

TUNA

THEORETICAL UNIFICATION OF NUCLEAR ARRANGEMENTS

Version 0.10.0

March 2026

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 a range of quantum chemical methods.

Unlike other Python-based programs for quantum chemistry such as Psi4 [1, 2] and PySCF [3], TUNA is not primarily a collection of modules to incorporate into a workflow, but is intended to be a complete program to be run directly from the command line. Parts of the syntax are inspired by the ORCA quantum chemistry package [4, 5] (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.

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

On the technical side, TUNA utilises numerical energy derivatives, so any implemented electronic structure method can be used to calculate geometries, harmonic 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 [6], 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 being 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 as well as a user-friendly tool for quantum chemistry education.

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

Harry Brough

Contents

1	Getting Started	5
1.1	Installation	5
1.2	Simple Input Line	5
1.3	Units and Constants	6
1.4	Program Components	7
1.5	Philosophy of TUNA	8
2	Summary of Features in TUNA	9
2.1	Calculation Types	9
2.2	Electronic Structure Methods	9
2.3	Basis Sets	10
2.4	Properties	12
2.5	Miscellaneous	13
3	Simple Input Line Examples	14
4	Detailed Documentation	16
4.1	Calculation Types	16
4.1.1	Single Point Energy	17
4.1.1.1	Koopmans' Theorem	19
4.1.1.2	Rotational Constant	19
4.1.1.3	Dipole Moment, Polarisability and Hyperpolarisability	20
4.1.1.4	Population Analysis	22
4.1.1.5	Natural Orbitals	24
4.1.1.6	Plotting Orbitals and Densities	24
4.1.2	Coordinate Scan	27
4.1.2.1	Plotting Coordinate Scans	29
4.1.3	Geometry Optimisation	31
4.1.3.1	Force Calculations	33
4.1.3.2	Dispersion Correction	34

4.1.4	Harmonic Frequency	34
4.1.4.1	Transition Intensity	37
4.1.4.2	Thermochemistry	39
4.1.5	Anharmonic Frequency	40
4.1.5.1	Transition Intensity	41
4.1.5.2	Plotting Nuclear Wavefunctions	41
4.1.6	Ionisation Potential and Electron Affinity	43
4.1.7	<i>Ab Initio</i> Molecular Dynamics	44
4.2	Electronic Structure Methods	46
4.2.1	Hartree–Fock Theory	48
4.2.1.1	Initial Guess	49
4.2.1.2	SCF Convergence	51
4.2.1.3	Hartree Theory	53
4.2.2	Møller–Plesset Perturbation Theory	54
4.2.2.1	Spin-component Scaling	56
4.2.2.2	Orbital-optimised MP2	57
4.2.2.3	Iterative MP2	57
4.2.2.4	Laplace Transform MP2	58
4.2.2.5	Freezing Core Electrons	59
4.2.3	Coupled Cluster Theory	60
4.2.3.1	Coupled Cluster Doubles	62
4.2.3.2	Coupled Cluster Singles and Doubles	62
4.2.3.3	Coupled Cluster Singles, Doubles and Triples	63
4.2.3.4	Coupled Cluster Singles, Doubles, Triples and Quadruples	63
4.2.3.5	Coupled Cluster with Perturbative Triples	64
4.2.3.6	Coupled Cluster with Perturbative Quadruples	64
4.2.3.7	Linearised Coupled Cluster	64
4.2.3.8	Quadratic Configuration Interaction	65
4.2.3.9	Coupled Cluster Density Matrix	65
4.2.4	Density Functional Theory	65
4.2.4.1	Local Density Approximation	67
4.2.4.2	Generalised Gradient Approximation	68
4.2.4.3	Hybrid Functionals	70
4.2.4.4	Double-hybrid Functionals	71
4.2.4.5	Numerical Quadrature	73
4.2.5	Excited States	74
4.2.5.1	Configuration Interaction Singles	74
4.2.5.2	Perturbative Doubles Correction to CIS	76

4.3	Basis Sets and Molecular Integrals	76
4.3.1	Custom Basis Sets	82
4.3.2	Basis Set Extrapolation	84
5	Performance of TUNA	86
6	Acknowledgements	88
7	List of Keywords	89
8	Detailed Changelog	94
8.1	TUNA 0.1.0	94
8.2	TUNA 0.2.0	94
8.3	TUNA 0.3.0	95
8.4	TUNA 0.4.0	95
8.5	TUNA 0.5.0	95
8.5.1	TUNA 0.5.1	97
8.6	TUNA 0.6.0	98
8.6.1	TUNA 0.6.1	99
8.7	TUNA 0.7.0	100
8.8	TUNA 0.8.0	102
8.8.1	TUNA 0.8.1	104
8.9	TUNA 0.9.0	105
8.10	TUNA 0.10.0	106
8.11	Next Steps	108
	Bibliography	113

1 Getting Started

1.1 Installation

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

```
pip install QuantumTUNA
```

It is also possible to install TUNA by cloning the GitHub repository, but this is not recommended as you will have to compile the Cython integrals module yourself. Installing with pip will also install TUNA's dependencies — NumPy [6], SciPy [7], Matplotlib [8] and TermColor [9].

Next locate the folder where TUNA was installed within the Python site packages, */TUNA/, where * is the path, and add it to the PATH environment variable on Windows.

On MacOS or Linux run the following commands to allow TUNA to be used from anywhere:

```
echo "alias tuna='noglob python3 */TUNA/tuna.py'" >> ~/.zshrc  
echo "alias TUNA='noglob python3 */TUNA/tuna.py'" >> ~/.zshrc  
source ~/.zshrc
```

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!

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 are listed in section 4.1, electronic structure methods in section 4.2 and basis sets in section 4.3.

For example, a single point energy calculation on an H₂ molecule with bond length of 1.0 Å, with restricted Hartree–Fock theory in the STO-3G basis set is requested by:

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

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

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

As many keywords as you like can be written after the 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 added. Many keywords can be used together in this way, and TUNA will figure out what you want. Further input examples to test out TUNA are found in section 3.

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. Units are set by the CODATA 2022 recommendations [10], in Table 1.1. All other constants are derived from these and the atomic masses in Table 4.5.

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 are also imported. All these should automatically be installed when `pip install QuantumTUNA` is run. The most important are NumPy [6] and SciPy [7], which speed up the code enormously compared to pure Python. In addition, TermColor [9] makes the terminal output more vibrant, while Matplotlib [8] enables the creation of two-dimensional plots.

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

Module	Description
tuna	Main program, input parsing
tuna_basis	Storing and setting up basis sets
tuna_cc	Coupled cluster
tuna_ci	Configuration interaction
tuna_dft	Density functional theory
tuna_energy	Calculating energies, building molecules, coordinate scans
tuna_freq	Harmonic and anharmonic frequencies
tuna_guess	Initial guess
tuna_integral	Evaluating one- and two-electron integrals
tuna_kernel	Low level functions for energy evaluation
tuna_md	<i>Ab initio</i> molecular dynamics
tuna_molecule	Storing and parsing molecular information
tuna_mp	Møller–Plesset perturbation theory
tuna_opt	Calculating gradients, Hessians and optimisation algorithm
tuna_out	Graphical plots and trajectories
tuna_postscf	Calculating and printing molecular properties
tuna_scf	Main self-consistent field loop, convergence acceleration
tuna_thermo	Thermochemistry after frequency calculations
tuna_util	General utility, units, frequently used functions

1.5 Philosophy of TUNA

Numerical differentiation underpins property calculations in TUNA. For atoms and diatomics it is very efficient, and it is also completely general: *any* electronic structure method can be combined with *any* calculation type. This makes unusual calculations straightforward and consistent. For example, a CCSDTQ hyperpolarisability or molecular dynamics calculation can, in principle, be treated with the same precision as a Hartree–Fock calculation. It also means that once a new method is implemented, it automatically becomes available across the program. As more compatible features are added, the capabilities of TUNA grow rapidly.

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

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} . \quad (1.1)$$

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

$$f''(x) \approx \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2} . \quad (1.2)$$

For numerical third derivatives, used in hyperpolarisability calculations, a six point stencil formula is used,

$$f'''(x) \approx \frac{f(x-3h) - 8f(x-2h) - 26f'(x) + 8f(x+2h) - f(x+3h)}{8h^3} . \quad (1.3)$$

The main drawback of numerical derivatives is their sensitivity to the finite-difference step size, h . A fixed value of 0.001 is used for all derivatives in TUNA, as this seems to give the most stable results.

These formulae also require well converged energies. For calculations involving first derivatives, TIGHT self-consistent field convergence criteria are used by default, while higher derivatives use EXTREME convergence criteria. Even the standard thresholds in TUNA are tighter than in many other programs, although they can still be overridden manually (see section 4.2.1.2).

A major advantage of TUNA's numerical philosophy is that these formulae can be combined with basis set extrapolation, allowing approximate complete basis set geometries, frequencies, properties and molecular dynamics trajectories to be calculated. See section 4.3.2 for details.

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
- Force, FORCE
- Coordinate scan, SCAN
- *Ab initio* molecular dynamics, MD
- Anharmonic frequency, ANHARM
- Ionisation potential, IP
- Electron affinity, EA

2.2 Electronic Structure Methods

- Hartree–Fock, HF
- Unrestricted Hartree–Fock, UHF
- Hartree theory, H
- MP2, MP2
- Spin-component-scaled MP2, SCS-MP2
- MP3, MP3
- Spin-component-scaled MP3, SCS-MP3
- MP4, MP4
- MP4 without triples, MP4[SDQ]
- MP4 without singles or triples, MP4[DQ]
- Orbital-optimised MP2, OMP2
- Iterative MP2, IMP2

- Laplace transform MP2, LMP2
- 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
- Coupled cluster singles, doubles, triples and perturbative quadruples, CCSDT[Q]
- Coupled cluster singles, doubles, triples and quadruples, CCSDTQ
- Quadratic configuration interaction singles and doubles, QCISD
- Quadratic configuration interaction singles, doubles and perturbative triples, QCISD[T]
- Local spin-density approximation density functional theory, LDA, SVWN, SPW, HFS
- Generalised gradient approximation density functional theory, PBE, BLYP, BP86, HFB
- *meta*-Generalised gradient approximation density functional theory, TPSS
- Hybrid density functional theory, PBE0, B1LYP, B3LYP, BHLYP, B1P86
- Double-hybrid density functional theory, PBE0-DH, PBE0-2, PBE-QIDH, B2PLYP, B2K-PLYP

2.3 Basis Sets

- | | |
|----------|--------------------------|
| ▪ STO-2G | ▪ 6-31++G |
| ▪ STO-3G | ▪ 6-311G |
| ▪ STO-4G | ▪ 6-311+G |
| ▪ STO-5G | ▪ 6-311++G |
| ▪ STO-6G | ▪ 6-31G* / 6-31G(d) |
| ▪ 3-21G | ▪ 6-31G** / 6-31G(d,p) |
| ▪ 4-31G | ▪ 6-311G* / 6-311G(d) |
| ▪ 6-31G | ▪ 6-311G** / 6-311G(d,p) |
| ▪ 6-31+G | ▪ 6-31+G* / 6-31+G(d) |

- 6-311+G* / 6-311+G(d)
- 6-31+G** / 6-31+G(d,p)
- 6-311+G** / 6-311+G(d,p)
- 6-31++G* / 6-31++G(d)
- 6-311++G* / 6-311++G(d)
- 6-31++G** / 6-31++G(d,p)
- 6-311++G** / 6-311++G(d,p)
- 6-31G(2df,p)
- 6-31G(3df,3pd)
- 6-311G(2df,2pd)
- 6-311+G(2d,p)
- 6-311++G(2d,2p)
- 6-311++G(3df,3pd)
- cc-pVDZ
- cc-pVTZ
- cc-pVQZ
- cc-pV5Z
- cc-pV6Z
- aug-cc-pVDZ
- aug-cc-pVTZ
- aug-cc-pVQZ
- aug-cc-pV5Z
- aug-cc-pV6Z
- d-aug-cc-pVDZ
- d-aug-cc-pVTZ
- d-aug-cc-pVQZ
- d-aug-cc-pV5Z
- d-aug-cc-pV6Z
- t-aug-cc-pVDZ
- t-aug-cc-pVTZ
- t-aug-cc-pVQZ
- t-aug-cc-pV5Z
- t-aug-cc-pV6Z
- cc-pCVDZ
- cc-pCVTZ
- cc-pCVQZ
- cc-pCV5Z
- aug-cc-pCVDZ
- aug-cc-pCVTZ
- aug-cc-pCVQZ
- aug-cc-pCV5Z
- cc-pwCVDZ
- cc-pwCVTZ
- cc-pwCVQZ
- cc-pwCV5Z
- aug-cc-pwCVDZ
- aug-cc-pwCVTZ
- aug-cc-pwCVQZ
- aug-cc-pwCV5Z
- ano-pVDZ
- ano-pVTZ
- ano-pVQZ
- ano-pV5Z
- aug-ano-pVDZ
- aug-ano-pVTZ
- aug-ano-pVQZ
- aug-ano-pV5Z
- pc-0
- pc-1
- pc-2

- pc-3
- pc-4
- aug-pc-0
- aug-pc-1
- aug-pc-2
- aug-pc-3
- aug-pc-4
- pcseg-0
- pcseg-1
- pcseg-2
- pcseg-3
- pcseg-4
- aug-pcseg-0
- aug-pcseg-1
- aug-pcseg-2
- aug-pcseg-3
- aug-pcseg-4
- def2-SVP
- def2-SVPD
- def2-TZVP
- def2-TZVPD
- def2-TZVPP
- def2-TZVPPD
- def2-QZVP
- def2-QZVPD
- def2-QZVPP
- def2-QZVPPD

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
- Calculations in finite electric fields
- Polarisability
- Hyperpolarisability
- Plotting molecular orbitals and natural orbitals
- Plotting electron densities, spin densities and difference densities

- Plotting anharmonic nuclear wavefunctions
- Anharmonic fundamental transition intensities

2.5 Miscellaneous

- McMurchie-Davidson integrals of Gaussian basis functions
- Interface to Matplotlib for plots for coordinate scan
- Ghost atoms
- Optimise to a minimum or maximum
- Decontract basis sets
- Rotate initial guess orbitals
- Self-consistent, superposition of atomic densities and core guess strategies
- 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)
- Coupled cluster amplitude extrapolation (DIIS)
- Dynamic (Zerner–Hehenberger) and static damping
- Freeze core orbitals
- D2 dispersion correction
- Coupled cluster T1 diagnostic
- Basis set extrapolation

3 Simple Input Line Examples

In this section, a number of examples of simple input lines are given. These can be simply copied and pasted into a terminal, to see TUNA in action. This should give a clear idea of the syntax to run a calculation, and then you can experiment!

Firstly, a single point energy calculation on a hydrogen atom:

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

Next, an MP2 geometry optimisation of lithium hydride, with D2 dispersion correction:

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

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

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

An MP4 frequency calculation on carbon monoxide, with 50% damping and no DIIS:

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

An optimisation and subsequent frequency calculation on dihydrogen, using coupled cluster theory and extreme SCF and geometry convergence criteria:

```
TUNA OPTFREQ : H H 0.74 : CCSD def2-TZVPP : EXTREME EXTREMEOPT
```

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

```
TUNA SPE : Li : CCSDT cc-pVDZ : CCMAXITER 60
```

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

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

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

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

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

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

An optimisation on triplet oxygen in a finite electric field:

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

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

TUNA SPE : C H 1.0 : CIS cc-pVDZ : DIFFDENS PLOT ROOT 3

A coupled cluster optimisation with single, double, triple and perturbative quadruples on carbon monoxide, freezing the core orbitals.

TUNA OPT : C O 1 : CCSDT[Q] STO-3G : FREEZECORE

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

TUNA ANHARM : H H 0.74 : CCSD def2-SVP : PLOTVIB

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

TUNA SPE : Mg : MP2 def2-TZVPP : EXTRAPOLATE

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

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

Finally, a full configuration interaction (via CCSDTQ) optimisation on the lithium hydride molecule, with extreme self-consistent field and optimisation convergence.

TUNA OPTFREQ : Li H 1.67 : CCSDTQ 6-31G : EXTREME EXTREMOPT

4 Detailed Documentation

4.1 Calculation Types

This section dives into the different calculation types available in TUNA, listed in Table 4.1 with their respective keywords.

Table 4.1 Calculation types available in TUNA

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

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

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

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

Trying to run a calculation for a molecule (eg. OPT, FREQ, MD) 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
```

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

At the end of any calculation — if it terminates successfully — TUNA prints the total time taken for the calculation. This time is counted from when the Python modules have been imported. More information about the times for each part of a TUNA calculation (integrals, SCF iterations, correlation, etc.) is shown when the additional print keyword, P, is used.

4.1.1 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 requested by:

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

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

```
~~~~~  
Molecule and Basis Information
```

```

~~~~~
Molecular structure: H ----- He

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

Charge: 1
Multiplicity: 1
Number of electrons: 2
Number of alpha electrons: 1
Number of beta electrons: 1
Number of occupied orbitals: 1
Number of virtual orbitals: 5

Point group: C1
Bond length: 2.0000
~~~~~

```

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

$$V_{nn} = \frac{Z_A Z_B}{R}, \quad (4.1)$$

where Z is the relative nuclear charge and R is the bond length. Next the optional dispersion energy is calculated (see section 4.1.3.2), before the one- and two-electron integrals are evaluated — for a discussion of the implementation of molecular integrals in TUNA, see section 4.3. The eigenvalues of the overlap matrix, \mathbf{S} , are then checked to make sure they're not too small and avoid errors from a linearly dependent basis set. The default threshold is 10^{-7} and can be adjusted with the `STHRESH` keyword. The Fock transformation matrix, $\mathbf{X} = \mathbf{S}^{-\frac{1}{2}}$, is formed, then the guess density matrix is calculated — this is the self-consistent field minimal basis guess via superposition of atomic densities by default. See section 4.2.1.1 for details about the initial guess.

If a multi-electron calculation is requested, the convergence accelerators are printed and the Hartree–Fock self-consistent field cycle begins, discussed in section 4.2.1.2. If this converges, the components of the energy are printed in addition to the Virial ratio, and the spin contamination for spin-unrestricted calculations. This is the ratio between the molecular potential and kinetic energy, and should approach 2.0 as the molecular structure approaches an equilibrium geometry or for an atom. If a correlated or excited state calculation has been requested, this then begins.

When the final energy has been calculated, the one electron reduced density matrix, \mathbf{P} is used in property calculations. In the current version, the RHF, UHF, unrelaxed CIS, unrelaxed (SCS-)MP2 and linearised coupled cluster density matrices are available in TUNA. The orbital-optimised MP2 (OMP2) density matrix is also implemented, for which there is no difference between an unrelaxed and a relaxed density — see section 4.2.2.2. If a higher order perturbation theory calculation is requested, properties will be calculated at the MP2 level. The properties calculated are detailed in this section. Calculations of rotational constants, multipole moments and population analysis are disabled for single atoms.

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

4.1.1.1 Koopmans' Theorem

Koopmans' theorem states that in closed-shell Hartree–Fock theory, the ionisation potential is equal to the negative of the orbital energy of the highest occupied molecular orbital (HOMO), $-\epsilon_{\text{HOMO}}$, and that the electron affinity is equal to the negative of the orbital energy of the lowest unoccupied molecular orbital (LUMO), $-\epsilon_{\text{LUMO}}$ [11]. This is exact within RHF theory if the ionic orbitals are assumed to be identical to the orbitals of the atom.

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

The “exact” ionisation potential or electron affinity also accounts for orbital relaxation, and can be calculated from two converged self-consistent field calculations (for vertical ionisations) or from two converged geometry optimisations (for adiabatic ionisations). These calculations can be performed easily using the IP or EA calculation types, which are discussed in detail in section 4.1.6.

4.1.1.2 Rotational Constant

The next property printed is the rotational constant, calculated by

$$B = \frac{1}{4\pi\mu R^2}, \quad (4.2)$$

where μ is the reduced mass of the molecule, using the masses in section 4.1.4. This is printed in units of GHz and cm^{-1} .

4.1.1.3 Dipole Moment, Polarisability and Hyperpolarisability

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

$$E(\mathbf{F}) = E - \sum_i \mu_i F_i - \frac{1}{2} \sum_{ij} \alpha_{ij} F_i F_j - \frac{1}{6} \sum_{ijk} \beta_{ijk} F_i F_j F_k + \dots, \quad (4.3)$$

where the dipole moment can be identified with the first derivative, polarisability with the second and hyperpolarisability with the third.

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, R_0 , calculated with the masses in section 4.1.4. The nuclear dipole moment,

$$\mu_{\text{nuc}} = Z_A(R_A - R_0) + Z_B(R_B - R_0), \quad (4.4)$$

and the electronic dipole moment,

$$\mu_{\text{ele}} = - \sum_{\mu\nu} P_{\mu\nu} D_{\mu\nu}, \quad (4.5)$$

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. Note that these dipole moments are one-dimensional by the symmetry of a diatomic molecule.

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

```

~~~~~
                                Dipole Moment
~~~~~
Nuclear: -1.49879292      Electronic: -1.47745744

Total: -2.97625036      H ----- He <-----+
~~~~~

```

This can be reproduced with the following input line:

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

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

$$\mu_{\text{ele}} = \frac{\partial E}{\partial F} . \quad (4.6)$$

The dipole moment can be calculated with the DIPOLE keyword. For methods with an one-electron density matrix implemented, this will give the same result as the version automatically printed after the SCF calculation. However this dipole moment will be correct for all methods, where the analytical one may not be — for example the analytical MP2 dipole moment uses the unrelaxed density matrix, whereas the numerical dipole moment will be correct:

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

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

$$\alpha_{ij} = \frac{\partial^2 E}{\partial F_i \partial F_j} . \quad (4.7)$$

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

$$\alpha_{\text{iso}} = \frac{1}{3}(\alpha_{\perp} + \alpha_{\perp} + \alpha_{\parallel}) \quad (4.8)$$

and the anisotropic polarisability is

$$\alpha_{\text{aniso}} = \alpha_{\parallel} - \alpha_{\perp} . \quad (4.9)$$

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

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

The dipole–dipole–dipole hyperpolarisability can also be calculated in TUNA with the HYPER keyword, which in general has 27 components since

$$\beta_{ijk} = \frac{\partial^3 E}{\partial F_i \partial F_j \partial F_k} . \quad (4.10)$$

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

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

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

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

$$\mathbf{F} = 0.05\hat{\mathbf{x}} + 0.06\hat{\mathbf{y}} + 0.10\hat{\mathbf{z}} \quad (4.11)$$

is requested by:

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

4.1.1.4 Population Analysis

Towards the end of a calculation, TUNA prints Mulliken [12], Löwdin [13] and Mayer [14] population analysis. Note that all the equations in this section are true for diatomics, not necessarily molecules in general. These calculations can be disabled with the T keyword, or requested after every geometry iteration in an optimisation with P.

The Mulliken analysis [12] is very widely used — despite its considerable weaknesses [15] — 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

$$N = \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} . \quad (4.12)$$

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

$$N_A = \sum_{\mu \in A} \sum_{\nu \in A} P_{\mu\nu} S_{\mu\nu} + \frac{1}{2} B_{AB} , \quad (4.13)$$

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

$$B_{AB} = 2 \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu} S_{\mu\nu} . \quad (4.14)$$

The charge of an atom in a molecule is then given by $Q_A = Z_A - N_A$. In a population analysis, 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, population analysis can be performed with non-localised orthogonal basis functions in Löwdin analysis [13] where the density matrix is rotated by

$$\mathbf{P}^L = \mathbf{S}^{\frac{1}{2}} \mathbf{P} \mathbf{S}^{\frac{1}{2}} . \quad (4.15)$$

Using this density matrix, atomic populations are calculated by

$$N_A = \sum_{\mu \in A} P_{\mu\mu}^L , \quad (4.16)$$

and the bond order is calculated by

$$B_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu}^L P_{\mu\mu}^L . \quad (4.17)$$

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

The results of a Mayer analysis [14] are also printed. The Mayer charges are the same as the Mulliken charges, but the bond order (and bonded valence) is given by

$$B_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} (\mathbf{P}\mathbf{S})_{\mu\nu} (\mathbf{P}\mathbf{S})_{\nu\mu} + (\mathbf{R}\mathbf{S})_{\mu\nu} (\mathbf{R}\mathbf{S})_{\nu\mu} , \quad (4.18)$$

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

$$V_A = 2N_A - \sum_{\mu \in A} \sum_{\nu \in B} (\mathbf{PS})_{\mu\nu} (\mathbf{PS})_{\nu\mu} \quad (4.19)$$

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.

4.1.1.5 Natural Orbitals

For UHF, MP2 and coupled cluster methods, natural orbitals can be calculated. These are the eigenfunctions of the one-electron reduced density matrix

$$\mathbf{P}\mathbf{\Lambda}_i = \lambda_i \mathbf{\Lambda}_i, \quad (4.20)$$

where λ_i represents the occupancy of the i th natural orbital, a value between 0.0 and 2.0. Natural orbitals can be shown to converge most efficiently to the true density of a system [16], and the occupancies are useful for determining systems which might benefit from a multireference treatment, beyond the single determinant model of Hartree–Fock theory.

The calculation of natural orbitals is disabled by default, but can be activated with the NATORBS keyword. For instance, a CCD calculation with natural orbitals is requested by:

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

The sum of the natural orbital occupancies is also printed along with the orbital occupancies, along with the trace of the density matrix. These values should be the same — the number of electrons in the system.

4.1.1.6 Plotting Orbitals and Densities

The infrastructure that allows a DFT calculation also permits the expression of orbitals and electron densities onto a grid, which can be plotted by TUNA. While the grids for DFT integration are more dense near the nuclei and more sparse further away, the plotting grids are two-dimensional Cartesian grids — with size defined by the bond length. These two-dimensional grids are much more efficient to compute than three-dimensional “cube” files, and contain all the information necessary to describe a diatomic, due to their symmetry.

The simplest plot is the electron density, requested via the DENSLOT keyword, which shows a magenta blob of the converged density.

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

For unrestricted calculations, the SPINDENSPLIT keyword plots the spin density. For example, the following calculation on the carbyne radical shows the vast majority of the spin density is localised on the carbon atom:

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

Spin density from UHF/CC-PVDZ calculation on C—H molecule

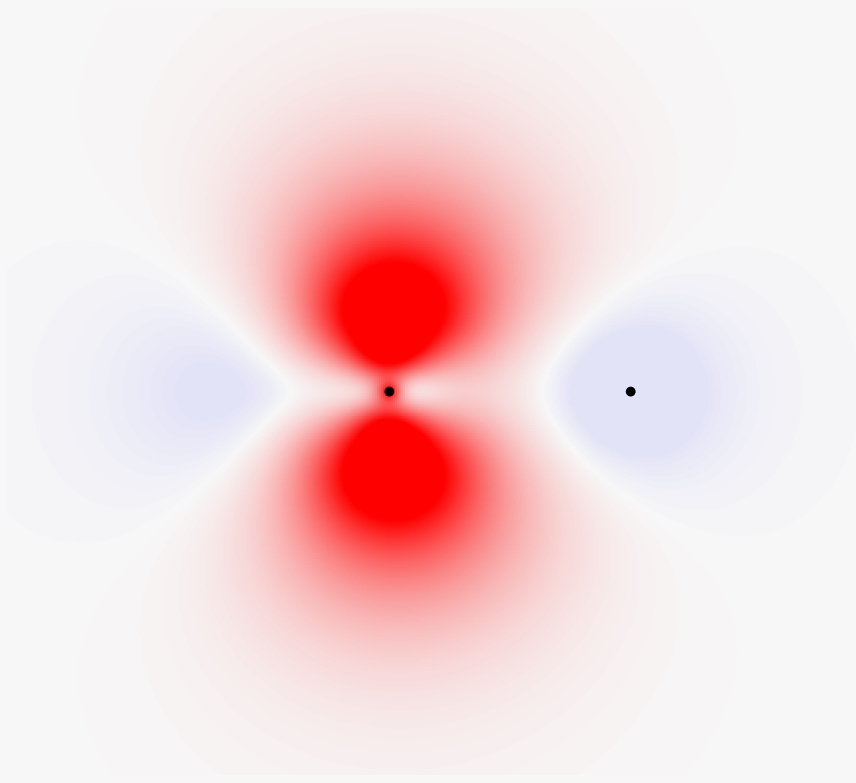


Figure 4.1 The spin density of carbyne, showing the vast majority of the spin density is found on the carbon atom, in the shape of a p orbital.

Difference densities of the i th excited state from excited state calculations (discussed in section 4.2.5, calculated by

$$P_{\text{diff}}^i = P_{\text{CIS}}^i - P_{\text{HF}}^0 \quad (4.21)$$

can be plotted with the DIFFDENSPLIT keyword. A similar equation calculates the difference spin density, given by the DIFFSPINDENSPLIT keyword. For example, the third excited state of the carbyne radical is plotted by:

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

Areas of positive difference density (which have gained electrons in the excited state) are shown in red, and areas of negative difference density are shown in blue. The converged molecular orbitals

Density from UCIS/CC-PVDZ calculation on C—H molecule

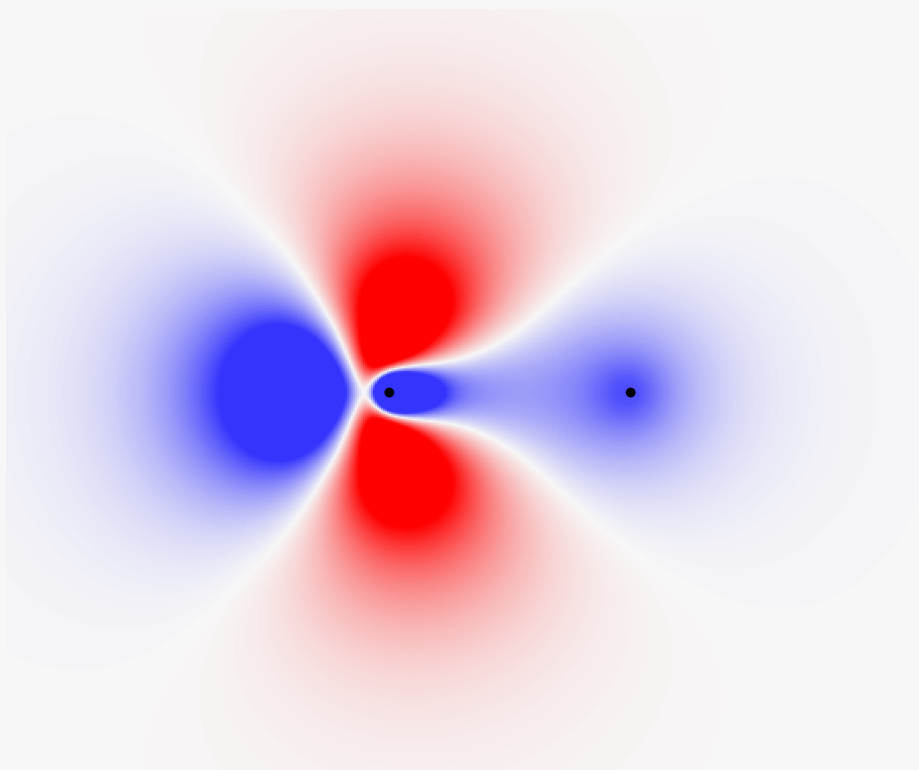


Figure 4.2 The difference density from an excited state of carbyne. This shows that the electron density has moved from around the lone pair on the carbon atom (deep blue), to the bond area.

from an SCF (Hartree–Fock or DFT) calculation can also be plotted. The PLOTMO keyword is used to do this, along with the number of molecular orbital that is desired to be plotted. The orbital numbers can be checked with the additional print keyword, P. For unrestricted references, the alpha orbitals are counted first, then the beta orbitals. The following calculation plots a 4f orbital from in the hydrogen atom:

```
TUNA SPE : H : PBE cc-pVQZ : PLOTMO 25
```

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

Orbital 25 from UPBE/CC-PVQZ calculation on H atom

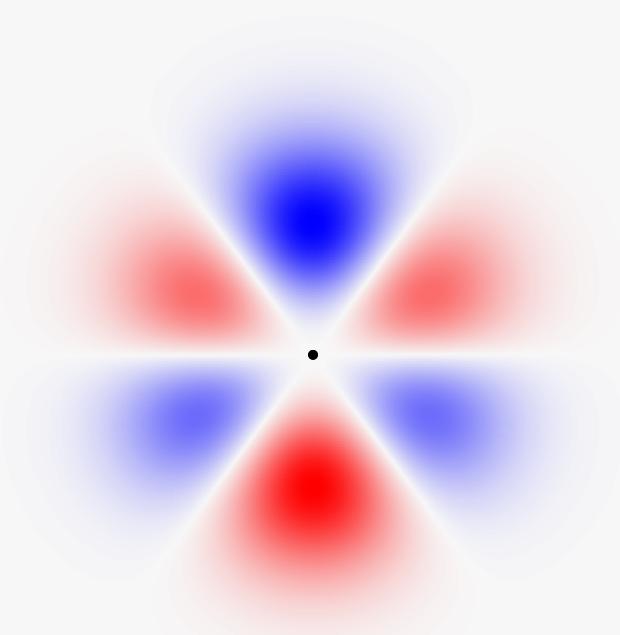


Figure 4.3 Using a large basis set allows high-lying unoccupied orbitals to be visualised. This is a virtual 4f orbital for the hydrogen atom.

```
TUNA SPE : 0 0 1.1 : B3LYP 6-311++G** : PLOTHOMO
```

If natural orbitals have been calculated (from a UHF, MP2 or coupled cluster calculation), these can be plotted similarly, using the PLOTNO keyword.

```
TUNA SPE : C 0 1.1 : CCSD cc-pVDZ : NATORBS PLOTNO 7
```

If no natural orbitals are found, the plot will just be blank.

4.1.2 Coordinate Scan

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

Orbital 8 from B3LYP/6-311++G** calculation on O—O molecule

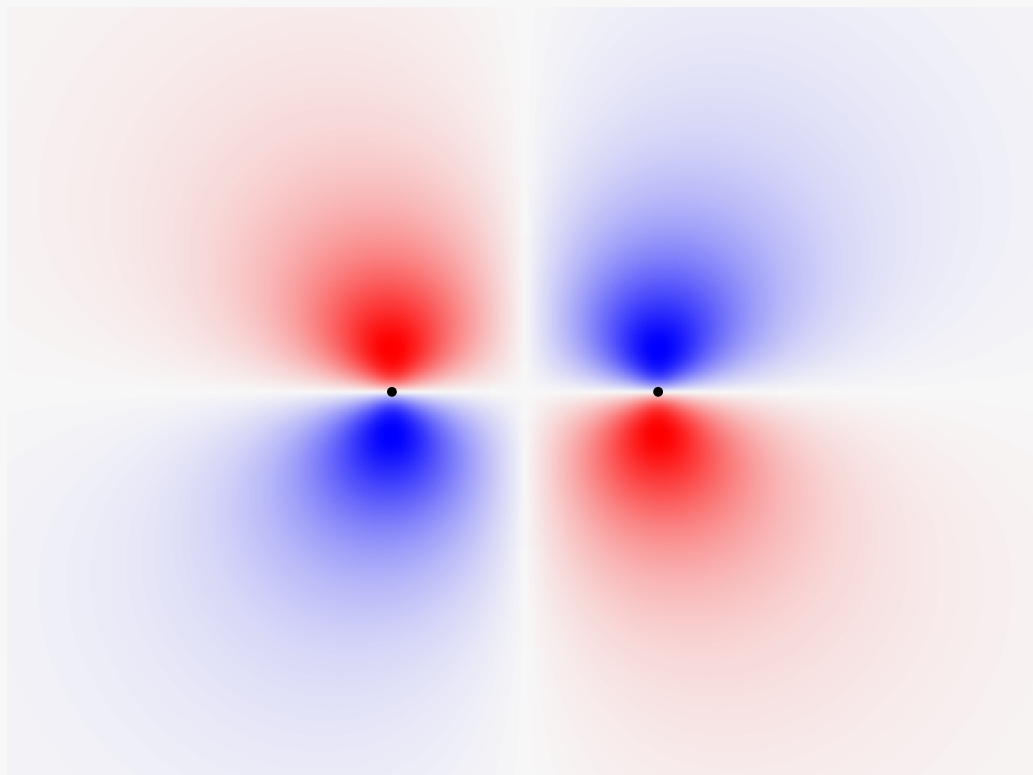


Figure 4.4 This is the HOMO of the singlet oxygen molecule, a π^* orbital split evenly over the two atoms.

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_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 : NUM 20 STEP 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 4.2.1.1 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, EXTRAPOLATE) can also be combined with the scan calculation. At the end of a coordinate scan calculation, a table of the steps, bond lengths and final energies is printed, which can be easily copied and pasted into a spreadsheet for further processing.

4.1.2.1 Plotting Coordinate Scans

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

For example, a plot with a magenta dotted line over the H_2 potential energy surface with UHF can be presented by:

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

This produces the plot in Figure 4.5, with the legend showing the electronic structure method and basis set. Plots can be saved by the keyword SAVEPLOT [filepath], for example SAVEPLOT "MAGENTAPLOT.PDF". Standard raster and vector image extensions are supported.

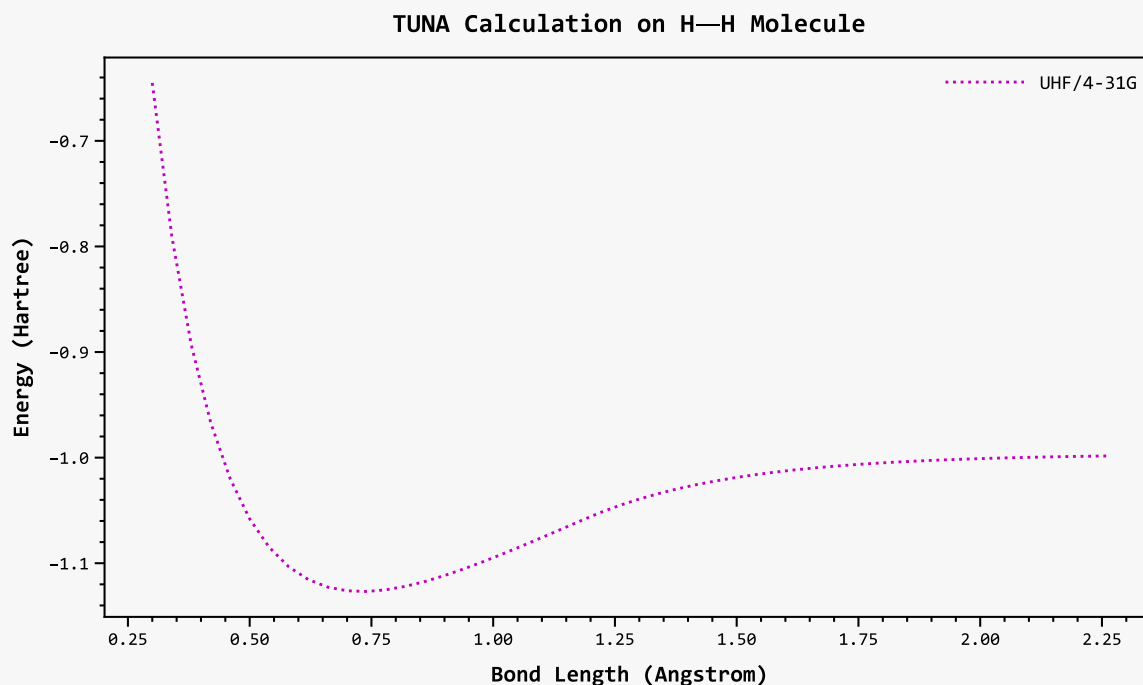


Figure 4.5 Potential energy surface of H_2 calculated in TUNA with UHF/4-31G

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 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 4.6.

```
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
```

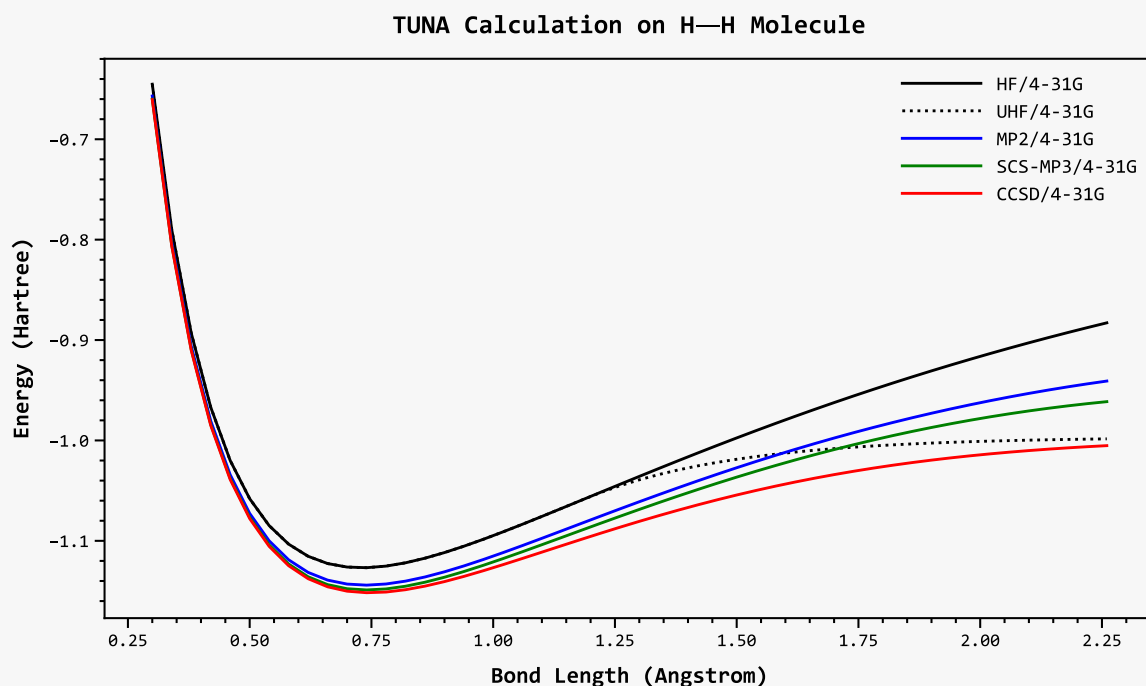


Figure 4.6 Potential energy surfaces of H_2 calculated in TUNA with various electronic structure methods in 4-31G basis set

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).

The following two calculations on the H-F molecule produced Figure 4.7:

```
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
```

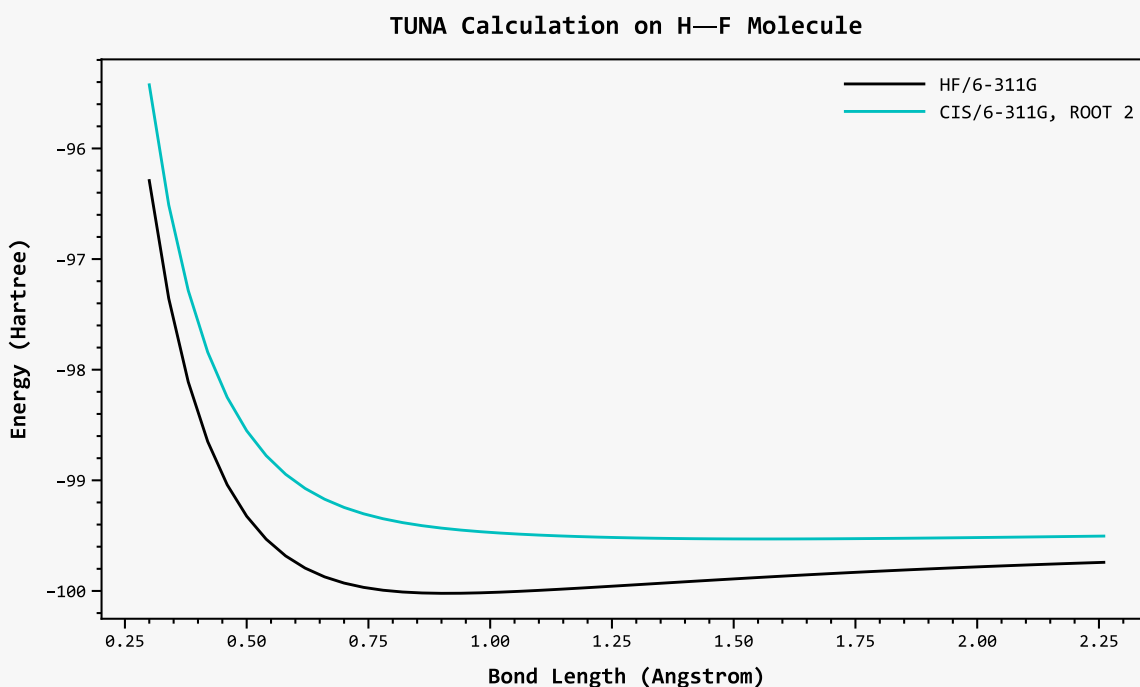


Figure 4.7 Potential energy surfaces of hydrogen fluoride ground state and first excited singlet state (root 2) calculated in TUNA with CIS/6-311G

4.1.3 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 strategy can be disabled using the NOMOREAD keyword.

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

$$\nabla E_i = \frac{\partial E}{\partial R} = \frac{E(R + \delta R) - E(R - \delta R)}{2\delta R}, \quad (4.22)$$

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

$$H_i = \frac{\nabla E_i - \nabla E_{i-1}}{R_i - R_{i-1}}. \quad (4.23)$$

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

$$R_{i+1} = R_i + \frac{\nabla E_i}{H_i}. \quad (4.24)$$

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

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

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

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 4.2. 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 4.2 Optimisation convergence criteria in TUNA

Convergence	Gradient / hartree bohr ⁻¹	Step / Å
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 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 H₂.

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
```

4.1.3.1 Force Calculations

To determine the force on the molecule in its current iteration, a calculation type with the FORCE keyword can be requested.

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

This calculates the energy and the gradient. This gradient and the calculated step length is printed to the terminal and the calculation terminates.

4.1.3.2 Dispersion Correction

Currently, the only available form of dispersion correction is the D2 semi-empirical scheme by Grimme [17]. Here, the energy due to dispersion is calculated by

$$E_{D2} = -\frac{s_6}{f_{\text{damp}}} \frac{C_6^{AB}}{R^6}, \quad (4.25)$$

where the damping function is given by

$$f_{\text{damp}} = 1 + \exp \left[-d \left(\frac{R}{R_{\text{vdW}}^{AB}} - 1 \right) \right], \quad (4.26)$$

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

The default value of the damping factor d is 20, and the s_6 value is 1.2 in line with the parameterisation of Hartree–Fock in ORCA [4, 5]. The values of the Van der Waals radii, R_{vdW} , and C_6 coefficients are element-specific (see Table 4.4). For correlated or 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). The optimised values of s_6 are implemented for various DFT methods, shown in Table 4.3

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

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

4.1.4 Harmonic Frequency

A harmonic frequency calculation in TUNA 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
```

This 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.

Table 4.3 Optimised values for D2 dispersion for various DFT methods

Exchange–Correlation Functional	s_6
[Default]	1.20
PBE	0.75
BLYP	1.20
B3LYP	1.05
B2PLYP	0.55
mPW2PLYP	0.40
BP86	1.05
TPSS	1.00

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

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

$$k = \frac{\partial^2 E}{\partial R^2} . \quad (4.27)$$

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

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

$$\omega = \sqrt{\frac{k}{\mu}} , \quad (4.28)$$

where μ is the reduced mass, calculated using the atomic masses shown in Table 4.5. 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. The atomic masses can be printed using P.

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

The vibrational frequency, reduced mass and force constant are all printed under the "harmonic

Table 4.4 Parameters for D2 semi-empirical dispersion correction in TUNA

Atom	C_6 / au	R_{vdW} / 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

frequency” header. 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 some different methods implemented in TUNA are shown in Table 4.6, where coupled cluster methods perform much better than perturbation theory. The following values were calculated with the parameters here, for instance for CCSD:

TUNA OPTFREQ : H H 1.0 : CCSD aug-cc-pVTZ : EXTREME EXTREMEOPT

Table 4.5 Default atomic masses in TUNA. Data from [18, 19]

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

4.1.4.1 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 mass-weighted normal coordinates,

$$\frac{\partial \mu}{\partial \mathbf{q}}, \quad (4.29)$$

which is calculated by the central differences method of numerical differentiation simultaneously

Table 4.6 Harmonic frequencies of H₂ calculated in the aug-cc-pVTZ basis set in TUNA

Method	Harmonic Frequency / cm ⁻¹
HF	4585
LSDA	4181
PBE	4314
B3LYP	4414
DSD-BLYP	4506
MP2	4514
OMP2	4509
SCS-MP2	4500
MP3	4464
SCS-MP3	4487
MP4	4429
CCD	4409
CCSD	4398
Exact	4401

as 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⁻¹. This coefficient is calculated by

$$\mathcal{A} = \frac{N_A}{12c^2\epsilon_0} \left(\frac{\partial \mu}{\partial \mathbf{q}} \right)^2, \quad (4.30)$$

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 [20]. This is printed by TUNA under the "transition intensity" header, along with the dipole moment derivative and the squared derivative in atomic units.

4.1.4.2 Thermochemistry

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

$$U = E + E_{\text{zero-point}} + E_{\text{translational}} + E_{\text{vibrational}} + E_{\text{rotational}} , \quad (4.31)$$

and entropy,

$$S = S_{\text{electronic}} + S_{\text{translational}} + S_{\text{vibrational}} + S_{\text{rotational}} , \quad (4.32)$$

are first calculated, before the enthalpy,

$$H = U + k_{\text{B}}T , \quad (4.33)$$

and Gibbs free energy,

$$G = H - TS , \quad (4.34)$$

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

$$E_{\text{vibrational}} = \frac{\omega}{\exp(\omega/k_{\text{B}}T) - 1} , \quad (4.35)$$

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

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

$$S_{\text{electronic}} = k_{\text{B}} \ln(2S + 1) . \quad (4.36)$$

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

$$S_{\text{translational}} = k_{\text{B}} \left[\frac{5}{2} + \ln\left(\frac{Mk_{\text{B}}T}{h}\right)^{\frac{3}{2}} + \ln\left(\frac{k_{\text{B}}T}{p}\right) \right] \quad (4.37)$$

$$S_{\text{vibrational}} = k_{\text{B}} \left[\frac{\omega}{k_{\text{B}}T(\exp(\omega/k_{\text{B}}T) - 1)} - \ln \left(1 - \exp \left(-\frac{\omega}{k_{\text{B}}T} \right) \right) \right] \quad (4.38)$$

$$S_{\text{rotational}} = k_{\text{B}} + k_{\text{B}} \ln \left(\frac{k_{\text{B}}T}{2\pi\sigma Bc} \right) \quad (4.39)$$

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

4.1.5 Anharmonic Frequency

Since the potential energy surface of a diatomic is one-dimensional, it is feasible to “exactly” solve the nuclear Schrödinger equation using the surface generated by TUNA as the potential, from any electronic structure method and basis set combination. The time taken to generate the potential energy surface is the limiting step, and the building of the nuclear Hamiltonian and subsequent diagonalisation for vibrational energy levels and wavefunctions is rapid.

The vibrational energy levels begin at the anharmonic zero-point energy — since the quantum harmonic oscillator model is not used, rather the full PES. The energy gap between this level and the first excited vibrational state is the fundamental transition, and comparison of this to the harmonic frequency allows the calculation of the anharmonicity constant, χ .

Assuming full configuration interaction and a complete one-electron basis set, this allows the calculation of the full anharmonic vibrational frequency to sub-cm^{-1} accuracy.

Since generation of the entire PES is very slow, TUNA uses some tricks to speed this up. The first and simplest is interpolation between calculated points, to get a smooth surface with many fewer energy evaluations.

A TUNA ANHARM calculation begins with optimisation to the equilibrium bond length, followed by calculation of the harmonic frequency. All calculations are used with ‘EXTREME’ SCF convergence

criteria by default. Next, from this equilibrium geometry, the PES is calculated in both directions by 0.5 angstroms. The Hamiltonian is formed and diagonalised, and the fundamental frequency determined. The PES is extended symmetrically in both directions for another 0.1 angstroms, and the fundamental frequency is calculated again and compared. This process repeats until the fundamental transition frequency is converged to 0.1 cm^{-1} , where the calculation terminated. The infrared spectrum is printed for the various transitions, and their intensities are calculated. These intensities are correct for any method with an analytical density matrix — this is used to evaluate the dipole moment along the path, which is used to multiply the vibrational overlap between states to evaluate the intensities. The anharmonic zero point energy is printed, and can be used in thermochemistry calculations along with the fundamental transition frequency for vibrational entropies via the additional print keyword, P.

4.1.5.1 Transition Intensity

4.1.5.2 Plotting Nuclear Wavefunctions

ANHARMCONV The nuclear wavefunctions and the potential energy surface can be printed with the PLOTVIB keyword.

The Hamiltonian of a diatomic moving through a potential energy surface calculated by TUNA is

$$\hat{H}_{\text{nuc}} = -\frac{1}{2\mu} \frac{\partial^2}{\partial x^2} + V_{\text{PES}} \quad (4.40)$$

The nuclear Schrödinger equation is solved on a grid, using a basis set of delta functions. This gives a matrix on the order of 1000x1000, but it can be diagonalised for the first few eigenvectors and eigenvalues efficiently since it is tridiagonal - only next door points are coupled due to the kinetic energy term so the matrix is very sparse.

$$\hat{H}_{\text{nuc}}\psi(x_n) = -\frac{1}{2\mu} \frac{\psi(x_{n+1}) - 2\psi(x_n) + \psi(x_{n-1}))}{(x_n - x_{n-1})^2} + V_{\text{PES}}(x_n)\psi(x_n) \quad (4.41)$$

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

$$\hat{H}_{\text{nuc}}\psi(x_n) = \left[\frac{1}{\mu\Delta x^2} + V_{\text{PES}}(x_n) \right] \psi(x_n) - \frac{\psi(x_{n+1}) + \psi(x_{n-1}))}{2\mu\Delta x^2} \quad (4.42)$$

With the diagonal part

$$\hat{H}_n^n = \frac{1}{\mu\Delta x^2} + V_{\text{PES}}(x_n) \quad (4.43)$$

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

$$\hat{H}_n^m = -\frac{\delta_{n+1,m} + \delta_{n-1,m}}{2\mu\Delta x^2} \quad (4.44)$$

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

$$\hat{H}_n^m = -\frac{1}{2\mu\Delta x^2} \quad (4.45)$$

and all the other elements are zero. This Hamiltonian is then formed and diagonalised for the i th vibrational energy level and wavefunction

$$\hat{H}\psi_i(x) = E_i\psi_i(x) \quad (4.46)$$

The fundamental transition frequency is

$$\omega_{\text{fund}} = E_1 - E_0 \quad (4.47)$$

where E_0 is the zero-point energy. The intensity of a vibrational transition depends on the transition dipole moment operator

$$\mathcal{A}_{ij} \propto \langle \psi_i | \hat{\mu} | \psi_j \rangle = \int \psi_i^*(x) \mu(x) \psi_j(x) dx \quad (4.48)$$

The anharmonicity constant is

$$\chi = \frac{\omega_0^1 - \omega_1^2}{2\omega_{\text{harm}}} \quad (4.49)$$

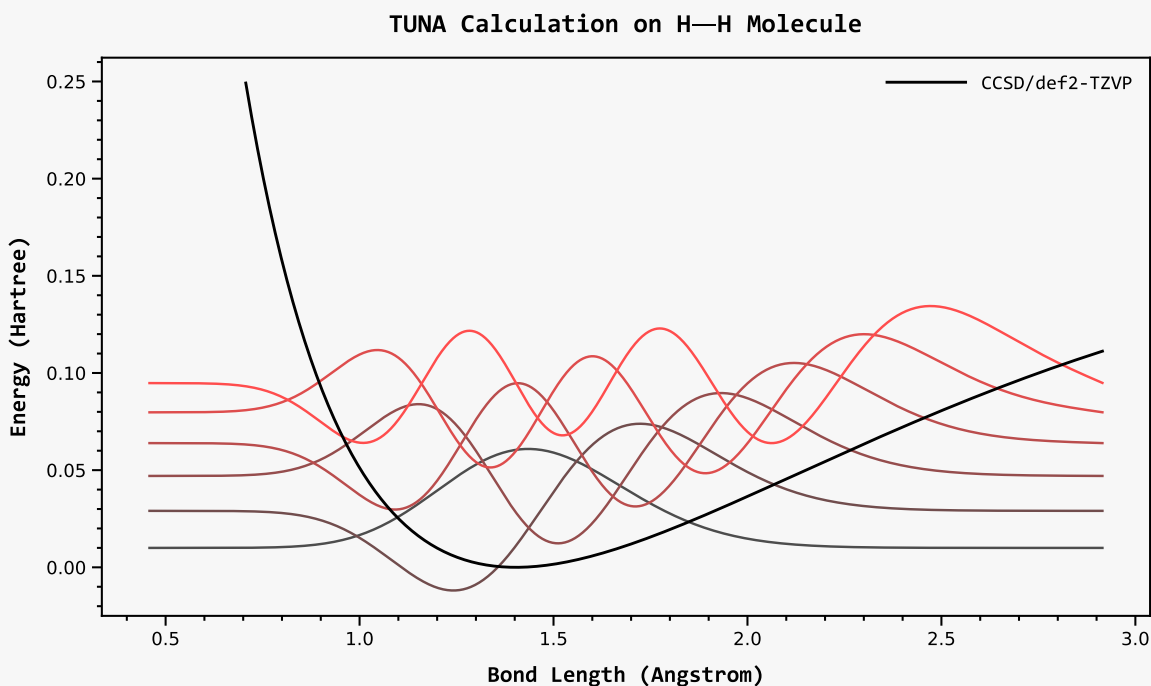
where ω_i^j is the transition element from the i th to j th nuclear state.

TUNA ANHARM : H H 1.0 : CCSD def2-TZVP : PLOTVIB

Switching to def2-QZVP gives 4159 /cm, the experimental value is 4161 per cm.

The step length is angstroms by default, which is sufficiently converged in most cases, but can be changed with the STEP keyword.

Bear in mind that despite the speedups, these are very intensive calculations and around 30 energy evaluations are necessary in total.



4.1.6 Ionisation Potential and Electron Affinity

There are inbuilt calculation types to determine the ionisation potential,

$$IP = E(X^+) - E(X) , \quad (4.50)$$

and the electron affinity

$$EA = E(X) - E(X^-) . \quad (4.51)$$

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

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

The analogous electron affinity calculation is:

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

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

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

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

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

$$\text{IP} = E(\text{Be}) - E(\text{Be}^{2+}) . \quad (4.52)$$

```
TUNA IP : Be : CCSDTQ cc-pVDZ : NELEC 2
```

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

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

4.1.7 *Ab Initio* Molecular Dynamics

The implementation in TUNA of *ab initio* molecular dynamics (AIMD) is Born–Oppenheimer molecular dynamics [21], 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 [22], the *ab initio* energy and gradient must be recalculated at every timestep.

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

For instance, a 200 step AIMD 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 200
```

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

$$\mathbf{R}(t + \Delta t) = \mathbf{R}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2, \quad (4.53)$$

and velocities are updated by

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t + \Delta t)}{2}\Delta t. \quad (4.54)$$

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

$$K = 3k_B T, \quad (4.55)$$

where

$$K = \frac{1}{2}m_A\mathbf{v}_A^2 + \frac{1}{2}m_B\mathbf{v}_B^2. \quad (4.56)$$

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.

4.2 Electronic Structure Methods

The electronic structure methods implemented in TUNA are shown in Table 4.7. Most of these methods are available for restricted and unrestricted references — the unrestricted reference can be requested for a singlet molecule by adding the letter "U" to the start of a method. For instance, a CCD calculation on a UHF reference becomes "UCCD".

Table 4.7 Wavefunction-based electronic structure methods implemented in TUNA

Keyword	Electronic Structure Method
HF	Hartree–Fock theory
UHF	Unrestricted Hartree–Fock theory
H	Hartree theory
UH	Unrestricted Hartree theory
MP2	Second-order Møller–Plesset perturbation theory
UMP2	Unrestricted second-order Møller–Plesset perturbation theory
SCS-MP2	Spin-component-scaled MP2 perturbation theory
USCS-MP2	Unrestricted spin-component-scaled MP2 theory
OMP2	Orbital-optimised MP2 theory
UOMP2	Unrestricted orbital-optimised MP2 theory
IMP2	Iterative MP2 theory
LMP2	Laplace transform MP2 theory
MP3	Third-order Møller–Plesset perturbation theory
UMP3	Unrestricted third-order Møller–Plesset perturbation theory
SCS-MP3	Spin-component-scaled MP3 perturbation theory
USCS-MP3	Unrestricted spin-component-scaled MP3 theory

Keyword	Electronic Structure Method
MP4	Fourth-order Møller–Plesset perturbation theory
MP4[DQ]	Restricted MP4 theory with no singles or triples
MP4[SDQ]	Restricted MP4 theory with no triples
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
CCSD[T]	Coupled cluster singles, doubles and perturbative triples
UCCSD[T]	Unrestricted coupled cluster singles, doubles and perturbative triples
CCSDT	Coupled cluster singles, doubles and triples
UCCSDT	Unrestricted coupled cluster singles, doubles and triples
CCSDT[Q]	Coupled cluster singles, doubles, triples and perturbative quadruples
CCSDTQ	Coupled cluster singles, doubles, triples and quadruples

Keyword	Electronic Structure Method
QCISD	Quadratic configuration interaction singles and doubles
UQCISD	Unrestricted quadratic CI singles and doubles
QCISD[T]	Quadratic CI singles, doubles and perturbative triples
UQCISD[T]	Unrestricted quadratic CI singles, doubles and perturbative triples

A variety of density functional theory methods are also available, listed in section 4.2.4.

4.2.1 Hartree–Fock Theory

Both restricted and unrestricted Hartree–Fock calculations are available. 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 [24, 25] turn the variational differential restricted Hartree–Fock equations [26] into a simpler eigenvalue problem,

$$\mathbf{FC} = \mathbf{SC}\epsilon, \quad (4.57)$$

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{V}_{ne} , Coulomb matrix, \mathbf{J} , and exchange matrix, \mathbf{K} , by

$$\mathbf{F} = \mathbf{T} + \mathbf{V}_{\text{ne}} + 2\mathbf{J} - \mathbf{K}. \quad (4.58)$$

These matrices are determined from the one- and two-electron integrals of basis functions, discussed in section 4.3. 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

$$\mathbf{F}' = \mathbf{X}^\dagger \mathbf{F} \mathbf{X}, \quad (4.59)$$

the Roothaan–Hall equations become a conventional eigenvalue problem,

$$\mathbf{F}'\mathbf{C}' = \mathbf{C}'\epsilon . \quad (4.60)$$

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,

$$\mathbf{C} = \mathbf{X}\mathbf{C}' . \quad (4.61)$$

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

$$P_{\mu\nu} = \sum_i^{\text{occ}} f_i C_{\mu i} C_{\nu i}^* . \quad (4.62)$$

The occupation numbers, f_i are one for UHF and two for RHF. Finally, the energy can be evaluated by contracting this new density matrix with the Fock and core Hamiltonian matrices, and adding on the constant nuclear repulsion energy,

$$E = V_{\text{nn}} + \frac{1}{2} \sum_{\mu\nu} P_{\mu\nu} (F_{\mu\nu} + H_{\mu\nu}^{\text{core}}) . \quad (4.63)$$

Here, $\mathbf{H}^{\text{core}} = \mathbf{T} + \mathbf{V}_{\text{ne}}$. 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,

$$2\mathbf{J} - \mathbf{K} = \sum_{\kappa\lambda} P_{\kappa\lambda} \left(\langle \mu\nu | \kappa\lambda \rangle - \frac{1}{2} \langle \mu\nu | \lambda\kappa \rangle \right) . \quad (4.64)$$

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

4.2.1.1 Initial Guess

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

$$\mathbf{F} = \mathbf{H}^{\text{core}} . \quad (4.65)$$

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

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

Another guess strategy, which is widely used in quantum chemistry codes, is the superposition of atomic densities guess.

Here the guess density matrix is calculated as a block diagonal matrix of atomic density matrices, assuming zero interaction between the atoms (ie. a simple superposition of the densities).

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_B \end{bmatrix} \quad (4.66)$$

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

$$\mathbf{P}_2 = \mathbf{S}_2^{-1} \mathbf{S}_{12} \mathbf{P}_1 \mathbf{S}_{12} \mathbf{S}_2^{-1}, \quad (4.67)$$

where \mathbf{P}_1 is the minimal basis set and \mathbf{P}_2 is the requested basis set. For this, the cross overlap matrix between the two basis sets is needed, which is happily calculated by the molecular integral engine in TUNA. A downside of this is that simple densities like the hydrogen molecule are not as well-represented, so may take longer to converge the SCF. The superposition of atomic densities guess strategy can be requested with the `SADGUESS` keyword.

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

The default guess strategy in TUNA embodies the best of both worlds, and can be accessed via the `SCFGUESS` keyword. Here, a STO-3G minimal basis set SCF calculation is first fully converged, using the guess for *that* calculation from superposition of atomic densities. The converged density matrix is then projected using equation 4.67 onto the larger basis set, and the true calculation begins.

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

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

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. An important example of this is dihydrogen at large bond lengths. To get around this, TUNA rotates the HOMO with the LUMO after the initial guess density is made, by

$$\begin{bmatrix} \psi'_{\text{HOMO}} \\ \psi'_{\text{LUMO}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \psi_{\text{HOMO}} \\ \psi_{\text{LUMO}} \end{bmatrix}. \quad (4.68)$$

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. Orbital rotation can be activated with the ROTATE keyword. If the default value of θ is not converging well, use the ROTATE keyword followed by the desired rotation angle in degrees to change it. The value of THETA is printed in the TUNA output when orbital rotation is used.

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 45
```

A well-considered guess rotation may allow a metastable energy state to be converged.

4.2.1.2 SCF Convergence

A number of SCF convergence criteria are available in TUNA, shown in Table 4.8. 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. Convergence is measured by the change in energy, maximum and root-mean-square change in the density matrix, and the DIIS error (labelled Error in the TUNA output), which is the root-mean-square of the commutator between **F** and **PS**.

There are several methods available to accelerate SCF convergence. The most powerful and

Table 4.8 Self-consistent field convergence criteria in TUNA

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

prevalent is probably Fock matrix extrapolation, also known as direct inversion of the iterative subspace (DIIS) [27].

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

$$\mathbf{SPF} - \mathbf{FPS} = \mathbf{0} . \quad (4.69)$$

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

$$\mathbf{SP}_i \mathbf{F}_i - \mathbf{F}_i \mathbf{P}_i \mathbf{S} = \mathbf{e}_i . \quad (4.70)$$

The main quantifier for error (distance from convergence) in TUNA is the root-mean-square of this error matrix, and is printed during the SCF cycle iterations.

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

$$\mathbf{F}_{j+1} = \sum_j c_j \mathbf{F}_j . \quad (4.71)$$

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

$$\mathbf{P}' = a\mathbf{P}_{\text{old}} + (1 - a)\mathbf{P} . \quad (4.72)$$

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 200 iterations and $a = 0.5$ can be called by

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

The value of a can be chosen specifically with the DAMP keyword. For example, a static damping factor of $a = 0.3$ can be requested by:

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

However, the default method is dynamic damping, where the damping factor, a , is some function of the distance from convergence. The implemented scheme for dynamic damping is that from Zerner and Hehenberger [28]. In this method, Mulliken gross atomic populations [12] are calculated in each cycle and extrapolated to derive an optimal damping factor, the average of the calculated atomic damping factors weighted by the proportion of basis functions on each atom. This uses the density matrix from the previous SCF iteration before and after damping, as well as the density matrix from the SCF iteration before that.

These several points of data allow an accurate extrapolation to give a damping factor that more rapidly converges, and avoids unnecessary damping in the static case. The dynamical damping method is enabled by default, can be disabled with the NODAMP keyword and is compatible with DIIS. The maximum damping factor is 0.7 by default, and controlled with the MAXDAMP keyword.

4.2.1.3 Hartree Theory

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

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

The Hartree bond length for dihydrogen is much longer (and wronger) compared to the Hartree-Fock bond length.

To more precisely investigate Fock exchange, it can be taken out of Hartree–Fock using the HFX parameter. The following calculation represents 50% of the exact exchange being used, and gives a somewhat better bond length.

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

4.2.2 Møller–Plesset Perturbation Theory

In TUNA, conventional Møller–Plesset perturbation theory [29] to second (MP2) and third (MP3) order are implemented for both restricted and unrestricted references, and fourth order perturbation theory (MP4) is implemented for restricted references. In addition, Møller–Plesset methods can be “spin-component-scaled” (SCS), which has been reported to increase their accuracy [30, 31]. These methods can be called by writing MP2, SCS-MP2, MP3, SCS-MP3 or MP4, 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 : USCS-MP2 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. The MP3 and MP4 methods also require more steeply scaling contractions to calculate the energy. However, at the number of basis functions involved in a typical TUNA calculation, these steps are small compared to the time to calculate two-electron integrals. If these methods do become slow, core electrons can be frozen (see section 4.2.2.5).

The (SCS-)MP2 and (SCS-)MP3 methods are implemented in both a spatial orbital and spin-orbital form. This means that restricted references benefit from a speedup due to the spin symmetry. For unrestricted references, the two electron integrals are transformed into a spin-orbital basis by

$$\langle ij|ab\rangle = \sum_{\mu\nu\kappa\lambda} C_{\mu i} C_{\nu j} \langle \mu\nu|\kappa\lambda\rangle C_{\kappa a} C_{\lambda b} . \quad (4.73)$$

These are then antisymmetrised to give

$$\langle ij||ab\rangle = \langle ij|ab\rangle - \langle ij|ba\rangle . \quad (4.74)$$

Correlated calculations in a spin orbital basis use these transformed integrals to calculate the energy. In TUNA, this energy is calculated via the double excitation amplitudes,

$$t_{ij}^{ab} = \langle ij || ab \rangle (\mathcal{E}_{ij}^{ab})^{-1} , \quad (4.75)$$

where

$$\mathcal{E}_{ij}^{ab} = \varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b , \quad (4.76)$$

by

$$E_{\text{MP2}} = \frac{1}{4} \sum_{ijab} \langle ij || ab \rangle t_{ij}^{ab} . \quad (4.77)$$

For both restricted and unrestricted references, TUNA calculates the same- and opposite-spin contributions to the MP2 energy. For unrestricted references, this is decomposed further into the energy due to α - α , β - β and α - β electron pairs.

The approximate doubles amplitudes are also used to update the occupied–occupied and virtual–virtual blocks of the density matrix, for the unrelaxed density which is automatically calculated during an MP2 calculation. For instance, the occupied–occupied blocks of the spin-orbital density matrix are given by

$$P_i^j = -\frac{1}{2} \sum_{abk} t_{jk}^{ab} t_{ik}^{ab} . \quad (4.78)$$

The unrelaxed density is implemented for all MP2-based methods, but not for MP3 or MP4 — in these methods the MP2 density will be calculated instead. 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.

In MP4, the first step is to calculate the second-order amplitudes and lagrange multipliers — these are then used to calculate the MP4 energy, which can be decomposed as

$$E_{\text{MP4}} = E_{\text{singles}}^{(4)} + E_{\text{doubles}}^{(4)} + E_{\text{triples}}^{(4)} + E_{\text{quadruples}}^{(4)} . \quad (4.79)$$

The calculation of the fourth-order triples is the steepest scaling step, at $\mathcal{O}(N^7)$, while the other steps scale at most at $\mathcal{O}(N^6)$.

Therefore, a speedup can be achieved by simply neglecting the calculation of the triples, yielding the MP4(SDQ) method. This can be calculated in TUNA using the calculation type MP4[SDQ], for instance:

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

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

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

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

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

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

4.2.2.1 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

$$E_{\text{SCS-MP2}} = E_{\text{HF}} + c_{\text{S}} E_{\text{MP2}}^{\text{SS}} + c_{\text{O}} E_{\text{MP2}}^{\text{OS}}, \quad (4.80)$$

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

```
TUNA SPE : H H 0.74 : SCS-MP2 6-31G : SSS 0 OSS 1.2
```

In SCS-MP3, the total energy is given by

$$E_{\text{SCS-MP3}} = E_{\text{SCS-MP2}} + c_{\text{MP3}} E_{\text{MP3}}, \quad (4.81)$$

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

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

4.2.2.2 Orbital-optimised MP2

The MP2 energy can be minimised with respect to orbital rotations, in orbital-optimised MP2, which lowers the MP2 energy further [32, 33]. 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, Θ , by

$$\mathbf{C}_{i+1} = \mathbf{C}_i \exp(\Theta) . \quad (4.82)$$

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 MP CONV keyword. The maximum number of OMP2 iterations is 30 by default, mutable with the MP MAXITER keyword.

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

4.2.2.3 Iterative MP2

Conventional MP2 requires a Hartree–Fock reference. For calculations with non-canonical orbitals (such as those using localised orbitals), the MP2 energy cannot be calculated in the conventional way, but can be evaluated through minimisation of the Hylleraas functional [34].

In iterative MP2, the following residuals are iteratively minimised

$$R_{ij}^{ab} = (ij|ab) + F_{ac} t_{ij}^{cd} S_{db} + S_{ac} t_{ij}^{cd} F_{db} - S_{ac} F_{ik} t_{kj}^{cd} S_{db} - S_{ac} F_{kj} t_{ik}^{cd} S_{db} . \quad (4.83)$$

Where the overlap matrix elements allow for the possibility of a non-orthogonal basis and Fock matrix elements allow for the possibility of non-canonical molecular orbitals. The doubles amplitudes are updated as

$$t_{ij}^{ab} \leftarrow t_{ij}^{ab} + (\mathcal{E}_{ij}^{ab})^{-1} R_{ij}^{ab} . \quad (4.84)$$

The pair energies can then be calculated as

$$e_{ij} = \sum_{ab} \left[(ij|ab) + R_{ij}^{ab} \right] (4t_{ij}^{ab} - 2t_{ji}^{ab}), \quad (4.85)$$

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

$$E_{\text{MP2}} = \frac{1}{2} \sum_{ij} e_{ij}, \quad (4.86)$$

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

An example of an iterative MP2 calculation on the carbon monoxide molecule, with a reduced number of maximum iterations is:

TUNA SPE : C O 1.6 : IMP2 def2-SVP : MPCONV 1e-10 MPMAXITER 8

This is a work in progress and only implemented for restricted references. The unrelaxed MP2 density matrix is also calculated.

4.2.2.4 Laplace Transform MP2

Conventional MP2 requires molecular orbitals and eigenvalues from diagonalisation of the Fock matrix. However, as matrix diagonalisation has formal scaling $\mathcal{O}(N^3)$, this cannot be implemented as a linear scaling method. One route towards linear scaling MP2 is to exploit the Laplace transform for the energy denominators and express the MP2 energy as a functional of the Hartree–Fock density matrix [35].

Using the Laplace identity for energy denominators,

$$(\mathcal{E}_{ij}^{ab})^{-1} = \int_0^\infty e^{-s(\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b)} \mathrm{d}s, \quad (4.87)$$

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

$$\mathbf{X}(s) = e^{s\mathbf{P}\mathbf{F}} \mathbf{P} \quad \mathbf{Y}(s) = e^{-s\mathbf{Q}\mathbf{F}} \mathbf{Q}, \quad (4.88)$$

where \mathbf{Q} is the “hole” density matrix, and contracts them with atomic orbital basis ERIs to obtain

an integrand

$$e(s) = \sum_{mnl\sigma} \sum_{gdke} X_{mg}(s) Y_{nd}(s) X_{kl}(s) Y_{e\sigma}(s) (gd|ke) L_{mnl\sigma}, \quad (4.89)$$

which can be integrated by numerical quadrature

$$E_{\text{MP2}} = - \int_0^\infty e(s) ds. \quad (4.90)$$

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

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

where $g'(r_k) = ds/dr$ is the analytic weight used in the TUNA. The Euler–Maclaurin quadrature method has been implemented which needs very few quadrature points to reach microhartree accuracy [36].

The only adjustable parameter for this is the number of quadrature points, which can be changed by the MPGRID keyword which is 20 by default. An example of a calculation with Laplace transform MP2 with 50 grid points is:

```
TUNA SPE : H H 0.6 : LMP2 6-311++G : MPGRID 50
```

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

4.2.2.5 Freezing Core Electrons

Because of the small size of TUNA calculations, all electrons are correlated by default. The frozen core approximation is often applied in correlated calculations and consists of neglecting correlation effects for electrons in the low lying core orbitals, like 1s orbitals in second period elements.

This is simple to do in TUNA: use the FREEZECORE keyword. Using FREEZECORE on its own will freeze the default number of core electrons depending on the elements used in the calculation. This number is 0 for H–Be, 2 for B–Mg and 10 for Al–Ar. Alternatively, if a specific number of *orbitals* (not electrons) is wanted frozen, this number can be given after the FREEZECORE keyword. For example, the following calculation will freeze the 3 lowest energy orbitals:

```
TUNA SPE : N O 1.0 : OMP2 6-311G : FREEZECORE 3
```

During the transformation to a molecular orbital basis, TUNA will print the number of frozen orbitals — which will be half the number of frozen electrons for RHF references, or the same as the number of frozen electrons for UHF references.

4.2.3 Coupled Cluster Theory

In coupled cluster theory, the exact wavefunction is given by

$$|\Psi\rangle = e^{\hat{T}}|\Phi_0\rangle, \quad (4.91)$$

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

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \dots, \quad (4.92)$$

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

$$\hat{T}_1 = \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i \quad (4.93)$$

$$\hat{T}_2 = \frac{1}{4} \sum_{ijab} t_{ij}^{ab} \hat{a}_b^\dagger \hat{a}_j \hat{a}_a^\dagger \hat{a}_i. \quad (4.94)$$

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, $e^{\hat{T}}$, as a Taylor series yields

$$e^{\hat{T}} = 1 + \hat{T} + \frac{1}{2!} \hat{T}^2 + \frac{1}{3!} \hat{T}^3 + \dots \quad (4.95)$$

which, upon substitution of the definition of \hat{T} , yields

$$e^{\hat{T}} = 1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2} \hat{T}_1^2 + \frac{1}{2} \hat{T}_1 \hat{T}_2 + \frac{1}{2} \hat{T}_2 \hat{T}_1 + \frac{1}{2} \hat{T}_2^2 + \dots. \quad (4.96)$$

Curtailing \hat{T} yields approximate wavefunctions. For instance curtailing \hat{T} to $\hat{T}_1 + \hat{T}_2$ yields the coupled cluster singles and doubles (CCSD) method, which, as demonstrated in equation 4.96, includes approximate contributions from higher than double excitations, through so-called “discon-

nected" excitations like $\hat{T}_1\hat{T}_2$. This means coupled cluster accounts for more electron correlation than configuration interaction curtailed to the same level.

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

$$E_{\text{CC}} = E_{\text{HF}} + \frac{1}{4} \sum_{ijab} \langle ij || ab \rangle t_{ij}^{ab} + \frac{1}{2} \sum_{ijab} \langle ij || ab \rangle t_i^a t_j^b. \quad (4.97)$$

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, the residual. This extrapolation is on by default and can be turned off with the NODIIS keyword, and the number of amplitudes that DIIS will remember (six by default) can be changed with the keyword DIIS [Num. Amps].

Damping of successive t -amplitudes can also be requested with the CCDAMP keyword, which, whose parameter determines how much old t -amplitudes should be mixed with new ones. Damping is off by default. A CCSD calculation with damping at 50% and DIIS remembering 7 sets of amplitudes 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 CC CONV keyword and $10^{-8} E_h$ by default. In addition to energy convergence, the amplitudes themselves must converge to within 10^{-8} , which can be adjusted with the AMP CONV keyword. 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. A maximum of CCMAXITER iterations are iterated through, which is 50 by default.

The norm of the singles amplitude and the \mathcal{T}_1 diagnostic is then printed, which gives an idea if a single reference calculation is valid [37]. The \mathcal{T}_1 diagnostic is related to the norm of the t_i^a

amplitudes by

$$\mathcal{T}_1 = \frac{||\mathbf{t}_1||}{\sqrt{N_{\text{occ}}}}, \quad (4.98)$$

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

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

All coupled cluster methods, except CCSDT(Q) and CCSDTQ, are available in both spin-unrestricted and restricted spin-adapted forms, making calculations on closed-shell singlet molecules much faster. Methods including connected quadruple excitations can only be used with a restricted Hartree–Fock reference.

4.2.3.1 Coupled Cluster Doubles

In CCD, the cluster operator is curtailed to

$$T = T_2, \quad (4.99)$$

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
```

4.2.3.2 Coupled Cluster Singles and Doubles

In CCSD, the cluster operator is curtailed to

$$T = T_1 + T_2, \quad (4.100)$$

giving equations where t_i^a and t_{ij}^{ab} need to be optimised together [38]. 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. However, in TUNA, if you can get through the two-electron integrals, CCSD is unlikely to be a problem.

4.2.3.3 Coupled Cluster Singles, Doubles and Triples

In CCSDT, the cluster operator is curtailed to

$$T = T_1 + T_2 + T_3 , \quad (4.101)$$

giving equations where t_i^a , t_{ij}^{ab} and t_{ijk}^{abc} need to be optimised together [39]. 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 CCSDT method is very slow, with formal $\mathcal{O}(N^8)$ scaling with respect to basis functions.

4.2.3.4 Coupled Cluster Singles, Doubles, Triples and Quadruples

In CCSDTQ, the cluster operator is curtailed to

$$T = T_1 + T_2 + T_3 + T_4 , \quad (4.102)$$

giving equations where t_i^a , t_{ij}^{ab} , t_{ijk}^{abc} and t_{ijkl}^{abcd} need to be optimised together [40, 41]. The CCSDTQ method can be requested using CCSDTQ. For example, a CCSDTQ single point energy on the beryllium atom in the cc-pVDZ basis set is called by:

TUNA SPE : Be : CCSDTQ cc-pVDZ

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

4.2.3.5 Coupled Cluster with Perturbative Triples

In CCSD(T), often called the “gold standard” of quantum chemistry, a perturbative correction is made to the CCSD energy, based on the MP4 and MP5 triple excitations [42, 43]. 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 [37].

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**
```

4.2.3.6 Coupled Cluster with Perturbative Quadruples

If the “gold standard” of CCSD(T) is not enough — for example in a multireference situation — the “platinum standard” of CCSDT(Q) can be used, which incorporates approximate connected quadruples through contributions based on MP5 and MP6 [44, 45]. The cost of this method is large — at $\mathcal{O}(N^9)$, but does not rely on the slow convergence of full CCSDTQ.

Because TUNA is run entirely from the terminal, and round parentheses are not allowed there, CCSDT(Q) is called by the CCSDT[Q] method keyword. A single point calculation on Li-H in the cc-pVTZ basis is given by:

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

4.2.3.7 Linearised Coupled Cluster

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

$$e^T \approx 1 + T = 1 + T_1 + T_2 + T_3 + \dots \quad (4.103)$$

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
```

4.2.3.8 Quadratic Configuration Interaction

A modification to configuration interaction to make it size extensive, quadratic configuration interaction singles and doubles (QCISD) is implemented in TUNA [37, 46]. This turns out to be the same as a simplified version of CCSD.

The keyword for this method is QCISD, and a spin-adapted implementation is available in addition to the spin-orbital method.

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

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

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

4.2.3.9 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. This density can also be used for plotting.

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.

4.2.4 Density Functional Theory

In Kohn–Sham density functional theory [47, 48], the electronic energy as a functional of the electron density, $n(\mathbf{r})$, is calculated by

$$E[n] = T_s[n] + V_{\text{ne}}[n] + J[n] + E_{\text{xc}}[n]. \quad (4.104)$$

In TUNA, the non-interacting kinetic energy, $T_s[n]$, the nuclear–electron energy $V_{ne}[n]$ and the Coulomb energy $J[n]$ are calculated from integrals over Kohn–Sham orbitals. The density is expressed from the occupied Kohn–Sham orbitals,

$$n(\mathbf{r}) = \sum_i^{\text{occ}} |\varphi_i(\mathbf{r})|^2 . \quad (4.105)$$

The implementation in TUNA defines all exchange–correlation functionals as follows

$$E_{xc}[n] = \int d\mathbf{r} f_{xc}[n] \quad (4.106)$$

where $f_{xc} = ne_{xc}$ with e_{xc} being the energy density per particle.

All density functional methods can be used with unrestricted references by prepending a “U” to the functional name. For all methods, the energy and density is calculated self-consistently. For double-hybrids, the MP2 perturbative correction is applied after self-consistency has been achieved.

At self-consistency, the Kohn–Sham equations,

$$\left(-\frac{1}{2}\nabla^2 + \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + \frac{\delta E_{xc}}{\delta n} + v_{\text{ext}}(\mathbf{r}) \right) \varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r}) , \quad (4.107)$$

are satisfied, where $\varphi_i(\mathbf{r})$ and ε_i are the Kohn–Sham orbitals and energies and $v_{\text{ext}}(\mathbf{r})$ is the external potential. Practically, the Roothaan–Hall equations from Hartree Fock are modified so the spin-restricted Fock matrix is calculated using an exchange–correlation matrix, \mathbf{V}_{xc}

$$\mathbf{F} = \mathbf{T} + \mathbf{V}_{ne} + 2\mathbf{J} + \mathbf{V}_{xc} \quad (4.108)$$

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

The exchange or correlation parts of a calculation can be turned off with the NOX or NOC keywords respectively. For example, this calculation reproduces the Hartree–Fock–Slater method with no DFT correlation:

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

Tables 4.9 and 4.10 list all the implemented exchange and correlation functionals currently in TUNA.

Table 4.9 Exchange functionals in TUNA

Functional	Description
S [49]	Slater LDA exchange
B88 [50]	Becke GGA exchange
PBE [51]	Perdew–Burke–Ernzerhof GGA exchange
PW91 [52]	Perdew–Wang GGA exchange
mPW [53]	Modified Perdew–Wang GGA exchange
TPSS [54]	Tao–Perdew–Staroverov–Scuseria <i>meta</i> -GGA exchange

4.2.4.1 Local Density Approximation

The local density contribution to the exchange–correlation matrix is

$$V_{xc}^{\mu\nu} = \int d\mathbf{r} \varphi_{\mu}(\mathbf{r}) \frac{df_{xc}}{dn} \varphi_{\nu}(\mathbf{r}) , \quad (4.109)$$

and the energy density is a functional of the local spin densities, $e_{xc} = e_{xc}[n_{\alpha}, n_{\beta}]$. The implemented local density functionals are listed in Table 4.11. The derivatives with respect to the density needed to form the exchange–correlation matrix are implemented analytically.

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

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

$$E_x[n] = -\alpha \frac{9}{8} \left(\frac{3}{\pi} \right)^{\frac{1}{3}} \int d\mathbf{r} n(\mathbf{r})^{\frac{4}{3}} , \quad (4.110)$$

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

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

Table 4.10 Correlation functionals in TUNA

Functional	Description
VWN3 [55]	Vosko–Wilk–Nusair LDA correlation, parameter set III
VWN5 [55]	Vosko–Wilk–Nusair LDA correlation, parameter set V
PW [56]	Perdew–Wang LDA correlation
P86 [57]	Perdew 1986 GGA correlation
PW91 [52]	Perdew–Wang GGA correlation
PBE [51]	Perdew–Burke–Ernzerhof GGA correlation
LYP [58]	Lee–Yang–Parr GGA correlation
TPSS [54]	Tao–Perdew–Staroverov–Scuseria <i>meta</i> -GGA correlation

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

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

Table 4.11 Local spin-density approximation exchange–correlation functionals in TUNA

Functional	$E_x[n]$	$E_c[n]$
LSDA [49, 55]	S	VWN5
SVWN [49, 55]	S	VWN5
SVWN5[49, 55]	S	VWN5
SVWN3 [49, 55]	S	VWN3
SPW [49, 56]	S	PW
HFS [49]	S	—

4.2.4.2 Generalised Gradient Approximation

The treatment of electronic structure can be improved by incorporating information about the gradients of the spin densities into the exchange–correlation functional, known as generalised

gradient approximation (GGA) functionals,

$$e_{xc} = e_{xc}[n_\alpha, n_\beta, \nabla n_\alpha, \nabla n_\beta] . \quad (4.111)$$

The GGA contribution to the exchange–correlation matrix depends on the derivative of the functional with respect to σ , which is the square density gradient, $\nabla n \cdot \nabla n$. It also depends on the basis function gradients, which are calculated analytically by TUNA and expressed on the integration grid.

$$V_{xc}^{\mu\nu} = \int d\mathbf{r} \varphi_\mu(\mathbf{r}) \frac{df_{xc}}{d\sigma} \nabla \varphi_\nu(\mathbf{r}) \nabla n(\mathbf{r}) \quad (4.112)$$

The implemented GGA functionals are listed in Table 4.12. Among these are the popular PBE and BLYP methods, as well as other widely used combinations of exchange and correlation functionals.

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

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

Table 4.12 (*meta*-)Generalised gradient approximation exchange–correlation functionals in TUNA

Functional	$E_x[n]$	$E_c[n]$
SLYP [49, 58]	S	LYP
HFB [50]	B88	—
BVWN [50, 55]	B88	VWN5
BVWN3 [50, 55]	B88	VWN3
BLYP [50, 58]	B88	LYP
PWP [52, 57]	PW91	P86
BP86 [50, 57]	B88	P86
PBE [51]	PBE	PBE
mPWLYP [58, 59]	mPW	LYP
mPWPW [52, 59]	mPW	PW91
TPSS [54]	TPSS	TPSS

The next step along Perdew’s “Jacob’s ladder” of exchange–correlation functionals [60] are known as *meta*-GGA functionals, which incorporate implicit dependence on the curvature (Laplacian) of

the density via the non-interacting kinetic energy density, $\tau(\mathbf{r})$,

$$e_{xc} = e_{xc}[n_\alpha, n_\beta, \nabla n_\alpha, \nabla n_\beta, \tau_\alpha, \tau_\beta] . \quad (4.113)$$

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

$$\tau(\mathbf{r}) = \frac{1}{2} \sum_i^{\text{occ}} |\nabla \varphi_i|^2 \quad (4.114)$$

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

$$\tau(\mathbf{r}) = \frac{1}{2} \sum_{ij} P_{ij} \nabla \varphi_i \nabla \varphi_j . \quad (4.115)$$

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

$$\alpha(\mathbf{r}) = \frac{\tau(\mathbf{r}) - \tau_W(\mathbf{r})}{\tau_U(\mathbf{r})} , \quad (4.116)$$

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

The exchange–correlation matrix contribution here requires the basis function gradients, as in GGA functionals, but also requires the derivative of the functional with respect to $\tau(\mathbf{r})$.

$$V_{xc}^{\mu\nu} = \int d\mathbf{r} \nabla \varphi_\mu(\mathbf{r}) \frac{df_{xc}}{d\tau} \nabla \varphi_\nu(\mathbf{r}) \quad (4.117)$$

The only currently implemented *meta*-GGA functional is TPSS, which is available self-consistently for restricted and unrestricted references.

TUNA SPE : Li H 1.0 : TPSS cc-pVDZ : PLOTHOMO LOOSEGRID

4.2.4.3 Hybrid Functionals

The next rung up Jacob’s ladder is “hybrid”, non-local exchange–correlation functionals. A hybrid exchange–correlation functional uses a proportion, β , of the Fock exchange expression evaluated

on the Kohn–Sham orbitals

$$E_{xc}[n] = \beta E_x^{\text{HF}} + (1 - \beta) E_x^{\text{KS}}[n] + E_c^{\text{KS}}[n] . \quad (4.118)$$

A number of hybrid functionals are implemented in TUNA, listed in Table 4.13. These include the popular PBE0 and B3LYP functionals, as well as the *meta*-hybrid TPSS0. The version of B3LYP used in the Gaussian program, which uses VWN-III local density correlation instead of VWN-V, can be requested by the B3LYP/G keyword.

The proportion of Hartree–Fock exchange can be changed with the HFX keyword, by stating the desired percentage of exchange. This will not change the proportion of exchange from the density functional expression, which can be changed with the DFX keyword. For example, the PBE0 functional can be built as:

```
TUNA SPE : H H 1.0 : PBE 4-31G : HFX 0.25 DFX 0.75
```

Similarly, the BHLYP functional can be specified by:

```
TUNA SPE : H H 1.0 : BLYP def2-TZVPD : HFX 0.50 DFX 0.50
```

4.2.4.4 Double-hybrid Functionals

The next rung contains “double-hybrid” functionals, which additionally incorporate a proportion, γ , of MP2 correlation.

$$E_{xc}[n] = \beta E_x^{\text{HF}} + (1 - \beta) E_x^{\text{KS}}[n] + \gamma E_c^{\text{MP2}} + (1 - \gamma) E_c^{\text{KS}}[n] \quad (4.119)$$

The MP2 part of the energy is not calculated self-consistently, but instead is applied to the converged Kohn–Sham orbitals at the end of the SCF. The unrelaxed density matrix is also implemented for double-hybrid functionals, allowing calculation of properties including natural orbitals and dipole moments.

The first double-hybrid functional developed was B2PLYP, which incorporates 53% Hartree–Fock exchange and 27% MP2 correlation. This functional can be requested by either of these two command strings.

```
TUNA SPE : Li H 1 : B2PLYP 6-311G :
```

```
TUNA SPE : Li H 1 : BLYP 6-311G : HFX 0.53 DFX 0.47 MPC 0.27 DFC 0.73
```


Table 4.13 Hybrid exchange–correlation functionals in TUNA

Functional	$E_x[n]$	$E_c[n]$	HFX / %
PBE0 [53]	PBE	PBE	25
B1LYP [61]	B88	LYP	25
BHLYP [62]	B88	LYP	50
B3LYP [63]	B88	LYP	20
B3LYP/G [63]	B88	LYP	20
B1P86 [50, 57]	B88	P86	25
B3P86 [50, 57]	B88	P86	20
PW1PW [53]	PW91	PW91	25
mPW1PW [53]	mPW91	PW91	25
mPW1LYP [53]	mPW91	LYP	25
B3PW91 [50, 52]	B88	PW91	20
TPSSH [64]	TPSS	TPSS	10
TPSS0 [64]	TPSS	TPSS	25

The MPC keyword defines the proportion of MP2 correlation, and the DFC keyword defines the proportion of density functional theory correlation. Similarly, the PBE0-DH functional (a non-empirical double-hybrid) can be built by:

```
TUNA SPE : Li H 1 : PBE 6-311G : HFX 0.50 DFX 0.50 MPC 0.125 DFC 0.875
```

A number of double-hybrids are implemented in TUNA, listed in Table 4.14. These include non-empirical double-hybrids like PBE0-DH, PBE0-2 and PBE-QIDH, as well as empirical double-hybrids including B2PLYP, mPW2PLYP and DSD-BLYP. The latter functional also incorporates spin-component scaling of the MP2 correlation part. The proportion of same- and opposite-spin scaling can be changed with the SSS and OSS keywords, as discussed in section 4.2.2.

The values of HFX, DFX, MPC and DFC are printed at the start of a hybrid or double-hybrid DFT calculation's SCF cycle. The values of OSS and SSS are printed during the MP2 calculation for a spin-scaled double-hybrid.

Table 4.14 Double-hybrid exchange–correlation functionals in TUNA

Functional	$E_x[n]$	$E_c[n]$	HFX / %	MPC / %
PBE0-DH [65]	PBE	PBE	50	12.5
PBE-QIDH [66]	PBE	PBE	69	33
PBE0-2 [67]	PBE	PBE	79	50
B2PLYP [68]	B88	LYP	53	27
B2K-PLYP [69]	B88	LYP	72	42
B2T-PLYP [69]	B88	LYP	60	31
B2G-PLYP [69]	B88	LYP	65	36
B2NC-PLYP [70]	B88	LYP	81	55
mPW2PLYP [71]	mPW91	LYP	55	25
DSD-BLYP [72]	B88	LYP	75	100

4.2.4.5 Numerical Quadrature

Density functional theory calculations involved numerical integrations over a grid, on which the density is expressed. The integral of the density should be the same as the number of electrons in the system, and prints for the guess density before the SCF cycle, and the final converged density. If the guess density is far from the ideal, a warning will be printed, and if it is totally wrong the calculation will crash politely.

The tightness of this grid is controlled via the INTACC keyword, which has a default value of 4.0. The extent of the grid (which may be important for large atoms) is controlled via the general integration grid keywords in Table 4.15.

Table 4.15 Integration grid tightness criteria in TUNA

Grid	Integral Accuracy	Grid Range
LOOSEGRID	3.0	0.7
MEDIUMGRID	4.0	0.9
TIGHTGRID	5.0	1.0
EXTREMEGRID	7.0	1.2

Integration on this grid is performed by numerical quadrature. The grids are atom centered, and then Becke weighting is applied to build the molecular grid [73]. The atom centered grids use Gauss–Legendre quadrature for the radial part and Lebedev quadrature for the angular part. The order of Lebedev integration is chosen by the closest Lebedev order available to match the desired integral accuracy. The following is a very high density grid, beyond that from EXTREMEGRID:

TUNA SPE : Ar : HFS 6-31G : INTACC 7 EXTREMEGRID

Becke’s heteroatomic weighting system is used, to distribute integration weights for heteronuclear systems. This relies on a proxy for atomic size, for which the van der Waals radius is used.

Both the density (and density gradient for GGA functionals) are needed to be expressed on the grid. This is done by expressing basis functions and basis function gradients on the grid. The mathematical expressions for the primitive Gaussians, weighted by their coefficients in an atomic orbital, are evaluated on the grid, centered around each atomic basis function centre.

4.2.5 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 and TUNA currently has no root following algorithm. 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.

4.2.5.1 Configuration Interaction Singles

The configuration interaction single excitations method is the analogue of Hartree–Fock for excited states [74], and therefore is 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,

$$|\Psi_{\text{CIS}}\rangle = \sum_{ia} c_i^a |\Psi_i^a\rangle . \quad (4.120)$$

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

$$H_{ia,jb} = (\varepsilon_a - \varepsilon_j) \delta_{ij} \delta_{ab} - \langle ja || ib \rangle . \quad (4.121)$$

The eigenvalues of this matrix give the excitation energies from the reference Hartree–Fock determinant and the excited states, ω . 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, μ , and uses these to find the oscillator strengths, by

$$f_{\text{osc}} = \frac{2}{3}\omega|\mu|^2. \quad (4.122)$$

These values, along with the excitation energy in eV, excitation frequency in 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

$$\Delta P_i^j = -\sum_{ab} c_i^a c_j^a. \quad (4.123)$$

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 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 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 : UCIS 6-31++G : ROOT 2 CH 1 CISTHRESH 0.1
```

4.2.5.2 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 [75]. 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 ω , consists of a “direct” contribution from the u_{ij}^{ab} tensor, which accounts for electron correlation effects of the electron involved in the excitation. The other term is “indirect”, accounting for correlation between electrons not involved in the excitation, calculated by contraction of the v_i^a tensor, which contains contributions from the MP2 amplitudes,

$$\omega_{\text{CIS(D)}} = \frac{1}{4} \sum_{ijab} \frac{(u_{ij}^{ab})^2}{\mathcal{E}_{ij}^{ab} + \omega} + \sum_{ia} c_i^a v_i^a. \quad (4.124)$$

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
```

4.3 Basis Sets and Molecular Integrals

One of the first steps in all TUNA calculations is the calculation of one- and two-electron integrals over contracted Gaussian basis functions, which are necessary to compute expectation values such as the energy. Molecular integrals in TUNA are computed with the McMurchie–Davidson scheme [76]. In this widely used method [77], recurrence relations between Hermite Gaussians simplify the calculation substantially.

The molecular integrals are stored in the `tuna_integral` module, which is written in Cython [78]. 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 so somewhat limiting how generally accessible TUNA is. Diatomic parity is leveraged in the calculation of these integrals — for a molecule aligned to the z axis, several of the two-electron integrals are necessarily zero, so their calculation can be skipped.

Starting from version 0.10.0, the integrals module has been refactored to demand the molecule aligned along the z -axis, or located at the origin for an atom. This allows all the symmetry of diatomics to be taken into account, leading to an approximately tenfold increase in speed from the previous version. A diatomic molecule can always be rotated onto the z -axis, which is the approach taken in the molecular dynamics module to allow three-dimensional rotation but energy evaluation in one dimension.

The molecular integrals implemented in the current version of TUNA are:

- Overlap integrals, $S_{\mu\nu} = \langle \varphi_\mu | \varphi_\nu \rangle$.
- Kinetic integrals, $T_{\mu\nu} = \langle \varphi_\mu | -\frac{1}{2} \nabla^2 | \varphi_\nu \rangle$.
- Dipole integrals, $D_{\mu\nu}^k = \langle \varphi_\mu | r_k | \varphi_\nu \rangle$.
- Nuclear–electron integrals, $V_{\mu\nu}^{\text{ne}} = \sum_A^{\text{nuclei}} \langle \varphi_\mu | \frac{Z_A}{r-R_A} | \varphi_\nu \rangle$.
- Two-electron integrals, $V_{\mu\kappa\nu\lambda}^{\text{ee}} = (\mu\kappa|\nu\lambda) = \langle \varphi_\mu \varphi_\kappa | \frac{1}{r} | \varphi_\nu \varphi_\lambda \rangle$.

Cartesian harmonics are used throughout the program, rather than spherical harmonics. This means that TUNA has, for example, six d orbitals rather than five, which slows calculations down but gives better answers due to the greater flexibility of a larger basis set.

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
```

Using the additional print keyword, P, will print the detailed basis data at the start of a TUNA calculation for the atoms included.

Basis options are listed in Table 4.16, and are taken from the Basis Set Exchange [79].

Table 4.16 Basis sets implemented in TUNA

Basis Set	Citation	Description
STO-2G	[80]	Minimal
STO-3G	[80]	Minimal
STO-4G	[80]	Minimal
STO-5G	[80]	Minimal
STO-6G	[80]	Minimal
3-21G	[81]	Double-zeta
4-31G	[82]	Double-zeta
6-31G	[82]	Double-zeta
6-31+G	[82]	Double-zeta, diffuse functions on heavy atoms
6-31++G	[83]	Double-zeta, diffuse functions
6-31G*	[84]	Double-zeta, polarisation functions on heavy atoms
6-31G**	[85]	Double-zeta, polarisation functions
6-31+G*	[86, 87]	Double-zeta, diffuse and polarisation functions on heavy atoms
6-31++G*	[86, 87]	Double-zeta, diffuse functions, polarisation functions on heavy atoms
6-31+G**	[86, 87]	Double-zeta, diffuse functions on heavy atoms, polarisation functions
6-31++G**	[86, 87]	Double-zeta, diffuse and polarisation functions
6-31G[2df,p]	[87–89]	Double-zeta, diffuse and polarisation functions
6-31G[3df,3pd]	[87, 88, 90]	Double-zeta, diffuse and polarisation functions
6-311G	[88]	Triple-zeta basis set
6-311+G	[83]	Triple-zeta basis set with diffuse functions on heavy atoms
6-311++G	[83]	Triple-zeta basis set with diffuse functions on all atoms
6-311G*	[88]	Triple-zeta basis set with polarisation functions on heavy atoms
6-311G**	[88]	Triple-zeta basis set with polarisation functions on all atoms

Basis Set	Citation	Description
6-311+G*	[83, 88]	Triple-zeta basis set with polarisation and diffuse functions
6-311++G*	[83, 88]	Triple-zeta basis set with polarisation and diffuse functions
6-311+G**	[83, 88]	Triple-zeta basis set with polarisation and diffuse functions
6-311++G**	[83, 88]	Triple-zeta basis set with polarisation and diffuse functions
6-311G[2df,2pd]	[88, 89]	Triple-zeta basis set with polarisation and diffuse functions
6-311+G[2d,p]	[83, 88]	Triple-zeta basis set with polarisation and diffuse functions
6-311++G[2d,2p]	[83, 88]	Triple-zeta basis set with polarisation and diffuse functions
6-311++G[3df,3pd]	[83, 88, 89]	Triple-zeta basis set with polarisation and diffuse functions
cc-pVDZ	[91, 92]	Polarised double-zeta
cc-pVTZ	[91, 92]	Polarised triple-zeta
cc-pVQZ	[91, 92]	Polarised quadruple-zeta
cc-pV5Z	[91, 92]	Polarised quintuple-zeta
cc-pV6Z	[93]	Polarised sextuple-zeta
aug-cc-pVDZ	[94]	Polarised double-zeta, diffuse functions
aug-cc-pVTZ	[94]	Polarised triple-zeta, diffuse functions
aug-cc-pVQZ	[94]	Polarised quadruple-zeta, diffuse functions
aug-cc-pV5Z	[94]	Polarised quintuple-zeta, diffuse functions
aug-cc-pV6Z	[94]	Polarised sextuple-zeta, diffuse functions
d-aug-cc-pVDZ	[94]	Polarised double-zeta, lots of diffuse functions
d-aug-cc-pVTZ	[94]	Polarised triple-zeta, lots of diffuse functions
d-aug-cc-pVQZ	[94]	Polarised quadruple-zeta, lots of diffuse functions
d-aug-cc-pV5Z	[94]	Polarised quintuple-zeta, lots of diffuse functions
d-aug-cc-pV6Z	[94]	Polarised sextuple-zeta, lots of diffuse functions
t-aug-cc-pVDZ	[94]	Polarised double-zeta, tons of diffuse functions
t-aug-cc-pVTZ	[94]	Polarised triple-zeta, tons of diffuse functions
t-aug-cc-pVQZ	[94]	Polarised quadruple-zeta, tons of diffuse functions
t-aug-cc-pV5Z	[94]	Polarised quintuple-zeta, tons of diffuse functions
t-aug-cc-pV6Z	[94]	Polarised sextuple-zeta, tons of diffuse functions

Basis Set	Citation	Description
cc-pCVDZ	[95]	Polarised double-zeta, core functions
cc-pCVTZ	[95]	Polarised triple-zeta, core functions
cc-pCVQZ	[95]	Polarised quadruple-zeta, core functions
cc-pCV5Z	[95]	Polarised quintuple-zeta, core functions
aug-cc-pCVDZ	[95]	Polarised double-zeta, core and diffuse functions
aug-cc-pCVTZ	[95]	Polarised triple-zeta, core and diffuse functions
aug-cc-pCVQZ	[95]	Polarised quadruple-zeta, core and diffuse functions
aug-cc-pCV5Z	[95]	Polarised quintuple-zeta, core and diffuse functions
cc-pwCVDZ	[96]	Polarised double-zeta, core–valence functions
cc-pwCVTZ	[96]	Polarised triple-zeta, core–valence functions
cc-pwCVQZ	[96]	Polarised quadruple-zeta, core–valence functions
cc-pwCV5Z	[96]	Polarised quintuple-zeta, core–valence functions
aug-cc-pwCVDZ	[96]	Polarised double-zeta, core–valence and diffuse functions
aug-cc-pwCVTZ	[96]	Polarised triple-zeta, core–valence and diffuse functions
aug-cc-pwCVQZ	[96]	Polarised quadruple-zeta, core–valence and diffuse functions
aug-cc-pwCV5Z	[96]	Polarised quintuple-zeta, core–valence and diffuse functions
ano-pVDZ	[97]	Polarised double-zeta, atomic natural orbitals
ano-pVTZ	[97]	Polarised triple-zeta, atomic natural orbitals
ano-pVQZ	[97]	Polarised quadruple-zeta, atomic natural orbitals
ano-pV5Z	[97]	Polarised quintuple-zeta, atomic natural orbitals
aug-ano-pVDZ	[97]	Polarised double-zeta, atomic natural orbitals, diffuse functions
aug-ano-pVTZ	[97]	Polarised triple-zeta, atomic natural orbitals, diffuse functions
aug-ano-pVQZ	[97]	Polarised quadruple-zeta, atomic natural orbitals, diffuse functions
aug-ano-pV5Z	[97]	Polarised quintuple-zeta, atomic natural orbitals, diffuse functions
pc-0	[98, 99]	Double-zeta, polarisation-consistent
pc-1	[98, 99]	Polarised double-zeta, polarisation-consistent

Basis Set	Citation	Description
pc-2	[98, 99]	Polarised triple-zeta, polarisation-consistent
pc-3	[98, 99]	Polarised quadruple-zeta, polarisation-consistent
pc-4	[98, 99]	Polarised quintuple-zeta, polarisation-consistent
aug-pc-0	[98–100]	Double-zeta, polarisation-consistent, diffuse functions
aug-pc-1	[98–100]	Polarised double-zeta, polarisation-consistent, diffuse functions
aug-pc-2	[98–100]	Polarised triple-zeta, polarisation-consistent, diffuse functions
aug-pc-3	[98–100]	Polarised quadruple-zeta, polarisation-consistent, diffuse functions
aug-pc-4	[98–100]	Polarised quintuple-zeta, polarisation-consistent, diffuse functions
pcseg-0	[101]	Double-zeta, polarisation-consistent
pcseg-1	[101]	Polarised double-zeta, polarisation-consistent
pcseg-2	[101]	Polarised triple-zeta, polarisation-consistent
pcseg-3	[101]	Polarised quadruple-zeta, polarisation-consistent
pcseg-4	[101]	Polarised quintuple-zeta, polarisation-consistent
aug-pcseg-0	[101]	Double-zeta, polarisation-consistent, diffuse functions
aug-pcseg-1	[101]	Polarised double-zeta, polarisation-consistent, diffuse functions
aug-pcseg-2	[101]	Polarised triple-zeta, polarisation-consistent, diffuse functions
aug-pcseg-3	[101]	Polarised quadruple-zeta, polarisation-consistent, diffuse functions
aug-pcseg-4	[101]	Polarised quintuple-zeta, polarisation-consistent, diffuse functions
def2-SVP	[102]	Polarised double-zeta
def2-SVPD	[102, 103]	Polarised double-zeta, diffuse functions
def2-TZVP	[102]	Polarised triple-zeta

Basis Set	Citation	Description
def2-TZVPD	[102, 103]	Polarised triple-zeta, diffuse functions
def2-TZVPP	[102, 103]	Very polarised triple-zeta
def2-TZVPPD	[102, 103]	Very polarised triple-zeta, diffuse functions
def2-QZVP	[102]	Polarised quadruple-zeta
def2-QZVPD	[102, 103]	Polarised quadruple-zeta, diffuse functions
def2-QZVPP	[102, 103]	Very polarised quadruple-zeta
def2-QZVPPD	[102, 103]	Very polarised quadruple-zeta, diffuse functions

4.3.1 Custom Basis Sets

If you want a basis set which is not implemented natively in TUNA, a custom basis set can be read in. This is done by setting the basis type to `CUSTOM` and using the `BASIS [filepath.tuna]` keyword, where `filepath.tuna` is the path to the custom basis file.

For example, reading in basis data from "basis-data.tuna" in the same directory as the terminal can be done by:

```
TUNA SPE : H H 1.0 : HF CUSTOM : BASIS BASIS-DATA.TUNA
```

Note that any file path typed in the terminal will be converted to *lower* case. So you need to have a fully lower case basis data file name.

The `.tuna` file type stores basis set data, and has essentially the same form of the `.orca` file type that can be downloaded from the Basis Set Exchange [79] — so `.orca` basis data files can be used directly. In this file, lines to skip begin with "\$", then elements are spelled out with their full names. Underneath this is the type of subshell (S, P, D, etc.) with the number of primitive Gaussians for that subshell. Next is a list of primitive Gaussians in three columns: the primitive number (1, 2, 3, etc.), the exponent and the coefficient. This process repeats for the different

subshells in the element type.

An example of this file type is shown, for the cc-pVDZ basis set for hydrogen and oxygen:

\$DATA

HYDROGEN

S 4

1	1.301000E+01	1.968500E-02
2	1.962000E+00	1.379770E-01
3	4.446000E-01	4.781480E-01
4	1.220000E-01	5.012400E-01

S 1

1	1.220000E-01	1.000000E+00
---	--------------	--------------

P 1

1	7.270000E-01	1.000000E+00
---	--------------	--------------

OXYGEN

S 9

1	1.172000E+04	7.100000E-04
2	1.759000E+03	5.470000E-03
3	4.008000E+02	2.783700E-02
4	1.137000E+02	1.048000E-01
5	3.703000E+01	2.830620E-01
6	1.327000E+01	4.487190E-01
7	5.025000E+00	2.709520E-01
8	1.013000E+00	1.545800E-02
9	3.023000E-01	-2.585000E-03

S 9

1	1.172000E+04	-1.600000E-04
2	1.759000E+03	-1.263000E-03
3	4.008000E+02	-6.267000E-03
4	1.137000E+02	-2.571600E-02
5	3.703000E+01	-7.092400E-02
6	1.327000E+01	-1.654110E-01
7	5.025000E+00	-1.169550E-01
8	1.013000E+00	5.573680E-01
9	3.023000E-01	5.727590E-01

S 1

1	3.023000E-01	1.000000E+00
---	--------------	--------------

P 4

```

1      1.770000E+01      4.301800E-02
2      3.854000E+00      2.289130E-01
3      1.046000E+00      5.087280E-01
4      2.753000E-01      4.605310E-01
P      1
1      2.753000E-01      1.000000E+00
D      1
1      1.185000E+00      1.000000E+00

```

\$END

4.3.2 Basis Set Extrapolation

Calculations that approach the complete basis set limit, with quadruple-zeta or higher basis sets, are very computationally expensive. In TUNA, two calculations can be run — for example with double- and triple-zeta basis sets — and the results from those extrapolated to give an approximation to the complete basis set limit. Because derivatives of the energy in TUNA are numerical, a complete basis set extrapolated energy can be carried forwards to give an extrapolated geometry, MD calculation or frequency. Extrapolation can be done starting with double-, triple- and quadruple-zeta basis sets.

Requesting a lower-zeta basis set and using the EXTRAPOLATE keyword tells TUNA to run two calculations — first with the lower-zeta basis, then the higher-zeta, then extrapolating those energies.

The convergence of the Hartree–Fock energy to the basis set limit [97] is expected to be

$$E_X^{\text{SCF}} = E_\infty^{\text{SCF}} + A \exp(-\alpha\sqrt{X}) , \quad (4.125)$$

where A and α are constants that depend on the basis set.

which can be rearranged, with X being the lower cardinal number and Y the higher, to an equation for the complete basis set limit

$$E_\infty^{\text{SCF}} = E_X^{\text{SCF}} + \frac{E_Y^{\text{SCF}} - E_X^{\text{SCF}}}{1 - \exp[\alpha(\sqrt{X} - \sqrt{Y})]} \quad (4.126)$$

The correlation energy is supposed to converge as

$$E_{\infty}^{\text{corr}} = \frac{X^{\beta} E_X^{\text{corr}} - Y^{\beta} E_Y^{\text{corr}}}{X^{\beta} - Y^{\beta}} \quad (4.127)$$

The values of α and β are dependent on the basis set. For double-zeta basis sets, β is always taken as 2.4, and for triple- and quadruple-zeta bases β is 3.0. The values of α are listed in [97] for double- and triple-zeta basis sets. The α value for quadruple-zeta sets is assumed to be the same as the value for triple-zeta sets.

An example of a single point calculation on a sodium atom with extrapolation with the cc-pVDZ and cc-pVTZ basis sets is requested by:

```
TUNA SPE : Na : MP2 cc-pVDZ : EXTRAPOLATE
```

And an example of an extrapolated optimisation and harmonic frequency calculation using def2-TZVPPD and def2-QZVPPD is:

```
TUNA OPTFREQ : H H 0.74 : CCSD def2-TZVPPD : EXTRAPOLATE
```

Finally, a CCSD energy evaluation of the helium atom with quadruple-to-quintuple-zeta basis set extrapolation:

```
TUNA SPE : He : CCSD aug-cc-pVQZ : EXTRAPOLATE
```

5 Performance of TUNA

Python is extremely slow. However, TUNA escapes this fate by having very few loops in pure Python — the computationally intensive elements of calculations are array manipulations, handled efficiently with NumPy's [6] vectorised operations. This makes TUNA surprisingly quick. An optimisation and harmonic frequency calculation with B3LYP/cc-pVTZ on a dihydrogen molecule takes about 13 seconds in TUNA and 3 minutes in ORCA, on a single core of the same device. However, the latter fish, of course, can be easily parallelised.

However, moving to larger basis sets, the story changes — ORCA becomes much *faster* than TUNA, like in real life. This is because of the only computationally intensive part of TUNA which is difficult to vectorise — the molecular integrals. Table 5.1 shows the relative speeds of ORCA and TUNA on a CCSD optimisation. For small basis sets, TUNA is the winner by far, but once there over a hundred basis functions ORCA dominates.

To overcome their slowness, the molecular integrals are written in Cython [78] (a compiled version of Python) which alleviates the problem for low angular momentum basis functions. However integrals over d, f and higher functions, which use recursion in the McMurchie–Davidson scheme, are still extremely slow. At the moment, TUNA is therefore very well suited to performing correlated calculations with small-to-medium sized basis sets. The routine use of larger basis sets will require a substantially rewritten molecular integrals module, which I do not want to do soon. It is possible to vectorise the integrals, to some extent, as has been done in ORCA's SHARK package [77]. Another reason that TUNA is slow for large basis set is due to using Cartesian harmonics instead of spherical harmonics, and not removing linearly dependent basis functions.

Table 5.1 Calculation times for a CCSD optimisation and frequency calculation on the hydrogen molecule on a single core, with extreme SCF convergence criteria

Basis Set	ORCA	TUNA
cc-pVDZ	97.2 s	0.4 s
cc-pVTZ	108.1 s	1.6 s
cc-pVQZ	129.3 s	30.0 s
cc-pV5Z	~6 min	> 1 hr

The use of Numpy's "einsum" function is ubiquitous throughout the TUNA code. This makes the implementation of tensor style equations with lots of indices very straightforward — increasing readability of the code, and dramatically increasing speed and reducing memory usage in some cases. For example, the transformation of the two-electron integrals into the molecular orbital basis is

$$(ij|ab) = \sum_{pqrs} C_p^i C_q^j (pq|rs) C_r^a C_s^b . \quad (5.1)$$

The implementation of this in pure Python would require at least four "for" loops – one for each summation index. However, with einsum the code practically writes itself:

```
ERI_MO = np.einsum("pi,qj,pqrs,ra,sb", C, C, ERI_AO, C, C, optimize=True)
```

This reduces the number of lines of code to one, but *increases* code readability!

The "optimize=True" statement tells Numpy to find the most efficient "path" to perform this contraction — it turns out that this is four $\mathcal{O}(N^5)$ steps, contracting one molecular orbital with the atomic orbital integrals at a time, and saving the resulting temporary array.

Complicated tensorial equations — as are found within coupled cluster theory — can therefore be efficiently and straight-forwardly implemented via einsum.

6 Acknowledgements

There are a variety of codes and tutorials that have served as valuable inspiration for TUNA.

The first is the Crawford group programming projects for computational chemistry which contain examples for wavefunction methods such as coupled cluster theory and MP2. The “Psi4Numpy” [\[2\]](#) collection of codes and tutorials has a diverse group of methods which were extremely useful.

The collection of Python codes of wavefunction methods based on Hartree–Fock theory, HarPy, was particularly useful for the CCSD and CCSDT code in TUNA. Finally, PyDFT, a simple educational DFT code that shows how basic exchange–correlation functionals can be implemented, was an extremely useful reference, as was its accompanying documentation.

I am very grateful to Hannah Whittome for, among other things, her brutal criticism of an early version of the TUNA logo — this led to the fish being put on an extreme weight loss regime.

7 List of Keywords

Table 7.1 lists all of the mandatory and optional keywords in TUNA, grouped by their functions.

Table 7.1 Parameter keywords in TUNA

Parameter	Description
P	Print more information to the terminal
T	Print less information to the terminal
DEBUG	Print extreme information to the terminal
CH [int] or CHARGE [int]	Molecular charge
ML [int] or MULTIPLICITY [int]	Molecular multiplicity
BASIS [string]	Custom basis filepath
EXTRAPOLATE	Extrapolate double-zeta basis
DECONTRACT	Decontract basis set
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
COREGUESS	Core Hamiltonian guess density
SADGUESS	Superposition of atomic densities guess density
SCFGUESS	Minimal self-consistent guess density
LOOSE or LOOSESCF	Loose SCF convergence
MEDIUM or MEDIUMSCF	Normal SCF convergence
TIGHT or TIGHTSCF	Tight SCF convergence

Parameter	Description
EXTREME or EXTREMESCF	Extreme SCF convergence
DAMP [float]	Static damping for SCF convergence
NODAMP	No damping for SCF convergence
MAXDAMP [float]	Maximum damping factor
DIIS [int]	Fock matrix extrapolation, and number of matrices
NODIIS	No Fock matrix extrapolation
SLOWCONV	High static damping (50%) for SCF convergence
VERYSLOWCONV	Extreme static damping (85%) for SCF convergence
MAXITER [int]	Maximum number of SCF iterations
LOOSEOPT	Loose geometry convergence
MEDIUMLOPT	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
ANHARMCONV [float]	Convergence criteria for anharmonic frequencies
NELEC [int]	Number of electrons for IP/EA calculation
VERTICAL	Vertical calculation for IP or EA

Parameter	Description
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
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
FREEZECORE [int]	Freeze core orbitals
SSS [float]	Same-spin scaling for SCS-MP2
OSS [float]	Opposite-spin scaling for SCS-MP2
MP3S [float]	Scaling for SCS-MP3
MPCONV [float]	Convergence criteria for MP2
MPMAXITER [int]	Maximum iterations for MP2
MPGRID [int]	Grid size for Laplace transform MP2
AMPCONV [float]	Convergence criteria for coupled cluster amplitudes

Parameter	Description
CCCONV [float]	Convergence criteria for coupled cluster energy
CCMAXITER [int]	Maximum iterations for coupled cluster
CCDAMP [float]	Use damping in coupled cluster, parameter
PRINTAMPS [int]	Print this number of largest t -amplitudes
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
LOOSEGRID	Loose DFT integration grid
MEDIUMGRID	Normal DFT integration grid
TIGHTGRID	Tight DFT integration grid
EXTREMEGRID	Extreme DFT integration grid
INTACC [float]	Integral accuracy for DFT
DFX [float]	Density functional theory exchange percentage
HFX [float]	Hartree–Fock theory exchange percentage
DFC [float]	Density functional theory correlation percentage
MPC [float]	Perturbation theory correlation percentage
NOX	Turn off DFT exchange
NOC	Turn off DFT correlation
XA [float]	Value of α in $X\alpha$ LDA exchange
DENSLOT	Plot the electron density
SPINDENSLOT	Plot the spin density
DIFFDENSLOT	Plot the difference electron density
DIFFSPINDENSLOT	Plot the difference spin density

Parameter	Description
PLOTMO [int]	Plot a molecular orbital
PLOTNO [int]	Plot a natural orbital
PLOTHOMO	Plot the highest occupied molecular orbital
PLOTLUMO	Plot the lowest unoccupied molecular orbital
PLOTVIB	Plot nuclear wavefunctions
EX [float]	Electric field in x direction
EY [float]	Electric field in y direction
EZ [float]	Electric field in z direction
DIPOLE	Calculate numerical dipole moment
POLAR	Calculate numerical polarisability
HYPER	Calculate numerical hyperpolarisability

8 Detailed Changelog

8.1 TUNA 0.1.0

Added

- Restricted Hartree–Fock
- Single point energy and coordinate scans
- New basis sets: ST0-3G, ST0-6G, 6-31G, 6-311G, 6-311++G
- Dynamic damping and level shift
- Ghost atoms
- Molecular orbitals and energies, Koopmans' theorem parameters
- Electron density 3D plots
- Dispersion correction with semi-empirical D2 scheme
- Convergence criteria keywords for SCF
- Interface with matplotlib for coordinate scan via SCANPLOT keyword

8.2 TUNA 0.2.0

Added

- Conventional and spin-component-scaled MP2
- Mulliken and Löwdin population analysis
- Keywords for additional print, P, and SCF damping, DAMP
- Identification of point group

Changed

- Updated to Python 3.12
- Significantly increased integral efficiency using vectorised operations

8.3 TUNA 0.3.0

Added

- Geometry optimisations
- Harmonic frequencies, optionally linked with prior optimisation with OPTFREQ calculation type
- Rotational constants
- Nuclear dipole moment
- Optional exact or approximate (default) Hessian for optimisation
- Keywords for geometry convergence tolerance and maximum iterations
- High static damping option for difficult SCF convergence cases, SLOWCONV

8.4 TUNA 0.4.0

Added

- Fock matrix extrapolation for SCF convergence (DIIS)
- Electronic and total dipole moment
- Unrelaxed MP2 density and natural orbitals
- Thermochemistry after frequency calculations, TEMPERATURE and PRESSURE keywords
- New 3-21G basis set

Changed

- Density matrix is now read by default from previous step in coordinate scans and optimisations

Fixed

- Unbroke level shift; added keywords

8.5 TUNA 0.5.0

Added

- *Ab initio* molecular dynamics
- Unrestricted Hartree-Fock energy and density
- Unrestricted MP2 energies
- Restricted and unrestricted MP3 energies
- Spin-component-scaled MP3 energies
- Keyword to decontract basis functions, DECONTRACT
- New basis sets: 4-31G, 6-31+G, 6-31++G, and 6-311+G
- Mayer bond order, free and total valences
- Spin contamination for UHF calculations
- Orbital rotation and ROTATE and NOROTATE keywords for UHF guess density
- Optimisations and molecular dynamics simulations optionally print to XYZ file with TRAJ and NOTRAJ keywords
- Option to optimize to a maximum with OPTMAX keyword
- Terminal output now has colour for warning and errors
- Increased speed of all TUNA calculations by 50–95% through making full use of permutational symmetry in the two-electron integrals
- Much better error handling and clear errors and warnings
- New changelog, manual, GitHub and PyPI pages
- TUNA can now be installed simply by `pip install QuantumTUNA`

Changed

- Rewrote all the code to make things object-oriented, improve efficiency and reduce redundancy
- Slimmed down the fish logo :(
- Optimised and simplified integral engine
- Reduced default SCF and optimisation convergence criteria by fixing associated bug
- Better handling of print levels; optimizations now only calculate properties at the end by default
- Now use more energy evaluations for gradients and Hessians, making them more robust but slower
- Generally refined the output, making information more precise and clear

Fixed

- When its equations can't be solved, DIIS now resets instead of crashing the program
- Fixed frequency calculations being far too sensitive to SCF convergence when guess density was read in
- SCF convergence was checking that ΔE was less than the criteria, rather than its magnitude leading to too early convergence
- Fixed the thermochemistry module mixing up the temperature and pressure variables
- Formatting issues with population analysis
- Fixed handling of ghost atoms, accessible by XH or XHe

8.5.1 TUNA 0.5.1

Added

- Keyword for the default Hessian used in geometry optimisations, DEFAULTHESS
- Keyword for the maximum step size for geometry optimisations, MAXSTEP
- Optional parameter can be used with the LEVELSHIFT keyword to adjust the degree of level shift

Changed

- Improved logging to console for a more consistent user experience
- Molecular dynamics simulations now read in the density matrix from the previous step by default
- Refactored and optimised all the code, ready for future updates and added comments and docstrings
- Separated two- and three-dimensional plotting functions into new `tuna_plot` module
- Keyword for the angle with which to rotate orbitals for initial guess, THETA has been removed and replaced by ROTATE [Angle]
- Reading in orbitals from previous coordinate scan step now turned off by default for UHF calculations

Fixed

- Point group and molecular structure are now detected correctly for ghost atom calculations
- Nuclear repulsion energy is now not calculated for ghost atoms

- Individual energy components now work correctly for UHF energies
- Removed Koopmans' theorem parameter calculations for UHF references
- Molecular orbital eigenvalues and coefficients are now printed correctly for UHF, split into alpha and beta orbitals
- Stopped reading in orbitals for one-electron systems in coordinate scans, as there is no SCF cycle
- Orbitals are now rotated correctly by the requested angle for UHF initial guesses
- DIIS for UHF now works as intended when the equations can't be solved

8.6 TUNA 0.6.0

Added

- Transition intensities for harmonic frequency calculations
- Excited state energy and density by configuration interaction singles, CIS
- Perturbative doubles correction to the CIS excitation energy with CIS(D) via CIS[D] keyword
- Excited state coordinate scans, geometry optimisations, harmonic frequencies and MD simulations
- Orbital-optimised MP2 energy and density, OMP2
- Support for unrestricted references for SCS-MP2, SCS-MP3 and OMP2
- Unrelaxed density matrix for unrestricted MP2, SCS-MP2 and OMP2
- Keywords for same-spin, opposite-spin and MP3 scaling for SCS-MP3: SSS, OSS and MP3S
- Keywords for orbital-optimised MP2 convergence criteria and maximum iterations: OMP2CONV and OMP2MAXITER
- Keywords for state of interest in CIS, threshold for printing contributions, and number of states to print: ROOT, CISTHRESH and NSTATES
- Optional spin contamination calculation for MP2 calculations
- Optional population analysis and dipole moment calculations using CIS unrelaxed density matrix
- Faster one- and two-electron integrals

Changed

- Rotational constant is now printed in both GHz and cm^{-1}

- Molecular information at the beginning of a calculation now prints number of alpha and beta electrons
- Reorganised and improved prettiness of console log
- Maximum SCF iterations is now 100 by default, instead of 50
- Various and widespread low level optimisations, and all code is now fully documented
- Keywords NORMALSCF and NORMALOPT for SCF and geometry convergence replaced by MEDIUMSCF and MEDIUMOPT
- Default geometry convergence criteria set to MEDIUMOPT rather than TIGHTOPT by default, except for OPTFREQ calculations
- Absorbed tuna_dispersion module into tuna_energy, and added new tuna_ci module for current and future spin orbital calculations
- Used more colour in the console log

Fixed

- Density matrix is now read in from previous optimisation step, except when initial guess orbitals are rotated, as previously intended
- Unrestricted MP2 was not working correctly for esoteric charge and multiplicity combinations
- Absolute change in density matrix is now checked, rather than signed change, for SCF convergence
- DIIS now works much more reliably for UHF, by combining the alpha and beta error vectors; converges faster for both RHF and UHF
- One extra SCF cycle is no longer undertaken for no reason
- Various miscellaneous bug fixes and improvements to error handling

8.6.1 TUNA 0.6.1

Added

- Keyword for plotting the spin density, SPINDENSLOT
- Virial ratio is calculated and printed, which indicates the proximity to an optimised geometry
- Degenerate excited states are now grouped and averaged before printing
- The singlet or triplet character of excited states is now printed for RHF references

Changed

- The default SCF convergence criteria for CIS calculations is now TIGHT
- Threshold for CIS contributions decreased from 5% to 1%
- Removed printing weights for RHF references, as these are calculated in a spin-orbital basis

Fixed

- Requested orbital rotation with a tiny basis no longer causes a crash
- Electron affinity calculation was crashing when no virtual orbitals were present
- Excited state calculations were crashing when no virtual orbitals were present
- Spin density matrix for one-electron systems was calculated incorrectly
- Error handling for non-existent root in CIS calculations

8.7 TUNA 0.7.0

Added

- Energy and linearised density from CEPA0, LCCD, LCCSD, CCD, CCSD, CCSD(T) and CCSDT
- Convergence control for coupled cluster calculations with damping via CCDAMP keyword (default off)
- Convergence control for coupled cluster calculations and DIIS via DIIS keyword (default on)
- Support for atoms beyond the first period — atoms up to argon are now implemented
- Cython-based Gaussian integral engine via the McMurchie-Davidson algorithm allows higher angular momentum orbitals
- New basis sets: 6-31G*, 6-31G**, 6-311G*, 6-311G**, cc-pVDZ, cc-pVTZ, cc-pVQZ, cc-pV5Z, cc-pV6Z
- Largest coupled cluster amplitudes are found and printed with the T1 diagnostic
- Keyword CCCONV for coupled cluster convergence control, CCMAXITER for maximum iterations
- Keyword NOSINGLES to turn off single excitations in coupled cluster
- Population analysis, dipole moment, natural orbitals using coupled cluster linearised density matrices
- Keyword for changing the number of DIIS matrices to store (for SCF and coupled cluster), DIIS [Num. Matrices]

- Keywords to turn on and off natural orbital calculations, NATORBS and NONATORBS
- Optional natural orbital calculation for UHF
- Keywords for custom atomic masses in AMU, M1 and M2
- The overlap matrix eigenvalues are now checked; use the STHRESH keyword to change the threshold
- Warning message for small overlap matrix eigenvalues that are close to the threshold
- Redesigned plotting functions in TUNA; various methods can be plotted on the same axes with the ADDPLOT keyword
- Keywords SAVEPLOT and DELPLOT to save and delete plots
- Dashed lines on plot via DASH and dotted lines via DOT
- Option to change colour in SCANPLOT with RED, BLUE, GREEN, CYAN, MAGENTA, YELLOW, BLACK, WHITE keywords

Changed

- By default, DIIS now remembers 6 matrices instead of 10
- The number of MD steps, MDNUMBER or NUM, now defaults to 50 rather than 30
- Orbital-optimised MP2 can now be accessed via (U)OOMP2 in addition to (U)OMP2
- TUNA now prints the number of minutes, in addition to the number of seconds, for long calculations
- Removed DENSXPLOT and SPINDENSXPLOT keywords, and the ability to generate a three-dimensional electron density plot
- Default atomic masses slightly altered
- Calculations with zero electrons now exit with error
- Molecule and basis information now includes number of occupied and virtual orbitals
- Condensed additional printing of molecular orbitals and eigenvalues for UHF
- TUNA now exits gracefully if interrupted by the user with Ctrl+C
- New tuna_molecule and tuna_cc modules; removed tuna_plot module; renamed tuna_mpn into tuna_mp
- Information is now printed during the spin-orbital transformation for post-Hartree-Fock
- Keywords SCANUMBER and MDNUMBER replaced by simpler keyword NUM
- Keywords SCANSTEP and TIMESTEP replaced by simpler keyword STEP

- Major code improvements and restructuring

Fixed

- Fixed error handling of requesting RHF calculation on a molecule with an odd number of electrons
- You can no longer put more electrons in a molecule than the number of spin-orbitals
- Incorrect unit conversions in SCANSTEP
- Incorrect unit conversions in SCANPLOT
- The GEOMMAXITER keyword was not working correctly
- The SCF convergence keyword was being overridden if an optimisation is requested
- The geometry convergence keyword was being overridden if a subsequent frequency calculation is requested

8.8 TUNA 0.8.0

Added

- Restricted MP4, MP4(SDQ) and MP4(DQ) energy
- Spatial orbital (SCS-)MP2 energy and density and (SCS-)MP3 energy, speeds up calculations on RHF references
- Core electrons can now be frozen for correlated calculations using the FREEZECORE keyword, with optional number of orbitals to freeze
- Amplitude convergence for coupled cluster calculations via AMPCONV keyword
- Improved (non-arbitrary) dynamic damping scheme, now uses Zerner–Hehenberger scheme based on Mulliken gross populations
- New DAMP keyword, followed by a number between 0 and 1 to define a static damping parameter
- Custom basis sets using CUSTOM basis with BASIS [filename.tuna] keyword
- Extrapolation to complete basis set limit compatible with all calculations via EXTRAPOLATE keyword
- New basis sets: STO-2G, STO-4G, STO-5G, 6-31+G*, 6-31++G*, 6-31+G**, 6-31++G**, 6-311+G*, 6-311++G*, 6-311+G**, 6-311++G**
- New basis sets: def2-SVP, def2-SVPD, def2-TZVP, def2-TZVPD, def2-TZVPP, def2-TZVPPD, def2-QZVP, def2-QZVPD, def2-QZVPP, def2-QZVPPD

- New basis sets: 6-31G(d,p), 6-31G(2df,p), 6-31G(3df,3pd), 6-311G(d,p), 6-311G(2df,2pd), 6-311++G(2d,p), 6-311++G(2d,2p), 6-311++G(3df,3pd)
- New basis sets: pc-0, pc-1, pc-2, pc-3, pc-4, aug-pc-0, aug-pc-1, aug-pc-2, aug-pc-3, aug-pc-4
- New basis sets: pcseg-0, pcseg-1, pcseg-2, pcseg-3, pcseg-4, aug-pcseg-0, aug-pcseg-1, aug-pcseg-2, aug-pcseg-3, aug-pcseg-4
- New basis sets: aug-cc-pVDZ, aug-cc-pVTZ, aug-cc-pVQZ, aug-cc-pV5Z, aug-cc-pV6Z, d-aug-cc-pVDZ, d-aug-cc-pVTZ, d-aug-cc-pVQZ, d-aug-cc-pV5Z, d-aug-cc-pV6Z
- New basis sets: cc-pCVDZ, cc-pCVTZ, cc-pCVQZ, cc-pCV5Z, aug-cc-pCVDZ, aug-cc-pCVTZ, aug-cc-pCVQZ, aug-cc-pCV5Z
- New basis sets: cc-pwCVDZ, cc-pwCVTZ, cc-pwCVQZ, cc-pwCV5Z, aug-cc-pwCVDZ, aug-cc-pwCVTZ, aug-cc-pwCVQZ, aug-cc-pwCV5Z
- New basis sets: ano-pVDZ, ano-pVTZ, ano-pVQZ, ano-pV5Z, aug-ano-pVDZ, aug-ano-pVTZ, aug-ano-pVQZ, aug-ano-pV5Z
- New basis sets: 6-31G(d), 6-31G(d,p), 6-31+G(d), 6-31+G(d,p), 6-31++G(d), 6-31++G(d,p), 6-311G(d), 6-311G(d,p), 6-311+G(d), 6-311+G(d,p), 6-311++G(d), 6-311++G(d,p)

Changed

- Dunning basis sets have been rotated and had redundant functions removed, increasing computational efficiency
- Decimals points now align nicely for all calculations with all elements
- Removed repeated logging of transforming integrals in MP2
- Nicer logging at the start of a geometry optimisation
- Basis information is now printed with the additional print keyword, P
- Design of logging is now more consistent
- Increased default orbital-optimised MP2 iterations to 30 (originally 20)
- Improved PyPI page and README

Fixed

- Distributions for MacOS and Linux now work correctly
- Python 3.13 and Python 3.14 are now supported in addition to Python 3.12
- Frequency calculations on a fixed geometry were not working at all

- Ghost atoms were not working correctly under some circumstances
- Convergence acceleration logging was sometimes contradictory

8.8.1 TUNA 0.8.1

Added

- Spin-adapted closed shell LCCD, LCCSD, CCD, CCSD and CCSD(T), speeding up correlated calculations on RHF references
- Spin-unrestricted and spin-adapted QCISD and QCISD(T)
- The additional print keyword, 'P', now gives a more detailed breakdown of time taken throughout a calculation
- Choose the number of the largest amplitudes to print with the PRINTAMPS keyword
- Significant speed improvements using new methods to find largest amplitudes in coupled cluster calculations
- Dynamic damping factor will now now exceed 70% by default, changeable with MAXDAMP keyword

Changed

- Dynamic damping now only starts after the second SCF step, to prevent issues with rotated guesses and overdamping
- For frequency calculations, the default SCF convergence is now EXTREME to prevent numerical issues
- Optimised and refactored coupled cluster module
- Improved formatting in output for consistency
- An error is now given before the SCF calculation if a correlated calculation is not possible

Fixed

- The disconnected doubles contribution to the coupled cluster energy was sometimes calculated incorrectly
- During geometry optimisation, the current bond length was not printing correctly
- Suppressed warnings for Matplotlib not being able to find a particular font
- Matplotlib graphs where ADDPLOT is used successively no longer get exponentially bigger on MacOS

- The TUNA manual is now distributed using PyPI
- Improved PyPI page formatting

8.9 TUNA 0.9.0

Added

- Density functional theory energy and density including (*meta*-)GGA, hybrid and double-hybrid exchange-correlation functionals
- Local density methods including SVWN3, SVWN5, SPW and HFS and XA keyword for X-alpha method
- (*Meta*-)Generalised gradient methods including PBE, BLYP, BP86, mPWLYP, PWP and TPSS
- Hybrid exchange-correlation functionals including PBE0, B3LYP, BHLYP, TPSS0, mPW3LYP and B3P86
- Double-hybrid exchange-correlation functionals including PBE0-DH, PBE0-QIDH, B2PLYP, B2NC-PLYP, mPW2PLYP and DSD-BLYP
- Keywords for proportion of DFT and Hartree-Fock exchange, DFX and HFX
- Keywords for proportion of DFT and MP2 correlation, DFC and MPC
- Turn off correlation and exchange via NOC and NOX keywords
- Change DFT integration grid tightness with LOOSEGRID, MEDIUMGRID, TIGHTGRID and EXTREMEGRID keywords
- Fine control over integration accuracy with INTACC keyword
- Iterative MP2 theory for calculations on non-canonical molecular orbitals, control with MPMAXITER and MPCONV keywords
- Numerical Laplace transform MP2 theory with controlling MPGRID keyword
- New output module, plotting orbitals, densities and spin densities
- Plot molecular orbitals with PLOTMO, PLOTHOMO and PLOTLUMO keywords
- Plot natural orbitals with PLOTNO keyword
- Plot density, spin density and difference densities with DENSLOT, SPINDENSLOT, DIFFDENSLOT, DIFFSPINDENSLOT keywords
- Hartree method with no Fock exchange, H
- Force calculation type, FORCE

Changed

- Increased two-electron integral speed by 2–10x through leveraging diatomic symmetry and code optimisations
- Keyword OMP2MAXITER is now MPMAXITER
- Keyword OMP2CONV is now MPCONV
- Major changes to SCF module, more robust convergence and stability
- Removed level shift
- Updated manual with acknowledgements and further detail

Fixed

- First SCF guess cycle energy now prints properly
- The MAXITER keyword was off by one
- Freezing core electrons was broken for unrestricted MP2
- Rotating guess orbitals was on by default for triplets
- Calling an atom "X" broke everything

8.10 TUNA 0.10.0

Added

- Spin-restricted CCSDT, CCSDT(Q) and CCSDTQ energy
- Anharmonic vibrational frequencies by solving the nuclear Schrodinger equation on the full PES via the ANHARM calculation type
- Anharmonic transition intensities using either analytical or numerical dipole moment
- Plot vibrational wavefunctions with PLOTVIB, control transition energy convergence with ANHARMCONV
- Numerical calculation of dipole moments for all electronic structure methods via DIPOLE
- Numerical calculation of polarisability and hyperpolarisability via POLAR and HYPER
- Calculations in applied electric field with EX, EY, and EZ keywords
- Adiabatic (or VERTICAL) ionisation potential and electron affinity calculation types, IP and EA for ionisation with NELEC electrons
- Superposition of atomic densities and self-consistent minimal basis guess strategies, making SCF up to 50% faster

- Keywords for core Hamiltonian guess, COREGUESS, superposition guess, SADGUESS or self-consistent guess, SCFGUESS
- Electronic contribution to entropy is now calculated
- New DEBUG keyword, for extreme levels of printing
- Took advantage of more diatomic symmetry in the molecular integrals, making calculations up to 10x faster
- New D2 dispersion parameters for DFT methods including PBE, BLYP, B3LYP, BP86, TPSS and B2PLYP
- Basis set extrapolation with triple/quadruple-zeta basis sets, and quadruple/quintuple-zeta basis sets
- New triply-augmented basis sets, t-aug-cc-pVDZ, t-aug-cc-pVTZ, t-aug-cc-pVQZ, t-aug-cc-pV5Z, and t-aug-cc-pV6Z

Changed

- Default SCF guess strategy is now a self-consistent minimal basis guess, via superposition of atomic densities
- Restructured code for Hessian evaluation, reducing cost by 20% with identical results
- Density matrix idempotency is now forced after the guess density, to increase stability
- Natural orbitals are no longer calculated by default for MP2
- Minimum requestable bond length decreased from 0.05 angstroms to 0.01 angstroms
- Atomic masses can now be printed in frequency calculations with additional print, P
- Coupled cluster maximum iterations default increased from 30 to 50
- A coupled cluster damping parameter now must be given when CCDAMP is used
- The input for the HFX, DFX, MPC and DFC keywords are now given as proportions, not percentages
- The exponent for correlated basis extrapolation is now always the theoretical value
- Trying to calculate a correlated method on a system with insufficient electrons no longer throws an error
- You can now request same and opposite spin scaling on all double-hybrid functionals with SSS and OSS
- Major refactors to code, including new modules written with type hints, made code more object-oriented

Fixed

- Only some molecular orbital information was printing with RHF/RKS calculations
- The unrestricted CCSDT energy was not including the disconnected doubles contribution
- Small basis sets would sometimes cause a crash for large diatomics
- PBE was giving the wrong energy for the hydrogen atom due to issues with cleaning the spin density
- Spin-component scaling keywords were not working for double-hybrid functionals
- Natural orbitals were not being transformed back to the atomic orbital basis correctly
- The NONATORBS keyword was causing crashes with MP2
- The CEPA method keywords were not working correctly
- Negative spin densities were not being plotted with SPINDENSLOT

8.11 Next Steps

A number of features are planned for TUNA 0.11.0 — the main priority will be improving the CIS module, including spin-adaptation and Davidson–Liu diagonalisation for speed. The configuration interaction capabilities of TUNA are set to expand, starting with CISD, then onwards. Time-dependent Hartree–Fock will be considered, as will stability analysis for the self-consistent field.

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) D. G. A. Smith, L. A. Burns, D. A. Sirianni, D. R. Nascimento, A. Kumar, A. M. James, J. B. Schriber, T. Zhang, B. Zhang, A. S. Abbott, E. J. Berquist, M. H. Lechner, L. A. Cunha, A. G. Heide, J. M. Waldrop, T. Y. Takeshita, A. Alenaizan, D. Neuhauser, R. A. King, A. C. Simmonett, J. M. Turney, H. F. Schaefer, F. A. Evangelista, A. E. I. DePrince, T. D. Crawford, K. Patkowski and C. D. Sherrill, *J. Chem. Theory Comput.*, 2018, **14**, 3504–3511.
- (3) Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters and G. K.-L. Chan, *WIREs Comput. Mol. Sci.*, 2018, **8**, e1340.
- (4) F. Neese, *WIREs Comput. Mol. Sci.*, 2012, **2**, 73–78.
- (5) F. Neese, *WIREs Comput. Mol. Sci.*, 2022, **12**, e1606.
- (6) C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke and T. E. Oliphant, *Nature*, 2020, **585**, 357–362.
- (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) J. D. Hunter, *Comput. Sci. Eng.*, 2007, **9**, 90–95.
- (9) *TermColor*, <https://pypi.org/project/termcolor/>.

- (10) P. J. Mohr, E. Tiesinga, D. B. Newell and B. N. Taylor, *Codata Internationally Recommended 2022 Values of the Fundamental Physical Constants*, en, 2024.
- (11) T. Koopmans, *Physica*, 1934, **1**, 104–113.
- (12) R. S. Mulliken, *J. Chem. Phys.*, 1955, **23**, 1833–1840.
- (13) P.-O. Löwdin, *J. Chem. Phys.*, 1950, **18**, 365–375.
- (14) I. Mayer, *Chem. Phys. Lett.*, 1983, **97**, 270–274.
- (15) E. R. Davidson and A. E. Clark, *Int. J. Quantum Chem.*, 2022, **122**, e26860.
- (16) E. R. Davidson, *Adv. Quantum Chem.*, 1972, **6**, ed. P.-O. Löwdin, 235–266.
- (17) S. Grimme, *J. Comput. Chem.*, 2006, **27**, 1787–1799.
- (18) G. Audi and A. Wapstra, *Nucl. Phys. A*, 1993, **565**, 1–65.
- (19) G. Audi and A. Wapstra, *Nucl. Phys. A*, 1995, **595**, 409–480.
- (20) J. Neugebauer, M. Reiher, C. Kind and B. A. Hess, *J. Comput. Chem.*, 2002, **23**, 895–910.
- (21) R. Iftimie, P. Minary and M. E. Tuckerman, *Proc. Natl Acad. Sci.*, 2005, **102**, 6654–6659.
- (22) R. Car and M. Parrinello, *Phys. Rev. Lett.*, 1985, **55**, 2471–2474.
- (23) W. C. Swope, H. C. Andersen, P. H. Berens and K. R. Wilson, *J. Chem. Phys.*, 1982, **76**, 637–649.
- (24) C. C. J. Roothaan, *Rev. Mod. Phys.*, 1951, **23**, 69–89.
- (25) G. G. Hall, *Proc. Roy. Soc. London A*, 1951, **205**, 541–552.
- (26) D. R. Hartree and W. Hartree, *Proc. Roy. Soc. London A*, 1935, **150**, 9–33.
- (27) P. Pulay, *Chem. Phys. Lett.*, 1980, **73**, 393–398.
- (28) M. C. Zerner and M. Hehenberger, *Chem. Phys. Lett.*, 1979, **62**, 550–554.
- (29) C. Møller and M. S. Plesset, *Phys. Rev.*, 1934, **46**, 618–622.
- (30) S. Grimme, *J. Chem. Phys.*, 2003, **118**, 9095–9102.
- (31) S. Grimme, *J. Comput. Chem.*, 2003, **24**, 1529–1537.
- (32) R. C. Lochan and M. Head-Gordon, *J. Chem. Phys.*, 2007, **126**, 164101.
- (33) F. Neese, T. Schwabe, S. Kossmann, B. Schirmer and S. Grimme, *J. Chem. Theory Comput.*, 2009, **5**, 3060–3073.
- (34) P. Pulay and S. Saeb, *Theor. Chim. Acta*, 1986, **69**, 357–368.
- (35) P. R. Surján, *Chem. Phys. Lett.*, 2005, **406**, 318–320.
- (36) M. Kobayashi and H. Nakai, *Chem. Phys. Lett.*, 2006, **420**, 250–255.
- (37) T. J. Lee, A. P. Rendell and P. R. Taylor, *J. Phys. Chem.*, 1990, **94**, 5463–5468.
- (38) S. Hirata, R. Podeszwa, M. Tobita and R. J. Bartlett, *J. Chem. Phys.*, 2004, **120**, 2581–2592.
- (39) J. Noga and R. J. Bartlett, *J. Chem. Phys.*, 1987, **86**, 7041–7050.
- (40) S. A. Kucharski and R. J. Bartlett, *The Journal of Chemical Physics*, 1992, **97**, 4282–4288.
- (41) A. Jiang, D. Matthews, D. Poole, C. Briggs, J. Turney, C. D. Sherrill and H. Schaefer, *ChemRxiv*, 2025, **2025**, DOI: [10.26434/chemrxiv-2025-qgc1q](https://doi.org/10.26434/chemrxiv-2025-qgc1q).
- (42) C. Riplinger, B. Sandhoefer, A. Hansen and F. Neese, *J. Chem. Phys.*, 2013, **139**, 134101.

- (43) A. P. Rendell, T. J. Lee and A. Komornicki, *Chem. Phys. Lett.*, 1991, **178**, 462–470.
- (44) M. Lesiuk, *Journal of Chemical Theory and Computation*, 2022, **18**, PMID: 36314739, 6537–6556.
- (45) A. Jiang, H. S. III and J. Turney, *ChemRxiv*, 2025, **2025**, DOI: 10.26434/chemrxiv-2025-77dmp-v2.
- (46) C. Hampel, K. A. Peterson and H.-J. Werner, *Chem. Phys. Lett.*, 1992, **190**, 1–12.
- (47) P. Hohenberg and W. Kohn, *Phys. Rev.*, 1964, **136**, B864–B871.
- (48) W. Kohn and L. J. Sham, *Phys. Rev.*, 1965, **140**, A1133–A1138.
- (49) J. C. Slater, *Phys. Rev.*, 1951, **81**, 385–390.
- (50) A. D. Becke, *Phys. Rev. A*, 1988, **38**, 3098–3100.
- (51) J. P. Perdew, K. Burke and M. Ernzerhof, *Phys. Rev. Lett.*, 1996, **77**, 3865–3868.
- (52) K. Burke, J. P. Perdew and Y. Wang, in *Electronic Density Functional Theory: Recent Progress and New Directions*, ed. J. F. Dobson, G. Vignale and M. P. Das, Springer US, Boston, MA, 1998, pp. 81–111.
- (53) C. Adamo and V. Barone, *J. Chem. Phys.*, 1998, **108**, 664–675.
- (54) J. Tao, J. P. Perdew, V. N. Staroverov and G. E. Scuseria, *Phys. Rev. Lett.*, 2003, **91**, 146401.
- (55) S. H. Vosko, L. Wilk and M. Nusair, *Can. J. Phys.*, 1980, **58**, 1200–1211.
- (56) J. P. Perdew and Y. Wang, *Phys. Rev. B*, 1992, **45**, 13244–13249.
- (57) J. P. Perdew, *Phys. Rev. B*, 1986, **33**, 8822–8824.
- (58) C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B*, 1988, **37**, 785–789.
- (59) C. Adamo and V. Barone, *J. Chem. Phys.*, 1999, **110**, 6158–6170.
- (60) J. P. Perdew and K. Schmidt, *AIP Conference Proceedings*, 2001, **577**, 1–20.
- (61) C. Adamo and V. Barone, *Chem. Phys. Lett.*, 1997, **274**, 242–250.
- (62) A. D. Becke, *J. Chem. Phys.*, 1993, **98**, 5648–5652.
- (63) P. J. Stephens, F. J. Devlin, C. F. Chabalowski and M. J. Frisch, *J. Phys. Chem.*, 1994, **98**, 11623–11627.
- (64) J. C. Sancho-García, *J. Chem. Phys.*, 2006, **124**, 124112.
- (65) E. Brémond and C. Adamo, *J. Chem. Phys.*, 2011, **135**, 024106.
- (66) É. Brémond, J. C. Sancho-García, Á. J. Pérez-Jiménez and C. Adamo, *J. Chem. Phys.*, 2014, **141**, 031101.
- (67) J.-D. Chai and S.-P. Mao, *Chem. Phys. Lett.*, 2012, **538**, 121–125.
- (68) S. Grimme, *J. Chem. Phys.*, 2006, **124**, 034108.
- (69) A. Karton, A. Tarnopolsky, J.-F. Lamère, G. C. Schatz and J. M. L. Martin, *J. Phys. Chem. A*, 2008, **112**, 12868–12886.
- (70) F. Yu, *J. Phys. Chem. A*, 2014, **118**, 3175–3182.
- (71) T. Schwabe and S. Grimme, *Phys. Chem. Chem. Phys.*, 2007, **9**, 3397–3406.
- (72) S. Kozuch, D. Gruzman and J. M. L. Martin, *J. Phys. Chem. C*, 2010, **114**, 20801–20808.

- (73) A. D. Becke, *J. Chem. Phys.*, 1988, **88**, 2547–2553.
- (74) J. B. Foresman, M. Head-Gordon, J. A. Pople and M. J. Frisch, *J. Phys. Chem.*, 1992, **96**, 135–149.
- (75) M. Head-Gordon, R. J. Rico, M. Oumi and T. J. Lee, *Chem. Phys. Lett.*, 1994, **219**, 21–29.
- (76) L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.*, 1978, **26**, 218–231.
- (77) F. Neese, *J. Comput. Chem.*, 2023, **44**, 381–396.
- (78) S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn and K. Smith, *Comput. Sci. Eng.*, 2011, **13**, 31–39.
- (79) B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibbsom and T. L. Windus, *J. Chem. Inf. Model.*, 2019, **59**, 4814–4820.
- (80) W. J. Hehre, R. F. Stewart and J. A. Pople, *J. Chem. Phys.*, 1969, **51**, 2657–2664.
- (81) J. S. Binkley, J. A. Pople and W. J. Hehre, *J. Am. Chem. Soc.*, 1980, **102**, 939–947.
- (82) R. Ditchfield, W. J. Hehre and J. A. Pople, *J. Chem. Phys.*, 1971, **54**, 724–728.
- (83) T. Clark, J. Chandrasekhar, G. W. Spitznagel and P. V. R. Schleyer, *J. Comput. Chem.*, 1983, **4**, 294–301.
- (84) V. A. Rassolov, M. A. Ratner, J. A. Pople, P. C. Redfern and L. A. Curtiss, *J. Comput. Chem.*, 2001, **22**, 976–984.
- (85) 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.
- (86) P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta*, 1973, **28**, 213–222.
- (87) W. J. Hehre, R. Ditchfield and J. A. Pople, *J. Chem. Phys.*, 1972, **56**, 2257–2261.
- (88) R. Krishnan, J. S. Binkley, R. Seeger and J. A. Pople, *J. Chem. Phys.*, 1980, **72**, 650–654.
- (89) M. J. Frisch, J. A. Pople and J. S. Binkley, *J. Chem. Phys.*, 1984, **80**, 3265–3269.
- (90) L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov and J. A. Pople, *J. Chem. Phys.*, 1998, **109**, 7764–7776.
- (91) J. Dunning, Thom H., *J. Chem. Phys.*, 1989, **90**, 1007–1023.
- (92) D. E. Woon and J. Dunning, Thom H., *J. Chem. Phys.*, 1993, **98**, 1358–1371.
- (93) A. K. Wilson, T. van Mourik and T. H. Dunning, *J. Mol. Struct.*, 1996, **388**, 339–349.
- (94) T. van Mourik, A. K. Wilson and T. H. Dunning, Jr, *Mol. Phys.*, 1999, **96**, 529–547.
- (95) D. E. Woon and T. H. Dunning, *J. Chem. Phys.*, 1995, **103**, 4572–4585.
- (96) K. A. Peterson and T. H. Dunning, *J. Chem. Phys.*, 2002, **117**, 10548–10560.
- (97) F. Neese and E. F. Valeev, *J. Chem. Theory Comput.*, 2011, **7**, 33–43.
- (98) F. Jensen, *J. Chem. Phys.*, 2001, **115**, 9113–9125.
- (99) F. Jensen, *J. Chem. Phys.*, 2002, **116**, 7372–7379.
- (100) F. Jensen, *J. Chem. Phys.*, 2002, **117**, 9234–9240.
- (101) F. Jensen, *J. Chem. Theory Comput.*, 2014, **10**, 1074–1085.
- (102) F. Weigend and R. Ahlrichs, *Phys. Chem. Chem. Phys.*, 2005, **7**, 3297.

(103) D. Rappoport and F. Furche, *J. Chem. Phys.*, 2010, **133**, 134105.