TUNA

Theoretical Unification of Nuclear Arrangements

Version 0.5.0

September 2024

Harry Brough

Foreword

Welcome to TUNA!

This is a personal project to deepen my understanding of the theory and algorithmic implementation of quantum chemistry. A simple to use terminal-based Python program, TUNA can calculate properties of diatomic molecules with a number of 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 rather 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 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 on diatomics.

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

Note — This manual was thrown together in a few days so equations and will definitely have errors!

Contents

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), 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 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. However, everything should work fine on a different operating system or with earlier versions of Python 3. 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] must be omitted. If two atoms are provided, a [Distance] must be too. Available calculation types and basis sets are discussed in section 3.1, and electronic structure methods are detailed in section 3.2.

For instance, a single point energy calculation on an H_2 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.

The output prints to the terminal by default, but by replacing the command `TUNA` with `TUNALOG` the output will print to a file called "tuna-calculation.log" in the terminal's directory.

1.3   Units and Constants

In the TUNA output, distances are printed in angstroms, times in femtoseconds and all other quantities 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 constants.

Table 1.1 Unit conversions and fundamental constants 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	$4\pi\varepsilon_0$	$1.11265005620 \times 10^{-10}$ 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}
Atomic Mass Unit	amu	$1.660539068911 \times 10^{-27}$ kg

1.4   Program Components

Table 1.2 lists the functions of the different Python modules in the TUNA program, which talk to each other during a calculation. The main program, `tuna.py`, initiates all calculations when the `TUNA` is called from a terminal.

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
```

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_dispersion	Semi-empirical dispersion correction
tuna_energy	Calculating energies, building molecules, coordinate scans
tuna_integral	Evaluating one- and two-electron integrals
tuna_md	*Ab initio* molecular dynamics
tuna_mpn	Møller–Plesset perturbation theory energy and density
tuna_optfreq	Calculating gradients, optimisation algorithm, harmonic frequencies
tuna_postscf	Calculating and printing molecular properties
tuna_scf	Main self-consistent field loop, convergence acceleration
tuna_thermo	Thermochemistry after frequency calculation
tuna_util	General utility, units, frequently used functions

In addition to these modules, some external modules are also imported. All these should automatically be installed when `pip install QuantumTUNA` is run. The most important are NumPy [5] and 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], Scikit-Image [10] and Plotly [11] enable the creation of two- and three-dimensional plots.

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`)
- Unrestricted MP2 (`UMP2`)
- Spin-component-scaled MP2 (`SCS–MP2`)
- MP3 (`MP3`)
- Unrestricted MP3 (`UMP3`)
- Spin-component-scaled MP3 (`SCS–MP3`)

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

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

- Electron density calculation and 3D plot

- Molecular orbitals and orbital energies

- Rotational constants

- Thermochemistry

- Restricted MP2 unrelaxed density, properties and natural orbitals

2.5   Miscellaneous

- D2 dispersion correction

- Interface to GUI for plots for coordinate scan and electron density

- 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 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	*Ab initio* molecular dynamics

All calculation types run on an initial molecule, defined by its atoms, charge and multiplicity. At the moment, only Gaussian integrals between s orbitals are implemented, so TUNA is (severely) limited to Period 1 elements with unpolarised basis sets — TUNA is breaking new ground as the only program with spin-component-scaled MP3 but no carbon atoms! So, the only atoms available are H and He.

However, "ghost atoms" for these 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 "XH" or "XHe". For example, the following command will calculate the energy of a hydrogen atom in the presence of a ghost hydrogen atom's basis functions, lowering the energy compared to the calculation on a lone atom:

```
TUNA SPE : H XH 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 exiting with an error.

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

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

Alternatively, shortened parameters for charge and multiplicity, CH and ML can 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 the 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, including the basis set. Basis options are listed in Table 3.2.

Table 3.2 Basis sets implemented in TUNA

Basis Set	Citation	Description
STO-3G	[12]	Minimal basis set
STO-6G	[12]	Minimal basis set
3-21G	[13]	Double-zeta basis set
4-31G	[14]	Double-zeta basis set
6-31G	[14]	Double-zeta basis set
6-31+G	[14]	Double-zeta basis set with diffuse functions on heavy atoms
6-31++G	[15]	Double-zeta basis set with diffuse functions on all atoms
6-311G	[16]	Triple-zeta basis set
6-311+G	[15]	Triple-zeta basis set with diffuse functions on heavy atoms
6-311++G	[15]	Triple-zeta basis set with diffuse functions on all atoms

This basis set usually consists of partially contracted Gaussian functions, but can be fully decontracted 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
```

The molecule is then set up as requested, and the molecular structure, number of atoms, number of basis functions, number of primitive Gaussians, charge, multiplicity, number of electrons, 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 as soon as 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 atoms:  2
 Number of basis functions:  6
 Number of primitive Gaussians:  10
 Charge:  1
 Multiplicity:  1
 Number of electrons:  2
 Point group:  Cinfv
 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, before 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. If a correlated calculation (eg. MP2, MP3) is requested, this then begins.

After all the energies are calculated, the density matrix is used in property calculations. This is the Hartree–Fock density unless a closed-shell MP2 or MP3 calculation is requested, in which case the unrelaxed MP2 density is used. Calculations of rotational constants, multipole moments and population analysis is 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. 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 [17]. These are the ionisation energy, $-\epsilon_{\text{HOMO}}$, electron affinity, $-\epsilon_{\text{LUMO}}$ and HOMO–LUMO gap, $\epsilon_{\text{LUMO}} - \epsilon_{\text{HOMO}}$.

For unrestricted Hartree–Fock calculations, the $alpha$-spin eigenvalues are used.

3.1.1.2 Rotational Constant

Next, the rotational constant of the molecule is printed, calculated by

$$B = \frac{1}{2\mu(\mathbf{R_A} - \mathbf{R_B})^2} \; , \tag{3.1}$$

where μ is the reduced mass of the molecule, using the masses in section 3.1.4. The value printed by TUNA has units of GHz.

3.1.1.3 Multipole Moments

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

$$\boldsymbol{\mu}_{\text{nuc}} = Z_A(\mathbf{R_A} - \mathbf{R}_0) + Z_B(\mathbf{R_B} - \mathbf{R}_0) \; , \tag{3.2}$$

and the electronic dipole moment,

$$\boldsymbol{\mu}_{\text{ele}} = -\sum_{\mu\nu} P_{\mu\nu} D_{\mu\nu} \; , \tag{3.3}$$

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$^+$ at 2Å gives this output, meaning the hydrogen atom is more negative than the helium atom:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                       Dipole Moment
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     Nuclear:   −1.498793      Electronic:   −1.477457



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

3.1.1.4   Population Analysis

By default, TUNA prints Mulliken [18], Löwdin [19] and Mayer [20] 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.

The Mulliken analysis [18] is very widely used (despite its considerable weaknesses [21]), 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

$$N = \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} \,. \tag{3.4}$$

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

$$N_{\mathsf{A}} = \sum_{\mu\in\mathsf{A}} \sum_{\nu\in\mathsf{A}} P_{\mu\nu} S_{\mu\nu} + \frac{1}{2} B_{\mathsf{AB}} \,, \tag{3.5}$$

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

$$B_{\mathsf{AB}} = 2 \sum_{\mu\in\mathsf{A}} \sum_{\nu\in\mathsf{B}} P_{\mu\nu} S_{\mu\nu} \,. \tag{3.6}$$

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 than generate a specific energy and density are not unique, population analysis can be performed with non-localised orthogonal basis functions, in Löwdin analysis [19] where the density matrix is rotated by

$$\mathbf{P}^L = \mathbf{S}^{\frac{1}{2}} \mathbf{P} \mathbf{S}^{\frac{1}{2}} . \tag{3.7}$$

Using this density matrix, atomic populations are calculated by

$$N_A = \sum_{\mu \in A} P_{\mu\mu}^L , \tag{3.8}$$

and the bond order is calculated by

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

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

The results of a Mayer analysis [20] are also printed. The Mayer charges are the same as the Mulliken charges, but the bond order (and bonded valence) is given by

$$B_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} (\mathbf{PS})_{\mu\nu} (\mathbf{PS})_{\nu\mu} + (\mathbf{RS})_{\mu\nu} (\mathbf{RS})_{\nu\mu} , \tag{3.10}$$

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), is

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

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

3.1.1.5 Electron Density Plot

A three-dimensional plot of the electron density can be calculated using the `DENSPLOT` keyword after an energy calculation. This opens a Plotly [11] image in your default browser.

This can be combined with optimisations to show the change in electron density over time, which is fun. However, the calculation of the cube files is very slow and the default parameters to view the iso-surface may crop the density incorrectly in some cases.

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 `SCANSTEP` keyword, and the total number of scan steps is specified by the `SCANNUMBER` 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 with 20 steps of 0.1 Å can be requested by:

```
TUNA SCAN : H H 0.3 : MP2 6-31G : SCANNUMBER 20 SCANSTEP 0.1
```

By default, the previous step's density matrix is read in at each scan step; 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`, `DENSPLOT`, `ROTATE`) can also be combined with the scan calculation. In addition, the `SCANPLOT` keyword will make a simple Matplotlib [9] graph of the potential energy surface. At the end of a coordinate scan calculation, a table of the bond lengths and final energies is printed, which can be easily copy and pasted into a spreadsheet.

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, such as:

```
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 strategy can be disabled using the NOMOREAD keyword.

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

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

where r is the interatomic distance. Next, an approximation to the Hessian, H, is calculated by

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

Instead of this approximate Hessian, the exact Hessian can be calculated using the CALCHESS keyword. This increases calculation time at each step significantly, but reduces the number of necessary iterations. For diatomics, the exact Hessian calculation is almost always 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.

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

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

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, to prevent exploding optimisations from a bad starting guess geometry. 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, normal, tight (default) and extreme. The gradient and step convergence criteria of these are listed in Table 3.3. 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 20 by default.

Table 3.3 Optimisation convergence criteria in TUNA

Convergence	Gradient / hartree bohr^{-1}	Step / bohr
Loose	10^{-3}	10^{-2}
Normal	10^{-4}	10^{-4}
Tight	10^{-6}	10^{-5}
Extreme	10^{-8}	10^{-7}

When the gradient and step fall below the convergence criteria, the optimisation will stop, 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, 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_2.

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

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

18

3.1.3.1 Dispersion Correction

Currently, the only available form of dispersion correction is the D2 semi-empirical scheme by Grimme [22]. Here, the energy due to dispersion is calculated by

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

where the damping function is given by

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

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. 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).

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 OPTRFEQ.

```
TUNA OPTFREQ : H H 1.0 : SCS-MP2 6-311++G
```

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

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

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.

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

$$\omega = \sqrt{\frac{k}{\mu}} \, , \tag{3.18}$$

where μ is the reduced mass, calculated using the atomic masses shown in Table 3.4.

Table 3.4 Atomic masses in TUNA

Atom	Major Isotope	Mass / amu
Hydrogen	^{1}H	1.00782503223
Helium	^{4}He	4.00260325413

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 appended with an "`i`".

The harmonic frequencies of H_2 calculated from all the different methods implemented in TUNA are shown in Table 3.5, where SCS-MP3 performs very well.

Table 3.5 Harmonic frequencies of H_2 calculated in the 6-311++g basis set in TUNA

Method	Harmonic Frequency / cm^{-1}
HF	4577
MP2	4456
SCS-MP2	4431
MP3	4367
SCS-MP3	4409
Exact	4401

3.1.4.1 Thermochemistry

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

$$U = E + E_{\text{zero-point}} + E_{\text{translational}} + E_{\text{vibrational}} + E_{\text{rotational}} \;, \tag{3.19}$$

and entropy,

$$S = S_{\text{electronic}} + S_{\text{translational}} + S_{\text{vibrational}} + S_{\text{rotational}} \;, \tag{3.20}$$

are first calculated, before the enthalpy,

$$H = U + k_{\text{B}}T \;, \tag{3.21}$$

and Gibbs free energy,

$$G = H - TS \;, \tag{3.22}$$

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

$$E_{\text{vibrational}} = \frac{\omega}{\exp\left(\omega/k_{\text{B}}T\right) - 1} \;, \tag{3.23}$$

assuming the molecule behaves as a harmonic oscillator — at high temperatures this may be a particularly bad poor approximation.

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:

$$S_{\text{translational}} = k_{\text{B}}\left[\frac{5}{2} + \ln\left(\frac{Mk_{\text{B}}T}{h}\right)^{\frac{3}{2}} + \ln\left(\frac{k_{\text{B}}T}{p}\right)\right] \tag{3.24}$$

$$S_{\text{vibrational}} = k_{\text{B}}\left[\frac{\omega}{k_{\text{B}}T\left(\exp(\omega/k_{\text{B}}T) - 1\right)} - \ln\left(1 - \exp\left(-\frac{\omega}{k_{\text{B}}T}\right)\right)\right] \tag{3.25}$$

$$S_{\text{rotational}} = k_{\text{B}} \left[1 + \ln \left(\frac{k_{\text{B}}T}{2\pi\sigma Bc} \right) \right] \tag{3.26}$$

For the rotational entropy, the symmetry number, σ, 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 [23], 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 [24], 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 TIMESTEP, TEMP and MDNUMBER respectively. By default, TIMESTEP is 0.1 fs, TEMP is 0 K and MDNUMBER is 30.

For instance, a 50 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 50
```

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

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2, \tag{3.27}$$

and velocities are updated by

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t + \Delta t)}{2}\Delta t \ . \tag{3.28}$$

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

$$K = 3k_\mathrm{B}T \ , \tag{3.29}$$

where

$$K = \frac{1}{2}m_1\mathbf{v}_1^2 + \frac{1}{2}m_2\mathbf{v}_2^2 \ . \tag{3.30}$$

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. The phrase "default spin" indicates that this will activate either a restricted or unrestricted Hartree–Fock calculation depending on the multiplicity of the requested molecule. A singlet molecule will default to RHF, while other multicplcities default to UHF. Unrestricted references can be requested for singlet states, but restricted references can not be requested for multiplet states.

Table 3.6 Electronic structure methods implemented in TUNA

Keyword	Electronic Structure Method
HF	Hartree–Fock theory (default spin)
UHF	Unrestricted Hartree–Fock theory
MP2	Second-order Møller–Plesset perturbation theory (default spin)
UMP2	Unrestricted second-order Møller–Plesset perturbation theory
SCS-MP2	Spin-component-scaled second-order Møller–Plesset perturbation theory
MP3	Third-order Møller–Plesset perturbation theory (default spin)
UMP3	Unrestricted third-order Møller–Plesset perturbation theory
SCS-MP3	Spin-component-scaled third-order Møller–Plesset perturbation theory

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 [26, 27] turn the variational differential restricted Hartree–Fock equations [28] into a simpler eigenvalue problem,

$$\mathbf{F}\mathbf{C} = \mathbf{S}\mathbf{C}\epsilon , \tag{3.31}$$

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 ϵ is a diagonal matrix of eigenvalues. The Fock matrix consists of contributions from the kinetic energy matrix, $\mathbf{T}$, nuclear–electron attraction matrix, $\mathbf{F}_{ne}$, Coulomb matrix, $\mathbf{J}$, and exchange matrix, $\mathbf{K}$, by

$$\mathbf{F} = \mathbf{T} + \mathbf{V}_{ne} + 2\mathbf{J} - \mathbf{K} . \tag{3.32}$$

These matrices are determined from the one- and two-electron integrals of basis functions. 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

$$\mathbf{F}' = \mathbf{X}^\dagger \mathbf{F} \mathbf{X} \,, \tag{3.33}$$

the Roothaan–Hall equations become a conventional eigenvalue problem,

$$\mathbf{F}'\mathbf{C}' = \mathbf{C}'\boldsymbol{\epsilon} \,. \tag{3.34}$$

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

$$\mathbf{C} = \mathbf{X}\mathbf{C}' \,. \tag{3.35}$$

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

$$P_{\mu\nu} = 2\sum_i^{\text{occ}} C_{\mu i} C_{\nu i}^* \,. \tag{3.36}$$

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,

$$E = V_{\text{nn}} + \frac{1}{2}\sum_{\mu\nu} P_{\mu\nu}(F_{\mu\nu} + H_{\mu\nu}^{\text{core}}) \,. \tag{3.37}$$

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,

$$2\mathbf{J} - \mathbf{K} = \sum_{\kappa\lambda} P_{\kappa\lambda}\left(\langle\mu\nu|\kappa\lambda\rangle - \frac{1}{2}\langle\mu\nu|\lambda\kappa\rangle\right) \,. \tag{3.38}$$

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

$$\mathbf{F} = \mathbf{T} + \mathbf{V}_{\text{ne}} \,. \tag{3.39}$$

While this can cause problems for polyatomic molecules, it normally works fine for diatomics, so no other guess options are implemented currently.

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 — 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

$$\begin{bmatrix} \psi'_{HOMO} \\ \psi'_{LUMO} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \psi_{HOMO} \\ \psi_{LUMO} \end{bmatrix} \tag{3.40}$$

By default, $\theta = 45°$. 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 θ is not converging well, use the `THETA` keyword followed by the desired rotation angle in degrees to change it.

An example of forcing 45° 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 THETA 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`, `NORMAL` (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`.

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) [29].

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

$$\mathbf{PF} - \mathbf{FP} = \mathbf{0} \; . \tag{3.41}$$

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

Table 3.7 Self-consistent field convergence criteria in TUNA

Convergence	ΔE	MAX($\Delta \mathbf{P}$)	RMS($\Delta \mathbf{P}$)	$[\mathbf{F}, \mathbf{PS}]$
LOOSE	10^{-6}	10^{-5}	10^{-6}	10^{-4}
NORMAL	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}

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

$$\mathbf{P}_i\mathbf{F}_i - \mathbf{F}_i\mathbf{P}_i = \mathbf{e}_i \ . \tag{3.42}$$

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,

$$\mathbf{F}_{j+1} = \sum_j c_j\mathbf{F}_j \ . \tag{3.43}$$

DIIS is very effective for closed shell systems, but can run into problems for radicals. TUNA stores 10 previous Fock matrices at each SCF step, 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.

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

$$\mathbf{P}' = a\mathbf{P}_{\text{old}} + (1-a)\mathbf{P} \ . \tag{3.44}$$

Static damping, where the value of a is fixed, can be invoked with the SLOWCONV or VERYSLOWCONV keywords, where $a = 0.5$ and $a = 0.15$ 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

$$a = 0.7 \tanh [\mathbf{F}, \mathbf{PS}] , \tag{3.45}$$

which is homemade and not necessarily any good. This dynamic damping turns off ($a = 0$) when the 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

$$\mathbf{F}' = \mathbf{F} - b\mathbf{P} \tag{3.46}$$

Level shift is off by default because it doesn't seem to help much when damping and DIIS are enabled (based on my testing), but can be activated with $b = 0.2$ by the LEVELSHIFT keyword.

3.2.2 Møller–Plesset Perturbation Theory

In TUNA, conventional Møller–Plesset perturbation theory [30] to second (MP2) and third (MP3) order are implemented. In addition, these methods can be "spin-component-scaled" (SCS), which has been reported to increase their accuracy [31, 32]. 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
```

Conventional MP2 and MP3 can be used with either unrestricted or restricted references, whereas spin-component-scaled methods are only available for restricted references presently. If the 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 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 negligible compared to the time to calculate two-electron integrals.

For restricted Hartree–Fock references, the two electron integrals are transformed into a spatial

molecular orbital basis by

$$\langle ij|ab\rangle = \sum_{\mu\nu\kappa\lambda} C_{\mu i} C_{\nu j} \langle \mu\nu|\kappa\lambda\rangle C_{\kappa a} C_{\lambda b} \,, \tag{3.47}$$

while for unrestricted references they are transformed into a spin orbital basis at higher computational cost. All MP3 calculations, regardless of reference, are performed in a spin orbital basis. There are currently no frozen core options.

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

$$E_{\text{MP2}} = E_{\text{HF}} + \frac{1}{4} \sum_{ijab} \frac{|\langle ij||ab\rangle|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,. \tag{3.48}$$

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
```

The same spin contribution is

$$E_{\text{MP2}}^{\text{SS}} = -\sum_{ijab} \frac{\langle ij|ab\rangle \langle ij|ab\rangle}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,, \tag{3.49}$$

and the opposite spin contribution is

$$E_{\text{MP2}}^{\text{SS}} = -\sum_{ijab} \frac{\langle ij|ab\rangle \left[\langle ij|ab\rangle - \langle ij|ba\rangle \right]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \,. \tag{3.50}$$

In SCS-MP2, the total energy is given by

$$E_{\text{SCS-MP2}} = E_{\text{HF}} + \alpha E_{\text{MP2}}^{\text{SS}} + \beta E_{\text{MP2}}^{\text{OS}} \,, \tag{3.51}$$

where by default, $\alpha = 1/3$ and $\beta = 6/5$.

In SCS-MP3, the total energy is given by

$$E_{\text{SCS-MP3}} = E_{\text{SCS-MP2}} + \gamma E_{\text{MP3}} \ , \tag{3.52}$$

where $\gamma = 1/4$ by default.

Restricted, conventional MP2 calculations also evaluate the unrelaxed density matrix (which neglects orbital relaxation) for property calculations and diagonalises this density for the natural orbitals, which are printed in TUNA's MP2 block. The unrelaxed density is not yet implemented for spin-component-scaled methods, unrestricted references or MP3.

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	Rotate orbitals for initial guess
NOROTATE	Don't rotate orbitals for initial guess
THETA [int]	Angle for guess orbitals rotation
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	Level shift for SCF convergence
NOLEVELSHIFT	No level shift for SCF convergence
DIIS	Fock matrix extrapolation for SCF convergence

Parameter	Description
NODIIS	No Fock matrix extrapolation for SCF convergence
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
D2	Semi-empirical dispersion correction with D2
MOREAD	Use density from previous step
NOMOREAD	Recalculate density from scratch each step
OPTMAX	Optimise to a local maximum, rather than a minimum
TRAJ	Write trajectory to tuna-trajectory.xyz
NOTRAJ	Do not write trajectory
MAXGEOMITER [int]	Maximum number of optimisation steps
SCANSTEP [float]	Distance increment for coordinate scan
SCANNUMBER [int]	Number of distance increments for coordinate scan
SCANPLOT	Plot potential energy surface from coordinate scan
DENSPLOT	Plot 3D electron density
TIMESTEP [float]	Timestep for MD simulation
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

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 Reconmmended 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, 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, *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)   S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart and T. Yu, *PeerJ*, 2014, **2**, e453.

(11)   P. T. Inc., *Collaborative data science*, Montreal, QC.

(12)   W. J. Hehre, R. F. Stewart and J. A. Pople, *J. Chem. Phys.*, 1969, **51**, 2657–2664.

(13)   J. S. Binkley, J. A. Pople and W. J. Hehre, *J. Am. Chem. Soc.*, 1980, **102**, 939–947.

(14) R. Ditchfield, W. J. Hehre and J. A. Pople, *J. Chem. Phys.*, 1971, **54**, 724–728.

(15) T. Clark, J. Chandrasekhar, G. W. Spitznagel and P. V. R. Schleyer, *J. Comput. Chem.*, 1983, **4**, 294–301.

(16) R. Krishnan, J. S. Binkley, R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1980, **72**, 650–654.

(17) T. Koopmans, *Physica*, 1934, **1**, 104–113.

(18) R. S. Mulliken, *J. Chem. Phys.*, 1955, **23**, 1833–1840.

(19) P.-O. Löwdin, *J. Chem. Phys.*, 1950, **18**, 365–375.

(20) I. Mayer, *Chem. Phys. Lett.*, 1983, **97**, 270–274.

(21) E. R. Davidson and A. E. Clark, *Int. J. Quantum Chem.*, 2022, **122**, e26860.

(22) S. Grimme, *J. Comput. Chem.*, 2006, **27**, 1787–1799.

(23) R. Iftimie, P. Minary and M. E. Tuckerman, *Proc. Natl Acad. Sci.*, 2005, **102**, 6654–6659.

(24) R. Car and M. Parrinello, *Phys. Rev. Lett.*, 1985, **55**, 2471–2474.

(25) W. C. Swope, H. C. Andersen, P. H. Berens and K. R. Wilson, *J. Chem. Phys.*, 1982, **76**, 637–649.

(26) C. C. J. Roothaan, *Rev. Mod. Phys.*, 1951, **23**, 69–89.

(27) G. G. Hall, *Proc. Roy. Soc. London A*, 1951, **205**, 541–552.

(28) D. R. Hartree and W. Hartree, *Proc. Roy. Soc. London A*, 1935, **150**, 9–33.

(29) P. Pulay, *Chem. Phys. Lett.*, 1980, **73**, 393–398.

(30) C. Møller and M. S. Plesset, *Phys. Rev.*, 1934, **46**, 618–622.

(31) S. Grimme, *J. Chem. Phys.*, 2003, **118**, 9095–9102.

(32) S. Grimme, *J. Comput. Chem.*, 2003, **24**, 1529–1537.