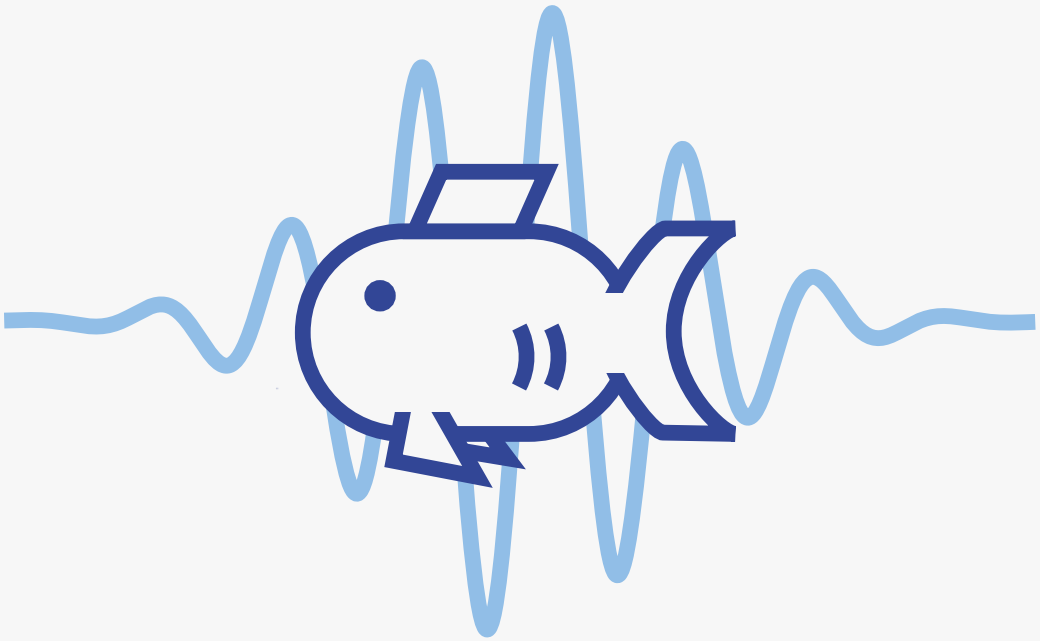

TUNA

Version 0.12.0

November 2026

Harry Brough

Foreword

Welcome to TUNA!

This is a project to deepen my understanding of the theory and implementation of quantum chemistry. A simple to use Python program with a command-line interface, TUNA can calculate energies and properties of diatomic molecules with a wide range of quantum chemical methods.

Unlike other Python-based programs for quantum chemistry such as Psi4 [1, 2] and PySCF [3], TUNA is not primarily a collection of modules to incorporate into a workflow, but is intended to be a program run directly from the command line. Parts of the syntax are inspired by the ORCA quantum chemistry package [4, 5] (although the tuna is the more majestic fish), but no input files are needed as the geometry of a diatomic molecule is easily specified in a single input line.

The key strength of TUNA is its low-friction, user-friendly interface, hopefully offering an intuitive experience that could serve as a teaching aid. Considerable effort has been taken to document all the code, and extensive examples of how the program works are provided in this manual. The program has grown into quite a large collection of quantum chemistry methods, showing their implementation in the simple yet instructive context of diatomic molecules.

On the technical side, TUNA utilises numerical energy derivatives, so any implemented electronic structure method can be used to calculate geometries, vibrational frequencies, response properties and molecular dynamics trajectories. Because these derivatives are fast for diatomics, almost nothing is written to the disk, and extensive use is made of vectorised operations in NumPy [6], TUNA is very fast for a Python program. Lots of use is made of the high symmetry of diatomics as well, speeding up calculations further.

While TUNA is currently fairly modest in scope, its capabilities are set to expand. The long term goal is to develop it into a simple and efficient testing ground for the performance of density functional approximations on diatomics as well as a capable tool for quantum chemistry education.

Who knows? Maybe one day, TUNA could make waves!

Harry Brough

Contents

1 Getting Started **6**

1.1 Installation 6

1.2 Simple Input Line 6

1.3 Units and Constants 7

1.4 Philosophy of TUNA 8

1.5 Program Components 8

2 Simple Input Line Examples **10**

3 Calculation Types **13**

3.1 Single Point Energy 14

3.1.1 Koopmans' Theorem 16

3.1.2 Rotational Constant 17

3.1.3 Multipole Moments 17

3.1.4 Polarisability and Hyperpolarisability 20

3.1.5 Population Analysis 21

3.1.6 Natural Orbitals 22

3.1.7 Plotting Orbitals and Densities 23

3.1.8 Dispersion Correction 25

3.2 Coordinate Scan 27

3.2.1 Plotting Coordinate Scans 28

3.3 Geometry Optimisation 30

3.3.1 Optimisation Convergence 32

3.3.2 Force Calculations 33

3.4 Harmonic Frequency 33

3.4.1 Transition Intensity 35

3.4.2 Thermochemistry . . . 36
3.4.3 Vibrational Perturbation Theory . . . 37
3.5 Anharmonic Frequency . . . 39
3.5.1 Transition Intensity . . . 41
3.5.2 Plotting Nuclear Wavefunctions . . . 43
3.6 Bond Dissociation Energy . . . 43
3.7 Ionisation Potential and Electron Affinity . . . 44
3.8 *Ab Initio* Molecular Dynamics . . . 45

4 Electronic Structure Methods **48**

4.1 Hartree–Fock Theory . . . 50
4.1.1 Initial Guess . . . 52
4.1.2 Symmetry Breaking . . . 53
4.1.3 Stability Analysis . . . 54
4.1.4 Self-consistent Field Convergence . . . 55
4.1.5 Hartree Theory . . . 57
4.2 Many-body Perturbation Theory . . . 57
4.2.1 Conventional MP2, MP3 and MP4 . . . 58
4.2.2 Freezing Core Electrons . . . 60
4.2.3 Spin-component Scaling . . . 60
4.2.4 Orbital-optimised MP2 . . . 61
4.2.5 Iterative MP2 . . . 61
4.2.6 Laplace Transform AO-MP2 . . . 62
4.2.7 Unrelaxed and Relaxed Density Matrices . . . 63
4.2.8 The GW Approximation . . . 65
4.2.9 Random Phase Approximation . . . 66
4.3 Configuration Interaction . . . 67
4.3.1 Full Configuration Interaction . . . 67
4.3.2 Configuration Interaction Singles and Doubles . . . 68
4.3.3 Configuration Interaction Doubles . . . 69
4.3.4 Configuration Interaction Singles, Doubles and Triples . . . 70
4.3.5 Complete Active Space Configuration Interaction . . . 70
4.3.6 Complete Active Space Self-consistent Field . . . 70

4.4 Coupled Cluster Theory . . . 71
4.4.1 Coupled Cluster Doubles . . . 74
4.4.2 Coupled Cluster Singles and Doubles . . . 74
4.4.3 Coupled Cluster Singles, Doubles and Triples . . . 75
4.4.4 Coupled Cluster Singles, Doubles, Triples and Quadruples . . . 75
4.4.5 Coupled Cluster with Perturbative Triples . . . 75
4.4.6 Coupled Cluster with Perturbative Quadruples . . . 76
4.4.7 Linearised Coupled Cluster . . . 76
4.4.8 Quadratic Configuration Interaction . . . 77
4.4.9 Approximate Coupled Cluster . . . 77
4.4.10 Linearised Density Matrix . . . 78
4.5 Density Functional Theory . . . 78
4.5.1 Local Spin Density Approximation . . . 82
4.5.2 Generalised Gradient Approximation . . . 83
4.5.3 *meta*-Generalised Gradient Approximation . . . 83
4.5.4 Hybrid Functionals . . . 85
4.5.5 Double-hybrid Functionals . . . 87
4.5.6 Efficient Grid Integration . . . 88
4.6 Excited States . . . 89
4.6.1 Time-dependent Hartree–Fock . . . 91
4.6.2 Configuration Interaction Singles . . . 91
4.6.3 Perturbative Doubles Correction . . . 91
4.6.4 Time-dependent Density Functional Theory . . . 92
4.6.5 Equation of Motion Coupled Cluster Theory . . . 93
4.6.6 Complete Active Space Methods . . . 94
4.6.7 Plotting Absorbance Spectra . . . 94

5 Basis Sets **96**

5.1 Basis Set Extrapolation . . . 96
5.2 Custom Basis Sets . . . 97
5.3 List of Basis Sets . . . 99

6 Molecular Integrals **105**

6.1 Types of Integrals 106
6.2 Spherical and Cartesian Harmonics 106

7 Performance of TUNA **108**

7.1 Speed and Efficiency 108
7.2 Memory Requirements 109
7.3 Accuracy and Precision 110

8 Acknowledgements **111**

9 List of Keywords **112**

Bibliography **125**

1 Getting Started

1.1 Installation

First ensure Python 3.12 or higher is installed, then run the following command from a terminal:

```
python3 -m pip install --upgrade quantumtuna
```

It is also possible to install TUNA by cloning the GitHub repository, but this is not recommended as you will have to compile the Cython integrals module yourself. Installing with pip will also install TUNA's three dependencies — NumPy [6], SciPy [7] and Matplotlib [8].

By default, TUNA will attempt to add itself to the system path on install. To check this has worked, open a new terminal and test that everything is working correctly by running:

```
TUNA --version
```

If the version of TUNA you installed prints to the terminal, you're good to go!

If this didn't work, locate the folder where TUNA was installed within the Python site packages, `/*/TUNA/`, where `*` is the path to TUNA, add it to the system path, then try again.

1.2 Simple Input Line

All calculations in TUNA are requested with a single line of commands in the terminal. The input line is not case sensitive and the spaces around colons are optional. It has the structure:

```
TUNA [Calculation] : [Atom A] [Atom B] [Distance] : [Method] [Basis Set]
```

A single atom can be provided, in which case `[Atom B]` and `[Distance]` are omitted. If two atoms are provided, a `[Distance]` must be given too. Available calculation types in TUNA are detailed in chapter 3, electronic structure methods in chapter 4 and basis sets in chapter 5.

For example, a single point energy calculation on an H_2 molecule with a bond length of 1.0 Å, with restricted Hartree–Fock theory in the STO-3G basis set is requested by:

```
TUNA SPE : H H 1.0 : HF STO-3G
```

Additional parameters can be requested by adding another colon at the end and appending keywords. For example, a B3LYP/6-31G* optimisation calculation using D2 semi-empirical dispersion correction on a H–He^+ molecule starting at a bond length of 0.8 Å is:

```
TUNA OPT : H He 0.8 : B3LYP 6-31G* : CHARGE +1 D2
```

As many keywords as you like can be written after the third colon. Some keywords such as `CHARGE` require a value (like +1 here) — just write this directly after the keyword, and then more keywords can be added. Many keywords can be used together in this way, and TUNA will figure out what you want. Further input examples to test out TUNA can be found in chapter 2.

1.3 Units and Constants

In the TUNA output, distances are printed in angströms, times in femtoseconds and everything else in atomic units unless otherwise stated. Units are set by the CODATA 2022 recommendations [9], in Table 1.1. All other constants are derived from these and the atomic masses in Table 3.3.

Table 1.1 Fundamental constants used in TUNA

Constant	Symbol	Value
Planck constant	h	$6.62607015 \times 10^{-34}$ J s
Elementary charge	e	$1.602176634 \times 10^{-19}$ C
Electron mass	m_e	$9.1093837139 \times 10^{-31}$ kg
Vacuum permittivity	ε_0	$8.8541878188 \times 10^{-12}$ F m^{-1}
Speed of light	c	2.99792458×10^{8} m s^{-1}
Boltzmann constant	k_B	1.380649×10^{-23} J K^{-1}
Avogadro constant	N_A	$6.02214076 \times 10^{23}$ mol^{-1}

1.4 Philosophy of TUNA

Numerical differentiation of the molecular energy underpins property calculations in TUNA. For atoms and diatomics it is very efficient, and it is also completely general, so:

any electronic structure method can be combined with **any** calculation type

This makes unusual calculations straightforward and consistent. For example, a coupled cluster hyperpolarisability calculation can, in principle, be treated with the same precision as a Hartree–Fock calculation. It also means that once a new method is implemented, it automatically becomes available across the program, rapidly expanding the capabilities of TUNA.

For numerical first derivatives, for instance used in geometry optimisations for the force and dipole moment derivatives for harmonic vibrational intensities, the central differences formula is used,

$$f'(x) \approx \frac{1}{2h}\left[f(x+h) - f(x-h)\right] .$$

For numerical second derivatives, used in harmonic frequency or polarisability calculations, a five point stencil formula is used,

$$f''(x) \approx \frac{1}{12h^2}\left[-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)\right] .$$

A drawback of numerical derivatives is their sensitivity to the finite difference step size, h. This value has been optimised for each derivative type in TUNA, and can be found in section 7.3. Numerical derivatives also require well converged energies. For calculations involving first derivatives, `TIGHT` energy convergence criteria are used by default, while higher derivatives use `EXTREME` convergence criteria. Even the standard thresholds in TUNA are tighter than in most programs.

A major advantage of TUNA's numerical philosophy is that these derivative formulae can be combined with basis set extrapolation, allowing approximate complete basis set geometries, vibrational frequencies, electric properties and molecular dynamics trajectories to be calculated.

1.5 Program Components

When TUNA is called by the terminal, the main program, `tuna.py`, begins a calculation in which it calls various modules, listed in Table 1.2. The most "fundamental" module is tuna_util, which contains all the defining constants for the program. The tuna_integral is unique as it is written in Cython [10], which is a compiled extension language to Python. This is because of the difficulty in vectorising the McMurchie–Davidson molecular integrals over basis functions.

Additionally, some external modules are used which are installed when `pip install quantumtuna` is run. The most important are NumPy [6] and SciPy [7], which speed up the code enormously compared to pure Python, while Matplotlib [8] enables the creation of two-dimensional plots.

Table 1.2 List of Python modules that constitute TUNA, in alphabetical order

Module	Description
tuna	Main program, input parsing
tuna_basis	Storing and setting up basis sets
tuna_calc	Processing keywords and storing calculation information
tuna_cas	Complete active space methods
tuna_cc	Coupled cluster
tuna_ci	Configuration interaction and excited states
tuna_dft	Density functional theory
tuna_energy	Calculating energies, building molecules, coordinate scans
tuna_freq	Harmonic and anharmonic frequencies
tuna_guess	Initial guess density
tuna_integral	Evaluating one- and two-electron integrals
tuna_kernels	Exchange–correlation kernels
tuna_md	*Ab initio* molecular dynamics
tuna_misc	Low level functions for energy evaluation
tuna_molecule	Storing and parsing molecular information
tuna_mbpt	Many-body perturbation theory
tuna_opt	Calculating gradients, Hessians and optimisation algorithm
tuna_out	Graphical plots and trajectories
tuna_props	Calculating and printing molecular properties
tuna_scf	Main self-consistent field loop, convergence acceleration
tuna_thermo	Thermochemistry after frequency calculations
tuna_util	General utility, units, frequently used functions
tuna_xc	Exchange–correlation functionals

2 Simple Input Line Examples

Several examples of simple input lines are shown here. These can be simply copied and pasted into a terminal window to see TUNA in action.

This should give a clear idea of the syntax to run a calculation, and then you can experiment!

Firstly, an energy calculation on the hydrogen atom with Hartree–Fock theory and a triple-ζ basis:

```
TUNA SPE : H : HF cc-pVTZ
```

Next, a second-order perturbation theory geometry optimisation of lithium hydride, with D2 semi-empirical dispersion correction and a polarised basis set:

```
TUNA OPT : Li H 1.7 : MP2 6-31+G* : D2
```

A single point DFT calculation on dinitrogen, with a decontracted basis and additional print:

```
TUNA SPE : N N 1.10 : B3LYP STO-3G : DECONTRACT P
```

A fourth-order perturbation theory vibrational frequency calculation on carbon monoxide, with 50% static damping and no Fock matrix extrapolation:

```
TUNA FREQ : C O 1.230 : MP4 3-21G : DAMP 0.5 NODIIS
```

An optimisation and subsequent frequency calculation on dihydrogen, using complete active space self-consistent field with extreme geometry convergence criteria:

```
TUNA OPTFREQ : H H 0.74 : CASSCF def2-TZVP : EXTREMEOPT NORB 2 NEL 2
```

A CCSDT calculation on a lithium atom, with reduced maximum coupled cluster iterations:

```
TUNA SPE : Li : CCSDT cc-pVDZ : CORRMAXITER 10
```

An orbital-optimised MP2 molecular dynamics simulation of dihydrogen, with a timestep of 0.1 fs and an initial temperature of 298.15 K, printing a trajectory to "tunatraj.xyz":

```
TUNA MD : H H 1.0 : OMP2 6-31G[d] : STEP 0.1 TEMP 298.15 TRAJ TUNATRAJ.XYZ
```

A single point calculation on a hydrogen atom using a ghost lithium basis set:

```
TUNA SPE : H XLi 0.50 : HF 6-311++G :
```

A full configuration interaction polarisability calculation on dihydrogen, using a diffuse basis set, plotting the second molecular orbital:

```
TUNA SPE : H H 0.74 : CISD t-aug-cc-pVDZ : POLAR PLOTMO 2
```

A geometry optimisation of triplet oxygen in a finite electric field:

```
TUNA OPT : O O 1.0 : SCS-MP3 3-21G : ML 3 EX 0.05 EY 0.06 EZ 0.10
```

A configuration interaction singles calculation that plots an excited state difference density:

```
TUNA SPE : C H 1.0 : CIS cc-pVDZ : DIFFDENSPLOT ROOT 3
```

A coupled cluster optimisation with single, double, triple and perturbative quadruple excitations on lithium hydride, freezing the core molecular orbital:

```
TUNA OPT : Li H 1 : CCSDT[Q] STO-3G : FREEZECORE
```

An anharmonic frequency calculation on dihydrogen, where the vibrational wavefunctions are plotted along with the potential energy surface when the calculation has converged:

```
TUNA ANHARM : H H 0.74 : CCSD def2-SVP : VIBPLOT
```

An MP2 atomic calculation on the magnesium atom extrapolated to the complete basis set limit:

```
TUNA SPE : Mg : MP2 def2-TZVPP : EXTRAPOLATE
```

An excited state optimisation and frequency calculation with a perturbative doubles correction:

```
TUNA OPTFREQ : H He 1.0 : CIS[D] 3-21G : ROOT 6
```

A full configuration interaction quadrupole moment calculation on the carbon atom:

```
TUNA SPE : C : FCI STO-6G : QUADRUPOLE
```

A *meta*-GGA excited state calculation in the Tamm–Dancoff approximation, plotting the spectrum:

```
TUNA SPE : Li H 1.2 : r2SCAN def2-SVP : TD TDA ABSPLOT
```

A double-hybrid electron affinity and population analysis using the relaxed density matrix:

```
TUNA EA : C C 1.2 : B2PLYP PC-0 : RELAXED
```

An equation of motion coupled cluster excited state single point calculation, plotting the absorbance spectrum:

```
TUNA SPE : H H 1.0 : EOM-CCSD 6-311G : ABSPLOT
```

A random phase approximation (on a generalised gradient approximaton DFT reference) perturbative anharmonic frequency calculation on hydrogen chloride:

```
TUNA FREQ : H Cl 1.0 : PBE cc-pVTZ : RPA VPT2
```

An ionisation potential calculation on dinitrogen with a hybrid exchange–correlation functional consisting of 50% Fock exchange:

```
TUNA IP : N N 1.0 : B97 d-aug-cc-pVDZ : HFX 0.5
```

An anharmonic frequency calculation with complete active space configuration interaction:

```
TUNA ANHARM : N N 1.0 : CASCI def2-SVP : NORB 4 NEL 2
```

Finally, a full configuration interaction polarisability calculation on the helium dimer, with increased damping for cluster amplitude convergence:

```
TUNA SPE : He He 2.0 : CCSDTQ cc-pVDZ : POLAR CORRDAMP 0.1
```

3 Calculation Types

The calculation types available in TUNA are listed in Table 3.1 with their respective keywords.

Table 3.1 Calculation types available in TUNA

Keyword	Calculation type
SPE	Single point energy
SCAN	Coordinate scan
OPT	Optimisation
FORCE	Force
FREQ	Harmonic frequency
OPTFREQ	Optimisation and subsequent harmonic frequency
ANHARM	Anharmonic frequency
BDE	Bond dissociation energy
IP	Ionisation potential
EA	Electron affinity
MD	*Ab initio* molecular dynamics

All calculation types run on an initial molecule, defined by its atoms, charge and multiplicity. Element types from hydrogen to argon are implemented, and are requested by their atomic symbols.

"Ghost" atoms are also available, which act as additional basis functions for a single atom and can be used to correct basis set superposition error [11]. The ghost atoms are accessible by typing an "X" before the atomic symbol, like "XH" or "XCl". For example, the following command will calculate the energy of a hydrogen atom in the presence of a ghost lithium atom's basis functions, lowering the energy compared to the calculation on a lone atom:

```
TUNA SPE : H XLi 0.735 : HF 6-311++G
```

Trying to run a calculation for a molecule (eg. `OPT`, `FREQ`, `MD`) with a ghost atom present will result in TUNA crashing politely with an error message.

Atoms or molecules can be given a charge and multiplicity with the `CHARGE` and `MULTIPLICITY` keywords. For example, triplet H_2 can be requested by:

```
TUNA SPE : H H 1.3 : UHF 6-311G : CHARGE 0 MULTIPLICITY 3
```

The shorter terms for charge and multiplicity `CH` and `ML` can also be used:

```
TUNA SPE : H H 1.3 : UHF 6-311G : CH 0 ML 3
```

If no charge or multiplicity is specified, TUNA defaults to a neutral molecule, which is assumed to be either a singlet or a doublet depending on whether the number of electrons is even or odd.

All calculations start by printing the TUNA logo to the terminal, importing libraries and printing the requested calculation type, electronic structure method and basis set. The molecule is then set up as requested, and the molecular structure, number of basis functions, number of primitive Gaussians, charge, multiplicity, number of (alpha and beta) electrons, number of occupied and virtual orbitals, point group and bond length are printed. The specified calculation then begins.

At the end of any calculation — if it terminates successfully — TUNA prints the total time taken for the calculation. This time is counted from when the Python modules have been imported.

More information about the times for each part of a TUNA calculation (molecular integrals, SCF iterations, correlation, etc.) is shown when the additional print keyword, `P`, is used.

3.1 Single Point Energy

A single point energy calculation in TUNA can be requested with the `SPE` calculation type keyword.

For instance, a single point calculation on H–He^+ at a distance of 2.0 Å with restricted Hartree–Fock theory in the 6-311G basis can be requested by:

```
TUNA SPE : H He 2.0 : RHF 6-311G : CH 1
```

The molecule is first set up, and information printed to the console:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Molecule and Basis Information
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Molecular structure:                                  H --- He

 Number of basis functions:                                   6
 Number of primitive Gaussians:                              10

 Charge:                                                      1
 Multiplicity:                                                1

 Number of electrons:                                         2
 Number of alpha electrons:                                   1
 Number of beta electrons:                                    1

 Number of occupied orbitals:                                 1
 Number of virtual orbitals:                                  5

 Point group:                                              Cinfv
 Bond length:                                            2.00000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

The nuclear repulsion energy is then calculated, which for diatomics is simply

$$V_{\text{nn}} = \frac{Z_{\text{A}} Z_{\text{B}}}{R} \,, \tag{3.1}$$

where Z is the relative nuclear charge and R is the bond length. Next the optional dispersion energy is calculated, before the one- and two-electron molecular integrals are evaluated. The eigenvalues of the overlap matrix, $\mathbf{S}$, are then checked to make sure they're not too small and avoid a linearly dependent basis set. The default threshold is 10^{-7} and can be adjusted with the `STHRESH` keyword. The Fock orthogonalisation matrix, $\mathbf{X} = \mathbf{S}^{-\frac{1}{2}}$, is formed, then the guess density matrix is calculated.

If a multi-electron calculation is requested, the convergence accelerators are printed and the self-consistent field cycle begins. If this converges, the components of the energy are printed in addition to the Virial ratio. This is the ratio between the molecular potential and kinetic energy, and should approach 2.0 as the molecular structure approaches an equilibrium geometry or for an atom. The spin contamination for spin-unrestricted calculations is printed. If a correlated wavefunction or

excited state calculation has been requested, this then begins.

When the final energy has been calculated, the one-electron reduced density matrix, $\mathbf{P}$, is used in property calculations. In the current version, the (un)restricted Hartree–Fock, unrelaxed CIS, (un)relaxed (spin-component scaled) MP2, linearised coupled cluster and full configuration interaction density matrices are available in TUNA. The orbital-optimised MP2 density matrix is also implemented, for which there is no difference between an unrelaxed and a relaxed density. If a higher order perturbation theory calculation is requested, properties will be calculated at the MP2 level. Calculations of rotational constants, multipole moments and population analysis are disabled for single atoms.

Finally, the energies are printed to the console again. If the additional print keyword, `P`, is used, molecular orbitals and their eigenvalues will also be printed, separated into α and β orbitals for spin-unrestricted calculations. Only the first ten virtual orbitals are printed by default, which can be overridden with the `PRINTMOS` keyword. If the reduced print keyword, `T`, is used, only the molecular information, SCF cycles and final energies are printed. If the `DEBUG` keyword is used, loads of information is printed throughout a calculation.

3.1.1 Koopmans' Theorem

Koopmans' theorem states that in closed-shell Hartree–Fock theory, the ionisation potential is equal to the negative of the orbital energy of the highest occupied molecular orbital (HOMO), $-\varepsilon_{\text{HOMO}}$, and that the electron affinity is equal to the negative of the orbital energy of the lowest unoccupied molecular orbital (LUMO), $-\varepsilon_{\text{LUMO}}$ [12]. This is exact within restricted Hartree–Fock (RHF) theory if the ionic orbitals are assumed to be identical to the orbitals of the atom.

The first properties printed in a TUNA calculation on a RHF reference are the Koopmans' parameters $-\varepsilon_{\text{HOMO}}$, $-\varepsilon_{\text{LUMO}}$ and the HOMO–LUMO gap, $\varepsilon_{\text{LUMO}} - \varepsilon_{\text{HOMO}}$. These values are not printed for calculations with unrestricted (UHF) references. They are also printed for restricted Kohn–Sham density functional theory (DFT) calculations, although the results are theoretically less meaningful due to the absence of an exact exchange–correlation functional.

The "exact" ionisation potential or electron affinity also accounts for orbital relaxation, and can be calculated from two converged self-consistent field calculations (for vertical ionisations) or from two converged geometry optimisations (for adiabatic ionisations). These calculations can be performed easily using the `IP` or `EA` calculation types, which are discussed in detail in section 3.7.

3.1.2 Rotational Constant

The next property printed, in units of GHz and cm^{-1}, is the rotational constant, calculated by

$$B = \frac{1}{4\pi\mu R^2} \, , \tag{3.2}$$

where μ is the reduced mass of the molecule, using the masses in section 3.4.

3.1.3 Multipole Moments

In the presence of a perturbative electric field, $\mathbf{F}$, the energy can be expanded as

$$E(\mathbf{F}) = E - \sum_i \mu_i F_i - \frac{1}{2}\sum_{ij} \alpha_{ij} F_i F_j - \frac{1}{6}\sum_{ijk} \beta_{ijk} F_i F_j F_k + \cdots \, , \tag{3.3}$$

where the dipole moment can be identified with the first derivative of the energy with respect to the field, polarisability with the second derivative and hyperpolarisability with the third.

The origin for dipole moment calculations (which is printed) is the centre of mass, R_0, calculated with the masses in section 3.4. The nuclear dipole moment,

$$\mu_{\mathsf{nuc}} = Z_{\mathsf{A}}(R_{\mathsf{A}} - R_0) + Z_{\mathsf{B}}(R_{\mathsf{B}} - R_0) \, , \tag{3.4}$$

and the analytical electronic dipole moment,

$$\mu_{\mathsf{ele}} = -\sum_{\mu\nu} P_{\mu\nu} D^z_{\mu\nu} \, , \tag{3.5}$$

are calculated, where $D^\alpha_{\mu\nu}$ are the dipole moment integrals, $\langle\mu|\mathbf{r} - \mathbf{R}_0|\nu\rangle$, evaluated with the rest of the one-electron integrals at the start of a calculation. Note that these dipole moments are one-dimensional by the symmetry of a diatomic molecule.

In addition to these values and the total dipole moment, $\mu = \mu_{\mathsf{nuc}} + \mu_{\mathsf{ele}}$, a diagram of the molecular structure is printed with an arrow pointing in the direction of the negative charge. For example, a RHF/6-311G calculation on H–He^+ at 2 Å gives the following output, meaning the hydrogen atom is more negatively charged than the helium atom.

This can be reproduced with the following input line:

```
TUNA SPE : H He 2.0 : HF 6-311G : CH 1
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        Dipole Moment
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Nuclear:  -1.4987929     Electronic:  -1.4774574

 Total:  -2.9762504           H --- He  <---+
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

The analytical dipole moment as described above is implemented for Hartree–Fock, pure and hybrid density functional theory, (un)relaxed MP2, unrelaxed CIS and linearised coupled cluster and configuration interaction density matrices. The numerical dipole moment can be calculated for all methods by taking the derivative of the energy with respect to the z component of an applied electric field,

$$\mu_{\text{ele}} = \frac{\partial E}{\partial F} \,. \tag{3.6}$$

This is requested with the `DIPOLE` keyword. For methods with a one-electron density matrix implemented, this will give the same result as the version automatically printed after the SCF calculation. However this dipole moment will be correct for all methods, where the analytical one may not be — for example the analytical MP2 dipole moment with the unrelaxed density matrix.

```
TUNA SPE : C O 1.0 : MP2 cc-pVDZ : DIPOLE
```

The quadrupole moment is also calculated analytically by default, and numerically with the `QUADRUPOLE` keyword. Just as for the dipole moment, these results are identical if an exact analytical density matrix is implemented in TUNA, but otherwise different, with the numerical results being correct.

The nuclear quadrupole moment is

$$Q_{\text{nuc}} = Z_{\text{A}}(R_{\text{A}} - R_0)^2 + Z_{\text{B}}(R_{\text{B}} - R_0)^2 \tag{3.7}$$

and the analytical electronic quadrupole moment is a rank-two tensor, with just two independent components for a diatomic aligned on the z axis,

$$Q^{\alpha}_{\text{ele}} = -\sum_{\mu\nu} P_{\mu\nu} Q^{\alpha}_{\mu\nu} \,, \tag{3.8}$$

where α is either xx or zz and $Q_{\mu\nu}$ are the quadrupole moment integrals over basis functions,

$\langle\mu|(\mathbf{r}-\mathbf{R}_0)^2|\nu\rangle$. The total zz component of the quadrupole tensor is then

$$Q^{zz} = Q_{\text{nuc}} + Q^{zz}_{\text{ele}} \,. \tag{3.9}$$

The isotropic quadrupole is

$$Q_{\text{iso}} = \frac{1}{3}(2Q^{xx} + Q^{zz}) \tag{3.10}$$

and the anisotropic quadrupole is

$$Q_{\text{aniso}} = Q^{zz} - Q^{xx} \,. \tag{3.11}$$

Similarly to the dipole moment, the nuclear, isotropic and anisotropic components of the analytical quadrupole moment are printed at the energy of a calculation by default, with a diagram showing the orientation of the quadrupole, here that the outsides of the molecule are more positive:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    Quadrupole Moment
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Nuclear:  10.2716330   Anisotropic:   9.0759675

 Isotropic:   2.2337484      H --- He  +-> <-+
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

When the `QUADRUPOLE` keyword is used, the quadrupole components are calculated as derivatives of the energy with respect to an applied electric field gradient,

$$Q^{\alpha}_{\text{ele}} = \frac{\partial E}{\partial(\partial_\alpha F_\alpha)} \,. \tag{3.12}$$

This requires four additional energy evaluations, but recalculation of the molecular integrals can be skipped, speeding things up. The xx and zz electronic components of the quadrupole, as well as the total isotropic and anisotropic quadrupole, are printed.

```
TUNA SPE : C O 1.0 : MP2 cc-pVDZ : QUADRUPOLE
```

3.1.4 Polarisability and Hyperpolarisability

Another electric property calculated by TUNA is the polarisability, requested with the POLAR keyword. In general, there are nine components of the polarisability matrix which are

$$\alpha_{\alpha\beta} = \frac{\partial^2 E}{\partial F_\alpha \partial F_\beta} \,. \tag{3.13}$$

Due to the symmetry of a diatomic molecule, there are only two unique components of the dipole–dipole polarisability matrix so only two second derivatives are necessary — eight additional energy calculations in total. These give the parallel and perpendicular components of polarisability, then the isotropic polarisability is

$$\alpha_{\text{iso}} = \frac{1}{3}(\alpha_\perp + \alpha_\perp + \alpha_\parallel) \tag{3.14}$$

and the anisotropic polarisability is

$$\alpha_{\text{aniso}} = \alpha_\parallel - \alpha_\perp \,. \tag{3.15}$$

These are both printed (along with the parallel and perpendicular component with P) with the dipole moment during a polarisability calculation. The following input line calculates polarisability on the hydrogen molecule, giving a value of 5.2 a_0^3, close to the experimental value of 5.4 a_0^3 [13].

```
TUNA SPE : H H 0.741 : CCSD aug-cc-pVTZ : POLAR
```

The dipole–dipole–dipole hyperpolarisability can also be calculated in TUNA with the HYPER keyword, and in general has $3 \times 3 \times 3 = 27$ components since

$$\beta_{\alpha\beta\gamma} = \frac{\partial^3 E}{\partial F_\alpha \partial F_\beta \partial F_\gamma} \,. \tag{3.16}$$

For diatomics, like the polarisability there are only two independent components, the parallel and perpendicular components, β_{zzz} and β_{xxz}. This means twelve energy evaluations are necessary so the calculation isn't too much slower than for polarisability.

The hyperpolarisability of a $D_{\infty\text{h}}$ molecule is necessarily zero due to symmetry, but can be calculated anyway. A $C_{\infty\text{v}}$ molecule has two independent components. For example:

```
TUNA SPE : H He 1.0 : CCSD aug-cc-pVTZ : HYPER CH 1
```

Calculations can also be requested where an external electric field is applied to the molecule. This can be requested with the EX, EY and EZ for the Cartesian electric field components, in atomic

units. For example an optimisation in the field

$$\mathbf{F} = 0.05\,\hat{\mathbf{x}} + 0.06\,\hat{\mathbf{y}} + 0.10\,\hat{\mathbf{z}} \tag{3.17}$$

is requested by:

```
TUNA OPT : H H 0.74 : SCS-MP3 cc-pVTZ : EX 0.05 EY 0.06 EZ 0.10
```

Similarly, calculations in an applied electric field gradient are requested with `EGX`, `EGY` and `EGZ`.

3.1.5 Population Analysis

Towards the end of a calculation, TUNA prints Mulliken [14], Löwdin [15] and Mayer [16] population analysis information. Note that all the equations in this section are true for diatomics, not necessarily molecules in general. These calculations can be disabled with the `T` keyword, or requested after every geometry iteration in an optimisation with `P`.

The Mulliken analysis [14] is very widely used — despite its considerable weaknesses [17] — and partitions the electron density by the basis functions assigned to each atom. The number of electrons, N, is the integral of the density and can be calculated by

$$N = \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} \ . \tag{3.18}$$

When the density is partitioned equally between atoms, the number of electrons on atom A is

$$N_{\mathsf{A}} = \sum_{\mu\in\mathsf{A}} \sum_{\nu\in\mathsf{A}} P_{\mu\nu} S_{\mu\nu} + \frac{1}{2} B_{\mathsf{AB}} \ , \tag{3.19}$$

where B_{AB} is the Mulliken bond order, given by

$$B_{\mathsf{AB}} = 2 \sum_{\mu\in\mathsf{A}} \sum_{\nu\in\mathsf{B}} P_{\mu\nu} S_{\mu\nu} \ . \tag{3.20}$$

The partial charge of an atom in a molecule is then given by $Q_{\mathsf{A}} = Z_{\mathsf{A}} - N_{\mathsf{A}}$. In a population analysis, TUNA prints these charges and the Mulliken bond order, as well as the sum of the charges which should be equal to the requested molecular charge.

Because the molecular orbitals that generate a specific energy and density are not unique, population analysis can be performed with non-localised orthogonal basis functions in Löwdin analysis

[15] where the density matrix is rotated by

$$\mathbf{P}^{\mathsf{L}} = \mathbf{S}^{\frac{1}{2}}\mathbf{P}\mathbf{S}^{\frac{1}{2}} \ . \tag{3.21}$$

Using this density matrix, atomic populations are calculated by

$$N_{\mathsf{A}} = \sum_{\mu \in \mathsf{A}} P^{\mathsf{L}}_{\mu\mu} \ , \tag{3.22}$$

and the bond order is calculated by

$$B_{\mathsf{AB}} = \sum_{\mu \in \mathsf{A}} \sum_{\nu \in \mathsf{B}} P^{\mathsf{L}}_{\mu\nu} P^{\mathsf{L}}_{\nu\mu} \ . \tag{3.23}$$

Similarly to in Mulliken analysis, the Löwdin charges and bond order are printed.

The results of a Mayer analysis [16] are also printed. The Mayer charges are the same as the Mulliken charges, but the bond order (and bonded valence) is given by

$$B_{\mathsf{AB}} = \sum_{\mu \in \mathsf{A}} \sum_{\nu \in \mathsf{B}} (\mathbf{PS})_{\mu\nu}(\mathbf{PS})_{\nu\mu} + (\mathbf{RS})_{\mu\nu}(\mathbf{RS})_{\nu\mu} \ , \tag{3.24}$$

where $\mathbf{R}$ is the spin density matrix, $\mathbf{P}_\alpha - \mathbf{P}_\beta$. In Mayer analysis the valence of each atom is calculated. The total valence, which is similar to the expected valence of an atom (ie. one for hydrogen, three for boron), is

$$V_{\mathsf{A}} = 2N_{\mathsf{A}} - \sum_{\mu \in \mathsf{A}} \sum_{\nu \in \mathsf{B}} (\mathbf{PS})_{\mu\nu}(\mathbf{PS})_{\nu\mu} \tag{3.25}$$

and the free valence, which is a measure of the ability to form further bonds, is calculated by $F_{\mathsf{A}} = V_{\mathsf{A}} - B_{\mathsf{AB}}$. These valences are printed under the Mayer analysis heading.

3.1.6 Natural Orbitals

For UHF, second-order perturbation theory, coupled cluster methods, and full configuration interaction approaches natural orbitals can be calculated. These are the eigenfunctions of the one-electron reduced density matrix

$$\mathbf{P}\mathbf{\Lambda}_i = \lambda_i \mathbf{\Lambda}_i \ , \tag{3.26}$$

where λ_i represents the occupancy of the ith natural orbital, a value between 0.0 and 2.0. Natural orbitals provide a basis that converges most efficiently to the exact wavefunction [18], and the occupancies are useful for determining systems which might benefit from a multireference treatment,

beyond the single determinant model of Hartree–Fock theory. Occupancies that deviate strongly from 2.0 (occupied) or 0.0 (virtual) indicate natural orbitals with strong correlation.

The calculation of natural orbitals is disabled by default, but can be activated with the `NATORBS` keyword. A coupled cluster doubles calculation with natural orbitals is requested by:

```
TUNA SPE : H H 2.0 : CCD def2-SVPD : NATORBS
```

The sum of the natural orbital occupancies is also printed along with the orbital occupancies, along with the density matrix trace. These values should be equal to the total number of electrons. Natural orbital calculations are possible with both unrelaxed and relaxed density matrices for MP2, as well as for complete active space methods.

Using the `PRINTMOS` keyword, the occupancies and coefficients of each natural orbital can be seen:

```
TUNA SPE : H H 3.0 : UHF cc-pVDZ : PRINTMOS 5 NATORBS
```

3.1.7 Plotting Orbitals and Densities

The infrastructure that supports density functional theory calculations enables the expression of orbitals and electron densities onto a grid, which can be plotted by TUNA. While the grids for DFT integration are more dense near the nuclei and more sparse further away, the plotting grids are two-dimensional Cartesian grids — with size defined by the bond length. These two-dimensional grids are much more efficient to compute than three-dimensional "cube" files, and contain all the information necessary to describe a diatomic, due to these molecules' symmetry.

The simplest plot is the electron density, requested via the `DENSPLOT` keyword, which shows a magenta blob of the converged electron density:

```
TUNA SPE : H H 0.74 : HF STO-3G : DENSPLOT
```

For unrestricted calculations, the `SPINDENSPLOT` keyword plots the spin density. For example, the following calculation on the carbyne radical, shown in Figure 3.1, shows the majority of the spin density is localised on the carbon atom. Red indicates α spin density and blue shows β density.

```
TUNA SPE : C H 1.0 : HF cc-pVDZ : SPINDENSPLOT COREGUESS
```

Difference densities of the ith excited state from excited state calculations, calculated by

$$\mathbf{P}^{i}_{\text{diff}} = \mathbf{P}^{i}_{\text{excited}} - \mathbf{P}_{\text{HF}} \tag{3.27}$$

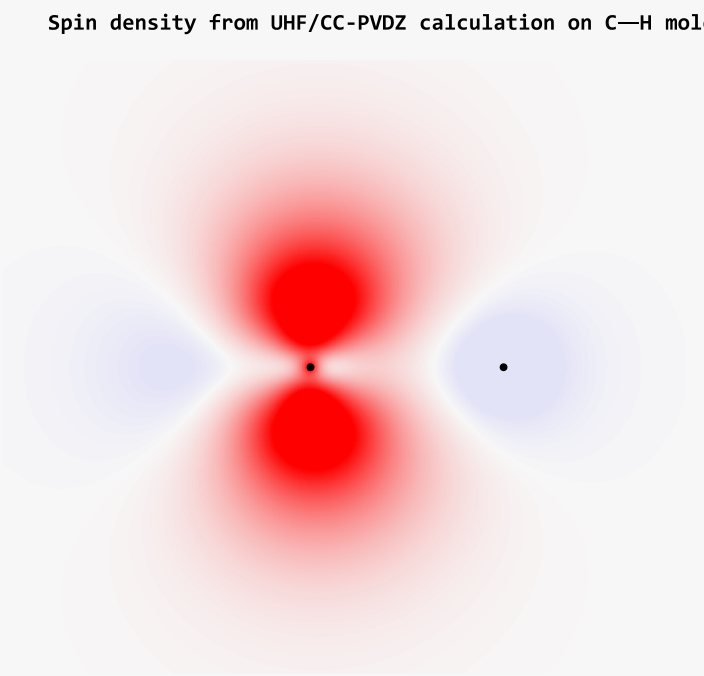

Figure 3.1: The spin density of carbyne, showing the vast majority of the spin density is found on the carbon atom, in the shape of a p orbital.

can be plotted with the `DIFFDENSPLOT` keyword. A similar equation calculates the difference spin density, given by the `DIFFSPINDENSPLOT` keyword. For example, the third excited state difference density of the carbyne radical, shown in Figure 3.2, is plotted by:

```
TUNA SPE : C H 1.0 : CIS cc-pVDZ : DIFFDENSPLOT ROOT 3
```

Areas of positive difference density, which have gained electrons in the excited state, are shown in red, and areas which have lost electrons are shown in blue.

The converged molecular orbitals from a Hartree–Fock or DFT calculation can also be plotted. The `PLOTMO` keyword is used to do this, along with the number of molecular orbital that is desired to be plotted. The orbital numbers can be checked with the additional print keyword, `P` or `PRINTMOS`. For unrestricted references, the α and β orbitals are interleaved. Figure 3.3 shows a virtual 4d orbital of a hydrogen atom, plotted by:

```
TUNA SPE : H : PBE cc-pVQZ : PLOTMO 29
```

The HOMO and LUMO can also be plotted with the `PLOTHOMO` and `PLOTLUMO` keywords, respectively. The following calculation shows the π^* orbital of the triplet oxygen molecule.

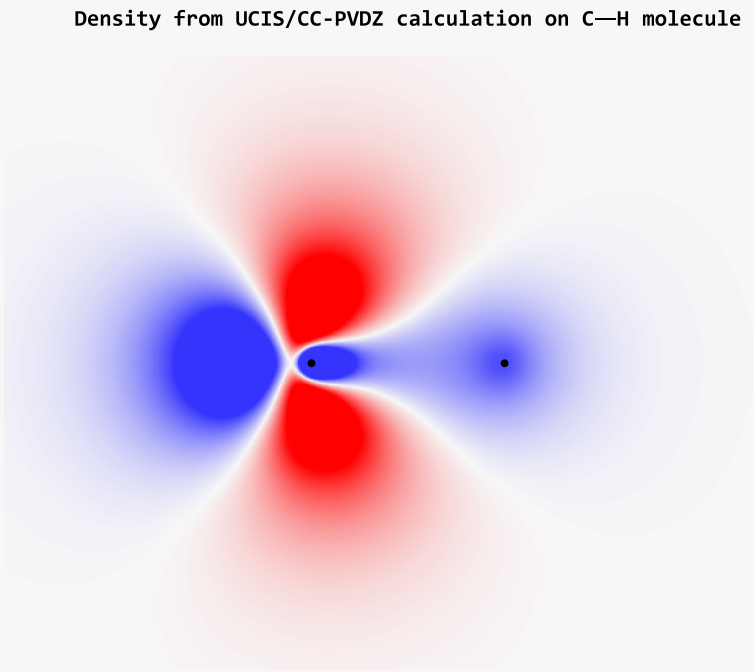

Figure 3.2: The difference density from an excited state of carbyne. This shows that the electron density has moved from around the lone pair on the carbon atom (deep blue), to the bond area.

```
TUNA SPE : O O 1.1 : B3LYP 6-311++G** : ML 3 PLOTHOMO
```

If natural orbitals have been calculated with `NATORBS` they can be plotted similarly, using the `PLOTNO` keyword. If no natural orbitals are found, the plot will just be blank.

```
TUNA SPE : C O 1.1 : CCSD cc-pVDZ : NATORBS PLOTNO 7
```

3.1.8 Dispersion Correction

The dispersion energy is long range and poorly described by most exchange–correlation functionals, so semi-empirical corrections are often used to improve the description of non-covalent interactions. There are two forms of dispersion correction available in TUNA: the atomic pairwise D2 scheme by Grimme [19], and the non-local VV10 method based on integrating the electron density [20].

The D2 correction is requested with the `D2` keyword:

```
TUNA SPE : H H 1.0 : PBE 6-311++G : D2
```

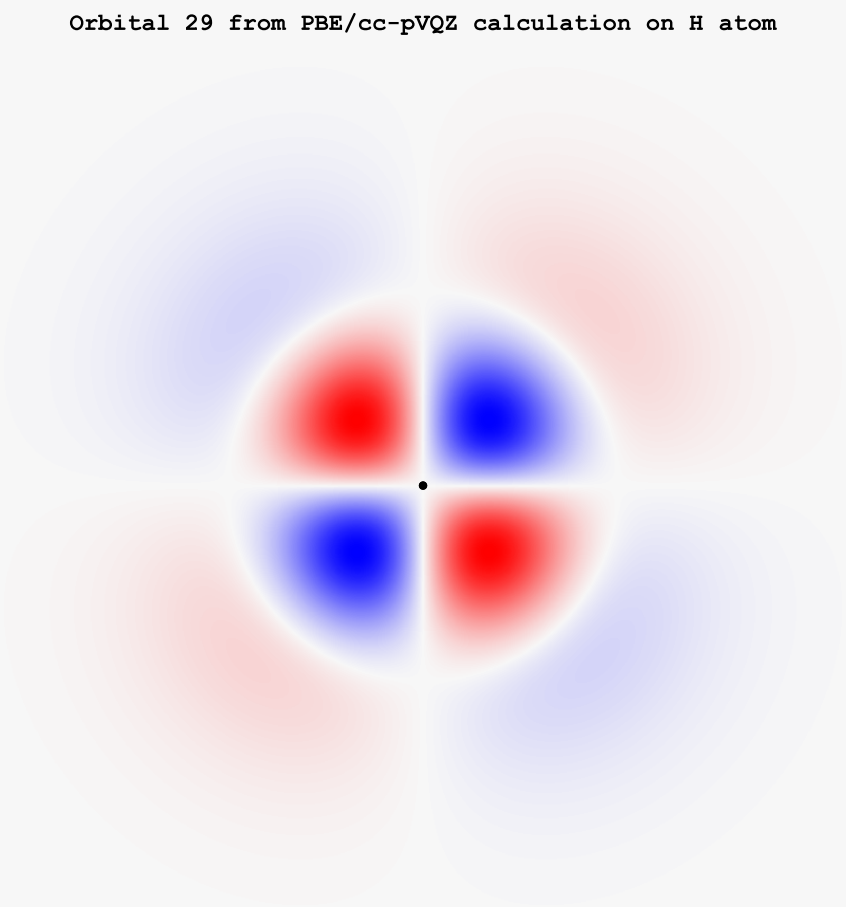

Figure 3.3: Using a large basis set like cc-pVQZ allows high-lying unoccupied orbitals to be visualised. This is a virtual 4d orbital for the hydrogen atom.

In this method, the dispersion energy is calculated by

$$E_{\text{disp}} = -\frac{s_6}{f_{\text{damp}}}\frac{C_6^{AB}}{R^6}\,, \tag{3.28}$$

where the damping function is given by

$$f_{\text{damp}} = 1 + \exp\left[-d\left(\frac{R}{R_{\text{VdW}}^{\text{AB}}} - 1\right)\right], \tag{3.29}$$

and $C_6^{\text{AB}} = \sqrt{C_6^{\text{A}} C_6^{\text{B}}}$ while $R_{\text{VdW}}^{\text{AB}} = R_{\text{VdW}}^{\text{A}} + R_{\text{VdW}}^{\text{B}}$.

The default value of the damping factor d is 20, and the s_6 value is 1.2 in line with the parameterisation of Hartree–Fock in ORCA [4, 5]. The values of the Van der Waals radii, R_{VdW}, and C_6

coefficients are element-specific. The optimised values of s_6 are implemented for various density functional theory methods [19]. These are PBE, BLYP, B3LYP, B2PLYP, mPW2PLYP, BP86 and TPSS. The s_6 value will print out to the console whenever the `P` keyword is used.

The non-local VV10 dispersion correction is requested with the `NL` keyword:

```
TUNA SPE : H H 1.0 : PBE 6-311++G : NL
```

In VV10, the dispersion energy is a non-local double integral,

$$E_{\text{disp}} = \int n(\mathbf{r}) \left[\beta + \frac{1}{2} \int \mathrm{d}\mathbf{r}' \, n(\mathbf{r}') \, \Phi(\mathbf{r}, \mathbf{r}') \right] \mathrm{d}\mathbf{r} \, , \tag{3.30}$$

where $n(\mathbf{r})$ is the electron density, β is an exchange–correlation functional-dependent parameter, and $\Phi(\mathbf{r}, \mathbf{r}')$ is known as the VV10 kernel.

In TUNA, the VV10 energy is computed at the end of the SCF loop, not self-consistently every iteration — this is not likely to make a meaningful difference to the final energy. The double integral is calculated with the same numerical quadrature used for density functional theory calculations, but is still very slow as the full non-local grid cannot be stored simultaneously in memory.

This method is parameterised for Hartree–Fock, and a number of density functional theory methods. These are PBE, RPBE, revPBE, PBE0, revPBE0, revPBE38, BLYP, B3LYP, B2PLYP, DSD-BLYP, BP86, TPSS, SCAN, rSCAN, r^{2}SCAN, TPSSh, TPSS0, r^{2}SCANh, r^{2}SCAN0, r^{2}SCAN50, PW1PW, B3PW91, B3P86, Pr^{2}SCAN50, Pr^{2}SCAN69. The B97M-V functional is also parameterised for VV10, and in that case the `NL` keyword does not need to be explicitly used:

```
TUNA SPE : H H 1.0 : B97M-V 6-311++G :
```

3.2 Coordinate Scan

A coordinate scan can be requested in TUNA with the `SCAN` calculation type. This calculates the potential energy surface between two points, increasing the interatomic distance at each step, using the chosen electronic structure method, basis set and miscellaneous keywords.

There are two mandatory parameters for a coordinate scan. The distance between steps must be specified by the `STEP` keyword, and the number of scan steps is specified by the `NUM` keyword. There are no default parameters here; these must be specified when a `SCAN` calculation is requested.

For instance, an MP2 scan over the potential energy surface of H_2, starting at a bond length of 0.3 Å and increasing with 20 steps of 0.1 Å can be requested by:

```
TUNA SCAN : H H 0.3 : MP2 6-31G : NUM 20 STEP 0.1
```

By default, the previous step's SCF density matrix is read in at each scan step to accelerate convergence; this can be disabled using the `NOMOREAD` keyword. For UHF calculations on singlet $D_{\infty h}$ molecules, the molecular orbitals are rotated after being read, unless the `NOROTATE` keyword is used. See subsection 4.1.1 for a discussion of the initial guess and `ROTATE` keywords.

Any of the keywords that can be used with a single point energy calculation (eg. `D2`, `SLOWCONV`, `DIPOLE`, `EXTRAPOLATE`) can also be combined with the coordinate scan calculation. At the end, a table of bond lengths and final energies is printed, which can be easily copied into a spreadsheet.

3.2.1 Plotting Coordinate Scans

Due to the Matplotlib [8] integration in TUNA, two-dimensional plots can easily be generated from coordinate scan calculations with the `SCANPLOT` keyword. The line is straight and blue by default, but can be changed to dashed using the keyword `DASH`, dotted using `DOT`, and the colour can be changed to any of `BLACK`, `WHITE`, `RED`, `BLUE`, `YELLOW`, `GREEN`, `CYAN`, and `MAGENTA`.

For example, a plot with a magenta dotted line over the UHF H_2 potential energy surface is:

```
TUNA SCAN : H H 0.3 : UHF 4-31G : NUM 40 STEP 0.04 SCANPLOT DOT MAGENTA
```

This produces the plot in Figure 3.4, with the legend showing the electronic structure method and basis set. Plots can be saved by the keyword `SAVEPLOT [filepath]`, for example `SAVEPLOT "MAGENTAPLOT.PDF"`. Standard raster and vector image extensions are supported.

Any plot colour can be chosen with the `COLOUR` keyword, followed by a hexadecimal hash:

```
TUNA SCAN : Li H 0.6 : HF def2-SVP : COLOUR #00FF00 NUM 40 STEP 0.1 SCANPLOT
```

Multiple calculations can be plotted on the same axes for easy comparison. To do this, use the keyword `ADDPLOT` with every calculation you want overlaid on the same axes. Then just run calculations one after the other. To clear the axes, use the `DELPLOT` keyword once. The `DELPLOT` keyword is parsed before `ADDPLOT`, so they can be used together to start from a blank canvas. The following calculations, run one after the other, produced Figure 3.5.

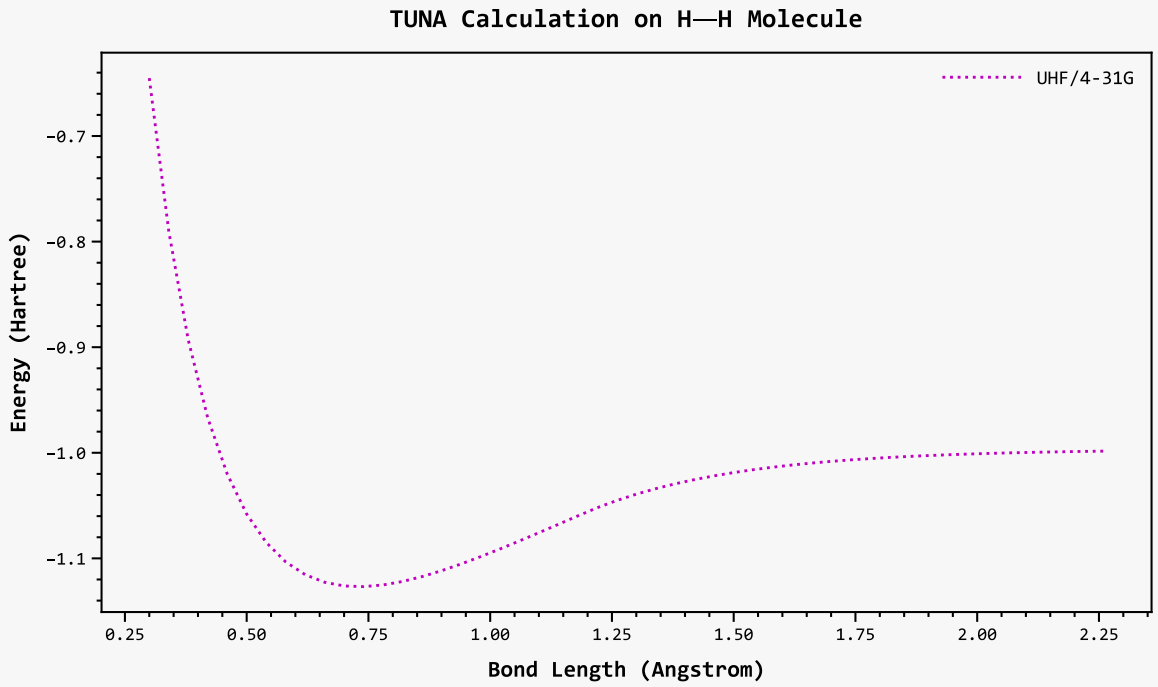

Figure 3.4: Potential energy surface of H_2 calculated in TUNA with UHF/4-31G

```
TUNA SCAN : H H 0.3 : HF 4-31G : NUM 50 STEP 0.04 SCANPLOT BLACK ADDPLOT
```

```
TUNA SCAN : H H 0.3 : UHF 4-31G : NUM 50 STEP 0.04 SCANPLOT BLACK DOT ADDPLOT
```

```
TUNA SCAN : H H 0.3 : MP2 4-31G : NUM 50 STEP 0.04 SCANPLOT BLUE ADDPLOT
```

```
TUNA SCAN : H H 0.3 : SCS-MP3 4-31G : NUM 50 STEP 0.04 SCANPLOT GREEN ADDPLOT
```

```
TUNA SCAN : H H 0.3 : CCSD 4-31G : NUM 50 STEP 0.04 SCANPLOT RED ADDPLOT
```

This kind of plotting can also be combined with excited state calculations to easily see the different potential energy surfaces. Note that TUNA does not currently have a root-following algorithm, so it's easy for potential energy surfaces to get mixed up part way through a coordinate scan.

The following calculations on the H–He molecule produced Figure 3.6. The ground state calculation shows a bond does not form, and the atoms dissociate. However absorbing a photon does allow a metastable molecule to form, with local minima found in the excited state surfaces. The sharp turn in the second excited surface indicates a crossing between two potential energy surfaces.

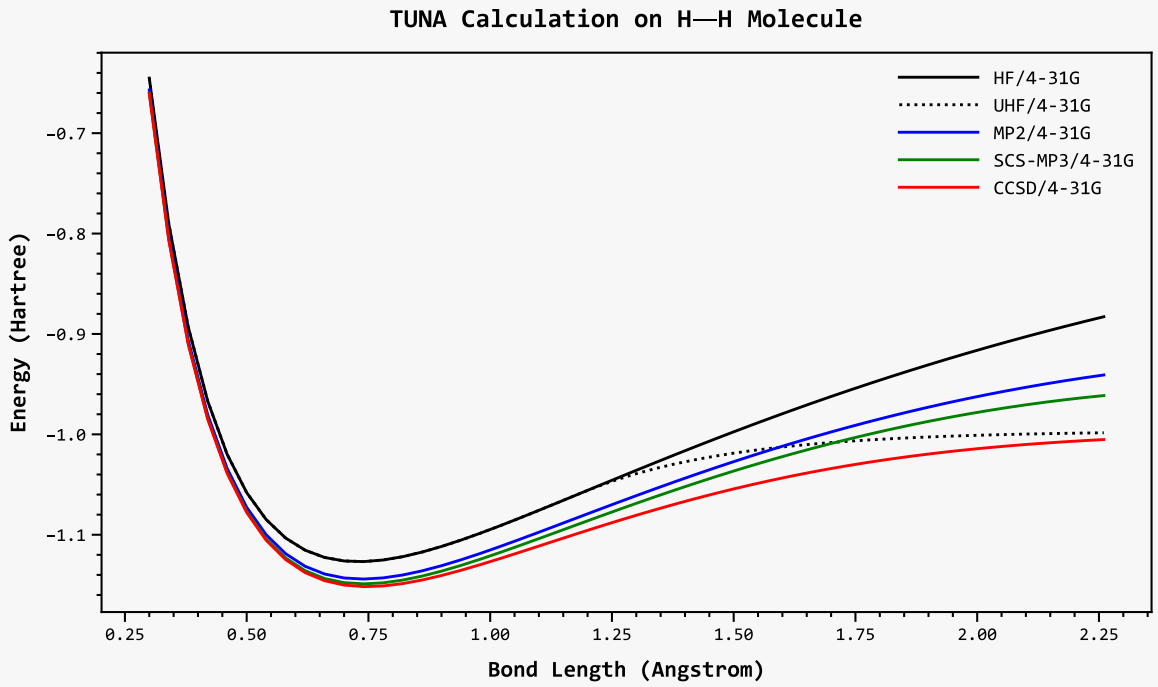

Figure 3.5: Potential energy surfaces of H_2 calculated in TUNA with various electronic structure methods in 4-31G basis set, overlaid using the ADDPLOT keyword and shown with SCANPLOT

```
TUNA SCAN : H He 0.3 : HF cc-pVTZ : NUM 50 STEP 0.025 SCANPLOT BLACK ADDPLOT
```

```
TUNA SCAN : H He 0.3 : CIS cc-pVTZ : NUM 50 STEP 0.025 SCANPLOT RED ADDPLOT
```

3.3 Geometry Optimisation

A geometry optimisation finds a stationary point on the potential energy surface, where

$$\frac{\partial E}{\partial R} = 0 \,, \tag{3.31}$$

and is requested by the OPT calculation type keyword:

```
TUNA OPT : H He 1.0 : RHF 6-311G : CH 1
```

At the start of an optimisation, the convergence criteria, maximum number of iterations and maximum allowed step are printed, before the calculation sets off. In each geometry iteration, the molecule is set up and the energy is calculated with the chosen method. Tight energy convergence criteria are used by default for optimisations and the SCF density matrix is read in from the previous

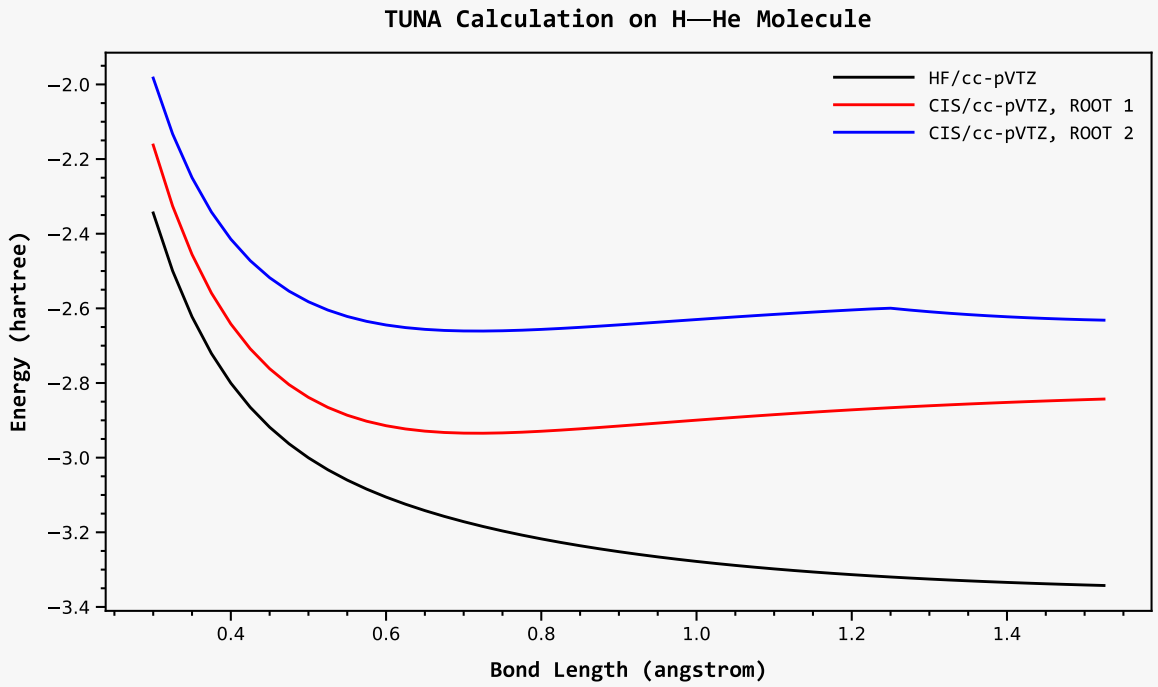

Figure 3.6: Potential energy surfaces of the H–He ground state and first two excited states calculated in TUNA with HF/cc-pVTZ and CIS/cc-pVTZ

optimisation step — this guess strategy can be disabled using the NOMOREAD keyword.

Next, the gradient is calculated using the central differences method, by

$$\nabla E_n = \frac{\partial E}{\partial R_n} = \frac{E(R_n + h) - E(R_n - h)}{2h} \,, \tag{3.32}$$

where R is the bond length and h is defined in section 7.3. Next, an approximation to the geometric Hessian is calculated by

$$H_n = \frac{\nabla E_n - \nabla E_{n-1}}{R_n - R_{n-1}} \,. \tag{3.33}$$

Alternatively, the exact Hessian can be calculated using the CALCHESS keyword. This increases calculation time at each step significantly, but tends to reduce the number of geometry iterations needed. For diatomics, the exact Hessian calculation is probably pointless unless the potential energy surface is very flat. For optimisations to a minimum, either the exact or approximate Hessian is used in the convex region of the potential energy surface, whereas for optimisations to maxima (using OPTMAX), these Hessians are used in the concave region. Beyond these regions, a default Hessian of $1/4$ is used. This value can be changed using the DEFAULTHESS keyword.

Table 3.2 Optimisation convergence criteria in TUNA

Convergence	Gradient / hartree bohr^{-1}	Step / Å
LOOSEOPT	10^{-3}	10^{-2}
MEDIUMOPT	10^{-4}	10^{-4}
TIGHTOPT	10^{-6}	10^{-5}
EXTREMEOPT	10^{-8}	10^{-7}

After the Hessian is determined, the bond length is updated by

$$R_{n+1} = R_n - \frac{\nabla E_n}{H_n} \,. \tag{3.34}$$

Note that for a one-dimensional potential energy surface there is no choice necessary about the update — this is the only method that uses derivatives up to second order.

The maximum value for the change in bond length is 0.2 Å by default, to prevent exploding optimisations from a bad starting guess geometry. This can be changed with the MAXSTEP keyword, giving the maximum allowed step in angstroms. The trajectory of a geometry optimisation can be printed to an .xyz file using the TRAJ keyword. The structures and energies at each point are printed to a file called "tuna-trajectory.xyz" in the directory of the terminal. An example of an optimisation with an exact Hessian, reduced maximum step and printed trajectory is:

```
TUNA OPT : H F 0.9 : HF 4-31G : CALCHESS MAXSTEP 0.1 TRAJ
```

3.3.1 Optimisation Convergence

Four optimisation convergence criteria keywords are available: LOOSEOPT, MEDIUMOPT, TIGHTOPT and EXTREMEOPT. Medium convergence is the default for standard optimisation calculations, while optimisation and frequency and anharmonic frequency calculations use tight default criteria.

The gradient and step convergence criteria of these are listed in Table 3.2. If the geometry has not converged to the chosen criteria within the maximum number of iterations, that maximum can be increased with the MAXGEOMITER keyword, which has a value of 30 by default.

When the gradient and step fall below the convergence criteria, the optimisation will terminate, and the final density will be used to calculate analytical properties. These can be calculated at every step with the additional print keyword, P. Finally, TUNA prints the bond length of the converged

geometry and the optimised energy. If a harmonic frequency calculation is desired immediately after the optimisation terminates, the `OPTFREQ` calculation type should be used instead of `OPT`.

3.3.2 Force Calculations

The `FORCE` keyword calculates the force on the current molecular geometry, without optimisation:

```
TUNA FORCE : H C 1.1 : mPW1LYP def2-SVP : CH 1
```

This calculates the energy and the gradient at the bond length and prints them to the terminal.

3.4 Harmonic Frequency

A harmonic frequency calculation can be requested on a molecule with the `FREQ` keyword:

```
TUNA FREQ : H H 0.7375 : RHF 6-311++G :
```

This can — and often should — be combined with a prior geometry optimisation, where the frequency calculation is run on the optimised structure, using the `OPTFREQ` calculation type. All the keywords available for the `OPT` calculation type are available for `OPTFREQ`.

```
TUNA OPTFREQ : H O 1.0 : CCSD[T] 6-31G* : CH -1
```

Harmonic frequency calculations in TUNA determine the numerical second derivative, which is the bond's force constant, k, (and the Hessian, H) by

$$k = \frac{\partial^2 E}{\partial R^2} \,. \tag{3.35}$$

The use of five single point calculations makes harmonic frequency calculations slow, but ensures the frequencies are numerically stable. For additional reliability, `EXTREME` SCF convergence criteria are used for frequency calculations by default, but by optimising the numerical step length, the frequencies have been found to be converged to 0.01 cm^{-1} even with `LOOSE` energy convergence.

After the second derivatives are determined, the vibrational frequency is calculated by

$$\omega = \sqrt{\frac{k}{\mu}} \,, \tag{3.36}$$

where μ is the reduced mass, calculated using the atomic masses in Table 3.3. These are the masses of the most abundant isotope. Custom masses can be used via the `M1` and `M2` keywords,

Table 3.3 Default atomic masses in TUNA, using data from [21, 22].

Atom	Major Isotope	Mass / amu
Hydrogen	^{1}H	1.007825
Helium	^{4}He	4.002603
Lithium	^{7}Li	7.016004
Beryllium	^{9}Be	9.012182
Boron	^{11}B	11.009305
Carbon	^{12}C	12.000000
Nitrogen	^{14}N	14.003074
Oxygen	^{16}O	15.994915
Fluorine	^{19}F	18.998403
Neon	^{20}Ne	19.992440
Sodium	^{23}Na	22.989770
Magnesium	^{24}Mg	23.985042
Aluminium	^{27}Al	26.981538
Silicon	^{28}Si	27.976927
Phosphorus	^{31}P	30.973762
Sulfur	^{32}S	31.972071
Chlorine	^{35}Cl	34.968853
Argon	^{40}Ar	39.962383

where the first and second atom's mass can be chosen. For example, this calculation gives the second atom — chlorine — a mass of 36.0. The atomic masses can be printed using `P`.

```
TUNA OPTFREQ : H Cl 1.30 : RHF 6-311++G : M2 36.0
```

The vibrational frequency, reduced mass and force constant are all printed to the console. This printed frequency has units of cm^{-1}. If k is negative, such as at a maximum on the potential energy surface, ω will be imaginary and TUNA will output the frequency prepended with an "`i`".

The harmonic frequencies of H_2 calculated from some different methods implemented in TUNA are shown in Table 3.4, where coupled cluster methods perform much better than perturbation theory and B3LYP does very well. These values were calculated using two-point basis set extrapolation with triple- and quadruple-ζ bases. For example:

Table 3.4 Harmonic frequencies of H_2 calculated with [cc-pVTZ, cc-pVQZ] extrapolation

Method	Harmonic Frequency / cm^{-1}
HF	4580.4
LSDA	4179.1
PBE	4310.5
B3LYP	4409.7
MP2	4517.3
MP3	4466.1
MP4	4431.7
CCD	4412.8
CCSD	4401.4
Exact [23]	4401.2

```
TUNA OPTFREQ : H H 1.0 : CCSD cc-pVTZ : EXTRAPOLATE
```

3.4.1 Transition Intensity

A commonly used measure of fundamental absorption intensity is the integral absorption coefficient, $\mathcal{A}$, which has units km mol^{-1}. This coefficient is calculated by

$$\mathcal{A} = \frac{\pi N_A}{3c^2}\left(\frac{\partial \boldsymbol{\mu}}{\partial \mathbf{q}}\right)^2. \tag{3.37}$$

This absorption coefficient is calculated as it is independent of the experimental setup — including the incident light intensity, molar concentration and length of sample cell [24]. This is printed under the "transition intensity" header, along with the dipole moment derivative and its square.

This intensity is proportional to the derivative of the dipole moment at the equilibrium geometry in mass-weighted normal coordinates,

$$\mathcal{A} \propto \frac{\partial \boldsymbol{\mu}}{\partial \mathbf{q}}, \tag{3.38}$$

which is determined by numerical differentiation while the Hessian is determined.

The dipole moment at each geometry is calculated analytically by default, but can be calculated numerically with the `DIPOLE` keyword. This will give the correct intensity for electronic structure

methods without an implemented relaxed density matrix — see subsection 3.1.3. For example, this input line calculates the full configuration interaction harmonic intensity for lithium hydride:

```
TUNA FREQ : Li H 1.54747 : CCSDTQ cc-pVDZ : DIPOLE
```

3.4.2 Thermochemistry

After the vibrational frequency calculation is finished TUNA begins a quick calculation of thermochemical parameters. The contributions to internal energy,

$$U = E + E_{\text{zero-point}} + E_{\text{translational}} + E_{\text{vibrational}} + E_{\text{rotational}} \,, \tag{3.39}$$

and entropy,

$$S = S_{\text{electronic}} + S_{\text{translational}} + S_{\text{vibrational}} + S_{\text{rotational}} \,, \tag{3.40}$$

are first calculated, before the enthalpy is determined,

$$H = U + k_{\text{B}}T \,, \tag{3.41}$$

where k_{B} is the Boltzmann constant and T is the absolute temperature. The Gibbs free energy,

$$G = H - TS \,, \tag{3.42}$$

is evaluated and printed. All these values, including entropies, are expressed in hartree.

The zero-point energy is given by $E_{\text{zero-point}} = \frac{1}{2}\omega$, and is excluded if the vibrational frequency is imaginary. The translational contribution to energy is $E_{\text{translational}} = \frac{3}{2}k_{\text{B}}T$ and the rotational contribution is $E_{\text{rotational}} = k_{\text{B}}T$. Finally, the vibrational contribution is calculated by

$$E_{\text{vibrational}} = \frac{\omega}{\exp\left(\omega/k_{\text{B}}T\right) - 1} \,, \tag{3.43}$$

assuming the molecule behaves as a harmonic oscillator, which is a bad approximation for diatomics, particularly at high temperatures. This entropy is also excluded for imaginary frequencies.

The electronic contribution to entropy is calculated from the multiplicity (related to the total spin S) assuming only the ground state is populated, by

$$S_{\text{electronic}} = k_{\text{B}} \ln(2S + 1) \,. \tag{3.44}$$

The translational, vibrational and rotational contributions are calculated by three fairly horrible formulas, where p is the pressure, B is the rotational constant and M is the molecular mass:

$$S_{\text{translational}} = k_{\text{B}}\left[\frac{5}{2} + \ln\left(\frac{Mk_{\text{B}}T}{h}\right)^{\frac{3}{2}} + \ln\left(\frac{k_{\text{B}}T}{p}\right)\right] \tag{3.45}$$

$$S_{\text{vibrational}} = k_{\text{B}}\left[\frac{\omega}{k_{\text{B}}T\left(\exp(\omega/k_{\text{B}}T) - 1\right)} - \ln\left(1 - \exp\left(-\frac{\omega}{k_{\text{B}}T}\right)\right)\right] \tag{3.46}$$

$$S_{\text{rotational}} = k_{\text{B}} + k_{\text{B}}\ln\left(\frac{k_{\text{B}}T}{2\pi\sigma Bc}\right) \tag{3.47}$$

For the rotational entropy, the symmetry number, σ, is determined from the molecular point group. Symmetric, $D_{\infty\text{h}}$ molecules have $\sigma = 2$ while asymmetric, $C_{\infty\text{v}}$ molecules have $\sigma = 1$.

The enthalpy, entropy, and Gibbs free energy depend on the temperature and pressure. Using the `TEMP` and `PRES` keywords, these quantities can be specified. For instance, a thermochemistry calculation at 340 K and 101500 Pa can be requested by:

```
TUNA FREQ : H H 0.6750 : RHF 6-311++G : TEMP 340 PRES 101500
```

3.4.3 Vibrational Perturbation Theory

The Taylor expansion of the potential energy surface around a stationary point is

$$E(R) = E + \frac{\partial E}{\partial R}R + \frac{1}{2}\frac{\partial^2 E}{\partial R^2}R^2 + \frac{1}{6}\frac{\partial^3 E}{\partial R^3}R^3 + \frac{1}{24}\frac{\partial^4 E}{\partial R^4}R^4 + \cdots \tag{3.48}$$

and since the gradient is zero and we can choose the energy gauge, this simplifies to

$$E(R) = \frac{1}{2}kR^2 + \frac{1}{6}\frac{\partial^3 E}{\partial R^3}R^3 + \frac{1}{24}\frac{\partial^4 E}{\partial R^4}R^4 + \cdots \tag{3.49}$$

where the first term is the harmonic potential, used to solve the nuclear Schrödinger equation in the quantum harmonic oscillator model of vibration. Vibrational perturbation theory provides an improved treatment by considering the next terms. Because first- and second-order vibrational perturbation theory share the same computational cost, the second-order treatment, VPT2, is used in practice although the first-order VPT1 is also implemented in TUNA.

This requires third and fourth derivatives of the energy, which are calculated in TUNA with eight-

and nine-point stencils respectively, discussed in section 7.3. These higher derivatives are used to calculate an approximate anharmonic fundamental frequency, overtone frequency and zero-point energy, which can then feed into a thermochemistry calculation.

Two intermediate quantities, based on third and fourth derivatives, are calculated for diatomic molecules. These are the terms from second and first order perturbation theory respectively. The VPT approach uses isoinertial normal coordinates involving the reduced mass [25].

$$g_3 = -\frac{1}{\mu^3\omega^4}\left(\frac{\partial^3 E}{\partial R^3}\right)^2 \tag{3.50}$$

$$g_4 = \frac{1}{\mu^2\omega^2}\frac{\partial^4 E}{\partial R^4} \tag{3.51}$$

The fundamental frequency then becomes

$$\omega_{\text{fund}} = \omega + \frac{5}{24}g_3 + \frac{1}{8}g_4 \ , \tag{3.52}$$

and the zero-point energy is

$$E_{\text{zero-point}} = \frac{1}{2}\omega + \frac{11}{288}g_3 + \frac{1}{32}g_4 \ . \tag{3.53}$$

The `VPT2` keyword is used with a frequency (or optimisation and frequency) calculation:

```
TUNA OPTFREQ : H H 1.0 : CCSD def2-TZVP : VPT2
```

During the execution, the potential energy surface is scanned symmetrically to calculate the third and fourth derivatives. Since several energy evaluations are already completed to calculate the harmonic frequency, only four additional calculations are necessary to achieve sufficient precision. The VPT2 results are often within a couple of cm^{-1} of the full anharmonic vibrational treatment. Naturally, the more "quadratic-like" the potential, and the lower the anharmonicity constant, the better results VPT2 will yield. The `VPT1` keyword is also implemented and provides a first-order treatment, but this saves no computational cost and loses a lot of accuracy.

After the four additional energies are evaluated, the new zero-point energy and equilibrium energy is printed, along with the fundamental frequency and first overtone in cm^{-1}. The anharmonicity constant, χ, is also printed. Running a VPT2 calculation on a non-stationary point is meaningless, so `TIGHTOPT` convergence criteria are used by default when `VPT2` is requested.

Table 3.5 Fundamental frequencies in cm^{-1} calculated with the quantum harmonic oscillator model, second-order vibrational perturbation theory and full nuclear Hamiltonian diagonalisation in TUNA. Results are from CCSD(T) in the def2-TZVP basis set for various molecules

Molecule	FREQ	VPT2	ANHARM
H — H	4430.4	4182.9	4184.6
H — F	4132.9	3954.0	3956.9
Li — H	1413.7	1360.4	1360.3
C — O	2468.9	2447.1	2447.1

Table 3.5 shows how the VPT2 frequencies hold up against our implementation of the full anharmonic treatment, for various molecules calculated with CCSD(T) and def2-TZVP. These complex calculations were performed with the notably simple input line:

```
TUNA ANHARM : H H 1.0 : CCSD[T] def2-TZVP : VPT2
```

3.5 Anharmonic Frequency

Since the potential energy surface of a diatomic molecule is one dimensional, it is feasible to "exactly" solve the nuclear Schrödinger equation using the surface generated by TUNA, using any electronic structure method and basis set combination with the ANHARM calculation type.

```
TUNA ANHARM : C O 1.0 : MP2 6-31G :
```

The time taken to generate the potential energy surface is the rate limiting step in these calculations, and the building and diagonalisation of the nuclear Hamiltonian is rapid. Assuming full configuration interaction and a complete one-electron basis set, this allows the calculation of full anharmonic vibrational frequencies to sub-cm^{-1} accuracy for many molecules.

The fundamental transition frequency is

$$\omega_{\text{fund}} = E_1 - E_0 \,, \tag{3.54}$$

where $\{E_i\}$ are the eigenvalues of the nuclear Hamiltonian for the ith wavefunction,

$$\hat{H}_{\text{nuc}}\Psi_i(R) = E_i\Psi_i(R) \,. \tag{3.55}$$

Anharmonic frequency calculations are performed by TUNA as follows:

- The molecule is optimised to find its equilibrium bond length
- The harmonic vibrational frequency is calculated at this bond length
- A small 0.4 Å section of the potential energy surface is scanned, forwards and backwards
- This potential energy surface is interpolated
- The nuclear Hamiltonian is formed and diagonalised
- A further 0.1 Å is scanned on either side of the potential energy surface

This process of further scanning repeats until diagonalisation of the nuclear Hamiltonian yields a fundamental transition frequency that is converged to within 0.01 cm^{-1}. This default convergence criteria can be changed with the `ANHARMCONV` keyword. A looser convergence is obtained as:

```
TUNA ANHARM : N N 1.0 : QCISD 6-31G : ANHARMCONV 0.1
```

The step length for the coordinate scans is 0.05 Å by default, which is almost always sufficiently converged, but can be changed with the `STEP` keyword. Each iteration, the fundamental transition frequency, harmonic frequency, anharmonicity constant and bond length range of the potential energy surface are printed into a table.

This iterative approach ensures that a minimal number of energy evaluations are performed, without any loss of accuracy. But bear in mind that anharmonic frequency calculations are still very intensive and require around 30 energy evaluations in total. All `ANHARM` calculations use `EXTREME` SCF convergence criteria by default. To avoid issues with linear independence, the bond length is not allowed to get too small — this doesn't affect the quality of results as at short bond length the wavefunctions become tiny. However, you'll probably want to avoid basis sets with diffuse functions, and may want to reduce the overlap matrix eigenvalue threshold with `STHRESH`.

The Hamiltonian of nuclei moving on a potential energy surface (PES) calculated by TUNA is

$$\hat{H}_{\text{nuc}} = -\frac{1}{2\mu}\frac{\partial^2}{\partial R^2} + V_{\text{PES}} \, . \tag{3.56}$$

The nuclear Schrödinger equation is solved on a grid, using a basis set of delta functions. This gives a matrix on the order of 1000 × 1000, but it can be diagonalised for the first few states efficiently since it is tridiagonal — only consecutive points are coupled due to the kinetic energy term so the matrix is very sparse. The second derivative is expressed as finite differences as

$$\hat{H}_{\text{nuc}}\Psi(R_n) = -\frac{1}{2\mu}\frac{\Psi(R_{n+1}) - 2\Psi(R_n) + \Psi(R_{n-1})}{(R_n - R_{n-1})^2} + V_{\text{PES}}(R_n)\Psi(R_n) \, . \tag{3.57}$$

This can be split into a diagonal and off diagonal part, for a uniform grid where $\Delta R = R_n - R_{n-1}$

$$\hat{H}_{\text{nuc}}\Psi(R_n) = \Big[\frac{1}{\mu\Delta R^2} + V_{\text{PES}}(R_n)\Big]\Psi(R_n) - \frac{\Psi(R_{n+1}) + \Psi(R_{n-1})}{2\mu\Delta R^2} \,. \tag{3.58}$$

This produces a Hamiltonian matrix with the diagonal part

$$\hat{H}_n^n = \frac{1}{\mu\Delta R^2} + V_{\text{PES}}(R_n) \,, \tag{3.59}$$

and the off-diagonal part, where $n \neq m$,

$$\hat{H}_n^m = -\frac{\delta_{n+1,m} + \delta_{n-1,m}}{2\mu\Delta R^2} \tag{3.60}$$

so the two closest off diagonal matrix elements are, where $m = n \pm 1$

$$\hat{H}_n^m = -\frac{1}{2\mu\Delta R^2} \tag{3.61}$$

and all the other elements are zero — this is the case for a "tridiagonal" matrix. This matrix is formed by TUNA after interpolation of the potential energy surface using cubic splines.

The lowest eigenvalue of the nuclear Hamiltonian, E_0, is the anharmonic zero-point energy. The energy gap between this and the first excited vibrational state is the fundamental transition frequency, and comparison to the harmonic frequency defines the anharmonicity constant,

$$\chi = \frac{\omega_0^1 - \omega_1^2}{2\omega_{\text{harm}}} \,, \tag{3.62}$$

where $\{\omega_i^j\}$ are the transition matrix elements from the ith to jth nuclear state.

3.5.1 Transition Intensity

After convergence is reached, the infrared spectrum is printed for various transitions between nuclear states $i \to j$, and their intensities are calculated. The transitions are printed as energies in hartree, wavenumbers in cm^{-1}, and wavelengths in nm. Intensities are printed in km mol^{-1}.

The intensity of a vibrational transition depends on the transition dipole moment operator

$$\mathcal{A}_{ij} \propto \langle\Psi_i|\hat{\boldsymbol{\mu}}|\Psi_j\rangle = \int \Psi_i^*(R)\mu(R)\Psi_j(R)\,\mathrm{d}R \,, \tag{3.63}$$

which is calculated in TUNA by simple numerical integration with the trapezium rule, interpolating the dipole moment at each bond length along the coordinate scans with cubic splines.

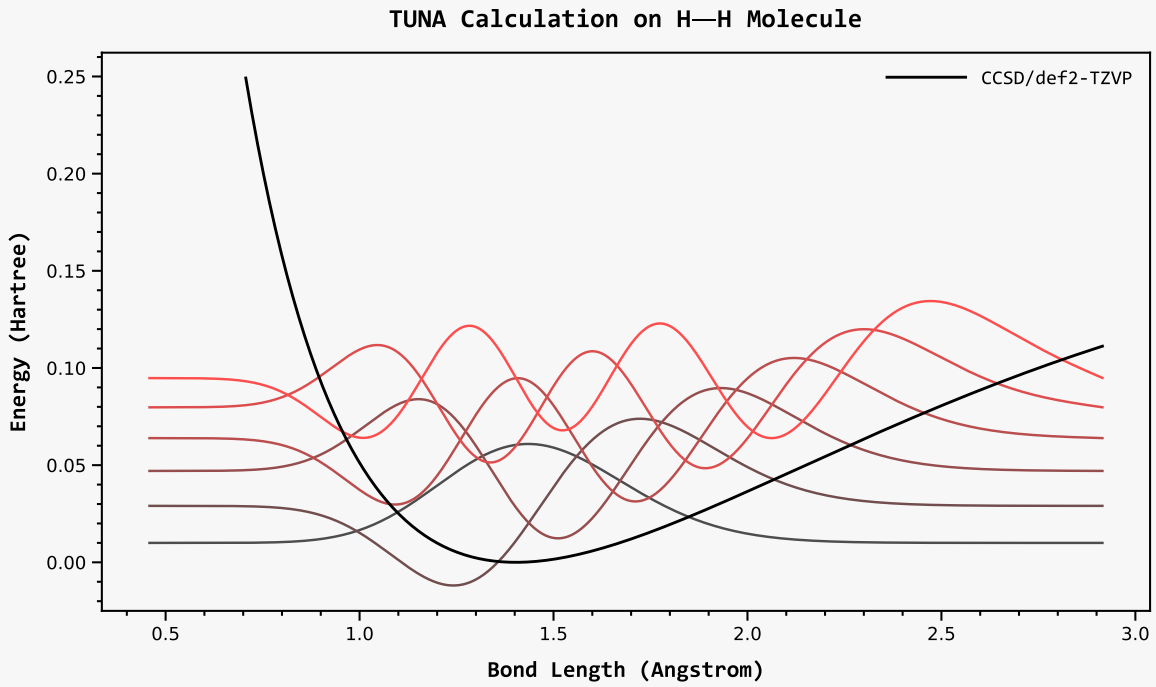

Figure 3.7: Potential energy surface (black) and nuclear wavefunctions (increasingly bright red colour with energy) calculated in TUNA and shown with the VIBPLOT keyword

By default, the intensities use the analytical dipole moment to calculate the transition dipole matrix elements. This is correct for any method with an implemented one-electron response density matrix, but not exact for methods with an approximate "unrelaxed" density matrix. However, the numerical dipole moment is correct for any electronic structure method, and can be requested with the DIPOLE keyword. The following calculation gives full configuration interaction anharmonic intensities — within the tiny one-electron basis set.

```
TUNA ANHARM : He H 1.0 : CISD 3-21G : CH 1 DIPOLE
```

The calculated intensities differ somewhat from (and are more accurate than) a calculation using analytical dipole moments from the linearised CCSD density matrix:

```
TUNA ANHARM : He H 1.0 : CISD 3-21G : CH 1
```

After the infrared spectrum is printed, so is the anharmonic zero-point energy, which can be used to calculate anharmonic thermochemical parameters with the additional print keyword, P.

3.5.2 Plotting Nuclear Wavefunctions

At the end of an ANHARM calculation, the nuclear wavefunctions and potential energy surface can be plotted with the VIBPLOT keyword. The SAVEPLOT keyword can be used in conjunction with this. The vibrational wavefunctions are plotted at their corresponding eigenstate energies, with more intense colouring the higher the energy. Only the first six nuclear eigenstates are plotted.

The following calculation will determine the quadruple-ζ basis set, full configuration interaction anharmonic vibrational transition frequencies for the hydrogen molecule. Running the (fairly long) calculation gives a result of 4161 cm^{-1}, within sub-cm^{-1} spectroscopic accuracy [26].

```
TUNA ANHARM : H H 0.74176 : CISD def2-QZVPP : VIBPLOT
```

3.6 Bond Dissociation Energy

Commonly calculated quantities for diatomics have been automated in TUNA, such as the bond dissociation energy which can be calculated by the simple BDE calculation type, which determines

$$E_{\text{BDE}} = E(\text{A}) + E(\text{B}) - E(\text{A–B}) \tag{3.64}$$

between atoms A and B. In these calculations, the first step is a geometry optimisation. Single point energies of the constituent atoms are then determined, in the presence of "ghost" basis functions to counteract basis set superposition error, by default [11]. This can be disabled with the NOCP keyword. In BDE calculations, TIGHT energy convergence is used by default.

To avoid redundant calculations, TUNA will only perform full configuration interaction if an even higher order calculation is requested. For example, a CCSDTQ bond dissociation energy calculation on lithium hydride will calculate the energy of the molecule with CCSDTQ, the lithium atom with CISDT and the hydrogen atom with Hartree–Fock — these are all the exact energies within the one-electron basis set. This approach can speed up the atomic calculations a lot, but can be disabled if desired with the FORCEMETHOD keyword.

```
TUNA BDE : Li H 1.5 : CCSDTQ 3-21G : NOCP
```

To improve agreement with experiment, the zero-point energy of the molecule can be calculated using the ZPE keyword. Second-order vibrational perturbation theory with VPT2 can also be requested, to provide a good approximation to anharmonic effects and improve the energetics.

This bond dissociation energy calculation on dihydrogen gives an energy of 0.16387 E_h, in pretty

good agreement with the experimental value of 0.16457 E_h [27].

```
TUNA BDE : H H 1.0 : CISD cc-pVQZ : ZPE
```

The discrepancy is partly due to the harmonic zero-point energy. The anharmonic zero-point energy could be calculated exactly with an `ANHARM` calculation, but it is simpler to just request second-order vibrational perturbation theory. The following calculation gives 0.16397 E_h.

```
TUNA BDE : H H 1.0 : CISD cc-pVQZ : ZPE VPT2
```

An even better result of 0.16475 E_h comes by extrapolation from triple-ζ to quadruple-ζ:

```
TUNA BDE : H H 1.0 : CISD cc-pVTZ : ZPE VPT2 EXTRAPOLATE
```

Finally, a very intensive basis set extrapolation from quadruple-ζ to quintuple-ζ gives a bond dissociation energy of 0.16460 E_h which is only 0.02 kcal mol^{-1} from experiment — bang on!

```
TUNA BDE : H H 1.0 : CISD cc-pVQZ : ZPE VPT2 EXTRAPOLATE
```

3.7 Ionisation Potential and Electron Affinity

There are also inbuilt calculation types to determine the ionisation potential,

$$\mathrm{IP} = E(\mathrm{X}^+) - E(\mathrm{X}) \,, \tag{3.65}$$

and the electron affinity

$$\mathrm{EA} = E(\mathrm{X}) - E(\mathrm{X}^-) \,. \tag{3.66}$$

These quantities can be calculated with the calculation types `IP` and `EA` respectively. Unlike the ionisation potential from Koopmans' theorem, the orbitals are allowed to fully relax. If a correlated calculation is requested, it will be performed on these converged orbitals. An ionisation potential calculation on the carbon monoxide molecule is:

```
TUNA IP : C O 1.0 : HF 3-21G :
```

The analogous electron affinity calculation is:

```
TUNA EA : C O 1.0 : HF 3-21G :
```

By default, an IP or EA calculation on a molecule will be adiabatic. First, the molecule is geometry optimised in its native charge state (normally neutral). The final energy of this molecule is saved as $E(\mathrm{X})$ and the final molecular structure is used as a starting guess for the ionised (or electron attached) calculation. With the new charge assigned, TUNA then optimises the geometry again to return the final charged energy such that the adiabatic IP or EA can be calculated.

For atoms, which can't be optimised, the vertical IP or EA is calculated instead. The vertical IP or EA can be calculated on molecules using the `VERTICAL` keyword:

```
TUNA EA : H H 0.7414 : PBE0 ano-pVDZ : VERTICAL
```

By default, the `IP` calculation will remove one electron and the `EA` calculation will add one electron to whatever charge and multiplicity state the calculation is initially called with. To calculate the IP or EA with more than one removed or added electron, the `NELEC` keyword can be used. For example, this calculation will determine the ionisation potential,

$$\mathrm{IP} = E(\mathrm{Be}^{2+}) - E(\mathrm{Be})\ . \tag{3.67}$$

```
TUNA IP : Be : CCSDT cc-pVTZ : NELEC 2
```

The output for IP or EA calculations first shows the charges of the two states, then prints the energy of the reference and charged system. For adiabatic calculations, the optimised bond lengths of the two systems are also printed, then for all calculations the final IP or EA is presented.

If one of the calculations to obtain the IP or EA will be on a zero-electron system, or produces any other weird state, the calculation will exit with an error message as it normally would.

3.8 *Ab Initio* Molecular Dynamics

The implementation in TUNA of *ab initio* molecular dynamics (AIMD) is Born–Oppenheimer molecular dynamics [28], where nuclear positions are updated by the classical equations of motion, $\mathbf{F} = m\mathbf{a}$, over the quantum chemical potential energy surface. Therefore, unlike other AIMD methods such as Car–Parrinello molecular dynamics where the electronic degrees of freedom are also dynamic [29], the *ab initio* energy and gradient must be recalculated at every timestep.

To run an AIMD calculation in TUNA, use the `MD` calculation type with any electronic structure method and basis, and set the timestep (fs), initial temperature (K) and number of steps with the keywords `STEP`, `TEMP` and `NUM` respectively. By default, `STEP` is 0.1 fs, `TEMP` is 0 K and `NUM` is 50.

For instance, a 200 step AIMD simulation using MP2/6-31G with a timestep of 0.2 fs at an initial temperature of 350 K can be requested by:

```
TUNA MD : H H 1.2 : MP2 6-31G : STEP 0.2 NUM 200 TEMP 350
```

As positions and velocities are continually updated, a trajectory is mapped out. By default, TUNA prints this trajectory to a file called `"tuna-trajectory.xyz"` in the terminal's directory — this logging can be disabled using the `NOTRAJ` keyword. The `TRAJ [filepath]` keyword can also be used to change the name of the file — just include the name in quotation marks.

The output prints the step number, time, bond length and temperature. The classical kinetic energy of the nuclei, as well as their potential energy (the total electronic energy) is also printed, with the total molecular energy. The timestep should be chosen to conserve the total system energy. At each timestep, all energy and gradient information can be printed using the additional print keyword, `P`, although this makes a big mess.

Because the Velocity Verlet algorithm [30] yields velocities and positions at the time point and is numerically stable, it is implemented in TUNA. Positions are updated in three dimensions by

$$\mathbf{R}(t+\Delta t) = \mathbf{R}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2, \tag{3.68}$$

and velocities are updated by

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t)+\mathbf{a}(t+\Delta t)}{2}\Delta t\ . \tag{3.69}$$

By default, the AIMD calculation begins with the atoms at rest. However, by using the `TEMP` keyword an initial temperature can be specified from which the total classical kinetic energy of the molecule is calculated by

$$K = 3k_{\mathrm{B}}T\ , \tag{3.70}$$

where

$$K = \frac{1}{2}m_{\mathrm{A}}\mathbf{v}_{\mathrm{A}}^2 + \frac{1}{2}m_{\mathrm{B}}\mathbf{v}_{\mathrm{B}}^2\ . \tag{3.71}$$

From this kinetic energy, velocities are randomly determined to satisfy the Maxwell–Boltzmann distribution. Next, net molecular translations are removed before the velocities are rescaled to produce the specified temperature exactly. Net rotations are not removed, so to handle this three-dimensional data — which is stressful for TUNA — the force vectors are scaled by the gradient in place.

The initial specified temperature will not be conserved, as thermostats don't really make sense for diatomics, so all AIMD calculations are run in the NVE ensemble. The shorter the timestep, the more faithful the energy convergence. Like the rest of TUNA, in principle any electronic structure method, basis set and keyword combination can be used for a molecular dynamics simulation.

```
TUNA MD : Li H 1.5 : CCSDT[Q] cc-pVDZ : NUM 500 STEP 0.01
```

4 Electronic Structure Methods

The wavefunction theory electronic structure methods available in TUNA are listed in Table 4.1. A plethora of density functional approximations are also available, which are presented in section 4.5.

Most of these methods are available for restricted and unrestricted references — the unrestricted reference can be requested for a singlet molecule by adding the letter "U" to the start of a method.

Table 4.1 Wavefunction-based electronic structure methods implemented in TUNA

Keyword	Electronic Structure Method
HF	Hartree–Fock theory
UHF	Unrestricted Hartree–Fock theory
H	Hartree theory
UH	Unrestricted Hartree theory
MP2	Second-order Møller–Plesset perturbation theory
UMP2	Unrestricted second-order Møller–Plesset perturbation theory
SCS-MP2	Spin-component-scaled MP2 perturbation theory
USCS-MP2	Unrestricted spin-component-scaled MP2 theory
OMP2	Orbital-optimised MP2 theory
UOMP2	Unrestricted orbital-optimised MP2 theory
IMP2	Iterative MP2 theory
AO-MP2	Laplace transform AO-MP2 theory
MP3	Third-order Møller–Plesset perturbation theory
UMP3	Unrestricted third-order Møller–Plesset perturbation theory
SCS-MP3	Spin-component-scaled MP3 perturbation theory
USCS-MP3	Unrestricted spin-component-scaled MP3 theory

Keyword	Electronic Structure Method
MP4	Fourth-order Møller–Plesset perturbation theory
MP4[DQ]	Restricted MP4 theory with no singles or triples
MP4[SDQ]	Restricted MP4 theory with no triples
RPA	Random phase approximation
TDHF	Time-dependent Hartree–Fock theory
UTDHF	Unrestricted time-dependent Hartree–Fock theory
CIS	Configuration interaction singles
UCIS	Unrestricted configuration interaction singles
CIS[D]	CIS with perturbative doubles
UCIS[D]	Unrestricted CIS with perturbative doubles
CID	Configuration interaction doubles
UCID	Unrestricted configuration interaction doubles
CISD	Configuration interaction singles and doubles
UCISD	Unrestricted configuration interaction singles and doubles
CISDT	Configuration interaction singles, doubles and triples
UCISDT	Unrestricted configuration interaction singles, doubles and triples
FCI	Full configuration interaction
CASCI	Complete active space configuration interaction
CASSCF	Complete active space self-consistent field
CEPA0	Coupled electron pair approximation
UCEPA0	Unrestricted coupled electron pair approximation
LCCD	Linearised coupled cluster doubles
ULCCD	Unrestricted linearised coupled cluster doubles
LCCSD	Linearised coupled cluster singles and doubles
ULCCSD	Unrestricted linearised coupled cluster singles and doubles

Keyword	Electronic Structure Method
`CC2`	Approximate coupled cluster singles and doubles
`CC3`	Approximate coupled cluster singles, doubles and triples
`CCD`	Coupled cluster doubles
`UCCD`	Unrestricted coupled cluster doubles
`CCSD`	Coupled cluster singles and doubles
`UCCSD`	Unrestricted coupled cluster singles and doubles
`CCSD[T]`	Coupled cluster singles, doubles and perturbative triples
`UCCSD[T]`	Unrestricted coupled cluster singles, doubles and perturbative triples
`CCSDT`	Coupled cluster singles, doubles and triples
`UCCSDT`	Unrestricted coupled cluster singles, doubles and triples
`CCSDT[Q]`	Coupled cluster singles, doubles, triples and perturbative quadruples
`CCSDTQ`	Coupled cluster singles, doubles, triples and quadruples
`EOM-CCSD`	Equation of motion coupled cluster singles and doubles
`UEOM-CCSD`	Unrestricted equation of motion coupled cluster singles and doubles
`QCISD`	Quadratic configuration interaction singles and doubles
`UQCISD`	Unrestricted quadratic CI singles and doubles
`QCISD[T]`	Quadratic CI singles, doubles and perturbative triples
`UQCISD[T]`	Unrestricted quadratic CI singles, doubles and perturbative triples

4.1 Hartree–Fock Theory

The simplest *ab initio* quantum chemistry method is Hartree–Fock theory [31].

Rather than calculating instantaneous electron–electron interactions, Hartree–Fock assumes each electron experiences an average field generated by all the others, described with one-electron wavefunctions known as molecular orbitals. Because the orbitals depend on the field and the field depends on the orbitals, the problem is solved iteratively: an initial guess for the orbitals is used to

construct the one-electron Hamiltonian (the Fock matrix), which then generates new, refined orbitals. This cycle repeats until the orbitals stop changing, resulting in a self-consistent field (SCF).

Both restricted and unrestricted Hartree–Fock calculations are available in TUNA. The `HF` method keyword defaults to `RHF` for singlet molecules and `UHF` for non-singlets. Unrestricted Hartree–Fock can also be requested on a singlet molecule using the `UHF` method, such as:

```
TUNA SPE : H H 1.85 : UHF 6-311++G
```

The Roothaan–Hall equations [32, 33] turn the variational differential restricted Hartree–Fock equations [31] into a simpler eigenvalue problem,

$$\mathbf{FM} = \mathbf{SM}\boldsymbol{\epsilon} \,, \tag{4.1}$$

where $\mathbf{F}$ is the Fock matrix, $\mathbf{M}$ is the molecular orbital matrix in the basis of atomic orbitals, $\mathbf{S}$ is the overlap matrix and $\boldsymbol{\epsilon}$ is a diagonal matrix of orbital energies. The Fock matrix consists of contributions from the kinetic energy matrix, $\mathbf{T}$, nuclear–electron attraction matrix, $\mathbf{V}$, Coulomb matrix, $\mathbf{J}$, and exchange matrix, $\mathbf{K}$, by

$$\mathbf{F} = \mathbf{T} + \mathbf{V} + 2\mathbf{J} - \mathbf{K} \,. \tag{4.2}$$

These matrices are determined from one- and two-electron integrals of atomic orbital basis functions. To solve the Roothaan–Hall equations, they need to be converted into a conventional eigenvalue problem; the overlap matrix needs to become the identity matrix. To do this, the basis functions can be rotated to an orthonormal basis. One method to achieve this defines $\mathbf{X} = \mathbf{S}^{-\frac{1}{2}}$, such that when the Fock matrix is rotated by

$$\mathbf{F}' = \mathbf{X}^{\dagger}\mathbf{F}\mathbf{X} \,, \tag{4.3}$$

the Roothaan–Hall equations become a conventional eigenvalue problem,

$$\mathbf{F}'\mathbf{M}' = \mathbf{M}'\boldsymbol{\epsilon} \,. \tag{4.4}$$

Now the rotated Fock matrix, $\mathbf{F}'$, is diagonalised for its eigenvectors, $\mathbf{M}'$, and eigenvalues, $\boldsymbol{\epsilon}$. The eigenvectors are then rotated back to the original basis, where they represent molecular orbitals,

$$\mathbf{M} = \mathbf{X}\mathbf{M}' \,. \tag{4.5}$$

Now the molecular orbitals have been determined the restricted Hartree–Fock density matrix, $\mathbf{P}$,

is constructed by summing over the lowest energy occupied orbitals,

$$P_{\mu\nu} = 2\sum_{i}^{\text{occ}} M_{\mu i} M_{\nu i}^{*} \,. \tag{4.6}$$

Finally, the energy can be evaluated by contracting this new density matrix with the Fock and one-electron matrices, and adding on the constant nuclear repulsion energy,

$$E = V_{\text{nn}} + \frac{1}{2}\sum_{\mu\nu} P_{\mu\nu}(F_{\mu\nu} + H_{\mu\nu}^{\text{core}}) \,. \tag{4.7}$$

Here, $\mathbf{H}^{\text{core}} = \mathbf{T}+\mathbf{V}$. The density matrix is contracted with the two-electron integrals to determine the new two-electron contribution to the Fock matrix,

$$2\mathbf{J} - \mathbf{K} = \sum_{\kappa\lambda} P_{\kappa\lambda}\big[(\mu\nu|\kappa\lambda) - \frac{1}{2}(\mu\kappa|\nu\lambda)\big] \,. \tag{4.8}$$

The Fock matrix is then constructed again, from which new molecular orbitals are determined as this procedure repeats. When the energy and density matrix change from the last iteration less than the convergence criteria, the mean field has become self-consistent and a stationary point on the surface of orbital rotations has been found — the Hartree–Fock wavefunction.

4.1.1 Initial Guess

In TUNA, the simplest choice of initial guess for a single point energy calculation is the density matrix from a one-electron calculation, where contributions from $\mathbf{J}$ and $\mathbf{K}$ are ignored and the one-electron Hamiltonian to diagonalise is simply

$$\mathbf{F} = \mathbf{H}^{\text{core}} \,. \tag{4.9}$$

This works well for diatomics of hydrogen and helium, but fails catastrophically even for relatively light diatomics in the p block. The guess density is so contracted in these molecules that the SCF can converge to a metastable state or a saddle point. The core Hamiltonian guess method can be requested with the `COREGUESS` keyword. Doing this for the hydrogen molecule will make the calculation slightly faster, without losing any accuracy — probably.

```
TUNA SPE : H H 1.0 : HF pc-1 : COREGUESS
```

Another widely used guess strategy is the superposition of atomic densities guess. Here the guess density matrix is evaluated as a block diagonal matrix of atomic density matrices, assuming zero

interaction between the atoms (ie. a simple superposition of the densities),

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_\mathrm{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\mathrm{B} \end{bmatrix} . \tag{4.10}$$

This produces a much less contracted density that tends to reduce the number of SCF iterations necessary to converge. The initial atomic density matrices are stored in TUNA as spherically averaged STO-3G density matrices. These are projected onto the requested basis set by

$$\mathbf{P}_2 = \mathbf{S}_2^{-1}\mathbf{S}_{12}\mathbf{P}_1\mathbf{S}_{12}\mathbf{S}_2^{-1} , \tag{4.11}$$

where $\mathbf{P}_1$ is the minimal basis set and $\mathbf{P}_2$ is the requested basis set. For this, the cross overlap matrix between the two basis sets is needed, which is happily provided by the molecular integral engine. A downside is that simple densities are not as well represented, so the SCF may take longer to converge in some cases. This guess strategy can be requested with the `SADGUESS` keyword.

```
TUNA SPE : N F 1.0 : HF pc-1 : SADGUESS
```

The default guess strategy in TUNA embodies the best of both worlds, and can be accessed with the `SCFGUESS` keyword. Here, a STO-3G minimal basis set SCF calculation is first fully converged, using the superposition of atomic densities guess for *that* calculation. The converged density matrix is then projected using Equation 4.11 onto the larger basis, and the true calculation begins.

We have found this to be the most robust guess strategy, and the one which minimises the number of needed SCF iterations on average, but it is naturally a fair bit slower than the other methods. This is felt most during DFT calculations as the basis functions need to be expressed on a grid, but considering the benefit in limiting the number of SCF iterations it is probably still worth it.

```
TUNA SPE : N F 1.0 : HF pc-1 : SCFGUESS
```

In all cases, the initial guess energy is contracted from the guess density matrix contracted with the core Hamiltonian matrix,

$$E_\mathrm{guess} = \sum_{\mu\nu} H_{\mu\nu}^\mathrm{core} P_{\mu\nu} . \tag{4.12}$$

4.1.2 Symmetry Breaking

If an unrestricted Hartree–Fock calculation is run on a singlet molecule with equal α and β guess densities the SCF will likely get caught in a local minimum or saddle point on the surface of orbital

rotations, the restricted Hartree–Fock solution. This happens for diatomics at long bond lengths.

To handle this, TUNA rotates the HOMO with the LUMO after the guess density is made, by

$$\begin{bmatrix} \psi'_{\text{HOMO}} \\ \psi'_{\text{LUMO}} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \psi_{\text{HOMO}} \\ \psi_{\text{LUMO}} \end{bmatrix} . \tag{4.13}$$

By default, $\theta = 45°$, which corresponds to a half-and-half mixing. This can cause some problems with convergence when the RHF and UHF solutions are (near-)degenerate, so the rotation can be disabled with the `NOROTATE` keyword. Orbital rotation can be activated with the `ROTATE` keyword. If the default value of θ is not converging well, use the `ROTATE` keyword with the desired rotation angle in degrees to change it. The value of `THETA` is printed when orbital rotation is used.

```
TUNA SPE : H He 1.2 : UHF 6-31G : ROTATE 45
```

A well-considered guess rotation may allow a metastable energy state to be converged. Whether an initial guess rotation is necessary to converge the lowest energy Hartree–Fock state can be checked by performing a stability analysis.

4.1.3 Stability Analysis

The self-consistent field process optimises the Hartree–Fock or Kohn–Sham wavefunction to a stationary point in the space of orbital rotations. However, this may be a saddle point rather than a local minimum. To determine whether there is an instability in the wavefunction with respect to an orbital rotation, a stability calculation can be requested with the `STAB` keyword:

```
TUNA SPE : H H 2.0 : RHF STO-3G : STAB
```

The above calculation shows that there is an instability with respect to unrestricted Hartree–Fock. Switching to UHF, we see that this instability has gone, and the energy has lowered:

```
TUNA SPE : H H 2.0 : UHF STO-3G : STAB
```

This only works because of the automatic symmetry-breaking in TUNA. Turning this off with `NOROTATE` still results in an instability due to the symmetric guess leading to the RHF solution:

```
TUNA SPE : H H 2.0 : UHF STO-3G : STAB NOROTATE
```

This works by forming and diagonalising the orbital Hessian. Similarly to diagonalising the geometric Hessian to check if an optimised structure is a local minimum, a negative eigenvalue of the orbital Hessian indicates an instability with respect to a molecular orbital rotation.

For a spin-restricted calculation, both the "singlet" and "triplet" Hessian are calculated. The former checks for RHF/RHF instabilities, and the latter checks for RHF/UHF instabilities.

The singlet orbital Hessian is calculated by

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} , \tag{4.14}$$

where

$$A_{ia,jb} = (\varepsilon_a - \varepsilon_i)\delta_{ij}\delta_{ab} + 2(ia|jb) - \beta(ij|ab) + (1-\beta)K^{\mathsf{XC}}_{ijab} \tag{4.15}$$

and

$$B_{ia,jb} = 2(ia|bj) - \beta(ib|aj) + (1-\beta)K^{\mathsf{XC}}_{ijab} . \tag{4.16}$$

Here the density functional kernel matrix, K^{XC}_{ijab}, is the same as in time-dependent DFT.

4.1.4 Self-consistent Field Convergence

A number of SCF convergence criteria are available in TUNA, shown in Table 4.2. These can be activated by the keywords `LOOSE`, `MEDIUM` (default for single point calculations), `TIGHT` (default for first derivative calculations) and `EXTREME` (default for higher derivative calculations).

The number of SCF iterations before TUNA gives up can be picked with the keyword `MAXITER`, which is 100 by default. Convergence is measured by the change in energy, maximum and root-mean-square change in the density matrix, and the DIIS error (labelled `Error` in the TUNA output), which is the root-mean-square of the commutator between $\mathbf{F}$ and $\mathbf{PS}$. More precise control over each of these can be found with the `ECONV`, `MAXDP`, `RMSDP` and `DIISERR` keywords, respectively. Note that `ECONV` also determines the energy convergence for post-SCF iterative methods.

There are several methods available to accelerate SCF convergence. The most powerful is probably Fock matrix extrapolation, known as direct inversion of the iterative subspace (DIIS) [34].

At SCF convergence, the density matrix commutes with the Fock matrix,

$$\mathbf{SPF} - \mathbf{FPS} = \mathbf{0} . \tag{4.17}$$

An error matrix can therefore be defined at the ith iteration, whose root-mean-square indicates

Table 4.2 Self-consistent field convergence criteria in TUNA

Convergence	ΔE	MAX($\Delta\mathbf{P}$)	RMS($\Delta\mathbf{P}$)	RMS($[\mathbf{F},\mathbf{PS}]$)
LOOSE	10^{-6}	10^{-5}	10^{-6}	10^{-4}
MEDIUM	10^{-7}	10^{-6}	10^{-7}	10^{-5}
TIGHT	10^{-9}	10^{-8}	10^{-9}	10^{-7}
EXTREME	10^{-11}	10^{-10}	10^{-11}	10^{-9}

how far the Hartree–Fock solution is from self-consistency,

$$\mathbf{SP}_i\mathbf{F}_i - \mathbf{F}_i\mathbf{P}_i\mathbf{S} = \mathbf{e}_i \ . \tag{4.18}$$

In DIIS, an error vector is built, where each error matrix is associated with a Fock matrix. This error vector is then optimised in a least-squares minimisation to yield the proportion of each Fock matrix, c_j, which can be used to make a new Fock matrix with a lower error matrix,

$$\mathbf{F}' = \sum_j c_j\mathbf{F}_j \ . \tag{4.19}$$

DIIS tends to be very robust. TUNA stores 6 previous Fock matrices at each SCF step by default — which can be changed using `DIIS [Num. Matrices]` — and will reset this stored array if the equations approach linear dependency and break, and a message will be printed to relay this at the end of the SCF iteration. The DIIS procedure begins after the second SCF step, to prevent extrapolation before convergence has properly begun, and can be deactivated with `NODIIS`.

Another convergence accelerator, which may be the most conceptually simple, is damping, where the density matrix, $\mathbf{P}$, is mixed with a fraction of the density from the previous SCF cycle, by

$$\mathbf{P}' = a\mathbf{P}_{\text{old}} + (1-a)\mathbf{P} \ . \tag{4.20}$$

Static damping, where the value of a is fixed, can be invoked with the `SLOWCONV` or `VERYSLOWCONV` keywords, where $a = 0.5$ and $a = 0.85$ respectively. These can be very helpful to force a difficult case to converge, although should be used with a very large value of `MAXITER`. A calculation with a maximum of 200 iterations and $a = 0.5$ can be called by:

```
TUNA SPE : H He 0.6 : HF 6-31+G : SLOWCONV MAXITER 200
```

The value of a can be chosen specifically with the `DAMP` keyword. For example, a static damping

factor of $a = 0.3$ can be requested by:

```
TUNA SPE : H He 0.6 : HF 6-31+G : DAMP 0.3
```

However, the default method is dynamic damping, where the damping factor, a, is some function of the distance from convergence. The implemented scheme for dynamic damping is that from Zerner and Hehenberger [35]. In this method, Mulliken gross atomic populations [14] are calculated in each cycle and extrapolated for an optimal damping factor, the average of the calculated atomic damping factors weighted by the proportion of basis functions on each atom.

This dynamic approach gives a damping factor that more rapidly converges, and avoids unnecessary damping in the static case. The dynamical damping method is enabled by default, can be disabled with the `NODAMP` keyword and is compatible with DIIS. The maximum damping factor is 0.7 by default, and can be controlled with the `MAXDAMP` keyword.

4.1.5 Hartree Theory

To isolate how the Fock exchange contribution affects energies and properties, calculations can be performed with Hartree theory with the `H` method, which sets the proportion of exchange to zero.

```
TUNA OPT : H H 1.0 : H ano-pVDZ : DAMP 0.3
```

The Hartree bond length, 1.06 Å, for dihydrogen is much longer (and less accurate) compared to the Hartree–Fock bond length of 0.74 Å.

The `HFX` parameter enables more precise investigations of Fock exchange. The following calculation represents 50% of the exact exchange being used, and gives an improved bond length of 0.87 Å.

```
TUNA OPT : H H 1.0 : HF ano-pVDZ : HFX 0.5
```

4.2 Many-body Perturbation Theory

The instantaneous electron–electron interaction neglected in Hartree–Fock theory makes up only a small proportion of the overall molecular energy, E, but is necessary to describe phenomena including dispersion forces and entanglement and to provide quantitatively accurate energetics.

Various approaches to calculate this molecular "correlation" energy,

$$E_{\text{corr}} = E - E_{\text{HF}} \,, \tag{4.21}$$

have been developed.

The simplest post-Hartree–Fock electronic structure method to introduce electron–electron correlation is found in perturbation theory, where the exact molecular Hamiltonian is split into the solvable one-electron part, the Fock operator, and a perturbative part,

$$\hat{H} = \hat{F} + \hat{V} \,, \tag{4.22}$$

then the standard equations of perturbation theory are applied for working formulae [36]. This approach defines the total molecular energy as a series of energy contributions,

$$E = E_{\text{MP0}} + E_{\text{MP1}} + E_{\text{MP2}} + E_{\text{MP3}} + E_{\text{MP4}} \cdots \tag{4.23}$$

The zeroeth and first order terms sum to the Hartree–Fock energy, so the first and most important contribution to electron correlation comes at second-order, and is known as MP2 [37].

4.2.1 Conventional MP2, MP3 and MP4

Conventional Møller–Plesset perturbation theory [37] to second (MP2) and third (MP3) order are available for both RHF and UHF references, and MP4 is implemented for restricted calculations.

An MP2 calculation can be requested by:

```
TUNA SPE : H H 1.0 : MP2 6-311G :
```

If your molecule has an even number of electrons but an unrestricted reference is desired, the method keywords `UMP2` and `UMP3` can be used:

```
TUNA SPE : H H 1.5 : UMP3 6-311G :
```

All correlated wavefunction methods require an initial $\mathcal{O}(N^5)$ transformation of the two-electron integrals into the molecular orbital basis. Calculations with MP3 and MP4 also require slower contractions to calculate the final energies. To speed these calculations up, core electrons can be excluded from the correlation treatment which only loses a small proportion of the accuracy.

Mathematically, this initial transformation into the spatial orbital basis involves matrix multiplications with the molecular orbital coefficients,

$$(ia|jb) = \sum_{\mu\kappa\nu\lambda} M^i_\mu M^a_\kappa M^j_\nu M^b_\lambda (\mu\kappa|\nu\lambda) \,. \tag{4.24}$$

This is the rate-limiting step of an MP2 calculation, and necessary for all correlated wavefunction methods. The spatial orbital MP2 energy can then be evaluated as

$$E_{\text{MP2}} = \sum_{ijab} \frac{(ia|jb)\left[2(ia|jb) - (ib|ja)\right]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,. \tag{4.25}$$

For unrestricted calculations, the two-electron integrals are instead transformed into the spin orbital basis, where α and β spins are treated separately by spin-blocked orbitals,

$$\langle ij|ab\rangle = \sum_{\mu\kappa\nu\lambda} M_\mu^i M_\kappa^a M_\nu^j M_\lambda^b \langle \mu\nu|\kappa\lambda\rangle \,. \tag{4.26}$$

These integrals are then antisymmetrised by

$$\langle ij||ab\rangle = \langle ij|ab\rangle - \langle ij|ba\rangle \,, \tag{4.27}$$

and the MP2 energy is evaluated as

$$E_{\text{MP2}} = \frac{1}{4} \sum_{ijab} \frac{|\langle ij||ab\rangle|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,. \tag{4.28}$$

The same-spin and opposite-spin contributions to the MP2 energy are calculated by TUNA. For unrestricted calculations, this is decomposed into the energy due to $\alpha\alpha$, $\beta\beta$ and $\alpha\beta$ electron pairs.

The MP4 energy can be decomposed as

$$E_{\text{MP4}} = E_{\text{MP4}}^{\text{singles}} + E_{\text{MP4}}^{\text{doubles}} + E_{\text{MP4}}^{\text{triples}} + E_{\text{MP4}}^{\text{quadruples}} \,. \tag{4.29}$$

The calculation of the triples is the steepest scaling step, at $\mathcal{O}(N^7)$, while other steps scale at most at $\mathcal{O}(N^6)$. Therefore, a speedup can be achieved by simply neglecting the triples, yielding the MP4(SDQ) method. This can be calculated in TUNA using the method type `MP4[SDQ]`:

```
TUNA SPE : H F 1.2 : MP4[SDQ] 6-31G :
```

Similarly, the calculation of singles can be skipped either with the `MP4[DQ]` method.

```
TUNA SPE : C O 1.2 : MP4[DQ] 6-31G :
```

Naturally, conventional MP4 can be requested by the `MP4` method type:

```
TUNA SPE : H H 0.7 : MP4 6-31G :
```

In all cases, TUNA prints the energy from the single, double, triple and quadruple excitations.

4.2.2 Freezing Core Electrons

Core electrons, like 1s electrons in second period elements, have a fairly consistent correlation energy between molecular geometries. This core correlation can often be safely ignored with the "frozen core" approximation. This is requested in TUNA with the `FREEZECORE` keyword.

Using `FREEZECORE` on its own will freeze the default number of core electrons depending on the elements used in the calculation. This number is 0 for H–Be, 2 for B–Mg and 10 for Al–Ar.

Alternatively, if a specific number of *orbitals* (not electrons) is wanted to be frozen, this number can be given after the `FREEZECORE` keyword. For example, the following calculation will freeze the three lowest energy spatial orbitals of nitric chloride:

```
TUNA SPE : N Cl 1.0 : MP2 6-311G :  FREEZECORE 3
```

Note that for unrestricted calculations there are twice the number of orbitals to freeze as in restricted calculations. During the transformation to a molecular orbital basis, TUNA will print the number of frozen orbitals — which will be half the number of frozen electrons for RHF references, or the same as the number of frozen electrons for UHF references.

4.2.3 Spin-component Scaling

Some Møller–Plesset perturbation theory methods can be "spin-component-scaled" (SCS), which has been reported to increase their accuracy [38, 39]. The total MP2 energy can be written in terms of same-spin and opposite-spin components,

$$E_{\text{MP2}} = E_{\text{MP2}}^{\text{SS}} + E_{\text{MP2}}^{\text{OS}} \,. \tag{4.30}$$

The SCS-MP2 approximation applies different scaling factors to the same spin, $E_{\text{MP2}}^{\text{SS}}$, and opposite spin, $E_{\text{MP2}}^{\text{OS}}$, contributions to correlation. These calculations can be called in TUNA by:

```
TUNA SPE : H He 1.2 : SCS-MP2 6-31G
```

In SCS-MP2, the correlation energy is given by

$$E_{\text{SCS-MP2}} = c_{\text{S}} E_{\text{MP2}}^{\text{SS}} + c_{\text{O}} E_{\text{MP2}}^{\text{OS}} \,, \tag{4.31}$$

where by default, $c_{\text{S}} = 1/3$ and $c_{\text{O}} = 6/5$. These values can be changed by the `SSS` and `OSS` keywords respectively. The following calculation removes the same-spin scaling entirely, and

increases the opposite spin scaling:

```
TUNA SPE : H H 0.74 : SCS-MP2 6-31G : SSS 0 OSS 1.5
```

In SCS-MP3, which often improves on conventional MP3, the total energy is given by

$$E_{\text{SCS-MP3}} = E_{\text{SCS-MP2}} + c_{\text{MP3}} E_{\text{MP3}} \, , \tag{4.32}$$

where $c_{\text{MP3}} = 1/4$ by default. The value of c_{MP3} can be changed with the `MP3S` keyword. For example a value of $c_{\text{MP3}} = 0.5$ is requested by:

```
TUNA SPE : H He 1.2 : SCS-MP3 6-31G :  MP3S 0.5
```

4.2.4 Orbital-optimised MP2

While the conventional theory is applied to the Hartree–Fock state, the MP2 energy can be minimised with respect to orbital rotations, in orbital-optimised MP2 [40, 41]. This can be requested with the `OMP2` method keyword, or `UOMP2` to force an unrestricted calculation:

```
TUNA SPE : H H 0.6 : OMP2 6-311++G :
```

The correlation energy is determined using the one- and two-particle reduced density matrices at each OMP2 iteration, where the molecular orbitals are updated using a rotation matrix, $\boldsymbol{\Theta}$, by

$$\mathbf{M}_{n+1} = \mathbf{M}_n \exp(\boldsymbol{\Theta}) \, . \tag{4.33}$$

This cycle continues until the change in energy falls below a convergence criteria of 10^{-8} by default. This value can be changed by using the `ECONV` keyword. The maximum number of OMP2 iterations is 30 by default, mutable with the `CORRMAXITER` keyword. The density matrix from orbital-optimised MP2 is fully relaxed, and is used to calculate molecular properties.

4.2.5 Iterative MP2

Conventional MP2 requires a Hartree–Fock reference. For calculations with non-canonical orbitals (such as those using localised molecular orbitals), the MP2 energy cannot be calculated in the conventional way, but can be evaluated through minimisation of the Hylleraas functional [42].

In iterative MP2, the following residuals are iteratively minimised

$$R_{ij}^{ab} = (ia|jb) + F_{ac}\, t_{ij}^{cd}\, S_{db} + S_{ac}\, t_{ij}^{cd}\, F_{db} - S_{ac}\, F_{ik}\, t_{kj}^{cd}\, S_{db} - S_{ac}\, F_{kj}\, t_{ik}^{cd}\, S_{db} \, . \tag{4.34}$$

Where the overlap matrix elements allow for the possibility of a non-orthogonal basis and Fock matrix elements allow for the possibility of non-canonical molecular orbitals.

The doubles amplitudes,

$$t_{ij}^{ab} = \frac{(ia|jb)}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,, \tag{4.35}$$

are updated as

$$t_{ij}^{ab} \leftarrow t_{ij}^{ab} + \frac{R_{ij}^{ab}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,. \tag{4.36}$$

The pair energy for each ij occupied orbital pair can then be calculated as

$$e_{ij} = \sum_{ab} \left[(ia|jb) + R_{ij}^{ab}\right](4t_{ij}^{ab} - 2t_{ji}^{ab}) \,, \tag{4.37}$$

and then the total MP2 energy is a sum of these pair energies

$$E_{\text{MP2}} = \frac{1}{2}\sum_{ij} e_{ij} \,. \tag{4.38}$$

This process repeats until the difference in energy falls below a convergence criterion, changed with `ECONV`. The maximum number of iterations can be changed with the `CORRMAXITER` keyword.

An iterative MP2 calculation on CO, with a reduced number of maximum iterations is:

```
TUNA SPE : C O 1.6 : IMP2 def2-SVP : ECONV 1E-10 CORRMAXITER 8
```

Iterative MP2 is only available, as a pilot implementation, for restricted references. The unrelaxed MP2 density matrix is also calculated and used for molecular property calculations.

4.2.6 Laplace Transform AO-MP2

Conventional MP2 requires molecular orbitals and eigenvalues from diagonalisation of the Fock matrix. However, as matrix diagonalisation has formal scaling $\mathcal{O}(N^3)$, this cannot be used for large systems. One route towards linear scaling MP2 is to exploit the Laplace transform for the energy denominators and express the MP2 energy as a functional of the HF density matrix [43].

Using the Laplace identity for energy denominators,

$$\frac{1}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} = \int_0^\infty e^{-s(\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j)} \, \mathrm{d}s \,, \tag{4.39}$$

the MP2 correlation energy can be written as an integral over the parameter s without explicit orbital energies. The implementation in TUNA forms energy-weighted density matrices

$$\mathbf{X}(s) = \exp(s\mathbf{PF})\,\mathbf{P} \qquad \mathbf{Y}(s) = \exp(-s\mathbf{QF})\,\mathbf{Q}\,, \tag{4.40}$$

where $\mathbf{Q}$ is the "hole" density matrix, containing unoccupied orbital information, and contracts them with atomic orbital basis electron repulsion integrals to obtain an integrand

$$e(s) \;=\; \sum_{mnl\sigma}\sum_{gdke} X_{mg}(s)\,Y_{nd}(s)\,X_{kl}(s)\,Y_{e\sigma}(s)\;(gd|ke)\;L_{mnl\sigma}, \tag{4.41}$$

which can be integrated by numerical quadrature

$$E_{\mathrm{MP2}} = -\int_0^\infty e(s)\,\mathrm{d}s\,. \tag{4.42}$$

The $[0,\infty)$ integral is mapped from $r \in (0,1)$ via $s = g(r)$ and evaluated by a τ-point quadrature,

$$E_{\mathrm{MP2}} \approx -\frac{1}{\tau+1}\sum_{k=1}^{\tau} e(s_k)\,g'(r_k), \qquad r_k = \frac{k}{\tau+1}, \qquad s_k = g(r_k),$$

where $g'(r_k) = \mathrm{d}s/\mathrm{d}r$ is the analytic weight used in the TUNA. The Euler–Maclaurin quadrature method is used which needs very few quadrature points to reach microhartree precision [44].

The number of quadrature points can be changed with the `MPGRID` keyword which is 10 by default. An example of a calculation using AO-MP2 with 5 grid points is:

```
TUNA SPE : H H 0.6 : AO-MP2 6-311++G : MPGRID 5
```

This method is a work in progress and is currently only available for restricted references. There is no current implementation of a Laplace transform AO-MP2 density matrix.

4.2.7 Unrelaxed and Relaxed Density Matrices

The approximate doubles amplitudes are used to update the occupied–occupied and virtual–virtual blocks of the density matrix to form the "unrelaxed" density, which is calculated automatically during an MP2 calculation. Specifically, the occupied–occupied and virtual–virtual blocks are

$$P_{ij} = -\frac{1}{2}\sum_{abk} t_{jk}^{ab} t_{ik}^{ab}\,, \tag{4.43}$$

and

$$P_{ab} = \frac{1}{2} \sum_{ijc} t_{ij}^{ac} t_{ij}^{bc} \ . \tag{4.44}$$

This unrelaxed density is evaluated for all MP2-based methods. For MP3 or MP4 calculations, the MP2 density is evaluated instead. The unrelaxed density is used for population analysis and evaluating multipole moments. If the additional print keyword `P` is used, this density is also used to compute the expectation value of the $\hat{S}^2$ operator to evaluate the spin contamination.

The relaxed MP2 density matrix, which includes the orbital response contributions in the occupied–virtual blocks, is also available and can be requested using the `RELAXED` keyword. This response density yields the correct analytic properties (such as molecular dipole moments) that precisely match the results obtained from numerical differentiation of the MP2 energy.

Computing the relaxed density is more demanding than the unrelaxed density because it requires solving the coupled-perturbed Hartree–Fock equations to account for orbital relaxation,

$$\sum_{jb} H_{ia,jb} Z_{jb} = -L_{ia} \ , \tag{4.45}$$

where $H_{ia,jb}$ represents the orbital Hessian and L_{ia} is the MP2 Lagrangian. The solutions to this linear system, Z_{ia}, define the occupied–virtual blocks of the relaxed contribution to the density.

These examples demonstrate the difference. Using `RELAXED` yields an analytic MP2 dipole moment that exactly matches numerical energy derivatives with `DIPOLE`, whereas the unrelaxed example does not. The relaxed analytical (or numerical) dipole moment is the correct way around, with the negative charge on the oxygen whereas the unrelaxed dipole makes carbon more negative.

```
TUNA SPE : C O 1.1 : MP2 6-31G : UNRELAXED DIPOLE
```

```
TUNA SPE : C O 1.1 : MP2 6-31G : RELAXED DIPOLE
```

Both the unrelaxed and relaxed density matrices are compatible with the frozen core approximation and spin-component scaling. Natural orbitals from MP2 can be calculated using either density:

```
TUNA SPE : H H 1.4 : MP2 6-31G : RELAXED NATORBS PRINTMOS
```

4.2.8 The GW Approximation

The GW approximation is a many-body perturbation theory method that corrects orbital energies so that they better match ionisation potentials and electron affinities. Koopmans' theorem approximates these as minus the HOMO and LUMO energies, but ignores how the other electrons relax when an electron is removed or added, and electron correlation. GW includes both through the electronic self-energy,

$$\Sigma = iGW, \tag{4.46}$$

where G is the one-particle Green's function and W is the screened Coulomb interaction. The self-energy splits into the Fock exchange, $\Sigma_{\mathrm{x}} = iGv$, and a correlation part, $\Sigma_{\mathrm{c}} = iG(W - v)$, where v is the bare Coulomb interaction. The exact reference is still a full IP or EA calculation, but GW estimates the IP or EA of every orbital from a single calculation on the neutral molecule.

Hartree–Fock uses the bare Coulomb operator v, whereas GW uses the screened interaction W. Screening happens because the other electrons respond to an added or removed charge. In terms of orbitals, this response is a set of occupied-to-virtual transitions with energies

$$\omega_{ia} = \varepsilon_a - \varepsilon_i \ , \tag{4.47}$$

where i labels occupied and a labels virtual orbitals. Small excitation energies give strong screening and large ones give weak screening.

The approach taken in TUNA is $\mathrm{G_0W_0}$, where G and W are built once from a Hartree–Fock or Kohn–Sham DFT starting point, and the GW equations are not solved self-consistently.

First, the two-electron integrals are transformed to the molecular orbital basis and used to form the excitation ($\mathbf{A}$) and de-excitation ($\mathbf{B}$) blocks of the orbital Hessian. Diagonalising these gives the excitation energies Ω_s, and eigenvectors $\mathbf{X}_s$ and $\mathbf{Y}_s$, of the collective excitations (plasmons). This is equivalent to finding the time-dependent Hartree–Fock excited states excluding the exchange interaction.

Next, the Coulomb interaction between each pair of molecular orbitals and the transition density of each excitation is calculated,

$$M_s^{pq} = \sum_{ia} (pq|ia)\,(X_{ia}^s + Y_{ia}^s) \ , \tag{4.48}$$

where i and a label occupied and virtual orbitals. The correlation self-energy of orbital p is then

$$\Sigma_{\mathrm{c}}^{p}(\omega) = \sum_{s} \left[\sum_{i} \frac{(M_{s}^{pi})^{2}}{\omega - \varepsilon_{i} + \Omega_{s}} + \sum_{a} \frac{(M_{s}^{pa})^{2}}{\omega - \varepsilon_{a} - \Omega_{s}} \right] , \tag{4.49}$$

and Newton's method is used to solve for the new orbital energy,

$$\varepsilon_{p}' = \varepsilon_{p} + \Sigma_{\mathrm{x}}^{p} - v_{\mathrm{xc}}^{p} + \Sigma_{\mathrm{c}}^{p}(\varepsilon_{p}') , \tag{4.50}$$

where Σ_{x}^{p} and v_{xc}^{p} are the Fock exchange and the exchange–correlation potential of the starting calculation, for orbital p — for Hartree–Fock these are equal, so they cancel. The correction and new value for each orbital energy are printed, along with the corrected ionisation potential, electron affinity and HOMO–LUMO gap.

G_0W_0 can be applied to a Hartree–Fock or DFT calculation, including hybrid functionals:

```
TUNA SPE : H H 1.0 : PBE0 6-311G : GW
```

The result depends on the starting point, as Hartree–Fock tends to give a HOMO–LUMO gap that is too large and PBE one that is too small, so it is worth comparing several:

```
TUNA SPE : C O 1.128 : HF 6-311G : GW
```

```
TUNA SPE : C O 1.128 : PBE 6-311G : GW
```

```
TUNA SPE : C O 1.128 : PBE0 6-311G : GW
```

The GW ionisation potential can be compared with a full IP calculation at the same geometry:

```
TUNA IP : C O 1.128 : HF 6-311G : VERTICAL
```

In this case, correcting the PBE0 quasiparticle energies with GW gives an ionisation potential in agreement with the exact calculation to two significant figures. The PBE0 reference performs better than either PBE or Hartree–Fock.

4.2.9 Random Phase Approximation

The random phase approximation is a method from condensed matter physics that approaches electron correlation differently to wavefunction methods, based on the response of the electron density to fluctuations in the Coulomb potential.

Instantaneous movement of an electron slightly changes the density, which leads to a slight change in the potential, which changes the density,

$$\delta n \to \delta v \to \delta n \to \delta v \to \cdots \tag{4.51}$$

The random phase approximations adds up this feedback process to infinite order. Unlike the Møller–Plesset perturbation series, correlated modulations of the exchange term are not included.

The RPA can be applied to either Hartree–Fock orbitals with the `RPA` method, or to Kohn–Sham orbitals with the `RPA` keyword. The RPA method begins with a time-dependent Hartree–Fock calculation without the exchange term, to determine the set of excitation energies, $\{\omega_n\}$. Then the RPA energy is simply

$$E_{\mathrm{RPA}} = \frac{1}{2}\sum_n \omega_n - \frac{1}{2}\mathrm{Tr}\,\mathbf{A}\ . \tag{4.52}$$

The RPA correlation energy can therefore be simply interpreted as the lowering of the excitation spectrum due to accounting for orbital deexcitation. It is the difference in zero-point energy of uncoupled particle-hole oscillators and the zero-point energy of coupled excitation–deexcitation processes.

4.3 Configuration Interaction

As the Hartree–Fock state cannot encode electron–electron correlation, configuration interaction (CI) constructs the many-body wavefunction as a superposition of the Hartree–Fock Slater determinant with various excited determinants.

These excited determinants are generated by promoting electrons from occupied spin orbitals (denoted by indices $i, j, \dots$) to virtual orbitals ($a, b, \dots$),

$$|\Psi\rangle = c_{\mathrm{HF}}|\Phi_{\mathrm{HF}}\rangle + \sum_{ia} c_i^a|\Phi_i^a\rangle + \sum_{i<j}\sum_{a<b} c_{ij}^{ab}|\Phi_{ij}^{ab}\rangle + \cdots \tag{4.53}$$

4.3.1 Full Configuration Interaction

When this linear combination includes every possible determinant — every way of distributing the electrons among the spin orbitals — the result is "full configuration interaction" (FCI), the exact wavefunction within the one-electron basis set.

This can be realised very simply by TUNA. The full Hamiltonian matrix is constructed,

$$\mathbf{H} = \begin{bmatrix} \langle \Phi_{\mathsf{HF}} | \hat{H} | \Phi_{\mathsf{HF}} \rangle & \langle \Phi_{\mathsf{HF}} | \hat{H} | \Phi_j^b \rangle & \langle \Phi_{\mathsf{HF}} | \hat{H} | \Phi_{kl}^{cd} \rangle & \cdots \\ \langle \Phi_i^a | \hat{H} | \Phi_{\mathsf{HF}} \rangle & \langle \Phi_i^a | \hat{H} | \Phi_j^b \rangle & \langle \Phi_i^a | \hat{H} | \Phi_{kl}^{cd} \rangle & \cdots \\ \langle \Phi_{ij}^{ab} | \hat{H} | \Phi_{\mathsf{HF}} \rangle & \langle \Phi_{ij}^{ab} | \hat{H} | \Phi_k^c \rangle & \langle \Phi_{ij}^{ab} | \hat{H} | \Phi_{kl}^{cd} \rangle & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{4.54}$$

using the Slater–Condon rules. These matrix elements are quite simple. For instance, by Brillouin's theorem $\langle \Phi_{\mathsf{HF}} | \hat{H} | \Phi_i^a \rangle = 0$, while $\langle \Phi_{\mathsf{HF}} | \hat{H} | \Phi_{ij}^{ab} \rangle = \langle ij||ab \rangle$, and any element between determinants differing by more than two spin orbitals vanishes. Diagonalising this matrix gives the full spectrum of states, with the energies and the CI coefficients $\{c_{\mathsf{occ}}^{\mathsf{virt}}\}$ for each. This is the exact solution to the (non-relativistic, Born–Oppenheimer) Schrödinger equation within the basis set.

However, the number of determinants grows combinatorially with the numbers of electrons and orbitals, so even storing the full Hamiltonian — let alone diagonalising it — quickly becomes impossible. The FCI method in TUNA is faster than iterative CI or CC methods for small basis sets, but becomes unusable for larger ones. It therefore exists mostly as a reference method, although it is compatible with all calculation types. The one-electron reduced density matrix is also calculated automatically, so properties such as dipole moments and transition intensities can be calculated at the FCI level. Natural orbitals can be requested with the `NATORBS` keyword.

The number of determinants is printed, and if it exceeds 50000 the calculation will not proceed. Even below this limit, be prepared to wait a long time. The weight of the reference determinant, c_{HF}, is printed at the end — the closer this is to 1, the better Hartree–Fock approximates the correlated state. The FCI method with natural orbitals is requested by:

```
TUNA SPE : Be H 1.0 : FCI STO-3G : NATORBS
```

4.3.2 Configuration Interaction Singles and Doubles

Once $\{c_{\mathsf{occ}}^{\mathsf{virt}}\}$ are known, the molecular energy is

$$E = E_{\mathsf{HF}} + \frac{1}{4} \sum_{ijab} c_{ij}^{ab} \langle ij||ab \rangle \, . \tag{4.55}$$

However, because the number of possible determinants scales factorially with system size, full configuration interaction is computationally intractable for all but the smallest systems. Consequently, the expansion is typically truncated at a practical level of excitation, such as CI with singles and

doubles (CISD). Systematically retaining higher-order excitations progressively increases the accuracy of the final computed energy. All the same keywords for coupled cluster calculations to speed up convergence can be used for truncated configuration interaction calculations.

Full configuration interaction for two-electron systems like the helium atom or hydrogen molecule only requires single and double excitations, where

$$|\Psi\rangle \approx c_{\mathsf{HF}}|\Phi_{\mathsf{HF}}\rangle + \sum_{ia} c_i^a|\Phi_i^a\rangle + \frac{1}{4}\sum_{ijab} c_{ij}^{ab}|\Phi_{ij}^{ab}\rangle \ . \tag{4.56}$$

This is a more efficient alternative to full CI compared to coupled cluster for these systems, as disconnected terms do not need to be calculated, although the computational scaling remains the same at $\mathcal{O}(N^6)$. A configuration interaction singles and doubles (CISD) optimisation and harmonic frequency calculation on the hydrogen molecule, within the cc-pVTZ basis, is called by:

```
TUNA OPTFREQ : H H 0.7 : CISD cc-pVTZ :
```

This gives the same answer as the CCSD calculation, in about half the time, and is available in TUNA for both restricted and unrestricted Hartree–Fock references.

Although more efficient for two-electron systems, CISD is rarely used for many-electron molecules due to its lack of size consistency — the energy of two well-separated molecules is not twice the energy of one isolated molecule. The preferred approach, coupled cluster theory, is size consistent.

4.3.3 Configuration Interaction Doubles

An approximation to CISD is configuration interaction doubles, where

$$|\Psi\rangle \approx c_{\mathsf{HF}}|\Phi_{\mathsf{HF}}\rangle + \frac{1}{4}\sum_{ijab} c_{ij}^{ab}|\Phi_{ij}^{ab}\rangle \ . \tag{4.57}$$

This approach is essentially a non-perturbative version of MP2 — the same space of double excitations is used, but these are refined iteratively. A configuration interaction doubles (CID) optimisation and frequency calculation on the hydrogen molecule, within cc-pVTZ, is requested:

```
TUNA OPTFREQ : H H 0.7 : CID cc-pVTZ :
```

4.3.4 Configuration Interaction Singles, Doubles and Triples

The CISDT method is exact for three-electron systems, and is defined by

$$|\Psi\rangle \approx c_{\mathrm{HF}}|\Phi_{\mathrm{HF}}\rangle + \sum_{ia} c_i^a|\Phi_i^a\rangle + \frac{1}{4}\sum_{ijab} c_{ij}^{ab}|\Phi_{ij}^{ab}\rangle + \frac{1}{8}\sum_{ijkabc} c_{ijk}^{abc}|\Phi_{ijk}^{abc}\rangle \tag{4.58}$$

This is available for both RHF and UHF references, but CISDT is not spin-adapted in TUNA. Like all truncated CI methods, CISDT is not size consistent so CCSDT is preferred for molecules.

```
TUNA OPTFREQ : Li H 1.0 : CISDT cc-pVDZ :
```

4.3.5 Complete Active Space Configuration Interaction

Rather than distributing all electrons among all orbitals, complete active space configuration interaction (CASCI) performs FCI only within a chosen subset of orbitals. The orbitals are partitioned into a doubly occupied core, an active space, and an empty secondary space. All determinants that can be formed by distributing the n active electrons among the m active orbitals are included, denoted CAS(n, m). The active space Hamiltonian is then built and diagonalised exactly as in full configuration interaction.

The number of determinants now depends only on the size of the active space, not the full basis, so larger basis sets become accessible. The price is that the Hartree–Fock orbitals are used as they are, without reoptimisation, so the result depends strongly on which orbitals are chosen as active. CASCI captures static correlation within the active space but almost none of the dynamic correlation outside it. As with FCI, the one-electron density matrix, one-electron properties, natural orbitals and the weight of the reference determinant are all available.

To use CASCI with six electrons in six orbitals, use the mandatory `NEL` and `NORB` keywords.

```
TUNA SPE : N N 1.1 : CASCI cc-pVDZ : NEL 6 NORB 6
```

The convergence criteria can be changed with the usual `ECONV` and `DIISERR` keywords.

4.3.6 Complete Active Space Self-consistent Field

Complete active space self-consistent field (CASSCF) removes the main weakness of CASCI by optimising the orbitals and the CI coefficients together. The wavefunction is written as

$$|\Psi\rangle = e^{-\hat{\kappa}}\sum_{I} c_I|\Phi_I\rangle, \tag{4.59}$$

where the c_I are the CI coefficients of the CAS determinants $|\Phi_I\rangle$, and $\hat{\kappa}$ is an operator with parameters κ_{pq} that rotates the orbitals. The energy is made stationary with respect to rotations between the core, active and virtual spaces, as well as with respect to the CI coefficients.

Rotations within the active space leave the energy unchanged, since the CAS wavefunction is invariant to them, so they need not be included. The calculation therefore alternates between diagonalising the active-space Hamiltonian, as in CASCI, and rotating the orbitals using the orbital gradient

$$g_{pq} = \frac{\partial E}{\partial \kappa_{pq}} = 2\left(F_{pq} - F_{qp}\right), \tag{4.60}$$

where $\mathbf{F}$ is the generalised Fock matrix, built from the one- and two-electron density matrices, and the derivative is taken at the current orbitals. This continues until both have converged, at which point $g_{pq} = 0$ for all p and q.

Because the orbitals are relaxed for the correlated wavefunction, CASSCF is much less sensitive to the starting orbitals than CASCI and gives a qualitatively correct description of bond breaking, diradicals and near-degenerate states. The starting guess still matters, though — the optimisation can converge to a local minimum or rotate unwanted orbitals into the active space, so it is worth checking the final active orbitals and their natural occupation numbers.

Each iteration costs roughly one CASCI calculation plus an integral transformation, so CASSCF is considerably more expensive, and like CASCI it recovers little dynamic correlation. The one-electron density matrix, natural orbitals and the weight of the reference determinant are available as usual.

To use CASSCF with six electrons in six orbitals, again use the mandatory `NEL` and `NORB` keywords.

```
TUNA SPE : N N 1.1 : CASSCF cc-pVDZ : NEL 6 NORB 6
```

4.4 Coupled Cluster Theory

In coupled cluster theory, the exact wavefunction is given by an exponential ansatz,

$$|\Psi\rangle = \exp(\hat{T})|\Phi_{\mathrm{HF}}\rangle\,, \tag{4.61}$$

where $|\Phi_{\text{HF}}\rangle$ is the Hartree–Fock reference determinant, and $\hat{T}$ is the cluster operator, which produces a linear combination of excited Slater determinants. This is written as

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \cdots \,, \tag{4.62}$$

where $\hat{T}_1$ is the operator of all single excitations, $\hat{T}_2$ is the operator of all double excitations, etc. In the formalism of second quantisation, these operators can be expressed as

$$\hat{T}_1 = \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i \,, \tag{4.63}$$

and

$$\hat{T}_2 = \frac{1}{4} \sum_{ijab} t_{ij}^{ab} \hat{a}_b^\dagger \hat{a}_j \hat{a}_a^\dagger \hat{a}_i \,. \tag{4.64}$$

In these formulae, $\hat{a}^\dagger$ and $\hat{a}$ are the second quantisation creation and annihilation operators, respectively, and t_i^a and t_{ij}^{ab} are the singles and doubles amplitudes, which are iteratively updated until convergence in a coupled cluster calculation.

Expanding the exponential operator, $\exp(\hat{T})$, as a Taylor series yields

$$\exp(\hat{T}) = 1 + \hat{T} + \frac{1}{2!}\hat{T}^2 + \frac{1}{3!}\hat{T}^3 + \cdots \tag{4.65}$$

which, upon substitution of the definition of T, yields

$$\exp(\hat{T}) = 1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2 + \frac{1}{2}\hat{T}_1\hat{T}_2 + \frac{1}{2}\hat{T}_2\hat{T}_1 + \frac{1}{2}\hat{T}_2^2 + \cdots \,. \tag{4.66}$$

Curtailing $\hat{T}$ yields approximate wavefunctions. For instance curtailing $\hat{T}$ to $\hat{T}_1 + \hat{T}_2$ yields the coupled cluster singles and doubles (CCSD) method, which, as demonstrated in Equation 4.66, includes approximate contributions from higher than double excitations, through so-called "disconnected" excitations like $\hat{T}_1\hat{T}_2$. This means coupled cluster accounts for more electron correlation than configuration interaction curtailed to the same level.

Because single excitations do not mix with the Hartree–Fock determinant, and differences of more than two excitations do not either, the expression for the coupled cluster energy only depends on connected and disconnected double excitations, no matter the curtailing. The difference in energy comes from effects from eg. triples on the converged singles and doubles t-amplitudes.

The unrestricted total coupled cluster energy is calculated by

$$E = E_{\text{HF}} + \frac{1}{4}\sum_{ijab}\langle ij||ab\rangle t_{ij}^{ab} + \frac{1}{2}\sum_{ijab}\langle ij||ab\rangle t_i^a t_j^b \,, \tag{4.67}$$

and the spin-restricted coupled cluster energy is

$$E = E_{\text{HF}} + \sum_{ijab}[2(ia|jb) - (ib|ja)](t_{ij}^{ab} + t_i^a t_j^b) \,. \tag{4.68}$$

In all coupled cluster methods, an iterative method is used to converge the t-amplitudes. Similarly to the Hartree–Fock self-consistent field, DIIS is used by default to dramatically accelerate convergence. The choice of error vector in TUNA to minimise is the difference between subsequent pairs of t-amplitudes, the residual. This extrapolation is on by default and can be turned off with the `NODIIS` keyword, and the number of amplitudes that DIIS will remember (six by default) can be changed with the keyword `DIIS [Num. Amps]`.

Damping of successive t-amplitudes can also be requested with the `CORRDAMP` keyword, whose parameter determines how much old t-amplitudes should be mixed with new ones,

$$t_\mu \leftarrow a\, t_\mu^{\text{old}} + (1-a) t_\mu^{\text{new}} \,. \tag{4.69}$$

This damping is off by default. A CCSD calculation with damping at 50% and DIIS remembering seven sets of t-amplitudes is given by:

```
TUNA SPE : O O 0.9 : CCSD 3-21G : DIIS 7 CORRDAMP 0.50
```

The TUNA output for coupled cluster calculation begins by printing the energy convergence for the coupled cluster iterations, which is mutable with the `ECONV` keyword — this is the same as the SCF convergence by default. In addition to energy convergence, the amplitudes themselves must converge to within 10^{-8}, which can be adjusted with the `AMPCONV` keyword. Next the MP2 energy, which is calculated from the MP2 approximate t_{ij}^{ab} amplitudes, which are used as the guess amplitudes, is printed. Information about the convergence acceleration is then printed, before the iterations begin. Once convergence is achieved, the final correlation energy is shown. A maximum of `CORRMAXITER` iterations are iterated through, which is 100 by default.

The norm of t_i^a and the $\mathcal{T}_1$ diagnostic is then printed, which gives an idea if a single reference calculation is valid [45]. The $\mathcal{T}_1$ diagnostic is related to the norm of the t_i^a amplitudes by

$$\mathcal{T}_1 = \frac{||t_i^a||}{\sqrt{N_{\text{occ}}}} \,, \tag{4.70}$$

where N_{occ} is the number of correlated occupied spin orbitals.

Next, the largest singles and doubles amplitudes are printed, showing the most important excitations out of the Hartree–Fock orbitals. The number of amplitudes shown is ten by default, and can be changed with the `PRINTAMPS` keyword. Finally, the linearised density is calculated and molecular property and population analysis calculations can begin.

All coupled cluster methods — except CC2, CC3, CCSDT(Q) and CCSDTQ which require a restricted Hartree–Fock reference — are available in both spin unrestricted and restricted spin-adapted forms, making calculations on closed-shell singlet molecules much faster.

4.4.1 Coupled Cluster Doubles

In CCD, the cluster operator is curtailed to

$$T = T_2 \,, \tag{4.71}$$

giving equations where t_{ij}^{ab} needs to be optimised. In TUNA, CCD can be requested with the `CCD` method type. A CCD single point energy calculation on carbon monoxide is requested by:

```
TUNA SPE : C O 0.9 : CCD 6-31+G
```

4.4.2 Coupled Cluster Singles and Doubles

In CCSD, the cluster operator is curtailed to

$$T = T_1 + T_2 \,, \tag{4.72}$$

giving equations where t_i^a and t_{ij}^{ab} need to be optimised together [46]. In TUNA, CCSD can be requested with the `CCSD` method type. For example, a CCSD optimisation and harmonic frequency calculation on the hydrogen molecule, in the cc-pVTZ basis set, can be called by:

```
TUNA OPTFREQ : H H 0.7 : CCSD cc-pVTZ
```

For hydrogen, with only two electrons, CCSD corresponds to full configuration interaction so this is the exact frequency (exact *harmonic* frequency, within the basis set, within the Born–Oppenheimer approximation, within non-relativistic quantum mechanics). The CCSD method is quite computationally intensive, with formal $\mathcal{O}(N^6)$ scaling with respect to basis functions.

4.4.3 Coupled Cluster Singles, Doubles and Triples

In CCSDT, the cluster operator is curtailed to

$$T = T_1 + T_2 + T_3 \,, \tag{4.73}$$

giving equations where t_i^a, t_{ij}^{ab} and t_{ijk}^{abc} need to be optimised together [47]. This can be requested using `CCSDT`. For example, a CCSDT single point energy on lithium with cc-pVDZ is called by:

```
TUNA SPE : Li : CCSDT cc-pVDZ :
```

As lithium has three electrons, the CCSDT energy corresponds to the full CI energy. The CCSDT method is very slow, with formal $\mathcal{O}(N^8)$ scaling with respect to basis functions. To improve convergence, CCSDT and CCSDTQ amplitudes that are not physically meaningful — and build up noise — are projected out each coupled cluster iteration. This can reduce the number of necessary iterations by a factor of 5, at little extra cost per iteration.

4.4.4 Coupled Cluster Singles, Doubles, Triples and Quadruples

In CCSDTQ, the cluster operator is curtailed to

$$T = T_1 + T_2 + T_3 + T_4 \,, \tag{4.74}$$

giving equations where t_i^a, t_{ij}^{ab}, t_{ijk}^{abc} and t_{ijkl}^{abcd} need to be optimised together [48, 49]. The CCSDTQ method can be requested using `CCSDTQ`. A CCSDTQ single point calculation on beryllium is:

```
TUNA SPE : Be : CCSDTQ cc-pVDZ
```

As beryllium has four electrons, the CCSDTQ energy corresponds to the full CI energy. The CCSDTQ method is extremely slow and memory intensive, with formal $\mathcal{O}(N^{10})$ scaling with respect to basis functions. This is currently the most intensive electronic structure method possible in TUNA, especially because the convergence of the quadruples amplitudes can be slow for difficult systems. All CCSDTQ calculations must be used with spin-restricted references.

4.4.5 Coupled Cluster with Perturbative Triples

In CCSD(T), often called the "gold standard" of quantum chemistry, a perturbative correction is made to the CCSD energy, based on the MP4 and MP5 triples amplitudes [50, 51]. This provides approximate treatment of triples, which often gives better results than CCSDT, with lower $\mathcal{O}(N^7)$ scaling, due to the undershooting of triples resembling the inclusion of quadruples [45].

Because TUNA is run entirely from the terminal CCSD(T) is called by the `CCSD[T]` method keyword. An optimisation of lithium hydride in the 6-311G** basis is given by:

```
TUNA OPT : Li H 1.0 : CCSD[T] 6-311G**
```

Note that the "CCSD[T]" method in the literature is not the same as the "CCSD(T)" method. Unfortunately, parentheses cannot be used directly in the terminal in Windows, so TUNA changes circular brackets to square brackets. Either will work fine in Linux or MacOS.

4.4.6 Coupled Cluster with Perturbative Quadruples

If the "gold standard" of CCSD(T) is not enough — for example in a multireference situation — the "platinum standard" of CCSDT(Q) can be used, which incorporates approximate connected quadruples excitations through contributions based on MP5 and MP6 [52, 53]. The cost of this method is huge — at $\mathcal{O}(N^9)$, but does not rely on the slow convergence of full CCSDTQ.

The CCSDT(Q) method is called by the `CCSDT[Q]` keyword. A single point calculation on lithium hydride in the cc-pVTZ basis is given by:

```
TUNA SPE : Li H 1.0 : CCSDT[Q] cc-pVTZ
```

4.4.7 Linearised Coupled Cluster

Linearised coupled cluster makes an approximation to $\exp(\hat{T})$, keeping only terms to first order

$$\exp(\hat{T}) \approx 1 + T = 1 + T_1 + T_2 + T_3 + \cdots \tag{4.75}$$

This comes at the cost of size consistency and is normally less accurate than conventional coupled cluster, however computation of the equations becomes much faster.

In TUNA, linearised CCD and linearised CCSD are implemented, and can be requested using the `LCCD` and `LCCSD` method types, respectively. In linearised CCD, $T = T_2$ and in linearised CCSD, $T = T_1 + T_2$. For instance, an optimisation on dinitrogen with LCCD can be requested by:

```
TUNA OPT : N N 1.2 : LCCD 6-31G
```

Linearised coupled cluster coincides with the coupled electron pair approximation (CEPA0). By default, the `CEPA0` method is the same as `LCCSD` in TUNA. A calculation using CEPA0, with single excitations (unshifted) on the carbon atom is shown:

```
TUNA SPE : C : CEPA0 3-21G : ML 3
```

4.4.8 Quadratic Configuration Interaction

A modification to configuration interaction to make it size consistent known as quadratic configuration interaction singles and doubles (QCISD) is implemented in TUNA [45, 54]. This turns out to be the same as a simplified version of CCSD.

The keyword for this method is `QCISD`, and a spin-adapted implementation is available:

```
TUNA OPT : H Li 1.3 : QCISD def2-SVP :
```

Moreover, the method can be combined with perturbative triples with the `QCISD[T]` method.

```
TUNA OPTFREQ : H Li 1.6 : QCISD[T] def2-TZVPPD :
```

4.4.9 Approximate Coupled Cluster

In the approximate coupled cluster singles and doubles method, CC2, the CCSD equations are simplified to only second order in the fluctuation potential, with iterative singles and a perturbative, MP2-like approach to the doubles amplitudes. This reduces the scaling of CCSD to $\mathcal{O}(N^5)$ [55].

Similarly, in the CC3 method the full CCSD equations are iteratively solved, but an approximation to connected triples with an MP4-like second-order triples expression. This reduces the scaling of CCSDT down to $\mathcal{O}(N^7)$, the same as CCSD(T).

These methods are implemented in TUNA with a T1-dressed Hamiltonian which reduces the complexity of the working equations, for spin-restricted references only. A polarisability calculation with CC2 on carbon monoxide is requested by:

```
TUNA SPE : C O 1.0 : CC2 cc-pVDZ : POLAR
```

Similarly, an optimisation and frequency calculation on lithium hydride with the iterative CC3 method can be requested by:

```
TUNA OPTFREQ : Li H 1.0 : CC3 def2-SVP :
```

4.4.10 Linearised Density Matrix

All implemented coupled cluster methods generate a density matrix after the energy is calculated, which is then used for property calculations such as natural orbitals, spin contamination, dipole moment and population analysis. This density can also be used for plotting.

However, bear in mind that this is not the relaxed response density, nor the full unrelaxed density – the implemented density is linearised. This means that only terms of t-amplitudes up to quadratic are included in the density matrix construction.

4.5 Density Functional Theory

Taking a very different approach to the correlated wavefunction methods, density functional theory (DFT) aims to recover electron–electron correlation directly from the electron density rather than the complex many-body wavefunction. Because the electron density depends on only three spatial coordinates — regardless of the number of electrons in the system — DFT offers a highly favourable balance between computational cost and physical accuracy.

In the widely used Kohn–Sham formulation of DFT [56, 57], the interacting many-body problem is mapped onto a mathematically equivalent, fictitious system of non-interacting electrons. The total electronic energy is expressed as a functional of the electron density, $n(\mathbf{r})$, by

$$E[n] = T_s[n] + V_{\mathsf{ne}}[n] + J[n] + E_{\mathsf{xc}}[n] \; . \tag{4.76}$$

Here, the terms represent the non-interacting kinetic energy (T_s), the external nuclear–electron attraction (V_{ne}), the classical Coulomb repulsion between electrons (J), and the exchange–correlation energy (E_{xc}), which accounts for quantum mechanical effects. To compute the kinetic energy term Kohn and Sham introduced a set of independent-particle orbitals [57]. In TUNA, $T_s[n]$, $V_{\mathsf{ne}}[n]$, and $J[n]$ are calculated from analytical integrals over these Kohn–Sham orbitals.

The ground state electron density is constructed from the occupied Kohn–Sham orbitals,

$$n(\mathbf{r}) = \sum_{i}^{\mathsf{occ}} |\psi_i(\mathbf{r})|^2 \; . \tag{4.77}$$

The $E_{\mathsf{xc}}[n]$ term accounts for all non-classical many-body effects, including quantum mechanical exchange, dynamic electron correlation, and the small correction to the kinetic energy not captured

by $T_s[n]$. The implementation in TUNA defines all exchange–correlation functionals as

$$E_{\mathrm{xc}}[n] = \int f_{\mathrm{xc}}[n]\,\mathrm{d}\mathbf{r} \tag{4.78}$$

where $f_{\mathrm{xc}}[n] = n e_{\mathrm{xc}}[n]$ with e_{xc} being the exchange–correlation energy density per particle. Because the exact mathematical form of E_{xc} remains unknown, various approximations are employed to evaluate this term numerically over a grid.

At self-consistency, the Kohn–Sham equations [57],

$$\left(-\frac{1}{2}\nabla^2 + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\mathrm{d}\mathbf{r}' + \frac{\delta E_{\mathrm{xc}}}{\delta n} + v_{\mathrm{ext}}(\mathbf{r}) \right) \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r}) \,, \tag{4.79}$$

are satisfied, where $\psi_i(\mathbf{r})$ and ε_i are the Kohn–Sham orbitals and energies and $v_{\mathrm{ext}}(\mathbf{r})$ is the external potential. Practically, the Roothaan–Hall equations from Hartree Fock are modified so

Table 4.3 Exchange functionals in TUNA

Functional	Description
`S` [58]	Slater LDA exchange
`B88` [59]	Becke GGA exchange
`PBE` [60]	Perdew–Burke–Ernzerhof GGA exchange
`revPBE` [61]	Revised Perdew–Burke–Ernzerhof GGA exchange
`RPBE` [62]	Modified Perdew–Burke–Ernzerhof GGA exchange
`PW91` [63]	Perdew–Wang GGA exchange
`mPW` [64]	Modified Perdew–Wang GGA exchange
`B97` [65]	Becke GGA exchange
`B97-D` [66]	Dispersion-corrected Becke GGA exchange
`TPSS` [67]	Tao–Perdew–Staroverov–Scuseria *meta*-GGA exchange
`revTPSS` [68]	Revised Tao–Perdew–Staroverov–Scuseria *meta*-GGA exchange
`SCAN` [69]	Strongly constrained and appropriately normed *meta*-GGA exchange
`rSCAN` [70]	Regularised SCAN *meta*-GGA exchange
`r`^{2}`SCAN` [71]	Regularised and restored SCAN *meta*-GGA exchange
`B97M-V` [72]	Non-local dispersion-corrected Becke *meta*-GGA exchange

Table 4.4 Correlation functionals in TUNA

Functional	Description
`VWN3` [73]	Vosko–Wilk–Nusair LDA correlation, parameter set III
`VWN5` [73]	Vosko–Wilk–Nusair LDA correlation, parameter set V
`PW` [74]	Perdew–Wang LDA correlation
`P86` [75]	Perdew 1986 GGA correlation
`PW91` [63]	Perdew–Wang GGA correlation
`PBE` [60]	Perdew–Burke–Ernzerhof GGA correlation
`LYP` [76]	Lee–Yang–Parr GGA correlation
`B97` [65]	Becke GGA correlation
`B97-D` [66]	Dispersion-corrected Becke GGA correlation
`TPSS` [67]	Tao–Perdew–Staroverov–Scuseria *meta*-GGA correlation
`revTPSS` [68]	Revised Tao–Perdew–Staroverov–Scuseria *meta*-GGA correlation
`SCAN` [69]	Strongly constrained and appropriately normed *meta*-GGA correlation
`rSCAN` [70]	Regularised SCAN *meta*-GGA correlation
`r`^{2}`SCAN` [71]	Regularised and restored SCAN *meta*-GGA correlation
`B97M-V` [72]	Non-local dispersion-corrected Becke *meta*-GGA correlation

the spin-restricted Fock matrix is calculated using an exchange–correlation matrix, $\mathbf{K}_{\text{xc}}$

$$\mathbf{F} = \mathbf{T} + \mathbf{V} + 2\mathbf{J} + \mathbf{K}_{\text{xc}} \,, \tag{4.80}$$

and a proportion of Hartree–Fock exchange can be simply implemented through the exchange matrix, $\mathbf{K}$. This Fock matrix can then be used in the standard Hartree–Fock procedure, leveraging the available convergence accelerators in TUNA.

All density functional methods in TUNA can be extended to open-shell systems by using unrestricted references. This is done simply by prepending a "U" to the functional name, which allows α and β electrons to occupy different spatial orbitals. For all standard methods, the energy and density are evaluated iteratively until self-consistency is achieved. For more advanced double-hybrid functionals, which mix exact exchange with perturbative correlation, a second-order Møller–Plesset correction is evaluated and applied only after self-consistency has been reached.

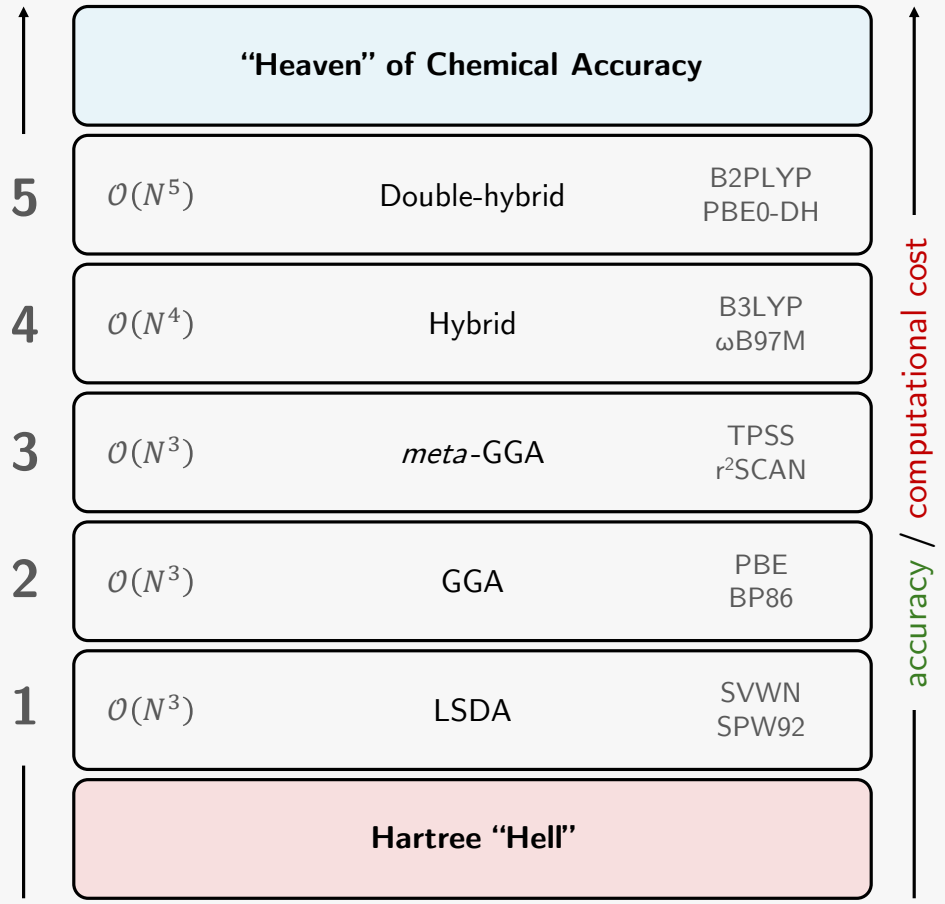

Figure 4.1: The Jacob's ladder of exchange–correlation functionals [77]. Climbing the ladder results in higher computational cost and, on average, higher accuracy

The exchange or correlation contributions can be turned off with the `NOX` or `NOC` keywords. For example, these reproduce the Hartree–Fock–Slater [58] method with no DFT correlation:

```
TUNA SPE : H H 1.0 : LDA 3-21G : NOC
```

```
TUNA SPE : H H 1.0 : LDA 3-21G : DFC 0
```

Table 4.3 and Table 4.4 list all the available exchange and correlation functionals in TUNA. Methods are often thought of as belonging to a certain rung of Perdew's "Jacob's ladder" of exchange–correlation functionals [77], shown in Figure 4.1.

4.5.1 Local Spin Density Approximation

The local density contribution to the exchange–correlation matrix is

$$K_{\mathrm{xc}}^{\mu\nu} = \int \varphi_\mu(\mathbf{r}) \frac{\delta f_{\mathrm{xc}}}{\delta n} \varphi_\nu(\mathbf{r})\, \mathrm{d}\mathbf{r} \,, \tag{4.81}$$

and the energy density is a functional of the local spin densities, $e_{\mathrm{xc}} = e_{\mathrm{xc}}[n_\alpha, n_\beta]$. The implemented local density functionals are listed in Table 4.5. The derivatives with respect to the density needed to form the exchange–correlation matrix have been implemented analytically.

The LSDA functional is equivalent to parameter set V of the VWN exchange–correlation functional, VWN5 [73]. The Perdew–Wang 1992 functional is a simpler and more accurate parameterisation of the correlation energy of the uniform electron gas [74]. Parameter set III for VWN can be requested with the SVWN3 functional [73]. The only local density exchange is that of the uniform electron gas, Slater exchange. This can be used in lieu of Hartree–Fock exchange by the Hartree–Fock–Slater (HFS) method keyword [58].

The exchange energy of the spin-restricted uniform electron gas is

$$E_{\mathrm{x}}[n] = -\alpha \frac{9}{8} \left(\frac{3}{\pi} \right)^{\frac{1}{3}} \int n(\mathbf{r})^{\frac{4}{3}} \mathrm{d}\mathbf{r} \,, \tag{4.82}$$

where $\alpha = 2/3$ by default. The XA keyword can be used to change this, in the X-α method. For example, changing the value of α to 0.7 in a HFS calculation is requested by:

```
TUNA SPE : N N 1.0 : HFS 3-21G : XA 0.7
```

Table 4.5 Local spin density approximation exchange–correlation functionals in TUNA

Functional	$E_{\mathrm{x}}[n]$	$E_{\mathrm{c}}[n]$
LSDA [58, 73]	S	VWN5
SVWN [58, 73]	S	VWN5
SVWN5[58, 73]	S	VWN5
SVWN3 [58, 73]	S	VWN3
SPW [58, 74]	S	PW
HFS [58]	S	—

An LSDA geometry optimisation can be requested in the usual way. Here, a medium integration grid is requested. The LSDA does worse for dihydrogen's bond length compared to Hartree–Fock.

```
TUNA OPT : H H 0.74 : LSDA cc-pVTZ : MEDIUMGRID
```

4.5.2 Generalised Gradient Approximation

The treatment of electronic structure can be improved by incorporating information about the gradients of the spin densities into the exchange–correlation functional, in "generalised gradient approximation" (GGA) functionals,

$$e_{\text{xc}} = e_{\text{xc}}[n_\alpha, n_\beta, \nabla n_\alpha, \nabla n_\beta] \,. \tag{4.83}$$

The GGA contribution to exchange–correlation depends on the derivative of f_{xc} with respect to

$$\sigma(\mathbf{r}) = \nabla n \cdot \nabla n \,. \tag{4.84}$$

It also depends on the basis function gradients, which are calculated analytically by TUNA and expressed on the integration grid,

$$K_{\text{xc}}^{\mu\nu} = \int \varphi_\mu(\mathbf{r}) \frac{\delta f_{\text{xc}}}{\delta \sigma} \nabla \varphi_\nu(\mathbf{r}) \nabla n(\mathbf{r}) \, \mathrm{d}\mathbf{r} \,. \tag{4.85}$$

The implemented GGA functionals are listed in Table 4.6, among which are the popular PBE and BLYP [59, 60, 76], as well as many other combinations of exchange and correlation.

```
TUNA OPT : Li Li 2.2 : BLYP def2-SVP : TIGHTGRID
```

```
TUNA SPE : H H 1.2 : UPBE def2-SVP : PLOTLUMO
```

4.5.3 *meta*-Generalised Gradient Approximation

On the next step along the Jacob's ladder of exchange–correlation functionals [77] are *meta*-GGA functionals, which incorporate implicit dependence on the curvature of the density, $\nabla^2 n$, through the non-interacting kinetic energy density, $\tau(\mathbf{r})$,

$$e_{\text{xc}} = e_{\text{xc}}[n_\alpha, n_\beta, \nabla n_\alpha, \nabla n_\beta, \tau_\alpha, \tau_\beta] \,. \tag{4.86}$$

Table 4.6 Generalised gradient approximation exchange–correlation functionals in TUNA

Functional	$E_\mathrm{x}[n]$	$E_\mathrm{c}[n]$
`SLYP` [58, 76]	S	LYP
`HFB` [59]	B88	—
`BVWN` [59, 73]	B88	VWN5
`BVWN3` [59, 73]	B88	VWN3
`BLYP` [59, 76]	B88	LYP
`PWP` [63, 75]	PW91	P86
`BP86` [59, 75]	B88	P86
`PBE` [60]	PBE	PBE
`revPBE` [78]	revPBE	PBE
`RPBE` [62]	RPBE	PBE
`mPWLYP` [76, 79]	mPW	LYP
`mPWPW` [63, 79]	mPW	PW91
`B97-D` [66]	B97-D	B97-D

The kinetic energy density is calculated as a sum of occupied orbital gradients, so is technically "non-local", but in practice has similar cost to GGA functionals so is normally called "semilocal",

$$\tau(\mathbf{r}) = \frac{1}{2}\sum_{i}^{\mathrm{occ}} |\nabla\psi_i|^2 \,. \tag{4.87}$$

In TUNA, this sum over occupied orbitals is achieved through the density matrix,

$$\tau(\mathbf{r}) = \frac{1}{2}\sum_{\mu\nu} P_{\mu\nu}\nabla\varphi_\mu(\mathbf{r})\nabla\varphi_\nu(\mathbf{r}) \,. \tag{4.88}$$

The kinetic energy density can be incorporated through the dimensionless iso-orbital indicator,

$$\alpha(\mathbf{r}) = \frac{\tau(\mathbf{r}) - \tau_\mathrm{W}(\mathbf{r})}{\tau_\mathrm{U}(\mathbf{r})} \,, \tag{4.89}$$

which compares the kinetic energy density to the von Weizsäcker kinetic energy density of a one orbital system, $\tau_\mathrm{W}(\mathbf{r})$ and the uniform electron gas kinetic energy density, $\tau_\mathrm{U}(\mathbf{r})$. This allows *meta*-GGA functionals to identify regions with one-electron character or slowly varying densities and adjust their behaviour accordingly, thereby satisfying more constraints of the exact exchange–correlation functional compared to GGAs.

Table 4.7 *meta*-Generalised gradient approximation exchange–correlation functionals in TUNA

Functional	$E_\mathrm{x}[n]$	$E_\mathrm{c}[n]$
`TPSS` [67]	TPSS	TPSS
`revTPSS` [68]	revTPSS	revTPSS
`SCAN` [69]	SCAN	SCAN
`rSCAN` [70]	rSCAN	rSCAN
`r2SCAN` [71]	r^{2}SCAN	r^{2}SCAN
`B97M-V` [72]	B97M-V	B97M-V

The exchange–correlation matrix contribution here requires the basis function gradients, as in GGA functionals, but also requires the derivative of the functional with respect to $\tau(\mathbf{r})$.

$$K_\mathrm{xc}^{\mu\nu} = \int \nabla\varphi_\mu(\mathbf{r}) \frac{\delta f_\mathrm{xc}}{\delta\tau} \nabla\varphi_\nu(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{4.90}$$

Table 4.7 lists the *meta*-GGA exchange–correlation functionals in TUNA, which are all available self-consistently for both restricted and unrestricted references. Because *meta*-GGAs rely on second derivatives of the density, they are particularly sensitive to the precision of the integration grid. It is probably a good idea to use `TIGHTGRID` for these functionals for high precision work.

```
TUNA SPE : Li H 1.0 : r2SCAN cc-pVDZ : PLOTHOMO TIGHTGRID
```

The B97M-V functional is unique in that it has been parameterised [72] with non-local VV10 dispersion correction. This dispersion calculation happens at the end of the SCF cycle.

```
TUNA SPE : C O 1.1 : B97M-V def2-SVP :
```

4.5.4 Hybrid Functionals

The next rung up Jacob's ladder is "hybrid", non-local exchange–correlation functionals. A hybrid exchange–correlation functional uses a proportion, β, of the Fock exchange expression evaluated on the Kohn–Sham orbitals

$$E_\mathrm{xc}[n] = \beta E_\mathrm{x}^\mathrm{HF} + (1-\beta)E_\mathrm{x}^\mathrm{KS}[n] + E_\mathrm{c}^\mathrm{KS}[n]\,. \tag{4.91}$$

A number of hybrid functionals are implemented in TUNA, listed in Table 4.8. These include the popular PBE0 and B3LYP functionals. The version of B3LYP used in the Gaussian program, which uses VWN-III LSDA correlation instead of VWN-V, can be requested with `B3LYP/G`.

The proportion of Hartree–Fock exchange can be changed with the HFX keyword, with the desired proportion of exchange. This will not change the proportion of exchange from the DFT expression, which can be changed with the DFX keyword. For example, the PBE0 functional can be built as:

```
TUNA SPE : H H 1.0 : PBE 4-31G : HFX 0.25 DFX 0.75
```

Similarly, the BHLYP functional can be specified by:

```
TUNA SPE : H H 1.0 : BLYP def2-TZVPD : HFX 0.50 DFX 0.50
```

Table 4.8 Hybrid exchange–correlation functionals in TUNA

Functional	$E_x[n]$	$E_c[n]$	HFX
PBE0 [64]	PBE	PBE	0.25
revPBE0 [78]	revPBE	PBE	0.25
revPBE38 [78]	revPBE	PBE	0.38
B1LYP [80]	B88	LYP	0.25
BHLYP [81]	B88	LYP	0.50
B3LYP [82]	B88	LYP	0.20
B3LYP/G [82]	B88	LYP	0.20
B1P86 [59, 75]	B88	P86	0.25
B3P86 [59, 75]	B88	P86	0.20
PW1PW [64]	PW91	PW91	0.25
mPW1PW [64]	mPW91	PW91	0.25
mPW1LYP [64]	mPW91	LYP	0.25
B3PW91 [59, 63]	B88	PW91	0.20
TPSSh [83]	TPSS	TPSS	0.10
TPSS0 [83]	TPSS	TPSS	0.25
SCAN0 [84]	SCAN	SCAN	0.25
r^{2}SCANh [85]	r^{2}SCAN	r^{2}SCAN	0.10
r^{2}SCAN0 [85]	r^{2}SCAN	r^{2}SCAN	0.25
r^{2}SCAN50 [85]	r^{2}SCAN	r^{2}SCAN	0.50
B97 [65]	B97	B97	0.19

4.5.5 Double-hybrid Functionals

The next rung up the ladder contains "double-hybrid" functionals, which additionally incorporate a proportion, γ, of second-order perturbation theory correlation.

$$E_{\mathrm{xc}}[n] = \beta E_{\mathrm{x}}^{\mathrm{HF}} + (1-\beta)E_{\mathrm{x}}^{\mathrm{KS}}[n] + \gamma E_{\mathrm{c}}^{\mathrm{MP2}} + (1-\gamma)E_{\mathrm{c}}^{\mathrm{KS}}[n] \tag{4.92}$$

The MP2 part of the energy is applied only to the converged Kohn–Sham orbitals at the end of the SCF. The unrelaxed and relaxed density matrices are also implemented for double-hybrid functionals, allowing calculation of properties including natural orbitals and dipole and quadrupole moments.

The first double-hybrid functional developed was B2PLYP [86], which incorporates 53% Hartree–Fock exchange and 27% MP2 correlation. This functional can be requested by either of these:

```
TUNA SPE : Li H 1 : B2PLYP 6-311G :
```

```
TUNA SPE : Li H 1 : BLYP 6-311G : HFX 0.53 DFX 0.47 MPC 0.27 DFC 0.73
```

The `MPC` keyword defines the proportion of MP2 correlation, and the `DFC` keyword defines the proportion of DFT correlation. Similarly, the non-empirical PBE0-DH functional can be built by:

```
TUNA SPE : Li H 1 : PBE 6-311G : HFX 0.50 DFX 0.50 MPC 0.125 DFC 0.875
```

A number of double-hybrids are implemented in TUNA, listed in Table 4.9. These include non-empirical double-hybrids like PBE0-DH, r^{2}SCAN0-DH and PBE-QIDH, as well as empirical double-hybrids including B2PLYP, mPW2PLYP and DSD-BLYP. The latter functional, as well as r^{2}SCAN-based double-hybrids, also incorporates spin-component scaling of the MP2 correlation part. The proportion of same- and opposite-spin scaling can be changed with the `SSS` and `OSS` keywords.

The values of `HFX`, `DFX`, `MPC` and `DFC` are printed at the start of a hybrid or double-hybrid DFT calculation's SCF cycle. The values of `OSS` and `SSS` are printed during the MP2 calculation for a spin-scaled double-hybrid, or when those keywords are used otherwise.

The following calculation uses the relaxed spin-component scaled double-hybrid density matrix to calculate the dipole moment, and the `DIPOLE` keyword shows that this analytical result agrees with the numerical calculation:

```
TUNA SPE : C N 1.4 : DSD-BLYP 4-31G : DIPOLE RELAXED
```

Be aware that calculation of the exchange–correlation kernel for the relaxed double-hybrid density is quite slow!

Table 4.9 Double-hybrid exchange–correlation functionals implemented within TUNA.

Functional	$E_x[n]$	$E_c[n]$	HFX	MPC	SSS	OSS
PBE0-DH [87]	PBE	PBE	0.50	1/8	1.00	1.00
PBE-QIDH [88]	PBE	PBE	0.69	0.33	1.00	1.00
PBE0-2 [89]	PBE	PBE	0.79	0.50	1.00	1.00
B2PLYP [86]	B88	LYP	0.53	0.27	1.00	1.00
B2K-PLYP [90]	B88	LYP	0.72	0.42	1.00	1.00
B2T-PLYP [90]	B88	LYP	0.60	0.31	1.00	1.00
B2G-PLYP [90]	B88	LYP	0.65	0.36	1.00	1.00
B2NC-PLYP [91]	B88	LYP	0.81	0.55	1.00	1.00
mPW2PLYP [92]	mPW91	LYP	0.55	0.25	1.00	1.00
DSD-BLYP [93]	B88	LYP	0.75	1.00	0.60	0.46
r^{2}SCAN0-DH [94]	r^{2}SCAN	r^{2}SCAN	0.50	1/8	0.00	4/3
r^{2}SCAN-CIDH [94]	r^{2}SCAN	r^{2}SCAN	0.55	4/3	0.00	4/3
r^{2}SCAN-QIDH [94]	r^{2}SCAN	r^{2}SCAN	0.69	1/3	0.00	4/3
r^{2}SCAN0-2 [94]	r^{2}SCAN	r^{2}SCAN	0.79	0.50	0.00	4/3
Pr^{2}SCAN50 [94]	r^{2}SCAN	r^{2}SCAN	0.50	0.25	0.00	4/3
Pr^{2}SCAN69 [94]	r^{2}SCAN	r^{2}SCAN	0.69	4/9	0.00	4/3

4.5.6 Efficient Grid Integration

Density functional theory calculations involved numerical integrations over a grid, on which the electron density is expressed. The integral of the density should be the same as the number of electrons in the system, and prints for the guess density before the self-consistent field cycle, and the final converged density. If the guess density is far from the ideal, a warning will be printed, and if it is totally wrong the calculation will crash politely.

The tightness of this grid can be controlled directly with the INTACC keyword, which has a default value of 4.0. The extent of the grid (which may be important for large atoms) is controlled via the general integration grid keywords in Table 4.10.

Table 4.10 Integration grid tightness criteria in TUNA

Grid	Integral Accuracy	Grid Range
LOOSEGRID	3.0	0.7
MEDIUMGRID	4.0	0.9
TIGHTGRID	5.0	1.0
EXTREMEGRID	7.0	1.2

Integration on this grid is performed by numerical quadrature. The grids are atom centered, and then Becke weighting is applied to build the molecular grid [95]. The atom centered grids use Gauss–Legendre quadrature for the radial part and Lebedev quadrature for the angular part. The order of Lebedev integration is chosen by the closest Lebedev order available to match the desired integral accuracy. The following is a very high density grid, beyond that from EXTREMEGRID:

```
TUNA SPE : Ar : HFS 6-31G : INTACC 8.0 TIGHTGRID
```

Becke's heteroatomic weighting system is used, to distribute integration weights for heteronuclear systems. This relies on a proxy for atomic size, for which the van der Waals radius is used.

Both the density (and density gradient for GGA functionals) need to be expressed on the grid. This is done through the density matrix, by expressing basis functions and basis function gradients on the grid. The mathematical expressions for the primitive Gaussians, weighted by their coefficients in an atomic orbital, are evaluated on the grid, centered around each atomic basis function centre.

4.6 Excited States

Excited state calculations, to evaluate excitation energies and absorption intensities, are available in TUNA. As with other electronic structure methods, any excited state method can be used with optimisations, vibrational frequency calculations and, in principle, molecular dynamics simulations. In reality, there is significant risk to doing this, as unintended crossings between excited state potential energy surfaces may occur and TUNA currently has no root following algorithm.

Excited state calculations are available for both restricted and unrestricted references. The default SCF convergence criteria is set to TIGHT for all excited state calculations.

Linear response time-dependent Hartree–Fock and density functional theory allow for the calcula-

tion of excitation energies, ω, through the Casida equations,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \omega \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} , \tag{4.93}$$

where the elements of the $\mathbf{A}$ and $\mathbf{B}$ matrices are the same as in stability analysis,

$$A_{ia,jb} = (\varepsilon_a - \varepsilon_i)\delta_{ij}\delta_{ab} + 2(ia|jb) - \beta(ij|ab) + (1-\beta)K^{\mathsf{xc}}_{ijab} \tag{4.94}$$

and

$$B_{ia,jb} = 2(ia|bj) - \beta(ib|aj) + (1-\beta)K^{\mathsf{xc}}_{ijab} , \tag{4.95}$$

where β is the proportion of Hartree–Fock exchange, which is unity for TDHF. In the Casida equations, $\mathbf{X}$ and $\mathbf{Y}$ represent the transition amplitudes for excitation and deexcitation, so an excited many-body wavefunction is

$$|\Psi\rangle = \sum_{ia} X_{ia}|\Phi^a_i\rangle + \sum_{ia} Y_{ia}|\Phi^i_a\rangle . \tag{4.96}$$

For spin-restricted calculations, the orbital Hessian is calculated separately for singlet-to-singlet and singlet-to-triplet excitations. By default, both singlet and triplet excited states are calculated, but can be disabled with the `NOSINGLETS` and `NOTRIPLETS` keywords respectively.

In an excited stated calculation, the excitation energies from the reference determinant to each state is printed, along with the percentage contribution of each occupied-to-virtual transition. The default threshold for the contributions to be printed is 1%, but this can be changed with the `EXTHRESH` parameter. The total number of excited states to print — counted from lowest excitation energy — can be controlled with the `NSTATES` keyword, which has a default value of 10. This will not speed up the calculation, as presently all states are calculated at the same time.

After the percentage contributions are calculated, TUNA determines the transition dipoles between the excited and ground state, $\boldsymbol{\mu}$, and uses these to find the oscillator strengths, by

$$f_{\mathsf{osc}} = \frac{2}{3}\omega|\boldsymbol{\mu}|^2 . \tag{4.97}$$

These values, along with the excitation energy in hartree, eV, and excitation wavelength in nm, are printed in the absorption spectrum. Finally, the unrelaxed density matrix is calculated.

This is added to the ground state density matrix to give the full unrelaxed one-particle reduced density matrix of a particular state. This state can then be investigated further in population

analysis and dipole moment calculations with the additional print keyword, `P`. The state of interest is the lowest energy state by default, but can be changed using the `ROOT` keyword. This determines for which state the energy and unrelaxed density matrix is calculated.

4.6.1 Time-dependent Hartree–Fock

The time-dependent Hartree–Fock method uses both the $\mathbf{A}$ and $\mathbf{B}$ matrices. It can be requested by the `TDHF` method keyword or through the `HF` method with the `TD` keyword.

```
TUNA SPE : H H 1.0 : HF STO-3G : TD
```

The following calculation performs a population analysis using the unrelaxed density matrix of the second excited state — which is the first *singlet* excited state in this case.

```
TUNA SPE : H H 1.0 : TDHF STO-3G : ROOT 2 P
```

4.6.2 Configuration Interaction Singles

The configuration interaction single excitations (CIS) method is the analogue of Hartree–Fock for excited states [96], and therefore is not very good. In CIS, the wavefunction is expressed as a linear combination of singly-excited determinants,

$$|\Psi_{\text{CIS}}\rangle = \sum_{ia} X_i^a |\Phi_i^a\rangle \,, \tag{4.98}$$

where only the $\mathbf{X}$ matrix is used and deexcitations from $\mathbf{Y}$ are ignored. This is equivalent to solving the much simpler eigenvalue problem

$$\mathbf{AX} = \omega \mathbf{X} \,. \tag{4.99}$$

Contributions from the $\mathbf{B}$ matrix of deexcitations are ignored. This is known as the Tamm–Dancoff approximation. Either the `TDA` keyword can be used in addition to `TD`, or the method keyword `CIS`:

```
TUNA SPE : H H 1.0 : HF cc-pVDZ : TD TDA
```

```
TUNA SPE : H H 1.0 : CIS cc-pVDZ :
```

4.6.3 Perturbative Doubles Correction

Analogous to the MP2 correlation correction to the Hartree–Fock ground state energy, CIS(D) is a perturbative doubles correction to an excited state's energy [97]. This has been shown to improve

excitation energies considerably compared to configuration interaction singles.

This method is callable in TUNA with the `CIS[D]` keyword and the correction will be applied to a chosen `ROOT` of interest only, as, like MP2, CIS(D) scales with $\mathcal{O}(N^5)$ with basis functions.

The correction to the excitation energy ω, consists of a "direct" contribution from the u_{ij}^{ab} tensor, which accounts for electron correlation effects of the electron involved in the excitation. The other term is "indirect", accounting for correlation between electrons not involved in the excitation, calculated by contraction of the v_i^a tensor, which contains contributions from the MP2 amplitudes,

$$\omega_{\text{CIS(D)}} = \frac{1}{4}\sum_{ijab}\frac{(u_{ij}^{ab})^2}{\mathcal{E}_{ij}^{ab}+\omega} + \sum_{ia} c_i^a v_i^a \ . \tag{4.100}$$

The state to which the energy correction should be applied is the first excited state by default, but can be changed with the `ROOT` keyword. For example, a CIS(D) optimisation of the fourth excited state of dihydrogen, which is a singlet, is requested by:

```
TUNA OPT : H H 1.0 : CIS[D] 6-311G : ROOT 4
```

The CIS(D) code is spin-adapted so will be fast for restricted references, on either singlet or triplet excited states. It is also available for unrestricted calculations.

The perturbative doubles can also be used with other linear response time-dependent methods with the `[D]` keyword. For instance, a single point energy calculation with a perturbative doubles correction on a time-dependent Hartree–Fock state can be requested by:

```
TUNA SPE : H H 1.0 : HF 6-311G : TD [D]
```

4.6.4 Time-dependent Density Functional Theory

The infrastructure for TDHF can be easily modified to allow TD-DFT calculations, which are requested with the `TD` keyword with the exchange–correlation functional of your choice. Currently, TD-DFT calculations are only available with LSDA functionals, although a proportion of Hartree–Fock exchange can be incorporated with the `HFX` keyword.

The `TDA` keyword activates the Tamm–Dancoff approximation, which is the CIS analogue for TD-DFT, ignoring contributions from deexcitation. This dramatically reduces the computational time for excited state calculations, halving matrix size, so is probably a good idea.

The TD-DFT method works the same as TDHF, except that the contribution of the exchange–correlation kernel matrix, K^{xc}_{ijab}, in Equation 4.94 and Equation 4.95 is not neglected. The matrix elements are integrals over the standard DFT integration grid in TUNA,

$$K^{\text{xc}}_{ijab} = \sum_{\mu\kappa\nu\lambda} \int \psi^{\mu}_{i}(\mathbf{r})\psi^{\kappa}_{a}(\mathbf{r}) f_{\text{xc}}(\mathbf{r})\psi^{\nu}_{j}(\mathbf{r})\psi^{\lambda}_{b}(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{4.101}$$

The integrals run over the occupied and virtual Kohn–Sham orbitals, $\psi(\mathbf{r})$, and the exchange–correlation kernel,

$$f_{\text{xc}} = \frac{\delta^2 E_{\text{xc}}[n]}{\delta n^2}\,. \tag{4.102}$$

These analytical second derivatives are hard-coded in TUNA. The K^{xc}_{ijab} matrix is also needed for orbital Hessian calculations for stability analysis. A TD-DFT calculation with LSDA is requested:

```
TUNA SPE : H H 1.0 : LSDA 6-311G : TD
```

The Tamm–Dancoff approximation to speed up calculations can be requested as:

```
TUNA SPE : H H 1.0 : LSDA 6-311G : TD TDA
```

In addition to speeding up the calculation, the Tamm–Dancoff approximation also allows TUNA to print the $\langle S^2 \rangle$ expectation value for each state.

Hybrid and double-hybrid excited state calculations can also be requested. The following tells TUNA to use 20% of Hartree–Fock exchange, and apply a perturbative doubles correction with 50% (D) correlation and 50% DFT correlation.

```
TUNA SPE : H H 1.0 : LSDA 6-311G : TD HFX 0.2 DFX 0.8 [D] MPC 0.5 DFC 0.5
```

For simplicity, you can use an inbuilt double-hybrid functionals for excited states too:

```
TUNA SPE : H H 1.0 : r2SCAN-QIDH 6-31G : TD
```

4.6.5 Equation of Motion Coupled Cluster Theory

A wavefunction theory approach for calculating excited states is equation of motion coupled cluster, implemented as EOM-CCSD in TUNA [98]. In this method, first the standard CCSD equations are solved to determine the singles and doubles amplitudes. The similarity transformed Hamiltonian,

$$\bar{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}}, \tag{4.103}$$

is formed and diagonalised for its eigenvalues and eigenvectors, which are the excited states. Even for the small system sizes handled by TUNA, full diagonalisation is still too expensive, so an iterative approach is taken.

```
TUNA SPE : H H 1.0 : EOM-CCSD 6-311G :
```

For consistency with the user experience for other excited state methods, the `TD` keyword can alternatively be used with the `CCSD` method for the same results.

```
TUNA SPE : H H 1.0 : CCSD 6-311G : TD
```

4.6.6 Complete Active Space Methods

In TUNA, electronic structure methods which use full configuration interaction — including CASCI and CASSCF — diagonalise the full Hamiltonian and therefore find all excited states, along with the ground state.

To show excited state information from these methods, which can be a useful reference against which to compare TD-DFT or EOM-CCSD states, simply use the `TD` keyword. The excitation energy to each state will be printed, along with its $\langle S^2 \rangle$ expectation value. However no orbital transitions will be printed as, unlike Hartree–Fock or Kohn–Sham DFT, correlated wavefunction methods include more than single excitations.

For example, a CASSCF excited state calculation on carbon monoxide is:

```
TUNA SPE : C O 1.0 : CASSCF def2-SVP : TD NEL 4 NORB 4
```

And a full configuration interaction optimisation of the second excited state of H–He is:

```
TUNA OPT : H He 1.0 : FCI cc-pVDZ : TD ROOT 2
```

4.6.7 Plotting Absorbance Spectra

The calculated transition peaks can be plotted, as a function of wavelength, with the `ABSPLOT` keyword. This works in the same way as coordinate scan plots with `SCANPLOT` in that the `ADDPLOT` and `DELPLOT` keywords allow multiple spectra to be overlaid on the same axes. Any of the colours or line styles for coordinate scan plots can also be used with absorbance plots.

```
TUNA SPE : Li H 1.0 : HF cc-pVTZ : TD ABSPLOT
```

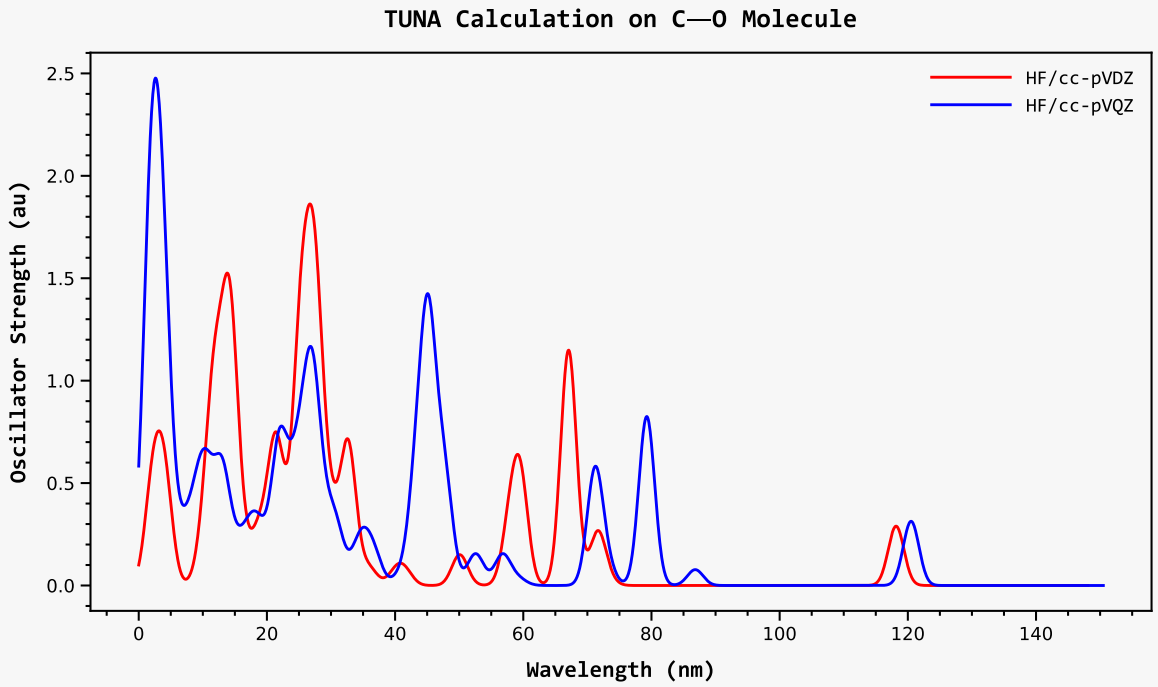

Figure 4.2: Absorbance spectra of carbon monoxide, calculated by TUNA with ABSPLOT

The absorbance peaks are plotted with their calculated oscillator strength by Gaussian functions. The width of these can be modified with the PEAKWIDTH keyword which is 3.0 by default.

```
TUNA SPE : Li H 1.0 : HF cc-pVTZ : TD ABSPLOT PEAKWIDTH 1.0
```

The following two calculations produced Figure 4.2, showing a change on using a larger basis set. Using the DELPLOT keyword in the first case clears any currently saved plots:

```
TUNA SPE : C O 1.0 : HF cc-pVDZ : ABSPLOT ADDPLOT TD RED DELPLOT
```

```
TUNA SPE : C O 1.0 : HF cc-pVQZ : ABSPLOT ADDPLOT TD BLUE
```

5 Basis Sets

Basis sets in TUNA consist of contracted Gaussian functions,

$$\varphi(\mathbf{r}) = \sum_{\gamma} C_{\gamma} g_{\gamma}(\mathbf{r}) \tag{5.1}$$

where $\{g_{\alpha}(\mathbf{r})\}$ are primitive Gaussians,

$$g_{\gamma}(\mathbf{r}) = N x^{l} y^{m} z^{n} e^{-\gamma(x^2+y^2+z^2)} \ , \tag{5.2}$$

where l, m and n are angular momenta and N is a normalisation constant. Using the additional print keyword, `P`, will show the detailed basis data at the start of a TUNA calculation, including the Gaussian exponent γ and contraction coefficient C_{γ} for each basis function.

A basis can be decontracted with the `DECONTRACT` keyword. The following command dedicates the three primitive Gaussians in the hydrogen STO-3G basis to three separate basis functions, rather than a single 1s orbital, giving a lower energy to the hydrogen atom than contracted STO-3G:

```
TUNA SPE : H : HF STO-3G : DECONTRACT
```

5.1 Basis Set Extrapolation

Calculations that approach the complete basis set limit, with quadruple-ζ or higher basis sets, are very computationally expensive. In TUNA, two calculations can be run — for example with double- and triple-ζ basis sets — and the results from these two energy evaluations extrapolated to give an approximation to the complete basis set limit.

Because in TUNA derivatives of the energy are numerical, a complete basis set extrapolated energy can be carried forwards to give a complete basis set extrapolated geometry, molecular dynamics simulation, electric response property or vibrational frequency. Extrapolation can be done starting with double-, triple-, quadruple-, and quintuple-ζ basis sets.

Requesting a lower-ζ basis set and using the `EXTRAPOLATE` keyword tells TUNA to run two calculations — first with the lower-ζ basis, then the higher-ζ calculation begins:

```
TUNA SPE : H H 1.0 : CISD cc-pVQZ : EXTRAPOLATE
```

The convergence of the Hartree–Fock energy to the basis set limit [99] is expected to be

$$E_X^{\mathrm{SCF}} = E_\infty^{\mathrm{SCF}} + A\exp(-\alpha\sqrt{X})\,, \tag{5.3}$$

where A and α are constants that depend on the basis set. This can be solved, with X being the lower cardinal number and Y the higher, for an equation for the complete basis set limit,

$$E_\infty^{\mathrm{SCF}} = E_X^{\mathrm{SCF}} + \frac{E_Y^{\mathrm{SCF}} - E_X^{\mathrm{SCF}}}{1-\exp\left[\alpha(\sqrt{X}-\sqrt{Y})\right]}\,. \tag{5.4}$$

The correlation energy is supposed to converge as

$$E_\infty^{\mathrm{corr}} = \frac{X^\beta E_X^{\mathrm{corr}} - Y^\beta E_Y^{\mathrm{corr}}}{X^\beta - Y^\beta}\,. \tag{5.5}$$

The values of α and β are dependent on the basis set. For double-ζ basis sets, β is always taken as 2.4 in TUNA, and for triple-ζ and higher bases β is 3.0. The values of α are listed in [99] for double- and triple-ζ basis sets. We have internally optimised the α values for quadruple-ζ and quintuple-ζ sets to reproduce the complete basis Hartree–Fock limit energy of the hydrogen atom.

An example of a single point energy calculation on a sodium atom using extrapolation with the cc-pVDZ and cc-pVTZ basis sets can be requested by:

```
TUNA SPE : Na : MP2 cc-pVDZ : EXTRAPOLATE
```

An extrapolated optimisation and frequency run using def2-TZVPPD and def2-QZVPPD is:

```
TUNA OPTFREQ : H H 0.74 : CCSD def2-TZVPPD : EXTRAPOLATE
```

Finally, the CISD energy of the helium atom with quadruple-to-quintuple-ζ basis set extrapolation:

```
TUNA SPE : He : CISD aug-cc-pVQZ : EXTRAPOLATE
```

5.2 Custom Basis Sets

If you want a basis set which is not implemented natively in TUNA, a custom basis set can be read in. This is done by setting the basis type to `CUSTOM` and using the `BASIS [filepath.tuna]` keyword, where `filepath.tuna` is the path to the custom basis file.

For example, reading in basis data from "basis-data.tuna" can be done by:

```
TUNA SPE : H H 1.0 : HF CUSTOM : BASIS BASIS-DATA.TUNA
```

Note that any file path typed in the terminal will be converted to *lower* case, so you need to have a *fully lower case* basis data file name.

The `.tuna` file type stores basis set data, and has essentially the same form of the `.orca` file type that can be downloaded from the Basis Set Exchange [100] — so `.orca` basis data files can be used directly. In this file, lines to skip begin with "`$`", then elements are spelled out with their full names. Underneath this is the type of subshell (`S`, `P`, `D`, etc.) with the number of primitive Gaussians for that subshell. Next is a list of primitive Gaussians in three columns: the primitive number (1, 2, 3, etc.), the exponent and the coefficient. This process repeats for all subshells.

An example of this file type is shown, for the cc-pVDZ basis set for hydrogen and oxygen:

```
$DATA

HYDROGEN
S   4
1      1.301000E+01     1.968500E-02
2      1.962000E+00     1.379770E-01
3      4.446000E-01     4.781480E-01
4      1.220000E-01     5.012400E-01
S   1
1      1.220000E-01     1.000000E+00
P   1
1      7.270000E-01     1.000000E+00


OXYGEN
S   9
1      1.172000E+04     7.100000E-04
2      1.759000E+03     5.470000E-03
3      4.008000E+02     2.783700E-02
4      1.137000E+02     1.048000E-01
5      3.703000E+01     2.830620E-01
6      1.327000E+01     4.487190E-01
7      5.025000E+00     2.709520E-01
8      1.013000E+00     1.545800E-02
9      3.023000E-01    -2.585000E-03
```

```
S   9
1      1.172000E+04    -1.600000E-04
2      1.759000E+03    -1.263000E-03
3      4.008000E+02    -6.267000E-03
4      1.137000E+02    -2.571600E-02
5      3.703000E+01    -7.092400E-02
6      1.327000E+01    -1.654110E-01
7      5.025000E+00    -1.169550E-01
8      1.013000E+00     5.573680E-01
9      3.023000E-01     5.727590E-01
S   1
1      3.023000E-01     1.000000E+00
P   4
1      1.770000E+01     4.301800E-02
2      3.854000E+00     2.289130E-01
3      1.046000E+00     5.087280E-01
4      2.753000E-01     4.605310E-01
P   1
1      2.753000E-01     1.000000E+00
D   1
1      1.185000E+00     1.000000E+00


$END
```

5.3 List of Basis Sets

Native basis options in TUNA are listed in Table 5.1, and are taken from the Basis Set Exchange [100]. There are a large variety, including sets with diffuse functions to describe the tails of electron density, core functions to describe near-nuclear electrons and polarisation functions to flexibly describe the molecular correlation energy which results from spontaneous multipole formation.

Table 5.1 Basis sets implemented in TUNA

Basis Set	Citation	Description
STO-2G	[101]	Minimal
STO-3G	[101]	Minimal
STO-4G	[101]	Minimal
STO-5G	[101]	Minimal

Basis Set	Citation	Description
STO-6G	[101]	Minimal
3-21G	[102]	Double-ζ
4-31G	[103]	Double-ζ
6-31G	[103]	Double-ζ
6-31+G	[103]	Double-ζ, diffuse functions on heavy atoms
6-31++G	[104]	Double-ζ, diffuse functions
6-31G*	[105]	Double-ζ, polarisation functions on heavy atoms
6-31G**	[106]	Double-ζ, polarisation functions
6-31+G*	[107, 108]	Double-ζ, diffuse and polarisation functions on heavy atoms
6-31++G*	[107, 108]	Double-ζ, diffuse functions, polarisation functions on heavy atoms
6-31+G**	[107, 108]	Double-ζ, diffuse functions on heavy atoms, polarisation functions
6-31++G**	[107, 108]	Double-ζ, diffuse and polarisation functions
6-31G[2df,p]	[108–110]	Double-ζ, many polarisation functions
6-31G[3df,3pd]	[108, 109, 111]	Double-ζ, many polarisation functions
6-311G	[109]	Triple-ζ basis set
6-311+G	[104]	Triple-ζ basis set with diffuse functions on heavy atoms
6-311++G	[104]	Triple-ζ basis set with diffuse functions on all atoms
6-311G*	[109]	Triple-ζ basis set with polarisation functions on heavy atoms
6-311G**	[109]	Triple-ζ basis set with polarisation functions on all atoms
6-311+G*	[104, 109]	Triple-ζ basis set with polarisation and diffuse functions
6-311++G*	[104, 109]	Triple-ζ basis set with polarisation and diffuse functions
6-311+G**	[104, 109]	Triple-ζ basis set with polarisation and diffuse functions

Basis Set	Citation	Description
6-311++G**	[104, 109]	Triple-ζ basis set with polarisation and diffuse functions
6-311G[2df,2pd]	[109, 110]	Triple-ζ basis set with polarisation and diffuse functions
6-311+G[2d,p]	[104, 109]	Triple-ζ basis set with polarisation and diffuse functions
6-311++G[2d,2p]	[104, 109]	Triple-ζ basis set with polarisation and diffuse functions
6-311++G[3df,3pd]	[104, 109, 110]	Triple-ζ basis set with polarisation and diffuse functions
cc-pVDZ	[112, 113]	Polarised double-ζ
cc-pVTZ	[112, 113]	Polarised triple-ζ
cc-pVQZ	[112, 113]	Polarised quadruple-ζ
cc-pV5Z	[112, 113]	Polarised quintuple-ζ
cc-pV6Z	[114]	Polarised sextuple-ζ
aug-cc-pVDZ	[115]	Polarised double-ζ, diffuse functions
aug-cc-pVTZ	[115]	Polarised triple-ζ, diffuse functions
aug-cc-pVQZ	[115]	Polarised quadruple-ζ, diffuse functions
aug-cc-pV5Z	[115]	Polarised quintuple-ζ, diffuse functions
aug-cc-pV6Z	[115]	Polarised sextuple-ζ, diffuse functions
d-aug-cc-pVDZ	[115]	Polarised double-ζ, lots of diffuse functions
d-aug-cc-pVTZ	[115]	Polarised triple-ζ, lots of diffuse functions
d-aug-cc-pVQZ	[115]	Polarised quadruple-ζ, lots of diffuse functions
d-aug-cc-pV5Z	[115]	Polarised quintuple-ζ, lots of diffuse functions
d-aug-cc-pV6Z	[115]	Polarised sextuple-ζ, lots of diffuse functions
t-aug-cc-pVDZ	[115]	Polarised double-ζ, tons of diffuse functions
t-aug-cc-pVTZ	[115]	Polarised triple-ζ, tons of diffuse functions
t-aug-cc-pVQZ	[115]	Polarised quadruple-ζ, tons of diffuse functions
t-aug-cc-pV5Z	[115]	Polarised quintuple-ζ, tons of diffuse functions
t-aug-cc-pV6Z	[115]	Polarised sextuple-ζ, tons of diffuse functions

Basis Set	Citation	Description
cc-pCVDZ	[116]	Polarised double-ζ, core functions
cc-pCVTZ	[116]	Polarised triple-ζ, core functions
cc-pCVQZ	[116]	Polarised quadruple-ζ, core functions
cc-pCV5Z	[116]	Polarised quintuple-ζ, core functions
aug-cc-pCVDZ	[116]	Polarised double-ζ, core and diffuse functions
aug-cc-pCVTZ	[116]	Polarised triple-ζ, core and diffuse functions
aug-cc-pCVQZ	[116]	Polarised quadruple-ζ, core and diffuse functions
aug-cc-pCV5Z	[116]	Polarised quintuple-ζ, core and diffuse functions
cc-pwCVDZ	[117]	Polarised double-ζ, core–valence functions
cc-pwCVTZ	[117]	Polarised triple-ζ, core–valence functions
cc-pwCVQZ	[117]	Polarised quadruple-ζ, core–valence functions
cc-pwCV5Z	[117]	Polarised quintuple-ζ, core–valence functions
aug-cc-pwCVDZ	[117]	Polarised double-ζ, core–valence and diffuse functions
aug-cc-pwCVTZ	[117]	Polarised triple-ζ, core–valence and diffuse functions
aug-cc-pwCVQZ	[117]	Polarised quadruple-ζ, core–valence and diffuse functions
aug-cc-pwCV5Z	[117]	Polarised quintuple-ζ, core–valence and diffuse functions
ano-pVDZ	[99]	Polarised double-ζ, atomic natural orbitals
ano-pVTZ	[99]	Polarised triple-ζ, atomic natural orbitals
ano-pVQZ	[99]	Polarised quadruple-ζ, atomic natural orbitals
ano-pV5Z	[99]	Polarised quintuple-ζ, atomic natural orbitals
aug-ano-pVDZ	[99]	Polarised double-ζ, atomic natural orbitals, diffuse functions
aug-ano-pVTZ	[99]	Polarised triple-ζ, atomic natural orbitals, diffuse functions
aug-ano-pVQZ	[99]	Polarised quadruple-ζ, atomic natural orbitals, diffuse functions
aug-ano-pV5Z	[99]	Polarised quintuple-ζ, atomic natural orbitals, diffuse functions

Basis Set	Citation	Description
pc-0	[118, 119]	Double-ζ, polarisation-consistent
pc-1	[118, 119]	Polarised double-ζ, polarisation-consistent
pc-2	[118, 119]	Polarised triple-ζ, polarisation-consistent
pc-3	[118, 119]	Polarised quadruple-ζ, polarisation-consistent
pc-4	[118, 119]	Polarised quintuple-ζ, polarisation-consistent
aug-pc-0	[118–120]	Double-ζ, polarisation-consistent, diffuse functions
aug-pc-1	[118–120]	Polarised double-ζ, polarisation-consistent, diffuse functions
aug-pc-2	[118–120]	Polarised triple-ζ, polarisation-consistent, diffuse functions
aug-pc-3	[118–120]	Polarised quadruple-ζ, polarisation-consistent, diffuse functions
aug-pc-4	[118–120]	Polarised quintuple-ζ, polarisation-consistent, diffuse functions
pcseg-0	[121]	Double-ζ, polarisation-consistent
pcseg-1	[121]	Polarised double-ζ, polarisation-consistent
pcseg-2	[121]	Polarised triple-ζ, polarisation-consistent
pcseg-3	[121]	Polarised quadruple-ζ, polarisation-consistent
pcseg-4	[121]	Polarised quintuple-ζ, polarisation-consistent
aug-pcseg-0	[121]	Double-ζ, polarisation-consistent, diffuse functions
aug-pcseg-1	[121]	Polarised double-ζ, polarisation-consistent, diffuse functions
aug-pcseg-2	[121]	Polarised triple-ζ, polarisation-consistent, diffuse functions
aug-pcseg-3	[121]	Polarised quadruple-ζ, polarisation-consistent, diffuse functions
aug-pcseg-4	[121]	Polarised quintuple-ζ, polarisation-consistent, diffuse functions
def2-SVP	[122]	Polarised double-ζ
def2-SVPD	[122, 123]	Polarised double-ζ, diffuse functions
def2-TZVP	[122]	Polarised triple-ζ

Basis Set	Citation	Description
def2-TZVPD	[122, 123]	Polarised triple-ζ, diffuse functions
def2-TZVPP	[122, 123]	Very polarised triple-ζ
def2-TZVPPD	[122, 123]	Very polarised triple-ζ, diffuse functions
def2-QZVP	[122]	Polarised quadruple-ζ
def2-QZVPD	[122, 123]	Polarised quadruple-ζ, diffuse functions
def2-QZVPP	[122, 123]	Very polarised quadruple-ζ
def2-QZVPPD	[122, 123]	Very polarised quadruple-ζ, diffuse functions

6 Molecular Integrals

One of the first steps in all TUNA calculations is the calculation of one- and two-electron integrals over contracted Gaussian basis functions, which are necessary to compute expectation values such as the molecular energy. These are based on Cartesian primitive Gaussians of the form

$$g(x, y, z) = Nx^l y^m z^n e^{-\gamma(x^2+y^2+z^2)} \tag{6.1}$$

where l, m, and n are the angular momenta, N is a normalisation constant and γ is the primitive Gaussian exponent, defining the "steepness" of the function.

Molecular integrals in TUNA are computed with the McMurchie–Davidson scheme [124]. In this widely used method recurrence relations between Hermite Gaussians simplify the calculation [125].

The molecular integral code is stored in the tuna_integral module, which is written in Cython [10]. Cython is a compiled version of Python which can easily interface with the rest of the TUNA program. This dramatically increases the speed of the integrals, at the cost of having to be compiled. Diatomic parity is leveraged in the calculation — for a molecule aligned to the z axis, several of the two-electron integrals are necessarily zero, so their calculation can be skipped.

The integrals module demands the molecule be aligned along the z-axis, or located at the origin for an atom. This allows all the symmetry of diatomics to be taken into account, leading to an approximately tenfold increase in speed from prior versions. A diatomic molecule can always be rotated onto the z-axis, which is the approach taken in the molecular dynamics module to allow three-dimensional rotation along the trajectory but energy evaluation in one dimension.

The two-electron integral evaluation is often the slowest step of a TUNA calculation. However, each integral is completely independent of the others, so their calculation can be trivially made parallel using OpenMP over the loops [126]. This is done by default with four OpenMP threads, and the number of threads can easily be changed with the `THREADS` keyword by:

```
TUNA SPE : H H 1.0 : HF cc-pV5Z : THREADS 8
```

Note that this only affects the calculation of the molecular integrals, not the rest of the code.

6.1 Types of Integrals

Several matrices of integrals over basis functions are calculated by TUNA, used in the Hartree–Fock self-consistent field method and in Kohn–Sham density functional theory.

The simplest are overlap integrals,

$$\mathbf{S} = S_{\mu\nu} = \int \varphi_\mu(\mathbf{r})\varphi_\nu(\mathbf{r})\,\mathrm{d}\mathbf{r}\ . \tag{6.2}$$

The kinetic matrix is

$$\mathbf{T} = T_{\mu\nu} = -\frac{1}{2}\int \varphi_\mu(\mathbf{r})\nabla^2\varphi_\nu(\mathbf{r})\,\mathrm{d}\mathbf{r}\ . \tag{6.3}$$

The nuclear–electron interaction matrix is

$$\mathbf{V} = V_{\mu\nu} = -\sum_{\mathsf{A}}^{\text{nuclei}} \int \varphi_\mu(\mathbf{r})\frac{Z_\mathsf{A}}{|\mathbf{r}-\mathbf{R}_\mathsf{A}|}\varphi_\nu(\mathbf{r})\,\mathrm{d}\mathbf{r}\ . \tag{6.4}$$

A Cartesian component of the dipole matrix is

$$\mathbf{D}^\alpha = D^\alpha_{\mu\nu} = \int \varphi_\mu(\mathbf{r})(\mathbf{r}_\alpha - \mathbf{R}_0)\varphi_\nu(\mathbf{r})\,\mathrm{d}\mathbf{r}\ , \tag{6.5}$$

where $\mathbf{R}_0$ is the dipole origin. A similar component of the quadrupole matrix is

$$\mathbf{Q}^\alpha = Q^\alpha_{\mu\nu} = \int \varphi_\mu(\mathbf{r})(\mathbf{r}_\alpha - \mathbf{R}_0)^2\varphi_\nu(\mathbf{r})\,\mathrm{d}\mathbf{r}\ . \tag{6.6}$$

Finally, and with most difficulty, the electron–electron repulsion integrals are calculated as

$$g_{\mu\kappa\nu\lambda} = (\mu\kappa|\nu\lambda) = \iint \varphi_\mu(\mathbf{r})\varphi_\kappa(\mathbf{r})\frac{1}{|\mathbf{r}-\mathbf{r}'|}\varphi_\nu(\mathbf{r}')\varphi_\lambda(\mathbf{r}')\,\mathrm{d}\mathbf{r}\,\mathrm{d}\mathbf{r}'\ . \tag{6.7}$$

6.2 Spherical and Cartesian Harmonics

After the two-electron integrals are calculated by TUNA, they are transformed from the redundant Cartesian harmonics to spherical harmonics by

$$(\mu\kappa|\nu\lambda) = \sum_{\bar\mu\bar\kappa\bar\nu\bar\lambda} U_{\mu\bar\mu}U_{\kappa\bar\kappa}U_{\nu\bar\nu}U_{\lambda\bar\lambda}(\bar\mu\bar\kappa|\bar\nu\bar\lambda)_{\mathsf{cart}}\ . \tag{6.8}$$

This is a formally $\mathcal{O}(N^5)$ scaling process, but is made dramatically faster in TUNA by the use of sparse matrix multiplication, since the transformation matrices U are mostly zeroes.

This transformation reduces the size of the basis set, and makes basis functions "proper" atomic orbitals — there are now 5 d orbitals rather than 6. Although it requires additional transformations, this approach reduces the basis set size for future parts of the calculation and reduces the linear dependence of the basis set. Regardless, it can be disabled with the CARTHARM keyword:

```
TUNA SPE : H H 1.0 : HF cc-pVTZ : CARTHARM
```

7 Performance of TUNA

7.1 Speed and Efficiency

Python is extremely slow. However, TUNA escapes this fate by having very few loops in pure Python — the computationally intensive elements of calculations are array manipulations, handled efficiently with NumPy's [6] vectorised operations. This makes TUNA surprisingly quick. An optimisation and harmonic frequency calculation with B3LYP/cc-pVTZ on a dihydrogen molecule takes about 13 seconds in TUNA and 3 minutes in ORCA, on a single core of the same device.

However, moving to larger basis sets, the story changes — ORCA becomes much *faster* than TUNA, like in real life. Table 7.1 shows the relative speeds of ORCA and TUNA on a CCSD optimisation. For small basis sets, TUNA is the winner by far, but once over a hundred basis functions ORCA dominates. Improvements in version 0.11.0 have made a significant difference to the speed. This is largely down to more efficient molecular integral evaluation and performing calculations in spherical harmonics.

To overcome their slowness, the molecular integrals are written in Cython [10] (a compiled version of Python) which alleviates the problem for low angular momentum basis functions. However integrals over d, f and higher functions, which use recursion in the McMurchie–Davidson scheme, are still extremely slow. At the moment, TUNA is therefore very well suited to performing correlated calculations with small-to-medium sized basis sets. The routine use of larger basis sets will require a substantially rewritten molecular integrals module, which I do not want to do soon. It is possible to vectorise the integrals, to some extent, as has been done in ORCA's SHARK package [125].

Table 7.1 Calculation times for a full CI optimisation and frequency calculation on dihydrogen

Basis Set	ORCA 6.1	TUNA 0.10.0	TUNA 0.11.0
cc-pVDZ	97.2 s	0.4 s	0.2 s
cc-pVTZ	108.1 s	1.6 s	0.6 s
cc-pVQZ	129.3 s	30.0 s	6.3 s
cc-pV5Z	~6 min	> 1 hr	~3 min

The use of Numpy's "einsum" function is ubiquitous throughout the TUNA code. This makes the implementation of tensorial equations with lots of indices very straightforward — increasing readability of the code, and dramatically increasing speed and reducing memory use in some cases.

For example, the transformation of the two-electron integrals into the molecular orbital basis is

$$(pr|qs) = \sum_{\mu\kappa\nu\lambda} M_{\mu}^{p} M_{\kappa}^{r} (\mu\kappa|\nu\lambda) M_{\nu}^{q} M_{\lambda}^{s} \, . \tag{7.1}$$

The implementation of this in pure Python would require at least four loops — one for each summation index. However, with `einsum` the code practically writes itself:

```
ERI_MO = np.einsum("mp,kr,mknl,nq,ls->prqs", M, M, ERI_AO, M, M, optimize = True)
```

This reduces the number of lines of code to one, but *increases* code readability!

The "`optimize = True`" statement tells Numpy to find the most efficient "path" to perform this contraction — it turns out that this is four $\mathcal{O}(N^5)$ steps, contracting one molecular orbital with the atomic orbital basis two-electron integrals at a time, and saving the resulting temporary array. Complicated tensorial equations — as are found within coupled cluster theory — can therefore be efficiently and easily be implemented with `einsum`.

7.2 Memory Requirements

Calculations in TUNA are fast due to the program avoiding writing anything to disk, and storing everything in memory. Since there are a maximum of two atoms, this is normally completely fine.

However, for extremely large basis sets memory will become a problem. The most memory-intensive part of the calculation is the storage of the four-centre two-electron integrals at the start of a calculation. If TUNA predicts that these can not be stored in memory, the calculation aborts.

Using the additional print keyword `P` will show you the memory footprint of these integrals. On dihydrogen, storing $(\mu\kappa|\nu\lambda)$ requires around 3 GB of RAM for cc-pV5Z — which is fine — and around 32 GB for cc-pV6Z — which is not fine! For these extreme calculations, a laptop will not suffice. However, they will do very well on a high memory node of a high-performance computing cluster. The highest memory possible calculation uses triply-augmented cc-pV6Z, and will require around a terabyte of RAM. These memory problems, as well as the slowness of huge basis set calculations, can be avoided by using basis set extrapolation with the `EXTRAPOLATE` keyword.

Table 7.2 Numerical derivative step length, h, for different energy perturbations

Perturbation	Property	Value of h
$\partial E/\partial R$	Geometry optimisation	0.00005
$\partial^2 E/\partial R^2$	Harmonic frequency	0.01
$\partial^3 E/\partial R^3$	Vibrational perturbation theory	0.025
$\partial^4 E/\partial R^4$	Vibrational perturbation theory	0.025
$\partial E/\partial F$	Dipole moment	0.00001
$\partial^2 E/\partial F^2$	Polarisability	0.001
$\partial^3 E/\partial F^3$	Hyperpolarisability	0.0015

7.3 Accuracy and Precision

All derivatives of the energy in TUNA are calculated numerically, allowing total transferability of electronic structure methods between calculation types and properties. This approach means that once a method is implemented it automatically becomes available throughout TUNA for any property or calculation type. The numerical derivative equations have been carefully optimised to maximise their precision, and the step lengths used are listed in Table 7.2 for geometric perturbations based on the bond length, R, and electric perturbations based on the electric field, F.

The energy convergence criteria have also been tested to make sure response properties are stable — conservative default choices have been made here, including `TIGHT` convergence for first derivative quantities, and `EXTREME` convergence for higher derivatives. This may sometimes cause a problem (particularly for CCSDT and CCSDTQ which are more difficult to converge) and if it does, the convergence criteria can be safely reduced to `TIGHT` for second derivative properties — third and fourth derivatives might be riskier but still will most likely be okay.

In terms of accuracy, new features implemented in TUNA are always — assuming they exist there — benchmarked against ORCA [127]. Otherwise features will be benchmarked against PySCF [3] and unit tests will be written. Naturally, there will be some bugs which we don't know about yet, but a wide range of systems is used to check that newly implemented features behave.

8 Acknowledgements

There are a variety of codes and tutorials that have served as valuable inspiration for TUNA over the years. First and foremost is the ORCA quantum chemistry package, which puts clear emphasis on the user experience with the "simple input line" approach [4, 5].

The Crawford group programming projects for computational chemistry which contain examples for wavefunction methods such as coupled cluster theory and MP2 were very useful for getting started [128]. The "Psi4Numpy" collection of codes and tutorials has a diverse group of methods which were extremely useful [2].

The collection of Python codes of wavefunction methods based on Hartree–Fock theory, HarPy [129], was particularly useful for checking the CCSD and CCSDT code in TUNA. Finally, PyDFT, a simple educational DFT code that shows how basic exchange–correlation functionals can be implemented, was an extremely useful reference, as was its accompanying documentation [130].

I am very grateful to Hannah Whittome for, among other things, her brutal criticism of an early version of the TUNA logo — this led to the fish being put on an extreme weight loss regime. Thank you also to Chris Field for suggesting features such as custom scan plot colours and Will Armstrong for improving and simplifying the molecular dynamics code.

9 List of Keywords

Table 9.1 lists all of the mandatory and optional keywords in TUNA, grouped by their functions.

Table 9.1 Parameter keywords in TUNA

Parameter	Description
`P`	Print more information
`T`	Print less information
`DEBUG`	Print extreme information
`PRINTLEVEL` [int]	Print levels of information
`CH` [int] or `CHARGE` [int]	Molecular charge
`ML` [int] or `MULTIPLICITY` [int]	Molecular multiplicity
`BASIS` [string]	Custom basis filepath
`EXTRAPOLATE`	Extrapolate double-zeta basis
`DECONTRACT`	Decontract basis set
`CARTHARM`	Cartesian harmonics Gaussian basis
`ROTATE` [float]	Rotate orbitals for initial guess (by degrees)
`NOROTATE`	Don't rotate orbitals for initial guess
`STHRESH` [float]	Threshold for overlap matrix eigenvalues
`PRINTMOS` [int]	Number of molecular orbitals to print
`M1` [float]	Custom mass for first atom
`M2` [float]	Custom mass for second atom
`THREADS` [int]	Number of OpenMP threads for integrals
`COREGUESS`	Core Hamiltonian guess density
`SADGUESS`	Superposition of atomic densities guess density
`SCFGUESS`	Minimal self-consistent guess density

Parameter	Description
LOOSE or LOOSESCF	Loose SCF convergence
MEDIUM or MEDIUMSCF	Normal SCF convergence
TIGHT or TIGHTSCF	Tight SCF convergence
EXTREME or EXTREMESCF	Extreme SCF convergence
DAMP [float]	Static damping for SCF convergence
NODAMP	No damping for SCF convergence
MAXDAMP [float]	Maximum damping factor
DIIS [int]	Fock matrix extrapolation, and number of matrices
NODIIS	No Fock matrix extrapolation
SLOWCONV	High static damping (50%) for SCF convergence
VERYSLOWCONV	Extreme static damping (85%) for SCF convergence
MAXITER [int]	Maximum number of SCF iterations
ECONV [float]	Iterative energy convergence criteria
MAXDP [float]	Maximum change in density matrix
RMSDP [float]	Root-mean-square change in density matrix
DIISERR [float]	Root-mean-square commutator
STAB	Do SCF stability analysis
LOOSEOPT	Loose geometry convergence
MEDIUMOPT	Normal geometry convergence
TIGHTOPT	Tight geometry convergence
EXTREMEOPT	Extreme geometry convergence
CALCHESS	Calculate exact Hessian for optimisation
DEFAULTHESS [float]	Default Hessian for optimisation
D2	Semi-empirical dispersion correction with D2
MAXSTEP [float]	Maximum step for optimisation
MOREAD	Use density from previous step

Parameter	Description
NOMOREAD	Recalculate density from scratch each step
OPTMAX	Optimise to a local maximum, rather than a minimum
TRAJ [string]	Write trajectory
NOTRAJ	Do not write trajectory
MAXGEOMITER [int]	Maximum number of optimisation steps
ANHARMCONV [float]	Convergence criteria for anharmonic frequencies
NELEC [int]	Number of electrons for IP or EA calculation
VERTICAL	Vertical calculation for IP or EA
ZPE	Calculate zero-point energy for BDE
NOCP	Do not use counterpoise correction in BDE
VPT1	First-order vibrational perturbation theory
VPT2	Second-order vibrational perturbation theory
STEP [float]	Distance increment for coordinate scan
NUM [int]	Number of distance increments for coordinate scan
SCANPLOT	Plot potential energy surface from coordinate scan
SAVEPLOT [string]	Save plot from coordinate scan
DELPLOT	Delete temporary plot
ADDPLOT	Add another plot to the axes
DOT	Use dotted lines on the coordinate scan plot
DASH	Use dashed lines on the coordinate scan plot
BLACK	Use black lines on the coordinate scan plot
BLUE	Use blue lines on the coordinate scan plot
GREEN	Use green lines on the coordinate scan plot
RED	Use red lines on the coordinate scan plot
YELLOW	Use yellow lines on the coordinate scan plot
MAGENTA	Use magenta lines on the coordinate scan plot
CYAN	Use cyan lines on the coordinate scan plot
WHITE	Use white lines on the coordinate scan plot
COLOUR [str]	Change colour of coordinate scan plot

Parameter	Description
STEP	Timestep for MD simulation
NUM	Number of timesteps for MD simulation
TEMP [float] or TEMPERATURE [float]	Sets temperature for thermochemistry and MD
PRES [float] or PRESSURE [float]	Sets pressure for thermochemistry calculations
FREEZECORE [int]	Freeze core orbitals
SSS [float]	Same-spin scaling for SCS-MP2
OSS [float]	Opposite-spin scaling for SCS-MP2
MP3S [float]	Scaling for SCS-MP3
MPGRID [int]	Grid size for Laplace transform MP2
AMPCONV [float]	Convergence criteria for coupled cluster amplitudes
CORRMAXITER [int]	Maximum iterations for correlation
CORRDAMP [float]	Use damping in correlation, parameter
PRINTAMPS [int]	Print this number of largest t-amplitudes
NATORBS	Calculate natural orbitals
RELAXED	Calculate relaxed density
NEL [int]	Number of electrons for CAS methods
NORB [int]	Number of orbitals for CAS methods
FORCEMETHOD	Turn off automatic full CI
TD	Do a time-dependent calculation
TDA	Use the Tamm–Dancoff approximation
EXTHRESH [float]	Threshold for printing excited state contributions
ROOT [int]	Choice of excited state
NSTATES [int]	Number of states to print
[D]	Calculate perturbative doubles
LOOSEGRID	Loose DFT integration grid
MEDIUMGRID	Normal DFT integration grid

Parameter	Description
TIGHTGRID	Tight DFT integration grid
EXTREMEGRID	Extreme DFT integration grid
INTACC [float]	Integral accuracy for DFT
DFX [float]	Density functional theory exchange percentage
HFX [float]	Hartree–Fock theory exchange percentage
DFC [float]	Density functional theory correlation percentage
MPC [float]	Perturbation theory correlation percentage
NOX	Turn off DFT exchange
NOC	Turn off DFT correlation
XA [float]	Value of α in Xα LDA exchange
NL	Calculate non-local VV10 dispersion
DENSPLOT	Plot the electron density
SPINDENSPLOT	Plot the spin density
DIFFDENSPLOT	Plot the difference electron density
DIFFSPINDENSPLOT	Plot the difference spin density
PLOTMO [int]	Plot a molecular orbital
PLOTNO [int]	Plot a natural orbital
PLOTHOMO	Plot the highest occupied molecular orbital
PLOTLUMO	Plot the lowest unoccupied molecular orbital
VIBPLOT	Plot nuclear wavefunctions
ABSPLOT	Plot absorbance spectrum
PEAKWIDTH [float]	Gaussian peak width
EX [float]	Electric field in x direction
EY [float]	Electric field in y direction
EZ [float]	Electric field in z direction
EGX [float]	Electric field gradient in x direction
EGY [float]	Electric field gradient in y direction

Parameter	Description
EGZ [float]	Electric field gradient in z direction
DIPOLE	Calculate numerical dipole moment
QUADRUPOLE	Calculate numerical quadrupole moment
POLAR	Calculate numerical polarisability
HYPER	Calculate numerical hyperpolarisability

Bibliography

(1) 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. P. Pritchard, B. R. Brooks, I. Schaefer, Henry F., A. Y. Sokolov, K. Patkowski, I. DePrince, A. Eugene, U. Bozkaya, R. A. King, F. A. Evangelista, J. M. Turney, T. D. Crawford and C. D. Sherrill, *J. Chem. Phys.*, 2020, **152**, 184108, https://doi.org/10.1063/5.0006002.

(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. I. DePrince, T. D. Crawford, K. Patkowski and C. D. Sherrill, *J. Chem. Theory Comput.*, 2018, **14**, 3504–3511, https://doi.org/10.1021/acs.jctc.8b00286.

(3) Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters and G. K.-L. Chan, *WIREs Comput. Mol. Sci*, 2018, **8**, e1340, https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1340.

(4) F. Neese, *WIREs Comput. Mol. Sci*, 2012, **2**, 73–78, https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.81.

(5) F. Neese, *WIREs Comput. Mol. Sci.*, 2022, **12**, e1606, https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1606.

(6) 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 and T. E. Oliphant, *Nature*, 2020, **585**, 357–362, https://doi.org/10.1038/s41586-020-2649-2.

(7) P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, *Nat. Methods*, 2020, **17**, 261–272.

(8) J. D. Hunter, *Comput. Sci. Eng.*, 2007, **9**, 90–95.
(9) P. J. Mohr, E. Tiesinga, D. B. Newell and B. N. Taylor, *Codata Internationally Recommended 2022 Values of the Fundamental Physical Constants*, en, 2024.
(10) S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn and K. Smith, *Comput. Sci. Eng.*, 2011, **13**, 31–39.
(11) S. Boys and F. Bernardi, *Mol. Phys.*, 1970, **19**, 553–566, https://doi.org/10.1080/00268977000101561.
(12) T. Koopmans, *Physica*, 1934, **1**, 104–113, https://www.sciencedirect.com/science/article/pii/S0031891434900112.
(13) J. Schmidt, M. Moldover and E. May, 2009, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=901652.
(14) R. S. Mulliken, *J. Chem. Phys.*, 1955, **23**, 1833–1840, https://doi.org/10.1063/1.1740588.
(15) P.-O. Löwdin, *J. Chem. Phys.*, 1950, **18**, 365–375.
(16) I. Mayer, *Chem. Phys. Lett.*, 1983, **97**, 270–274, https://www.sciencedirect.com/science/article/pii/0009261483800050.
(17) E. R. Davidson and A. E. Clark, *Int. J. Quantum Chem.*, 2022, **122**, e26860, https://onlinelibrary.wiley.com/doi/abs/10.1002/qua.26860.
(18) E. R. Davidson, *Adv. Quantum Chem.*, 1972, **6**, ed. P.-O. Löwdin, 235–266, https://www.sciencedirect.com/science/article/pii/S006532760860547X.
(19) S. Grimme, J. Antony, S. Ehrlich and H. Krieg, *J. Chem. Phys.*, 2010, **132**, DOI: 10.1063/1.3382344, http://dx.doi.org/10.1063/1.3382344.
(20) O. A. Vydrov and T. Van Voorhis, *J. Chem. Phys.*, 2010, **133**, 244103, http://dx.doi.org/10.1063/1.3521275.
(21) G. Audi and A. Wapstra, *Nucl. Phys. A*, 1993, **565**, 1–65, https://www.sciencedirect.com/science/article/pii/037594749390024R.
(22) G. Audi and A. Wapstra, *Nucl. Phys. A*, 1995, **595**, 409–480, https://www.sciencedirect.com/science/article/pii/0375947495004459.
(23) I. Komasa, K. Pachucki, M. Przybytek and B. Jeziorski, *J. Chem. Theory Comput.*, 2011, **7**, 3105–3115.
(24) J. Neugebauer, M. Reiher, C. Kind and B. A. Hess, *J. Comput. Chem.*, 2002, **23**, 895–910, https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.10089.
(25) M. Piccardo, J. Bloino and V. Barone, *Int. J. Quantum Chem.*, 2015, **115**, 948–982.
(26) G. D. Dickenson, M. L. Niu, E. J. Salumbides, J. Komasa, K. S. E. Eikema, K. Pachucki and W. Ubachs, *Phys. Rev. Lett.*, 2013, **110**, 193601.
(27) J. Liu, E. J. Salumbides, U. Hollenstein, J. C. J. Koelemeij, K. S. E. Eikema, W. Ubachs and F. Merkt, *J. Chem. Phys.*, 2009, **130**, 174306, https://doi.org/10.1063/1.3120443.

(28) R. Iftimie, P. Minary and M. E. Tuckerman, *Proc. Natl Acad. Sci.*, 2005, **102**, 6654–6659, https://www.pnas.org/doi/abs/10.1073/pnas.0500193102.

(29) R. Car and M. Parrinello, *Phys. Rev. Lett.*, 1985, **55**, 2471–2474, https://link.aps.org/doi/10.1103/PhysRevLett.55.2471.

(30) W. C. Swope, H. C. Andersen, P. H. Berens and K. R. Wilson, *J. Chem. Phys.*, 1982, **76**, 637–649.

(31) D. R. Hartree and W. Hartree, *Proc. Roy. Soc. London A*, 1935, **150**, 9–33, https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1935.0085.

(32) C. C. J. Roothaan, *Rev. Mod. Phys.*, 1951, **23**, 69–89, https://link.aps.org/doi/10.1103/RevModPhys.23.69.

(33) G. G. Hall, *Proc. Roy. Soc. London A*, 1951, **205**, 541–552, http://dx.doi.org/10.1098/rspa.1951.0048.

(34) P. Pulay, *Chem. Phys. Lett.*, 1980, **73**, 393–398.

(35) M. C. Zerner and M. Hehenberger, *Chem. Phys. Lett.*, 1979, **62**, 550–554, https://www.sciencedirect.com/science/article/pii/0009261479807617.

(36) E. Schrödinger, *Annal. Phys.*, 1926, **385**, 437–490, http://dx.doi.org/10.1002/andp.19263851302.

(37) C. Møller and M. S. Plesset, *Phys. Rev.*, 1934, **46**, 618–622, https://link.aps.org/doi/10.1103/PhysRev.46.618.

(38) S. Grimme, *J. Chem. Phys.*, 2003, **118**, 9095–9102, https://doi.org/10.1063/1.1569242.

(39) S. Grimme, *J. Comput. Chem.*, 2003, **24**, 1529–1537, https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.10320.

(40) R. C. Lochan and M. Head-Gordon, *J. Chem. Phys.*, 2007, **126**, 164101, https://doi.org/10.1063/1.2718952.

(41) F. Neese, T. Schwabe, S. Kossmann, B. Schirmer and S. Grimme, *J. Chem. Theory Comput.*, 2009, **5**, 3060–3073, https://doi.org/10.1021/ct9003299.

(42) P. Pulay and S. Saeb , *Theor. Chim. Acta*, 1986, **69**, 357–368, http://dx.doi.org/10.1007/BF00526697.

(43) P. R. Surján, *Chem. Phys. Lett*, 2005, **406**, 318–320, https://www.sciencedirect.com/science/article/pii/S000926140500357X.

(44) M. Kobayashi and H. Nakai, *Chem. Phys. Lett.*, 2006, **420**, 250–255, https://www.sciencedirect.com/science/article/pii/S000926140600025X.

(45) T. J. Lee, A. P. Rendell and P. R. Taylor, *J. Phys. Chem.*, 1990, **94**, 5463–5468, https://doi.org/10.1021/j100377a008.

(46) S. Hirata, R. Podeszwa, M. Tobita and R. J. Bartlett, *J. Chem. Phys.*, 2004, **120**, 2581–2592, https://doi.org/10.1063/1.1637577.

(47) J. Noga and R. J. Bartlett, *J. Chem. Phys.*, 1987, **86**, 7041–7050, https://doi.org/10.1063/1.452353.

(48) S. A. Kucharski and R. J. Bartlett, *J. Chem. Phys.*, 1992, **97**, 4282–4288, https://doi.org/10.1063/1.463930.

(49) A. Jiang, D. Matthews, D. Poole, C. Briggs, J. Turney, C. D. Sherrill and H. Schaefer, *ChemRxiv*, 2025, **2025**, 144102, https://chemrxiv.org/doi/abs/10.26434/chemrxiv-2025-qgc1q.

(50) C. Riplinger, B. Sandhoefer, A. Hansen and F. Neese, *J. Chem. Phys.*, 2013, **139**, 134101, https://doi.org/10.1063/1.4821834.

(51) A. P. Rendell, T. J. Lee and A. Komornicki, *Chem. Phys. Lett.*, 1991, **178**, 462–470, https://www.sciencedirect.com/science/article/pii/000926149187003T.

(52) M. Lesiuk, *J. Chem. Theory Comput.*, 2022, **18**, 6537–6556, https://doi.org/10.1021/acs.jctc.2c00460.

(53) A. Jiang, H. S. III and J. Turney, *ChemRxiv*, 2025, **2025**, 2386–2401, https://chemrxiv.org/doi/abs/10.26434/chemrxiv-2025-77dmp-v2.

(54) C. Hampel, K. A. Peterson and H.-J. Werner, *Chem. Phys. Lett.*, 1992, **190**, 1–12, https://www.sciencedirect.com/science/article/pii/000926149286093W.

(55) T. Helgaker, P. Jorgensen and J. Olsen, in *Molecular Electronic-Structure Theory*, John Wiley and Sons, Ltd, Chichester, 2000, ch. 13, pp. 648–723.

(56) P. Hohenberg and W. Kohn, *Phys. Rev.*, 1964, **136**, B864–B871, https://link.aps.org/doi/10.1103/PhysRev.136.B864.

(57) W. Kohn and L. J. Sham, *Phys. Rev.*, 1965, **140**, A1133–A1138, https://link.aps.org/doi/10.1103/PhysRev.140.A1133.

(58) J. C. Slater, *Phys. Rev.*, 1951, **81**, 385–390, https://link.aps.org/doi/10.1103/PhysRev.81.385.

(59) A. D. Becke, *Phys. Rev. A*, 1988, **38**, 3098–3100, https://link.aps.org/doi/10.1103/PhysRevA.38.3098.

(60) J. P. Perdew, K. Burke and M. Ernzerhof, *Phys. Rev. Lett.*, 1996, **77**, 3865–3868, https://link.aps.org/doi/10.1103/PhysRevLett.77.3865.

(61) Y. Zhang and W. Yang, *Phys. Rev. Lett.*, 1998, **80**, 890–890, http://dx.doi.org/10.1103/PhysRevLett.80.890.

(62) B. Hammer, L. B. Hansen and J. K. Nørskov, *Phys. Rev. B*, 1999, **59**, 7413–7421, http://dx.doi.org/10.1103/PhysRevB.59.7413.

(63) K. Burke, J. P. Perdew and Y. Wang, in *Electronic Density Functional Theory: Recent Progress and New Directions*, ed. J. F. Dobson, G. Vignale and M. P. Das, Springer US, Boston, MA, 1998, pp. 81–111, https://doi.org/10.1007/978-1-4899-0316-7_7.

(64) C. Adamo and V. Barone, *J. Chem. Phys,* 1998, **108**, 664–675, https://doi.org/10.1063/1.475428.

(65) A. D. Becke, *J. Chem. Phys.*, 1997, **107**, 8554–8560, http://dx.doi.org/10.1063/1.475007.

(66) S. Grimme, *J. Comput. Chem.*, 2006, **27**, 1787–1799, http://dx.doi.org/10.1002/jcc.20495.

(67) J. Tao, J. P. Perdew, V. N. Staroverov and G. E. Scuseria, *Phys. Rev. Lett.*, 2003, **91**, 146401, https://link.aps.org/doi/10.1103/PhysRevLett.91.146401.

(68) J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin and J. Sun, *Phys. Rev. Lett.*, 2009, **103**, DOI: 10.1103/physrevlett.103.026403, http://dx.doi.org/10.1103/PhysRevLett.103.026403.

(69) J. Sun, A. Ruzsinszky and J. P. Perdew, *Phys. Rev. Lett.*, 2015, **115**, DOI: 10.1103/physrevlett.115.036402, http://dx.doi.org/10.1103/PhysRevLett.115.036402.

(70) A. P. Bartók and J. R. Yates, *J. Chem. Phys.*, 2019, **150**, DOI: 10.1063/1.5094646, http://dx.doi.org/10.1063/1.5094646.

(71) J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew and J. Sun, *J. Phys. Chem. Lett.*, 2020, **11**, 8208–8215, http://dx.doi.org/10.1021/acs.jpclett.0c02405.

(72) N. Mardirossian and M. Head-Gordon, *J. Chem. Phys.*, 2015, **142**, DOI: 10.1063/1.4907719, http://dx.doi.org/10.1063/1.4907719.

(73) S. H. Vosko, L. Wilk and M. Nusair, *Can. J. Phys.*, 1980, **58**, 1200–1211, https://doi.org/10.1139/p80-159.

(74) J. P. Perdew and Y. Wang, *Phys. Rev. B*, 1992, **45**, 13244–13249, https://link.aps.org/doi/10.1103/PhysRevB.45.13244.

(75) J. P. Perdew, *Phys. Rev. B*, 1986, **33**, 8822–8824, https://link.aps.org/doi/10.1103/PhysRevB.33.8822.

(76) C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B*, 1988, **37**, 785–789, https://link.aps.org/doi/10.1103/PhysRevB.37.785.

(77) J. P. Perdew and K. Schmidt, *AIP Conf. Proc.*, 2001, **577**, 1–20, https://doi.org/10.1063/1.1390175.

(78) L. Goerigk and S. Grimme, *Phys. Chem. Chem. Phys.*, 2011, **13**, 6670, http://dx.doi.org/10.1039/C0CP02984J.

(79) C. Adamo and V. Barone, *J. Chem. Phys.*, 1999, **110**, 6158–6170, https://doi.org/10.1063/1.478522.

(80) C. Adamo and V. Barone, *Chem. Phys. Lett.*, 1997, **274**, 242–250, https://www.sciencedirect.com/science/article/pii/S0009261497006519.

(81) A. D. Becke, *J. Chem. Phys.*, 1993, **98**, 5648–5652, https://doi.org/10.1063/1.464913.

(82) P. J. Stephens, F. J. Devlin, C. F. Chabalowski and M. J. Frisch, *J. Phys. Chem.*, 1994, **98**, 11623–11627, https://doi.org/10.1021/j100096a001.

(83) J. C. Sancho-García, *J. Chem. Phys.*, 2006, **124**, 124112, https://doi.org/10.1063/1.2180774.
(84) K. Hui and J.-D. Chai, *J. Chem. Phys.*, 2016, **144**, 044114, http://dx.doi.org/10.1063/1.4940734.
(85) M. Bursch, H. Neugebauer, S. Ehlert and S. Grimme, *J. Chem. Phys.*, 2022, **156**, 134105, http://dx.doi.org/10.1063/5.0086040.
(86) S. Grimme, *J. Chem. Phys.*, 2006, **124**, 034108, https://doi.org/10.1063/1.2148954.
(87) E. Brémond and C. Adamo, *J. Chem. Phys.*, 2011, **135**, 024106, https://doi.org/10.1063/1.3604569.
(88) É. Brémond, J. C. Sancho-García, Á. J. Pérez-Jiménez and C. Adamo, *J. Chem. Phys.*, 2014, **141**, 031101, https://doi.org/10.1063/1.4890314.
(89) J.-D. Chai and S.-P. Mao, *Chem. Phys. Lett.*, 2012, **538**, 121–125, https://www.sciencedirect.com/science/article/pii/S000926141200526X.
(90) A. Karton, A. Tarnopolsky, J.-F. Lamère, G. C. Schatz and J. M. L. Martin, *J. Phys. Chem. A*, 2008, **112**, 12868–12886, https://doi.org/10.1021/jp801805p.
(91) F. Yu, *J. Phys. Chem. A*, 2014, **118**, 3175–3182, https://doi.org/10.1021/jp5005506.
(92) T. Schwabe and S. Grimme, *Phys. Chem. Chem. Phys.*, 2007, **9**, 3397–3406, http://dx.doi.org/10.1039/B704725H.
(93) S. Kozuch, D. Gruzman and J. M. L. Martin, *J. Phys. Chem. C*, 2010, **114**, 20801–20808, https://doi.org/10.1021/jp1070852.
(94) L. Wittmann, H. Neugebauer, S. Grimme and M. Bursch, *J. Chem. Phys.*, 2023, **159**, 224103, https://doi.org/10.1063/5.0174988.
(95) A. D. Becke, *J. Chem. Phys,* 1988, **88**, 2547–2553, https://doi.org/10.1063/1.454033.
(96) J. B. Foresman, M. Head-Gordon, J. A. Pople and M. J. Frisch, *J. Phys. Chem.*, 1992, **96**, 135–149, https://doi.org/10.1021/j100180a030.
(97) M. Head-Gordon, R. J. Rico, M. Oumi and T. J. Lee, *Chem. Phys. Lett.*, 1994, **219**, 21–29, https://www.sciencedirect.com/science/article/pii/0009261494000700.
(98) J. F. Stanton and R. J. Bartlett, *The Journal of Chemical Physics*, 1993, **98**, 7029–7039.
(99) F. Neese and E. F. Valeev, *J. Chem. Theory Comput.*, 2011, **7**, 33–43, https://doi.org/10.1021/ct100396y.
(100) B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson and T. L. Windus, *J. Chem. Inf. Model.*, 2019, **59**, 4814–4820.
(101) W. J. Hehre, R. F. Stewart and J. A. Pople, *J. Chem. Phys.*, 1969, **51**, 2657–2664.
(102) J. S. Binkley, J. A. Pople and W. J. Hehre, *J. Am. Chem. Soc.*, 1980, **102**, 939–947.
(103) R. Ditchfield, W. J. Hehre and J. A. Pople, *J. Chem. Phys.*, 1971, **54**, 724–728.

(104) T. Clark, J. Chandrasekhar, G. W. Spitznagel and P. V. R. Schleyer, *J. Comput. Chem.*, 1983, **4**, 294–301.
(105) V. A. Rassolov, M. A. Ratner, J. A. Pople, P. C. Redfern and L. A. Curtiss, *J. Comput. Chem.*, 2001, **22**, 976–984, https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.1058.
(106) M. M. Francl, W. J. Pietro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFrees and J. A. Pople, *J. Chem. Phys.*, 1982, **77**, 3654–3665.
(107) P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta*, 1973, **28**, 213–222.
(108) W. J. Hehre, R. Ditchfield and J. A. Pople, *J. Chem. Phys.*, 1972, **56**, 2257–2261.
(109) R. Krishnan, J. S. Binkley, R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1980, **72**, 650–654.
(110) M. J. Frisch, J. A. Pople and J. S. Binkley, *J. Chem. Phys.*, 1984, **80**, 3265–3269.
(111) L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov and J. A. Pople, *J. Chem. Phys.*, 1998, **109**, 7764–7776.
(112) J. Dunning, Thom H., *J. Chem. Phys.*, 1989, **90**, 1007–1023.
(113) D. E. Woon and J. Dunning, Thom H., *J. Chem. Phys.*, 1993, **98**, 1358–1371.
(114) A. K. Wilson, T. van Mourik and T. H. Dunning, *J. Mol. Struct.*, 1996, **388**, 339–349, https://www.sciencedirect.com/science/article/pii/S0166128096800480.
(115) T. van Mourik, A. K. Wilson and T. H. Dunning, Jr, *Mol. Phys.*, 1999, **96**, 529–547.
(116) D. E. Woon and T. H. Dunning, *J. Chem. Phys.*, 1995, **103**, 4572–4585.
(117) K. A. Peterson and T. H. Dunning, *J. Chem. Phys.*, 2002, **117**, 10548–10560.
(118) F. Jensen, *J. Chem. Phys.*, 2001, **115**, 9113–9125.
(119) F. Jensen, *J. Chem. Phys.*, 2002, **116**, 7372–7379.
(120) F. Jensen, *J. Chem. Phys.*, 2002, **117**, 9234–9240.
(121) F. Jensen, *J. Chem. Theory Comput.*, 2014, **10**, 1074–1085.
(122) F. Weigend and R. Ahlrichs, *Phys. Chem. Chem. Phys.*, 2005, **7**, 3297, http://dx.doi.org/10.1039/B508541A.
(123) D. Rappoport and F. Furche, *J. Chem. Phys.*, 2010, **133**, 134105.
(124) L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.*, 1978, **26**, 218–231, https://www.sciencedirect.com/science/article/pii/002199917890092X.
(125) F. Neese, *J. Comput. Chem.*, 2023, **44**, 381–396, https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.26942.
(126) L. Dagum and R. Menon, *IEEE Comput. Sci. Eng.*, 1998, **5**, 46–55, http://dx.doi.org/10.1109/99.660313.
(127) F. Neese, F. Wennmohs, U. Becker and C. Riplinger, *J. Chem. Phys.*, 2020, **152**, 224108, /aip/jcp/article/152/22/224108/1061982/The-ORCA-quantum-chemistry-program-package.
(128) T. D. Crawford, *C++ Programming Tutorial in Chemistry*, https://github.com/CrawfordGroup/ProgrammingProjects, 2020.

(129) P. W. Borthwick, *HarPy*, https://github.com/pwborthwick/harpy, 2022.

(130) I. A. W. Filot, *Elements of Electronic Structure Theory*, Eindhoven University of Technology, 2025.