



POLITECNICO
MILANO 1863

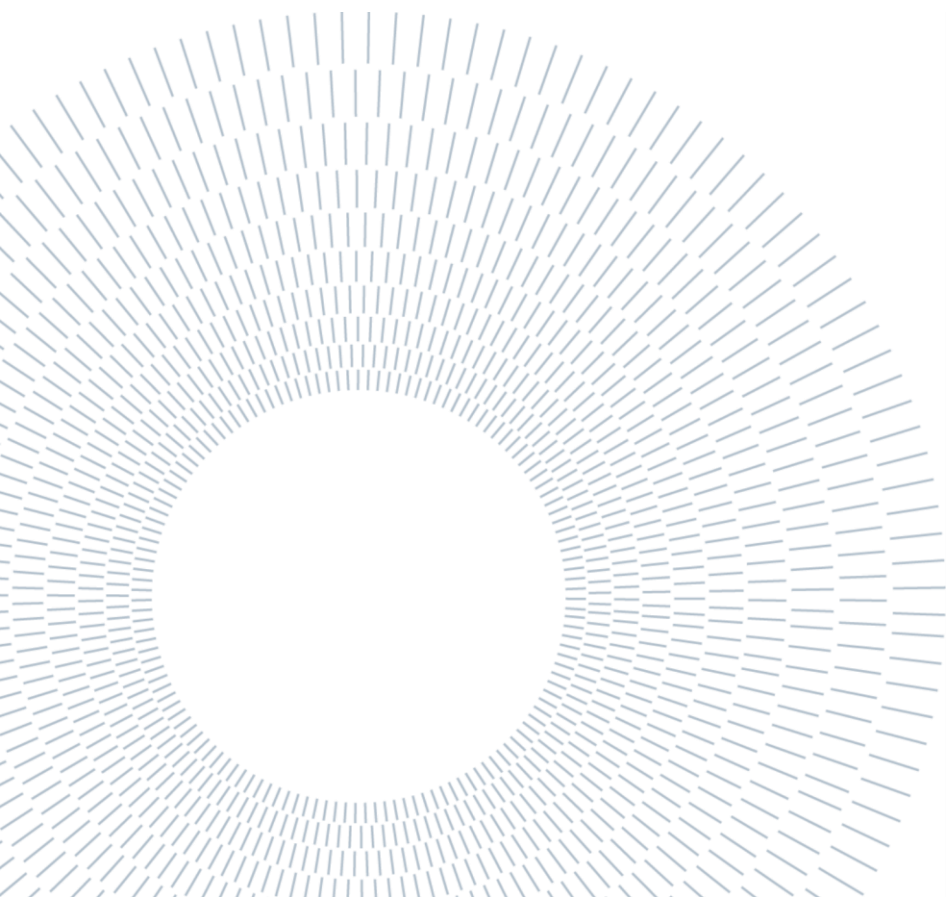
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Fostering automation in chemical kinetics: a protocol for bond energy computation and the implementation of a hierarchical approach for thermochemistry calculations

TESI DI LAUREA MAGISTRALE IN
CHEMICAL ENGINEERING-INGEGNERIA CHIMICA

Author: **Marcello Ferraro**

Student ID: 10602732
Advisor: Carlo Alessandro Cavallotti
Co-advisor: Andrea Della Libera
Academic Year: 2022-23



Abstract

Automatic PES exploration and thermochemical parameters estimation are fundamental challenges to overcome to fully automate calculation procedures in predictive chemical kinetics from first principles. Our in-house software EStokTP was born with this final purpose and is continuously updated to foster automation and minimize human time spent on the calculations and thus the possibility of human mistakes. In the present work, three main protocols written in Python that work in synergy with EStokTP are presented: 1) the automatic generation of input data for EStokTP calculations, 2) the estimation of bond energy and 3) thermochemical parameters estimation in the form of NASA polynomials.

The automatic generation of input data avoids the generation of the Z-matrix by the user, which is sometimes source of error caused by non-trivial geometry definition (bad first guess structure) or bad atom order in the Z-matrix (e.g., specific dihedral angle definition), and it is a tedious process if the required calculations involve a high number of species.

The bond energies are calculated by generating the possible fragments obtained by the rupture of a bond of the original molecule; the information can be used for the selection of the main reaction channels to explore in a more detailed way and avoiding side reaction channels with too high activation energy. The algorithm was tested on 1,3-butadiene-2-ol and simulation results are compared with correspondent experimental evaluations. In particular, the estimated bond energy for fragmentation number 4 (H-abstraction from oxygen atom) is only 0.9 [kcal mol⁻¹] higher than the latest literature theoretical estimate.

Estimation of $C_p^0(T)$ and $S^0(T)$ is made by exploiting molecular translational, vibrational, and rotational partition function contributions and 1D hindered rotor theory. $H^0(T)$ is evaluated by estimating $\Delta H^0(0 K)$ using the Connectivity Based Hierarchy method, which is a rung-based atomization scheme. $\Delta H^0(0 K)$ is then corrected to $\Delta H^0(298.15 K)$, with an extrapolation scheme based on experimental atomization enthalpies. Finally, a series of regressions evaluate the set of 14 coefficients required for the computation of NASA polynomials.

The estimation of $\Delta H^0(0\text{ K})$ is tested on a set of 142 species, reporting a mean absolute error of 0.39 [kcal mol⁻¹], while the influence of the level of theory and the correction of $\Delta H^0(0\text{ K})$ to $\Delta H^0(298.15\text{ K})$ are tested on a set of 8 molecules with a mean absolute error of 0.54 [kcal mol⁻¹]. Finally, the NASA polynomials of isoprene and 1,3-butadiene-2-ol are evaluated and confronted with an external database result.

Key-words: bond energy, thermochemistry, atomization, NASA polynomials

Abstract in lingua italiana

L'esplorazione automatica delle superfici di potenziale e la stima dei parametri termochimici sono sfide fondamentali da superare per automatizzare completamente le procedure di calcolo nella cinetica chimica predittiva. Il software EStokTP è nato con questo scopo finale ed è continuamente aggiornato per favorire l'automazione e ridurre al minimo il tempo speso dall'utente per i calcoli e quindi la possibilità di errori umani. Nel presente lavoro vengono presentati tre protocolli principali scritti in Python che lavorano in sinergia con EStokTP: 1) la generazione automatica di dati di input per i calcoli di EStokTP, 2) la stima dell'energia di legame e 3) la stima dei parametri termochimici sotto forma di polinomi NASA.

La generazione automatica degli input evita la scrittura a mano della matrice z da parte dell'utente, che talvolta risulta essere fonte di errori causati dalla definizione non banale della geometria (struttura di primo tentativo errata) o dall'ordine di definizione degli atomi nella Z -matrix (per esempio nella definizione degli angoli diedri) e risulta essere un processo tedioso se i calcoli richiesti includono un elevato numero di specie dalla geometria complessa.

Le energie di legame sono calcolate generando tutti i possibili frammenti ottenuti dalla rottura di un unico legame della molecola di partenza; le informazioni possono essere usate per la selezione dei principali canali di reazione da esplorare in maniera più dettagliata, evitando i canali di reazione secondari aventi una energia di attivazione troppo elevata. L'algoritmo è stato testato sul 1,3-butadiene-2-olo, e le simulazioni comparate con i risultati sperimentali in letteratura. In particolare, l'energia stimata per l'estrazione dell'idrogeno dal gruppo ossidrilico è stata sovrastimata di sole 0.9 [kcal mol⁻¹] rispetto alla più recente stima teorica in letteratura.

La stima di $C_p^0(T)$ e $S^0(T)$ è effettuata utilizzando i contributi traslazionali, vibrazionali e rotazionali delle funzioni di partizione molecolare, e il trattamento di alcune vibrazioni come rotori impediti nella teoria dei rotori monodimensionali. $H^0(T)$ è valutata stimando dapprima $\Delta H^0(0\text{ K})$ usando un metodo basato sulla gerarchia di connettività, che è uno schema di atomizzazione a più livelli. Successivamente $\Delta H^0(0\text{ K})$ è corretto per valutare $\Delta H^0(298.15\text{ K})$, utilizzando uno

schema di estrapolazione basato sulle entalpie sperimentali dei singoli atomi. Una serie di regressioni calcola infine il set di 14 coefficienti richiesti per i polinomi NASA.

La stima del $\Delta H^0(0 K)$ è stata testata su 142 specie, riportando un errore assoluto medio di 0.39 [kcal mol⁻¹], mentre l'influenza del livello di teoria usato e la correzione da $\Delta H^0(0 K)$ a $\Delta H^0(298.15 K)$ è stata testata su un set di 8 specie, con un errore medio assoluto di 0.54 [kcal mol⁻¹]. Infine i polinomi NASA di isoprene e 1,3-butadiene-2-olo sono stati valutati e confrontati con i risultati forniti da un database esterno.

Parole chiave: energia di legame, termochimica, atomizzazione, polinomi NASA

Page intentionally left blank

Contents

Abstract.....	i
Abstract in lingua italiana	iii
Contents.....	vii
1. Introduction	11
1.1 Quantum chemistry and chemical kinetics.....	12
1.1.1 Status of chemical kinetics	12
1.1.2 Quantum chemistry development.....	13
1.2 Thermochemical parameters.....	15
1.2.1 Group contribution methods.....	15
1.2.2 Atomization scheme methods	16
1.2.3 Connectivity Based Hierarchy method	17
1.2.4 Future improvements	17
1.2.5 NASA polynomials	18
1.2.6 CHEMKIN format of NASA polynomials.....	19
1.3 Chemical kinetics software.....	20
1.3.1 EStokTP.....	20
1.3.2 RMG	22
1.3.3 AutoMech.....	24
1.3.4 KinBot.....	25
1.3.5 Genesys	27
1.3.6 Arkane.....	28
1.5 Purpose of this work	28
2. Methods.....	30
2.1 EStokTP input files	30
2.1.1 estoktp.dat	30

2.1.2	Species data file (XXXX.dat)	33
2.1.3	theory.dat.....	36
2.1.4	me_head.dat.....	37
2.1.5	Molpro theory files.....	38
2.2	RDKit.....	38
2.3	InChI and SMILES	39
2.4	InChI2data	40
2.4.1	XYZ format structure generation	40
2.4.2	Cartesian coordinates conversion to Z-matrix.....	42
2.4.3	data subdirectory generation.....	44
2.4.4	InChI2data user manual.....	45
2.5	FragGen	46
2.5.1	Bond breakage	46
2.5.2	Equivalent fragmentations.....	47
2.5.3	Bond energy calculation	47
2.5.4	FragGen user manual.....	49
2.6	CHEMTP	49
2.6.1	CBH-0, Isogyric scheme.....	53
2.6.2	CBH-1, Isodesmic scheme	55
2.6.3	CBH-2, Homodesmotic scheme.....	60
2.6.4	Reference species database	64
2.6.5	Correction of $\Delta H^0(0 K)$	64
2.6.6	Estimation of NASA polynomials coefficients.....	66
2.6.7	CHEMTP user manual.....	68
2.7	Summary of the codes.....	49
3.	Results and discussion	70
3.1	Fragmentation of 1,3-butadiene-2-ol.....	70
3.2	Thermochemical parameters.....	73
3.2.1	Estimation of $\Delta H^0(0 K)$ at ω B97X-D/jun-cc-pVTZ	73
3.2.2	Estimation of $\Delta H^0(0 K)$: influence of the level of theory.....	78
3.2.3	Correction of $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$	81

3.2.4 NASA polynomials comparison	82
4. Conclusion and future development	87
Bibliography.....	89
A. CBH reference species	97
B. Estimated standard enthalpy @ 0 K.....	103
C. Isoprene and 1,3-butadiene-2-ol NASA polynomials.....	107
List of Figures.....	109
List of Tables	111
List of Symbols.....	112
Acknowledgements	115

Page intentionally left blank

1. Introduction

The recent advancements in quantum chemistry and the automation of chemical kinetics went hand in hand with the exponential increase in the capability of high-performance computers (HPC). The latest frontier in these fields is the generation of kinetic mechanisms using an automated procedure to determine all possible reaction channels; while rate constants for elementary steps such as abstraction, addition, beta-scission, and isomerization reactions can be determined accurately, the investigation of a complex potential energy surface (PES) is still challenging. The complexity of this task lies not only in the determination of all possible reactant structures, but also in the determination of the Transition State (TS) structure, for the application of canonical Transition state theory (TST) and its variational form (VTST).

The automated estimation of thermochemical parameters has advanced even further: the current protocols are capable of estimating entropies, heat capacities and enthalpies with accuracy comparable to experiments; the bottleneck of the procedure is the time requirement of HPC in the determination of electronic energy and zero-point energy (ZPE) of large chemical compounds (8 or more non-hydrogen atoms) and the selection of the proper level of theory for the calculations.

Software like EStokTP, RMG, AutoMech, Kinbot, Genesys and Arkane implement different algorithms for the estimation of kinetic constants and thermochemical parameters, at various levels of theory.

Chapter one will present state of chemical kinetics, focusing both on postdictive and predictive kinetics. Then the historic development of quantum chemistry will be introduced, emphasizing the importance of computational chemistry software and Post-Hartree-Methods to solve the Schrödinger equation. A panoramic on thermochemical parameters estimation is given, concentrating on methods such as group contribution and atomization schemes. The final part is dedicated to the presentation of chemical kinetics software and their crucial importance in predicting and understanding the rates and mechanisms of chemical reactions.

1.1 Quantum chemistry and chemical kinetics

1.1.1 Status of chemical kinetics

Kinetic mechanisms are fundamental in the modelling of chemical reactors [1], industrial equipment such as turbines [2] and in computational fluid dynamics studies [3]. As suggested by Green [4], the status of chemical kinetics has reached a stage where predictive kinetics can override postdictive kinetics.

Historically, postdictive kinetics has relied on experimental observations, based on a hypothetical mechanism, and numerical fitting over small temperature and pressure ranges. The most common form of kinetic schemes obtained using this approach is a list of power laws-like expressions (1.1), in which the kinetic constant is an Arrhenius or modified Arrhenius expression (1.2). For a second order reaction $A + B \rightarrow C$ the equation for its rate is:

$$r(T, P) = K C_A^\alpha C_B^\beta \quad (1.1)$$

$$K(T) = K_0 e^{-\frac{E_A}{RT}} \quad (1.2)$$

With:

- C_A and C_B concentration of reactant A and B, respectively [$mol\ m^{-3}$]
- α and β fitting exponentials of C_A and C_B , respectively [–]
- K_0 pre-exponential factor, temperature dependent in the case of modified Arrhenius expressions [$m^3\ mol^{-1}s^{-1}$]
- E_A activation energy [$kcal\ mol^{-1}$]
- R ideal gas constant [$kcal\ mol^{-1}K^{-1}$]
- r reaction rate [$mol\ m^{-3}s^{-1}$]

This method, although relatively fast and cheap, has several disadvantages. The most evident limitation is the need of a guessed mechanism for parameter fitting: for complex reaction schemes like combustion of heavy hydrocarbons is nearly impossible to guess all elementary steps. The order of magnitude of an 87-octane gasoline/ethanol blended is about tens of thousands of reactions [5].

This limitation can be partially overcome using lumped mechanisms, in which a series of molecules with similar reacting behaviour are “lumped” into a single representative species [6]. This method reduces the number of reactions involved in a complete mechanism, but the volume of reactions remains untreatable in a satisfying manner.

Another limitation is the impossibility of extrapolation outside the fitting ranges. Since a “black box” model is implemented for data fitting, with no molecular dynamics nor quantum chemistry studies, the obtained parameters cannot be safely used outside the fitting temperature/pressure ranges; there is no general accepted extrapolation scheme for this type of dependencies, which leads to the need of several experiments in order to acquire sufficient data over a large temperature/pressure range.

The temperature dependence of the rate constant, with no pressure dependence, is also a hypothesis which leads to unacceptable results in the fall-off regime of the Lindemann theory. This hypothesis has been corrected by Marcus, in the context of RRKM theory [7].

The last disadvantage of postdictive kinetics is the impossibility of identifying species with a short lifetime, such as radicals; these species have a strong impact on reaction rates and in many cases lead to unacceptable estimations of rate constants, for reacting environments in which these radicals have a longer lifetime, such as space environments.

The issues of postdictive kinetics can be partially solved moving to predictive kinetics [4]; the driving forces of this shift are the progress in the field of theoretical chemical kinetics [8] and the increase of the available computational resources. Several software has been developed with this aim, following the Ab Initio Transition State based Master Equation (AITSTME) approach; the first step is a preliminary electronic structure calculation for successive PES investigation. The second step is to increase the level of theory as requested by the type of calculations, for reactants, products, and TS. As last step, the Master Equation is solved, obtaining the values of the rate constants [9].

Different software implements this protocol coupled with a series of high-level corrections (such as anharmonic oscillator [10] for Rigid Rotor Harmonic Oscillator corrections (RRHO) and quantum tunnelling [11]), with the aim of automatically generate a complete kinetic mechanism with results comparable with postdictive kinetics predictions.

1.1.2 Quantum chemistry development

The accurate results of ab initio chemical kinetics calculations are possible due to the higher levels of theory developed in the field of quantum chemistry (QC). Quantum chemistry development started in the '30 of the 20th century, with the formulation of Schrödinger's wave equation [12], based on de Broglie hypothesis [13].

Although Schrödinger equation has been a crucial step in quantum physics development, its analytical solution is possible only for special case systems (single electron hydrogen); for more complicated systems, only an approximated solution can be derived. Due to limited computational power, the approximated solutions were still limited to small molecular systems. The development of quantum mechanical methods and the increase of computational power in the '70 allowed the extension of approximated solutions to larger systems.

One of the first and most used theory at that time was the Restricted Hartree-Fock (RHF) [14] theory, which consists in an approximated method of treating the electron-electron interaction in closed shell systems (stable species). Although the method fails in describing a series of species such as radicals [15], it represents the starting point of successive approaches which make corrections in the electron-electron interaction treatment, named Post-Hartree-Fock methods, to highlight the conceptual base of this theories [16].

Among all, Configuration Interaction (CI) [17] and Møller-Plesset Multi-Body Perturbation Theory (MPMBPT) [18] are the most employed by far, balancing the required precision and the necessary computational time.

Another method used in modern quantum chemistry is Density Functional Theory (DFT), based on Hohenberg-Kohn theorem [19], which describes molecular properties studying the electron density distribution, rather than describing each molecular orbital as CI and MPMBPT do.

Software implementing these different theories have been developed through the years and continuously updated. Examples include Gaussian [20] (first released in 1970 as Gaussian70 and at present updated in 2016 as Gaussian16), developed by J.A. Pople et al. (1970), and Molpro [21], developed by Werner et al. (latest update in December 2022).

At present, quantum chemistry software can predict thermochemical parameters under chemical accuracy (1 kcal mol^{-1}) for small systems (less than 6 heavy atoms) and even achieving sub-chemical accuracy (around $0.1 \text{ kcal mol}^{-1}$). For medium-large system (6-11 heavy atoms) the chemical accuracy is not obtained easily, because of the difficulty in the estimation of electronic and zero-point energy, and low-quality frequencies using the RRHO approximation.

1.2 Thermochemical parameters

Accurate estimation of thermochemical parameters is one of the challenges of modern quantum chemistry. Specific heat C_p , enthalpy H and entropy S are fundamental to track the evolution of energy distribution in reacting environments, and they have a direct influence on reaction rates through temperature and equilibrium conditions, because of their relationship with the Gibbs free energy $G = H - TS$.

A series of empirical and semi-empirical methods have been developed, mostly based on different types of group contributions; more modern approaches exploit quantum chemistry calculations for the implementation of high-level atomization procedures. Some of the popular schemes are called Isogyric [22] by Snyder and Basch, Isodesmic [23-25] by Pople et al., Homodesmotic [26-28] by George et al., Hyperhomodesmotic [29] by Hess and Schaad, Semi-homodesmotic [30] by Nyulaszi et al., Quasihomodesmotic [31-32] by Vianello et al., Homomolecular homodesmotic [33] by Chestnut and Davis, Isogeitonic [34] by El-Nahas et al., Isoplesitoic and Homoplesitoic [35] by George et al., and s-homodesmotic [36-38] by Zhao et al. Such schemes allow the estimations of $\Delta H^0(0 K)$, which are usually coupled with thermochemical expressions for $C_p(T)$ and $S(T)$ [39] estimations.

1.2.1 Group contribution methods

Group contributions are the first attempt for predictive estimation of thermochemical parameters. They are based on the idea of determining thermochemical properties by breaking the molecule of interest into sub-blocks of different dimensions (based on the specific fragmentation implemented) and summing the contribution of every block to obtain the desired property of the original molecule. The contribution of each block is determined by experiments on reference species and data fitting on expected results (obtained by calorimetry experiments, for example). The estimations obtained are reliable if the reference species contribution is estimated precisely and the choice of how to consider the reference groups is done with care.

Joback [40-41] proposed a group contribution method based on single heavy atoms and small molecular groups; the main advantage of this approach is that it requires a single structure analysis for the estimation of 11 thermochemical properties. Although easily implemented, Joback's method fails when different types of interaction are established between heavy atoms (like single, double, triple bonds or aromatic/non-aromatic rings). Joback's theory resembles the CBH-0 rung implementation (Isogyric [22] scheme).

Constantinou and Gani (CG) [42] made a distinction between first order and second order groups, making a classification of the immediate environment surrounding every heavy atom. First order groups are structures composed by a limited number of atoms and are used for low level estimations, while second order groups are formed by combination of first order group molecules, giving a higher-level estimation. Since a combinatory approach is used for second order groups, the dimensions of such database are greatly increased with respect to Joback's approach. Although the great improvement compared to Joback's method, CG scheme still fails for a series of common molecules and gives unsatisfactory results, with an error higher than chemical accuracy of 1 kcal mol⁻¹. CG resemble a mixed CBH-1/CBH-2 rung implementation (Isodesmic [23-25] and Hypomodesmotic [26-28] scheme, respectively).

Benson et al. [43] developed a method based on the number of ligands (but not the type) each heavy atom makes. This approach works better than Joback and CG because it implements a series of corrections such as non-next-nearest neighbour interactions (NNI), which considers interactions between atoms separated by at least 2 atoms, and ring strain corrections, to consider restrained ring, such a norbornane. The implementation of Benson's theory is like a CBH-1 rung.

1.2.2 Atomization scheme methods

Atomization schemes are protocols in which a molecule is conceptually broken into its atomic constituents; they are widely used for the estimation of chemical properties, such as standard enthalpy of formation. The introduction by Snyder and Basch [22] of an atomization scheme as saturation of heavy atoms of closed shell molecules was the first attempt of atomization procedure relying on precise experimental standard enthalpy of formation data. A series of atomization schemes have been developed since then, ranging from bond-centred Isogyric scheme [23-25] to atomic-centred Hybridization-based homodesmotic scheme [26-28].

Atomization schemes are used to study the energy changes associated with different fragmentation of a parent molecule down to its constituent atoms/groups; they permit to predict thermochemical parameters with high precision, if an adequate scheme is implemented. The improvements of atomization schemes, along with quantum chemistry theory, led to an initially precision around 30 [kcal mol⁻¹] by Snyder and Basch [22] to chemical precision of 1 [kcal mol⁻¹]. Also, the ease of implementation makes them the perfect choice for modern chemical kinetics software.

Although atomization schemes are widely used, they have some limitations; first, the dependency on experimental data for reference standard enthalpy of formation limits the applicability of such schemes on fragmentations that produce species with known $\Delta H^0(0 K)$. Another limitation is the neglect of pressure and temperature effects: typically, atomization schemes assume standard conditions, so pressure and temperature influence on energy changes are not considered. The main limitations are electron correlation effects, which are not taken fully into consideration, but can be significant in cases of highly correlated electronic structures.

1.2.3 Connectivity Based Hierarchy method

Connectivity Based Hierarchy (CBH) is a method for the estimation of the enthalpy of formation of pure components @ 0 [K], $\Delta H^0(0 K)$. A comprehensive work done by Ramabhadran and Raghavachari [44-47] summarizes different hybridization scheme levels. CBH is based on the construction of successive rungs on a scale, each one providing information for higher rungs and receiving information from lower rungs. The most used levels are the Isogyric scheme [22], namely CBH-0 rung from now on, Isodesmic bond separation scheme [23-25], namely CBH-1 rung from now on, and Hybridization-based homodesmotic scheme [26-28], namely CBH-2 rung from now on.

A rung lower than CBH-0 lower rung is also possible (namely atomization scheme), but it gives poor results for molecules with more than two heavy atoms and so it has not been taken into consideration in the algorithm implementation.

1.2.4 Future improvements

CBH methods are the last frontier of ab initio enthalpy estimation: they provide reliable prediction of $\Delta H^0(0 K)$ basing the calculations on quantum chemistry calculations. The bottlenecks of the estimation process are:

- The reliability of the reference data used for the construction of each rung.
- The computational time required for the estimation of the electronic and zero-point energy of the molecule of interest.
- The size of reference database for each rung.

The first two issues are partially solved by modern quantum chemistry theories and efficient ways in solving the Schrödinger equation and the rapid development of powerful HPC.

The last problem can be partially solved by carefully tuning the target of molecules the user is interested in estimating the $\Delta H^0(0 K)$. For the CBH-0 the maximum

number of heavy atoms in a reference species is 1; for the CBH-1 is 2; for the CBH-2 is 5. If a CBH-3 rung must be computed (Hyperhomodesmotic scheme [29]), the maximum number of heavy atoms in a reference species is 8. The number of possible reference species that can be computed with such number of heavy atoms, including radical species, makes the construction of a general database comprehensive of the most common organic heavy atoms (carbon C, oxygen O and nitrogen N) a challenging task. The size of the database can be reduced if no double/triple bonded species are included, or unstable species are excluded a priori.

Another issue related to reference species is the need of reliable electronic energy, zero-point energy and $\Delta H^0(0 K)$ data. The first two are related to computational times and for reference species up to CBH-2 modern HPC have no problem in dealing this type of computations, even with high level theories and large basis sets. The $\Delta H^0(0 K)$ are more problematic because they are either obtained by means of corrections to experiments or by lower rungs estimations (i.e., propane $\Delta H^0(0 K)$ can not be estimated at CBH-2, but only at CBH-1, so if $\Delta H^0(0 K)$ would not be available, the value estimated at CBH-1 is the only database source).

The estimation of $\Delta H^0(0 K)$ at high level of theory is now available only at CBH-2 as highest rung possible; a method for the automatic construction of a set of reference species for CBH-3 rung would be a significant improvement in the estimation procedure. The CBH-3 reference species $\Delta H^0(0 K)$, if not available, can be estimated at CBH-2 rung.

1.2.5 NASA polynomials

A convenient way of storing information regarding thermochemical parameters is the use of NASA polynomials, a set of coefficients used for handling the temperature dependence of a series of thermodynamical properties.

The NASA polynomials were introduced in the early '70 of 20th century at the NASA Lewis (now Glenn) Research Center. Earlier versions used a fourth-order polynomial as empirical representation of the quantity $C_P^0(T)/R$ over a temperature range of 300 to 5000 K [48-53].

A more modern format consists in a set of 14 coefficients (half for low temperature range and half for high temperature range), able to precisely describe the behaviour of the quantities $C_P^0(T)/R$, $H^0(T)/RT$ and $S^0(T)/R$, in their standard form (for gas it corresponds to ideal gas @ 1 bar,) [54]. Their expressions are reported below:

$$\frac{C_P^0(T)}{R} = a_0 + a_1T + a_2T^2 + a_3T^3 + a_4T^4 \quad (1.3)$$

$$\frac{H^0(T)}{RT} = a_0 + \frac{1}{2}a_1T + \frac{1}{3}a_2T^2 + \frac{1}{4}a_3T^3 + \frac{1}{5}a_4T^4 + \frac{a_5}{T} \quad (1.4)$$

$$\frac{S^0(T)}{R} = a_0 \ln(T) + a_1T + \frac{1}{2}a_2T^2 + \frac{1}{3}a_3T^3 + \frac{1}{4}a_4T^4 + a_6 \quad (1.5)$$

A modified expression adds two more coefficients for more accuracy over a wider range of temperatures, bringing the total amount of coefficients to 18. The addition of these coefficients has been avoided in the present work, due to little estimation improvements at the price of increased implementing complications.

The use of NASA polynomials has some clear advantages in the estimation of thermochemical parameters. First, it avoids the necessity of tabular interpolations since the polynomial functions are described continuously in the entire range of temperatures. Second, it permits analytical integration. Third, it condenses tabulated thermochemical information into a reduced set of coefficients, allowing fast prediction at a specific temperature of a wide range of parameters.

1.2.6 CHEMKIN format of NASA polynomials

CHEMKIN [55] format of NASA polynomials is one the most used way of storing information related to NASA polynomials parameters. There are several versions of CHEMKIN format, based on how much information every set should carry (i.e., the number of coefficients). One of the most used forms of the CHEMKIN format is the 14 coefficients, stored in three lines of five coefficients each (first the high temperature interval coefficients then the low temperature interval coefficients). A comment line is used to store useful information, such as molecular formula, phase of the molecule, high temperature limit, low temperature limit and split temperature. A 15 coefficients form adds the sum of electronic and zero-point energy, placing it in the last available spot of the fourth line. The CHEMKIN format, 14 coefficients of NASA polynomials of methane is reported below:

```
CH4                                G   200.000  3500.000  1000.000   1
  7.48514950E-02  1.33909467E-02-5.73285809E-06  1.22292535E-09-1.01815230E-13  2
-9.46834459E+03  1.84373180E+01  5.14987613E+00-1.36709788E-02  4.91800599E-05  3
-4.84743026E-08  1.66693956E-11-1.02466476E+04-4.64130376E+00  4
```

The CHEMKIN format of NASA polynomials is widely used for computational purposes for a series of reason. First, it gives a consistent and standardized way of representing thermochemical parameters information, which simplify the exchanging process between different research groups and software. Second, its flexibility in handling different types of information, such as chemical reaction mechanism and thermochemical parameters, allows the modelling of complex scientific and industrial applications. Third, the ease of use of the format: its simple

text-based nature makes it a suitable choice for handling and editing the CHEMKIN format with any programming language and software. Lastly, CHEMKIN format has been around for many years, hence many researchers are already versatile in its use and many software already use this format in their implementations.

1.3 Chemical kinetics software

The latest aim of quantum chemistry software is the complete automatization, decreasing the required time and effort from the user side, and the speed up of the calculations, as the levels of theory developed in the last years allow predictions of rate constants and thermochemical parameters under chemical precision. New chemical kinetics software integrates electronic structure software such as Gaussian and Molpro, adding further corrections and calculations, for example hindered rotors, Master Equation Solver (MES) codes and thermochemical parameter estimations.

A general analysis is carried out for EStokTP [9], RMG [56], AutoMech [57,58], KinBot [61], Genesys [62] and Arkane [63], paying particular attention in thermochemical parameters estimation, being the core argument of the present work.

1.3.1 EStokTP

EStokTP [9] is a computational environment developed in Fortran77; it is designed to directly couple electronic structure, TST and Master Equation evaluations, obtaining direct estimations of rate constants, ideally with a precision of no more than a factor of 2 from experiments. The design of the code has been done with the precise aim of reducing, if not eliminating, any human intervention.

One of the main strengths of EStokTP is the differentiation of protocols for each class of reactions supported (abstraction, addition, beta-scission, isomerization and barrierless). This makes EStokTP particularly versatile; the combination of its versatility with high accuracy predictions makes EStokTP one of the most powerful programs for rate constants estimation of gas-phase reactions, in particular combustion kinetics and atmospheric kinetics.

The workflow schematization is provided in Figure 1.1.

The first step is the optimization of reactants and products at a lower level of theory (level 0); this optimization needs the preparation of input data in form of a Z-matrix, with suitable characteristics, correlated with other input files stored in the **data** directory. This step can be tedious and time consuming if the species involved have complicated geometries or many atoms. A part of the present work covers the

automatization of this process. The optimization requires the Monte Carlo sampling of selected dihedral angles to locate the absolute minimum energy structure. A guess structure of the TS is then generated by performing a one-dimensional grid search as a function of a reaction class-dependent interatomic distance; then a first approximation of the TS is determined from standard protocols directly integrated in Gaussian or Molpro.

The results of the level 0 are used as input for level 1 calculations (which present a higher level of theory, compared with level 0), obtaining the final geometries and ZPE. RRHO-based conventional TST analysis can be applied at this level. Torsional scans (1D, 2D, 3D) are performed on specified dihedral angles; the torsional motions are then projected out of the level 1 Hessian, obtaining a set of vibrational frequencies and hindered rotors potentials.

Single point energies are then computed at a high level of theory on level 1 geometries for all the stationary points.

Reaction path scan and evaluation of symmetry numbers are finally performed.

All data computed are then collected in the folder **me_blocks**, which contains all blocks of input for the Master Equation Solver Software (MESS), which in turn extract the values of the rate constants.

An explicit subroutine for the estimation of thermochemical parameters is not yet implemented in EStokTP: the present work tries to cover this gap, making EStokTP able to predict thermochemical parameters, in the CHEMKIN form of NASA polynomials.

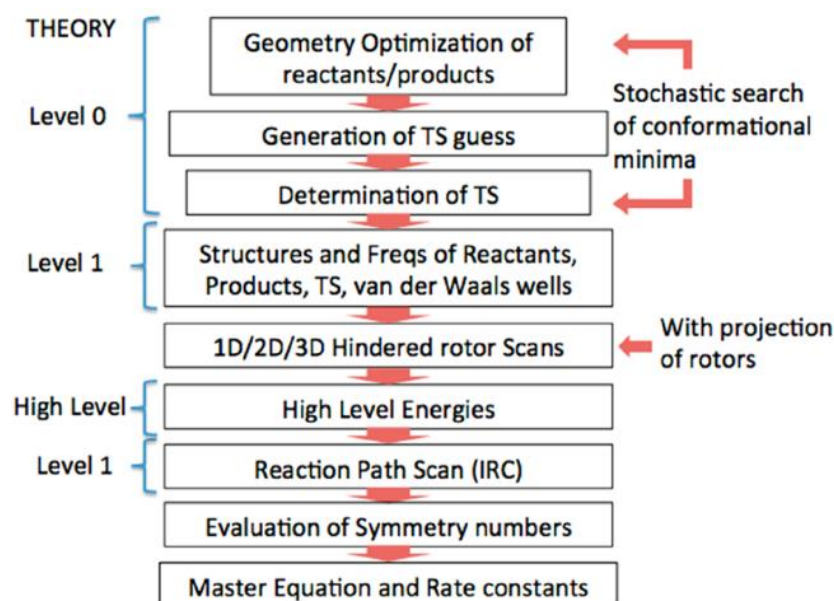


Figure 1.1: EStokTP program structure [9]

1.3.2 RMG

RMG [56] functions as an automated mechanism generator that leverages existing chemical knowledge housed within a database, coupled with parameter estimation techniques, to produce chemical kinetic mechanisms. It employs template definitions of reaction families that manipulate matching functional groups to convert reactants into products.

The kinetic constants estimation is based on the definition of a series of reaction templates, which describe the general evolution of the chemical species involved, such as bond connectivity changes and abstraction sites. Associated with every reaction family there is a hierarchical tree of rate estimation rules, which can be modified by the user to increase the amount of information contained. The inverse rate constant of the reaction of interest is computed exploiting thermodynamic consistency. The mechanism generation flowchart is reported in Figure 1.2.

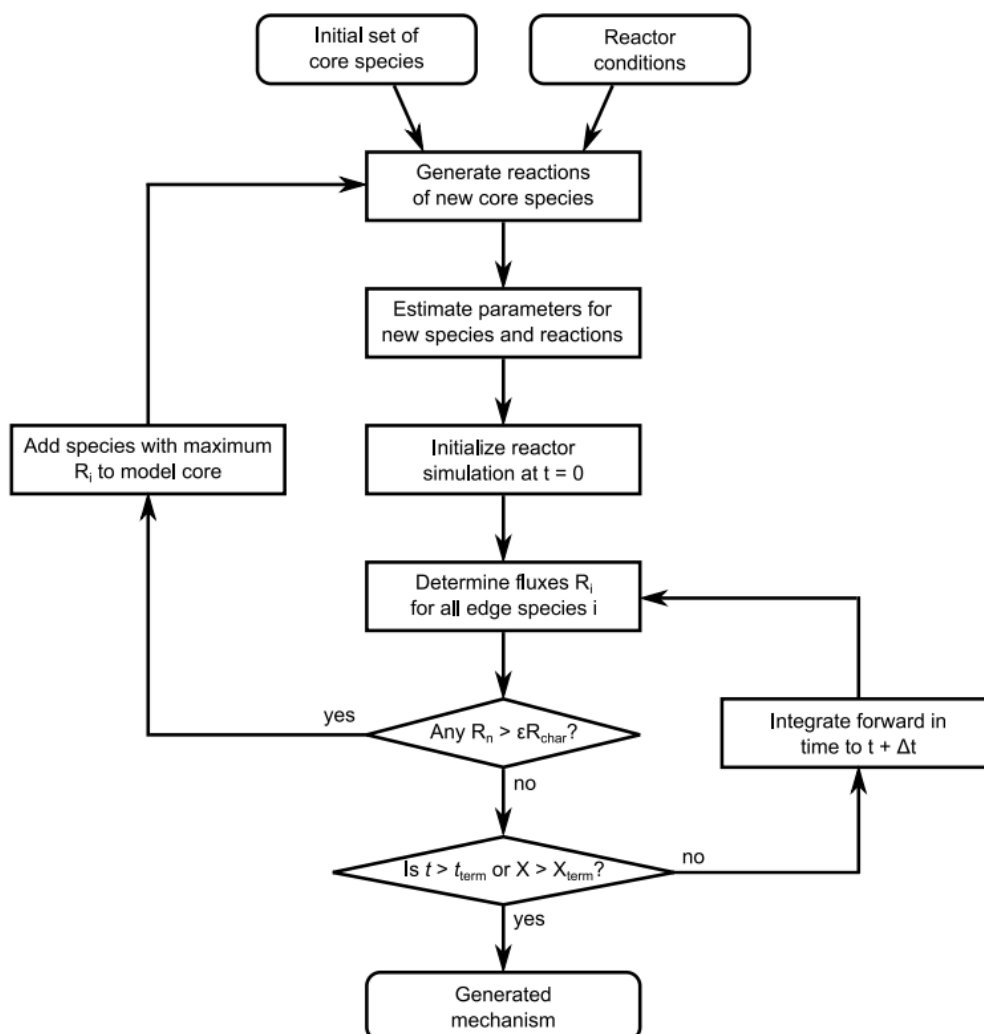


Figure 1.2: Flowchart of RMG rate-based algorithm [56]

Thermochemical parameters are estimated using Benson group additivity, with a series of corrections for radical, cyclic, and gauche structures. After generation of resonance isomers, thermodynamic properties of each isomer are calculated by identification of every group in the molecule. Symmetry number correction to entropy is then implemented. When all isomers have been identified, RMG chooses the one with the most stable enthalpy to represent the thermochemistry for the overall species. The general flowchart is reported in Figure 1.3

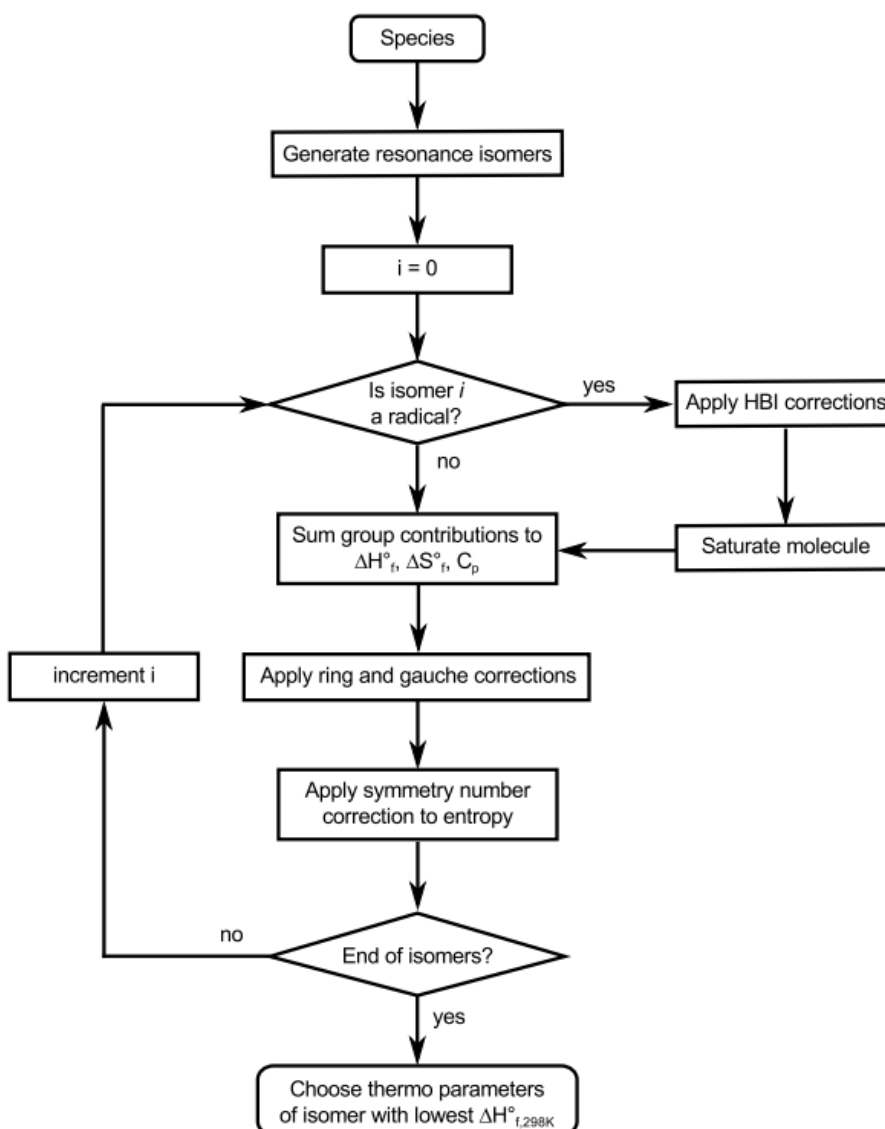


Figure 1.3: Flowchart of RMG group additivity-based thermodynamic parameter estimation algorithm [56]

1.3.3 AutoMech

Auto-Mech [57,58] is an open-source programming package for high level thermochemistry and kinetics estimations. It is designed to fully couple electronic structure calculations and rate constant calculations via TST; AutoMech is based on the concepts implemented in EStokTP previously presented.

The main strength of AutoMech is the managing of large-scale mechanism, involving 1000 to 10000 reactions and the estimation of thermochemical parameters for 100 to 1000 species.

The entire workflow is managed by MechDriver, which consists in a series of drivers and routines that execute simple tasks by calling libraries of low-level functions; this makes the code flexible to new adaptations and updates.

AutoMech flowchart is presented in Figure 1.4.

The first step is the analysis of all Potential Energy Surfaces, after providing a reaction mechanism (e.g., generated with RMG). Subsequently, completely connected reaction channels are determined by the analysis of PESs connectivity, identifying the so called sub-PESs. When the PESs has been defined, following the protocol also implemented in EStokTP, estimation of rate constants via AITSTME protocol is done.

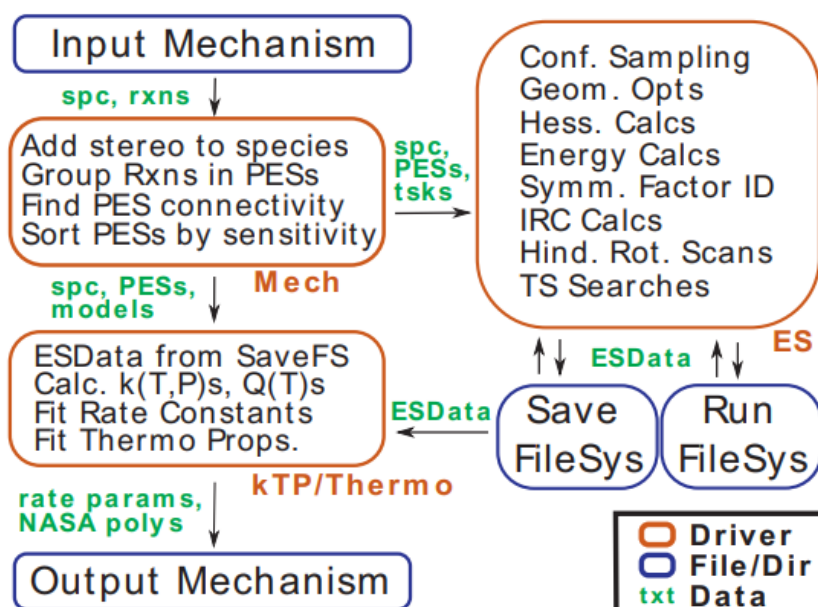


Figure 1.4: Flowchart of AutoMech [57]

Thermochemical estimations are managed through the predictive automated computational thermochemistry (PACT) software package [59]. It exploits electronic structure calculations (vibrational frequencies, anharmonic corrections, hindered

rotors, rotational symmetries) for the computation of partition functions contribution to each thermochemical parameter. Thermochemical information are then converted in NASA polynomials in the CHEMKIN format. The enthalpy estimation is exploited using the CBH-ANL approach [60], an automated procedure for the generation of CBH reactions and the estimation of $\Delta H^0(0 K)$.

A general flowchart of PACT implementation in AutoMech for automatic thermochemistry information is reported in Figure 1.5.

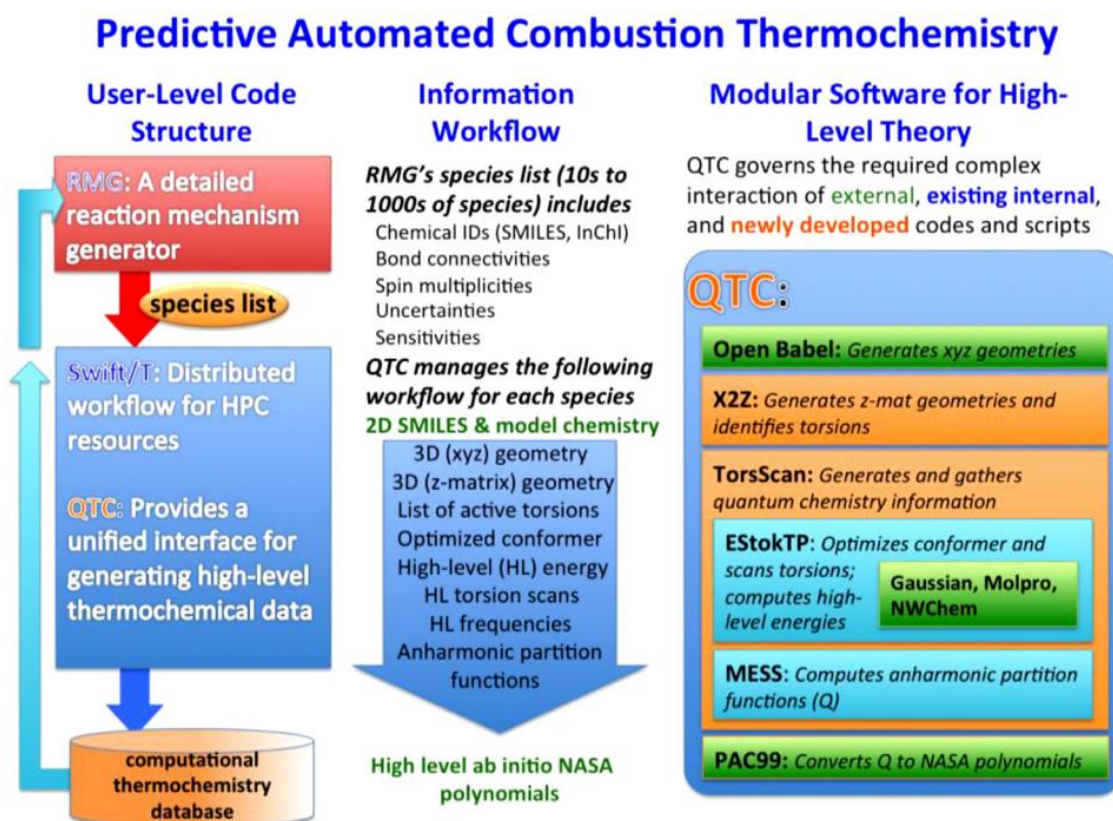


Figure 1.5: Flowchart of PACT software package [59]

1.3.4 KinBot

KinBot [61] is a Python code aimed to the automatic characterization of stationary points on Potential Energy Surfaces and successive Master Equation calculations.

Search of stationary points on PESs is conducted by iterative changes in reactants geometry to obtain an initial guess structure for reactive saddle points; such structures are then optimized using external quantum chemistry codes (such as Gaussian and Molpro). After the optimization, KinBot verifies the connectivity of the saddle point with the reactants and identifies the products using intrinsic reaction coordinate calculations. The analysis of products and saddle points structures

include conformational search of minimum energy structures, hindered rotors, and rotational symmetry numbers.

When connections between reactants, saddle points and products have been established, the code automatically creates input files for the solution of RRKM Master Equation, enabling the study of temperature and pressure rate dependencies.

The KinBot flowchart is presented in Figure (1.6).

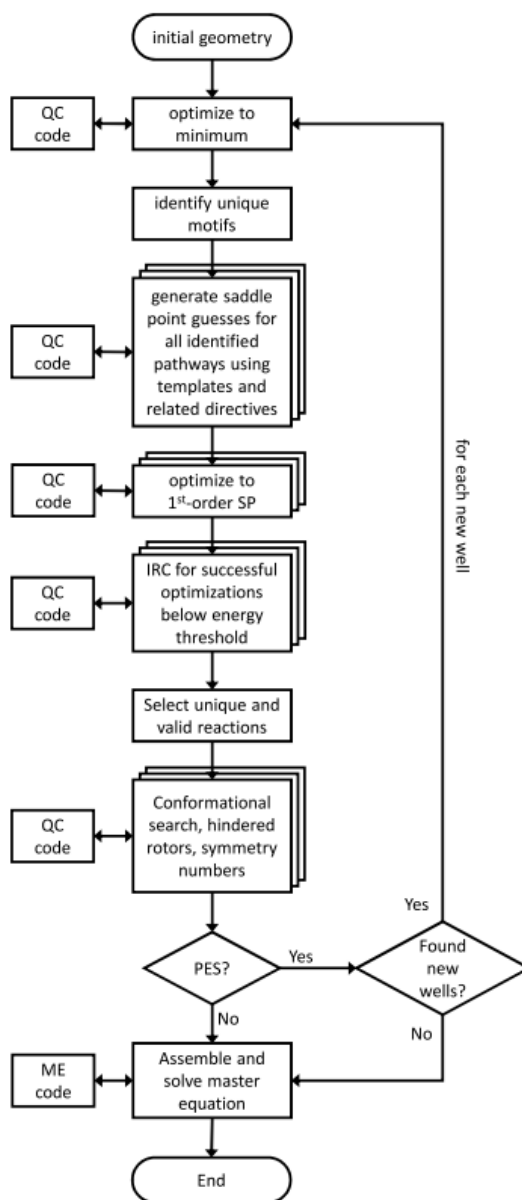


Figure 1.6: Flowchart of KinBot [61]

1.3.5 Genesys

Genesys [62] is a chemical kinetics software for automatic generation of reaction mechanisms. Genesys bases the creation of a mechanism on user-defined reaction families; this avoids the consideration of negligible reaction channels that would lead to a combinatorial explosion of possible species and untreatable mechanisms. Definition of reaction families is performed considering only a part of molecular bonds as suitable candidates for a chemical reaction, leaving other bonds untouched. After the analysis of all possible reaction families, a rule-based termination criterion is applied to prevent endless generation of new reactions and species. It applies a priori network reduction, based on chemical principles, which not only limits the size of the generated network, but also it guarantees that all the reactions involved are relevant for the mechanism.

After network generation has ended, a postprocessing thermochemistry subroutine, based on Benson group contribution method, is used for the estimation of thermochemical properties of all species generated. The flowchart of Benson group contribution algorithm implemented by Genesys is reported in Figure (1.7).

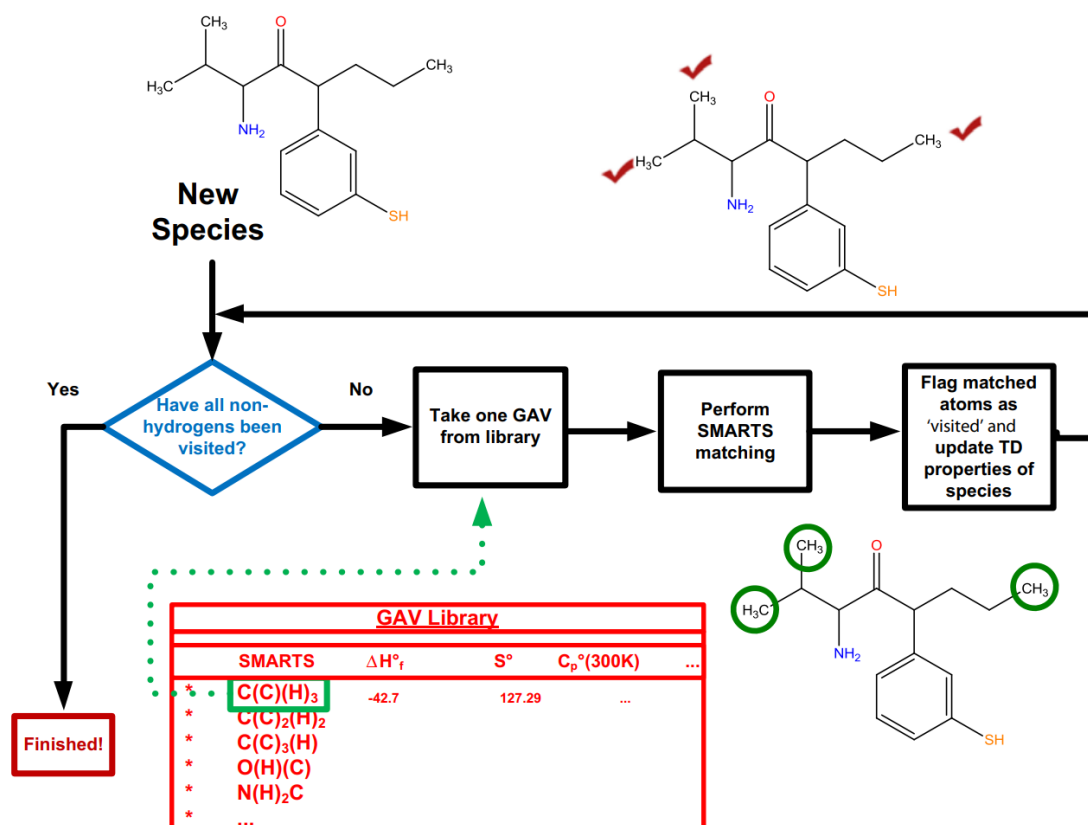


Figure 1.7: Flowchart of Genesys thermochemistry algorithm [62]

1.3.6 Arkane

Arkane [63] is a Python code aimed to the calculation of thermochemical properties and high-pressure limits as well as pressure-dependant reaction rate coefficients. Arkane is designed to work synergistically with RMG [51]. Arkane can process output files from quantum mechanical computations when used alone and augment the input with data from RMG database when it's coupled with RMG itself. It also provides complementary procedures for gas-phase model generation by performing pressure-dependent reaction rate computations. Since Arkane is closely connected to RMG, it can exploit RMG estimation methods to determine missing well energies and rate coefficient parameters.

Thermochemical parameters are estimated using the complete atomization scheme (the molecule is broken into its atomic components) with bond additivity corrections or, alternatively, a modified Isodesmic bond separation scheme. The considered molecule is not broken into each single bond; instead, a hypothetical reaction is constructed to preserve each element and bond present in the original molecule, searching for suitable reference molecules with known standard enthalpy of formation. An example of Isodesmic scheme implemented by Arkane for the estimation of standard enthalpy of formation is reported in Figure (1.8).

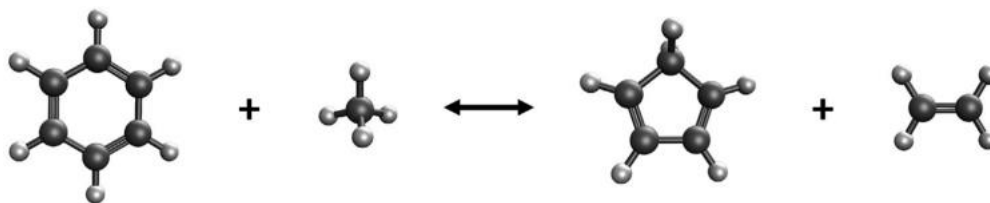


Figure 1.8: Arkane Isodesmic scheme for benzene [63]

1.5 Purpose of this work

This work has two main purposes: the automatization of input preparation for EStokTP code starting just from the InChI name of any chemical species and the development of a protocol able to predict thermochemical parameters in the form of CHEMKIN NASA polynomials, based on data post-processing of EStokTP calculations.

The first purpose is a further step in the automatization process on which EStokTP is based through a protocol named InChI2data: starting from a list of univocal identifiers, such as InChI [64], the Python code creates a series of directories *data* containing all the required blocks for an EStokTP simulation. For this part of the code (named InChI2data), InChI identifiers have been chosen, since they are unambiguous

schematization of a molecule, not having stereochemical identification problems, such as SMILES [65] identifiers.

The automatization part developed with InChI2data represent a building block for a successive Python code, named FragsGen, which, given a InChI identifier as input, produces a series of directories containing all possible fragmentations caused by the breakage of a bond (single, double, or triple), avoiding ring breakage over more probable breakage like H-abstractions.

The last part of this work is devoted to the implementation of a Python code, named CHEMTP (Connectivity Hierarchy Estimation Model for Thermochemical Parameters), that, starting from simulation data of EStokTP, construct CBH-0, CBH-1, CBH-2 rungs used for the estimation of $\Delta H^0(0 K)$ of the molecule of interest. Then $\Delta H^0(0 K)$ is corrected to obtain an estimation of $\Delta H^0(298.15 K)$.

Exploiting a subroutine that produces values of $C_p^0(T)$ and $S^0(T)$ (corrected for the presence of possible hinder rotors) over a selected interval of temperature, a non-linear regression is performed to obtain a_0 to a_4 from (1.3), for both low and high temperature intervals. The split temperature is determined automatically by the code to have the minimum estimation difference using low and high temperature coefficients, at the middle boundary. The last two coefficients for both high and low temperature are calculated using the explicit expression derived from (1.4) and (1.5).

After the estimation of the entire set of 14 coefficients, they are formatted and saved in the CHEMKIN format, adding information as the molecular formula, the phase of the molecule (for this code, aimed mainly to combustion and atmospheric kinetics, gas is chosen as standard phase), the low boundary temperature, the high boundary temperature, and the split temperature. A 15th coefficient is added, and it corresponds to the electronic energy at the highest level of theory available from the simulations.

2. Methods

Chapter two initially describes the structure of the EStokTP input files, all contained in a subdirectory called **data**, and discusses which steps are necessary for the automatization of the creation of the subdirectory **data** and which files require user modification. Then the algorithms implemented in the codes developed in the context of the present work are examined: InChI2data for the automatic generation of **data** input files, FragsGen for the automatic generation of possible fragmentation products, consequential to a bond breakage, and CHEMTP for the estimation of thermochemical parameters, which relies on CBH method for the estimation of $\Delta H^0(0 K)$. The Python version used in the present work for the construction of the codes is 3.6.8.

2.1 EStokTP input files

A general analysis of EStokTP input is given in the following subsections [66]. The code InChI2data exploits pre-compiled templates created by the user and adjusts their syntax to fulfil the input requirements of EStokTP. All input files are simple text files that are stored in the subdirectory **./data**. The final aim is the generation of the input starting just from the name of the chemical species of interest.

Only part of the generation of the input **data** directory was automatized; the code is designed to generate the XXXX.dat input files (which contain specifications on either reactants, products or transition states), and correct a series of parameters in different files (such as the species molecular weight in the header file for the Master Equation Solver *me_head.dat*), but the definition of which modules that should be used (defined in *estoktp.dat*) or the level of theory to be implemented (defined in *theory.dat*) is up to the user. Such input files are stored in a known template subdirectory and modified by the user, when needed.

2.1.1 estoktp.dat

The file *estoktp.dat* contains all information required to perform an EStokTP job, in the form of calls to its different modules. The file is divided in two sequential parts. From *estoktp.dat* the code interprets the type of job required and which reactive chemical species is being treated and the characteristics of the computational environment used for the simulation.

The type of job required specifies which kind of calculations EStokTP needs to run; for example, the type of reaction, through the keyword **ReactionType**, is specified in *estoktp.dat*. Also van der Waals wells can be automatically searched using the keywords **Wellr** (for reactant side) or **Wellp** (for product side). Quantum tunnelling can also be explicitly treated, using the keyword **MdTunnel** or excluded a priori via keyword **NoTunnel**. **Debug** keyword specifies the writing in the output; it helps with debugging procedure, as it determines how long will the output files be (usually set as “debug 2” for dense output). The **Recover** keyword recovers the output from an interrupted calculation after generating output prior to post-processing it; this is particularly useful for high level calculations if time-limited calculations are set by the HPC used.

The second block contained in *estoktp.dat* specifies the modules called in the EStokTP computation. It is used to list which type of reactive species needs to be optimized and at which level of theory (specified in *theory.dat*, section 2.1.3). Level 0 optimizations are specified by means of the keyword **Opt_XXXX** where **XXXX** can be **Reac1**, **Reac2**, **Prod1**, **Prod2**, **WellR**, **WellP**; they are performed at level of theory 0. Structural parameters are stored in the **./output** subdirectory as **XXXX_opt.out**; Cartesian coordinates of optimized structures are saved in the subdirectory **./geoms** as **XXXX_YY.xyz**, where **YY** is the number of structure found by Monte Carlo sampling (01, 02, 03 and so on). Transition state calculation are performed through two different keywords: **Grid_Opt_TS** performs a grid scan along a distance coordinate to determine a guess for TS search that will be used for successive calculations; the highest energy structure is saved in subdirectory **./output** as **ts_opt.out**. A log file of this calculation is saved in **./output** as **grid_opt.out**. If a transition state structure needs to be computed at level 0, keyword **Opt_TS_0** is used; the computed structure is saved in subdirectory **./output** as **ts_opt.out**. Transition state rotational conformers are searched using a Monte Carlo approach along the selected dihedral using the keyword **TauO_TS**, saving the results in **./output/ts_opt.out** for successive calculations. Exploiting level 0 results as input, level 1 calculations are carried out (as for level 0, level 1 theory is specified in *theory.dat*). The minimum energy structure can be determine using the keyword **Opt_XXXX_1** where **XXXX** can be **Reac1**, **Reac2**, **Prod1**, **Prod2**, **WellR**, **WellP**, **TS**; structural parameters are saved in **./output/XXXX_opt.out**, overwriting level 0 structural parameters results. Cartesian coordinates of the optimized structure are saved in the subdirectory **./geoms** as **XXXX_11.xyz**. Level 1 energies (in Hartree) are saved in the second line of **XXXX_11.xyz**. Frequencies are also calculated and, if no hindered rotors are specified in the species data file, they can be used for the computation of partition functions and the estimation of thermochemical parameters and rate constants. If hindered rotors are specified in the data file, a subsequent

module, either **1dTau_XXXX** or **MdTau_XXXX**, needs to be called first. Zero point energy are computed and saved in the subdirectory **./me_files** as **XXXX_zpe.me** for successive Master Equation calculations. The **1dTau_XXXX** module, where **XXXX** can be **Reac1**, **Reac2**, **Prod1**, **Prod2**, **WellR**, **WellP**, **TS**, performs one dimensional hindered rotor scans along the dihedral coordinates specified in the species data file. Log files are saved in the subdirectory **./output** as **XXXX_hr.out**. The calculated 1D potential energy surface is saved in the subdirectory **./me_files** as **XXXX_hr.me**, while the Hessian matrix and the projected frequencies are saved as **./me_files/XXXX_fr.me**. **MdTau_XXXX** performs a multidimensional hindered rotor scan along the dihedral coordinates specified in the species data file. At present, only one multidimensional hindered rotor scan is allowed for a molecule to preserve the correct coupling between internal and external rotational motions as implemented in the Mess Master Equation solver. Rotational and optical symmetry number are determined using the module **Symm_XXXX**. It exploits the `symmetry_number` code contained in the Mess Master Equation solver, by changing the tolerance from 0.001 to 0.07 for better results; then optical isomers are searched through a rotational scan of the hindered rotor PESs. The species external optical isomer number is then divided by the number of rotational optical isomers found. High level energy calculations are performed with the module **HL_XXXX**, which uses the geometry found in level 1 calculations; the output of high level calculations is saved in the subdirectory **./hl_logs** as **XXXX_molpro.out** if Molpro is adopted or **XXXX_g09.out** if Gaussian is adopted. The **kTP** module generates, if the keyword *reaction* and the specification of a valid reaction type are present in *estoktp.dat*, the file *me_ktp.inp*, which is the input for the MESS solver; all data required for this type of calculation are stored in the subdirectory **./me_files** and assembled in the directory **./me_blocks** by the **kTP** module (after successive calculations at level 0 and level 1). The output file, named **rate.out**, is saved in the subdirectory **./output**.

The last four lines of *estoktp.dat* define the computational environment. The first line contains the cores (individual processing units that can independently execute operations within a central processing unit) used for low (level 0 and 1) and high levels calculations; the second is a comment line; the third line contains the memory requested for low and high levels calculations, expressed in Mega words (MW); the fourth line is a comment line.

An example of *estoktp.dat* input file for the computation of level 0, level 1, 1D hindered rotor, high level, symmetry and kTP modules of ethane is reported below. Since ethane is the only molecule of interest, the computation is just a single well optimization and single point calculation, with explicit consideration of one vibrational degree of freedom as hindered rotation. Ethane is assigned to **Reac1** by default if no reaction type is defined.

```
C2H6
Debug 2
! blank line !
Opt_Reac1
Opt_Reac1_1
1dTau_Reac1
HL_Reac1
Symm_reac1
kTP
! blank line !
End
48,4
numprocl1,numproch1
800MW 1000MW
gmeml1 gmemh1
```

As mentioned, the code can not decide which module should be included in *estoktp.dat*, nor the amount of memory needed for low and high levels calculations. The template *estoktp.dat* must be modified by the user depending on the aim of the study and the computational environment in which one operates.

2.1.2 Species data file (XXXX.dat)

A total of seven species can be included in a single EStokTP job: two reactants (*reac1.dat* and *reac2.dat*), two products (*prod1.dat* and *prod2.dat*), two van der Waals wells (*wellr.dat* and *wellp.dat*) and a transition state (*ts.dat*). The code InChI2data is designed to generate by default an input file of type *reac1.dat*: this is not a limiting factor since the definition of reactant 1, reactant 2, product 1 and product 2 is arbitrary, and the structure optimization is requested and conducted in the same manner for all four kinds of species. A single EStokTP job optimizing *reac1*, *reac2*, *prod1* and *prod2* in series is completely equivalent to four parallel jobs, named all *reac1*; the same information is obtained and can be post-processed by the user, deciding if a species should act as reactant or product, depending on the specific case. At present the second option is exploited, reducing the computational time by parallelizing four different jobs. The *wellr.dat* and *wellp.dat* are trivial input files: they contain only the following keywords, *nosmp*, *ntau*, *nhind*, *charge*, *symmetryfactor*, and *nelec*, followed by the respective text blocks, if the *findgeom* option is associated to the *wellr* and *wellp* keywords (which is recommended). The

TS, not having a unique identifier, is not treated by InChI2data and so the *ts.dat* needs to be written by the user depending on the type of reaction of interest.

The input file *reac1.dat* contains a series of blocks, with no specific order imposed. After the block presentation, a complete example of ethane *reac1.dat* is reported.

- **nosmp** block: specifies the number of guesses generated in the Monte Carlo sampling, the geometric (in degrees) and energetic (in Hartree) thresholds for the stochastic search of minimum conformation geometry. A **nosmp** block requesting 6 sampling points with a geometric and energetic threshold of 1.0 and 0.00001, respectively, is reported in the complete example.
- **ntau** block: specifies the number and names of the coordinates stochastically sampled to search for the absolute minimum energy structure. The coordinates are usually dihedral angles, defined in the Z-matrix inside the **charge** block, but can also contain distance or planar angle coordinates. The block contains the number coordinates analysed and the name and interval of each coordinate analysed. In the complete example is reported a **ntau** block containing one coordinate (dih6, the sixth dihedral angle), analysed in a range of 0-360 degrees.
- **nhind** block: specifies the number and names of dihedral coordinates treated as hindered rotors. The total number of hindered rotors is specified, along with a list of names, interval of interests, number of points analysed within the interval and periodicity; the periodicity is specified to speed up hindered rotor calculations of groups presenting a symmetry (e.g., the periodicity of methyl groups is 3); in this way the hindered rotor analysis for a methyl group is done only for 120 degrees, and not 360. To achieve this, also the scanning interval must be defined as starting from 0 degrees to up to 120 degrees. The automatic generating procedure avoids this type of treatment for symmetries since for 1D hindered rotors the computational cost is not very large and so this approximation can be avoided. In the complete example, **nhind** block requires the scan of one dihedral angle (dih6) from 0 to 360 degrees, for 12 points with periodicity equal to 1.
- **natom** block: specifies the number of atoms contained in the molecule and its linearity; three values are specified. The total number of atoms, excluding dummy atoms; the total number of atoms, including dummy atoms; an index for linearity (equal to 0 for non-linear molecule, equal to 1 for linear molecule). The complete example, being ethane a non-linear molecule, thus not requesting definition of dummy atoms, reports 8 8 0 as **natom** block.
- **charge** block: specifies the charge and spin multiplicity of the molecule, followed by the Z-matrix of the molecule. Since ethane is not a charged

molecule nor a radical, its charge and spin multiplicity correspond to 0 and 1, respectively. This block is reported in the complete example.

- **intcoor** block: contains the list of first guess values for distances, planar angles and dihedral angles specified in the Z-matrix contained in the **charge** block.
- **symmetryfactor** block: specifies the global symmetry factor of the molecule, defined as the ratio between the external rotational symmetry number and the external optical symmetry number. Although it can be specified, it gets recalculated when the **Symmetry** block is called, so it is good practice to initially set it to unity, remembering to include **Symm_reac1** block in *estoktp.dat*.
- **nelec** block: specify the number of electronic states for the considered species and their multiplicity and energy relative to the ground state, including the ground state.

The complete example of *react1.dat* for an ethane molecule, generated using InChI2data code, is reported below:

```

nosmp dthresh ethresh
6 1.0 0.00001

ntau
1
→ name and sampling interval
dih6 0. 360.

Nhind
1
→namehind,hindmn,hindmx,nhindsteps
dih6 0. 360. 12 1

natom natomt ilin
8 8 0
charge spin atomlabel
2 1
c1
c2 c1 cc2
h3 c1 hc3 c2 hcc3
h4 c1 hc4 c2 hcc4 h3 dih4
h5 c1 hc5 c2 hcc5 h3 dih5
h6 c2 hc6 c1 hcc6 h3 dih6
h7 c2 hc7 c1 hcc7 h6 dih7

```

```
h8      c2      hc8      c1      hcc8     h6      dih8
```

```
intcoor
```

```
cc2     1.5037358220500001
```

```
hc3     1.11616129548
```

```
hc4     1.10674723665
```

```
hc5     1.0979734819900002
```

```
hc6     1.1268295038
```

```
hc7     1.09971447432
```

```
hc8     1.1064773563799999
```

```
hcc3    107.196
```

```
hcc4    107.971
```

```
hcc5    114.133
```

```
hcc6    106.172
```

```
hcc7    113.524
```

```
hcc8    113.467
```

```
dih4    114.161
```

```
dih5    241.969
```

```
dih7    244.206
```

```
dih8    121.653
```

```
SymmetryFactor
```

```
1.
```

```
nelec
```

```
1
```

```
0. 1.
```

The generation of *reac1.dat* is completely automatized by InChI2data, starting only from the InChI identifier of the requested molecule.

2.1.3 theory.dat

The *theory.dat* file specifies the level of theory used for each of the blocks contained in *estoktp.dat*. Each block contained in *theory.dat* reports the module keyword (**level0**, **level1**, **hind_rotor**, **symmetry** and so on) followed by the code used for the calculations: if Gaussian is used, the keyword **g09** or **g16** (depending on the version used) can directly follow the module keyword; if Molpro is used, the keyword **molpro** follows the module call, together with a separate input file that specifies the level of theory (section 2.1.5). A series of additional keywords, if Gaussian is used, are directly specified in *theory.dat*, requesting the optimization in internal coordinates

(**opt=internal**), the frequency calculation (**freq**), the deactivation of Gaussian symmetry module (**nosym**) and so on.

An example of *theory.dat* specifying the level of theory of level 0, level 1, hindered rotor calculations, symmetry calculations and high-level calculations (in Molpro) is reported below.

```
Level0 g09
uwb97xd/6-311+g(d,p)  opt=(internal,MaxCycles=50)
int=ultrafine nosym
```

```
level1 g09
uwb97xd/jun-cc-pvtz  opt=(internal,MaxCycles=50)
int=ultrafine nosym freq
```

```
hind_rotor g09
b3lyp/6-311+g(d,p)  opt=internal
int=ultrafine nosym
```

```
symmetry g09
b3lyp/6-311+g(d,p)  opt=internal
int=ultrafine nosym
```

```
hlevel molpro
```

```
End
```

As for *estoktp.dat*, the code can not determine a universal level of theory adapted to all kinds of calculations; thus *theory.dat* should be modified based on the user necessities and it is then copied by the code InChI2data developed in this work in each subdirectory **./data** created.

2.1.4 *me_head.dat*

Input file *me_head.dat* contains the input for master equation simulations; it should always be contained in subdirectory **./data**. As suggested by [66], *me_head.dat* is self-explanatory and modified by the user at needed; syntax explanation is provided by [67].

The only automatization performed by InChI2data is the automatic change of the buffer gas molecule mass, expressed in amu, coherent with the molecule expressed

by the InChI identifier. Other changes such as temperature and pressure values should be performed by the user.

2.1.5 Molpro theory files

If **molpro** keyword is present in *theory.dat* (as for **hlevel** in section (2.1.3) example), a different input file must be specified. The name of the input file should be the module contained in *theory.dat*, followed by *_molpro.dat*. For example, a level0 optimization should be specified using the file *level0_molpro.dat*; if more species are optimized in the same job of EStokTP, Molpro input file should contain the name of the specific species (e.g. *level0_reac1_molpro.dat* or *level0_prod1_molpro.dat*).

The typical EStokTP job uses Gaussian for level 0 and level 1 calculations, while high-level calculations are performed by Molpro. Thus, the only Molpro input file specified is *hl_molpro.dat*. It performs CCSD(T) calculations followed by MP2 corrections for basis set size and CCSD(T) core correlation corrections. An example of *hl_molpro.dat* can be found in [67].

Being a standardised file, *hl_molpro.dat* is simply reported by the code in *./data* subdirectory.

2.2 RDKit

RDKit [68] is an open-source toolkit for cheminformatics. It provides an intuitive environment for analysis and modification of chemical molecules, usually provided to the program through identifiers such as InChI or SMILES; the information provided by the codes developed in the present work relies mostly on RDKit tools. Characteristics like bond order, presence of a radical, presence of a ring are obtained using RDKit. Also, the conversion of InChI identifier to Mol file (containing the Cartesian representation of the molecule) is performed using RDKit. The RDKit version used in the present work is 2023.03.1.

InChI2data relies on RDKit for generation of Cartesian coordinates, determination of number of atoms and determination of mass for a molecule.

FragGen relies on RDKit for identification of the fragments from a bond breakage, generation of structures in Cartesian coordinates for the fragments and identification of rings.

CHEMTP relies on RDKit for the identification of radicals, bond orders, saturation numbers and atom numeration.

2.3 InChI and SMILES

InChI and SMILES are structure-based chemical identifier. They are essential for computational chemistry databases because they describe in the most unique way possible chemical compounds. Programs like RDKit can extract stereochemical information from this type of identifiers.

The structure and the purpose of InChI and SMILES are quite different.

An InChI (acronymous of International Chemical Identifier) is a unique machine-readable string, able to store precise information about chemical structures. The main downside of InChI is not being user-friendly since the structure is not easily readable. The strength of InChI identifiers is that they are unique, i.e., there is just one possible molecule-InChI coupling. Because of this characteristic, they have been chosen as starting point for the Python codes developed in the present work.

An InChI presents a layered structure; each layer is separated by a slash symbol. The version of the InChI identifier used is determined by the first layer (1S or 2S); the second layer contains the chemical formula of the compound, while the third layer defines the non-hydrogen atom connections. The fourth layer determines the connectivity of the hydrogen atoms. Additional sublayers define the charge and the stereochemistry of the molecule.

As example, the InChI of ethane is reported: InChI=1S/C2H6/c1-2/h1-2H3.

SMILES (acronymous of Simplified Molecular Input Line Entry System) is a chemical descriptor in the form of line notation, using short ASCII strings. The SMILES form has been developed to have an intermediate form between the human readable IUPAC form and the computer readable Wiswesser line notation (WLN) [69]. They provide a concise representation of molecules and enable easy generation (even by hand). Their main limitation is in the describing complicated 3D structures (e.g., highly branched); also, they are not unique, and this can lead to multiple definition of the same molecule in large databases, even if a canonical representation exists.

In SMILES notation, every non-hydrogen atom is identified by its atomic symbol in (uppercase for atoms not contained in aromatic compounds, lowercase vice versa). Hydrogens are not specified if the atom is not a radical.

If an atom is not a radical, its saturated with hydrogens by default (e.g., C represents methane CH₄, O represents water H₂O); elementary compounds, radicals and ions have hydrogens and charges specified in squared brackets (e.g., methyl-radical group is [CH3], atomic nitrogen is [N], sodium cation is [Na⁺]).

Double and triple bonds are identified by = and # symbols, respectively (e.g., C=C is ethylene, C#C is acetylene). Although single bonds can be identified by – symbol, it's usually omitted to keep the notation compact.

Branching is defined by round brackets, and the branched chain refers to the last non-hydrogen atom before brackets (e.g., CC(C)C is isobutane).

Cyclic molecules are represented by starting from an arbitrary non-hydrogen atom and following the structure until the atom connected to the starting position is reached (e.g., C1CCCCC1 is cyclohexane, c1ccccc1 is benzene).

Because of their simple representation of small molecules, SMILES have been selected for the representation of the thermodynamic reactions generated by CHEMTP in the context of CBH estimation of $\Delta H^0(0 K)$ and for the reference species database.

Both forms are saved in `./data` subdirectory in a file `name.dat`, generated using InChI2data.

2.4 InChI2data

InChI2data is a Python code, developed in the context of the present work, for the automatic generation of the input subdirectory `./data` suitable for a single species (identified as `react1`) job in EStokTP. Two versions of InChI2data exist: for single and multiple `data` subdirectory generation. The single generation version is used for quick simulation of a chemical species, even without checking the input files `theory.dat` or `estoktp.dat` from the templates directory and changing them after the creation of `data`. A preliminary check of `theory.dat` and `estoktp.dat` is nevertheless recommended before running EStokTP jobs. The extension to multiple data generation is immediate, with the generation of a series of directories, in the form `./dir_num/data`, with `dir_num` equal to 0001, 0002 and so on.

The single data generation version of InChI2data is presented in detail, while the extension to multiple data generation is discussed at the end of the section.

2.4.1 XYZ format structure generation

The first step is the generation of the Cartesian coordinates from the InChI identifier. They are the representation of atoms distribution in 2D or 3D space; the Cartesian structure file contains the number of atoms (without dummy atoms) as first line, while the second line is a comment line. The successive lines are occupied by the spatial definition of atoms, one for each line. A single line contains the atomic symbol

of the atom and three numbers, representing the x , y , and z coordinate of the atomic centre with respect to an arbitrary origin.

Using RDKit, the InChI identifier, from the input file *inchi_file.dat*, is converted into a Mol file. A Mol file is a widely used format for representing molecular structures, containing information about molecule atoms, bonds, connectivity, charges, and stereochemistry. The Mol file of ethane is reported below:

```

10  9  0  0  0  0          999 v2000
    0.0000    0.0000    0.0000 C  0  0  0  0  0  0
    1.0186    0.0000   -0.3968 H  0  0  0  0  0  0
   -0.5097   -0.8818   -0.3968 H  0  0  0  0  0  0
   -0.5088    0.8825   -0.3966 H  0  0  0  0  0  0
    0.0000    0.0000    1.5261 C  0  0  0  0  0  0
   -1.0186    0.0015    1.9228 H  0  0  0  0  0  0
    0.5101    0.8818    1.9227 H  0  0  0  0  0  0
    0.5084   -0.8825    1.9232 H  0  0  0  0  0  0
  1  2  1  0
  2  3  1  0
  2  4  1  0
  2  5  1  0
  2  6  1  0
  2  7  1  0
  2  8  1  0
  2  9  1  0

```

M END

The information of interest is highlighted in bold; these lines are extracted from the Mol file, placing the atomic symbol on the left side of the line, and putting as first line the number of atoms, plus a blank second line.

By default, RDKit generates Mol files in 2D coordinates (3D coordinates projected on x - y plane, with all z coordinates set to zero) and without hydrogens; these options are changed to have a fully tri-dimensional representation of the entire set of atoms composing the molecule. The obtained Cartesian structure is reported below:

```

8
C    0.00000    0.00000    0.00000
H    1.01866    0.00000   -0.39684
H   -0.50975   -0.88187   -0.39688
H   -0.50887    0.88258   -0.39661
C    0.00000    0.00000    1.52614

```

H	-1.01868	0.00155	1.92287
H	0.51013	0.88186	1.92272
H	0.50845	-0.88255	1.92321

2.4.2 Cartesian coordinates conversion to Z-matrix

The next step is the conversion of the Cartesian coordinates to Z-matrix. The Z-matrix is a geometrical representation of the molecule in terms of internal coordinates: this means that the definition of each atom is made on the base of other atoms present in the molecule. The optimizations are done in terms of internal coordinates, without relying on external coordinate systems. The typical definition of an atom consists in setting distance, planar angle, and dihedral angle with respect to three different atoms. These atoms are usually one or two bonds distant from the defined atom.

The first atom has no parameter definition, being the centre of the molecule internal coordinate system; the second atom has only a distance (from the first atom) defined; the third atom has distance and planar angle defined with respect to the other two atoms; starting from the fourth atom, the parameters defined will be three, the third being the dihedral angle. An example of Z-matrix is reported in the block **charge**, presented in section (2.1.2). A list of first guess values of distances, planar angles and dihedrals is also generated in this step. The construction of *reac1.dat* is conducted using the software *x2z* [70], a Python code that makes heuristic analysis of molecular bonding based on Cartesian information and produces electronic configurations, internal rotation structure and Z-matrix.

A convenient Z-matrix definition has firstly defined all the heavy atoms, creating a connected structure of heavy atoms only, and then the definition of hydrogens, saturating the heavy atoms according to the numeration assigned. *x2z* does not follow this rule for a randomly ordered Cartesian coordinates; instead, it orders the atoms in such a way of completing the atomic saturation of each atom defined before continuing in the Z-matrix construction. A Z-matrix of ethane, generated using its Cartesian structure from section (2.4.1), is reported below:

```

C1
H2  C1  R1
H3  C1  R2  H2  A2
H4  C1  R3  H2  A3  H3  D3
C5  C1  R4  H2  A4  H3  D4
H6  C5  R5  C1  A5  H2  D5
H7  C5  R6  C1  A6  H6  D6

```

```
H8  C5  R7  C1  A7  H6  D7
```

It clearly shows the definition of methyl-group hydrogens before the definition of the second carbon. Because of this definition, the Cartesian system is reordered to firstly define all the heavy atoms and then hydrogens, and only then passed to x2z code. The order of definition of heavy atoms is pseudo-randomly changed by switching two atoms at a time and the Z-matrix continuously regenerated, until a suitable form is obtained. The final form of the ethane Z-matrix is reported below:

```
C1
C2  C1  R1
H3  C1  R2  C2  A2
H4  C1  R3  C2  A3  H3  D3
H5  C1  R4  C2  A4  H3  D4
H6  C2  R5  C1  A5  H3  D5
H7  C2  R6  C1  A6  H6  D6
H8  C2  R7  C1  A7  H6  D7
```

The other blocks described in section (2.1.2) are also generated by x2z.

The Z-matrix quantities are also renamed to make them more intuitive, stating clearly the atoms defined by a distance or planar angle. The distance R_i is replaced by XYN, where X is the atom defined in the current line N, Y is the atom which defines the distance with respect to atom X, N is the atom number (e.g., the distance R6, defining atom H7 with respect to C2, becomes hc6). Following the same approach, planar angles are defined by three atoms (e.g., planar angle A6, defining atom H7 with respect to C2 and C1, becomes hcc6). The dihedral angles are simply defined as dihN, where N is the atom number (e.g., dihedral angle D6 becomes dih6).

The first guess value list and the dihedrals defined in the blocks **ntau** and **nhind** are renamed following the renaming procedure of the Z-matrix.

An example of renamed Z-matrix is reported below:

```
c1
c2  c1  cc2
h3  c1  hc3  c2  hcc3
h4  c1  hc4  c2  hcc4  h3  dih4
h5  c1  hc5  c2  hcc5  h3  dih5
h6  c2  hc6  c1  hcc6  h3  dih6
h7  c2  hc7  c1  hcc7  h6  dih7
h8  c2  hc8  c1  hcc8  h6  dih8
```

2.4.3 data subdirectory generation

Once the *reac1.dat* file has been successfully generated, the subdirectory **data** is created; as stated in section (2.1), *theory.dat*, *estoktp.dat* and *hl_molpro.dat* files are copied in **data**, being modified in advance by the user as needed. The *me_head.dat* is automatically corrected for the molecular weight (expressed in amu) and copied in **data**. An additional file *name.dat*, containing InChI and SMILES identifiers of the molecule is created and stored in **data** and used for successive calculations; the SMILES identifier is generated by conversion of the Mol file, exploiting the RDKit functions. Even if the SMILES identifier is not canonical, its purpose is to give an indication of which molecule is modelled: this is useful when a high number of **data** subdirectories are created, to keep track of which molecules have been generated, without having to check each InChI identifier from external databases.

Multiple generation of **data** subdirectories is exploited by passing to InChI2data a list of InChIs through an input file *inchi_file.dat*; The text file *inchi_file.dat* contains the list of InChIs, one per line, without blank lines. The code generates a series of directories (0001, 0002...), each one containing a **data** subdirectory with the proper *reac1.dat*, *name.dat* and *me_head.dat*; the directory definition order follows the order specified in *inchi_file.dat*. The files *theory.dat*, *estoktp.dat* and *hl_molpro.dat* are the same for every directory created. The flow diagram of InChI2data algorithm is reported below:

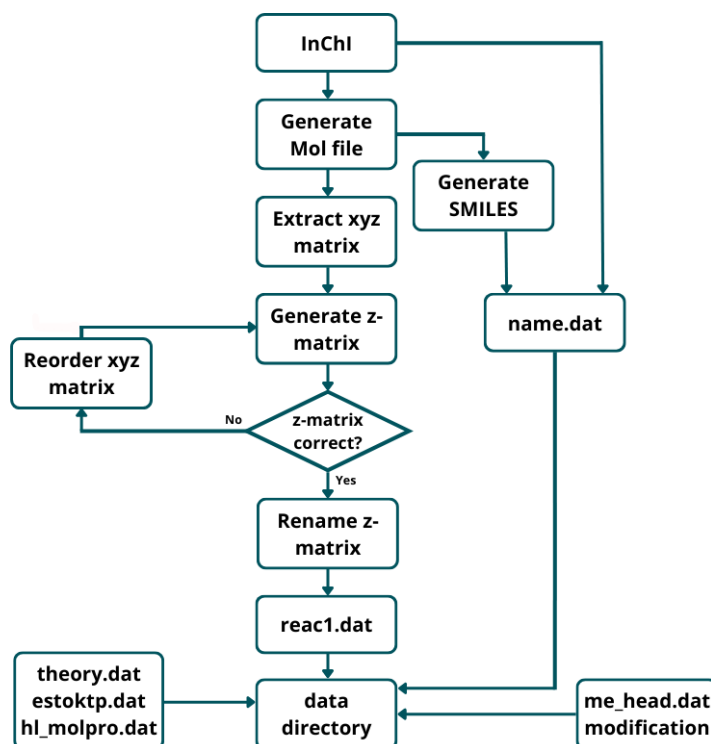


Figure 2.1: Flowchart of InChI2data algorithm

2.4.4 InChI2data user manual

The present sub-section introduces the utilization procedure of the code InChI2data by the user.

Single **data** directory generation:

- Move to the working directory where the **data** directory will be created.
- Create a text file called *inchi_file.dat*; in this file copy the InChI identifier in the first line, without spaces or blank lines.
- Change the line of the code InChI2data which defines the path of the model data files (this information is hard-coded; therefore, it has to be changed in the script any time the source of the template input files is changed). The model data files are the text files *estoktp.dat*, *theory.dat*, *me_head.dat* and *hl_molpro.dat*.
- Modify the template data files based on the requested calculation (level of theory, calculation blocks etc.).
- Run the code InChI2data (single **data** generation version).

The **data** directory created in this way will contain *estoktp.dat*, *theory.dat* and *hl_molpro.dat* copied from the model data path, and the updated *me_head.dat*, *reac1.dat* and *name.dat* files. The directory **data** is already ready for an EStokTP job to start.

Multiple **data** directory generation:

- Move to the working directory where the *./XXXX/data* directories will be created (XXXX are 0001, 0002, 0003 and so on).
- Create a text file called *inchi_file.dat*; in this file copy the InChI identifiers, one per line, without spaces or blank lines.
- Change the line of the code InChI2data which defines the path of the model data files (this information is hard-coded; therefore, it has to be changed in the script any time the source of the template input files is changed). The model data files are the text files *estoktp.dat*, *theory.dat*, *me_head.dat* and *hl_molpro.dat*.
- Modify the template data files based on the requested calculation (level of theory, calculation blocks etc.).
- Run the code InChI2data (multiple **data** generation version).

The directories created in this way are 0001, 0002, 0003 and so on, following the order of the InChI identifiers contained in *inchi_file.dat*. Each directory XXXX contains a directory **data** with *estoktp.dat*, *theory.dat* and *hl_molpro.dat* copied from the model data path, and the updated *me_head.dat*, *reac1.dat* and *name.dat* files. Each *me_head.dat*, *reac1.dat* and *name.dat* refer to the XXXX InChI contained in *inchi_file.dat*. Each directory XXXX is already ready for an EStokTP job to start.

2.5 FragsGen

FragGen is a Python code, developed in the context of the present work, for the automatic generation of fragmentation products of an initial molecule. The purpose of this code falls in the context of automatic exploration of Potential Energy Surfaces and determination of bond energies, making predictions of which bonds are more likely to break.

Since ring structures (aromatic or not) are far more stable than non-cyclic structures and because of rings breakage does not produce two molecules, the code is designed to break only bonds not contained in ring structures (i.e., the only breakage in a benzene molecule would be a C-H bond, while C-C bonds are not considered as candidates for bond breakage).

2.5.1 Bond breakage

The code analyses the structure of the molecule, provided by a single InChI identifier stored in the input file *inchi.dat*, using the tools provided by RDKit. After creation of a subdirectory **./fragmentation**, where the fragments will be stored, the code cycles through all bonds; for each bond, if not contained in a ring, two Mol files, corresponding to the fragments that would result from the n^{th} bond breakage, are created. From the Mol files, the SMILES of each fragment are generated, along with the extraction of the Cartesian coordinates from each Mol file. InChI2data (section 2.4) is used for conversion of the Cartesian representation of the molecule in Z-matrix and creation of two **data** subdirectories. The n^{th} fragmentation molecules are saved in the subdirectory **./fragmentation/frags_N**, where N is the n^{th} fragmentation number. A **data** subdirectory is created also from the original molecule, and is stored in the subdirectory **./fragmentation/originalmol**.

The **data** subdirectories of each fragmentation are stored in the subdirectory **./fragmentation/frags_N/fragment1** and **./fragmentation/frags_N/fragment2** for the first and second fragment, respectively, generated by the n^{th} fragmentation.

In order to understand which fragment is stored in every subdirectories, an output file *smilesfrags.out* is created in **./fragmentation**, which collects the two SMILES corresponding to the molecules created by every bond breakage considered. Even if this step does not allow the generation of canonical SMILES due to RDKit conversion of Mol files generated by bond fragmentation (e.g., canonical representation of methyl-radical group would be [CH3], while the SMILES created with this procedure is * C), for relatively small molecules is enough to clearly understand the molecules generated.

2.5.2 Equivalent fragmentations

Taking as example the structure of pentane (Figure 2.2) it is clear that a series of fragmentations are equivalent, giving the same products. Bond breakages like C(1)-C(2) and C(4)-C(5) give the same radicals (i.e., methyl- and n-propyl-radical). This is automatically managed by the code, ensuring that only one subdirectory per bond breakage is created, even for multiple equivalent fragmentation. This is done by keeping track of the SMILES couples generated at each fragmentation and comparing the n^{th} SMILES couple with the couples already created. The number of equivalent fragmentations is stored in every subdirectory `./fragmentation/frags_N` in the output file `num_equivfrags.out`.

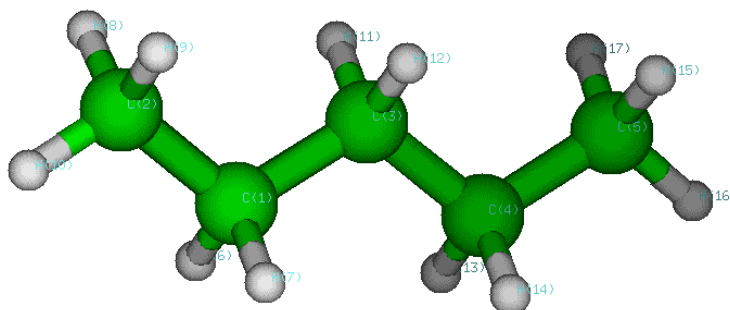


Figure 2.2: Pentane structure

Once all possible fragmentations have been analysed an output file `nfrags.out`, containing the number of fragmentations, counting multiple equivalent fragmentations once, is created in subdirectory `./fragmentation`.

The code runs automatically all the simulations prepared.

2.5.3 Bond energy calculation

Bond energy can be calculated as difference between electronic energy plus ZPE of fragmentation products and original molecule.

$$E_{bond} = \left[\sum_{i=1}^2 (E_{el} + ZPE)_i \right] - (E_{el} + ZPE)_{orig.mol.} \quad (2.1)$$

Bond energy estimation is done at level 1 and high-level theory, producing two estimated values. They are saved in the subdirectory `./fragmentation` in the output file `bondenergies.out` for level 1 and `bondenergies_hl.out` for high level. To compute the bond energy, all three molecules jobs should be ended successfully; if the original molecule simulation has failed at level 1, `bondenergies.out` and `bondenergies_hl.out` will

contain an error message, while if the high level has failed, level 1 energies will be successfully computed, while *bondenergies_hl.out* will contain an error message. If the original molecule has been computed successfully both at level 1 and high level, the code checks for fragment 1 and fragment 2 simulations.

At level 1, if both fragments have been successfully simulated, the energy is calculated, while if fragment 1, or fragment 2, or both, have failed, the output file *bondenergies.out* contains an indication error of which fragment(s) have failed. For example, if both simulations have failed, the corresponding error would be “failed 1 2”.

For high level estimation the procedure followed by the code is the same.

The flow diagram of FragsGen algorithm is reported below:

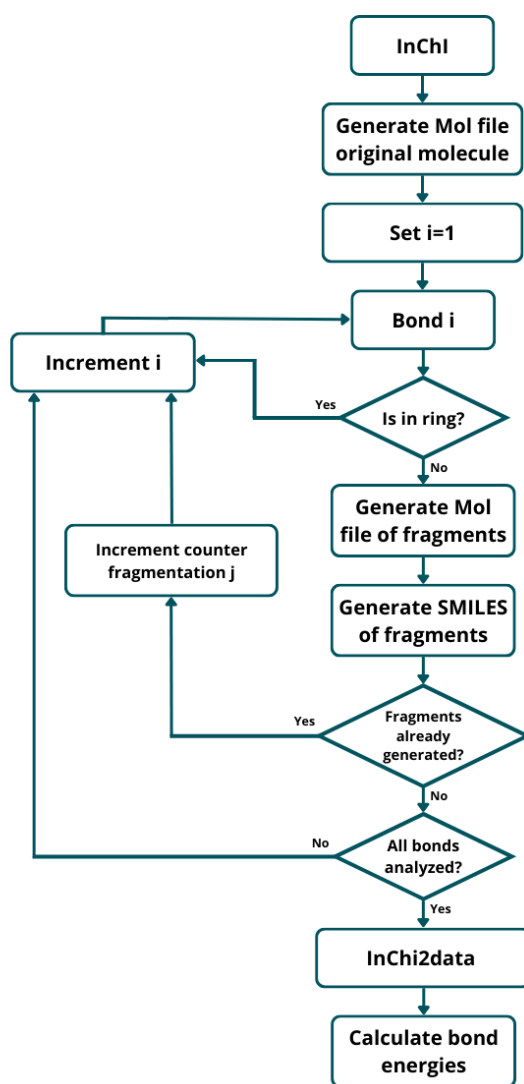


Figure 2.3: Flowchart of FragsGen algorithm

2.5.4 FragsGen user manual

The present subsection introduces the utilization procedure of the code FragsGen by the user.

- Move to the desired working directory where **fragmentation** directory will be created.
- Create a text file called *inchi.dat*; in this file, copy the single InChI identifier in the first line, without spaces or blank lines.
- Change the line of the code InChI2data which defines the path of the model data files (this information is hard-coded; therefore, it has to be changed in the script any time the source of the template input files is changed). The model data files are the text files *estoktp.dat*, *theory.dat*, *me_head.dat* and *hl_molpro.dat*.
- Modify the model data files based on the requested calculation (level of theory, calculation blocks etc.).
- Run the code FragsGen.

Inside the directory **fragmentation**, a series of directories frags_N (where N=1,2,3,4 etc.) are created; each frags_N contains two directories **./fragmentation/frags_N/fragmentX** (where X=1,2). Each directory **fragmentX** contains the **data** directory referred to the proper X fragment generated by the N fragmentation (generated using the code InChI2data (Section 2.4)). The code automatically submits all the requested calculations as EStokTP jobs.

Once all the calculations have ended, the second part of the code must be executed, which collects the results of the calculation needed for the computation of bond energies. They are stored in the **fragmentation** directory as *bondenergies.out* and *bondenergies_hl.out* for level 1 and high-level calculation, respectively. A list of computed fragments, in the form of non-canonical SMILES, is stored in the text file **./fragmentation/smilesfrags.out**. The collecting code must be executed inside the **fragmentation** directory.

2.6 CHEMTP

CHEMTP is a Python code for the estimation of thermochemical parameters in the form of NASA polynomials in CHEMKIN format. Temperature dependent specific heat $C_p^0(T)$ and entropy $S^0(T)$ estimation is based on the RRHO approximation and the derived translational, vibrational, and rotational contributions [39]. Specific heat and entropy contain contributions for the presence of internal motions, such as methyl-groups rotation; internal motions are treated as 1D hindered rotors [71]. The information is obtained from the output of the **1dTau_Reac1** module, presented in section (2.1.1).

There are no explicit ways of estimating enthalpy directly from molecular partition functions, so a different approach is applied. Rather than attempting to implement complicated electronic structure models, the usual procedure is the implementation of error cancellation methods. This type of models describes molecular quantities by relating them to single atoms/small atomic groups, relying on the error compensation in the approximated treatment of atom-atom interactions. Because a given method produces more or less consistent errors for each atom-atom interaction, the error cancellation leads to consistent estimation of $\Delta H^0(0 K)$.

Consider a generic reaction between reactants and products (Figure 2.4); the reaction energy is evaluated in terms of electronic and zero-point energy of products and electronic and zero-point energy of reactants. It represents the energy change to form the product(s) starting from the reactant(s). The reaction energy can be written also in terms of standard enthalpy of formation $\Delta H^0(0 K)$; this leads to an explicit expression:

$$\Delta H_R^0(0 K) = \sum_{prod} \Delta H_i^0(0 K) - \sum_{react} \Delta H_i^0(0 K) = \sum_{prod} (E_{El} + ZPE)_i - \sum_{react} (E_{El} + ZPE)_i \quad (2.2)$$

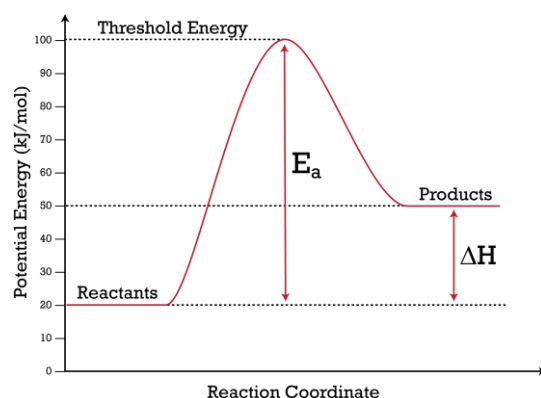


Figure 2.4: Potential Energy Surface

that allows the estimation of $\Delta H^0(0 K)$ of the parent molecule (PM) based on electronic and zero-point energies of PM itself and a series of reference molecules (RM), and RM standard enthalpy of formations. The only unknown left in equation (2.2) is the $\Delta H^0(0 K)$ of the PM, which is the objective of this code.

A series of corrections to the estimation of the electronic energy can be implemented; their complete implementation would lead to a complicated, even though more precise, algorithm. Tajti et al. [72] estimate a correlation energy correction at the CCSD(T) level of theory and extrapolation to the basis set limit, at least one order of magnitude higher than other plausible corrections. This type of adjustment is

computed in this work using the extrapolation scheme presented by Martin [73] in 1996. For level 1 computations no corrections are implemented. Other types of corrections were not considered. Because of this neglects, accurate values of electronic and ZPE and standard enthalpy of formation of RM are required.

The reactions exploited for the estimation of $\Delta H^0(0\text{ K})$ of the PM using equation (2.2) can be of different types, based on the scheme implemented. CHEMTP uses the CBH approach [44-47], since it offers the best compromise between accuracy and effort required for the computation of reference species, being composed at most by 5 heavy atoms. The CBH approach consists of three levels (also called rungs): CBH-0 rung (implementing the isogyric fragmentation scheme), CBH-1 rung (implementing the isodesmic bond separation scheme) and CBH-2 rung (implementing the hybridization-based homodesmotic scheme).

Information relative to electronic and zero-point energy of the PM are obtained from initial EStokTP jobs; the estimation can be carried out at two different levels. Level 1 estimation is done at ω B97X-D or B2PLYP-D3 level of theory and then extracting the value of electronic energy from *react1_l1.xyz* stored in subdirectory **./geoms** and ZPE stored in **./me_files/react1_zpe.me**, while at high level estimation requires the same ZPE as level 1 and electronic energy stored in **./me_files/react1_en.me**.

Information relative to electronic energy, ZPE and standard enthalpy of formation of RM are obtained from in-house built database, containing both level 1 and high-level reference information.

Once $\Delta H^0(0\text{ K})$ of the PM has been computed at the highest rung available, it is corrected to estimate $\Delta H^0(298.15\text{ K})$; this is necessary because molecular partition functions can not be computed at 0 K. The extrapolation scheme implemented by Ochterski [39] is based on enthalpy corrections of the atomic elements composing the PM; the contribution of each atom is obtained from purely experimental data for the heats of formation by Pople et al. [74].

After the calculation of $\Delta H^0(298.15\text{ K})$, the regression of $C_p^0(T)$ as a function of temperature is performed, allowing the computation of the high and low temperature a_0 to a_4 coefficients. High and low temperature a_6 are computed making them explicit from (1.5). Low temperature interval a_5 is computed making it explicit from (1.4), using $\Delta H^0(298.15\text{ K})$. H^0 in the high temperature interval is estimated to the same temperature of S^0 (2500 K), exploiting the relation between $H^0(T)$ and $C_p^0(T)$. Since there are two temperature intervals, the expression becomes:

$$H^0(2500\text{ K}) = \Delta H^0(298.15\text{ K}) + \int_{298.15}^{T_{split}} C_{P,lowT}^0(T) dT + \int_{T_{split}}^{2500} C_{P,highT}^0(T) dT \quad (2.3)$$

Expression (2.3) is computed analytically because $C_p^0(T)$ is already determined for both temperature intervals in terms of expression (1.3).

Coefficients a_0 to a_4 are estimated with a non-linear regression of equation (1.3); coefficients a_5 and a_6 are obtained rearranging equations (1.4) and (1.5). They are stored in the subdirectory `./thermo` in the output files `cp.out` and `S.out`.

$$\frac{C_p^0(T)}{R} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 \quad (1.3)$$

$$\frac{H^0(T)}{RT} = a_0 + \frac{1}{2} a_1 T + \frac{1}{3} a_2 T^2 + \frac{1}{4} a_3 T^3 + \frac{1}{5} a_4 T^4 + \frac{a_5}{T} \quad (1.4)$$

$$\frac{S^0(T)}{R} = a_0 \ln(T) + a_1 T + \frac{1}{2} a_2 T^2 + \frac{1}{3} a_3 T^3 + \frac{1}{4} a_4 T^4 + a_6 \quad (1.5)$$

After computing all 14 coefficients, they are formatted in the CHEMKIN format and saved in the `./thermo` subdirectory, in the output file `nasa_polyn.out`.

The code automatically determines the level of theory to use for the estimation of $\Delta H^0(0\text{ K})$ from `./data/theory.dat` and reports it in the output files.

The output files of the CBH reactions are stored in the subdirectory `./thermo`; `DH0K.out` is the output file for the level 1 estimation, while `hl_DH0K.out` is the output file for high level estimation. The reactions reported are the same for both files. Each level computed, along with the current estimate of $\Delta H^0(0\text{ K})$, is reported, along with a list of the reference species used, with electronic plus zero-point energy and $\Delta H^0(0\text{ K})$ of the reference species.

If a rung can not be computed (it is usually the case for the rung CBH-2, because the PM is a small molecule or there is a missing reference species in the database), a warning message is reported in the output files, while lower rungs are still computed.

The implementation of the algorithm is reported in next subsections, with a step-by-step example of calculation for 2-hydroperoxybutyl (Figure 2.5).

The procedure for the database generation is reported in subsection (2.6.4) along with the methodology used for the reference species determination and a description of database structure.

The code is then tested on 142 species from the thermochemical database by Klippenstein et al. [60], which includes alkanes, alkyl radicals, alkyl-hydroperoxides, alkylperoxy radicals and hydroperoxy-alkyl radicals.

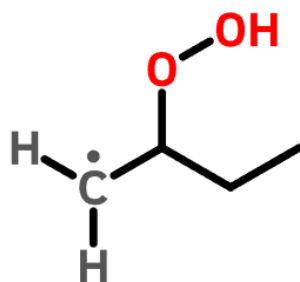


Figure 2.5: 2-hydroperoxybutyl structure

2.6.1 CBH-0, Isogyric scheme

The Isogyric fragmentation is an atomic-centred scheme; it divides the PM considering the nature of the heavy atoms present in the molecule. The products of CBH-0 (defined as rung 0 products) are determined by breaking every bond present in the initial molecule and saturating the connections with hydrogens; the information about each atom stability is preserved (i.e., presence of a radical atom is preserved). Since hydrogen atoms are added to each heavy atom forming the rung 0 products, a proper amount of hydrogen molecules will constitute, along with parent molecule, the rung 0 reactants.

The code, exploiting the InChI identifier of the molecule contained in *./data/name.dat* and RDKit functions, analyses every atom contained in the molecule; if the atom is heavy, the presence of unpaired electron is checked (i.e., if the atom is radical). After assessing the nature of the atom, the corresponding SMILES for the reference species that represents it is constructed using heuristic rules based on SMILES structure [65].

If the atom is not a radical, the SMILES corresponds to the atomic symbol; if the atom is a radical, the corresponding SMILES is formed by the atomic symbol of the atom and a number of hydrogens corresponding to the saturation number minus one, all between square brackets (i.e., the SMILES of methyl group CH_3^* corresponds to $[\text{CH}_3]$, hydroxyl radical group OH^* corresponds to $[\text{OH}]$ and amino radical group NH_2^* corresponds to $[\text{NH}_2]$).

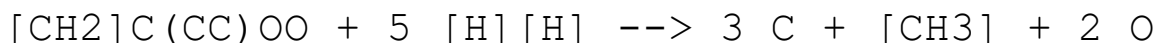
The amount of hydrogen molecules required to balance rung 0 reaction is calculated as difference between the amount of hydrogen atoms contained in products and hydrogen atoms contained in the parent molecule, exploiting the saturation number

of heavy atoms (it's considered that the saturation number corresponds to the most stable form of the heavy atoms).

$$N_{H_2} = \frac{1}{2} \left(\sum_{Stab\ prod} N_{i,S} \sigma_i + \sum_{Rad\ prod} N_{i,R} (\sigma_i - 1) - N_{H,PM} \right) \quad (2.4)$$

N_{H_2} is the number of hydrogen molecules in rung 0 reactants; $N_{i,S}$ and $N_{i,R}$ are the number of stable and radical products of type i ; σ_i is the saturation number of atom type i (equal for stable and radical species, not being a function of hydrogens bonded to the specific atom); $N_{H,PM}$ is the number of hydrogen atoms in the PM.

CBH-0 reaction for 2-hydroperoxybutyl, considering the presence of four carbons (one of which is a radical) and two oxygens, is reported below:



Having constructed the rung 0 reaction, the PM $\Delta H^0(0 K)$ can be estimated from Expression (2.2). Remembering $\Delta H_{H_2}^0(0 K) = 0$,

$$\Delta H_{PM}^0(0 K) = \sum_{prod} N_i \Delta H_i^0(0 K) - \left[\sum_{prod} (E_{El} + ZPE)_i \right] + N_{H_2} (E_{El} + ZPE)_{H_2} + (E_{El} + ZPE)_{PM} \quad (2.5)$$

The estimated $\Delta H^0(0 K)$ using $\omega B97X-D/jun-cc-pVTZ$ level of theory is -14.69 [kcal mol⁻¹] at level 1 of EStokTP, while the estimated $\Delta H^0(0 K)$ by [60] is 4.47 [kcal mol⁻¹]. The relative error is 428.6%, making the estimation at CBH-0 highly unreliable.

The Isogyric scheme construction is useful not only because it provides the starting reactants for rung 1 construction, but also because during this step the heavy atoms are analysed and counted; this information will be used for rung 1 reactants correction due to terminal moieties and branching.

The flowchart of CBH-0 algorithm is reported in Figure (2.6):

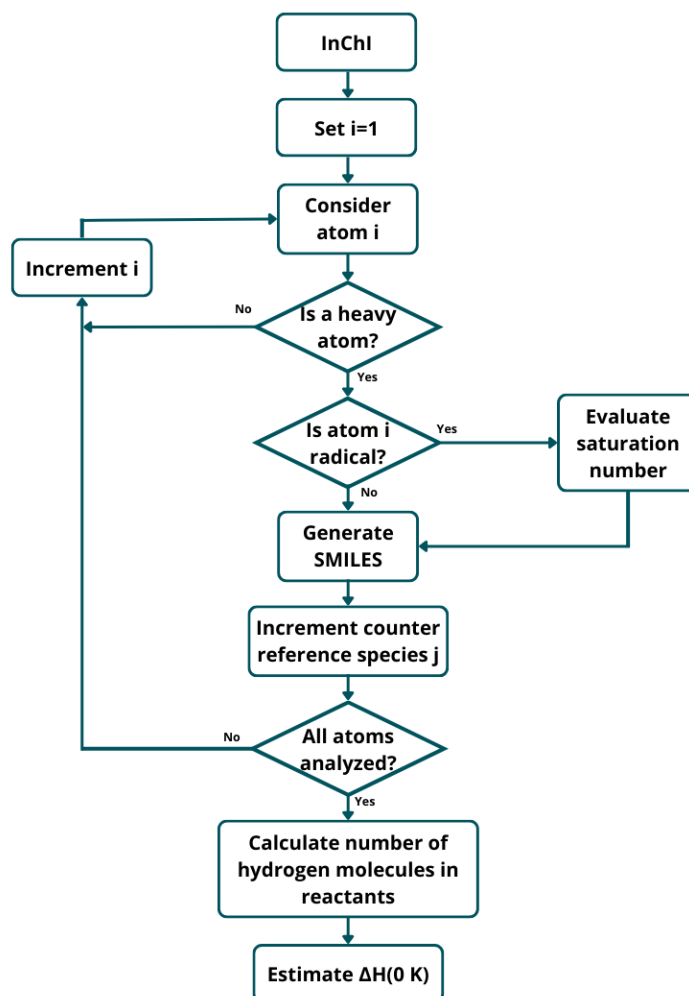


Figure 2.6: Flowchart of CBH-0 algorithm

2.6.2 CBH-1, Isodesmic scheme

The Isodesmic bond separation is a bond-centred scheme; it divides the PM preserving each bond environment in terms of the nature of the heavy atoms. The products of CBH-1 (defined as rung 1 reaction) are determined by isolating every bond formed by two heavy atoms present in the molecule and analysing the type of bond (single, double, or triple), and the type of heavy atoms present in the bond (which element and its saturation). Rung 1 reactants are determined starting from rung 0 products; they are corrected for the presence of terminal moieties and branched parts of the molecule.

Terminal moieties are groups which do not have two (or more) heavy atoms bonded; terminal moieties in 2-hydroperoxybutyl (Figure 2.5) are for example methyl-,

methyl radical and hydroxyl-groups. The molecules representing such moieties need to be treated differently with respect to others and get cancelled from rung 1 reactant.

Terminal moieties corrections are also applied to CBH-2 rung, because terminal atoms lead to reference species with only two heavy atoms, which are of the same type of CBH-1 products (i.e., CBH-2 reactants). This will be explained in section (2.6.3).

Branching is the formation of more than two bonds by a heavy atom; it takes place due to the affinity of carbon toward catenation. This leads to a double counting per branch. Branching corrections are not applied to atom-centred schemes (CBH-0 and CBH-2) because, while an atom can be at the intersection of multiple bonds, a covalent bond is formed exactly by two atoms, thus not requiring this type of correction.

The code, using nested loops, analyses the structure of the molecules (exploiting InChI identifier and RDKit functions) searching for all bonds formed by atom *i*. Nested loops are used for avoiding double counting of bonds. If atom *j*, bonded to atom *i*, is found, the reference species SMILES, representing the bond environment, is constructed. A SMILES structure comprising two heavy atoms is well defined; the general structure is {group atom *i*}{bond symbol}{group atom *j*}. For example, the reference species for the terminal bond circled in red of 2-hydroperoxybutyl (Figure 2.7) containing primary and secondary non-radical carbons is ethane, CC, while for the bond circled in green, containing tertiary non-radical carbon and non-radical oxygen, is methanol, CO.

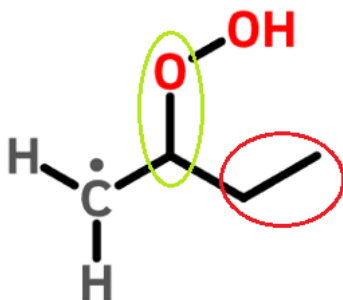


Figure 2.7: 2-hydroperoxybutyl CBH-1 example

Single bond symbol – is omitted in the SMILES structure, to keep the syntax compact; double and triple bonds symbols are represented by = and #, respectively. (e.g., ethylene and acetylene SMILES correspond to C=C and C#C).

Once the type of bond has been determined, the two atomic groups should be constructed; if the heavy atom considered is stable, the construction is trivial since the SMILES corresponds to the atomic symbol. The construction of groups involving

radicals is trickier: the number of hydrogens contained in the group SMILES is found using the following expression:

$$N_H = \sigma_{HA} - \sigma_B - 1 \quad (2.6)$$

N_H is the number of hydrogens in the group considered, σ_{HA} is the saturation number of the heavy atom and σ_B is the number of electrons shared to form the bond with the other heavy atom (i.e., not forming bonds with hydrogens). For example, the construction of the reference SMILES for the primary-radical carbon in Figure (2.7) would consider $\sigma_{HA} = 4$, $\sigma_B = 1$, and so $N_H = 2$, leading to SMILES [CH2].

For non-symmetric SMILES structures such as C[CH2] the code checks which version is present in the reference database (i.e., both C[CH2] and [CH2]C are searched in the database). This avoids the necessity of searching for canonical SMILES when constructing the reference database, since any valid form of equivalent SMILES structure is accepted by the code; care must be taken in the construction of the database since only one valid form per structure should be present. The presence of two valid SMILES forms of the same molecule leads to wrong counting of the reference species, with bad construction of the rung reactions.

The expression of rung 1 reaction after products identification, before terminal moieties and branching corrections, is reported below:



Without moieties and branching corrections, the balance between number and type of atoms/bonds is not correct.

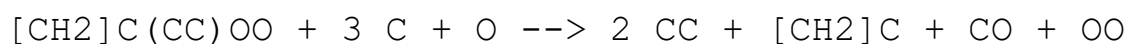
A simple correction procedure can be implemented by noticing that only the bold highlighted species in the expression above need to change stoichiometry. This is because the reference species for stable and radical heavy atoms is unique, no matter the surrounding environment of such atoms (e.g., a primary, secondary, and tertiary non-radical carbons, isolated from the surrounding environment, are always represented by a methane molecule C). Thus, counting the type of heavy atoms (radical or not) on both reactants and products sides gives a clear indication of the change in stoichiometry required to balance the rung 1 reaction. The reactants side counting has been implemented during the construction of CBH-0 level, while the products side counting is carried out during the construction of CBH-1 products. Expression (2.7) estimates the stoichiometric changes required for every single atom reference species:

$$\Delta N_i = \sum_{prod}^{CBH1} N_i - 2 \sum_{prod}^{CBH0} N_i \quad (2.7)$$

ΔN_i is the difference of heavy atoms of type i between products and reactants side (radical and non-radical species are considered different species). CBH-0 products are counted twice because, when reported as CBH-1 reactants, they are added to the PM counter, which contains an equivalent number of atoms of each type N_i , from which CBH-0 products were derived. Three possible situations arise:

- $\Delta N_i > 0$: excess of N_i -type atoms on products side; addition of ΔN_i to the corresponding species reactants side is needed.
- $\Delta N_i < 0$: excess of N_i -type atoms on reactants side; removal of ΔN_i to the corresponding species reactants side is needed.
- $\Delta N_i = 0$: N_i -type atoms are balanced, no need for stoichiometry changes.

The corrected expression for CBH-1 reaction of 2-hydroperoxybutyl is reported below:



After terminal moieties and branching correction, $\Delta H^0(0 \text{ K})$ can be estimated rearranging Expression (2.2) into Expression (2.8):

$$\Delta H_{PM}^0(0 \text{ K}) = \sum_{prod} N_i \Delta H_i^0(0 \text{ K}) - \left[\sum_{prod} (E_{El} + ZPE)_i \right] - \sum_{react} N_i \Delta H_i^0(0 \text{ K}) + \left[\sum_{react} (E_{El} + ZPE)_i \right] + (E_{El} + ZPE)_{PM} \quad (2.8)$$

Both summations over reactants do not include the parent molecule.

The estimated $\Delta H^0(0 \text{ K})$ at the $\omega\text{B97X-D/jun-cc-pVTZ}$ level of theory is 18.87 [kcal mol⁻¹]. The relative error is 322.1% compared to the estimate by Klippenstein et al. [60]; even if an improvement has been attained with respect to CBH-0 estimation, the result remains unreliable.

The products of CBH-1 rung are the starting point for the construction of CBH-2 level, which implements the hybridization-based homodesmotic scheme.

The CBH-1 algorithm also stores the number of bonds each atom forms: this will be used for the construction of the successive rung.

The flowchart of CBH-1 algorithm is reported below:

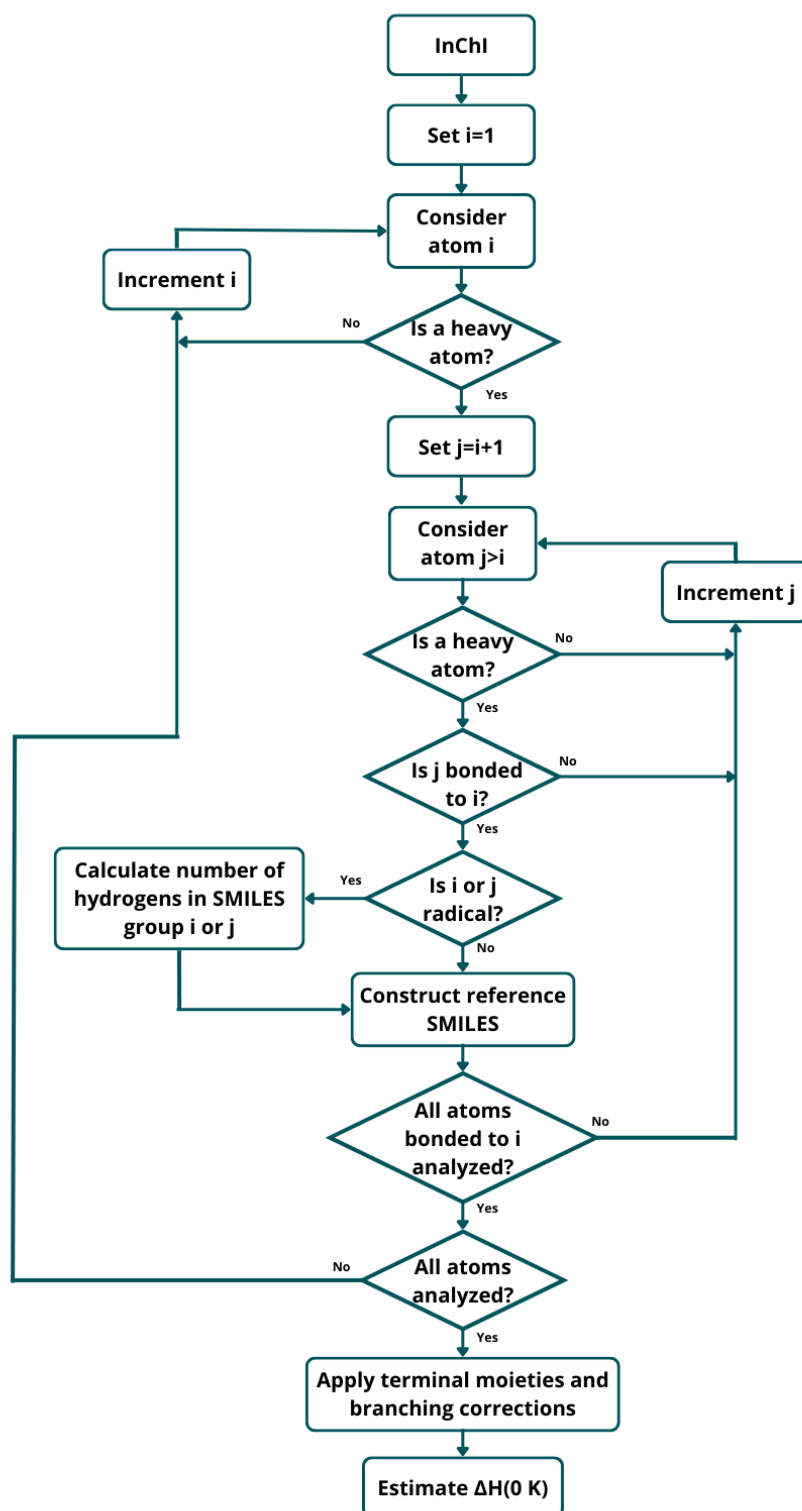


Figure 2.8: Flowchart of CBH-1 algorithm

2.6.3 CBH-2, Homodesmotic scheme

The hybridization-based homodesmotic separation is an atomic-centred scheme and it is the highest-level atomization procedure implemented in CHEMTP. It divides the PM preserving the immediate environment of each heavy atom: a reference species will be formed by the non-hydrogen atom considered and all the atoms to which it bond. The products of CBH-2 (defined as rung 2 reaction) are determined by isolating each non-hydrogen atom, considering the nature of each atom connected and the type of bond formed (single, double, or triple). The reactants of CBH-2 are determined starting from rung 1 products; as in CBH-1, corrections for terminal moieties are needed in order to balance the reaction.

Rung 2 terminal moieties are species formed by two heavy atoms. Taking 2-hydroperoxybutyl (Figure 2.5) as example, the surrounding environment of the primary radical carbon is represented by the species C[CH₂]. Since C[CH₂] is also present as a rung 2 reactant (i.e., product rung 1), it appears on both sides and thus it needs species cancellation.

Being the hybridization-based homodesmotic scheme an atomic-centred one, branching corrections are not needed, since the entire set of connections formed by an atom is taken into consideration, thus branched atoms are considered once automatically.

The first step is the determination of all non-hydrogen atoms bonded to each heavy atom of the molecule; as for CBH-0 and CBH-1 rungs, this is done using the molecule InChI identifier stored in *./data/name.dat* and RDKit functions. The logic of the algorithm is the same implemented for rung 1 construction (Figure 2.8), which means cycling through all atoms and identifying the ones bonded to the atom of interest.

If the number heavy atoms connected to the atom of interest is equal to one, it means that the atom of interest is a terminal moiety; the construction of SMILES structure follows the same procedure shown in Section (2.6.2). Once the correct arrangement has been determined, the corresponding species stoichiometric coefficient on reactants side is decreased by one (i.e., equivalent to species cancellation).

If the number of bonds is greater than one (i.e., three, four, or five, as for the database constructed in the present work), the SMILES construction is different. First, the nature of each bond is determined; then each group structure bonded to the atom of interest is established. For stable species the SMILES corresponds to the atomic symbol, while for radical species the number of hydrogens is determined by Expression (2.6). As last step, the target atom group is determined; if the atom of interest is stable, the SMILES arrangement is trivial, while if it's radical, the number of hydrogens is established by Expression (2.9):

$$N_H = \sigma_{HA} - \sum_{\text{Bonds}} \sigma_{Bi} - 1 \quad (2.9)$$

N_H is the number of atoms contained in the target atom SMILES structure, σ_{HA} is the saturation number of the atom of interest, σ_{Bi} is the number of electrons shared to form the bond i (i.e., not forming bonds with hydrogens). As example, the SMILES corresponding to isopropyl-radical (Figure 2.9a) and tert-butyl-radical (Figure 2.9b) are C[CH]C and CCC, respectively.



Figure 2.9: isopropyl-radical and tert-butyl-radical structure

The construction of the complete reference SMILES is done exploiting well-defined tri-, tetra- and pent-atomic SMILES structure.

GS = Group SMILES, B = Bond symbol; i is the atom of interest, while j, k, x, y are the atoms bonded to i .

Triatomic SMILES: $GS_j B_j GS_i B_k GS_k$

Tetratomic SMILES: $GS_j B_j GS_i (B_x GS_x) B_k GS_k$

Pentatomic SMILES: $GS_j B_j GS_i (B_x GS_x)(B_y GS_y) B_k GS_k$

For example, the reference SMILES for the blue-circled group in Figure 2.10 is CC(O)[CH2] while for yellow-circled group in Figure 2.10 is CCC.

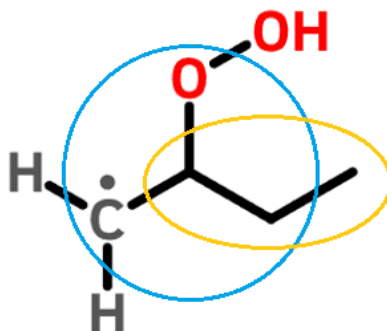
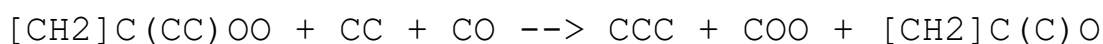


Figure 2.10: 2-hydroperoxybutyl CBH-2 example

The problem of multiple SMILES definitions of the same molecular structure (e.g., the blue-circled structure can be identified also by CC([CH2])O or [CH2]C(C)O) is solved as it was done in rung 1: a series of nested loops cycles every possible combination by switching group SMILES and bonds j,k,x,y ; the position of the atom of interest does not change, thus is left untouched by the combinatorial approach. A maximum of $4! = 24$ possibilities are checked, searching for the only structure present in the reference database; attention must be paid to have one structure only defined in the database.

Once every non-hydrogen atom environment has been checked and all SMILES structures have been identified, the final form of CBH-2 reaction is constructed. The rung 2 reaction for 2-hydroperoxybutyl is reported below:



As it can be noticed, reactants OO, [CH2]C and one CC have been cancelled because they represent terminal moieties.

$\Delta H^0(0\text{ K})$ is estimated using Expression (2.9).

The estimated $\Delta H^0(0\text{ K})$ using $\omega\text{B97X-D}/\text{jun-cc-pVTZ}$ level of theory is 4.66 [kcal mol⁻¹] at level 1 of EStokTP; comparison with the $\Delta H^0(0\text{ K})$ estimated by Klippenstein et al. [60] (4.47 [kcal mol⁻¹]) results in an absolute overestimation of 0.19 [kcal mol⁻¹] and a relative error of 4.1%.

As it can be seen, the $\Delta H^0(0\text{ K})$ estimated at CBH-2 is more precise than CBH-0 and CBH-1, with the error decreasing from 424.5% and 318% to 4.1%. This highlights the necessity of computing all rungs present in the algorithm (which also implies constructing a reference species database sufficiently large). Although some of the molecules tested can be computed only at CBH-1 because of the small number of atoms present, relying on CBH-1 estimations of molecules that can be computed at CBH-2 can lead to erroneous estimations, and it would always be preferable to expand the reference species database to include all the required molecules, whenever possible.

The estimation accuracy can be even incremented if higher-level electronic and zero-point energies are used; if present, the code will automatically estimate $\Delta H^0(0\text{ K})$ both at level 1 and at high-level defined in the EStokTP input files.

The reference species database contains both level 1 and high-level reference data. Its structure is described in Section (2.6.4).

The flowchart of CBH-2 algorithm is reported below:

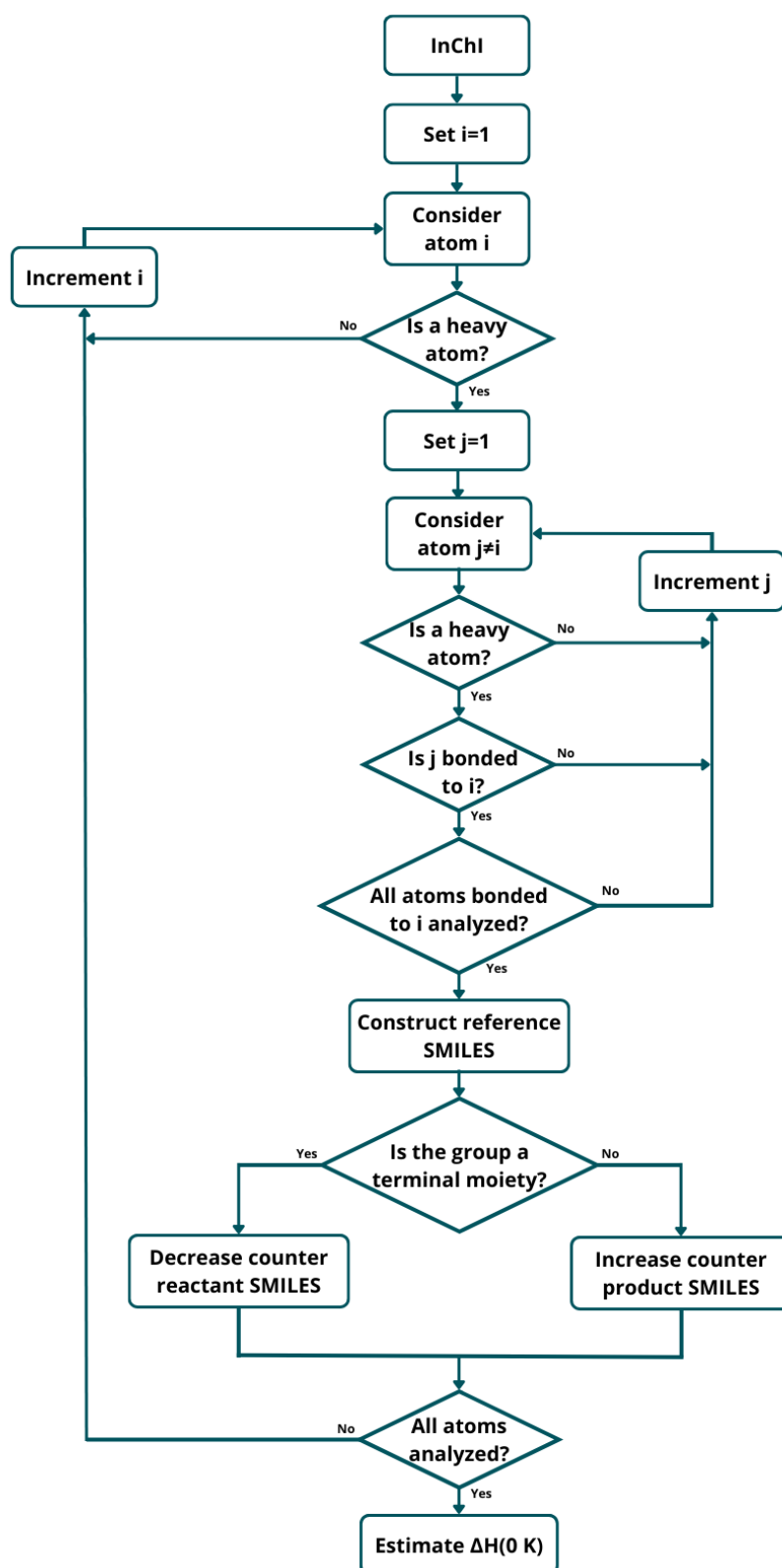


Figure 2.11: Flowchart of CBH-2 algorithm

2.6.4 Reference species database

The implementation of Connectivity Based Method requires the availability of information about electronic energy, zero-point energy and $\Delta H^0(0 K)$ of the reference species, to apply Expressions (2.5) and (2.8). In the present work the choice of constructing the reference database with C, O, N and H was done to cover the most relevant species involved in combustion and atmospheric kinetic mechanisms; the extension to species containing different heavy atoms is in any case possible.

The first step is the determination of all the species that will be included in the database using a combinatorial approach to generate all the feasible structures involving carbon, oxygen, nitrogen, and hydrogen. All the reference species used by CHEMTP for the construction of CBH reactions were selected and constructed by hand.

The second step is obtaining reference data (electronic and zero-point energy, and $\Delta H^0(0 K)$). For $\Delta H^0(0 K)$ Argonne National Laboratory - Active Thermochemical Tables (ANL-ATcT, v 1.124 (complete)) database was used [75]. For electronic and zero-point energies, in-house calculations were made using EStokTP. Two different levels of theory were used: ω B97X-D/jun-cc-pVTZ and B2PLYP-D3/jun-cc-pVTZ. These correspond to two possible level 1 levels of theory defined for EStokTP jobs. Moreover, starting from such level 1 geometries, energies were also estimated at CCSD(T) level of theory with extrapolation to basis set limit and correction for core electrons correlation. Therefore, in total four different levels of theory were defined and an equivalent number of database were built.

The four levels, ω B97X-D/jun-cc-pVTZ, B2PLYP-D3/jun-cc-pVTZ, and the CCSD(T) starting from both DFT geometries will be indicated as ω B97X-D, B2PLYP-D3, ω B97X-D-HL and B2PLYP-D3-HL.

Once all the reference data have been collected, they are organized in two different text files with their SMILES, one for level 1 estimations and one for high level estimations.

All the reference species input files, contained in each subdirectory **./data**, have been created using the code InChI2data developed in this work.

The list of reference species determined at level 1 and high level are reported in Appendix A.

2.6.5 Correction of $\Delta H^0(0 K)$

Partition functions can not be estimated at 0 Kelvin, thus $\Delta H^0(0 K)$ can not be directly used for estimating the coefficient a_5 from Expression (1.4); an estimation of

enthalpy at a higher temperature is then required. The simplest procedure is the application of the extrapolating scheme reported by Ochterski [39]. The approach is summarized by Expression (2.10):

$$\Delta H^0_{PM}(298.15\text{ K}) = \Delta H^0_{PM}(0\text{ K}) + H_{PM}^{corr} - \sum_{Atoms} N_i H_i^{corr} \quad (2.10)$$

$\Delta H^0_{PM}(298.15\text{ K})$ is the enthalpy of formation of the parent molecule evaluated at 298.15 K, $\Delta H^0_{PM}(0\text{ K})$ is the enthalpy of formation of the parent molecule estimated at 0 K (using the CBH method), H_{PM}^{corr} is the thermal correction to enthalpy for the parent molecule, N_i is the number of atoms of type i present in the parent molecule and H_i^{corr} is the thermal correction to enthalpy for atoms of type i .

The thermal correction to enthalpy of the parent molecule H_{PM}^{corr} is obtained from the log file of level 1 calculations `./geoms/reacl_1l.log`; care must be taken to subtract the zero-point energy from the value reported in the log file, as highlighted by Ochterski [39].

The thermal corrections to enthalpy of atomic species are obtained from experimental values by Pople et al. [74]; they are reported in (Table 2.1).

Table 2.1: Thermal correction to enthalpy of the atomic elements, from Pople et al. [74].

Element	H_i^{corr} [kcal mol ⁻¹]
H	1.01
Li	1.1
Be	0.46
B	0.29
C	0.25
N	1.04
O	1.04
F	1.05
Na	1.54
Mg	1.19
Al	1.08
Si	0.76
P	1.28
S	1.05
Cl	1.1

The output of the thermal correction to $\Delta H^0(0\text{ K})$ is stored in the directory `./thermo` as `DH298K.out`; it contains the estimated $\Delta H^0(298.15\text{ K})$, the level of estimation (level

1 or high level of EStokTP) and the number of atoms of type i with the associated thermal correction.

2.6.6 Estimation of NASA polynomials coefficients

The final step of CHEMTP code is the estimation of coefficients reported in Expressions (1.3), (1.4) and (1.5).

The first step is the estimation of coefficients a_0 to a_4 (Expression 1.3). This is done by two successive non-linear regressions using two different Python libraries (NumPy [76] and SciPy [77]); $C_p^0(T)$ data over a sufficiently large temperature interval are extracted from *cp.out* file, contained in the directory *./thermo*. Rather than choosing an arbitrary split temperature for high and low temperature ranges and trying to fit the $C_p^0(T)$ data over fixed intervals, the code selects the most suitable value to have the best possible fitting. As first guess value, the middle temperature of the entire interval found in *cp.out*, T_{split}^0 , is selected; three preliminary regressions are made at T_{split}^0 , $T_{split}^0 + 20$, $T_{split}^0 - 20$ using *numpy.polyfit()* function. The correspondent coefficients of determination R^2 are calculated: based on R^2 , the code chooses the “direction” in which the highest R^2 can be estimated (at higher or lower temperature with respect to the T_{split}^0). If both R^2 at higher and lower temperature are worse than R^2 estimated at the first guess temperature, the NASA polynomials are calculated using the temperature initially considered.

Once the “direction” is established, the code uses an iterative procedure schematized as follow:

- Construct high and low temperature intervals, with a partial superposition (the lowest 10 temperature data of the high temperature interval are included in the low temperature interval, while the highest 10 temperature data of the low temperature interval are included in the high temperature interval) of the spans at the split value to have better junction estimation (i.e., avoids different estimation at the split value using the high and low temperature coefficients).
- Use *numpy.polyfit()* from NumPy [76] to estimate a first guess set of high and low temperature coefficients: this function does not require the choosing of initial values by the user, so this avoids erroneous estimation due to bad preliminary data.
- Use *optimize.curve_fit()* from SciPy [77], employing the Trust Region Reflective algorithm by Coleman and Li [78], using as first guess values the ones obtained using *numpy.polyfit()*.
- Check for continuity at the split value with first and second derivative; if continuity is guaranteed the recursive method stops, otherwise the split

temperature is changed according to the “direction” in which R^2 grows as it was determined previously, and a new regression is done.

When coefficients a_0 to a_4 have been estimated in a satisfactory manner, a_5 and a_6 from rearrangement of Expressions (1.4) and (1.5), reported below, are calculated for both ranges; $H^0(2500\text{ K})$ is calculated using Expression (2.3). Both S^0 data are obtained from *S.out* stored in **thermo** directory.

$$a_5 = \frac{H^0(T)}{R} - a_0 T - \frac{1}{2} a_1 T^2 - \frac{1}{3} a_2 T^3 - \frac{1}{4} a_3 T^4 - \frac{1}{5} a_4 T^5 \quad (2.11)$$

$$a_6 = \frac{S^0(T)}{R} - a_0 \ln(T) - a_1 T - \frac{1}{2} a_2 T^2 - \frac{1}{3} a_3 T^3 - \frac{1}{4} a_4 T^4 \quad (2.12)$$

At this point the entire set of coefficients has been determined; the code formats the set in the CHEMKIN format and saves it in an output file *nasa_poly.n.out*, stored in the directory **./thermo**.

Having estimated the coefficients of NASA polynomials, the user can have access to thermochemical information such as $C_p^0(T)$, $H^0(T)$ and $S^0(T)$ at every temperature inside the range reported in the first line of the CHEMKIN format of NASA polynomials. Another quantity that can be directly estimated from *nasa_poly.n.out* is the standard Gibbs free energy, being $G^0(T) = H^0(T) - TS^0(T)$.

The flowchart of the regression algorithm implemented in CHEMTP is reported below:

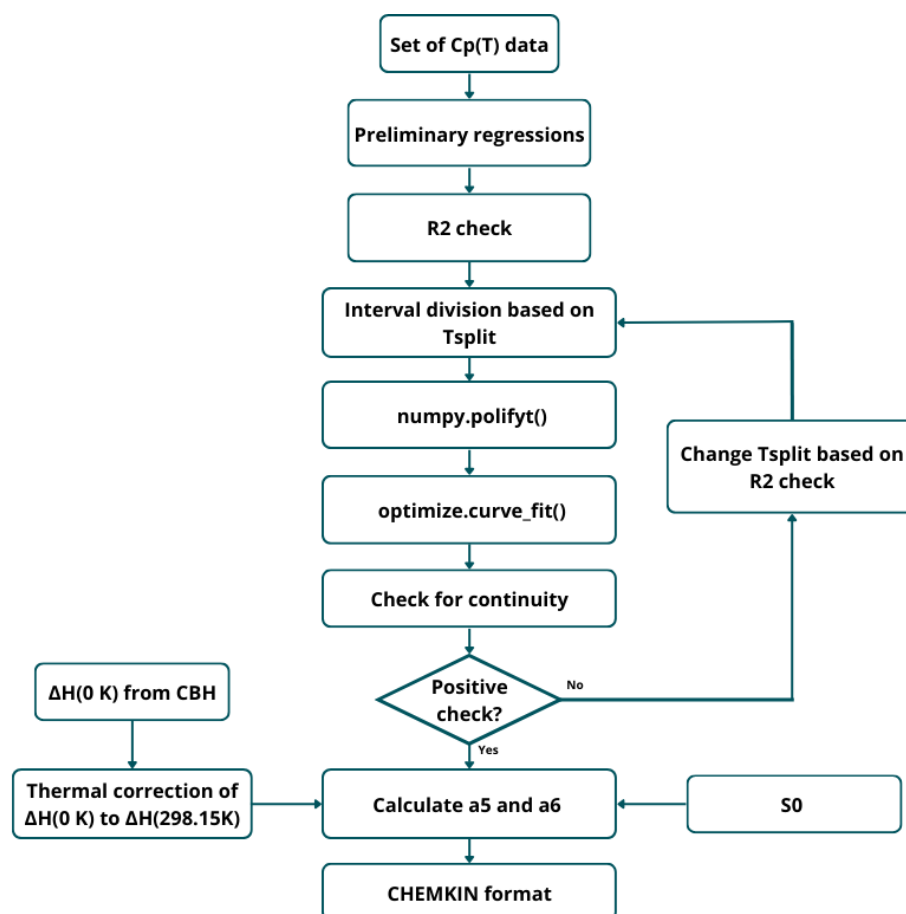


Figure 2.13: Flowchart of regression algorithm

2.6.7 CHEMTP user manual

The present chapter introduces the utilization procedure of the code CHEMTP by the user.

- Move to the working directory of a completed EStokTP job (i.e., containing the subdirectories **data**, **geoms**, **output** etc. and the empty *finished* file).
- Check if *name.dat* text file, with the InChI identifier of the molecule in the first line, is contained in the subdirectory **data**. If not, it must be created.
- Run the CBH part of the CHEMTP code; it will generate the subdirectory **./thermo** (if not already present). Inside the subdirectory **./thermo** the output files *DH0K.out* and *hl_DH0K.out* will contain the CBH estimation results at level 1 and high-level of EStokTP, respectively.
- Run the code *Hcorrection_0_to_298K.py* for correction from 0 to 298.15 K of ΔH^0 . It will create the output file *DH298K.out* inside the **thermo** subdirectory.

- Run the code *nasa_estimator.py* for the estimation of the coefficients of NASA polynomials; it will create an output file *nasa_polyn.out* containing the coefficient of the NASA polynomials in the CHEMKIN format.

2.7 Summary of the codes

In this section a panoramic of the codes developed in the present work and their main functionalities are presented.

InChI2data: this code generates automatically the input subdirectory **./data** for a single species job in EStokTP, starting just from the InChI identifier of the chemical species of interest. InChI2data can generate a single subdirectory **./data** or multiple subdirectories **./dir_num/data**, with **dir_num** equal to 0001, 0002 and so on, depending on the number of InChI identifiers passed to the code.

FragmentsGen: this code generates all the fragmentation products of an initial molecule and calculates the correspondent bond energies. This allows the prediction of which bond is more likely to break and so which reaction channel may be more important in the Potential Energy Surface.

CHEMTP (Connectivity Hierarchy Estimation Model for Thermochemical Parameters): this code estimates the NASA polynomials in the CHEMKIN format of the molecule of interest. It exploits RRHO approximation and explicit treatment of 1D hindered rotors for the computation of translational, vibrational, and rotational contributions to calculate $C_p^0(T)$ and $S^0(T)$ over a defined range of temperatures. $\Delta H^0(0 K)$ is determined using the Connectivity Based Hierarchy method and corrected to obtain $\Delta H^0(298.15 K)$. After the correction of $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$, a series of regressions estimate the NASA coefficients, which are formatted in the CHEMKIN format and saved in the output file *nasa_polyn.out*.

3. Results and discussion

Chapter 3 initially reports the results obtained when the FragsGen code is used for a case study estimation of bond energies for a selected set of chemical species; the molecule considered is 1,3-butadiene-2-ol. Considerations on estimated bond energies with respect to experimental values in different molecules are also reported, validating the protocol implemented in the FragsGen code for the preliminary exploration of possible fragmentation pathways of any chemical species, a preliminary step for the construction of Potential Energy Surfaces for unimolecular decomposition reactions.

Afterwards, the results of the estimation of $\Delta H^0(0\text{ K})$ of 142 species from the species database by Klippenstein et al. [60] are discussed. They include 16 alkanes, 43 alkyls, 26 alkylhydroperoxides, 27 alkylperoxides and 30 hydroperoxy-alkyl. An error analysis is performed to evaluate the accuracy of CHEMTP, in terms of standard deviation 2σ .

A sub-set of 8 medium-size molecules, with molecular weight comprised between 64 and 75 amu, are then selected to evaluate the difference between different levels of theory estimation, at the level 1 and the high levels of theory as defined and implemented in EStokTP. The same sub-set is used to estimate the accuracy of the correction of $\Delta H^0(0\text{ K})$ for the calculation of $\Delta H^0(298.15\text{ K})$, in terms of standard deviation 2σ .

Finally, a comparison of the estimated NASA polynomials by CHEMTP and RMG [56] is carried out.

3.1 Fragmentation of 1,3-butadiene-2-ol

The fragmentation of 1,3-butadiene-2-ol (Figure 3.1), using the code FragsGen, is performed. The estimation of bond energies is reported both at level 1 and high-level calculations of EStokTP, which in this case study were $\omega\text{B97X-D}/\text{jun-cc-pVTZ}$ and CCSD(T) level of theory with extrapolation to basis set limit and correction for core electrons correlation, respectively.

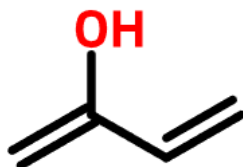


Figure 3.1: 1,3-butadiene-2-ol structure

The selection of 1,3-Butadiene-2-ol was made because it contains the main substructures of interests (e.g., single and double carbon-carbon bonds, and hydroxy groups) for combustion kinetics and atmospheric kinetics, which are among the main targets of EStokTP calculations.

The bond breakage of this molecule leads to eight different fragmentation pathways, with multiplicity equal to one for every breakage (i.e., no equivalent pairs of fragments are produced).

The fragmentations generated are reported in Figure (3.2); the bond energies are reported in Table (3.1).

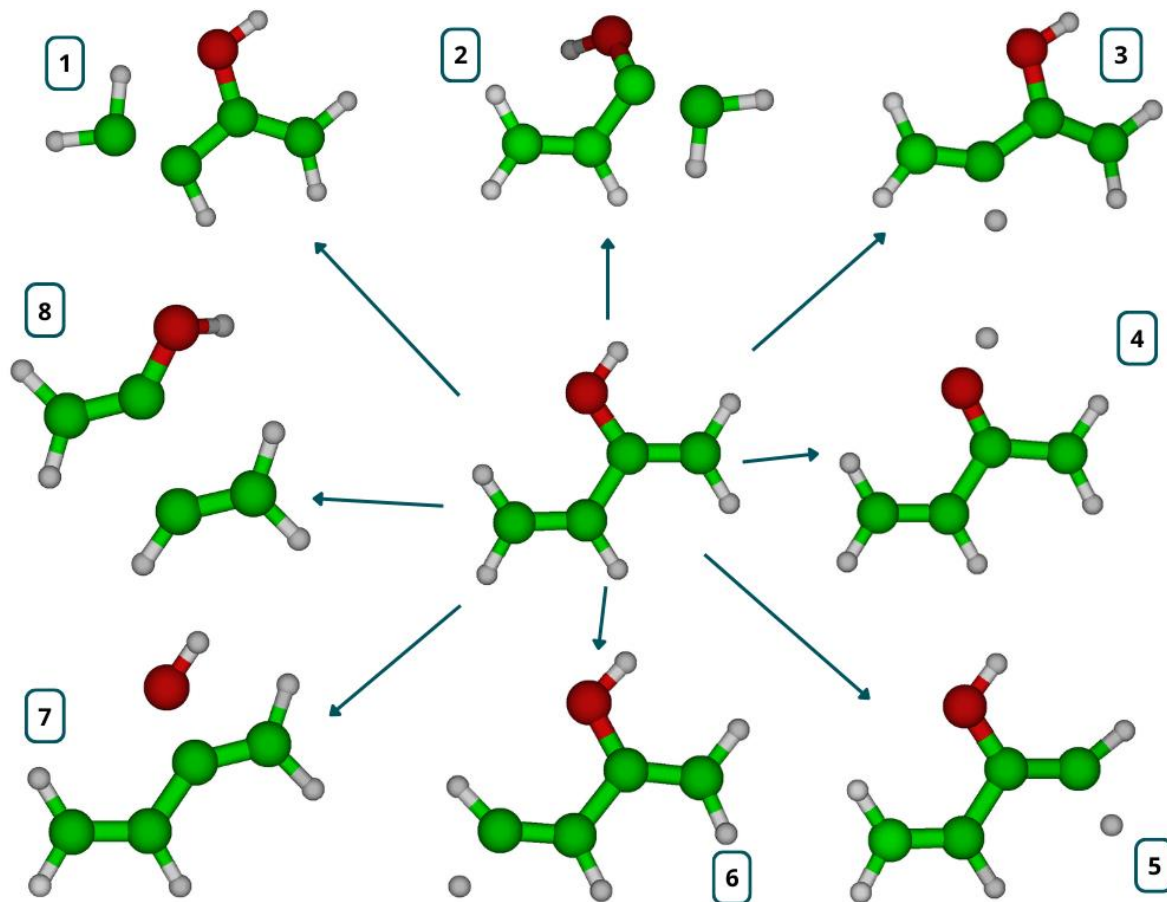


Figure 3.2: 1,3-butadiene-2-ol fragmentation products

Table 3.1: 1,3-butadiene-2-ol bond energies at the ω B97X-D/jun-cc-pVTZ level of theory (Level 1) and at the CCSD(T) level of theory with extrapolation to basis set limit and correction for core electrons correlation (High level)

N. Fragments	Level 1 [kcal mol⁻¹]	High level [kcal mol⁻¹]
1	144.28	144.77
2	155.64	154.91
3	107.43	109.86
4	81.54	85.40
5	112.43	113.94
6	109.44	111.32
7	105.29	107.84
8	110.17	113.98

The fragments generated consider every possible bond breakage, ranging from H-abstraction as in case of fragmentations 3,4,5 and 6, chain ruptures as in 1,2, and 8, and hydroxy abstraction in case 7.

The bond with the lowest energy is the one which leads to fragmentation number 4, the H-abstraction from the oxygen atom; the estimated values are 81.54 [kcal mol⁻¹] at level 1 and 85.40 [kcal mol⁻¹] at high level of EStokTP. So et al. [79] estimate the abstraction of hydrogen from 1,3-butadiene-2-ol oxygen as the H-abstraction with the lowest energy, with a bond energy of 84.50 [kcal mol⁻¹]. The estimation protocol is similar to the one implemented in EStokTP: the structure optimization is carried out using M06-2X/6-31G(2df,p) level of theory; from the obtained structures, G3X-K and M06-2X/aug-cc-pVTZ level of theory are used for higher level calculations. The final energy values are estimated incorporating also CCSD(T) level of theory with extrapolation to basis set limit.

Based on the results of So et. al, it can thus be concluded that FragsGen code can identify the weakest bond of 1,3-butadiene-2-ol correctly.

The strongest bonds in 1,3-butadiene-2-ol are the double carbon-carbon bonds; their rupture results in fragmentations 1 and 2, with an estimated energy of 144.77 [kcal mol⁻¹] and 154.91 [kcal mol⁻¹].

Blanksby and Ellison [80] estimate an experimental C=C bond energy of 174.1 [kcal mol⁻¹]; this experimental value is referred to ethylene: the surrounding atomic neighbourhood present in 1,3-butadiene-2-ol may cause discrepancy from the value estimated by FragsGen (using EStokTP calculations) and the value estimated by Blanksby and Ellison. Also, the estimated experimental value of the C-H bond breakage in ethylene reported [80] is 110.7 [kcal mol⁻¹], which is close to the estimated

values using FragsGen of 109.86, 113.94 and 111.32 [kcal mol⁻¹] for fragmentations 3,5 and 6, respectively. The lower value of bond energy calculated for fragmentation 3 can be interpreted considering the higher stability of the secondary radical carbon with respect to primary radical carbon that formed through fragmentations 5 and 6.

The results obtained in the computation of the bond energies of 1,3-butadiene-2-ol confirm the possibility of using the code FragsGen for the preliminary exploration of the most important PES reaction channels concerning the rupture of a single bond, such as H-abstractions.

The code can also be used for the estimation of bond energies in different molecular vicinities. Since the same bond does not have the exact same energy if present in different molecules or with different surrounding atoms, like the C=C bonds in 1,3-butadiene-2-ol, the experimental procedures for the estimation of all possible cases are usually long and complicated. The use of the code FragsGen allows the determination of bond energies in different atomic neighbourhood, expanding the range of possible bond energy estimation.

3.2 Thermochemical parameters

3.2.1 Estimation of $\Delta H^0(0 K)$ at the ω B97X-D/jun-cc-pVTZ level

Klippenstein et al. [60] estimated the $\Delta H^0(0 K)$ of the 142 test species considered in this study at the CCSD(T)-F12b/cc-pVTZ-F12//B2PLYP-D3/cc-pVTZ level of theory, which considers extrapolation to basis set limit and correction for core electrons correlation. Corrections for ZPE anharmonicity were also introduced. This level of accuracy is similar or slightly better with respect to the high-level estimation at the ω B97X-D-HL or B2PLYP-D3-HL levels, introduced in subsection (2.6.4), of CHEMTP, although anharmonicities for ZPE were not considered. This type of calculations requires long computational time, which was not available for the test of the code. So, in the present work the $\Delta H^0(0 K)$ of the 142 species were estimated at the ω B97X-D/jun-cc-pVTZ level of theory, without extrapolation to basis set limit or correction for core electrons correlation and without corrections for ZPE anharmonicity.

Although the lower level of theory used for the test estimation of the $\Delta H^0(0 K)$, the absolute error and relative percentage error of most of the species taken into consideration is similar to the one obtained by Klippenstein et al, but it has been obtained at a much lower computational cost. Moreover, Klippenstein et al. used a limited number of sample points in the Monte Carlo sampling over internal coordinates; the number of sample points was selected as the lower value between 100 and $6 + 2 \times 3^N$, where N is the number of non-methyl torsional angles. The

calculations performed with CHEMTP used Monte Carlo sampling results with no limitation in terms of sampling points using an in-house built Python code named ParConf, which permitted to split and parallelize the Monte Carlo sampling for the search of the minimum energy configuration; the number of sampling points used was up to 500. The difference in terms of number of sampling points could lead to different geometry minimization and different estimation in electronic energy and ZPE.

Figure (3.3) to (3.7) report the absolute value of the relative percentage error of the estimated $\Delta H^0(0 K)$ of alkanes, alkyls, alkylhydroperoxides, alkylperoxides and hydroperoxy-alkyls using the code CHEMTP and the $\Delta H^0(0 K)$ estimated by Klippenstein et al. [60].

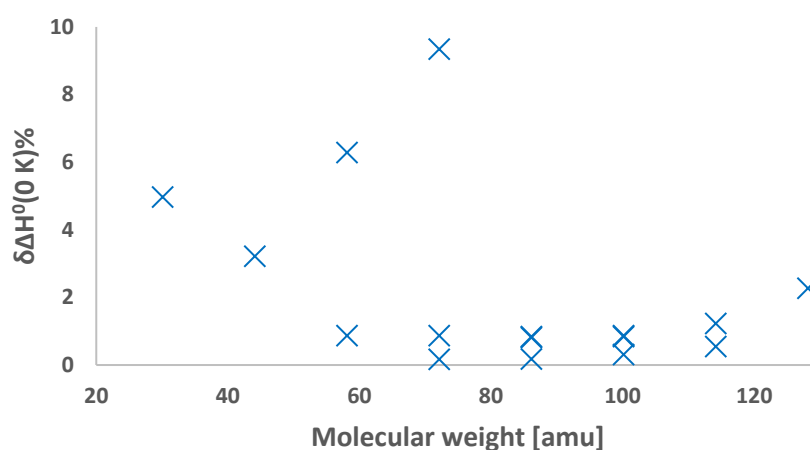


Figure 3.3: Relative percentage error of the estimated $\Delta H^0(0 K)$ of alkanes using the code CHEMTP and the $\Delta H^0(0 K)$ estimated by Klippenstein et al. [60]

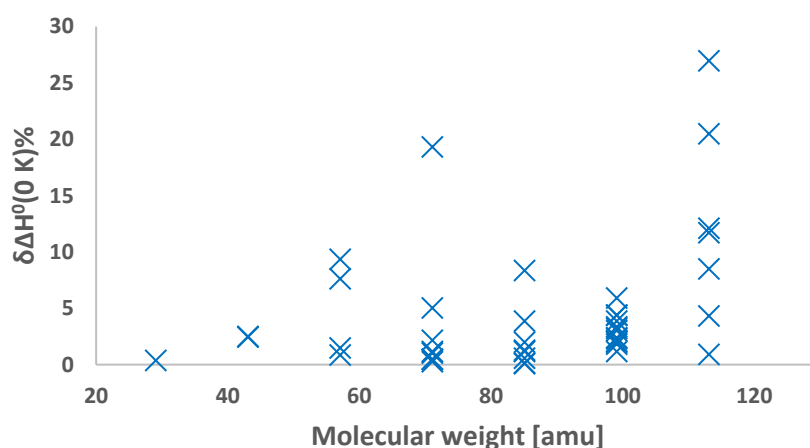


Figure 3.4: Relative percentage error of the estimated $\Delta H^0(0 K)$ of alkyls using the code CHEMTP and the $\Delta H^0(0 K)$ estimated by Klippenstein et al. [60]

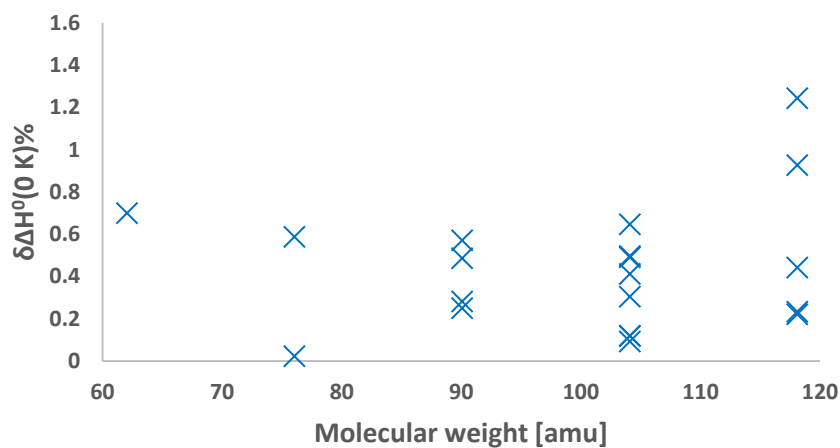


Figure 3.5: Relative percentage error of the estimated $\Delta H^0(0 K)$ of alkylhydroperoxides using the code CHEMTP and the $\Delta H^0(0 K)$ estimated by Klippenstein et al. [60]

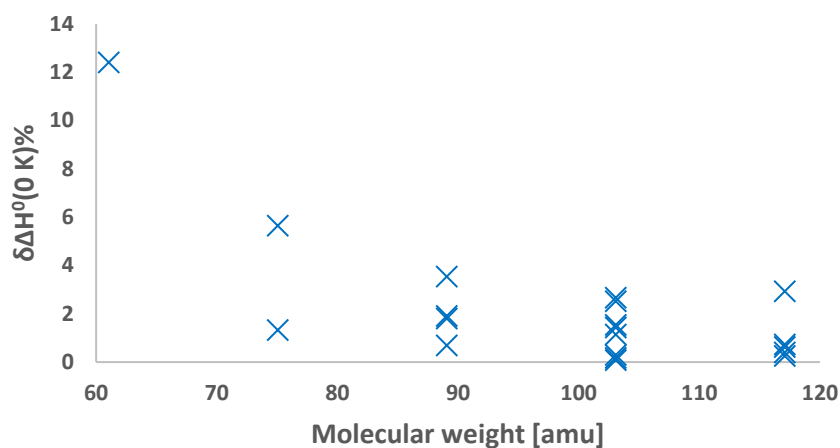


Figure 3.6: Relative percentage error of the estimated $\Delta H^0(0 K)$ of alkylperoxides using the code CHEMTP and the $\Delta H^0(0 K)$ estimated by Klippenstein et al. [60]

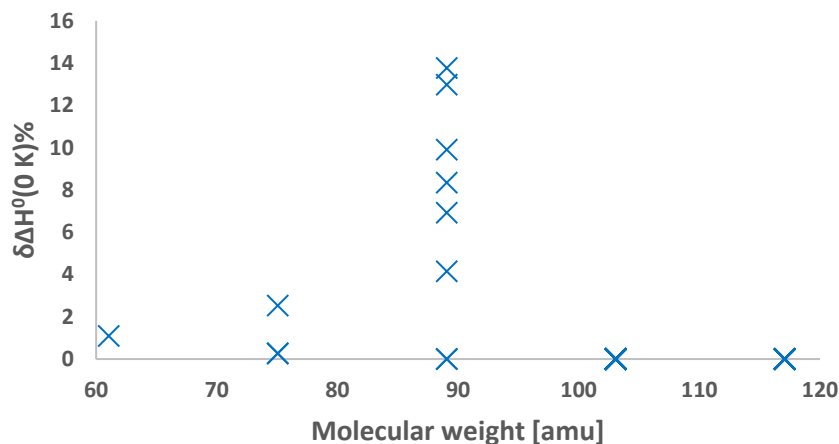


Figure 3.7: Relative percentage error of the estimated $\Delta H^0(0 K)$ of hydroperoxy-alkyls using the code CHEMTP and the $\Delta H^0(0 K)$ estimated by Klippenstein et al. [60]

For each group of species, the standard deviation of the absolute error and the standard deviation relative percentage error are calculated. They are summarized in Table (3.2).

The standard deviation for both absolute error and relative percentage error is calculated using Expression (3.1):

$$2\sigma = \sqrt{\frac{\sum(Y_i - \bar{Y})^2}{N}} \quad (3.1)$$

Where Y_i is the value of the absolute error or the relative percentage error of species i , \bar{Y} is the mean value of the absolute error or the relative percentage error, N is the total number of values of absolute error or relative percentage error.

Table 3.2: Absolute and relative standard deviation of $\Delta H^0(0 K)$ estimation

Species	2σ - Absolute error	2σ - Relative % error
alkanes	0.73	2.53
alkyls	0.59	5.75
alkylhydroperoxides	0.18	0.34
alkylperoxides	1.43	6.37
hydroperoxy-alkyl	0.39	4.19

Based on the 142 species from Klippenstein et al. [60] considered in this study, CHEMTP was able to predict the $\Delta H^0(0 K)$ with a 2σ of 0.54 for the absolute error and 5.09 for the relative percentage error. The estimated values of $\Delta H^0(0 K)$ by CHEMTP are in the uncertainty region of 0.39 ± 0.54 [kcal mol⁻¹] in terms of absolute error and 3.39 ± 5.09 % in terms of relative percentage error, with ω B97X-D/jun-cc-pVTZ level of theory.

The differences in the $\Delta H^0(0 K)$ estimation, using different levels of theory, are discussed in Section (3.2.2).

The estimated $\Delta H^0(0 K)$ with the absolute and relative percentage error of the entire set of 142 analysed in this work is reported in Appendix B.

3.2.2 Estimation of $\Delta H^0(0 K)$: influence of the level of theory

The influence of the level of theory on the estimation of $\Delta H^0(0 K)$ was analysed using a set of 8 molecules. Their SMILES, molecular weight (expressed in [amu]) and the estimated values of electronic energy plus ZPE, at every level of theory introduced in Section (2.6.4), are reported in Table (3.3).

Table 3.3: Sum of electronic energy and ZPE at different levels of theory

N	SMILES	MW [amu]	Eel+ZPE [Ha]			
			ω B97X-D	B2PLYP-D3	ω B97X-D-HL	B2PLYP-D3-HL
1	C1=C[C]C=C1	65.09	-193.459	-193.446	-193.358	-193.446
2	C1=CCCC1	68.12	-195.331	-195.215	-195.315	-195.315
3	C=CC(=C)C	68.12	-195.311	-195.200	-195.296	-195.297
4	C=CC(=C)O	70.09	-231.225	-231.120	-231.212	-231.212
5	C=CC(C)=O	70.09	-231.243	-231.136	-231.227	-231.227
6	C=COCC	72.11	-232.445	-232.328	-232.425	-232.425
7	CCC(C)=O	72.11	-232.483	-232.364	-232.461	-232.461
8	CCOCC	74.12	-233.677	-233.550	-233.653	-233.653

The $\Delta H^0(0 K)$ were estimated at every level of theory by CHEMTP and confronted with the values reported in the Active Thermochemical Tables of Argonne National Laboratory [75]. The absolute difference between the value estimated by CHEMTP and the value reported in ATcT [75], at every level of theory, is summarized in Table (3.4). A graphical comparison is reported in Figure (3.9).

Table 3.4: Absolute error of the $\Delta H^0(0 K)$ estimation at different levels of theory

N	SMILES	$\delta\Delta H^0(0 K)$ [kcal mol ⁻¹]			
		ω B97X-D	B2PLYP-D3	ω B97X-D-HL	B2PLYP-D3-HL
1	C1=C[C]C=C1	1.45	0.40	-0.44	-0.08
2	C1=CCCC1	0.35	0.14	-0.30	-0.30

3	C=CC(=C)C	0.59	-0.33	0.25	0.18
4	C=CC(=C)O	0.58	-0.37	0.13	0.07
5	C=CC(C)=O	0.46	-0.17	0.19	0.16
6	C=COCC	0.23	-0.10	-0.05	-0.05
7	CCC(C)=O	0.34	0.28	0.21	0.21
8	CCOCC	0.05	-0.15	-0.20	-0.21

From Table (3.4) it is clear that the extrapolation to the basis set limit and correction for core electrons correlation employed by EStokTP for high level calculations have a great impact on the estimation performed by CHEMTTP. For species 1, the precision was increased even by two orders of magnitude, passing from an absolute error of 1.45 [kcal mol⁻¹] using ω B97X-D/jun-cc-pVTZ level of theory to an absolute error of -0.08 [kcal mol⁻¹] using CCSD(T) with extrapolation to basis set limit and correction for core electrons correlation, starting from B2PLYP-D3/jun-cc-pVTZ geometries.

The increase in the level of theory is not a guarantee that the estimation will improve. Species 8 with an absolute error of 0.05 [kcal mol⁻¹] at ω B97X-D/jun-cc-pVTZ level of theory and an absolute error of 0.21 [kcal mol⁻¹] for both ω B97X-D-HL and B2PLYP-D3-HL is an example of such case where probably error cancellation effects allow to obtain an estimate closer to the one by Klippenstein et al. [60] at ω B97X-D/jun-cc-pVTZ level of theory. It is also important to remember that CHEMTTP results are compared with other theoretical estimates, therefore there is no guarantee that they are less in agreement with the real value of the enthalpy of formation (for which unfortunately experimental results do not exist). The case of species 8 is probably an example where the advanced corrections introduced by Klippenstein et al. are more relevant as the level of theory increases. Even if the estimation of $\Delta H^0(0\text{ K})$ can become less precise by increasing the level of theory used, the 87 % of the species considered in the set of 8 molecules showed an increased accuracy with high-level estimation with respect to ω B97X-D/jun-cc-pVTZ and B2PLYP-D3/jun-cc-pVTZ levels of theory.

Figure (3.9) illustrates the difference between the absolute error, at different levels of theory, of the species reported in Table (3.3).

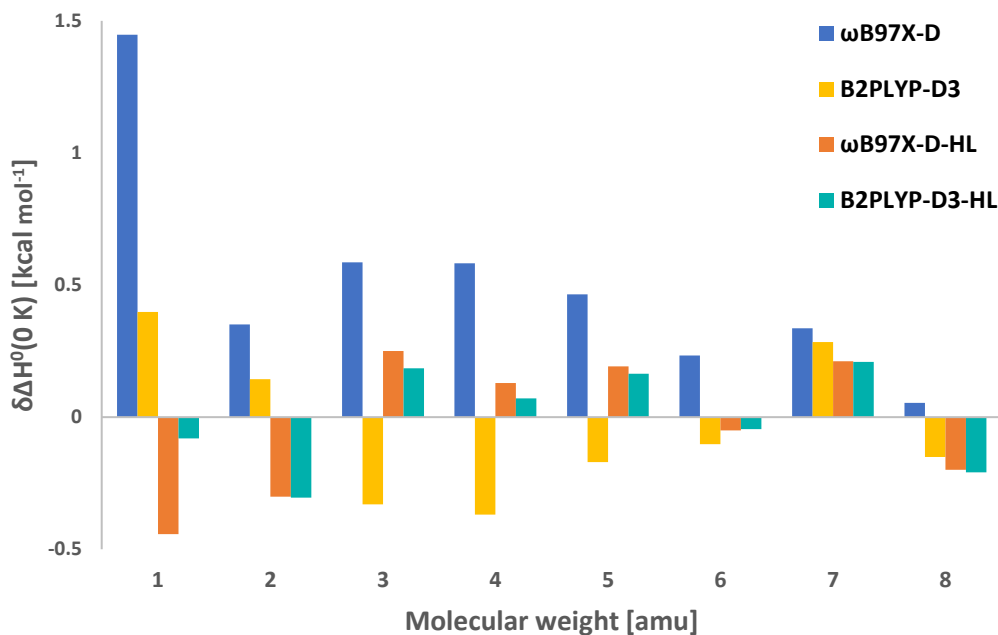


Figure 3.9: Absolute error of $\Delta H^0(0 K)$ estimation at different levels of theory

The standard deviation of the absolute error was calculated for every level of theory using Expression (3.1). They are reported in Table (3.5).

Table 3.5: Standard deviation of absolute error of $\Delta H^0(0 K)$ estimation at different levels of theory

ω B97X-D	B2PLYP-D3	ω B97X-D-HL	B2PLYP-D3-HL
0.39	0.26	0.24	0.17

An improvement in the estimation of $\Delta H^0(0 K)$, increasing the level of theory used, was confirmed by the standard deviation of the absolute error. ω B97X-D/jun-cc-pVTZ-HL calculations led to a standard deviation of 0.24, with respect to ω B97X-D/jun-cc-pVTZ 2σ of 0.39; B2PLYP-D3/jun-cc-pVTZ-HL standard deviation of 0.17 is also improved with respect to B2PLYP-D3/jun-cc-pVTZ 2σ of 0.26.

The results reported in Figure (3.9) highlighted the necessity of doing the calculations at the highest level possible, to have the most reliable estimation of $\Delta H^0(0 K)$. Although estimations using CCSD(T) with extrapolation to basis set limit and correction for core electrons correlation (high-level calculations) were more accurate, the computational time required for this type of calculations is greatly increased with respect to ω B97X-D/jun-cc-pVTZ and B2PLYP-D3/jun-cc-pVTZ calculations. If the estimation required is not very high, ω B97X-D/jun-cc-pVTZ and B2PLYP-D3/jun-cc-pVTZ can be used anyways for first guess estimations and refined in a second time.

3.2.3 Correction of $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$

The $\Delta H^0(298.15 K)$ of the set of 8 molecules reported in Table (3.3) are estimated by correction of the $\Delta H^0(0 K)$ estimated using high-level calculations, as described in Section (2.6.5).

The estimated $\Delta H^0(298.15 K)$ are compared to the $\Delta H^0(298.15 K)$ from Active Thermochemical Tables from Argonne National Laboratory [75]; the absolute error is calculated as difference between the ATcT value, and the value estimated by CHEMTP, in [kcal mol⁻¹]. The results are reported in Table (3.6).

Table 3.6: Absolute error of thermal correction from $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$

N	SMILES	ATcT [kcal mol ⁻¹]	CHEMTP [kcal mol ⁻¹]	$\delta\Delta H^0(298.15 K)$ [kcal mol ⁻¹]
		$\Delta H^0(298.15 K)$	$\Delta H^0(298.15 K)$	
1	C1=C[C]C=C1	62.62	62.60	0.02
2	C1=CCCC1	8.60	8.09	0.51
3	C=CC(=C)C	18.08	18.10	-0.02
4	C=CC(=C)O	-18.39	-16.77	-1.62
5	C=CC(C)=O	-27.47	-26.51	-0.96
6	C=COCC	-33.49	-33.98	0.49
7	CCC(C)=O	-57.00	-56.96	-0.04
8	CCOCC	-60.25	-60.92	0.67

The standard deviation of the absolute error for the thermal correction of $\Delta H^0(0 K)$ estimated with the data in Table (3.5) is 0.74.

The estimated error for the thermal correction of $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$ is thus -0.12 ± 0.74 .

The starting $\Delta H^0(0 K)$ are estimated at B2PLYP-D3/jun-cc-pVTZ-HL level of theory. 7 out of 8 species have an absolute error below chemical accuracy [kcal mol⁻¹] and only species 4 enthalpy differs by 1.5 [kcal mol⁻¹] from the reference ATcT value [75].

The results of the thermal correction of $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$, expressed as absolute difference between CHEMTP estimation and reference value reported in ATcT [75], are reported in Figure (3.10).

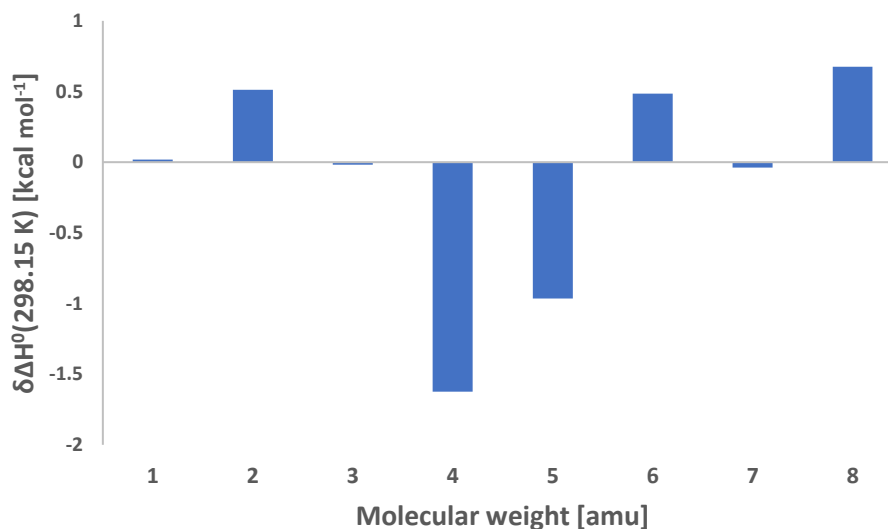


Figure 3.10: Absolute error of thermal correction of $\Delta H^0(0\text{ K})$ to $\Delta H^0(298.15\text{ K})$

3.2.4 NASA polynomials comparison

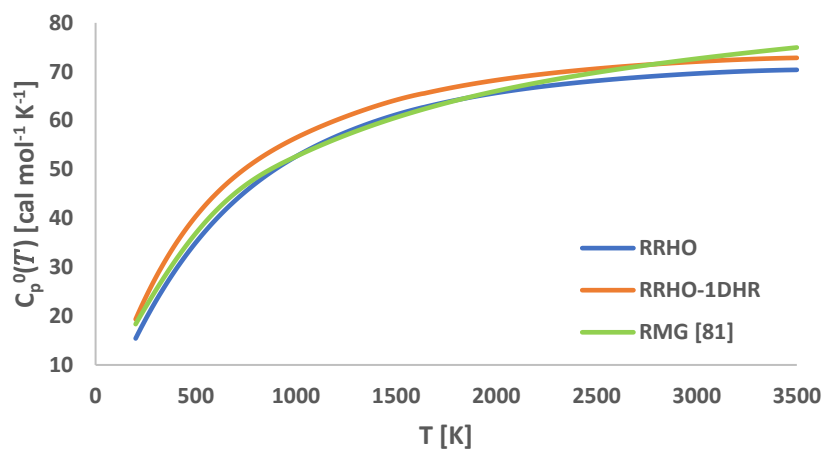
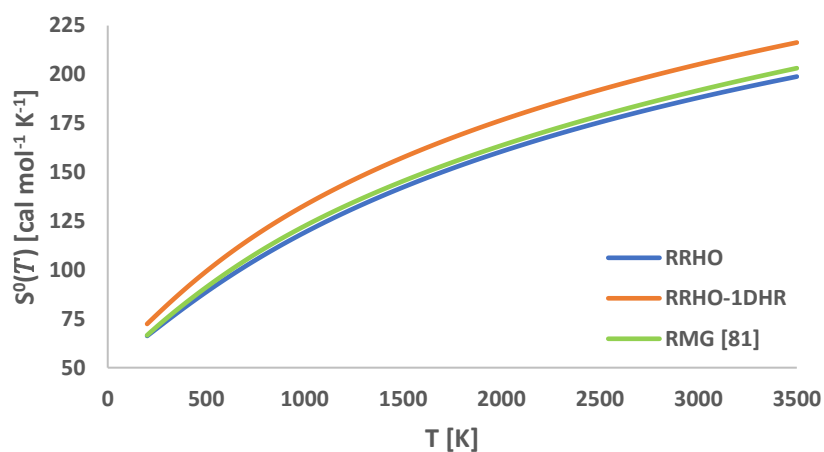
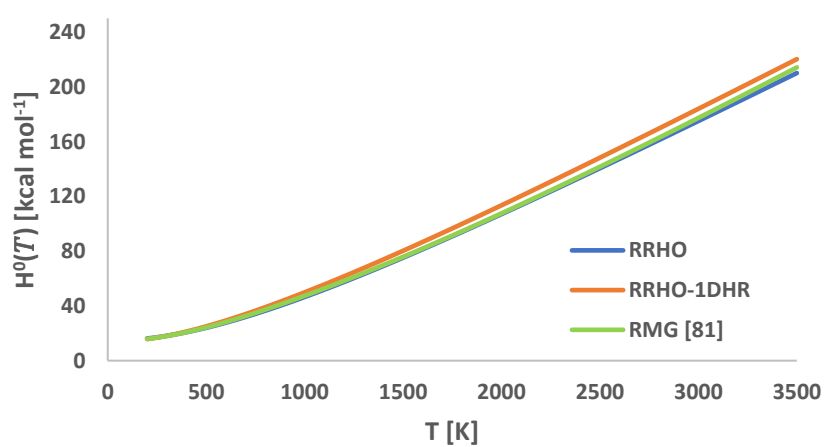
The NASA polynomials of two species selected from the set of 8 species in Table (3.3) were estimated (isoprene, species number 3, and 1,3-butadiene-2-ol, species number 4). The determination of the NASA polynomials of these two species should consider the presence of hindered rotors and the different type of theory used to describe their behaviour. The estimation of NASA polynomials with and without the explicit treatment of hindered rotors showed the importance of such description of this internal motion.

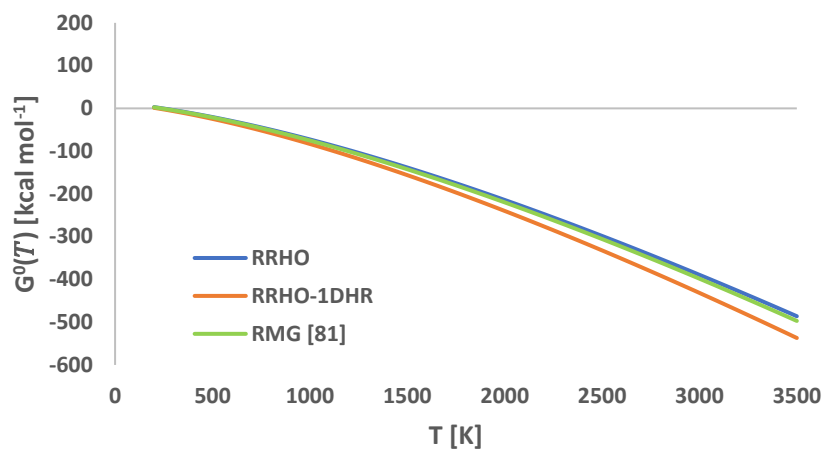
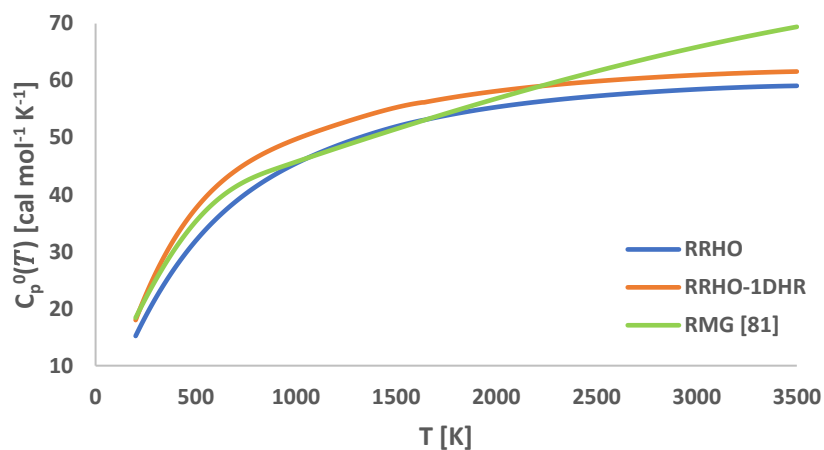
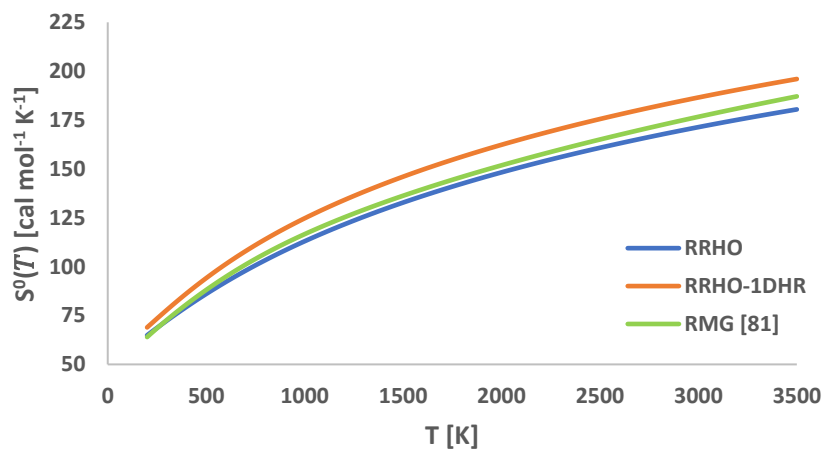
The NASA polynomials obtained with CHEMTP were then compared with the thermochemical parameters estimation using Benson group additivity [43] present in RMG database by Green et al. [81], between 200 [K] and 3500 [K].

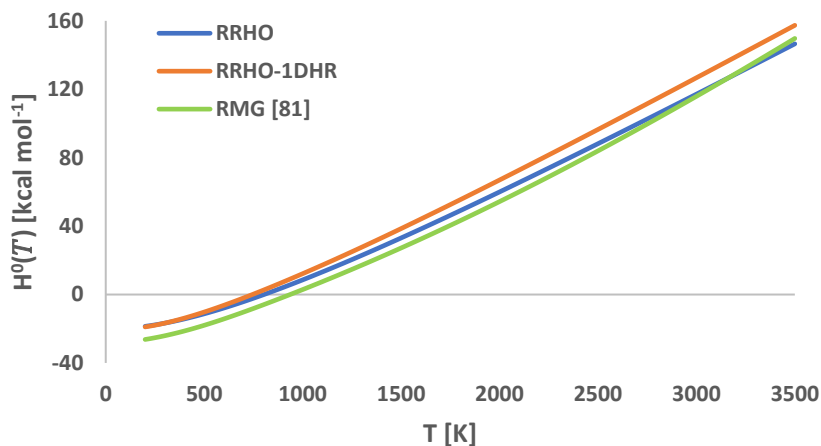
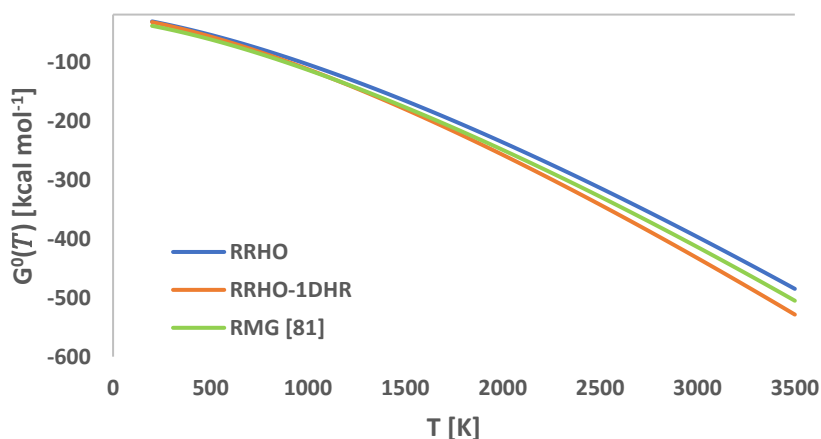
The estimated isoprene $C_p^0(T)$, $S^0(T)$, $H^0(T)$ and $G^0(T)$ using pure RRHO approximation of internal motions, RRHO approximation + 1D hindered rotor approximation and RMG data are reported in Figures (3.11) to (3.14).

The estimated 1,3-butadiene-2-ol $C_p^0(T)$, $S^0(T)$, $H^0(T)$ and $G^0(T)$ using pure RRHO approximation of internal motions, RRHO approximation + 1D hindered rotor approximation and RMG data are reported in Figures (3.15) to (3.18).

The estimated NASA polynomials coefficients are reported in Appendix C.

Figure 3.11: Isoprene $C_p^0(T)$ Figure 3.12: Isoprene $S^0(T)$ Figure 3.13: Isoprene $H^0(T)$

Figure 3.14: Isoprene $G^0(T)$ Figure 3.15: 1,3-butadiene-2-ol $C_p^0(T)$ Figure 3.16: 1,3-butadiene-2-ol $S^0(T)$

Figure 3.17: 1,3-butadiene-2-ol $H^0(T)$ Figure 3.18: 1,3-butadiene-2-ol $G^0(T)$

The $C_p^0(T)$ estimated by CHEMTP was in good agreement with the $C_p^0(T)$ estimated by Green et al. [81] for both isoprene and 1,3-butadiene-2-ol; the absolute difference was around 2 [cal mol⁻¹ K⁻¹] for both RRHO and RRHO+1DHR estimation with respect to RMG estimation up to 2500 [K]. For higher temperatures, the difference between the estimation of CHEMTP and the estimation of RMG increased up to 5 [cal mol⁻¹] for 1,3-butadiene-2-ol at 3500 [K] with respect to RRHO estimation. Using RRHO-1DHR approximation, the maximum error reported was around 10 [cal mol⁻¹ K⁻¹] for 1,3-butadiene-2-ol, at 3500 [K]. The agreement between CHEMTP and RMG estimation at lower temperatures was higher with respect to high temperature calculations. At 300 [K] the estimated values by RRHO and RRHO-1DHR differed from RMG evaluation by 3.2 [cal mol⁻¹ K⁻¹] and 1.4 [cal mol⁻¹ K⁻¹] respectively for 1,3-butadiene-2-ol and by 1.9 [cal mol⁻¹ K⁻¹] and 2.4 [cal mol⁻¹ K⁻¹] respectively for isoprene.

The $S^0(T)$ and $H^0(T)$ estimated by CHEMTP were consistent with RMG predictions; the estimated standard entropy and enthalpy by Green et al. was closer to the RRHO estimation by CHEMTP, while the RRHO+1DHR results overestimated the values of $S^0(T)$ and $H^0(T)$ by a maximum factor of 13 [$\text{cal mol}^{-1} \text{K}^{-1}$] for $S^0(T)$ and 8 [kcal mol^{-1}] for $H^0(T)$ in isoprene, at 3500 [K]. At lower temperatures, as for $C_p^0(T)$, the agreement between CHEMTP and RMG was increased with respect to high temperatures.

For isoprene, at 300 [K], RRHO and RRHO-1DHR approximations gave a discrepancy from RGM of 1.4 [$\text{cal mol}^{-1} \text{K}^{-1}$] and 6.3 [$\text{cal mol}^{-1} \text{K}^{-1}$] for $S^0(300 \text{ K})$, while for $H^0(300 \text{ K})$ the reported error was 0.11 [kcal mol^{-1}] and 0.12 [kcal mol^{-1}], respectively.

For 1,3-butadiene-2-ol, at 300 [K], RRHO and RRHO-1DHR approximations gave a discrepancy from RGM of 0.4 [$\text{cal mol}^{-1} \text{K}^{-1}$] and 5 [$\text{cal mol}^{-1} \text{K}^{-1}$] for $S^0(300 \text{ K})$, while for $H^0(300 \text{ K})$ the reported error was 8.1 [kcal mol^{-1}] and 8.2 [kcal mol^{-1}], respectively. Even if the estimated error at 300 [K] for $H^0(300 \text{ K})$ was close to the maximum error reported also at 3500 [K], the calculated $H^0(298.15 \text{ K})$ using RRHO and RRHO-1DHR (both equal to -16.76 [kcal mol^{-1}] = $-70,16$ [kJ mol^{-1}]) was closer to the experimental value of -77.0 ± 5.0 [kJ mol^{-1}] reported by Turecek [82] with respect to the estimated -101.05 [kJ mol^{-1}] by RMG.

The $G^0(T)$, calculated using the relation $G^0(T) = H^0(T) - TS^0(T)$, was in good agreement with $G^0(T)$ estimated by Green et al. [81]. At high temperatures the RRHO+1DHR model underestimated the values of Gibbs free energy by a maximum factor of 39.9 [kcal mol^{-1}] for isoprene at 3500 [K]. At lower temperatures the estimation difference is significantly decreased.

For isoprene, at 300 [K], RRHO and RRHO-1DHR approximations gave a discrepancy from RGM of 0.5 [kcal mol^{-1}] and 2 [kcal mol^{-1}], respectively.

For 1,3-butadiene-2-ol, at 300 [K], RRHO and RRHO-1DHR approximations gave a discrepancy from RGM of 7 [kcal mol^{-1}] and 5 [kcal mol^{-1}], respectively.

The RRHO and RRHO+1DHR estimations were consistent with the predictions of RMG [81] for most of the temperature range analysed; the major differences were reported in high temperature ranges (higher than 2000-2500 [K]), where the approximations of RRHO and RRHO+1DHR are likely to fail. Even though clear differences were reported in Figures (3.11) to (3.18), CHEMTP was able to predict the thermochemical parameters of isoprene and 1,3-butadiene-2-ol in an acceptable value range.

4. Conclusion and future development

The automation of chemical kinetics is a challenging task. The gradual transition from postdictive kinetics to predictive kinetics is possible thanks to the theories developed in the last decades, along with an exponentially increase in computer power and its availability. The accuracy achieved by computational chemistry is even better than experimental precision, in many cases; one of the main limitations remains the availability of resources to provide enough computational power for the calculations.

The reduction of the user effort, if not the total elimination, is desired in order to limit a possible source of errors and to have faster calculations, without the need of a debugging phase and the processing of input and output files. Part of this work was devoted to the creation of a protocol for the automatic creation of input data for EStokTP calculations. Using the Python code InChI2data, the user can create single or multiple `./data` subdirectories, avoiding the creation by hand of the Z-matrix of the molecule of interests, which is complicated for complex geometries. The possible construction of bad first guess structures and the improper definition of atomic order are overcome using InChI2data.

The exploration of a PES is another complex task; many reaction channels can be taken into consideration, but usually a limited number of channels are reactively interesting, having an activation energy lower with respect to other pathways. The code FragsGen developed in this work permits to understand the strength of the bonds present in the molecule and which is more likely to break, allowing a fast initial exploration of possible fragmentation pathways. FragsGen was successfully tested on 1,3-butadiene-2-ol, with results consistent with the literature ones. Even though an exact value of the bond energy is not assured, FragsGen can be used for implementation of low level of theory calculations and the determination of the most important reaction channels involving the rupture of a bond in a molecule.

The estimation of thermochemical parameters has advanced independently with respect to other topics involved in computationally chemistry; modern software like Auto-Mech, RMG, Genesys and Arkane implement various algorithms for the estimation of thermochemical parameters, ranging from group contribution methods like Joback's method or Benson's method, to atomization schemes at different level of complexity. In the present work a protocol that relies on EStokTP calculations has been developed through a Python code named CHEMTP; this allows to exploit EStokTP to predict the most important thermochemical parameters of a molecule,

namely $C_p^0(T)$, $S^0(T)$, $H^0(T)$ and $G^0(T)$ for a wide range of temperatures. The estimation of $\Delta H^0(0 K)$ has been performed on a set of 142 species studied by Klippenstein et al. [60] and compared with their collection of species; the study showed that, even without extrapolation to basis set limit and correction for core electrons correlation, the error committed with respect to the reference database is well below the chemical precision of 1 [kcal mol⁻¹], with a confidence of 95%. Increasing the level of theory led to a general, but not guaranteed decrease of the error; the order of magnitude of absolute error decreased even by a factor of 3 for one of the eight species of the subset tested. The thermal correction on $\Delta H^0(0 K)$ to obtain $\Delta H^0(298.15 K)$ was applied on the same subset of eight species and resulted in seven out of eight species' enthalpies with an absolute error lower than 1 [kcal mol⁻¹]; this shows that the correction step of enthalpy in CHEMTP can consistently estimate $\Delta H^0(298.15 K)$ for the successive computation of NASA polynomials. Finally, isoprene and 1,3-butadiene-2-ol NASA polynomials were estimated both within the full RRHO approximation and with explicit treatment of 1D hindered rotors. The thermochemical parameters were compared with Green et al. RMG [81] estimations in the 300-3500 [K] temperature range. The most marked discrepancy was always observed in the high temperature regime (2000-3500 [K]), where the RRHO approximation is more likely to give poor results and the rotors assume more and more the characteristics of free rotors. Despite this, a general agreement with Green et al. estimations was guaranteed in the range 300-2000 [K], which is of interest for both the atmospheric and combustion kinetic.

More precise thermochemical estimations could be computed introducing a series of corrections in the calculation of $C_p^0(T)$ and $S^0(T)$ or in the estimation of $\Delta H^0(0 K)$. 2D and 3D hindered rotor with a multiconfigurational sampling of torsional angles would surely benefit the evaluation of $C_p^0(T)$ and $S^0(T)$, but this is not feasible for complex molecular structures due to lack of computational resources, leaving the 1D hindered rotor treatment as the best possible approximation of this internal motion as of today. Different type of corrections for electronic energy estimation, described by Attila et al. [72], along with corrections for possible ZPE anharmonicity, could be implemented for better estimation of $\Delta H^0(0 K)$. A different approach for CBH rung construction, using different types of atomization schemes with respect to the ones used in the present work or the implantation of higher-level atomization schemes, still represent a possible improvement of the protocol developed in CHEMTP.

Bibliography

- [1] Sébastien Thion, Pascal Diévar, Pierre Van Cauwenberghe, Guillaume Dayma, Zeynep Serinyel, Philippe Dagaut, *An experimental study in a jet-stirred reactor and a comprehensive kinetic mechanism for the oxidation of methyl ethyl ketone*, Proceedings of the Combustion Institute, Volume 36, Issue 1 (2017) 459-467
- [2] Xiao, H., Valera-Medina, A., *Chemical Kinetic Mechanism Study on Premixed Combustion of Ammonia/Hydrogen Fuels for Gas Turbine Use*, ASME. J. Eng. Gas Turbines Power (2017), <https://doi.org/10.1115/1.4035911>
- [3] Zettervall N, Fureby C, Nilsson EJK., *Evaluation of Chemical Kinetic Mechanisms for Methane Combustion: A Review from a CFD Perspective*. Fuels. (2021); 210-240, <https://doi.org/10.3390/fuels2020013>
- [4] Green, W.H., *Moving from postdictive to predictive kinetics in reaction engineering*, AIChE Journal 66 (2020), e17059
- [5] Alberto Cuoci, C. Thomas Avedisian, Jordan D. Brunson, Songtao Guo, Alireza Dalili, Yujie Wang, Marco Mehl, Alessio Frassoldati, Kalyanasundaram Seshadri, John E. Dec, Dario Lopez-Pintor, *Simulating combustion of a seven-component surrogate for a gasoline/ethanol blend including soot formation and comparison with experiments*, Fuel, Volume 288 (2021), ISSN 0016-2361, <https://doi.org/10.1016/j.fuel.2020.119451>
- [6] Alessandro Stagni, Alberto Cuoci, Alessio Frassoldati, Tiziano Faravelli, and Eliseo Ranzi, *Lumping and Reduction of Detailed Kinetic Schemes: an Effective Coupling*, Industrial & Engineering Chemistry Research, 53-22 (2014), <https://doi.org/10.1021/ie403272f>
- [7] R. A. Marcus; *Unimolecular Dissociations and Free Radical Recombination Reactions*, J. Chem. Phys. 1 March 1952, 20 (3): 359–364. <https://doi.org/10.1063/1.1700424>
- [8] Stephen J. Klippenstein, Carlo Cavallotti, *Ab initio kinetics for pyrolysis and combustion systems*, Computer Aided Chemical Engineering, Volume 45 (2019), 115-167, <https://doi.org/10.1016/B978-0-444-64087-1.00002-4>
- [9] Cavallotti, C., Pelucchi, M., Georgievskii, Y. & Klippenstein, S. J. *EStokTP: Electronic Structure to Temperature- and Pressure-Dependent Rate Constants—A Code for Automatically Predicting the Thermal Kinetics of Reactions*. J. Chem. Theory Comput. 15, 1122–1145 (2019).
- [10] David O. Harris, Gail G. Engerholm, William D. Gwinn; *Calculation of Matrix Elements for One-Dimensional Quantum-Mechanical Problems and the*

- Application to Anharmonic Oscillators*. J. Chem. Phys. 1 September 1965; 43 (5): 1515–1517. <https://doi.org/10.1063/1.1696963>
- [11] Merzbacher, E. *The Early History of Quantum Tunneling*. Phys. Today 55, 44–49 (2002).
- [12] Erwin Schrödinger, *Quantisierung als Eigenwertproblem (Erste Mitteilung)* [Quantization as an eigenvalue problem (first communication)], in *Annalen der Physik*, vol. 79, 1926, pp. 361-376.
- [13] Louis De Broglie, *Recherches sur la théorie des Quanta*, in *Annales de Physique*, vol. 10, n. 3, 1925, pp. 22-128, DOI:10.1051/anphys/192510030022
- [14] Hartree, D. (1928). *The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods*. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1), 89-110, doi:10.1017/S0305004100011919
- [15] Lykos, P. & Pratt, G. W. *Discussion on The Hartree-Fock Approximation*. *Rev. Mod. Phys.* 35, 496–501 (1963).
- [16] Bartlett, R. J. & Stanton, J. F. *Applications of Post-Hartree-Fock Methods: A Tutorial*. in *Reviews in Computational Chemistry* (1994)
- [17] C. David Sherrill, Henry F. Schaefer, *The Configuration Interaction Method: Advances in Highly Correlated Approaches*, Editor(s): Per-Olov Löwdin, John R. Sabin, Michael C. Zerner, Erkki Brändas, *Advances in Quantum Chemistry*, Academic Press, Volume 34 (1999), Pages 143-269, [https://doi.org/10.1016/S0065-3276\(08\)60532-8](https://doi.org/10.1016/S0065-3276(08)60532-8)
- [18] Chr. Møller, M. S. Plesset, *Note on an Approximation Treatment for Many-Electron Systems*, *American Physical Society*, 46-7 (1934), <https://link.aps.org/doi/10.1103/PhysRev.46.618>
- [19] Hohenberg, P. & Kohn, W. *Inhomogeneous Electron Gas*. *Phys. Rev.* 136, B864– B871 (1964).
- [20] Frisch, M. J.; G. W. T., Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; ; Raghavachari, K.; A. R., J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V.

- G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, Fox, D. J.; *Gaussian 09*, Gaussian, Inc.: Wallingford CT, 2009W. J. Hehre, W. A. Lathan, R. Ditchfield, M. D. Newton, and J. A. Pople, *Gaussian 70* (Quantum Chemistry Program Exchange, Program No. 237, 1970)
- [21] Werner, H.-J., Knowles, P.J., Knizia, G., Manby, F.R. and Schütz, M. (2012), *Molpro: a general-purpose quantum chemistry program package*. WIREs Comput Mol Sci, 2: 242-253. <https://doi.org/10.1002/wcms.82>
- [22] Snyder, L. C.; Basch, H. J. Am. Chem. Soc. 1969, 91, 2189–2198.
- [23] Hehre, W. J.; Ditchfield, R.; Radom, L.; Pople, J. A. J. Am. Chem. Soc. 1970, 92, 4796–4801
- [24] Radom, L.; Hehre, W. J.; Pople, J. A. J. Am. Chem. Soc. 1971, 93, 289–300
- [25] Hehre, W. J.; Radom, L.; Schleyer, P. v. R.; Pople, J. A. *Ab Initio Molecular Orbital Theory*; Wiley-Interscience: New York, 1986.
- [26] George, P.; Trachtman, M.; Bock, C. W.; Brett, A. M. Theor. Chem. Acc. 1975, 38, 121–129.
- [27] George, P.; Trachtman, M.; Bock, C. W.; Brett, A. M. Tetrahedron 1976, 32, 317–32
- [28] George, P.; Trachtman, M.; Bock, C. W.; Brett, A. M. J. Chem. Soc., Perkin Trans. 2 1976, 1222–1227
- [29] Hess, B. A., Jr.; Schaad, L. J. J. Am. Chem. Soc. 1983, 105, 7500–7505.
- [30] Nyulaszi, L.; Vamai, P.; Veszpremi, T. J. Mol. Struct. THEO-CHEM 1995, 358, 55–61.
- [31] Vianello, R.; Liebman, J. F.; Maksic, Z. B. Chem.-Eur. J. 2004, 10, 5751–5760.
- [32] Vianello, R.; Liebman, J. F.; Maksic, Z. B.; Muller, T. J. J. Phys. Chem. A 2005, 109, 10594–10606.
- [33] Chestnut, D. B.; Davis, K. M. J. Comput. Chem. 1997, 18, 584–593.
- [34] El-Nahas, A. M.; Bozelli, J. W.; Simmie, J. M.; Navarro, M. V.; Black, G.; Curran, H. J. J. Phys. Chem. A 2006, 110, 13618–13623.
- [35] George, P.; Bock, C. W.; Trachtman, M. Theor. Chim. Acta 1987, 71, 289–298.
- [36] Gimarc, B. M.; Zhao, M. J. Phys. Chem. 1994, 98, 1596–1600.
- [37] Zhao, M.; Gimarc, B. M. J. Phys. Chem. 1993, 97, 4023–4030.
- [38] Warren, D. S.; Gimarc, B. M. J. Phys. Chem. 1993, 97, 4031–4035.
- [39] Joseph W. Ochterski, *Thermochemistry in Gaussian* (2000), Gaussian Inc.
- [40] Joback, Kevin G. *A unified approach to physical property estimation using multivariate statistical techniques*, Diss. Massachusetts Institute of Technology, 1984.
- [41] K.G. Joback and R.C. Reid, *Estimation of pure-component properties from group-contributions*, Chemical Engineering Communications, Volume 57 (1987), 1-6

- (233-243), doi:10.1080/00986448708960487
- [42] Constantinou, L. and Gani, R. (1994), *New group contribution method for estimating properties of pure compounds*, AIChE J., 40: 1697-1710. <https://doi.org/10.1002/aic.690401011>
- [43] Benson, S. W., Cruickshank, F. R., Golden, D. M., Haugen, G. R., O'neal, H. E., Rodgers, A. S., ... & Walsh, R. (1969). *Additivity rules for the estimation of thermochemical properties*. Chemical Reviews, 69(3), 279-324.
- [44] Ramabhadran, Raghunath O. and Raghavachari, Krishnan, *Theoretical Thermochemistry for Organic Molecules: Development of the Generalized Connectivity-Based Hierarchy*, Journal of Chemical Theory and Computation, (2011) 7 (7), 2094-2103, DOI: 10.1021/ct200279q
- [45] Ramabhadran, Raghunath O. and Raghavachari, Krishnan, *Connectivity-Based Hierarchy for Theoretical Thermochemistry: Assessment Using Wave Function-Based Methods*, The Journal of Physical Chemistry A (2012) 116 (28), 7531-7537, DOI: 10.1021/jp301421a
- [46] Ramabhadran, Raghunath O. and Raghavachari, Krishnan, *Extrapolation to the Gold-Standard in Quantum Chemistry: Computationally Efficient and Accurate CCSD(T) Energies for Large Molecules Using an Automated Thermochemical Hierarchy*, Journal of Chemical Theory and Computation (2013) 9 (9), 3986-3994, DOI: 10.1021/ct400465q
- [47] Sengupta, Arkajyoti and Raghavachari, Krishnan, *Prediction of Accurate Thermochemistry of Medium and Large Sized Radicals Using Connectivity-Based Hierarchy (CBH)*, Journal of Chemical Theory and Computation (2014) 10 (10), 4342-4350, DOI: 10.1021/ct500484f
- [48] Gordon, S., and McBride, B.J., 1971, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations*, NASA SP-273
- [49] Svehla, R.A., and McBride, B.J., 1973, *FORTTRAN IV Computer Program for Calculation of Thermodynamic and Transport Properties of Complex Chemical Systems*, NASA TN D-7056.
- [50] Gordon, S., and McBride, B.J., 1976, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations*, NASA SP-273, Interim Revision
- [51] Gordon, S., McBride, B.J., and Zeleznik, F.J., 1984, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. Supplement I: Transport Properties*, NASA TM-86885.
- [52] Gordon, S., and McBride, B.J., 1988, *Finite Area Combustor Theoretical Rocket Performance*, NASA TM-100785.

- [53] McBride, B.J, Reno, M.A., and Gordon, S., 1994, *CET93 and CETPC: An Interim Updated Version of the NASA Lewis Computer Program for Calculating Complex Chemical Equilibria With Applications*, NASA TM-4557.
- [54] Bonnie J. McBride, Michael J. Zehe, and Sanford Gordon, 2002, *NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species*, NASA/TP-2002-211556
- [55] CHEMKIN webpage: <https://www.ansys.com/products/fluids/ansys-chemkin-pro> (2023)
- [56] Connie W. Gao, Joshua W. Allen, William H. Green, Richard H. West, *Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms*, *Computer Physics Communications*, Volume 203 (2016) 212-225, <https://doi.org/10.1016/j.cpc.2016.02.013>.
- [57] Sarah N. Elliott, Kevin B. Moore, Andreas V. Copan, Murat Keçeli, Carlo Cavallotti, Yuri Georgievskii, Henry F. Schaefer, Stephen J. Klippenstein, *Automated theoretical chemical kinetics: Predicting the kinetics for the initial stages of pyrolysis*, *Proceedings of the Combustion Institute*, Volume 38, Issue 1, Pages 375-384 (2021), <https://doi.org/10.1016/j.proci.2020.06.019>.
- [58] AutoMech Github web page: <https://github.com/Auto-Mech> (2023)
- [59] Murat Keçeli, Sarah N. Elliott, Yi-Pei Li, Matthew S. Johnson, Carlo Cavallotti, Yuri Georgievskii, William H. Green, Matteo Pelucchi, Justin M. Wozniak, Ahren W. Jasper, Stephen J. Klippenstein, *Automated computational thermochemistry for butane oxidation: A prelude to predictive automated combustion kinetics*, *Proceedings of the Combustion Institute*, Volume 37, Issue 1, Pages 363-371 (2019), <https://doi.org/10.1016/j.proci.2018.07.113>.
- [60] Elliott, Sarah N. and Keçeli, Murat and Ghosh, Manik K. and Somers, Kieran P. and Curran, Henry J. and Klippenstein, Stephen J., *High-Accuracy Heats of Formation for Alkane Oxidation: From Small to Large via the Automated CBH-ANL Method*, *The Journal of Physical Chemistry A*, 127-6 1512-1532 (2023), <https://doi.org/10.1021/acs.jpca.2c07248>
- [61] Ruben Van de Vijver, Judit Zádor, *KinBot: Automated stationary point search on potential energy surfaces*, *Computer Physics Communications*, Volume 248 (2020), <https://doi.org/10.1016/j.cpc.2019.106947>.
- [62] Nick M. Vandewiele, Kevin M. Van Geem, Marie-Françoise Reyniers, Guy B. Marin, *Genesys: Kinetic model construction using chemo-informatics*, *Chemical Engineering Journal*, Volumes 207-208, Pages 526-538 (2012), <https://doi.org/10.1016/j.cej.2012.07.014>.
- [63] Dana, AG, Johnson, MS, Allen, JW, et al. *Automated reaction kinetics and network exploration (Arkane): A statistical mechanics, thermodynamics, transition state theory, and master equation software*. *Int J Chem Kinet*. 2023; 55: 300-

323. <https://doi.org/10.1002/kin.21637>
- [64] Heller, S., McNaught, A., Stein, S. et al. *InChI - the worldwide chemical structure identifier standard*. *J Cheminform* 5, 7 (2013). <https://doi.org/10.1186/1758-2946-5-7>
- [65] Weininger, David, *SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules*, *Journal of Chemical Information and Computer Sciences*, 28 1 32-36 (1988), <https://doi.org/10.1021/ci00057a005>
- [66] EStokTP manual: <https://github.com/EStokTP/EStokTP/tree/main/manual> (2023)
- [67] Y. Georgievskii, J. A. Miller, M. P. Burke, and S. J. Klippenstein, *Reformulation and Solution of the Master Equation for Multiple-Well Chemical Reactions*, *J. Phys. Chem. A*, 117, 12146-12154 (2013), <https://doi.org/10.1021/jp4060704>
- [68] RDKit: Open-source cheminformatics. <https://www.rdkit.org> (2023)
- [69] Deena A. Koniver et al., *Wiswesser Line Notation: Simplified Techniques for Converting Chemical Structures to WLN*, *Science* 176, 1437-1439 (1972). DOI:10.1126/science.176.4042.1437
- [70] Georgievskii, Y.; Klippenstein, S. J. *x2z, A Code for Converting from Cartesians to Internals with Well-Defined Torsional Coordinates*. <https://github.com/Auto-Mech/x2z> (2023).
- [71] Sandeep Sharma, Sumathy Raman, and William H. Green, *Intramolecular Hydrogen Migration in Alkylperoxy and Hydroperoxyalkylperoxy Radicals: Accurate Treatment of Hindered Rotors*, *The Journal of Physical Chemistry A* 2010 114 (18), 5689-5701, DOI: 10.1021/jp9098792
- [72] Attila Tajti, Péter G. Szalay, Attila G. Császár, Mihály Kállay, Jürgen Gauss, Edward F. Valeev, Bradley A. Flowers, Juana Vázquez, John F. Stanton; *HEAT: High accuracy extrapolated ab initio thermochemistry*. *J. Chem. Phys.* 15 December 2004; 121 (23): 11599–11613. <https://doi.org/10.1063/1.1811608>
- [73] J.M.L. Martin, *Ab initio total atomization energies of small molecules – towards the basis set limit*, *Chem. Phys. Lett.* 259 (1996) 669-678.
- [74] Larry A. Curtiss, Krishnan Raghavachari, Paul C. Redfern, John A. Pople; *Assessment of Gaussian-2 and density functional theories for the computation of enthalpies of formation*. *J. Chem. Phys.* 15 January 1997; 106 (3): 1063–1079. <https://doi.org/10.1063/1.473182>
- [75] Argonne National Laboratory–Active Thermochemical Tables (ANL-ATcT): <https://atct.anl.gov/Thermochemical%20Data/version%201.124/index.php> (2023)

- [76] NumPy website: <https://numpy.org/> (2023)
- [77] SciPy website: <https://scipy.org/> (2023)
- [78] Thomas F. Coleman, Yuying Li, *An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds*, SIAM Journal on Optimization, 6, 418-445 (1996), doi: 10.1137/0806023
- [79] Sui So, Uta Wille, and Gabriel da Silva, *Photoisomerization of Methyl Vinyl Ketone and Methacrolein in the Troposphere: A Theoretical Investigation of Ground-State Reaction Pathways*, ACS Earth and Space Chemistry (2018) 2 (8), 753-763, DOI: 10.1021/acsearthspacechem.8b00066
- [80] Stephen J. Blanksby and G. Barney Ellison, *Bond Dissociation Energies of Organic Molecules*, Accounts of Chemical Research (2003) 36 (4), 255-263, DOI: 10.1021/ar020230d
- [81] RMG database website: <https://rmg.mit.edu/database> (2023)
- [82] Turecek, F., *2-Hydroxybutadiene: preparation, ionization energy and heat of formation*, Tetrahedron Lett., 1984, 25, 5133-5134.

A. CBH reference species

A. CBH reference species

Reference species for level 1 estimations; both basis sets for the ω B97X-D and B2PLYP-D3 levels of theory are jun-cc-pVTZ. All quantities are expressed in [Ha].

	<i>SMILES</i>	<i>ωB97X-D</i>	<i>B2PLYP-D3</i>	<i>$\Delta H^0(0\text{ K})$</i>
<i>CBH-0</i>				
1)	C	-40.475078	-40.442432	-0.02535
2)	[CH3]	-39.809241	-39.781952	0.057083
3)	O	-76.418056	-76.392238	-0.10903
4)	[OH]	-75.731988	-75.709854	0.0142
5)	N	-56.531974	-56.503121	-0.01469
6)	[NH2]	-55.862925	-55.83888	0.071956
7)	[H][H]	-1.166500	-1.159819	0.000000
<i>CBH-1</i>				
8)	CC	-79.757932	-79.699436	-0.02605
9)	[CH2]C	-79.099494	-79.045654	0.05004
10)	C=C	-78.537396	-78.490257	0.023192
11)	[CH]=C	-77.863524	-77.820021	0.114744
12)	C#C	-77.300283	-77.266043	0.087176
13)	[C]#C	-76.586668	-76.555498	0.214725
14)	CO	-115.683041	-115.631411	-0.07238
15)	[CH2]O	-115.032973	-114.985397	-0.00395
16)	C[O]	-115.021798	-114.972083	0.011015
17)	C=O	-114.482108	-114.440616	-0.04014
18)	[CH]=O	-113.843672	-113.807557	0.015761
19)	OO	-151.538086	-151.492851	-0.04929
20)	[O]O	-150.906045	-150.86209	0.005767
21)	CN	-95.802531	-95.747577	-0.00249
22)	[CH2]N	-95.158034	-95.107312	0.060693
23)	C[NH]	-95.147101	-95.096082	0.071564
24)	C=N	-94.592421	-94.548833	0.036724
25)	[CH]=N	-93.941971	-93.903655	0.10505
26)	C=[N]	-93.955065	-93.914647	0.092207
27)	C#N	-93.40611	-93.376088	0.049392

28)	[C]#N	-92.699788	-92.676401	0.166342
29)	NO	-131.691802	-131.642994	-0.0126
30)	[NH]O	-131.060109	-131.01497	0.042765
31)	N[O]	-131.075972	-131.028964	0.026924
32)	N=O	-130.466536	-130.427839	0.041874
33)	[N]=O	-129.892336	-129.858798	0.034522
34)	NN	-111.826644	-111.774375	0.042552
35)	[NH]N	-111.199658	-111.151303	0.089419
36)	N=N	-110.618145	-110.577184	0.078888
37)	[N]=N	-110.019874	-109.983941	0.096035

CBH-2

38)	CCC	-119.044808	-118.960776	-0.03151
39)	[CH2]CC	-118.38572	-118.306402	0.045089
40)	C[CH]C	-118.39195	-118.312017	0.040209
41)	C=CC	-117.829267	-117.756689	0.013274
42)	[CH2]C=C	-117.192464	-117.123374	0.068539
43)	C=[C]C	-117.161031	-117.091906	0.100175
44)	[CH]=CC	-117.153851	-117.085142	0.105957
45)	C#CC	-116.597702	-116.537614	0.073384
46)	C#C[CH2]	-115.955567	-115.89837	0.13485
47)	[C]#CC	-115.883885	-115.8272	0.201306
48)	COC	-154.956175	-154.879443	-0.06342
49)	[CH2]OC	-154.305883	-154.233478	-0.00498
50)	CCO	-154.974498	-154.897393	-0.08277
51)	[CH2]CO	-154.314601	-154.242225	-0.00498
52)	C[CH]O	-154.327513	-154.254196	-0.01632
53)	CC[O]	-154.314387	-154.23913	0.00048
54)	C=CO	-153.766095	-153.699357	-0.04318
55)	[CH]=CO	-153.08512	-153.02235	0.052561
56)	C=[C]O	-153.095058	-153.031653	0.046208
57)	C=C[O]	-153.133984	-153.069941	0.008775
58)	CC=O	-153.783471	-153.716332	-0.05905
59)	C[C]=O	-153.143672	-153.081493	-0.00132
60)	COO	-190.814923	-190.745121	-0.04375
61)	[CH2]OO	-190.162058	-190.096232	0.027842
62)	CO[O]	-190.185331	-190.116847	0.008577
63)	OCO	-190.914973	-190.844571	-0.14441
64)	[O]CO	-190.25441	-190.184926	0.008577
65)	O=CO	-189.745167	-189.68429	-0.14133
66)	[O]C=O	-189.068876	-189.15562	-0.04793
67)	O=[C]O	-189.09085	-189.034698	-0.06901
68)	N=C=O	-168.669947	-168.669947	-0.04417
69)	N#CO	-168.624681	-168.624681	-0.0047

70)	N#C[O]	-167.994379	-167.994379	0.048349
71)	C=NO	-169.779137	-169.779136	0.011617
72)	[CH]=NO	-169.108823	-169.051088	0.103561
73)	NC=O	-169.842016	-169.831043	-0.06802
74)	N=C[O]	-167.994379	-167.994379	0.038546
75)	N=[C]O	-169.181874	-169.181873	0.028223
76)	CC(C)C	-158.33411	-158.224859	-0.04033
77)	[CH2]C(C)C	-157.674004	-157.569738	0.037025
78)	CCC	-157.684799	-157.579268	0.028688
79)	CC(C)O	-194.267353	-194.165253	-0.09471
80)	[CH2]C(C)O	-193.606566	-193.509437	-0.01661
81)	CCO	-193.622326	-193.523555	-0.02982
82)	CC(C)[O]	-193.604138	-193.503716	-0.00957
83)	C=C(C)C	-157.122076	-157.024366	0.001592
84)	[CH]=C(C)C	-156.445053	-156.351176	0.095658
85)	[CH2]C=C(C)C	-156.482833	-156.38865	0.059364
86)	CC(C)=O	-193.082478	-192.98967	-0.07617
87)	C=C(C)O	-193.061741	-192.969953	-0.05765
88)	[CH2]C=C(O)	-192.420058	-192.331917	0.001524
89)	CC(=O)[CH2]	-192.430968	-192.341869	-0.00704
90)	[CH]=C(C)O	-192.380519	-192.292794	0.038686
91)	[CH2]C(=O)O	-228.389465	-228.30722	-0.08635
92)	CC([O])=O	-228.373753	-228.298763	-0.0694
93)	CC(=O)O	-229.044598	-228.958011	-0.15941
94)	C=C(O)O	-229.001406	-228.914768	-0.11718
95)	CC(=N)O	-209.134064	-209.04589	-0.08364
96)	CN(C)C	-174.361567	-174.25629	0.000849
97)	[CH2]N(C)C	-173.717665	-173.61691	0.063264
98)	CC(C)(C)C	-197.624358	-197.490237	-0.05113
99)	CC(C)(C)O	-233.560045	-233.432936	-0.11906
100)	CC(C)(C)N	-213.674928	-213.544506	-0.03313
101)	[CH2]C(C)(C)C	-196.962968	-196.833967	0.027681
102)	[CH2]C(C)(O)C	-232.899577	-232.879892	-0.02909

Reference species for high level estimations; both basis sets for level 1 estimations using the ω B97X-D and B2PLYP-D3 levels of theory are jun-cc-pVTZ. CCSD(T) level of theory with extrapolation to basis set limit and correction for core electrons correlation is also implemented. All quantities are expressed in [Ha].

	SMILES	ω B97X-D	B2PLYP-D3	$\Delta H^0(0\text{ K})$
		CBH-0		
1)	C	-40.467366	-40.467232	-0.02535
2)	[CH3]	-39.802698	-39.802515	0.057083

3)	O	-76.414421	-76.414675	-0.10903
4)	[OH]	-75.726387	-75.726465	0.0142
5)	N	-56.527212	-56.5273	-0.01469
6)	[NH2]	-55.857959	-55.858006	0.071956
7)	[H][H]	-1.165225	-1.16516	0.000000

CBH-1

8)	CC	-79.74515	-79.745056	-0.02605
9)	[CH2]C	-79.086548	-79.086366	0.05004
10)	C=C	-78.530703	-78.530865	0.023192
11)	[CH]=C	-77.856427	-77.856455	0.114744
12)	C#C	-77.309239	-77.309835	0.087176
13)	[C]#C	-76.590939	-76.591475	0.214725
14)	CO	-115.672796	-115.67293	-0.07238
15)	[CH2]O	-115.021581	-115.021673	-0.00395
16)	C[O]	-115.006829	-115.006765	0.011015
17)	C=O	-114.475337	-114.475504	-0.04014
18)	[CH]=O	-113.836735	-113.836933	0.015761
19)	OO	-151.529706	-151.530589	-0.04929
20)	[O]O	-150.892156	-150.892627	0.005767
21)	CN	-95.79187	-95.79196	-0.00249
22)	[CH2]N	-95.14599	-95.145975	0.060693
23)	C[NH]	-95.134747	-95.13479	0.071564
24)	C=N	-94.587407	-94.587673	0.036724
25)	[CH]=N	-93.936474	-93.936666	0.10505
26)	C=[N]	-93.948798	-93.94888	0.092207
27)	C#N	-93.409414	-93.40995	0.049392
28)	[C]#N	-92.709447	-92.709899	0.166342
29)	NO	-131.682996	-131.683514	-0.0126
30)	[NH]O	-131.049073	-131.049468	0.042765
31)	N[O]	-131.061041	-131.061184	0.026924
32)	N=O	-130.462897	-130.463545	0.041874
33)	[N]=O	-129.88748	-129.887868	0.034522
34)	NN	-111.817401	-111.817746	0.042552
35)	[NH]N	-111.187944	-111.188099	0.089419
36)	N=N	-110.61529	-110.61588	0.078888
37)	[N]=N	-110.015643	-110.016063	0.096035

CBH-2

38)	CCC	-119.027547	-119.027516	-0.03151
39)	[CH2]CC	-118.368402	-118.368226	0.045089
40)	C[CH]C	-118.373345	-118.373199	0.040209
41)	C=CC	-117.817461	-117.817636	0.013274
42)	[CH2]C=C	-117.179759	-117.179672	0.068539
43)	C=[C]C	-117.147838	-117.147844	0.100175

44)	[CH]=CC	-117.142048	-117.14207	0.105957
45)	C#CC	-116.592334	-116.592343	0.073384
46)	C#C[CH2]	-115.948178	-115.948263	0.13485
47)	[C]#CC	-115.881844	-115.882375	0.201306
48)	COC	-154.94054	-154.940557	-0.06342
49)	[CH2]OC	-154.289324	-154.289286	-0.00498
50)	CCO	-154.959787	-154.959945	-0.08277
51)	[CH2]CO	-154.299791	-154.299805	-0.00498
52)	C[CH]O	-154.310946	-154.311015	-0.01632
53)	CC[O]	-154.294763	-154.295389	0.00048
54)	C=CO	-153.7553	-153.755755	-0.04318
55)	[CH]=CO	-153.074465	-153.074782	0.052561
56)	C=[C]O	-153.082811	-153.083158	0.046208
57)	C=C[O]	-153.12037	-153.120366	0.008775
58)	CC=O	-153.7711	-153.77124	-0.05905
59)	C[C]=O	-153.131025	-153.1311	-0.00132
60)	COO	-190.801152	-190.80188	-0.04375
61)	[CH2]OO	-190.146983	-190.147743	0.027842
62)	CO[O]	-190.165941	-190.166204	0.008577
63)	OCO	-190.902973	-190.903309	-0.14441
64)	[O]CO	-190.236511	-190.236641	0.008577
65)	O=CO	-189.734536	-189.734909	-0.14133
66)	[O]C=O	-189.055754	-189.075164	-0.04793
67)	O=[C]O	-189.079532	-189.079921	-0.06901
68)	N=C=O	-168.661397	-168.661397	-0.04417
69)	N#CO	-168.621824	-168.621824	-0.0047
70)	N#[C]O	-167.985353	-167.985353	0.048349
71)	C=NO	-169.770281	-169.77028	0.011617
72)	[CH]=NO	-169.098946	-169.098946	0.103561
73)	NC=O	-169.832764	-169.832657	-0.06802
74)	N=C[O]	-167.985353	-167.985353	0.038546
75)	N=[C]O	-169.171624	-169.171624	0.028223
76)	CC(C)C	-158.31312	-158.313164	-0.04033
77)	[CH2]C(C)C	-157.653218	-157.653131	0.037025
78)	CCC	-157.661451	-157.66134	0.028688
79)	CC(C)O	-194.249054	-194.249285	-0.09471
80)	[CH2]C(C)O	-193.588341	-193.588422	-0.01661
81)	CCO	-193.601296	-193.601404	-0.02982
82)	CC(C)[O]	-193.580913	-193.580946	-0.00957
83)	C=C(C)C	-157.106005	-157.106189	0.001592
84)	[CH]=C(C)C	-156.429055	-156.429068	0.095658
85)	[CH2]C(=C)C	-156.466177	-156.46605	0.059364
86)	CC(C)=O	-193.065441	-193.065582	-0.07617
87)	C=C(C)O	-193.046773	-193.047203	-0.05765

88)	[CH2]C(=C)O	-192.404335	-192.404524	0.001524
89)	CC(=O)[CH2]	-192.412994	-192.412988	-0.00704
90)	[CH]=C(C)O	-192.365762	-192.36607	0.038686
91)	[CH2]C(=O)O	-228.373411	-228.373648	-0.08635
92)	CC([O])=O	-228.355076	-228.402483	-0.0694
93)	CC(=O)O	-229.029377	-229.029701	-0.15941
94)	C=C(O)O	-228.987361	-228.98801	-0.11718
95)	CC(=N)O	-209.120835	-209.121288	-0.08364
96)	CN(C)C	-174.34198	-174.342106	0.000849
97)	[CH2]N(C)C	-173.696795	-173.696806	0.063264
98)	CC(C)(C)C	-197.600449	-197.600566	-0.05113
99)	CC(C)(C)O	-233.538637	-233.538909	-0.11906
100)	CC(C)(C)N	-213.653342	-213.653626	-0.03313
101)	[CH2]C(C)(C)C	-196.962968	-196.833967	0.027681
102)	[CH2]C(C)(O)C	-232.898434	-232.885882	-0.02909

B. Estimated standard enthalpy @ 0 K

B. Estimated standard enthalpy @ 0 K

$\delta\Delta H^0(0\text{ K})$ is the absolute error between the $\Delta H^0(0\text{ K})$ estimated by Klippenstein et al. and the $\Delta H^0(0\text{ K})$ using the CHEMTP protocol, in [kcal mol⁻¹]. $\delta\Delta H^0(0\text{ K})\%$ is the relative percentage error. The $\omega\text{B97X-D/jun-cc-pVTZ}$ level of theory is used for all the species estimated with CHEMPT.

SMILES	KL et al. [60]	CHEMTP	$\delta\Delta H^0(0\text{ K})$	$\delta\Delta H^0(0\text{ K})\%$
<i>Alkanes</i>				
CC	-16.49	-15.67	-0.82	4.97
CCC	-19.95	-19.31	-0.64	3.21
CC(C)C	-25.39	-23.79	-1.60	6.28
CCCC	-23.61	-23.40	-0.21	0.87
CC(C)(C)C	-31.85	-28.87	-2.98	9.34
CCC(C)C	-28.50	-28.45	-0.05	0.17
CCCCC	-27.27	-27.03	-0.24	0.87
CC(C(C)C)C	-32.61	-32.67	0.06	0.18
CCC(C)(C)C	-34.28	-34.56	0.28	0.81
CCCCCC	-30.97	-30.71	-0.26	0.84
CC(CC(C)C)C	-37.29	-37.41	0.12	0.31
CCC(C(C)C)C	-35.57	-35.87	0.30	0.84
CCCCCCC	-34.69	-34.39	-0.30	0.87
CC(C(C)C)C(C)C	-39.46	-39.68	0.22	0.55
CC(CC(C)(C)C)C	-41.04	-41.54	0.50	1.22
CC(CC(C)(C)C)(C)C	-43.80	-44.79	0.99	2.27
<i>Alkyls</i>				
C[CH2]	31.30	31.41	-0.11	0.36
CC[CH2]	28.14	28.84	-0.70	2.50
C[CH]C	24.99	25.60	-0.61	2.43
CCC	17.88	19.55	-1.67	9.34
CC([CH2])C	23.23	25.00	-1.77	7.61
CCC[CH2]	24.46	24.67	-0.21	0.84
C[CH]CC	21.59	21.91	-0.32	1.46
[CH2]CC(C)C	19.15	19.24	-0.09	0.47

CC([CH2])(C)C	17.37	20.72	-3.35	19.31
CCCC	14.90	15.05	-0.15	1.03
CC[CH]CC	18.19	18.58	-0.39	2.15
CCC([CH2])C	20.01	19.92	0.09	0.46
CCCC[CH2]	20.80	21.04	-0.24	1.17
C[CH]C(C)C	16.94	16.98	-0.04	0.26
C[CH]CCC	17.91	18.81	-0.90	5.01
[CH2]CC(C)(C)C	13.24	13.07	0.17	1.29
CC[CH]CCC	14.45	15.66	-1.21	8.34
CCC([CH2])(C)C	14.43	14.35	0.08	0.55
C[C](C(C)C)C	10.53	10.41	0.12	1.15
C[CH]C(C)(C)C	11.34	11.35	-0.01	0.09
C[CH]CCCC	14.50	15.06	-0.56	3.86
CC(C(C)C)[CH2]	15.46	15.15	0.31	1.99
CCCCC[CH2]	17.39	17.37	0.02	0.13
[CH2]CC(C(C)C)C	11.96	11.69	0.27	2.29
CC([CH]C(C)C)C	8.91	8.52	0.39	4.40
CC[CH]CCCC	11.54	11.99	-0.45	3.86
CCC(C(C)C)[CH2]	12.02	11.78	0.24	1.99
CCC[CH]CCC	11.52	11.91	-0.39	3.35
C[C](CC(C)C)C	5.57	5.39	0.18	3.19
C[CH]C(C(C)C)C	9.03	8.76	0.27	3.02
C[CH]CCCCC	11.10	11.36	-0.26	2.37
CC(CC(C)C)[CH2]	11.23	11.10	0.13	1.16
CC[C](C(C)C)C	7.58	7.25	0.33	4.37
CCC(CC)C	6.72	6.32	0.40	5.89
CCC(C(C)[CH2])C	12.43	12.17	0.26	2.07
CCCCCC[CH2]	13.66	13.91	-0.25	1.80
[CH2]C(C(C)C)C(C)C	7.66	7.33	0.33	4.33
C[C](C(C)C)C(C)C	3.09	2.26	0.83	26.95
CC(CC)C(C)C	2.22	1.77	0.45	20.47
CC([CH]C(C)(C)C)C	3.00	2.64	0.36	12.11
CC(C([CH2])C)C(C)C	8.33	8.26	0.07	0.90
CC(CC([CH2])(C)C)C	6.87	6.29	0.58	8.49
CC(CC(C)(C)C)[CH2]	6.81	6.01	0.80	11.70

Alkylhydroperoxides

CCOO	-34.16	-33.92	-0.24	0.70
CCCOO	-37.63	-37.41	-0.22	0.59
OOC(C)C	-41.44	-41.43	-0.01	0.02
CCCCOO	-41.23	-41.03	-0.20	0.49
CC(OO)CC	-44.86	-44.73	-0.13	0.28
OOC(C)(C)C	-49.16	-49.44	0.28	0.57
OOC(C)C	-43.12	-43.01	-0.11	0.25

CC(CC)COO	-45.90	-46.13	0.23	0.50
CCCCCOO	-44.67	-44.63	-0.04	0.09
CC(OO)CCC	-48.19	-48.39	0.20	0.41
CCC(OO)(C)C	-52.40	-52.34	-0.06	0.12
CCC(OO)CC	-48.21	-47.97	-0.24	0.49
OOC(C(C)C)C	-49.16	-49.48	0.32	0.65
OCCC(C)(C)C	-48.91	-48.97	0.06	0.12
OCCCC(C)C	-46.01	-46.15	0.14	0.30
CCC(COO)(C)C	-51.67	-51.79	0.12	0.23
OOC(C(C)(C)C)C	-54.55	-54.79	0.24	0.44
OOC(C(C)C)(C)C	-56.17	-56.29	0.12	0.22
OCCC(C(C)C)C	-49.99	-50.45	0.46	0.93
OCCCC(C)(C)C	-51.26	-51.90	0.64	1.24
OOC(C(C)C)C(C)C	-55.84	-55.92	0.08	0.15
OOC(CC(C)C)(C)C	-60.10	-60.43	0.33	0.55
OOC(C(C)(C)C)C(C)C	-60.16	-60.27	0.10	0.17
OOC(CC(C)(C)C)(C)C	-62.90	-62.76	-0.14	0.22
OCCC(CC(C)(C)C)C	-59.75	-60.61	0.85	1.43
OCCC(CC(C)C)(C)C	-58.89	-59.08	0.19	0.33

Alkylperoxides

CCO[O]	-1.76	-1.54	-0.22	12.40
[O]OC(C)C	-9.55	-9.42	-0.13	1.33
CCCO[O]	-5.28	-4.98	-0.30	5.64
CCCCO[O]	-8.95	-8.63	-0.32	3.54
[O]OC(C)(C)C	-17.84	-17.96	0.12	0.69
[O]OCC(C)C	-10.73	-10.53	-0.20	1.90
CC(O[O])CC	-13.10	-12.86	-0.24	1.80
CC(CC)CO[O]	-13.82	-13.62	-0.20	1.44
CCCCCO[O]	-12.67	-12.33	-0.34	2.67
[O]OC(C(C)C)C	-17.52	-17.55	0.03	0.17
[O]OCC(C)(C)C	-16.54	-16.53	-0.01	0.09
[O]OCCCC(C)C	-13.92	-13.76	-0.16	1.14
CC(O[O])CCC	-16.80	-16.54	-0.26	1.54
CCC(O[O])(C)C	-20.90	-20.84	-0.06	0.29
CCC(O[O])CC	-16.50	-16.08	-0.42	2.52
[O]OC(C(C)(C)C)C	-23.04	-23.18	0.14	0.62
[O]OC(C(C)C)(C)C	-24.88	-24.94	0.06	0.25
[O]OCC(C(C)C)C	-17.92	-18.05	0.13	0.73
[O]OCCCC(C)(C)C	-19.06	-19.62	0.56	2.93
CCC(CO[O])(C)C	-19.21	-19.29	0.08	0.40
[O]OC(C(C)C)C(C)C	-25.43	-25.32	-0.11	0.43
[O]OC(CC(C)C)(C)C	-28.75	-35.92	7.17	24.95
[O]OCC(CC(C)C)C	-22.79	-22.84	0.05	0.20

[O]OC(C(C)(C)C)C(C)C	-28.36	-28.51	0.15	0.53
[O]OC(CC(C)(C)C)C(C)C	-31.92	-39.49	7.57	23.71
[O]OCC(CC(C)(C)C)C	-27.26	-27.63	0.37	1.35
[O]OCC(CC(C)C)C(C)C	-26.28	-26.44	0.16	0.62

Hydroperoxy-alkyls

[CH2]COO	15.59	15.42	0.17	1.09
[CH2]CCOO	10.69	10.66	0.03	0.26
OOC([CH2])C	7.96	7.98	-0.02	0.27
C[CH]COO	8.58	8.80	-0.22	2.52
[CH2]CCCCOO	6.89	7.37	-0.48	6.93
[CH2]C(OO)CC	4.47	4.66	-0.19	4.15
C[CH]CCOO	4.09	4.62	-0.53	12.99
CC[CH]COO	5.23	5.75	-0.52	9.91
CC(OO)[CH]C	0.93	1.06	-0.13	13.77
CC(OO)C[CH2]	3.11	3.37	-0.26	8.35
OOC(CC)C(C)C	-15.64	-16.10	0.46	2.92
OOC([CH]C(C)C)C(C)C	-14.84	-14.62	-0.22	1.51
OOC(C([CH2])C)C(C)C	-8.32	-8.88	0.56	6.68
OOC(CC(C)C)([CH2])C	-10.52	-10.50	-0.02	0.18
OOC[C](CC(C)C)C	-11.6	-12.47	0.87	7.49
OCCC([CH]C(C)C)C	-9.52	-9.55	0.03	0.29
OCCC(CCC)C	-12.5	-12.85	0.35	2.80
OCCC(CC(C)C)[CH2]	-6.57	-6.93	0.36	5.50
OOC([CH]C(C)C)C(C)C	-18.2	-18.13	-0.07	0.38
OOC(C([CH2])C)C(C)C(C)C	-11.45	-11.70	0.25	2.15
OOC(C(C)C)CCC	-20.66	-21.42	0.76	3.68
OOC(C(C)C)C(C)C[CH2]	-11.47	-13.34	1.87	16.30
OOC(CC(C)C)C([CH2])C	-14.36	-14.78	0.42	2.93
OOC[C](CC(C)C)C(C)C	-17.3	-18.38	1.08	6.23
OCCC([CH]C(C)C)C(C)C	-15.11	-15.79	0.68	4.52
OCCC([CH]C(C)C)C(C)C	-14.45	-15.26	0.81	5.59
OCCC(CCC)C(C)C	-18.13	-18.51	0.38	2.11
OCCC(CC([CH2])C)C(C)C	-11.54	-11.62	0.08	0.70
OCCC(CC([CH2])C)C(C)C	-11.09	-11.62	0.53	4.82
OCCC(CC(C)C)C[CH2]	-11.69	-12.70	1.01	8.64

C. Isoprene and 1,3-butadiene-2-ol NASA polynomials

Isoprene NASA polynomials.

InChI: InChI=1S/C5H8/c1-4-5(2)3/h4H,1-2H2,3H3 **SMILES:** C=CC(=C)C

RRHO

C4H6O				G	250.000	3000.000	1630.000	1
8.89393988e+00	2.18700908e-02	-9.61134989e-06	2.06023509e-09	-1.75167298e-13				2
-1.28237898e+04	-2.22759665e+01	-1.13354071e+00	5.20728947e-02	-4.44848859e-05				3
2.01196502e-08	-3.68104927e-12	-1.00619854e+04	2.91072901e+01	-1.45087553e+05				4

RRHO+1DHR

C4H6O				G	250.000	3000.000	1630.000	1
1.34777480e+01	1.78914600e-02	-7.70922500e-06	1.63136812e-09	-1.37242132e-13				2
-1.41218543e+04	-4.48693229e+01	-2.77805988e+00	7.32671385e-02	-7.82300425e-05				3
4.10271099e-08	-8.24413927e-12	-1.02547168e+04	3.62259923e+01	-1.45087553e+05				4

Green et al. [81]

C4H6O	C	4H	6O	1	G	100.000	5000.000	917.00	1
1.67447426E+01	6.49283171E-03	-1.29078943E-07	-8.27008753E-11	4.09137317E-15					2
-1.84768160E+04	-6.33495634E+01	1.62503270E+00	3.98215745E-02	-1.28140528E-06					3
-3.80424422E-08	2.09303043E-11	1.43321992E+04	1.57632916E+01						4

1,3-butadiene-2-ol NASA polynomials.

InChI: InChI=1S/C4H6O/c1-3-4(2)5/h3,5H,1-2H2 **SMILES:** C=CC(=C)O

RRHO

C5H8				G	300.000	3000.000	1660.000	1
7.83085338e+00	2.93901619e-02	-1.30636012e-05	2.81928448e-09	-2.40614089e-13				2
4.54131986e+03	-1.78765096e+01	-1.88066221e+00	5.57647515e-02	-4.08385542e-05				3
1.61219915e-08	-2.66084705e-12	7.52221946e+03	3.29300123e+01	-1.22550360e+05				4

RRHO+1DHR

C5H8				G	250.000	3000.000	1630.000	1
1.16177281e+01	2.62596289e-02	-1.15691725e-05	2.48806136e-09	-2.11953159e-13				2
3.62321811e+03	-3.45415119e+01	-1.80030161e+00	6.93315758e-02	-6.40556982e-05				3
3.08708072e-08	-5.91217060e-12	7.07276738e+03	3.33132138e+01	-1.22550360e+05				4

Green et al. [81]

C5H8	C	5H	8	G	100.000	5000.000	967.48	1
1.20176009E+01	2.03483441E-02	-7.02494884E-06	1.25892728E-09	-8.94202102E-14				2
3.69935053E+03	-3.85851566E+01	1.83928733E+00	3.69282576E-02	6.80788292E-06				3
-3.55181225E-08	1.64541534E+11	6.86231883E+03	1.63509440E+01					4

List of Figures

Figure 1.1: EStokTP program structure.....	21
Figure 1.2: Flowchart of RMG rate-based algorithm.....	22
Figure 1.3: Flowchart of RMG group additivity-based algorithm.....	23
Figure 1.4: Flowchart of AutoMech.....	24
Figure 1.5: Flowchart of PACT software package.....	25
Figure 1.6: Flowchart of KinBot.....	26
Figure 1.7: Flowchart of Genesys thermochemistry algorithm.....	27
Figure 1.8: Arkane Isodesmic scheme for benzene.....	28
Figure 2.1: Flowchart of InChI2data algorithm.....	44
Figure 2.2: Pentane structure.....	47
Figure 2.3: Flowchart of FragsGen algorithm.....	48
Figure 2.4: Potential energy surface.....	50
Figure 2.5: 2-hydroperoxybutyl structure.....	53
Figure 2.6: Flowchart of CBH-0 algorithm.....	55
Figure 2.7: 2-hydroperoxybutyl CBH-1 example.....	56
Figure 2.8: Flowchart of CBH-1 algorithm.....	59
Figure 2.9: isopropyl-radical and tert-butyl-radical structure.....	61
Figure 2.10: 2-hydroperoxybutyl CBH-2 example.....	61
Figure 2.11: Flowchart of CBH-2 algorithm.....	63
Figure 2.12: Flowchart of regression algorithm.....	68

Figure 3.1: 1,3-butadiene-2-ol structure	71
Figure 3.2: 1,3-butadiene-2-ol fragmentation products	71
Figure 3.3: Relative percentage error alkanes	74
Figure 3.4: Relative percentage error alkyls	74
Figure 3.5: Relative percentage error alkylhydroperoxides.....	75
Figure 3.6: Relative percentage error alkylperoxides.....	75
Figure 3.7: Relative percentage error hydroperoxy-alkyls.....	76
Figure 3.8: Relative percentage error of the entire set.....	77
Figure 3.9: Absolute error of $\Delta H^0(0 K)$ estimation at different levels of theory	80
Figure 3.10: Absolute error of thermal correction of $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$	82
Figure 3.11: Isoprene $C_p^0(T)$	83
Figure 3.12: Isoprene $S^0(T)$	83
Figure 3.13: Isoprene $H^0(T)$	83
Figure 3.14: Isoprene $G^0(T)$	84
Figure 3.15: 1,3-butadiene-2-ol $C_p^0(T)$	84
Figure 3.16: 1,3-butadiene-2-ol $S^0(T)$	84
Figure 3.17: 1,3-butadiene-2-ol $H^0(T)$	85
Figure 3.18: 1,3-butadiene-2-ol $G^0(T)$	85

List of Tables

Table 2.1: Thermal correction to enthalpy of the atomic elements	65
Table 3.1: 1,3-butadiene-2-ol bond energies at the ω B97X-D/jun-cc-pVTZ level of theory (Level 1) and at the CCSD(T) level of theory with extrapolation to basis set limit and correction for core electrons correlation (High level)	72
Table 3.2: Absolute and relative standard deviation of $\Delta H^0(0 K)$ estimation.....	76
Table 3.3: Sum of electronic and ZPE at different levels of theory	78
Table 3.4: Absolute error of the $\Delta H^0(0 K)$ estimation at different levels of theory	78
Table 3.5: Standard deviation of absolute error of $\Delta H^0(0 K)$ estimation at different levels of theory.....	80
Table 3.6: Absolute error of thermal correction from $\Delta H^0(0 K)$ to $\Delta H^0(298.15 K)$	81

List of Symbols

Variable	Description	SI unit
C_i	Concentration species i	[mol m ⁻³]
α, β	Fitting coefficients	[-]
K_0	Pre-exponential factor	[m ³ mol ⁻¹ s ⁻¹]
E_A	Activation energy	[kcal mol ⁻¹]
R	Ideal gas constant	[kcal mol ⁻¹ K ⁻¹]
r	Reaction rate	[mol m ⁻³ s ⁻¹]
T	Temperature	[K]
C_p^0	Standard specific heat @ 1 bar	[cal mol ⁻¹ K ⁻¹]
S^0	Standard entropy @ 1 bar	[cal mol ⁻¹ K ⁻¹]
H^0	Standard enthalpy @ 1 bar	[kcal mol ⁻¹]
G^0	Standard Gibbs free energy @ 1 bar	[kcal mol ⁻¹]
$\Delta H^0(0\text{ K})$	Standard enthalpy of formation @ 0 K, 1 bar	[kcal mol ⁻¹]
$\Delta H^0(298.15\text{ K})$	Standard enthalpy of formation @ 298.15 K, 1 bar	[kcal mol ⁻¹]
$a_0 - a_6$	Nasa polynomials coefficients	[S.I.]
E_{bond}	Bond energy	[kcal mol ⁻¹]
E_{el}	Electronic energy	[kcal mol ⁻¹]
ZPE	Zero-point energy	[kcal mol ⁻¹]

$\Delta H_R^0(0\text{ K})$	Standard enthalpy of reaction @ 0 K, 1 bar	[kcal mol ⁻¹]
T_{split}	Split range temperature	[K]
N_{H_2}	Number of H ₂ molecules (CBH-0)	[-]
$N_{i,S}$	Number of stable species of type i	[-]
$N_{i,R}$	Number of radical species of type i	[-]
$N_{H,PM}$	Number of hydrogen atoms in the parent molecule	[-]
σ_i	Saturation number atom i	[-]
N_H	Number of hydrogen atoms in SMILES group (CBH-1, CBH-2)	[-]
σ_{HA}	Saturation number non-hydrogen atom	[-]
σ_B	Number of shared electrons by a non-hydrogen atom to form a bond	[-]
ΔN_i	Correction factor for CBH-1 reactants	[-]
GS	Group SMILES (for single non-hydrogen atom)	[-]
B _i	Bond i symbol	[-]
H_{PM}^{corr}	Standard thermal correction to enthalpy of the parent molecule @ 1 bar	[kcal mol ⁻¹]
H_i^{corr}	Standard thermal correction to enthalpy of the type i atom @ 1 bar	[kcal mol ⁻¹]
2σ	Standard deviation	[-]
Y_i	i value in standard deviation summatory	[S.I.]
\bar{Y}	Mean value	[S.I.]

Acknowledgements

Acknowledgements

The CINECA award HP10BCGTXF, under the ISCRA initiative, is acknowledged for the availability of high-performance computing resources and support.

Grazie al mio relatore, Professor Carlo Cavallotti, per il continuo supporto che ha permesso di sviluppare questa tesi e per avermi permesso di dare un contributo diretto al miglioramento di EStokTP.

Grazie al mio correlatore, Andrea Della Libera, per l'immenso aiuto datomi in questi mesi; il suo apporto è stato fondamentale per la realizzazione di questo lavoro.

Grazie al Dottor Alberto Baggioli per i preziosi consigli sullo sviluppo di molti punti del codice. A lui un ringraziamento non di circostanza.

Grazie a Rebecca, che in questi due anni ha vissuto con me gioie e dolori della laurea magistrale, e che non mi ha mai fatto mancare il supporto che solo una vera amica può dare. Spero di essere stato altrettanto per lei.

Grazie a Francesca, che è una splendida amica su cui sono sicuro di poter sempre fare affidamento, e lo stesso vale per lei.

Grazie a Giuseppe, che non mi ha mai fatto mancare momenti di spensieratezza e per il costante incoraggiamento nel corso degli anni.

Grazie a Elisa, con cui ho condiviso tanto dal punto di vista universitario e non. Il progetto di CFD rimane una delle mie più grandi soddisfazioni e sono contento di averlo condiviso con lei.

Grazie a Matteo, Maristella, Federico, Ilaria, Alice e Andrea per i mesi passati in centro di calcolo e per aver reso meno pesante la scrittura di questa tesi.

Grazie a Riccardo, Jacopo, Mario, Laura e Gaia per aver condiviso con me un viaggio ricco di emozioni.

Grazie a Giorgio e Tommaso, che sono per me una seconda famiglia.

Grazie a Davide, Elena e Leonardo, che sono stati un supporto costante per tutta la durata dei miei studi.

Grazie a Gabriele per aver creduto in me e per tutto l'aiuto che mi ha dato.

Infine, grazie ai miei genitori, la mia famiglia, per tutto ciò che hanno fatto per me.

