

TUNA

Theoretical Unification of Nuclear Arrangements

Version 0.7.0

August 2025

Harry Brough

Foreword

Welcome to TUNA!

This is a personal project to deepen my understanding of the theory and implementation of quantum chemistry. A simple to use terminal-based Python program, TUNA can calculate energies and properties of diatomic molecules with several quantum chemical methods.

Unlike other Python programs for quantum chemistry such as Psi4 [1] and PySCF [2], TUNA is not primarily a collection of modules to incorporate into a workflow, but is intended to be a complete program to be run directly from the command line. The program is inspired by the ORCA quantum chemistry package [3, 4] (although the tuna is clearly the more majestic animal), but no input files need to be set up as the geometry of a diatomic molecule is easily specified in a single input line to the terminal.

The key strength of TUNA is its user-friendly interface, hopefully offering an intuitive experience that could potentially serve as a valuable teaching aid in the future. On the technical side, TUNA utilises numerical energy derivatives, so any electronic structure method can be used to calculate geometries, frequencies 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 [5], TUNA is surprisingly fast for a Python program and significantly outpaces ORCA for calculations with small basis sets, although larger basis sets are agonisingly slow — the molecular integrals are mostly to blame.

While TUNA is currently 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.

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

Harry Brough

Contents

1	Getting Started	4
1.1	Installation	4
1.2	Simple Input Line	4
1.3	Units and Constants	5
1.4	Program Components	6
2	Summary of Features in TUNA	7
2.1	Calculation Types	7
2.2	Electronic Structure Methods	7
2.3	Basis Sets	8
2.4	Properties	8
2.5	Miscellaneous	9
3	Detailed Documentation	10
3.1	Calculation Types	10
3.1.1	Single Point Energy	11
3.1.1.1	Ionisation Energy and Electron Affinity	13
3.1.1.2	Rotational Constant	13
3.1.1.3	Natural Orbitals	13
3.1.1.4	Multipole Moments	13
3.1.1.5	Population Analysis	14
3.1.2	Coordinate Scan	15
3.1.2.1	Plotting Coordinate Scans	16
3.1.3	Geometry Optimisation	18
3.1.3.1	Dispersion Correction	21
3.1.4	Harmonic Frequency	23
3.1.4.1	Transition Intensity	24
3.1.4.2	Thermochemistry	25
3.1.5	<i>Ab Initio</i> Molecular Dynamics	27
3.2	Electronic Structure Methods	28

3.2.1	Hartree–Fock Theory	30
3.2.1.1	Initial Guess	31
3.2.1.2	SCF Convergence	32
3.2.2	Møller–Plesset Perturbation Theory	34
3.2.2.1	Spin-component Scaling	36
3.2.2.2	Orbital-optimised MP2	36
3.2.3	Coupled Cluster Theory	37
3.2.3.1	Coupled Cluster Doubles	39
3.2.3.2	Coupled Cluster Singles and Doubles	39
3.2.3.3	Coupled Cluster Singles, Doubles and Triples	39
3.2.3.4	Coupled Cluster Singles, Doubles and Perturbative Triples	40
3.2.3.5	Linearised Coupled Cluster	40
3.2.3.6	Coupled Cluster Density Matrix	41
3.2.4	Excited States	41
3.2.4.1	Configuration Interaction Singles	41
3.2.4.2	Perturbative Doubles Correction to CIS	43
3.3	Basis Sets and Molecular Integrals	43
4	List of Keywords	45
	Bibliography	49

1 Getting Started

1.1 Installation

The easiest way to install TUNA is first ensuring a recent version of Python 3 is installed (ideally Python 3.12 onwards), then run the following command from a terminal:

```
pip install QuantumTUNA
```

If this proceeds according to plan, next locate the folder where TUNA was installed, and add it to the PATH environment variable. On Windows, where * is your user, this folder may be:

```
C:\Users\*\AppData\Local\Programs\Python\Python312\Lib\site-packages\TUNA
```

Once the folder containing the TUNA Python files is added to PATH, 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!

Installing TUNA has only been tested with Python 3.12 on Windows 11 — and not tested very well. In principle, it will work on up to date MacOS and Linux systems, as long as Python 3.12 is installed. Let me know if it doesn't!

1.2 Simple Input Line

All calculations in TUNA are requested via 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]
```

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 and basis sets are listed in section 3.1, and electronic structure methods are detailed in section 3.2.

For example, a single point energy calculation on an H₂ molecule with bond length of 1.0 Å, with 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, an MP2/6-31G 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 : MP2 6-31G : CHARGE +1 D2
```

As many keywords as you like can be written after the second exclamation mark. Some keywords such as CHARGE require a value (like +1 here) — just write this directly after the keyword, and then more keywords can be written afterwards.

1.3 Units and Constants

In the TUNA output, distances are printed in angstroms, times in femtoseconds and everything else in atomic units unless otherwise stated. Internally, units in TUNA are set by the CODATA 2022 recommendations [6], shown in Table 1.1. All other units and values are derived from these.

Table 1.1 Fundamental constants used in TUNA

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

1.4 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. In addition to these modules, some external modules

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

Module	Description
<code>tuna</code>	Main program, input parsing
<code>tuna_basis</code>	Storing and setting up basis sets
<code>tuna_cc</code>	Coupled cluster
<code>tuna_ci</code>	Configuration interaction
<code>tuna_energy</code>	Calculating energies, building molecules, coordinate scans
<code>tuna_integral</code>	Evaluating one- and two-electron integrals
<code>tuna_md</code>	<i>Ab initio</i> molecular dynamics
<code>tuna_molecule</code>	Storing and parsing molecular information
<code>tuna_mp</code>	Møller–Plesset perturbation theory
<code>tuna_optfreq</code>	Calculating gradients, optimisation algorithm, harmonic frequencies
<code>tuna_postscf</code>	Calculating and printing molecular properties
<code>tuna_scf</code>	Main self-consistent field loop, convergence acceleration
<code>tuna_thermo</code>	Thermochemistry after frequency calculation
<code>tuna_util</code>	General utility, units, frequently used functions

are also imported. All these should automatically be installed when `pip install QuantumTUNA` is run. The most important are NumPy [5], SciPy [7], which speed up the code considerably compared to pure Python. In addition, TermColor [8] makes the terminal output more vibrant, while Matplotlib [9] enables the creation of two-dimensional plots. The latter module is not strictly necessary to run TUNA calculations.

In the directory where TUNA was installed, the program can be run without the shell script by calling `tuna.py` directly. This would be useful if the shell scripts don't work, but Python does.

```
python tuna.py SPE : H H 1.0 : HF STO-3G
```

2 Summary of Features in TUNA

2.1 Calculation Types

- Single point energy, SPE
- Geometry optimisation, OPT
- Harmonic frequency, FREQ
- Optimisation and harmonic frequency, OPTFREQ
- Coordinate scan, SCAN
- *Ab initio* molecular dynamics, MD

2.2 Electronic Structure Methods

- Hartree–Fock, HF
- Unrestricted Hartree–Fock, UHF
- MP2, MP2
- Spin-component-scaled MP2, SCS-MP2
- MP3, MP3
- Spin-component-scaled MP3, SCS-MP3
- Orbital-optimised MP2, OMP2
- Configuration interaction singles, CIS
- Configuration interaction singles with perturbative doubles, CIS[D]
- Coupled electron pair approximation, CEPA
- Linearised coupled cluster doubles, LCCD
- Coupled cluster doubles, CCD
- Linearised coupled cluster singles and doubles, LCCSD
- Coupled cluster singles and doubles, CCSD
- Coupled cluster singles, doubles and perturbative triples, CCSD[T]
- Coupled cluster singles, doubles and triples, CCSDT

2.3 Basis Sets

- STO-3G
- STO-6G
- 3-21G
- 4-31G
- 6-31G
- 6-31+G
- 6-31++G
- 6-311G
- 6-311+G
- 6-311++G
- 6-31G*
- 6-31G**
- 6-311G*
- 6-311G**
- cc-pVDZ
- cc-pVTZ
- cc-pVQZ
- cc-pV5Z
- cc-pV6Z

2.4 Properties

- Nuclear and electronic dipole moment
- Mulliken and Löwdin charges and bond orders
- Mayer valences and bond order
- Spin contamination
- Koopmans' theorem electron affinity and ionisation energy, HOMO–LUMO gap
- Molecular orbitals and orbital energies
- Rotational constants
- Thermochemistry

- Harmonic fundamental transition intensities

2.5 Miscellaneous

- McMurchie-Davidson integrals of Gaussian basis functions
- D2 dispersion correction
- Interface to GUI for plots for coordinate scan
- Ghost atoms
- Optimise to a minimum or maximum
- Decontract basis sets
- Rotate initial guess orbitals
- Temperature and pressure keywords for thermochemistry
- Optimisation and SCF convergence keywords
- Identification of point group
- Approximate or exact Hessian for optimisations
- Maximum iterations keywords
- Fock matrix extrapolation (DIIS)
- Dynamic and static damping
- Slow convergence option
- Very slow convergence option
- Level shift

3 Detailed Documentation

3.1 Calculation Types

This section dives into the different calculation types available in TUNA, listed in Table 3.1 with their respective keywords.

Table 3.1 Calculation types available in TUNA

Keyword	Calculation type
SPE	Single point energy
OPT	Optimisation
FREQ	Harmonic frequency
OPTFREQ	Optimisation and subsequent harmonic frequency
SCAN	Coordinate scan
MD	<i>Ab initio</i> molecular dynamics

All calculation types run on an initial molecule, defined by its atoms, charge and multiplicity. Atom types from hydrogen to argon are implemented.

However, “ghost atoms” are also available, which act as additional basis functions for a single atom and can be used to counteract basis set superposition error. 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 argon atom’s basis functions, lowering the energy compared to the calculation on a lone atom:

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

Trying to run a calculation for a molecule (eg. OPT, FREQ, MD) a ghost atom 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 H₂ can be requested by:

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

Shorter parameters 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 begin by printing the TUNA logo to the terminal, importing libraries and printing the requested calculation type 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 electrons, number of occupied and virtual orbitals, point group and bond length are printed. The specified calculation then starts.

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.

3.1.1 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 in the 6-311G basis can be called 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: C₁fv
Bond length: 2.0000

~~~~~

The nuclear repulsion energy and optional dispersion energy (see section 3.1.3.1) are then calculated. One- and two-electron integrals are evaluated. The eigenvalues of the overlap matrix are checked to make sure they are not too small, to avoid linear dependence in the basis set. The default threshold is  $10^{-7}$  and can be adjusted with the `STHRESH` keyword. The Hartree–Fock self-consistent field cycle begins, discussed in section 3.2.1.2. When this is finished, the energy components will be printed as well as the Virial ratio. This is the ratio between the molecular kinetic and potential energy, and should approach 2.0 as the molecular structure approaches an equilibrium geometry or for an atom. If a correlated calculation (eg. MP2, CCSD), or an excited state calculation (eg. CIS), is requested, this then begins.

After all the energies are calculated, the density matrix is used in property calculations. Currently, the RHF, UHF, CIS, unrelaxed MP2 and linearised coupled cluster density matrices are available in TUNA. The MP2 density is implemented for either RHF or UHF references, with and without spin-component scaling, as well as for orbital-optimised MP2. The CIS and MP2 density matrices are unrelaxed (see section 3.2.2), while there is no difference between the unrelaxed and relaxed densities for OMP2. If an MP3 calculation is requested, properties will be calculated with the relevant unrelaxed MP2 density matrix. 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  $\alpha$  and  $\beta$  orbitals for spin-unrestricted calculations. If the reduced print keyword, `T`, is used, only the molecular information, SCF cycles and final energies are printed.

### 3.1.1.1 Ionisation Energy and Electron Affinity

The first properties printed are the Koopmans' theorem parameters [10]. These are the ionisation energy,  $-\varepsilon_{\text{HOMO}}$ , electron affinity,  $-\varepsilon_{\text{LUMO}}$  and HOMO–LUMO gap,  $\varepsilon_{\text{LUMO}} - \varepsilon_{\text{HOMO}}$ . These parameters are only determined in calculations with an RHF reference.

### 3.1.1.2 Rotational Constant

The rotational constant of the molecule is printed, calculated by

$$B = \frac{\hbar^2}{2\mu(\mathbf{R}_A - \mathbf{R}_B)^2}, \quad (3.1)$$

where  $\mu$  is the reduced mass of the molecule, using the masses in section 3.1.4. This is printed in units of GHz and  $\text{cm}^{-1}$ .

### 3.1.1.3 Natural Orbitals

For UHF, MP2 and coupled cluster methods, a natural orbital calculation can be requested via the NATORBS keyword. Here, the one-particle reduced density matrix,  $\mathbf{P}$  is diagonalised, to give a list of natural orbital occupancies between 0.0 and 2.0, which are printed. The calculation of natural orbitals is the default behaviour for MP2-based calculations, and can be disabled with the NONATORBS keyword.

### 3.1.1.4 Multipole Moments

The only multipole moment implemented in TUNA is the dipole moment. The origin for dipole moment calculations (which is printed) is the centre of mass,  $\mathbf{R}_0$ , calculated with the masses in section 3.1.4. The nuclear dipole moment,

$$\boldsymbol{\mu}_{\text{nuc}} = Z_A(\mathbf{R}_A - \mathbf{R}_0) + Z_B(\mathbf{R}_B - \mathbf{R}_0), \quad (3.2)$$

and the electronic dipole moment,

$$\boldsymbol{\mu}_{\text{ele}} = - \sum_{\mu\nu} P_{\mu\nu} D_{\mu\nu}, \quad (3.3)$$

are calculated, where  $D_{\mu\nu}$  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.

In addition to these values and the total dipole moment, 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<sup>+</sup> at 2 Å gives this output, meaning the hydrogen atom is more negatively charged than the helium atom:

```

~~~~~
 Dipole Moment
~~~~~
Nuclear: -1.498793      Electronic: -1.477457

Total: -2.976250      H ----- He  <-----+
~~~~~

```

### 3.1.1.5 Population Analysis

By default, TUNA prints Mulliken [11], Löwdin [12] and Mayer [13] population analysis after single point energy calculations and optimisations. Note that all the equations in this section are true for diatomics, not necessarily molecules in general. This can be disabled with the T keyword.

The Mulliken analysis [11] is very widely used — despite its considerable weaknesses [14] — 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

$$N = \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} . \quad (3.4)$$

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

$$N_A = \sum_{\mu \in A} \sum_{\nu \in A} P_{\mu\nu} S_{\mu\nu} + \frac{1}{2} B_{AB} , \quad (3.5)$$

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

$$B_{AB} = 2 \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu} S_{\mu\nu} . \quad (3.6)$$

The charge of an atom in a molecule is then given by  $Q_A = Z_A - N_A$ . At the bottom of the single point energy output, TUNA prints these charges and the Mulliken bond order, as well as the sum of the charges which should equal the requested molecular charge.

Because the molecular orbitals that generate a specific energy and density are not unique, popu-

lation analysis can be performed with non-localised orthogonal basis functions, in Löwdin analysis [12] where the density matrix is rotated by

$$\mathbf{P}^L = \mathbf{S}^{\frac{1}{2}} \mathbf{P} \mathbf{S}^{\frac{1}{2}} . \quad (3.7)$$

Using this density matrix, atomic populations are calculated by

$$N_A = \sum_{\mu \in A} P_{\mu\mu}^L , \quad (3.8)$$

and the bond order is calculated by

$$B_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu}^L P_{\nu\mu}^L . \quad (3.9)$$

Similarly to the Mulliken analysis, Löwdin charges and bond order are printed at the end of a calculation.

The results of a Mayer analysis [13] are also printed. The Mayer charges are the same as the Mulliken charges, but the bond order (and bonded valence) is given by

$$B_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} (\mathbf{P}\mathbf{S})_{\mu\nu} (\mathbf{P}\mathbf{S})_{\nu\mu} + (\mathbf{R}\mathbf{S})_{\mu\nu} (\mathbf{R}\mathbf{S})_{\nu\mu} , \quad (3.10)$$

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 lithium), is

$$V_A = 2N_A - \sum_{\mu \in A} \sum_{\nu \in B} (\mathbf{P}\mathbf{S})_{\mu\nu} (\mathbf{P}\mathbf{S})_{\nu\mu} \quad (3.11)$$

and the free valence, which is a measure of the ability to form further bonds, is calculated by  $F_A = V_A - B_{AB}$ . These valences are printed under the Mayer analysis heading at the end of a TUNA calculation.

### 3.1.2 Coordinate Scan

A coordinate scan can be requested in TUNA by the SCAN calculation type. This calculates the potential energy surface, increasing the interatomic distance at each step, using the chosen electronic structure method.

There are two mandatory parameters for a coordinate scan. The distance between steps must



be specified by the STEP keyword, and the total 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<sub>2</sub>, starting at a bond length of 0.3 Å 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 density matrix is read in at each scan step for RHF-based calculations and UHF-based calculations on non-singlet molecules; 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 section 3.2.1.1 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, ROTATE) can also be combined with the scan calculation. At the end of a coordinate scan calculation, a table of the bond lengths and final energies is printed, which can be easily copied and pasted into a spreadsheet.

### 3.1.2.1 Plotting Coordinate Scans

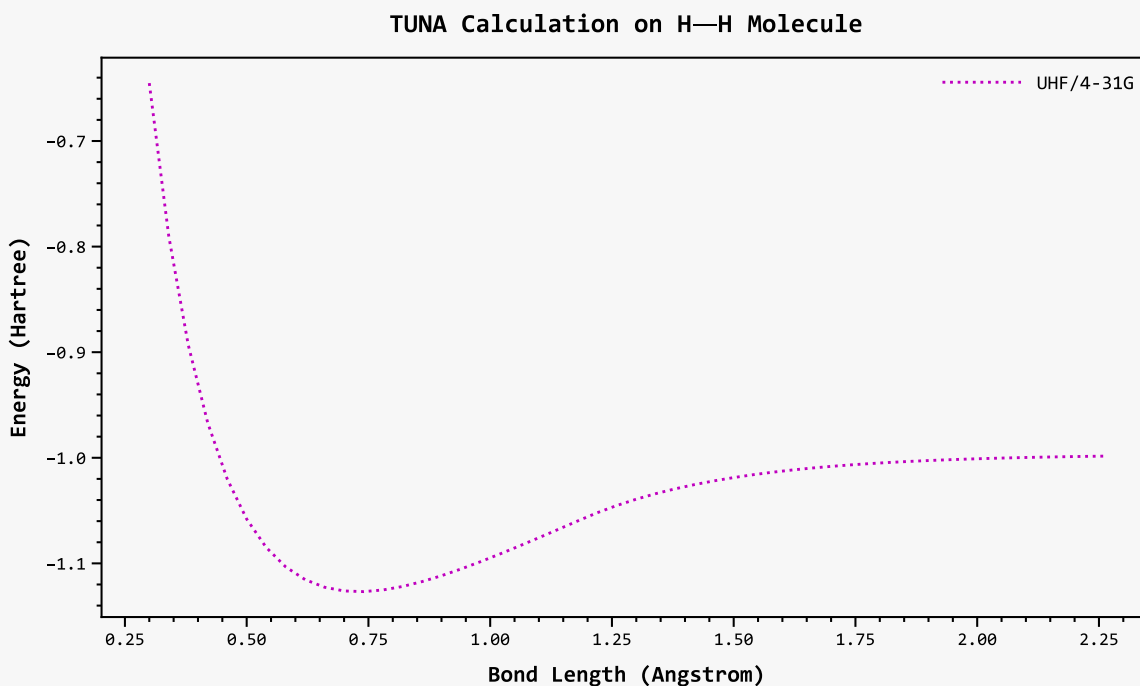
Due to the Matplotlib [9] integration in TUNA, two-dimensional plots can easily be generated from coordinate scan calculations with the SCANPLOT keyword. The line is straight and blue by default, but can be changed to dashed using the keyword DASH, dotted using DOT, and the colour can be changed to any of BLACK, WHITE, RED, BLUE, YELLOW, GREEN, CYAN, and MAGENTA.

For example, a plot with a magenta dotted line over the H<sub>2</sub> potential energy surface with UHF can be presented by:

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

This produces the plot in Figure 3.1, 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 rasterised and vector image extensions are supported.

Multiple calculations can be plotted on the same axes, for easy comparison. To do this, add the keyword ADDPLOT to every calculation you want on the same axes. Then just run calculations one



**Figure 3.1** Potential energy surface of H<sub>2</sub> calculated in TUNA with UHF/4-31G

after the other. To clear the axes, just 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 3.2.

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

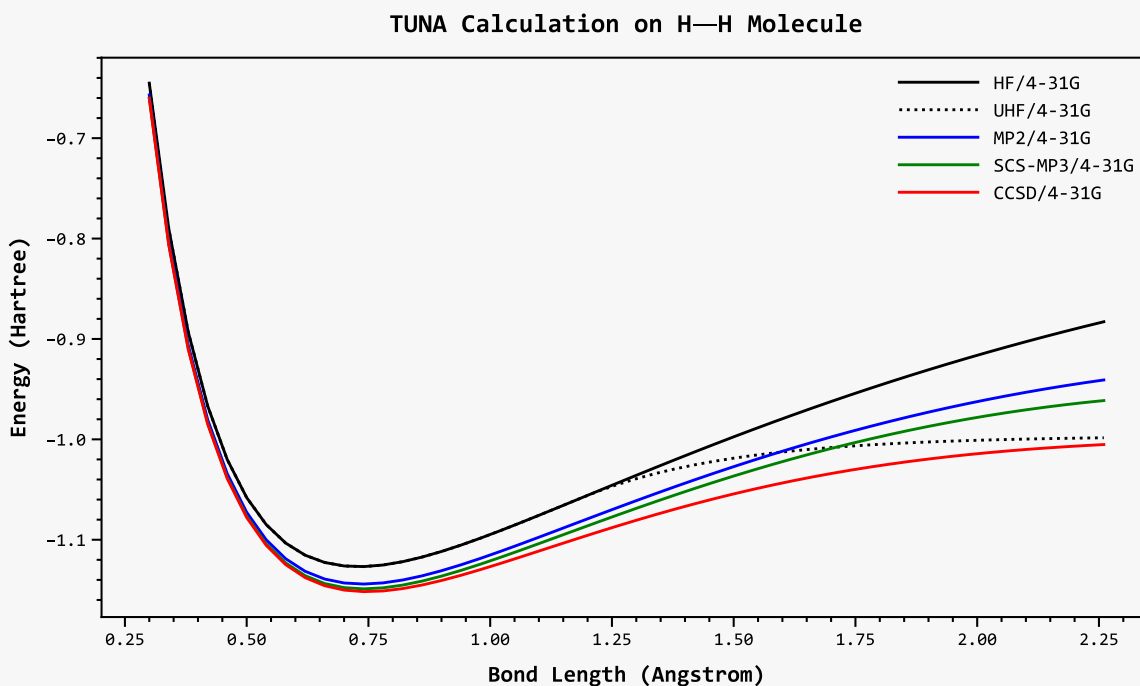
```
TUNA SCAN : H H 0.3 : UHF 4-31G : NUM 50 STEP 0.04 SCANPLOT BLACK DOT ADDPLOT
```

```
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 its easy for potential energy surfaces to get mixed up part way through a coordinate scan (or optimisation).



**Figure 3.2** Potential energy surfaces of  $H_2$  calculated in TUNA with various electronic structure methods in 4-31G basis set

The following two calculations on the H—F molecule produced Figure 3.3:

```
TUNA SCAN : H F 0.3 : HF 6-311G : NUM 50 STEP 0.04 SCANPLOT BLACK ADDPLOT
```

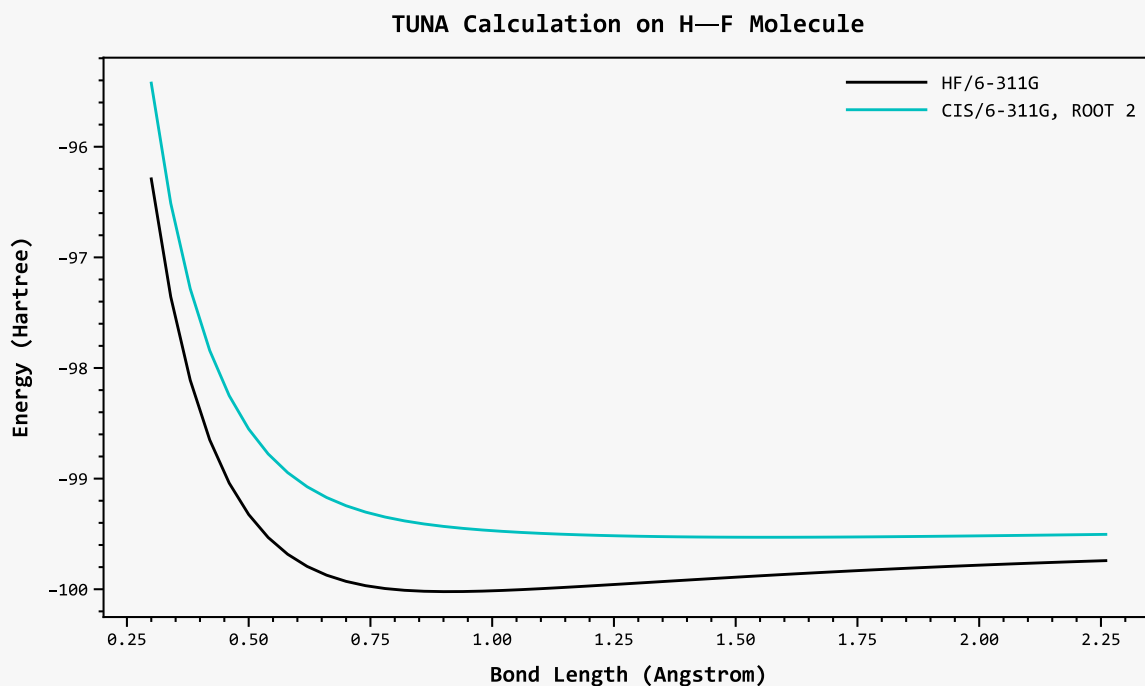
```
TUNA SCAN : H F 0.3 : CIS 6-311G : NUM 50 STEP 0.04 SCANPLOT CYAN ADDPLOT
ROOT 2
```

### 3.1.3 Geometry Optimisation

A geometry optimisation finds the location of a stationary point on the potential energy surface, 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 iteration, the molecule is set up and the energy is calculated. Tight SCF convergence criteria are used by default for optimisations and the density matrix is read in from the previous optimisation step — this guess



**Figure 3.3** Potential energy surfaces of hydrogen fluoride ground state and first excited singlet state (root 2) calculated in TUNA with CIS/6-311G

strategy can be disabled using the NOMOREAD keyword.

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

$$\nabla E_i = \frac{\partial E}{\partial r} = \frac{E(r + \delta r) - E(r - \delta r)}{2\delta r}, \quad (3.12)$$

where  $r$  is the interatomic distance and  $\delta r = 0.0001$ . Next, an approximation to the Hessian,  $H$ , is calculated by

$$H_i = \frac{\nabla E_i - \nabla E_{i-1}}{r_i - r_{i-1}}. \quad (3.13)$$

Instead of this approximate Hessian, 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 local minimum, either the exact or approximate Hessian is used in the convex region of the potential energy surface, whereas for optimisations to maxima, these Hessians are used in the concave region. Outside of these regions, a default Hessian with value 1/4 is used. This value can be changed using the

DEFAULTHESS keyword.

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

$$r_i = r_{i-1} + \frac{\nabla E_i}{H_i} . \quad (3.14)$$

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 angstroms by default, to prevent exploding optimisations from a bad starting guess geometry. This can be changed with the MAXSTEP keyword, giving the maximum 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.

Four optimisation convergence criteria are available: loose, medium, tight and extreme. Medium convergence is the default for standard optimisation calculations, while optimisation and frequency calculations use tight default criteria. The optimisation convergence criteria can be changed with the keywords LOOSEOPT, MEDIUMOPT, TIGHTOPT and EXTREMEOPT. 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, the maximum can be increased with the MAXITER keyword, which has a value of 30 by default.

**Table 3.2** Optimisation convergence criteria in TUNA

| Convergence | Gradient / hartree bohr <sup>-1</sup> | Step / bohr      |
|-------------|---------------------------------------|------------------|
| Loose       | 10 <sup>-3</sup>                      | 10 <sup>-2</sup> |
| Normal      | 10 <sup>-4</sup>                      | 10 <sup>-4</sup> |
| Tight       | 10 <sup>-6</sup>                      | 10 <sup>-5</sup> |
| Extreme     | 10 <sup>-8</sup>                      | 10 <sup>-7</sup> |

When the gradient and step fall below the convergence criteria, the optimisation will terminate, and the final density will be used to calculate properties. These properties can also be calculated at each step with the additional print keyword, P. Finally, TUNA prints the bond length of the converged geometry and the minimum energy. If a harmonic frequency calculation is desired immediately after the optimisation terminates (on the final geometry), the OPTFREQ calculation

type should be used instead of OPT.

Geometries can also be optimised to a local maximum, rather than a local minimum, using the OPTMAX keyword. This simply inverts the treatment of the Hessian and step update calculations, bringing the structure to a local maximum on the potential energy surface. This probably isn't very useful, but could find, for instance, the unphysical local maximum predicted by restricted MP2 on the potential energy surface of H<sub>2</sub>.

An example of a calculation to do this, using 30 maximum iterations, with extreme geometry convergence criteria is:

```
TUNA OPT : H H 2.0 : MP2 6-31++G : OPTMAX MAXITER 30 EXTREMEOPT
```

### 3.1.3.1 Dispersion Correction

Currently, the only available form of dispersion correction is the D2 semi-empirical scheme by Grimme [15]. Here, the energy due to dispersion is calculated by

$$E_{\text{disp}} = -\frac{s_6}{f_{\text{damp}}} \frac{C_6^{ij}}{R_{ij}^6}, \quad (3.15)$$

where the damping function is given by

$$f_{\text{damp}} = 1 + \exp \left[ -d \left( \frac{R_{ij}}{R_r} - 1 \right) \right]. \quad (3.16)$$

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 [3, 4]. The values of the Van der Waals radii,  $R_r$ , and  $C_6$  coefficients are element-specific (see Table 3.3). For correlated calculations, the Hartree–Fock parameterisation is also used, so D2 is not recommended in these cases (although it's not really any good for Hartree–Fock either).

**Table 3.3** Parameters for D2 semi-empirical dispersion correction in TUNA

| Atom       | $C_6$ / au | Van der Waals Radius / $a_0$ |
|------------|------------|------------------------------|
| Hydrogen   | 2.4283     | 1.8916                       |
| Helium     | 1.3876     | 1.9124                       |
| Lithium    | 27.92545   | 1.55902                      |
| Beryllium  | 27.92545   | 2.66073                      |
| Boron      | 54.28985   | 2.80624                      |
| Carbon     | 30.35375   | 2.74388                      |
| Nitrogen   | 21.33435   | 2.63995                      |
| Oxygen     | 12.1415    | 2.53601                      |
| Fluorine   | 13.00875   | 2.43208                      |
| Neon       | 10.92735   | 2.34893                      |
| Sodium     | 99.03995   | 2.16185                      |
| Magnesium  | 99.03995   | 2.57759                      |
| Aluminium  | 187.15255  | 3.09726                      |
| Silicon    | 160.09435  | 3.24277                      |
| Phosphorus | 135.9848   | 3.22198                      |
| Sulfur     | 96.61165   | 3.18041                      |
| Chlorine   | 87.93915   | 3.09726                      |
| Argon      | 79.96045   | 3.01411                      |

Dispersion correction is requested by the D2 keyword, for example:

```
TUNA OPT : H H 1.0 : HF 6-311++G : D2
```

### 3.1.4 Harmonic Frequency

A harmonic frequency calculation can be requested on a molecule at a set bond length with the FREQ keyword by:

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

These can 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 : SCS-MP2 6-311G** : CH -1
```

Harmonic frequency calculations in TUNA determine the numerical second derivative, which is the bond's force constant,  $k$ , by

$$k = \frac{\partial^2 E}{\partial r^2} = \frac{-E(r + 2\delta r) + 16E(r + \delta r) - 30E(r) + 16E(r - \delta r) - E(r - 2\delta r)}{12(\delta r)^2}. \quad (3.17)$$

The use of five single point calculations makes harmonic frequency calculations slow, but ensures the frequencies are numerically stable. For additional reliability, TIGHT SCF convergence criteria are used for frequency calculations by default, as are TIGHTOPT optimisation convergence criteria.

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

$$\omega = \sqrt{\frac{k}{\mu}}, \quad (3.18)$$

where  $\mu$  is the reduced mass, calculated using the atomic masses shown in Table 3.4. These are the masses of the most abundant isotope. Custom masses can be used via the M1 and M2 keywords, 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.

```
TUNA OPTFREQ : H Cl 0.650 : RHF 6-311++G : M2 36.0
```

This printed frequency has units of  $\text{cm}^{-1}$ . If  $k$  is negative (such as at a maximum on the potential energy surface),  $\omega$  will be imaginary and TUNA will output the frequency appended with an “i”.

The harmonic frequencies of  $\text{H}_2$  calculated from some different methods implemented in TUNA are shown in Table 3.5, where coupled cluster methods perform much better than perturbation theory. The following values were calculated with the parameters here, for instance for OMP2:



**Table 3.4** Default atomic masses in TUNA

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

TUNA OPTFREQ : H H 1.0: OMP2 cc-pVTZ : EXTREME EXTREMEOPT

#### 3.1.4.1 Transition Intensity

In addition to the frequency, the fundamental absorption intensity is calculated and printed to the console. These intensities are calculable for any density implemented in TUNA.

The intensity is proportional to the dipole moment derivative at the equilibrium geometry in

**Table 3.5** Harmonic frequencies of H<sub>2</sub> calculated in the cc-pVTZ basis set in TUNA

| Method  | Harmonic Frequency / cm <sup>-1</sup> |
|---------|---------------------------------------|
| HF      | 4587                                  |
| MP2     | 4524                                  |
| OMP2    | 4518                                  |
| SCS-MP2 | 4511                                  |
| MP3     | 4474                                  |
| SCS-MP3 | 4499                                  |
| CCD     | 4420                                  |
| CCSD    | 4407                                  |
| Exact   | 4401                                  |

mass-weighted normal coordinates,

$$\frac{\partial \mu}{\partial \mathbf{q}}, \quad (3.19)$$

which is calculated by the central differences method of numerical differentiation when the Hessian is determined.

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

$$\mathcal{A} = \frac{N_A}{12c^2\epsilon_0} \left( \frac{\partial \mu}{\partial \mathbf{q}} \right)^2, \quad (3.20)$$

where  $N_A$  is the Avogadro constant. This absorption coefficient is calculated as it is independent of the experimental setup — factors including the incident light intensity, molar concentration and length of sample cell [16].

### 3.1.4.2 Thermochemistry

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

$$U = E + E_{\text{zero-point}} + E_{\text{translational}} + E_{\text{vibrational}} + E_{\text{rotational}}, \quad (3.21)$$

and entropy,

$$S = S_{\text{electronic}} + S_{\text{translational}} + S_{\text{vibrational}} + S_{\text{rotational}} , \quad (3.22)$$

are first calculated, before the enthalpy,

$$H = U + k_{\text{B}}T , \quad (3.23)$$

and Gibbs free energy,

$$G = H - TS , \quad (3.24)$$

are 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

$$E_{\text{vibrational}} = \frac{\omega}{\exp(\omega/k_{\text{B}}T) - 1} , \quad (3.25)$$

assuming the molecule behaves as a harmonic oscillator, which is a bad approximation for diatomics, particularly at high temperatures.

The electronic contribution to entropy is assumed to be zero, but the translational, vibrational and rotational contributions are calculated by three fairly horrible formulas, where  $p$  is the pressure and  $B$  is the rotational constant:

$$S_{\text{translational}} = k_{\text{B}} \left[ \frac{5}{2} + \ln \left( \frac{M k_{\text{B}}T}{h} \right)^{\frac{3}{2}} + \ln \left( \frac{k_{\text{B}}T}{p} \right) \right] \quad (3.26)$$

$$S_{\text{vibrational}} = k_{\text{B}} \left[ \frac{\omega}{k_{\text{B}}T (\exp(\omega/k_{\text{B}}T) - 1)} - \ln \left( 1 - \exp \left( -\frac{\omega}{k_{\text{B}}T} \right) \right) \right] \quad (3.27)$$

$$S_{\text{rotational}} = k_{\text{B}} \left[ 1 + \ln \left( \frac{k_{\text{B}}T}{2\pi\sigma B} \right) \right] \quad (3.28)$$

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

The entropy, and therefore the Gibbs free energy, depends 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 called by:

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

### 3.1.5 *Ab Initio* Molecular Dynamics

The implementation in TUNA of *ab initio* molecular dynamics (AIMD) is Born–Oppenheimer molecular dynamics [17], 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 [18], 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 100 step AIMD calculation using MP2/6-31G with a timestep of 0.2 fs at an initial temperature of 350 K can be called by:

```
TUNA MD : H H 1.2 : MP2 6-31G : TIMESTEP 0.2 TEMP 350 MDNUMBER 100
```

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.

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 [19] yields velocities and positions at the same point in time and is numerically stable, it is implemented in TUNA. Positions are updated by

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2, \quad (3.29)$$

and velocities are updated by

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t + \Delta t)}{2} \Delta t . \quad (3.30)$$

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

$$K = 3k_{\text{B}}T , \quad (3.31)$$

where

$$K = \frac{1}{2}m_1\mathbf{v}_1^2 + \frac{1}{2}m_2\mathbf{v}_2^2 . \quad (3.32)$$

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 coordinates are rotated onto one dimension to calculate the energy and gradient before forces are back-transformed into three dimensions.

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.

## 3.2 Electronic Structure Methods

The electronic structure methods implemented in TUNA are shown in Table 3.6. All of these methods are available for restricted or unrestricted references, and an unrestricted reference can be requested for a singlet molecule by adding the letter "U" to the start of the method. For instance, a CCD calculation on a UHF reference becomes "UCCD".

**Table 3.6** Electronic structure methods implemented in TUNA

| Keyword | Electronic Structure Method                     |
|---------|-------------------------------------------------|
| HF      | Hartree–Fock theory                             |
| UHF     | Unrestricted Hartree–Fock theory                |
| MP2     | Second-order Møller–Plesset perturbation theory |

| Keyword  | Electronic Structure Method                                  |
|----------|--------------------------------------------------------------|
| UMP2     | Unrestricted second-order Møller–Plesset perturbation theory |
| SCS-MP2  | Spin-component-scaled MP2 perturbation theory                |
| USCS-MP2 | Unrestricted spin-component-scaled 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                |
| OMP2     | Orbital-optimised MP2 theory                                 |
| UOMP2    | Unrestricted orbital-optimised MP2 theory                    |
| CIS      | Configuration interaction singles                            |
| UCIS     | Unrestricted configuration interaction singles               |
| CIS[D]   | CIS with perturbative doubles                                |
| UCIS[D]  | Unrestricted CIS with perturbative doubles                   |
| CEPA     | Coupled electron pair approximation                          |
| UCEPA    | 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  |
| CCD      | Coupled cluster doubles                                      |
| UCCD     | Unrestricted coupled cluster doubles                         |
| CCSD     | Coupled cluster singles and doubles                          |
| UCCSD    | Unrestricted coupled cluster singles and doubles             |

| Keyword  | Electronic Structure Method                                            |
|----------|------------------------------------------------------------------------|
| 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              |

### 3.2.1 Hartree–Fock Theory

Both restricted and unrestricted Hartree–Fock calculations are available. Restricted open-shell Hartree–Fock is not yet implemented. 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 [20, 21] turn the variational differential restricted Hartree–Fock equations [22] into a simpler eigenvalue problem,

$$\mathbf{F}\mathbf{C} = \mathbf{S}\mathbf{C}\epsilon, \quad (3.33)$$

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

$$\mathbf{F} = \mathbf{T} + \mathbf{V}_{\text{ne}} + 2\mathbf{J} - \mathbf{K}. \quad (3.34)$$

These matrices are determined from the one- and two-electron integrals of basis functions, discussed in section 3.1. To solve the Roothaan–Hall equations, they need to be converted into a conventional eigenvalue problem — ie. 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

$$\mathbf{F}' = \mathbf{X}^\dagger \mathbf{F} \mathbf{X}, \quad (3.35)$$

the Roothaan–Hall equations become a conventional eigenvalue problem,

$$\mathbf{F}'\mathbf{C}' = \mathbf{C}'\epsilon . \quad (3.36)$$

Now the rotated Fock matrix,  $\mathbf{F}'$ , can be diagonalised for the eigenvectors,  $\mathbf{C}'$ , and eigenvalues,  $\epsilon$ . The eigenvectors are then rotated back to the original atomic orbital basis, where they represent molecular orbitals,

$$\mathbf{C} = \mathbf{X}\mathbf{C}' . \quad (3.37)$$

Now the molecular orbitals have been determined the restricted Hartree–Fock density matrix,  $\mathbf{P}$ , is constructed by summing over the doubly occupied orbitals,

$$P_{\mu\nu} = \gamma \sum_i^{\text{occ}} C_{\mu i} C_{\nu i}^* . \quad (3.38)$$

Here,  $\gamma$  is one for UHF and two for RHF. Finally, energy can be calculated by contracting this new density matrix with the Fock and core Hamiltonian matrices, and adding on the constant nuclear repulsion energy,

$$E = V_{\text{nn}} + \frac{1}{2} \sum_{\mu\nu} P_{\mu\nu} (F_{\mu\nu} + H_{\mu\nu}^{\text{core}}) . \quad (3.39)$$

The resulting molecular orbitals are then read in and contracted with the two-electron integrals to determine the new two-electron contribution to the Fock matrix,

$$2\mathbf{J} - \mathbf{K} = \sum_{\kappa\lambda} P_{\kappa\lambda} (\langle \mu\nu | \kappa\lambda \rangle - \frac{1}{2} \langle \mu\nu | \lambda\kappa \rangle) . \quad (3.40)$$

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.

### 3.2.1.1 Initial Guess

The 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 Fock matrix to diagonalise is

$$\mathbf{F} = \mathbf{T} + \mathbf{V}_{\text{ne}} . \quad (3.41)$$



This works well for first row diatomics, but goes horribly wrong for heavier elements. Keep this in mind considering comparison to other quantum chemistry programs, which use a better default guess.

If an unrestricted Hartree–Fock calculation is run on a singlet molecule with equal  $\alpha$  and  $\beta$  guess densities the SCF will likely get caught in a local — but not global — minimum on the surface of orbital rotations, the restricted Hartree–Fock solution. To get around this, TUNA rotates the HOMO with the LUMO after the initial guess density is made, by

$$\begin{bmatrix} \psi'_{\text{HOMO}} \\ \psi'_{\text{LUMO}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \psi_{\text{HOMO}} \\ \psi_{\text{LUMO}} \end{bmatrix} \quad (3.42)$$

By default,  $\theta = 45^\circ$ . 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. Similarly, orbital rotation can be activated with the ROTATE keyword. If the default value of  $\theta$  is not converging well, use the ROTATE keyword followed by the desired rotation angle in degrees to change it.

An example of forcing  $45^\circ$  rotation on a UHF calculation, corresponding to symmetric and anti-symmetric linear combinations of the HOMO and LUMO, is:

```
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.

### 3.2.1.2 SCF Convergence

A number of SCF convergence criteria are available in TUNA, shown in Table 3.7. These can be activated by the keywords LOOSE, MEDIUM (default for single point calculations), TIGHT (default for optimisations, frequencies and MD calculations) and EXTREME. The number of SCF iterations before TUNA gives up can be picked with the keyword MAXITER, which is 100 by default.

There are several methods available to accelerate SCF convergence. The most powerful is probably Fock matrix extrapolation, also known as direct inversion of the iterative subspace (DIIS) [23].

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

$$\mathbf{PF} - \mathbf{FP} = \mathbf{0} . \quad (3.43)$$

**Table 3.7** Self-consistent field convergence criteria in TUNA

| Convergence | $\Delta E$ | MAX( $\Delta \mathbf{P}$ ) | RMS( $\Delta \mathbf{P}$ ) | [ $\mathbf{F}$ , $\mathbf{PS}$ ] |
|-------------|------------|----------------------------|----------------------------|----------------------------------|
| 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     | $10^{-11}$ | $10^{-10}$                 | $10^{-11}$                 | $10^{-9}$                        |

An error matrix can therefore be defined at the  $i$ th iteration, the root-mean-square of which indicates how far the Hartree–Fock solution is from self-consistency.

$$\mathbf{P}_i \mathbf{F}_i - \mathbf{F}_i \mathbf{P}_i = \mathbf{e}_i . \quad (3.44)$$

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 vector,

$$\mathbf{F}_{j+1} = \sum_j c_j \mathbf{F}_j . \quad (3.45)$$

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. DIIS begins after SCF step 2, to prevent extrapolation before convergence has properly begun, and can be deactivated with the NODIIS keyword.

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

$$\mathbf{P}' = a \mathbf{P}_{\text{old}} + (1 - a) \mathbf{P} . \quad (3.46)$$

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 large value of MAXITER. A calculation with a maximum of 100 iterations and  $a = 0.5$  can be called by

```
TUNA SPE : H He 0.6 : HF 6-31+G : SLOWCONV MAXITER 100
```

However, the default method is dynamic damping, where the damping factor,  $a$ , is a function of the DIIS error. The equation used in TUNA is

$$a = 0.7 \tanh [\mathbf{F}, \mathbf{PS}] , \quad (3.47)$$

which is homemade and not necessarily any good. This dynamic damping turns off ( $a = 0$ ) when the root-mean-square DIIS error falls below 0.01 units.

In level shifting, the energies of the virtual orbitals are shifted so that after diagonalisation of the Fock matrix, the occupied and virtual orbitals mix less strongly and the calculation converges more smoothly. Level shifting is applied via a parameter,  $b$ , which updates the Fock matrix by

$$\mathbf{F}' = \mathbf{F} - b\mathbf{P} \quad (3.48)$$

Level shift is off by default because it doesn't seem to help much when damping and DIIS are enabled), but can be activated with  $b = 0.2$  by the LEVELSHIFT keyword. This keyword can also be used with a different value of  $b$  if 0.2 is not effective. For instance, a calculation using a level shift of 0.35 without DIIS can be called by:

```
TUNA SPE : H He 0.8 : UHF 6-31+G : LEVELSHIFT 0.35 NODIIS
```

### 3.2.2 Møller–Plesset Perturbation Theory

In TUNA, conventional Møller–Plesset perturbation theory [24] to second (MP2) and third (MP3) order are implemented for both restricted and unrestricted references. In addition, these methods can be “spin-component-scaled” (SCS), which has been reported to increase their accuracy [25, 26]. These methods can be called by writing MP2, SCS-MP2, MP3, or SCS-MP3, such as:

```
TUNA SPE : H H 0.6 : MP3 6-311G
```

Both conventional and spin-component-scaled MP2 and MP3 can be used with either unrestricted or restricted references. If the molecule has an even number of electrons but an unrestricted reference is desired, the method keywords UMP2, USCS-MP2, UMP3 or USCS-MP3 can be used:

```
TUNA SPE : H H 0.6 : UMP2 6-311G
```

All of these methods require an  $\mathcal{O}(N^5)$  transformation of the two-electron integrals into the spatial orbital or spin orbital basis, and MP3 requires a further  $\mathcal{O}(N^6)$  contraction to calculate the energy.

However, at the number of basis functions involved in a TUNA calculation, these steps are small compared to the time to calculate two-electron integrals. There are currently no options to freeze core electrons.

The two electron integrals are transformed into a spin-orbital basis by

$$\langle ij|ab\rangle = \sum_{\mu\nu\kappa\lambda} C_{\mu i} C_{\nu j} \langle \mu\nu|\kappa\lambda\rangle C_{\kappa a} C_{\lambda b} . \quad (3.49)$$

These are then antisymmetrised to give

$$\langle ij||ab\rangle = \langle ij|ab\rangle - \langle ij|ba\rangle . \quad (3.50)$$

Correlated calculations use these transformed integrals to calculate the energy. For instance, the MP2 energy is calculated by

$$E_{\text{MP2}} = \frac{1}{4} \sum_{ijab} \frac{|\langle ij||ab\rangle|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} . \quad (3.51)$$

In TUNA, this energy is calculated via the double excitation amplitudes,

$$t_{ij}^{ab} = \langle ij||ab\rangle (\mathcal{E}_{ij}^{ab})^{-1} , \quad (3.52)$$

where

$$\mathcal{E}_{ij}^{ab} = \varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b , \quad (3.53)$$

by

$$E_{\text{MP2}} = \frac{1}{4} \sum_{ijab} \langle ij||ab\rangle t_{ij}^{ab} . \quad (3.54)$$

These amplitudes are also used to update the occupied–occupied and virtual–virtual blocks of the density matrix, for the unrelaxed density. For instance, the occupied–occupied blocks are given by

$$P_i^j = -\frac{1}{2} \sum_{abk} t_{jk}^{ab} t_{ik}^{ab} . \quad (3.55)$$

The unrelaxed density is implemented for all MP2-based methods, but not for MP3. This density is used for population analysis and dipole moment calculations, and to calculate natural orbital occupancies, which are printed in the MP2 block in TUNA. If the additional print keyword, P, is

used, the density will also be used to compute the expectation value of the  $\hat{S}^2$  operator, and the spin contamination. The natural orbital calculation can be disabled using NONATORBS.

### 3.2.2.1 Spin-component Scaling

SCS-MP2 is a semi-empirical modification to MP2 that applies different scaling factors to the same spin,  $E_{\text{MP2}}^{\text{SS}}$ , and opposite spin,  $E_{\text{MP2}}^{\text{OS}}$ , components of the MP2 correlation energy. These calculations can be called in TUNA by:

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

In SCS-MP2, the total energy is given by

$$E_{\text{SCS-MP2}} = E_{\text{HF}} + \alpha E_{\text{MP2}}^{\text{SS}} + \beta E_{\text{MP2}}^{\text{OS}}, \quad (3.56)$$

where by default,  $\alpha = 1/3$  and  $\beta = 6/5$ . These values can be changed by the SSS and OSS keywords respectively.

In SCS-MP3, the total energy is given by

$$E_{\text{SCS-MP3}} = E_{\text{SCS-MP2}} + \gamma E_{\text{MP3}}, \quad (3.57)$$

where  $\gamma = 1/4$  by default. The value of  $\gamma$  can be changed with the MP3S keyword.

### 3.2.2.2 Orbital-optimised MP2

The MP2 energy can be minimised with respect to orbital rotations, in orbital-optimised MP2, which lowers the MP2 energy further [27, 28]. This method can be requested with the OMP2 method keyword, or UOMP2 to force an unrestricted reference on a singlet state:

```
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,  $\mathbf{R}$ , by

$$\mathbf{C}_{i+1} = \mathbf{C}_i \exp(\mathbf{R}). \quad (3.58)$$

This cycle continues until the change in energy falls below a convergence criteria of  $10^{-8}$  by default. This value can be changed with the OMP2CONV keyword. The maximum number of OMP2 iterations is 20 by default, mutable with the OMP2MAXITER keyword.

The density matrix from orbital-optimised MP2 is fully relaxed, and is used to calculate molecular properties and in population analysis.

### 3.2.3 Coupled Cluster Theory

In coupled cluster theory, the exact wavefunction is given by

$$|\Psi\rangle = e^T |\Phi_0\rangle, \quad (3.59)$$

where  $|\Phi_0\rangle$  is the reference function, a Hartree–Fock state in TUNA, and  $T$  is the cluster operator, which produces a linear combination of excited determinants. This is written in the form

$$T = T_1 + T_2 + T_3 + \dots, \quad (3.60)$$

where  $T_1$  is the operator of all single excitations,  $T_2$  is the operator of all double excitations, etc. In the formalism of second quantisation, these operators can be expressed as

$$T_1 = \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i \quad (3.61)$$

$$T_2 = \frac{1}{4} \sum_{ijab} t_{ij}^{ab} \hat{a}_b^\dagger \hat{a}_a^\dagger \hat{a}_j \hat{a}_i. \quad (3.62)$$

In these formulae,  $\hat{a}^\dagger$  and  $\hat{a}$  are the 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,  $e^T$ , as a Taylor series yields

$$e^T = 1 + T + \frac{1}{2!}T^2 + \frac{1}{3!}T^3 + \dots \quad (3.63)$$

which, upon substitution of the definition of  $T$ , yields

$$e^T = 1 + T_1 + T_2 + \frac{1}{2}T_1^2 + \frac{1}{2}T_1T_2 + \frac{1}{2}T_2T_1 + \frac{1}{2}T_2^2 + \dots. \quad (3.64)$$

Curtailling  $T$  yields approximate wavefunctions, such as curtailling  $T$  to  $T_1 + T_2$  yields the coupled cluster singles and doubles (CCSD) method, which, as demonstrated in equation 3.64, includes approximate contributions from higher than double excitations, through so-called “disconnected” excitations like  $T_1T_2$ . This makes coupled cluster more accurate 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 is always the same, no matter the curtailing. The difference in energy comes from effects from eg. triples on the converged  $t$ -amplitudes.

$$E_{CC} = E_{HF} + \frac{1}{4} \sum_{ijab} \langle ij || ab \rangle t_{ij}^{ab} + \frac{1}{2} \sum_{ijab} \langle ij || ab \rangle t_i^a t_j^b \quad (3.65)$$

In all coupled cluster methods, an iterative method is used to converge the  $t$ -amplitudes. Similarly to the Hartree–Fock energy, 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. 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 CCDAMP keyword, which, with an optional parameter determines how much old  $t$ -amplitudes should be mixed with new ones. Damping is off by default, and CCDAMP defaults to 0.25 if no parameter is given. For example, a CCSD calculation with damping at 50% and DIIS remembering 7 matrices is given by:

TUNA SPE : 0 0 0.9 : CCSD 3-21G : DIIS 7 CCDAMP 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 CCONV keyword and  $10^{-8}$  by default. Next the MP2 energy, which is calculated from the MP2  $t_{ij}^{ab}$  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 printed.

The norm of the singles amplitude and the T1 diagnostic is then printed, which gives an idea if a single reference calculation is valid. The T1 diagnostic is related to the norm of the  $t_i^a$  amplitudes by

$$T1 = \sqrt{\frac{\sum_{ia} t_i^a t_i^a}{N_{occ}}} . \quad (3.66)$$

Next, the largest singles and doubles amplitudes are searched for and printed, showing the most important excitations out of the Hartree–Fock orbitals. Finally, the linearised density is calculated and property calculations can begin.

### 3.2.3.1 Coupled Cluster Doubles

In CCD, the cluster operator is curtailed to

$$T = T_2 , \quad (3.67)$$

giving equations where  $t_{ij}^{ab}$  needs to be optimised. In TUNA, CCD can be requested with the CCD method type. For example, a CCD single point energy calculation on carbon monoxide, in the 6-31+G basis set, can be called by:

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

### 3.2.3.2 Coupled Cluster Singles and Doubles

In CCSD, the cluster operator is curtailed to

$$T = T_1 + T_2 , \quad (3.68)$$

giving equations where  $t_i^a$  and  $t_{ij}^{ab}$  need to be optimised together. 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 slow, with formal  $\mathcal{O}(N^6)$  scaling with respect to basis functions.

### 3.2.3.3 Coupled Cluster Singles, Doubles and Triples

In CCSDT, the cluster operator is curtailed to

$$T = T_1 + T_2 + T_3 , \quad (3.69)$$

giving equations where  $t_i^a$ ,  $t_{ij}^{ab}$  and  $t_{ijk}^{abc}$  need to be optimised together. The CCSDT method can be requested using CCSDT. For example, a CCSDT single point energy on the lithium atom in the cc-pVDZ basis set is called by:

```
TUNA SPE : Li : CCSDT CC-PVDZ
```



As lithium has three electrons, the CCSDT energy corresponds to the full CI energy. The CCSD method is very slow, with formal  $\mathcal{O}(N^8)$  scaling with respect to basis functions.

#### 3.2.3.4 Coupled Cluster Singles, Doubles and 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 triple excitations. This provides approximate treatment of triples, which often manages to give better results than CCSDT, with lower  $\mathcal{O}(N^7)$  scaling, due to the undershooting of triples resembling the inclusion of quadruple excitations.

Because TUNA is run entirely from the terminal, and round parentheses are not allowed there, CCSD(T) is called by the CCSD[T] method keyword. An optimisation of Li-H in the 6-311G\*\* basis is given by:

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

#### 3.2.3.5 Linearised Coupled Cluster

Linearised coupled cluster makes quite a dramatic approximation to  $e^T$ , keeping only terms to first order

$$e^T \approx 1 + T = 1 + T_1 + T_2 + T_3 + \dots \quad (3.70)$$

This comes at the cost of size-extensivity and is normally less accurate than conventional coupled cluster, however computation of the equations is made 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. If the NOSINGLES keyword is used in conjunction with the CEPA method, this will instead use LCCD. A calculation using CEPA0, with single excitations (unshifted) on the carbon atom is shown:

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

### 3.2.3.6 Coupled Cluster 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.

However, this is not the 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.

### 3.2.4 Excited States

Excited state calculations are available in TUNA using the configuration interaction singles (CIS) method, with an optional perturbative doubles correction. This module can be used to calculate excitation energies and absorption intensities. Any excited state method for energies can be used with optimisations, harmonic frequency calculations and, in principle, MD simulations. In reality, there is significant risk to doing this, as unintended crossings between excited state potential energy surfaces may occur. Excited state calculations are available for both RHF and UHF references. The default SCF convergence criteria is set to TIGHT for all excited state calculations.

#### 3.2.4.1 Configuration Interaction Singles

The configuration interaction single excitations method is the analogy of Hartree–Fock for excited states [29], and therefore not very good. CIS in TUNA is not yet spin-adapted.

In CIS, the wavefunction is expressed as a linear combination of singly-excited determinants,

$$|\Psi_{\text{CIS}}\rangle = \sum_{ia} c_i^a |\Psi_i^a\rangle . \quad (3.71)$$

The weights of these determinants,  $c_i^a$ , are determined by diagonalisation of the shifted CIS Hamiltonian, whose elements, in the basis of determinants, are

$$H_{ia,jb} = (\varepsilon_a - \varepsilon_j) \delta_{ij} \delta_{ab} - \langle ja || ib \rangle . \quad (3.72)$$

The eigenvalues of this matrix give the excitation energies from the reference Hartree–Fock determinant and the excited states,  $\omega$ . TUNA prints these for each state, along with the percentage contribution of each transition between occupied and virtual orbitals to the state. The default threshold for the contributions to be printed is 1%, but this can be changed with the CISTHRESH 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. For restricted Hartree–Fock references, whether the excited state is a singlet or a triplet is also printed.

After the percentage contributions are calculated, TUNA determines the transition dipoles between the excited and ground state,  $\mu$ , and uses these to find the oscillator strengths, by

$$f_{\text{osc}} = \frac{2}{3} \omega |\mu|^2. \quad (3.73)$$

These values, along with the excitation energy in eV, excitation frequency in  $\text{cm}^{-1}$  and excitation wavelength in nm, are printed in the CIS absorption spectrum in TUNA.

Finally, the CIS unrelaxed density matrix is calculated using the weights matrix. For instance, the change to the occupied–occupied block upon excitation is

$$\Delta P_i^j = - \sum_{ab} b_i^a b_j^a. \quad (3.74)$$

This is added to the Hartree–Fock density matrix to give the full CIS 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 excited state by default, but can be changed using the ROOT keyword. This determines for which state the density matrix is calculated, as well as which energy is printed at the end of the single point calculation. Therefore, if a geometry optimisation is desired on the fourth excited state of  $\text{H}_2$ , the following should be requested:

```
TUNA OPT : H H 1.0 : CIS 6-311G : ROOT 4
```

Note that TUNA does not currently have a root-following algorithm, so its easy for potential energy surfaces to get mixed up part way through an optimisation, (or coordinate scan).

Excited states for singlet molecules can be requested with an unrestricted reference, with the UCIS method. For example, an unrestricted CIS calculation on  $\text{H-He}^+$  where the second excited state will be used for population analysis and lots of information will be printed about the percentage contribution to the states, is:

```
TUNA SPE : H He 1.0 : CIS 6-31++G : P ROOT 2 CH 1 CISTHRESH 0.1
```

### 3.2.4.2 Perturbative Doubles Correction to CIS

The CIS(D) method is analogous to the MP2 correlation perturbative correction to the Hartree–Fock ground state, to an excited state’s excitation energy [30]. This has been shown to improve excitation energies considerably compared to CIS.

This method is callable in TUNA with the CIS[D] keyword — normal parentheses don’t work nicely in the terminal — and the correction will be applied to a state of interest only, as, like MP2, CIS(D) scales with  $\mathcal{O}(N^5)$  with basis functions.

The correction to the excitation energy  $\omega$ , 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,

$$\omega_{\text{CIS(D)}} = \frac{1}{4} \sum_{ijab} \frac{(u_{ij}^{ab})^2}{\mathcal{E}_{ij}^{ab} + \omega} + \sum_{ia} b_i^a v_i^a. \quad (3.75)$$

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 is requested by:

```
TUNA OPT : H H 1.0 : CIS[D] 6-311G : ROOT 4
```

## 3.3 Basis Sets and Molecular Integrals

One of the first steps in all TUNA calculations is the calculation of integrals over basis functions, which are necessary to compute expectation values such as the energy.

Molecular integrals in TUNA are computed with the McMurchie-Davidson scheme [31]. Here, recurrence relations between Hermite Gaussians simplify the calculation.

The molecular integrals are stored in the `tuna_integral` module, which is written in Cython [32]. 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.

This basis set usually consists of partially contracted Gaussian functions, but can be fully decon-

tracted with the DECONTRACT keyword. For example, the following command will dedicate 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

Basis options are listed in Table 3.8, and are taken from the Basis Set Exchange [33].

**Table 3.8** Basis sets implemented in TUNA

| Basis Set | Citation | Description                                                      |
|-----------|----------|------------------------------------------------------------------|
| STO-3G    | [34]     | Minimal basis set                                                |
| STO-6G    | [34]     | Minimal basis set                                                |
| 3-21G     | [35]     | Double-zeta basis set                                            |
| 4-31G     | [36]     | Double-zeta basis set                                            |
| 6-31G     | [36]     | Double-zeta basis set                                            |
| 6-31+G    | [36]     | Double-zeta basis set with diffuse functions on heavy atoms      |
| 6-31++G   | [37]     | Double-zeta basis set with diffuse functions on all atoms        |
| 6-31G*    | [38]     | Double-zeta basis set with polarisation functions on heavy atoms |
| 6-31G**   | [39]     | Double-zeta basis set with polarisation functions on all atoms   |
| 6-311G    | [40]     | Triple-zeta basis set                                            |
| 6-311+G   | [37]     | Triple-zeta basis set with diffuse functions on heavy atoms      |
| 6-311++G  | [37]     | Triple-zeta basis set with diffuse functions on all atoms        |
| 6-311G*   | [40]     | Triple-zeta basis set with polarisation functions on heavy atoms |
| 6-311G**  | [40]     | Triple-zeta basis set with polarisation functions on all atoms   |
| cc-pVDZ   | [41, 42] | Dunning polarised valence double-zeta basis set                  |
| cc-pVTZ   | [41, 42] | Dunning polarised valence triple-zeta basis set                  |
| cc-pVQZ   | [41, 42] | Dunning polarised valence quadruple-zeta basis set               |
| cc-pV5Z   | [41, 42] | Dunning polarised valence quintuple-zeta basis set               |
| cc-pV6Z   | [43]     | Dunning polarised valence sextuple-zeta basis set                |

## 4 List of Keywords

Table 4.1 lists all of the mandatory and optional keywords in TUNA, grouped by their functions.

**Table 4.1** Parameter keywords in TUNA

| Parameter                      | Description                                    |
|--------------------------------|------------------------------------------------|
| P                              | Print more information to the terminal         |
| T                              | Print less information to the terminal         |
| CH [int] or CHARGE [int]       | Molecular charge                               |
| ML [int] or MULTIPLICITY [int] | Molecular multiplicity                         |
| DECONTRACT                     | Decontract basis set                           |
| ROTATE [float]                 | Rotate orbitals for initial guess (by degrees) |
| NOROTATE                       | Don't rotate orbitals for initial guess        |
| STHRESH [float]                | Threshold for overlap matrix eigenvalues       |
| M1 [float]                     | Custom mass for first atom                     |
| M2 [float]                     | Custom mass for second atom                    |
| LOOSE or LOOSESCF              | Loose SCF convergence                          |
| NORMAL or NORMALSCF            | Normal SCF convergence                         |
| TIGHT or TIGHTSCF              | Tight SCF convergence                          |
| EXTREME or EXTREMESCF          | Extreme SCF convergence                        |
| DAMP                           | Dynamic damping for SCF convergence            |
| NODAMP                         | No dynamic damping for SCF convergence         |
| LEVELSHIFT [float]             | Level shift for SCF convergence, and parameter |
| NOLEVELSHIFT                   | No level shift for SCF convergence             |

| Parameter                        | Description                                        |
|----------------------------------|----------------------------------------------------|
| 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                   |
| LOOSEOPT                         | Loose geometry convergence                         |
| NORMALOPT                        | 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                     |
| 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               |
| STEP [float] or SCANSTEP [float] | Distance increment for coordinate scan             |
| NUM [int] or SCANNUMBER [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       |

| Parameter                           | Description                                        |
|-------------------------------------|----------------------------------------------------|
| 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        |
| STEP [float] or TIMESTEP [float]    | Timestep for MD simulation                         |
| NUM [int] or MDNUMBER [int]         | 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     |
| SSS [float]                         | Same-spin scaling for SCS-MP2                      |
| OSS [float]                         | Opposite-spin scaling for SCS-MP2                  |
| MP3S [float]                        | Scaling for SCS-MP3                                |
| OMP2CONV [float]                    | Convergence criteria for OMP2                      |
| OMP2MAXITER [int]                   | Maximum iterations for OMP2                        |
| CCCONV [float]                      | Convergence criteria for coupled cluster           |
| CC2MAXITER [int]                    | Maximum iterations for coupled cluster             |
| CCDAMP [float]                      | Use damping in coupled cluster, parameter          |
| NOSINGLES                           | No single excitations                              |
| NATORBS                             | Calculate natural orbitals                         |
| NONATORBS                           | Do not calculate natural orbitals                  |
| CISTHRESH [float]                   | Threshold for printing excited state contributions |
| ROOT [int]                          | Choice of excited state                            |
| NSTATES [int]                       | Number of states to print                          |



# 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.
- (2) 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.
- (3) F. Neese, *WIREs Comput. Mol. Sci.*, 2012, **2**, 73–78.
- (4) F. Neese, *WIREs Comput. Mol. Sci.*, 2022, **12**, e1606.
- (5) 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.
- (6) P. J. Mohr, E. Tiesinga, D. B. Newell and B. N. Taylor, *Codata Internationally Recommended 2022 Values of the Fundamental Physical Constants*, en, 2024.
- (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, Í. 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, *Nature Methods*, 2020, **17**, 261–272.
- (8) *TermColor*, <https://pypi.org/project/termcolor/>.
- (9) J. D. Hunter, *Comput. Sci. Eng.*, 2007, **9**, 90–95.
- (10) T. Koopmans, *Physica*, 1934, **1**, 104–113.
- (11) R. S. Mulliken, *J. Chem. Phys.*, 1955, **23**, 1833–1840.
- (12) P.-O. Löwdin, *J. Chem. Phys.*, 1950, **18**, 365–375.
- (13) I. Mayer, *Chem. Phys. Lett.*, 1983, **97**, 270–274.
- (14) E. R. Davidson and A. E. Clark, *Int. J. Quantum Chem.*, 2022, **122**, e26860.

- (15) S. Grimme, *J. Comput. Chem.*, 2006, **27**, 1787–1799.
- (16) J. Neugebauer, M. Reiher, C. Kind and B. A. Hess, *J. Comput. Chem.*, 2002, **23**, 895–910.
- (17) R. Iftimie, P. Miny and M. E. Tuckerman, *Proc. Natl Acad. Sci.*, 2005, **102**, 6654–6659.
- (18) R. Car and M. Parrinello, *Phys. Rev. Lett.*, 1985, **55**, 2471–2474.
- (19) W. C. Swope, H. C. Andersen, P. H. Berens and K. R. Wilson, *J. Chem. Phys.*, 1982, **76**, 637–649.
- (20) C. C. J. Roothaan, *Rev. Mod. Phys.*, 1951, **23**, 69–89.
- (21) G. G. Hall, *Proc. Roy. Soc. London A*, 1951, **205**, 541–552.
- (22) D. R. Hartree and W. Hartree, *Proc. Roy. Soc. London A*, 1935, **150**, 9–33.
- (23) P. Pulay, *Chem. Phys. Lett.*, 1980, **73**, 393–398.
- (24) C. Møller and M. S. Plesset, *Phys. Rev.*, 1934, **46**, 618–622.
- (25) S. Grimme, *J. Chem. Phys.*, 2003, **118**, 9095–9102.
- (26) S. Grimme, *J. Comput. Chem.*, 2003, **24**, 1529–1537.
- (27) R. C. Lochan and M. Head-Gordon, *J. Chem. Phys.*, 2007, **126**, 164101.
- (28) F. Neese, T. Schwabe, S. Kossmann, B. Schirmer and S. Grimme, *J. Chem. Theory Comput.*, 2009, **5**, 3060–3073.
- (29) J. B. Foresman, M. Head-Gordon, J. A. Pople and M. J. Frisch, *J. Phys. Chem.*, 1992, **96**, 135–149.
- (30) M. Head-Gordon, R. J. Rico, M. Oumi and T. J. Lee, *Chem. Phys. Lett.*, 1994, **219**, 21–29.
- (31) L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.*, 1978, **26**, 218–231.
- (32) S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn and K. Smith, *Comput. Sci. Eng.*, 2011, **13**, 31–39.
- (33) B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibbsom and T. L. Windus, *J. Chem. Inf. Model.*, 2019, **59**, 4814–4820.
- (34) W. J. Hehre, R. F. Stewart and J. A. Pople, *J. Chem. Phys.*, 1969, **51**, 2657–2664.
- (35) J. S. Binkley, J. A. Pople and W. J. Hehre, *J. Am. Chem. Soc.*, 1980, **102**, 939–947.
- (36) R. Ditchfield, W. J. Hehre and J. A. Pople, *J. Chem. Phys.*, 1971, **54**, 724–728.
- (37) T. Clark, J. Chandrasekhar, G. W. Spitznagel and P. V. R. Schleyer, *J. Comput. Chem.*, 1983, **4**, 294–301.
- (38) V. A. Rassolov, M. A. Ratner, J. A. Pople, P. C. Redfern and L. A. Curtiss, *J. Comput. Chem.*, 2001, **22**, 976–984.
- (39) 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.
- (40) R. Krishnan, J. S. Binkley, R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1980, **72**, 650–654.
- (41) J. Dunning, Thom H., *J. Chem. Phys.*, 1989, **90**, 1007–1023.
- (42) D. E. Woon and J. Dunning, Thom H., *J. Chem. Phys.*, 1993, **98**, 1358–1371.
- (43) A. K. Wilson, T. van Mourik and T. H. Dunning, *J. Mol. Struct.*, 1996, **388**, 339–349.