# TUNA

*Theoretical Unification of Nuclear Arrangements*

Version 0.5.0

August 2024

Harry Brough

# Foreword

Welcome to TUNA!

Beginning as a project in my spare time, developing TUNA was intended to deepen my understanding of the theory and algorithmic implementation of quantum chemistry. It's a simple, command line-based Python program designed to calculate properties of diatomic molecules. While modest in scope, TUNA aims to be a helpful tool at the intersection of molecular physics and quantum chemistry, providing an accessible way to explore quantum mechanical calculations in small molecules.

One of TUNA's strengths is its user-friendly interface, which makes it a suitable teaching aid. With clear command line outputs, the program offers an straightforward experience for those looking to learn about quantum chemistry. On the more technical side, TUNA most unique feature is combining the calculation of potential energy surfaces with the numerical solution of the nuclear Schrödinger equation, enabling users to compare anharmonic vibrational frequencies of diatomics using a variety of (post-)Hartree-–Fock methods. The program is also surprisingly efficient, considering it's written in Python.

While TUNA is modest in scope, its capability will improve in the near future. The primary long term aim is to enable it to act as a testing ground for the performance of density-functional approximations on diatomics.

One day, somebody might find the program useful and TUNA could make waves!

Harry Brough

# Contents

# 1   Using TUNA

## 1.1   Installation

Installing TUNA has only been tested on Windows 11 — and not tested very well. The easiest way to install TUNA is first ensuring Python 3.12 or higher is installed, then from a terminal, run:

```
pip install QuantumTUNA
```

If this proceeds according to plan, next locate the folder where TUNA was installed, and add it to PATH. 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, you're good to go!

## 1.2   Simple Input Line

The input line reads "TUNA calculation-type: atom1 [atom2 bond-length] ! method basis-set ! parameters". The input line is not case sensitive.

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

For instance, a single point energy calculation on an H2 molecule with bond length of 1.0 Angstroms, with Hartree–Fock in the STO-3G basis set is:

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

Additional parameters can be added by adding another exclamation mark at the end, and appending the parameters.

An optimisation calculation with the D2 dispersion scheme on a H—He$^+$ molecule starting at a bond length of 0.8 Å , using the 6-31++G basis and spin-component-scaled MP2, is:

```
TUNA OPT: H He 0.8 ! SCS-MP2 6-31++G ! D2 CHARGE +1 DENSPLOT
```

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

# 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)
- Anharmonic frequency (ANHARM)
- Coordinate scan (SCAN)
- *Ab initio* molecular dynamics (MD)

## 2.2 Methods

- Restricted Hartree–Fock
- Unrestricted Hartree–Fock
- MP2
- Spin-component-scaled MP2
- D2 dispersion correction

## 2.3 Properties

- Nuclear and electronic dipole moment
- Mulliken and Löwdin charges and bond orders
- Koopman's theorem electron affinity and ionisation energy, HOMO–LUMO gap
- Electron density calculation and 3D plot
- Molecular orbitals, energies
- Rotational constants
- Thermochemistry

- MP2 unrelaxed density and natural orbitals

## 2.4  SCF Convergence Options

- DIIS
- Dynamic and static damping
- Slow convergence option
- Very slow convergence option
- Level shift

## 2.5  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

## 2.6  Miscellaneous

- Plot for coordinate scan
- Ghost atoms
- Temperature and pressure keywords for thermochemistry
- SCF convergence keywords
- Optimisation convergence keywords
- Identification of point group
- Approximate or exact Hessian for optimisations
- Maximum iterations keywords

- More accurate energies than ORCA

# 3    Detailed Documentation

## 3.1    Calculation Types

### 3.1.1    Single Point Energy

### 3.1.2    Geometry Optimisation

### 3.1.3    Harmonic Frequency

### 3.1.4    Anharmonic Frequency

### 3.1.5    Coordinate Scan

### 3.1.6    *Ab Initio* Molecular Dynamics

The implementation in TUNA of *ab initio* molecular dynamics (AIMD) is Born–Oppenheimer molecular dynamics, where forces are calculated with quantum chemical methods before atomic positions are updated by the classical equations of motion, $\mathbf{F}_i = m_i \mathbf{a}_i$ [1].

Because the Velocity Verlet algorithm [2] yields velocities and positions at the same point in time, as well as maintaining good numerical stability, it is implemented in TUNA, where positions are updated by

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2, \tag{3.1}$$

and velocities are then updated by

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t + \Delta t)}{2}\Delta t . \tag{3.2}$$

The initial velocities are determined from the temperature (set by the TEMP keyword), before total system translations are removed and the velocities rescaled to match the input temperature. Presently in TUNA, no thermostat is implemented so all calculations run in the NVE ensemble. As positions and velocities are continually updated, a trajectory is mapped out. By default, TUNA prints this trajectory to a file called "[Atom A]-[Atom-B]-trj.xyz" in the directory where the program was called.

To run an AIMD calculation in TUNA, use the $\mathrm{MD}$ calculation type with any electronic structure method and basis, and set the timestep, initial temperature, and maximum number of steps with the keywords TIMESTEP, TEMP, NSTEPS respectively.

## 3.2  Electronic Structure Methods

### 3.2.1  Hartree–Fock

#### 3.2.1.1  SCF Convergence

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 smoothly converges.

Level shifting is applied via a parameter, $b$, which updates the Fock matrix by

$$\mathbf{F}' = \mathbf{F} - b\mathbf{P} \tag{3.3}$$

In TUNA, $b = 0.2$ by default. Level shift can be activated by the LEVELSHIFT keyword.

### 3.2.2  Møller–Plesset Perturbation Theory

### 3.2.3  Basis Sets

Presently, only s orbital integrals are implemented in TUNA, so basis sets are limited to period 1 elements without polarisation functions. Only Pople-style basis sets are available, and custom sets can not yet be read in.

Note: the 6-311++G basis set is not parameterised for helium.

### 3.2.4  Dispersion Correction

Currently, the only available form of dispersion correction is the D2 semi-empirical scheme by Grimme [3]. Here, the energy due to dispersion is calculated by

$$E_{\mathsf{disp}} = -s_6 \frac{C_6^{ij}}{R_{ij}^6} f_{\mathsf{damp}} \; , \tag{3.4}$$

where the damping function is given by

$$f_{\mathsf{damp}} = \frac{1}{1 + \exp(-d(\frac{R_{ij}}{R_r} - 1))} \; . \tag{3.5}$$

Table 3.1: Basis sets in TUNA

| Basis Set | Description |
| --- | --- |
| STO-3G | Minimal basis set |
| STO-6G | Minimal basis set |
| 3-21G | Double-zeta basis set |
| 4-31G | Double-zeta basis set |
| 6-31G | Double-zeta basis set |
| 6-31+G | Double-zeta basis set with diffuse functions on heavy atoms |
| 6-31++G | Double-zeta basis set with diffuse functions on all atoms |
| 6-311G | Triple-zeta basis set |
| 6-311+G | Triple-zeta basis set with diffuse functions on heavy atoms |
| 6-311++G | Triple-zeta basis set with diffuse functions on all atoms |

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. The values of the Van der Waals radii, $R_r$, and $C_6$ coefficients are element-specific.

## 3.3 Properties

### 3.3.1 Dipole Moments, Bond Orders and Charges

### 3.3.2 Electron Density Plot

### 3.3.3 Thermochemistry

# 4    List of Parameters

| Parameter | Meaning |
|---|---|
| P | Additional print |
| CHARGE [int] or CH [int] | Charge |
| D2 | D2 dispersion correction |
| DENSPLOT | Electron density plot |
| | |
| NODIIS | Turn off DIIS in SCF convergence |
| NODAMP | Turn off damping in SCF convergence |
| LEVELSHIFT | Turn on level shift in SCF convergence |
| SLOWCONV | Permanent high static damping |
| VERYSLOWCONV | Permanent very high static damping |
| LOOSE or LOOSESCF | Loose SCF convergence criteria |
| MEDIUM or MEDIUMCF | Medium SCF convergence criteria |
| TIGHT or TIGHTSCF | Tight SCF convergence criteria |
| EXTREME or EXTREMESCF | Extreme SCF convergence criteria |
| MAXITER [int] | Maximum SCF cycle iterations |
| | |
| CALCHESS | Calculate exact Hessian for optimisation |
| LOOSEOPT | Loose optimisation convergence criteria |
| MEDIUMOPT | Medium optimisation convergence criteria |
| TIGHTOPT | Tight optimisation convergence criteria |
| EXTREMEOPT | Extreme optimisation convergence criteria |
| GEOMMAXITER [int] or MAXGEOMITER [int] | Maximum optimisation iterations |
| | |
| SCANPLOT | Plot of coordinate scan potential energy surface |
| SCANSTEP [float] | Step size for coordinate scan |
| SCANNUMBER [int] | Number of steps for coordinate scan |
| MOREAD | Read density from previous scan step |
| | |
| TEMP [float] | Temperature for thermochemistry |
| PRES [float] | Pressure for thermochemistry |

# Bibliography

(1)  R. Iftimie, P. Minary and M. E. Tuckerman, *Proceedings of the National Academy of Sciences*, 2005, **102**, 6654–6659.

(2)  W. C. Swope, H. C. Andersen, P. H. Berens and K. R. Wilson, *The Journal of Chemical Physics*, 1982, **76**, 637–649.

(3)  S. Grimme, *Journal of Computational Chemistry*, 2006, **27**, 1787–1799.