EStokTP

From Electronic Structure to Temperature and Pressure Dependent Rate Constants

Carlo Cavallotti, Matteo Pelucchi

Dipartimento di Chimica, Materiali e Ingegneria Chimica "Giulio Natta", Politecnico di Milano, 20131 Milano, Italy

Stephen J. Klippenstein

Chemical Sciences and Engineering Division, Argonne National Laboratory, Argonne, IL, 60439, USA

Table of Contents

1. Overview of EStokTP

1.1. Purpose

In recent years, theoretical chemical kinetics has transformed from an empirical to a predictive science, with achievable accuracies approaching or even exceeding that of experiments. This transformation arises from the dramatic progress in the accuracy of electronic structure calculations, as well as improved procedures for incorporating such predictions within treatments of chemical dynamics and kinetics. The *ab initio* transition state theory based master equation (AITSTME) approach, which provides the basis for much of the work in this area, is finding widespread utility as a tool for chemical kinetic modeling. This utility could be greatly enhanced by reducing the human effort involved in the implementation of such calculations.

Typically, such calculations are performed in three distinct steps: First the molecular properties of the stationary points on the potential energy surface are predicted with *ab initio* (AI) electronic structure codes. Microcanonical rates for chemical transfer are then predicted with transition state theory (TST) codes that incorporate these stationary point properties. Finally, thermal rate estimates are obtained with master equation (ME) solver codes, which rely on the microcanonical and energy transfer rates. While, the transition state theory and master equation calculations are often performed within the same code, the electronic structure calculations are almost always performed independently. This decoupling of the electronic structure evaluations leads to considerable human effort wasted on the transfer of data from one set of codes to the other.

The EStokTP computational environment is designed to directly couple the electronic structure, TST, and ME evaluations in automatically obtaining *a priori* predictions of the temperature and pressure dependent rate constants. The code requires as input the reference molecular structures, the transition states to be searched, and the electronic structure and transition state theory methodologies to be employed. Given this set of input files, the code performs sequentially the series of calculations listed in Figure 1.
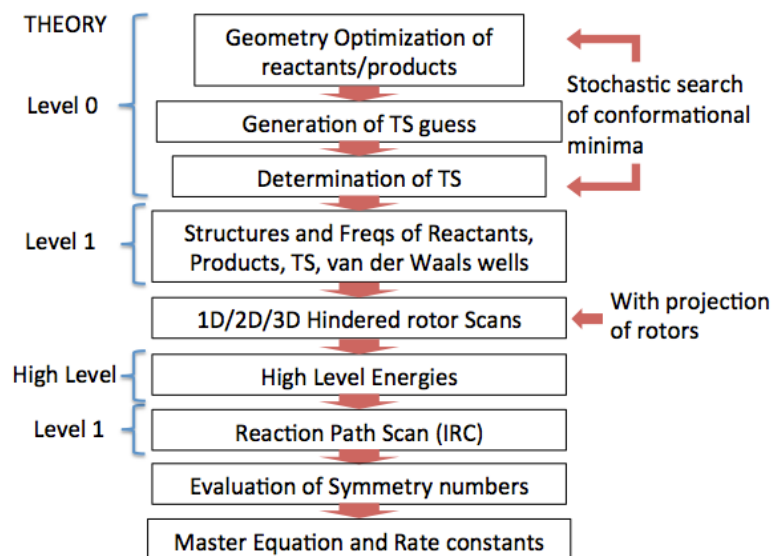

Figure 1: EStokTP program structure.

EStokTP relies on calls to external codes to perform electronic structure and master equation calculations. Internal torsions can be treated as hindered rotors, in which case they are projected out from the Hessian so that no identification and removal from the frequency list is necessary. Tunneling can be accounted for using both Eckart and multidimensional models, such as small curvature theory. The code has been written following two principles. The first is that it must be able to reliably and automatically determine a large number of rate constants for classes of reactions of a target molecule (e.g. H-abstractions). This requirement is motivated by the perspective of the upcoming availability of exascale computational resources. The code is thus robust, with several automatic fallback options implemented so that human intervention can be limited. The second is that it must be capable of determining highly accurate rate constants, differing by no more than a factor of two from experiments. At present the following reaction types are implemented: abstraction, addition, isomerization, and beta-decomposition.

EStokTP can be used also to investigate complicated potential energy surfaces. For this purpose all reaction steps are studied separately applying repeatedly EStokTP. There is no need of repeating calculations for reactants or products more than once as scripts allow copying all reactants and products data necessary for a calculation from one directory to the other. After all the desired reaction steps have been performed, a multiple well master equation input can be written using a wrapper script.

1.2. Capabilities

1.2.1. Current
- determination of minimum energy structures for wells and saddle points performing Monte Carlo conformational analysis over specified list of internal coordinates;
- 1D, 2D or 3D rotational scan over list of dihedral coordinates;
- project out torsional motions from Hessian (implies no need to identify torsional vibrational frequencies manually);
- possibility to project out of the Hessian the motion along the reaction path
- automatic multilevel determination of structures (level0 followed by level1 calculations);
- search for high quality guesses for abstraction, addition, beta-scission and isomerization reactions;
- determination of internal and external rotational symmetry numbers, including optical symmetry numbers;
- search for structures of van der Waals wells on reactant and product side;
- IRC calculations with PES evaluation and projection from the Hessian of hindered rotors along the IRC path at selected points with interpolation for intermediate points; rescaling of energy at a high level quality along the IRC path is also supported;
- Small Curvature Tunneling calculations, implemented using reaction path Hamiltonian frequencies;
- preparation of input for the Mess master equation solver, which is at present used to compute thermodynamic data, multi dimensional partition functions for hindered rotors, and pressure dependent rate constants;
- two electronic structure calculation codes are currently supported: G09 and molpro (2008-2015 are supported). All calculation steps can be done with any of the two

codes made exception for the internal reaction coordinate calculations, which are supported only for G09.

1.2.2. Planned
- automatic determination of rate constants for barrierless reactions
- anharmonic analysis

1.3. EStokTP Jobs: theory and sequential module calls

EStokTP is made of a series of modules that can be called separately. Each module requires a set of data, which are reported in the **./data** subdirectory for the initial calculations (e.g. level 0 geometry optimizations), while geometries and Hessian are stored in the **./output** and **./geoms** subdirectories for later stages of the calculations (such as for saddle point determination or high level calculations).

EStokTP modules can be used to perform five types of calculations (jobs): well, abstraction, addition, beta-scission, or isomerization. For each job we suggest a set of modules to call, as listed below. Changes to the proposed call list are possible, though in order to obtain the production of the **master equation blocks** and, for the reactions, of the **master equation input** for the Mess solver the level0, level1 and high-level calculations must always be performed (see section 1.4 for a description of what is stored in a block or master equation input file). The hindered rotor analysis, IRC, and symmetry calculations can be omitted, if thought unnecessary.

1.3.1. Well

1) Opt_Reac1
2) Opt_Reac1_1
3) 1dTau_Reac1
4) mdtau_reac1
5) HL_Reac1
6) Symm_reac1
7) kTP

In step 1, the minimum energy structure for reactant 1 is searched at level 0 of theory. In step 2 geometry and Hessian are determined at level 1. In step 3 1D hindered rotor scans are performed. In step 4 multidimensional hindered rotor scans are performed. If 1d and multi dimensional HR scans are not requested, step 3 and 4 can be omitted. In step 5 energies are determined at the specified high level of theory on level1 geometries. In step 6 symmetry numbers are determined. In step 7 the output are written in the Mess block format (briefly, me block), which can be used to generate the input necessary to compute thermodynamic parameters or for multi well master question calculations. The calculation output, which comprises level 1 geometry, level 1 frequencies from which hindered rotors have been projected, list of 1d hindered rotors, with rotating groups, axis and rotational pes, and symmetry is summarized in one file that is written by the kTP module in the **./me_block** subdirectory in the reac1.me file. The high level energy expressed in Hartrees is stored in the same subdirectory in the reac_en.me file. If multi dimensional hindered rotors calculations have been requested, the PES files are copied as well in

the **./me_block** subdirectory and are named reac1_2dhr_01.dat and reac1_2dhr_nofr01.dat, with the first file containing the PES and the frequencies calculated and projected along the hindered rotor path and the second one only the PES.

1.3.2. Abstraction

Rate constant calculations for abstraction reactions can be performed using three different PESs, which differ depending on whether van der Waals wells are considered both on the reactant and product sides (3TS calculation), only on the reactant side (2TS calculation) or not considered (1TS calculations). The approach is chosen with the 'reactiontype' keyword (see section 3.1.1). A scheme of the 3TS PES is sketched in Figure 2.
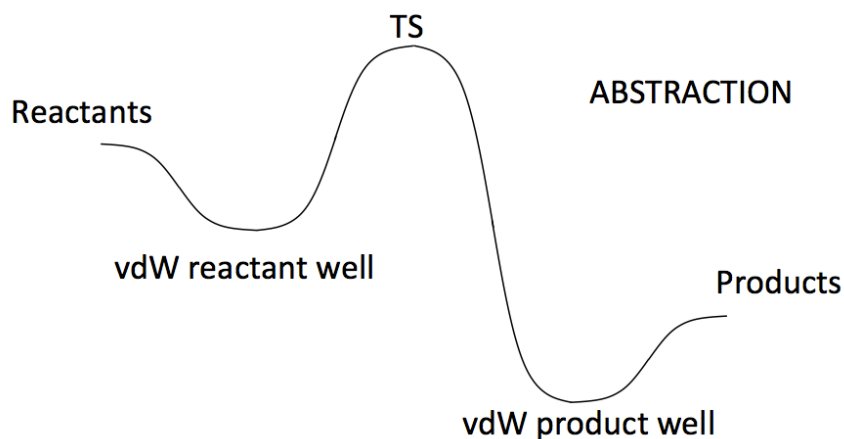

Figure 2: PES used to calculated the abstraction reaction using a three transition states (3TS) model.

In the 3TS model, the abstraction rate constant is computed solving a master equation consisting of three channels: entrance, exit and reaction. The rates of the entrance and exit channels are computed using phase space theory, while that of the reaction channel is determined using transition state theory. Phase space theory calculations are set up and performed automatically by the Mess solver, which requires only information on the wells it connects. In this case they are the bimolecular reactants and the vdW reactant well for the entrance channel and the vdW product well and the bimolecular products for the exit channel. Usually the rates of entrance and exit channels largely exceed those of the reaction channel, so that the latter becomes the controlling step. However at sufficiently low temperatures the van der Waals well may live long enough time to get collisionally stabilized, in which case they would appear as a possible product channel in the master equation results. The depth of the van der Waals well may have a significant impact, especially at low temperatures and when the reaction barrier height is small with respect to reactants, on the quantum tunneling contribution to the rate constant. It is thus advised to compute the wells energy and structure at the same level of theory as that of reactants and products, though it is it is not necessary, unless the rate of formation of the van der Waals well is desired, to perform a hindered rotor study of the van der Waals wells. In case of need, the hindered rotor study of the van der Waals wells should be carefully checked as it may easily lead to the finding of other van der Waals wells, which may not be those connected to the reaction transition state. The 2TS and 1TS abstraction PESs are similar to that reported in Figure 2, from which they differ for the absence of the van der Waals product well and for the absence of both van der Waals wells, respectively.

The search for the transition state for the reaction channel is performed through a set of restrained geometry optimizations where the energy of the structure consisting of the two reactants is minimized while keeping fixed a coordinate, named RTS, which describes the approach of reactant 2 to the reaction site (isite) of reactant 1. In particular the z-matrix containing the two reactants is automatically constructed using as input the z-matrixes specified for reactant 1 and reactant 2 in the reac1.dat and reac2.dat files. The relative position of reactant 2 with respect to reactant 1 is defined through the definition of 6 degrees of freedom (3 if reactant 2 is an atom, 5 if it is a biatomic species). Since the abstracting atom is often co-linear with the bond of the abstracted atom, a dummy atom is positioned at 2.0 Å from the abstraction site (same as abstracted atom) and perpendicularly to the bond that is being broken in reactant 1 through the abstraction reaction. To construct the transition state z-matrix the following information is needed: a list of three atoms of reactant 1 used to define the relative position of reactant 2 and 5 angle coordinates: aabs1, babs1, aabs2, babs2, and babs3. The three atoms, defined as isite, jsite and ksite are the abstraction site (isite), the atom to which the abstracted atom is bound (jsite), and an atom, usually connected to jsite, which is used to define the dihedral angle for the dummy atom. The dihedral angle is placed at 2.0 Å from isite, at 90 degrees with respect to isite and jsite and on a dummy atom-isite-jsite plane that forms a 180 angle with the isite-jsite-ksite plane. The first atom of reactant 2 is assumed to be the extracting atom (that is the atom that will bind the extracted atom).

The definition of the 5 angle coordinates (transitional degrees of freedom) is the following:

aabs1: angle between first atom of reactant2, isite and dummy atom (usually 90°).

babs1: dihedral angle between first atom of reactant2, isite, dummy atom, and jsite (usually 180°).

aabs2: angle between second atom of reactant 2, fisrt atom of reactant2, and isite. It depends on the reactant, often a good choice is 90°, if it is about 180° then it may give problems in the optimization.

babs2: dihedral angle between second atom of reactant 2, first atom of reactant2, isite, and the dummy atom. Its value depends on the system. It is likely that it is a torsional degree of freedom, and should thus be considered as a hindered rotor and randomly scanned in the tauo_ts module.

babs3: dihedral angle between third atom of reactant 2, the two atoms to which it is connected in the reactant2 z-matrix structure, and isite. Its value depends on the system. It is likely that it is a torsional degree of freedom, and should thus be considered as a hindered rotor and randomly scanned in the tauo_ts module.

An example of a possible definition of isite, jsite, ksite, dummy atom, scanned coordinate as well as of aabs1 and aabs2 is shown in Figure 3. Once the transition state z-matrix has been constructed, the RTS distance is scanned in the interval and for the number of points specified in the ts.dat input file (see section 3.2.3 for further details).
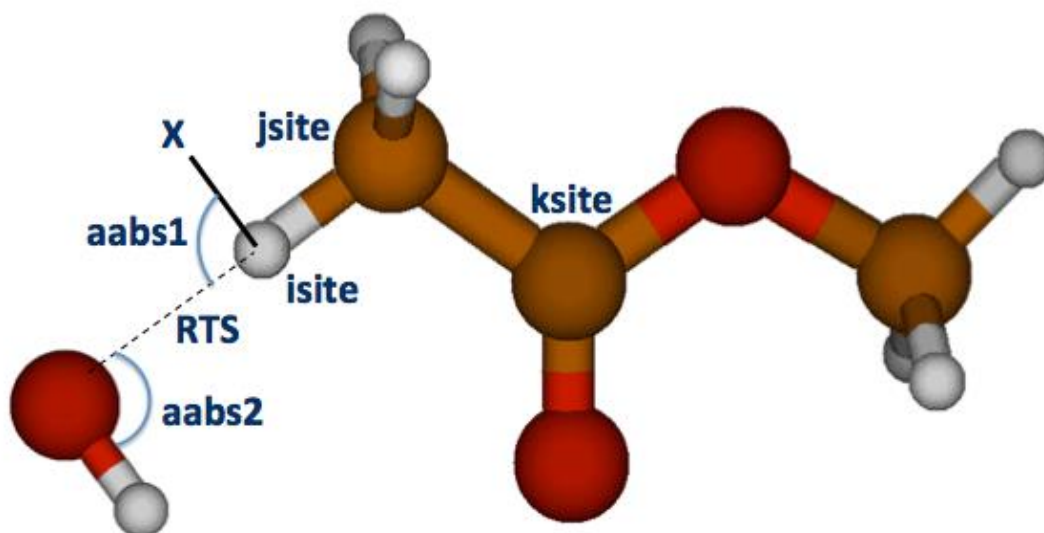
Figure 3: Transition state geometry constructed automatically for H abstraction in the OH + methylactetate reaction. The isite, jsite, and ksite atoms specified for the calculation and the resulting position of the dummy atom (X) are explicitly shown, together with some of the geometrical variables used in the calculations (the grid scanned distance RTS, and two of the angles defining the position of the OH residue relatively to methyl acetate.

For a 3 TS abstraction reaction calculation the sequential list of modules that should be called is the following:

First the geometries of reactants and products is determined at level 0 of theory.
　Opt_Reac1
　Opt_Reac2
　Opt_Prod1
　Opt_Prod2

Then the RTS coordinate is scanned in the interval and for the number of points requested in the ts.dat file.
　Grid_Opt_TS

The structure with the highest energy is used as guess for a saddle point search, which is performed at level 0 of theory.
　Opt_TS_0

Once the saddle point has been found, a random search is performed on the dihedral angles specified in the ts.dat input for the requested number of times.
　Tauo_Ts

Once the TS minimum energy structure has been determined, the structure of reactants, products and transition state are determined at level 1 of theory.
　Opt_Reac1_1
　Opt_Reac2_1
　Opt_Prod1_1

```
Opt_Prod2_1
Opt_TS_1
```

If the 'wellr findgeom level1' and 'wellp findgeom level1' keywords have been used in the estoktp.dat file, than the structures of wellr and wellp are searched at level 1 of theory. A minimum input is requested for the well input files (see section 3.2.2).
```
Opt_wellr_1
Opt_wellp_1
```

Hindered rotor scans are then performed for all the considered stationary points and wells.
```
1dTau_Reac1
1dTau_Reac2
1dTau_Prod1
1dTau_Prod2
1dTau_ts
1dTau_wellr
1dTau_wellp
```

High level energies are determined for all the considered stationary points and wells.
```
HL_Reac1
HL_Reac2
HL_Prod1
HL_Prod2
HL_WellR
HL_WellP
HL_Ts
```

Symmetry numbers are determined for all the considered stationary points and wells.
```
Symm_reac1
Symm_reac2
Symm_prod1
Symm_prod2
Symm_wellr
Symm_wellp
Symm_ts
```

An IRC calculation is performed for the number of points and using the algorithm specified in the theory.dat file. It can be used to determine variationally the rate constant if the 'variational' keyword is present in the estoktp.dat file.
```
Irc
```

Having performed all the above calculation, we are now ready to compute the rate constant calling the ktp module.
```
kTP
```

1.3.3. Addition

Rate constant calculations for addition reactions can be performed using two different PESs, which differ depending on whether a van der Waals wells is considered on the reactant side (2TS calculation) or not (1TS calculations). The selection is made with the 'reactiontype' keyword (see section 3.1.1). A scheme of the 2TS PES is sketched in Figure 4.
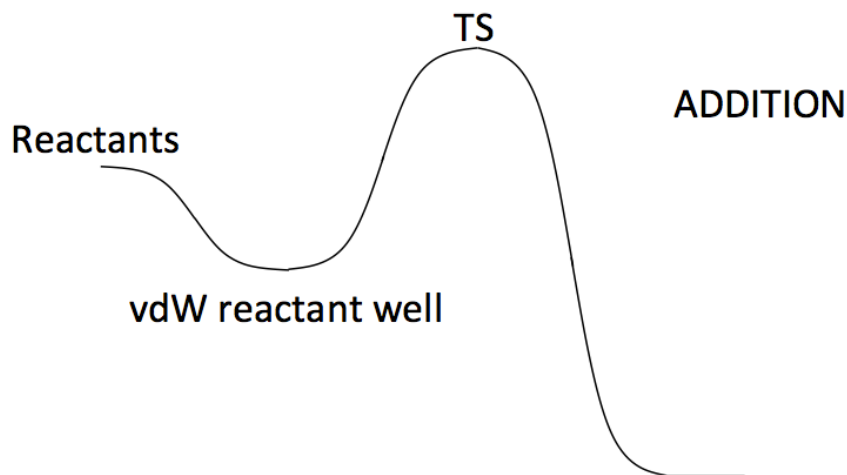
Figure 4: PES used to calculated an addition reaction using a two transition states (2TS) model.

Similarly to what seen for abstraction reactions, also in this case in the 2TS model the abstraction rate constant is computed solving a master equation consisting of two channels: entrance, and reaction. The rate of the entrance channel is computed using phase space theory, while that of the reaction channel is determined using transition state theory. We refer to section 1.3.2 for a discussion of what implies including a van der Waals well in a master equation and to set up a calculation for the well.

The algorithm used to search for the reaction transition state is similar to the one used for abstraction reactions. The main difference is that no dummy atom is used in this case to construct the z-matrix of the transition state. Also in this case, however, a list of three atoms (isite, jsite, and ksite) must be given in order to define the orientation of reactant 2 with respect to reactant1 in the z-matrix. In this case however isite, jsite and ksite are used to define the position of the first atom of reactant 2, which is assumed to be the atom with which a bond between reactant 1 and 2 will be formed. The parameters aabs1, babs1, aabs2, babs2, and babs3 have the following meaning:

aabs1: angle between first atom of reactant2, isite and jsite.
babs1: dihedral angle between first atom of reactant2, isite, jsite, and ksite.
aabs2: angle between second atom of reactant 2, first atom of reactant2, and isite.
babs2: dihedral angle between second atom of reactant 2, first atom of reactant2, isite and jsite. Its value depends on the system. It is likely that it is a torsional degree of freedom, and should thus be considered as a hindered rotor and randomly scanned in the tauo_ts module.
babs3: dihedral angle between third atom of reactant 2, the two atoms to which it is connected in the reactant2 z-matrix structure, and isite. Its value depends on the system. It is likely that

it is a torsional degree of freedom, and should thus be considered as a hindered rotor and randomly scanned in the tauo_ts module.

An example of a possible definition of isite, jsite, ksite, dummy atom, scanned coordinate as well as of aabs1 and aabs2 is shown in Figure 5. Once the transition state z-matrix has been constructed, the RTS distance is scanned in the interval and for the number of points specified in the ts.dat input file (see section 3.2.3 for further details).
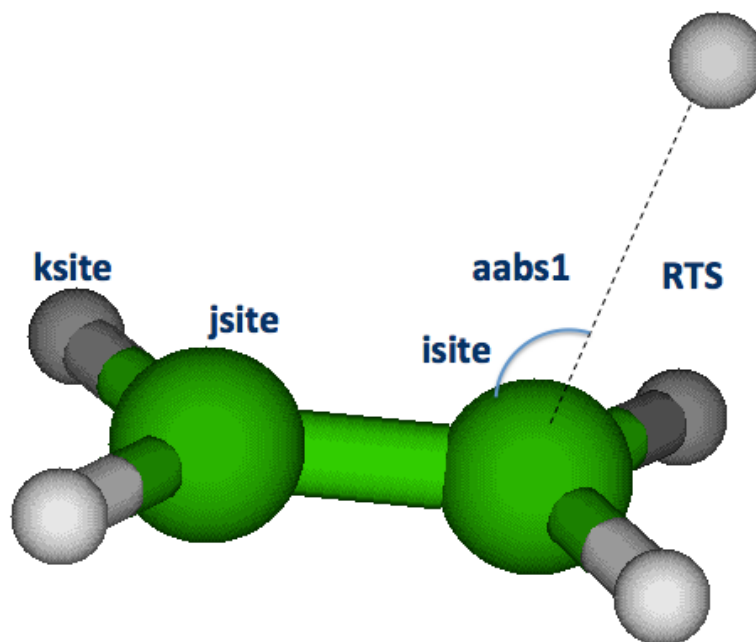
Figure 5: Transition state geometry constructed automatically for H addition to C_2H_4. The isite, jsite, and ksite atoms specified for the calculation are explicitly shown, together with some of the geometrical variables used in the calculations (the grid scanned distance RTS, and two of the angles defining the position of the H atom relatively to ethylene.

For a 2 TS addition reaction calculation the sequential list of modules that should be called is the following:

First the geometries of the two reactants are determined at level 0 of theory. Note that no product needs to be specified here as it will be determined by the code as wellp using the findgeom keyword.
  Opt_Reac1
  Opt_Reac2

Then the RTS coordinate is scanned in the interval and for the number of points requested in the ts.dat file.
  Grid_Opt_TS

The structure with the highest energy is used as guess for a saddle point search, which is performed at level 0 of theory.
  Opt_TS_0

Once the saddle point has been found, a random search is performed on the dihedral angles specified in the ts.dat input for the requested number of times.
  Tauo_Ts

Level 1 structures and vibrational frequencies are determined for all the considered stationary points and wells. The structure of the van der Waals reactant well is determined using the findgeom algorithm. The same algorithm is used to locate the product well, wellp, which in this case will coincide with the reaction product.
  Opt_Reac1_1
  Opt_Reac2_1
  Opt_TS_1
  Opt_wellr_1
  Opt_wellp_1

Hindered rotor scans are then performed for all the considered stationary points and wells.
  1dTau_Reac1
  1dTau_Reac2
  1dTau_ts
  1dTau_wellr
  1dTau_wellp

High level energies are determined for all the considered stationary points and wells.
  HL_Reac1
  HL_Reac2
  HL_WellR
  HL_WellP
  HL_Ts

Symmetry numbers are determined for all the considered stationary points and wells.
  Symm_reac1
  Symm_reac2
  Symm_wellr
  Symm_wellp
  Symm_ts

An IRC calculation is performed for the number of points and using the algorithm specified in the theory.dat file. It can be used to determine variationally the rate constant if the 'variational' keyword is present in the estoktp.dat file.
  Irc

Having performed all the above calculation, we are now ready to compute the rate constant calling the ktp module.
  kTP

1.3.4. BetaScission

Beta-scission reactions are the inverse of addition, so that their rate constant may be determined through detailed balance from the addition rate constant. Alternatively, given a reactant, a rate constant for a beta-scission reaction can be determined using the betascission algorithm. This is a relatively simple bond pulling algorithm that stretches the breaking bond for a given number of steps, specified in the ts.dat input file. To use this algorithm the 'reactiontype' keyword (see section 3.1.1) of the estoktp.dat file must be set to 'betascission'. The rate of the bet-decomposition reaction is determined solving the master equation calling the Mess code, so that pressure dependence is properly accounted for. The parameters of the master equation calculation are set in the me_head.dat file, located in the **./data** subdirectory. No van der Waals well is used for this type of calculation. The breaking bond is determined through the data specified in the ts.dat input file. The number of the first of the two atoms of the breaking bond is identified as isite. The jsite and ksite atom number should be specified as well in the input file but are at present not used in the calculation, so that they can assume any value. The second atom is the last number specified in the ts.dat input identified by the rmax1 keyword, namely ireact. The other numbers are distance1, step1, distance2, and step2. The beta-scission algorithm is a pushing algorithm working so that the distance between two atoms, isite and ireact, is progressively increased from that of the minimum energy structure to distance1 with step step1 and then to distance2 with step step2. See section 3.2.3 for further details and an example.

The sequential list of modules that should be called for a standard beta-scission calculation is the following:

First the geometries of the reactant and two products are determined at level 0 of theory.
 Opt_Reac1
 Opt_Prod1
 Opt_Prod2

Then the breaking bond coordinate is scanned in the intervals requested in the ts.dat file using the specified steps.
 Grid_Opt_TS

The structure with the highest energy is used as guess for a saddle point search, which is performed at level 0 of theory.
  Opt_TS_0

Once the saddle point has been found, a random search is performed on the dihedral angles specified in the ts.dat input for the requested number of times.
  Tauo_Ts

Level 1 structures and vibrational frequencies are determined for all the considered stationary points and wells.
 Opt_Reac1_1
 Opt_Prod1_1
 Opt_Prod2_1
 Opt_TS_1

Hindered rotor scans are then performed for all the considered stationary points and wells.
 1dTau_Reac1

1dTau_Prod1
1dTau_Prod2
1dTau_ts

High level energies are determined for all the considered stationary points and wells.
HL_Reac1
HL_Prod1
HL_Prod2
HL_Ts

Symmetry numbers are determined for all the considered stationary points and wells.
Symm_reac1
Symm_prod1
Symm_prod2
Symm_ts

An IRC calculation is performed for the number of points and using the algorithm specified in the theory.dat file. It can be used to determine variationally the rate constant if the 'variational' keyword is present in the estoktp.dat file.
Irc

Having performed all the above calculation, we are now ready to compute the rate constant calling the ktp module.
kTP

1.3.5. Isomerization

Isomerization reactions are pressure independent unimolecular reactions that involve one reactant and one product. Transition states for isomerizations are searched constructing a product like z-matrix where the bond that is to be formed as a result of the reaction is explicitly expressed as a z-matrix parameter. This is performed by substituting the z-matrix line that contains the reacting atom (ireact) with a line where the forming bond appears explicitly. To use this algorithm the 'reactiontype' keyword (see section 3.1.1) of the estoktp.dat file must be set to 'isomerization'. The breaking bond is determined through the data specified in the ts.dat input file. The number of the first of the two atoms of the breaking bond is the last number specified in the ts.dat input identified by the rmin1 keyword, namely ireact. The other numbers are distance1, step1, distance2, and step2. The second atom is identified as isite. The jsite and ksite atom numbers are used to define the connectivity (angle and dihedral) of the ireact and isite atoms. The isomerization algorithm is a pulling algorithm working so that the distance between two atoms, isite and ireact, is progressively decreased from that of the starting structure to distance1 with step step1 and then to distance2 with step step2. The starting structure is not the minimum energy structure, but the conformer found in the Monte Carlo search for the minimum energy structure of reactant 1 that exhibits the minimum distance between ireact and isite. See section 3.2.3 for further details and an example.
The sequential list of modules that should be called for a standard isomerization calculation is the following:

First the geometry of the reactant and is optimized at level 0 of theory.
 Opt_Reac1

Then the forming bond coordinate is scanned in the specified intervals (distance1 and distance 2) using the step specified in the ts.dat file. An additional internal coordinate can also be simultaneously scanned in a 2D search.
Grid_Opt_TS
2dGrid_Opt_TS

The structure with the highest energy is used as guess for a saddle point search, which is performed at level 0 of theory.
  Opt_TS_0

Once the saddle point has been found, a random search is performed on the dihedral angles specified in the ts.dat input for the requested number of times.
  Tauo_Ts

Level 1 structures and vibrational frequencies are determined for all the considered stationary points and wells. The findgeom wellp algorithm is in this case used to find the product structure.
 Opt_Reac1_1
 Opt_TS_1
 Opt_wellp_1

Hindered rotor scans are then performed for all the considered stationary points and wells.
 1dTau_Reac1
 1dTau_wellp
 1dTau_ts

High level energies are determined for all the considered stationary points and wells.
 HL_Reac1
 HL_wellp
 HL_Ts

Symmetry numbers are determined for all the considered stationary points and wells.
 Symm_reac1
 Symm_wellp
 Symm_ts

Having performed all the above calculations, we are now ready to compute the rate constant calling the ktp module.
   kTP

1.3.6. Barrierless

The reaction rates for barrierless channels may be computed with both variational transition state theory (VTST) and variable reaction coordinate transition state theory (VRC-TST). Contrary to other reaction types, in this case there is no well-defined barrier. The bond-fission is considered in the direction of dissociation, i.e. A➔B+C. The MEP is scanned at fixed distances of the breaking

bond and the orientation of the two fragments is automatically defined from that of the reactant. Hence, the Z-matrix along the MEP is the same as that of the reactant. The ts.dat file must be specified as described in Section 4.1.6. The connectivity of the two fragments is retrieved from the reactant matrix, however the reactive sites must be specified in ts.dat. A possible definition of the sites is shown in Figure 6: ibond, jbond and kbond are the atoms of the first, second and third atoms (if present) of the second fragment as specified in the Z-matrix of the reactant. Isite, jsite and ksite are instead the atoms of the Z-matrix of product 1 defining the connectivity of ibond to the first fragment as specified in the Z-matrix of the reactant. The reaction coordinate RTS is the distance between ibond and isite. Further details are provided in Section 4.1.6.
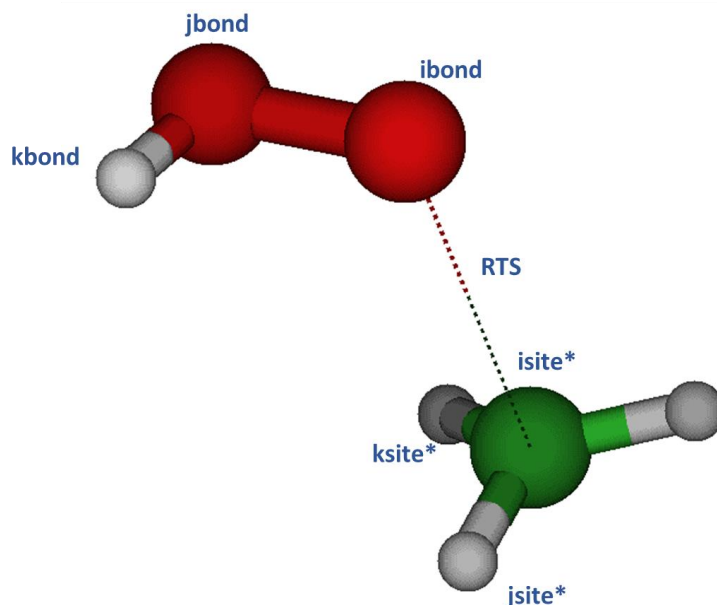
*Figure 6: Transition state geometry constructed automatically for CH3+HO2. The isite, jsite, and ksite ibond, jbond, kbond atoms specified for the calculation are explicitly shown, together with the grid scanned distance RTS. * symbol indicates that the atoms numbers refer to the z-matrix of the first fragment (prod1), whereas all the others refer to the z-matrix of the reactant.*

Several steps are required to perform the calculations for a barrierless reaction:
- STEP0: The information about the reactants and the products is generated in **ref** directory.
- STEP1: a first directory at a fixed representative RTS distance **dist_x.y** is generated. Here, both level1 and high level calculations are performed.
- STEP2: The information of STEP1 is used to generate other directories describing the system at fixed RTS distances along the MEP.
- STEP3: After all calculations are finished, the vtst input is assembled and run in the directory **variational**. At this point, the variational rate constant is available.
- STEP4: To proceed with vrctst calculations, a correction potential is computed at each dist_x.y.
- STEP5: the **vrctst** directory is set up and the input to vrctst calculations is generated. **rotd** program is used for these calculations (see Section 4.1.6.2).
- STEP6: after vrctst calculations are finished, MESS input is setup and run.

A detailed description of each step is provided below. Steps 1-7 may be performed automatically with a **run.sh** script. The only modification required to **run.sh** is the list of distances to be considered to build the MEP.

STEP0: setting up the **ref** directory.

The **ref** directory contains the information of the reactants and products and the required input for all the following steps. The **ref/data** directory contains several files necessary to run the initial step of the calculation. The main operations are performed first with **estoktp.dat_reacsprods** and then with **estoktp.dat_vrc**. These files are renamed as estoktp.dat before running the steps listed in them.

Estoktp.dat_reacsprods: this file contains the keywords necessary for the optimization of the reactant (the well) and products (the two fragments). This sequential operations called by this file are listed below:

First the geometry of the reactants and products are optimized at level 0 of theory.
Opt_Reac1
Opt_Prod1
Opt_Prod2
Level 1 structures and vibrational frequencies are determined for all the considered stationary points.
Opt_Reac1_1
Opt_Prod1_1
Opt_Prod2_1
Hindered rotor scans are then performed for all the considered stationary points.
1dTau_Reac1
1dTau_Prod1
1dTau_Prod2
High level energies are determined for all the considered stationary points.
HL_Reac1
HL_Prod1
HL_Prod2
Symmetry numbers are determined for all the considered stationary points.
Symm_reac1
Symm_prod1
Symm_prod2

Products 1 and 2 are optimized consistently with the level of theory of all the following calculations for the points along the MEP:
Opt_Pr1_vrc
Opt_Pr2_vrc

The level of theory of the products must be consistent with that used for the study of the MEP at level 1. If multireference methods are used for level1 (as specified in ts.dat, see Section 4.1.6.1), keywords level1_31 molpro (for prod1) and level1_41 molpro (for prod2) must be specified in theory.dat. These require the files **level1_prod1vrc_molpro.dat** and **level1_prod2vrc_molpro.dat**, where input for molpro calculations is specified.

After all calculations in ref are concluded, the estoktp.dat file is copied from estoktp.dat_vrc, which serves as template for all the following steps.

estoktp.dat_vrc
ReactionType Barrierless decomp STEPN
 Prods
 nvariational
 nGrid_Opt_ts
 nOpt_ts_1
 n1dTau_ts
 nHL_ts
 nSymm_ts
 npot_corr
 nvrc_tst
 nkTP
End

STEP1: optimization of the reference distance. The calculations at a fixed RTS distance are used to generate the guess for both the geometries and orbitals (for multireference calculations) of all the other geometries of the MEP.
The directory **dist_X.Y** (X.Y is the RTS distance in Angstrom) is created. The estoktp.dat input contains:
ReactionType Barrierless decomp Step1

A first guess of the structure along the MEP at the fixed reference distance specified in ts.dat is optimized at level0 (as specified in theory.dat).
Grid_Opt_ts

For multireference calculations: the optimized geometry is copied at the beginning of the file **output/ts_asl1_step2.inp** and is used as a guess for the active space selection for all the other scanned distances along the MEP.

The resulting structure is optimized at level1 according to what specified in theory.dat.
 Opt_ts_1

If needed, hindered rotors are scanned.
 1dTau_ts

The energy is finally refined according to what specified in theory.dat. For further details on the required theory files, see Section 4.1.6.
 HL_ts

The internal symmetry number of the structure is determined.
 Symm_ts

STEP2: optimization of the points along the MEP

Directories **dist_x.y** for a given list (possibly provided in run.sh) are generated. In each directory, **data/ts.dat** file is updated specifying the fixed RTS distance.

In each x.y directory, **data/estoktp.dat** contains the following in structions:
ReactionType Barrierless decomp Step2

The point of the MEP at fixed distance is optimized at level1. For multireference calculations, an initial guess for the orbitals is generated using the geometry of the reference distance optimized in step1 (**output/grid_opt.out**). The input template for the guess generation is **output/ts_asl1_step2.inp**, whereas the input file for the final geometry optimization is generated from **data/level1_ts_molpro.dat**.
Opt_ts_1

Hindered rotors energies are computed according to what specified in theory.dat and ts.dat.
1dTau_ts

High level energy is calculated according to what specified in theory.dat. For multireference calculations, the **output/ts_asl1_step1.inp** is used as a guess for the active space selection. For larger active spaces than level1 calculations, the active space is opened directly on the geometry of the final structure (see section 4.1.6.1).
HL_ts

The internal symmetry number of the ts is calculated.
Symm_ts

The directory **dist_10.0** (RTS=10 Å) is mandatory and it contains a slightly different input in **data/estoktp.dat**:
ReactionType Barrierless decomp Step21
Prods

The structure is optimized starting from the level1 optimized geometry of the reference MEP distance generated in STEP1 **output/ts_opt.out**. For multireference calculations, the level of theory is specified in **data/level1_ts_molpro.dat** and no guess is considered for the active space (at least not automatically). In this case, a preliminary uhf calculation generates a guess for the orbitals.
Opt_ts_1

High level and symmetry calculations are then performed.
HL_ts
Symm_ts

STEP3: high level calculations scaling and variational analysis
The directory **10.0** is copied as link **100**. This directory is used as reference for the energy scaling in all the other directories. In each **dist_x.y** directory, the input in **data/estoktp.dat** is:

ReactionType Barrierless decomp Step3
HL_ts
kTP

At each fixed distance, the high level energy is rescaled to ensure consistency with the level of theory at which the ΔH(0K) of the reaction is computed in the **ref** directory. In particular, at each fixed distance the energy of the **ts** is:

$$E_{TS} = \left(E_{PR1,HL} + E_{PR2,HL}\right) + \left(E_{distX.Y,HL} - E_{dist10.0,HL}\right) + \left(ZPE_{PR1,l1} + ZPE_{PR2,l2} - ZPE_{dist10.0}\right)$$

The ΔE of the scaling and the reference ΔH(0K) are stored in **output/hl_en.out**.
For instance, if the reactants and products are computed at CCSD(T)//DFT level and the points along the MEP are optimized at CASPT2(HL)//CASPT2(L1) (where HL and L1 may indicate a larger and smaller active spaces/basis sets, respectively), the final energy for each point along the MEP is:

$$E_{TS} = \left(E_{PR1,CCSD(T)} + E_{PR2,CCSD(T)}\right) + \left(E_{distX.Y,CASPT2(HL)} - E_{dist10.0,CASPT2(HL)}\right) + \left(ZPE_{PR1,DFT} + ZPE_{PR2,DFT} - ZPE_{dist10.0,CASPT2(L1)}\right)$$

Once the energy is scaled, kTP computes with cTST the rate constants at the selected fixed **x.y** distance along the MEP.

At this point the rate constant can be computed with vTST. The script **run.sh** generates the directory **variational**, which assembles the results obtained. In this directory, **data/blocks.dat** contains the list of the directories to be included in the variational analysis. The file **data/estoktp.dat** is:

ReactionType Barrierless decomp Step3
variational
kTP

Mess input is assembled from the information on the distances listed in **blocks.dat**. The energies along the MEP are computed at each point with respect to the energy of the products:
$$E_{MEP_{distX.Y}} = (E_{TS} + ZPE_{TS}) - \left(E_{PR1,HL} + E_{PR2,HL} + ZPE_{PR1,L1} + ZPE_{PR2,L1}\right)$$
Where E_{TS} is computed as described above.

STEP4: calculation of the correction potential for vrcTST calculations.
Energy corrections for both the geometry and the high level energies are considered. In vrcTST calculations, the two dissociating fragments are considered at the fixed optimized opt_pr1_vrc and opt_pr2_vrc geometries; the energies are computed as specified in **data/vrc_as.inp**, generally at CASPT2 level of theory and using a small active space. Hence, the former correction considers a geometry optimization involving only the transitional modes between the dissociating fragments; the latter considers the energy at the level of theory used for vrcTST calculations.

In each **dist_x.y** directory, the file **data/estoktp.dat** performs the following steps:
ReactionType Barrierless decomp Step4

The correction potential is computed in two steps:
- Correction for the geometry: the geometry is optimized at level1 of theory. For multireference calculations, the level of theory is specified in **level1_ts_molpro.dat**. The only active optimization parameters are the transitional modes between the two dissociating fragments; warning: lines with internal coordinates specifically set as active in level1_ts_molpro.dat must be commented out manually.
  The geometry considered is that of the two fragments optimized separately at level1 (command opt_pt1_vrc, opt_pr2_vrc for the **ref** directory) at fixed **x.y** distance, as in vrcTST calculations. The initial transitional modes are those optimized at level1 at the selected **x.y** distance, read from **output/ts_opt.out**.
  The outputs of the calculations are stored in **geoms/potcorr_geom.log, me_files/potcorr_geom.me** and **geoms/potcorr_geom.molden** (for multireference calculations only). In **me_files/potcorr_geom.me**, both the energies of the current optimization and the full level1 optimization are reported for comparison.
- Correction for the high level energy: the energy is computed at the level of theory of the following vrctst calculations, as specified in **data/theory.dat** (see Section 4.1.6.1), **data/vrc_as.inp**. The geometry used for the calculations is the one optimized at level1 (**output/ts_opt.out**). The energy obtained is stored and compared with high level calculations (STEP2) in **me_files/potcorr_hl.me**.

potcorr

The directory **10.0** is mandatory and it contains a slightly different input in **data/estoktp.dat**:
ReactionType Barrierless decomp Step4

The correction potential is computed as for the other **x.y** distances. Then, the energy is computed at the same level of vrctst calculations (data/vrc_as.inp) on the geometry of the fragments optimized independently (as for the correction for the geometry) and the transitional modes correspond to those of **output/ts_opt.out**. The corresponding energy is stored in **me_files/potcorr_inf.me** and it is used as a reference for ALL vrctst calculations (it corresponds to the **ENCORR** parameter of the file **data/vrc_as.inp** and it is used to generate the file **vrc_molpro.xml** for vrctst calculations).
Pot_corr inf

The final correction potential of STEP4 is used to scale the energy obtained in the following vrctst calculations (geometry of the fragments fixed as from opt_pr1_vrc, opt_pr2_vrc, and active space specified in data/vrc_as.inp) as follows:

$$E_{CORR,x.y}$$
$$= \left[\left(E_{l1,x.y} - E_{potcorrGE,x.y}\right) - \left(E_{l1,10.0} - E_{potcorrGE,10.0}\right)\right]$$
$$+ \left[\left(E_{HL,x.y} - E_{potcorrHL,x.y}\right) - \left(E_{HL,10.0} - E_{potcorrHL,10.0}\right)\right]$$
These values are written in step 5 in the file **pot.dat** (see Section 4.1.6.3).

STEP5: generation of vrctst calculation input.

The directory **variational** is copied as **vrctst**. The final input to vrctst calculations is then found in the subdirectorys **vrctst/vrc_tst/sr, vrctst/vrc_tst/lr** for short range and long range calculations, respectively. The operations performed are listed in estoktp.dat:

ReactionType Barrierless decomp Step5
Prods

- The input to vrctst calculations is prepared according to the template **data/vrc_tst.inp** and the face symmetry of the fragments in **ts.dat**.
- Pivot points are chosen automatically for both short and long range and the corresponding inputs are **vrctst/vrc_tst/sr/divsruf.inp**, **vrctst/vrc_tst/lr/divsruf.inp**.
- Nodes on which calculations are run are automatically copied from **vrctst/data/machines**.
- The file **pot.dat** containing the correction potential in kcal/mol is generated. The correction potential at each distance **x.y** is computed with respect to the energy ad distance 10.0 computed at the level used for vrctst calculations (**dist_10.0/me_files/potcorr_inf.me**) as described above.

vrc_tst

Calculations are run in **vrctst/vrc_tst/sr** and **vrctst/vrc_tst/lr** directorys independently with the command

rotd –f machines tst.inp >& stdout.log &

The calculations are stored in the directory **scratch,** whereas output structures and corresponding energies and fluxes are stored in **flux.out**. Additional information is provided in sections 4.1.6.2, 4.1.6.3.

STEP6: Rate constant calculation

Once the flux is computed in **vrctst/vrc_tst/sr** and **vrctst/vrc_tst/lr** directorys, the final output is assembled as follows:

- In each **sr lr** directory, obtain the final flux running the commands
  convert_multi convert.inp
  mc_flux mc_flux.inp
  the file mc_flux.out is obtained.
- Run estoktp after setting the **vrctst/data/estoktp.dat** as:
  variational rotd_sr rotd_lr
  ktp

The me_ktp.inp file refers to the output fluxes generate with both short and long range calculations.

The symmetry factor of the ts is derived as $\frac{(symm_{prod1} \cdot symm_{prod2})}{dynamic\ correction}$.

1.4. Input/Output Overview

In order to run EStokTP it is suggested to create a dedicated directory for each calculation type, may it be just a well or a full reaction calculation. The directory must contain the **./data** subdirectory, which must contain all input files necessary to run EstokTP.

The primary input is contained in the **estoktp.dat** file, where the type of calculation and list of modules to call, as well as the computational environment (number of nodes and memory to use) are defined. See section 3.1 for details on what should go in estoktp.dat.

A dedicated file must be used for each species considered in the calculations, so that it may be necessary to compile up to seven species input files: **reac1.dat, reac2.dat, wellr.dat, ts.dat, wellp.dat, prod1.dat, prod2.dat**. See section 3.2 for details.

The level of theory for the Electronic Structure calculations is specified in the **theory.dat** file. This may be sufficient if only g09 is used to perform the calculations. If molpro is used as well or alternatively, module specific theory files will be needed. These are **level0_molpro.dat**, **level1_molpro.dat**, **hl_molpro.dat**, and **onedtau_molpro.dat**. Additionally, specific inputs for saddle point calculations can be requested, in which case we should add **level0_ts_molpro.dat**, **level1_ts_molpro.dat**, **hl_ts_molpro.dat**, and **onedtau_ts_molpro.dat**. See section 3.3 for details. Finally, high level calculations can be performed using a composite (multi-step) approch, requiring an **hl_composite.dat** input file. See section 3.3 for details.

The input necessary to run master equation simulations using Mess is specified in the **me_head.dat** file, while if only thermochemical data are requested, the **thermp.dat** file must be compiled.

The calculation outputs are stored into eight subdirectories that are created contextually to the first call of EstokTP: level0 and level1 geometries are stored in the **./geoms** subdirectory; modules output and z-matrix parameters of geometries are stored in the **./output** subdirectory; files used in irc calculations are stored in the **./irc_files** subdirectory; input and output files for multi dimensional tunneling calculations are stored in the **./md_tunn** subdirectory; the log files of all high level calculations are stored in the **./hl_logs** subdirectory; geometries of hindered rotor structures computed while performing the PES scan of 1D rotors are stored in the **./hr_geoms** subdirectory; data necessary to build the input for the master question calculations as well as for the me block files are stored in the **./me_files** subdirectory; the me blocks and energies (multi dimensional rotational PESs and high level energies) are stored in the ./**me_blocks** subdirectory.

The main results of an EstokTP job are contained in two different sets of files. The first is the input for the Mess solver, namely the **me_ktp.inp** file. The second is the **me_block** file produced for each species (reactant1, reactant2, product1, product2, reactant van derWaals well, product van der Waals well, or transition state). Me block files are named after the species whose data they contain as reac1.me, reac2.me, prod1.me, prod2.me, wellr.me, wellp.me, or ts.me and contain essentially the same information as the me_ktp.inp file, that is geometry, hindered rotor data, frequencies, symmetry, and electronic energy levels. It is not difficult to assemble an input for the Mess solver starting from the me_block files.

2. Installation

2.1. Distribution

EstokTP is distributed as a tar file. Once opened it will create three subdirectories: example, exe, and sources. Commands:

2.2. Compiling

EstokTP comprises two separate codes: the EstokTP source and the source code used to performe the projection of the Hessian and to apply multidimensional tunneling theory. The two codes are reported in two subdirectories in the sources subdirectory: estoktp_1.6 and projRot_1.0. Both must be compiled. Makefiles are already present in the subdirectory and should be edited to choose the preferred compilers and options. The pre-selected compilers and gfortran and gcc. The code has also been compiled with intel fortran without problems. To compile the codes, it is sufficient to hit 'make' in each subdirectory. One executable file should be created in the **./exe** subdirectory for each code: estoktp.x and RPHt.exe. Commands:

```
tar –xvf estoktp_dis_1.0.tar
cd estoktp_dis_1.0
cd sources
cd estoktp_1.6
make
cd ../projRot_1.0
make
```

2.3. Execution Environment

EstokTP uses the executables and the scripts that are present in the **./exe** subdirectory. It is thus sufficient to initialize the execution environment to put the **./exe** subdirectory in your path. If the code is run on a cluster the same applies.

To run EstokTP up to the production of a rate constants it is necessary to have installed at least one electronic structure code (at present Gaussian 09, Gaussian 16, and molpro 2008-2015 are both supported) and the Mess master equation solver (see https://tcg.cse.anl.gov/papr/codes/mess.html or, probably better, the GitHub repository https://github.com/PACChem for distribution and installation instructions).

EstokTP calls Gaussian with the 'g09' and 'g16' commands, so that if Gaussian is properly installed no further action is needed. Molpro and MRCC are called through molprop and mrccp scripts reported in the exe directory. The molprop script takes as input the number of processors to use. These scripts should be edited to point to the proper molpro and mrcc executables. Also, if compiling with gfortran, it is suggested to set the gfortran_unbuffered_all variable to Y, to increase the frequency at which output is written. Finally, external python scripts in **./exe** require the path to the python interpreter (version 3 or higher) be exported as a $python variable. Commands examples:

Bash:
```
export PATH=$estoktpdir/estoktp_dis_1.0/exe:$PATH
export GFORTRAN_UNBUFFERED_ALL=Y
```

```
export python=$python3executable
```

csh/tcsh
```
        set PATH = ($estoktpdir/estoktp_dis_1.0/exe $PATH)
        setenv GFORTRAN_UNBUFFERED_ALL Y
        setenv python = ($python3executable)
```

2.4. Running

A script that launches EstokTP on a node of a cluster is the run_estoktp.com script reported in the **./exe** subdirectory. It should probably be edited to reflect your needs. It is used to launch EstokTP as: 'run_estoktp.com nodename &'. Alternatively EstokTP can be launched directly from a node through the command 'estoktp.x'.

3. Input files

All data files necessary to perform an EstokTP job are stored in the **./data** subdirectory of the main job directory. Section 1.4 reports an overview of the input files needed. In the following a detailed description of each file is reported. It should be noted that the input is not case sensitive.

3.1. estoktp.dat: job selection and modules

Estoktp.dat is made of three sequential parts. The first contains a list of keywords, the second the list of modules to use, and the third a concise description of the computational environment.

3.1.1. Reaction Type/Global Keywords

These keywords define the type of EstokTP job that should be performed. Each one should stay on a separate line. The order is not important.

- **ReactionType** *reaction option1 option2:* this keyword defines which type of reaction is studied. The *reaction* field can be either Addition, Abstraction, Isomerization, Betascission, MECP, or barrierless. In the absence of this keyword the code assumes that the job is dedicated to studying a single well, with data in the reac1.dat file. The *option1* parameter, which must be specified, can be either 1TS (active for all reaction types made exception for barrierless), 2TS (active only for addition and abstraction), 3TS (active only for abstraction), or decomp (for barrierless reactions). See section 1.3 for how the choice of *tsnumber* determines the model used for the calculations. Example: 'ReactionType abstraction 3TS'. The parameter *option2* is used only for barrierless calculations and refer to the calculations step (step1, step2, step3, step4, or step5).

- **Wellr** *algorithm leveloftheory*: it means that a van der Waals well must be searched on the reactant side. If the *algorithm* and *leveloftheory* keywords are left blank than the search is performed starting from the z-matrix structure that must be specified in the wellr.dat file. If the *algorithm* keyword is 'findgeom' then a guess structure is generated from the TS geometry. No z-matrix specification is in this case needed in the wellr.dat file. This is the recommended option when a van der Waals well is searched. The *leveloftheory* keyword, which can be either 'level1' or 'level0', specifies whether the search should be performed starting from level0 of theory or directly from level1. Example: 'wellr findgeom level1'.

- **Wellp** *algorithm leveloftheory*: it means that a van der Waals well must be searched on the product side. If the *algorithm* and *leveloftheory* keywords are left blank than the search is performed starting from the z-matrix structure that must be specified in the wellp.dat file. If the *algorithm* keyword is 'findgeom' then a guess structure is generated from the TS geometry. No z-matrix specification is in this case needed in the wellp.dat file. This is the recommended option when a van der Waals well is searched. The *leveloftheory* keyword, which can be either 'level1' or 'level0', specifies whether the search should be performed starting from level0 of theory or directly from level1. Example: 'wellp findgeom level1'

- **Prods**: it tells the code that the formation of products should be explicitly considered in writing the Mess input.

 - **Variational** : it specifies that an IRC scan will be performed (and thus the IRC module will be called and should be activated in the requested series of calculations) and variational transition state theory will be used to determine the rate constant.

 - **Variational Internal**: it specifies that after performing the IRC scan frequencies will be computed in internal coordinates using the theory described in Jackels, C. F. ; Gu, Z.; Truhlar, D. G. Reaction-Path Potential and Vibrational Frequencies in Terms of Curvilinear Internal Coordinates, J. Chem. Phys. 1995, 102, 3188-3201.

- **FrozRTS** *distance*: it means calculations are performed keeping the RTS coordinate frozen. If activated, the code will be able to run up to the determination of the rate constant determining geometries and frequencies and performing hindered rotor scans performing all geometry optimizations constraining the RTS coordinate. This option can be used to perform several types of calculations. One possibility for example is to use it to perform variational TST calculations as a function of the RTS coordinate, so that the minimum energy path can be determined for a barrierless reaction. The *distance* parameter is used by the Opt_TS_0 module to set the RTS value. It should be noted (see sections 1.3.2 and 1.3.3) that RTS is the name of the distance coordinate used by EstokTP to define the distance between react1 and react2 in abstraction and addition reaction. Example: 'FrozRTS 2.5'.

- **ResIRC**: this keyword requests a linear rescaling of the potential energy profile calculated in the IRC module to high level quality energies. For this purpose two additional high level calculations are performed for the first and last geometries computed along the IRC path.

- **ResIRCAll**: same as ResIRC, except that high level energies are computed for all points determined along the IRC path. It is, obviously, quite expansive computationally.

- **MdTunnel**: it requests to compute tunneling contributions using multi dimensional tunneling theory, in particular the present implementation uses small curvature theory with reaction path Hamiltonian frequencies. It should be noted that this calculation requires that a large number of points is taken along the reaction path (on the average 200 points with a stepsize of 0.3 Bohr). It is suggested to use this keyword in conjunction with the ResIRC keyword. By default, in the absence of the MdTunnel keyword, the Eckart tunneling model is used. When used also the **projrcoo** keyword should be present.

- **Projrcoo**: it projects the reaction coordinate from the Hessian as suggested by Miller, W. H.; Handy, N. C.; Adams, J. E. Reaction Path Hamiltonian for Polyatomic Molecules. J. Chem. Phys. 1980, 72, 99-112.

- **NoTunnel**: this keyword disables the use of tunneling in rate constant calculations.

- **Recover**: Recovers the output from a calculation that was interrupted after generating output prior to processing it (e.g., a high level MOLPRO calculation). It can be used only if a single

module is active (that is, the one from which we want to recover the data).

 - **Debug** *debuglevel*: it is a keyword that defines the density of writing in the output. At present, as this is the first release of the code, use 2 (that is dense output), as it will help with debugging the code. Example: "debug 2"

- Stoichiometry *atomstypenumber* : this keyword specifies the number of atoms of each type present in the molecule. It is an input for an external call to a code that is to be used to determine Lennard Jones parameters, which are needed for the master equation calculation. It is not used at present. Example: "stoichiometry C4H8"

3.1.2. List of Modules

An EstokTP job typically consists of a set of electronic structure calculations, each used to produce certain data (geometries, frequencies, or energies most typically) that will then be necessary either to perform successive higher level calculations or, eventually, to write a master equation input file or a master equation block. These calculations are grouped in modules, which can be called independently, provided that the necessary input is available. Each job type (be it a reaction or a well job) requires that a minimal set of modules is called, either sequentially or one by one. See sections 1.3.1-1.3.5 for a list of the modules requested by each job type. The full list of all the available modules is reported in the following, with a brief description. The Module section must be terminated with a line containing the 'End' word.

Opt_Reac1
Opt_Reac2
Opt_Prod1
Opt_Prod2
Opt_WellR
Opt_WellP
Modules that perform the search for the minimum energy structure of reactants, products or wells at level 0 of theory. Structural parameters are saved in the **./output** subdirectory as reac1_opt.out, reac2_opt.out, prod1_opt.out, prod2_opt.out, wellr_opt.out, wellp_opt.out (briefly, $species_opt.out). Structural data for all minimum energy structures found in the stochastic scan are saved as well in the order in which they are found as reac1_opt_1.out, reac1_opt_2.out, and so on. Geometries in xyz format are saved in the **./geoms** subdirectory as reac1_01.xyz, reac1_02.xyz, etc. Level 0 energies are reported in the title line of the xyz geometries and as last line of the $species_opt.out geometries. A log of the calculations is reported in the reac1.out, reac2.out and so on files in the **./output** subdirectory. It can be used to check the progress of the calculations. If the 'wellr findgeom level0' or 'wellp findgeom level0' keywords are used then the guess geometry for the van der Waals well optimization will be taken from the ./output/ts_opt.out file and modified by changing the bond breaking/bond forming length.

Grid_Opt_TS
Module that performs a grid scan along a distance coordinate (usually RTS) to determine a suitable guess for a TS search. The structure (z-matrix parameters) with the highest energy found along the

scan is saved in the ./output/ts_opt.out file and used for successive calculations. A log of the grid scan is saved in the ./output/grid_opt.out file and can be used to check the status of the calculations.

2dGrid_Opt_TS
Module that performs a two dimensional grid scan along a distance coordinate (usually RTS) and another internal coordinated to determine a suitable guess for a TS search. The structure (z-matrix parameters) with the highest energy found along the scan is saved in the ./output/ts_opt.out file and used for successive calculations. A log of the grid scan is saved in the ./output/grid_opt.out file and can be used to check the status of the calculations.

Opt_TS_0
Module used to determine the transition state structure at level 0 of theory. The structure so determined is saved in the ./output/ts_opt.out file.

TauO_TS
Module that performs a MonteCarlo search along the selected dihedral coordinates to search for rotational conformers of the transition state. The one with the minimum energy is saved in the ./output/ts_opt.out file and used for successive calculations.

Opt_Reac1_1
Opt_Reac2_1
Opt_Prod1_1
Opt_Prod2_1
Opt_TS_1
Opt_WellR_1
Opt_WellP_1
These modules are used to determine the minimum energy structure of reactants, products or wells at level 1 of theory. Structural parameters are saved in the **./output** subdirectory as reac1_opt.out, reac2_opt.out, prod1_opt.out, prod2_opt.out, wellr_opt.out, wellp_opt.out (briefly, $species_opt.out). Thus overwriting the files written by level0 calculations. Geometries in xyz format are saved in the **./geoms** subdirectory as reac1_l1.xyz, reac2_l1.xyz, etc. Level 1 energies are reported in the title line of the xyz geometries and as last line of the $species_opt.out geometries. A log of the calculations is reported in the level1.out file in the **./output** subdirectory. Structures determined at level1 of theory are then used for me calculations and are saved in the **./me_files** subdirectory as reac1_ge.me, reac2_ge.me and so on. Frequencies are calculated in these modules from the Hessian matrix if no hindered rotor calculation is requested and are saved in the **./me_files** subdirectory as reac1_fr.me, reac2_fr.me and so on. If hindered rotors are present, then the Hessian matrix (reac1_fcmat.log, reac2_fcmat.log and so on) is saved in the **./output** subdirectory, and will then be used by the hindered rotor module (see the 1dTau_xxx module description) to compute a set of frequencies from which hindered rotors have been projected. Zero point energies are computed by these modules and saved in the **./me_files** subdirectory as reac1_zpe.me, reac2_zpe.me, and so on. If the 'wellr findgeom level1' or 'wellp findgeom level1' keywords are used then the guess geometry for the van der Waals well optimization will be taken form the ./output/ts_opt.out file and modified by changing the bond breaking/bond forming length. It means that no preliminary call to the Opt_WellR or Opt_WellP module is necessary. The global symmetry number (ratio of external rotational and external optical symmetry numbers, which

should account for optical isomers found in hindered rotor scans) is here set to 1 and should then be edited or corrected calling the symmetry modules (see below).

1dTau_Reac1
1dTau_Reac2
1dTau_Prod1
1dTau_Prod2
1dTau_WellR
1dTau_WellP
1dTau_TS

The 1dTau modules perform one dimensional hindered rotor scans along the dihedral coordinates indicated in the input data files for reactants, products, wells or transition state (nhind keyword). These modules should also be called when a multidimensional hindered rotor scan is requested for the dihedral coordinates treated multidimensionally. Log files of the calculations are saved in the **./output** subdirectory as reac1_hr.out, reac2_hr.out, and so on. The calculated 1D hindered rotor PES, the list of atoms in the rotating top and that belong to the rotation axis are saved in the **./me_files** subdirectory as reac1_hr.me, reac2_hr.me, and so on. The Hessian matrix is then projected and projected frequencies are saved in the **./me_files** subdirectory as reac1_fr.me, reac2_fr.me and so on. Hindered rotor structures for each point are saved in the **./hr_geoms** subdirectory.

MdTau_Reac1
MdTau_Reac2
MdTau_Prod1
MdTau_Prod2
MdTau_WellR
MdTau_WellP
MdTau_TS

The mdTau modules perform a multidimensional hindered rotor scan along the dihedral coordinates indicated in the input data files for reactants, products, wells or transition state (nhind2D or nhind3D keywords). At the present level of theory, only one multidimensional hindered rotor scan is allowed for molecule in order to preserve the correct coupling between internal and external rotational motions as implemented in the Mess master equation solver. The calculated potential energy surface is saved in the **./me_files** subdirectory in the reac1_2dhr_nofr01.dat (or reac2_.. or reac1_3dhr_.. and so on) if no frequencies were estimated at each point of the PES and in the reac1_2dhr_01.dat (or reac2_.. or reac1_3dhr_.. and so on) file otherwise (if the mhr_freqs keyword has been used in the species input data). The frequencies, if calculated along the multi dimensional PES, are determined from a Hessian from which the corresponding hindered rotors have been projected out. Details on rotating groups necessary to solve the multi dimensional hindered rotor equation as implemented in Mess are saved in the reac1_mdhr.me file (or reac2, prod1, and so on).

Symm_Reac1
Symm_Reac2
Symm_Prod1
Symm_Prod2

Symm_WellR
Symm_WellP
Symm_TS
The symmetry keywords are used to determine the rotational and optical symmetry numbers for external and internal rotations. The implemented algorithm works as follows. The external rotational and optical symmetry numbers are determined through the symmetry_number code that is part of the Mess suite of codes. A tolerance of 0.07 is used as input to symmetry_number, as it was found to give better predictions than obtained using the default 0.001 parameter. Successively the hindered rotor PES is scanned to check if optical isomers are found along the rotational scan. For this purpose structures saved in the **./hr_geoms** subdirectory are used to find local minima along the rotational PES for each hindered rotor. If the computed energy differs by less than 0.00001 Hartrees from the absolute minimum than it is considered as an optical isomer. The species external optical isomer number is then divided by the number of rotational optical isomers. It should be mentioned that this algorithm is not yet extended to treat optical symmetry in multidimensional hindered rotors. For this reason if a multidimensional rotor treatment has been requested all optical symmetry numbers (internal and external) for the considered species are set to 1, as it is supposed than an eventual external optical symmetry would be captured by the multi dimensional hindered rotor analysis. The log files for symmetry calculations is saved in the **./output** subdirectory as reac1_sym.out (or reac2_sym.out, and so on). The reac1_ge.me (or reac2_ge.me and so on) file, which contains the global symmetry number for each species (ratio of external rotational and external optical symmetry numbers), which is set to 1 when written for the first time in the level1 calculations (opt_reac1_1 and so on), is then updated on the basis of what calculated in the symmetry module.

HL_Reac1
HL_Reac2
HL_Prod1
HL_Prod2
HL_WellR
HL_WellP
HL_TS
These modules perform high level calculations for all species. The geometry adopted for this purpose is the one determine at level 1 of theory and saved in the ./output/reac1_opt.out and so on files. The output of all high-level calculations are saved in the **./hl_logs** subdirectory as reac1_molpro.out, reac1_g09.out, or reac1_hlcomp.out and so on.

IRC
The internal reaction coordinate module is at present set to process files produced through g09 IRC calculations. The required information are Hessians and energies calculated along the reaction path, which is started from the transition state structure located using level 1 calculations whose geometry is stored in the ./output/ts_opt.out. It is important that the level of theory at which IRC calculations are performed (specified in the theory.dat file) is the same as that used to find the transition state. All files used in IRC calculations are stored in the **./irc_files** subdirectory. When the IRC module is called the 'variational' keyword should also be present in the estoktp.dat input, and vice-versa. The output of the IRC module is saved in the variational.me file, which is stored in the **./me_files** subdirectory. Additionally, the IRC module should be called whenever multi

dimensional tunneling calculations are requested. As MD tunneling calculations are fast compared to electronic structure calculations, they are performed always when the IRC module is called. The input and output files for such calculations are saved in the **./md_tunn** subdirectory. Notably, the trajec.xyz contains all the geometries of each point of the IRC scan saved in xyz format and thus easily readable using molecular structure visualization softwares (such as molden or vmd), while the mdtunn.out files contains the log file of the multi dimensional tunneling calculation. The tunneling coefficient is reported as kSCT at the end of the file. The file created and used by the Mess solver for multi dimensional tunneling calculations is the imactint.dat file, which contains the natural logarithm of the imaginary action integral as a function of the energy (in cm-1), measured from the energy of the saddle point. Please look at the **mdTunnel** keyword description for further information on how to run multidimensional tunneling calculations.

kTP
The kTP module generates, when the 'reaction' keyword is present in the estoktp.dat file, the me_ktp.inp file, which is the input for the Mess solver. The data used for this purpose are those saved by previous calls to all modules necessary for the desired calculation in the **./me_files** subdirectory. Also, the me_blocks for all species participating to the reaction are created and saved in the **./me_blocks** subdirectory. The output of the Mess solver is saved in the rate.out file, which can be found in the job directory and is also copied to the **./output** subdirectory.
If the reaction keyword is not present in the estoktp.dat file then it is assumed that this is a well job and thus only the me_block files for the reac1 species are generated.

ModArr
This module is not yet active. Its purpose is to fit all rate constants reported in the rate.out file produced by the Mess solver to the modified Arrhenius form.

3.1.3. Computational environment

The computational environment is defined by the last 4 lines of the estoktp.dat file, which must be preceded by the 'End' line that marks the end of the module section. The first line contains the number of cores used for the low and high level calculations, separated by a comma, the second is a comment line, the third line contains the memory requested for low and high level calculations (expressed in Mega words, with the MW letters attached to the number), and the fourth line is a comment line. In practice, the calculations for each module are performed using the low level parameters except for high level modules, which use the high level parameters.
Example:
12,4
numprocll, numprochl
200MW 300MW
gmemll gmemhl

3.2. Species Data Files

An EstokTP job can include up to seven species, each one requiring a dedicated input file: 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 one transition state (ts.dat). Each file should terminate with a line containing the 'end' word. EstokTP retrieves data from the input files searching for keywords that correspond to a specific set of input data. For example the geometry is reported after the charge keyword, while the hindered rotor section is reported after the nhind keyword. The format of each file is described in detail in the following.

3.2.1. reac1.dat, reac2.dat, prod1.dat, prod2.dat

Each of these files should contain the following data blocks. The order in which the blocks are reported in the file is not relevant.

- **nosmp**: it gives the number of sampling points to be used in the Monte Carlo sampling of the configurational space defined by the coordinates listed in the ntau block, the geometric (in degrees) and the energy (in Hartree) threshold to check if the new geometry found in the stochastic search differs from geometries found in previous scans. The whole block is formed by two lines (keyword and data).
Example (requests 5 sampling points):

  nosmp
  5 1.0 0.00001

- **ntau**: lists the number and names of the coordinates that are stochastically sampled to search for the absolute minimum energy structure. The whole block is formed by three lines (keyword, number of sampled coordinates and comment line) plus a list of the names of the sampled coordinates, each followed by its sampling interval. It is assumed that the sampled coordinates are dihedral angles whose names and z-matrix coordinates are defined in the 'charge' block.
Example (scans b1 and b5 coordinates randomly sampling their values in the 0-360 interval for b1 and in the 0-120 interval for b2):

  ntau
  2
   -- name and sampling interval
  b1 0 360.
  B2 0 120.

- **nhind**: lists the number and names of the dihedral coordinates that are to be treated as hindered rotors. The whole block is formed by three lines (keyword, number of dihedral coordinates and a comment line) plus a list of the names of the dihedral coordinates, each followed by its scan interval, the number of points to take on the PES, and periodicity. The scan step is obtained dividing the scan interval by the number of PES points. It is assumed that the first and last point of the scan interval correspond to the same molecular structure, thus the calculation is performed only once for the first point, so that the total number of electronic structure calculations is equal to the number of PES points indicated. Though the ntau and nhind coordinates are often the same this may not always be necessarily the case and is thus not enforced.

Example (treats b1 and b5 coordinates as hindered rotors, using a symmetry number of 1 for b1 and of 3 for b2, that is in fact scanned between 0 and 120. The calculations at 360. And 120. Degress for b1 and b2 are not performed, as they are assumed to give the same energy as those performed at 0 degrees).

```
Nhind
2
-- name hindmin hindmax points periodicity
b1 0 360. 12 1
b2 0 120. 4 3
```

- **natom**: it must be followed by three numbers: the total number atoms in the molecule excluding dummy atoms, the total number of atoms including dummy atoms and an index, which can be either 0 and 1, that indicates whether the molecule is not linear (0) or is linear (1) (not necessary for diatomic molecules).
  Example:

```
natom natomt ilin
6 6 0
```

- **charge**: this block contains the charge and spin multiplicity and is followed by the z-matrix of the molecule. In specifying the z-matrix each coordinate name must differ from the others for the same species, and for the reac1 and reac2 species in abstraction and addition reactions.
  Example (ethylene charge, spin and z-matrix):

```
charge  spin  atomlabel
0   1
c1
c2  c1 rcc1
h3  c1 rch1 c2 ahcc1
h4  c1 rch2 c2 ahcc2 h3 b1
h5  c2 rch3 c1 ahcc3 h3 b2
h6  c2 rch4 c1 ahcc4 h5 b3
```

- **intcoor**: this block contains first guess values for all coordinates reported in the z-matrix, made exception for the coordinates listed in the ntau block, whose value is determined through stochastic sampling. Coordinate names and guess values must be separated by a space.
  Example (coordinates for ethylene):

```
intcoor
rcc1 1.5
rch1 1.08
rch2 1.08
rch3 1.08
rch4 1.08
ahcc1 110.
Ahcc2 110.
Ahcc3 110.
Ahcc4 110.
B1   178.
```

```
B2   5.
B3   177.
```

- **symmetryfactor**: it is the global symmetry factor for the molecule, defined as the ratio between external rotational symmetry number and external optical symmetry number. It should also account for optical isomers found in the hindered rotor scan. Though it has to be specified in this file, it is recalculated when the Symmetry modules are called.
Example (symmetry for ethylene):

```
SymmetryFactor
4.
```

- **nelec**: it lists the number of electronic states for the considered species and their multiplicity and energy relative to the ground state, including the ground state. The nelec keyword must be followed by one line giving the number Ne of electronic states and by Ne lines giving energy (in cm-1) and multiplicity of each electronic state.
Example (electronic states for OH):

```
nelec
2
0.  2
140. 2
```

In addition to these data blocks, which should always be included in the data file, even in the absence of hindered rotors or coordinates to sample stochastically (in which case the nhind and ntau parameters should be set to 0), the following data blocks can be optionally added:

- **nhind2D** (same rules for **nhind1D** and **nhind3D**): number and name of rotors for which a multidimensional scan is requested. The block is structured similarly to the nhind block, thus it is formed by three lines (keyword, number of multi dimensional rotors and a comment line) plus nlines containing the list of the names of the dihedral coordinates for which a multidimensional treatment is requested (nlines = rotor dimensionality x number of multidimensional rotors). At present, this keyword is designed to generate inputs for the Mess solver, and thus for the multidimensional hindered rotor model it implements. This is a sophisticated model that accounts for the coupling between internal and external rotations. Because of this it does not make sense (and it is thus not supported at present) to use this treatment for more than 1 multi dimensional rotor at a time, as the correct theoretical treatment would be to increase the dimensionality of the rotor, rather than have several rotors of lower dimensionality (which also means that nhind1d, nhind2d, and nhind3d calculations are alternative ones with respect to the other). This block is read by the mdTau modules, which must therefore be present in the estoktp.dat input file.
Example (treats b1 and b6 as 2D hindered rotors, using a symmetry number of 3 for each, so that 4 points are used to scan the coordinates between 0 and 120 for 4 times, for a total number of 16 points taken along the PES).

```
nhind2D
1
```

→namehind,hindmn,hindmx,hindstep period
b1 0 120 4 3
b6 0 120 4 3

3.2.2. wellr.dat, wellp.dat

The wellr.dat and wellp.dat data files should always be filled in when the wellp and wellr keywords are present in the estoktp.dat file. They should be filled in as described in section 3.2.1 unless the findgeom option is associated to the wellr and wellp keywords (which is recommended). In that case, the only blocks that should be included are nosmp, ntau, nhind, charge, Symmetryfactor, and nelec. If the findgeom option is used, it is suggested to set the number of sampling points to 1 and that of sampling coordinates to 0, as the findgeom algorithm should help to find the van der Waals wells connected to the TS, which may not be that of absolute minimum energy which may be found using the stochastic sampling strategy. It is also suggested to set the nhind keyword to 0 (that is, do not consider hindered rotors), as what is necessary for rate calculations is the energy of the well, rather than an accurate evaluation of its density of states so that the RRHO approximation should be sufficient.

3.2.3. ts.dat
The data file for the transition state contains several blocks that are the same as those of the species data described in section 3.2.1. In particular it must contain the: **nosmp**; **ntau**, **nhind**, **Symmetryfactor**, and **nelec** blocks, organized as described in section 3.2.1. The natom and intcoor blocks should not be present, as the geometry of the TS is constructed from EstokTP following rules that depend on the reaction type starting from the data reported in the reac1.dat and reac2.dat data files. In addition, the following blocks must be present:

- **charge**: requires two integers: charge and spin multiplicity.
  Example:

  charge spin
  0 2

- **isite**: requires three integers: the atom numbers for isite, jsite, and ksite. The meaning of these three numbers depends on the reaction type. They are used to construct the z-matrix of the transition state and to define the reacting atoms. For example isite is the abstracted atom in an abstraction reaction, the site to which an atom or molecule is added in an addition reaction; the site to which an atom is transferred in an isomerization reaction, and one of the two atoms whose bond is being broken in a beta-scission reaction. See sections 1.3.2 – 1.3.5 and keywords **rmin**, **rmin1** and **rmax1** for the meaning.
  Example:

  isite jsite ksite
  11 3 2

- **rmin**: this block must be present in abstraction and addition reactions. It is not read in isomerization and beta-scission reactions. It reports data for constructing the TS z-matrix and

defines the TS search grid. It is a four lines block, with the first line being the keyword, the second three numbers defining the interval of the grid search. The grid search is a restrained geometry optimization with the RTS coordinate kept frozen. The first number is the lowest value of RTS, the second the highest and the last the number of steps to progressively move from the lowest to the highest value of RTS. The third line is a comment line while the fourth contains 5 numbers, separated by commas, which defines coordinates aabs1, babs1, aabs2, babs2, and babs3 that are used to construct the transition state z-matrix. The 5 coordinates are two angles (aabs1 and aabs2) and three dihedrals (babs1, babs2, and babs3) that are used to define the initial relative orientation of fragments reac1 and reac2. See sections 1.3.2 – 1.3.5 for a precise definition of these coordinates.

Example (scans RTS between 1.4 and 2.2 in 5 steps):

```
rmin rmax nr
1.4 2.2 5
→aabs1,babs1,aabs2,babs2,babs3
90., 90., 90., 175., 90.
```

- **rmin1**: this block is used in isomerization reactions. It consists of the keyword line and a line containing 5 numbers: distance1, step1, distance2, step2, and ireact. The isomerization algorithm is a pulling algorithm working so that the distance between two atoms, the isite and ireact is progressively diminished from the minimum value found among all the react1 conformers located in the stochastic conformational analysis to distance1 with step step1 and then decreased to distance2 with step step2.
  Example (atom 4 is moved toward atom isite in steps of 0.2 Å down to 1.7 Å and then down to distance 1.2 Å in steps of 0.1 Å):

```
rmin1 rstp1 rmin2 rstp2 ireact
1.7 0.2 1.2 0.1 4
```

- **rmin1_1** and **rmin1_2**: these blocks are used in isomerization reactions to specify a two-dimensional grid. The first block consists of the rmin1_1 keyword line and a line containing 5 numbers: distance1, step1, distance2, step2, and ireact. The isomerization algorithm is a pulling algorithm working so that the distance between two atoms, the isite and ireact is progressively diminished from the minimum value found among all the react1 conformers located in the stochastic conformational analysis to distance1 with step step1 and then decreased to distance2 with step step2. The second block consistes of the rmin1_2 keyword line and a line containing 4 number (distance1, step1, distance2, step2) followed by the name of the internal coordinate to be scanned.
  Example (atom 4 is moved toward atom isite in steps of 0.2 Å down to 1.7 Å and then down to distance 1.2 Å in steps of 0.1 Å, while angle A6 is moved towards 120° and then 90° in -6° and -3° steps):

```
rmin1_1 rstp1 rmin2 rstp2 ireact
1.7 0.2 1.2 0.1 4
rmin1_2 rstp1 rmin2 rstp2 ireact
120. -6. 90. -3. A9
```

- **rmax1**: this block is used in beta-scission reactions. It consists of the keyword line and a line containing 5 numbers: distance1, step1, distance2, step2, and ireact. The beta-scission algorithm is a pushing algorithm working so that the distance between two atoms, the isite and ireact is progressively increased from that of the minimum energy structure to distance1 with step step1 and then to distance2 with step step2.
  Example (the distance between atom 8 and isite is increased in steps of 0.2 Å up to 1.8 Å and then up to distance 2.4 Å in steps of 0.1 Å):

  rmax1 rstp1 rmax2 rstp2 ireact
  1.8 0.2 2.4 0.1 8

Additional blocks for the ts.dat file are the **nhind1D, nhind2D**, and **nhind3D** blocks, organized as described in section 3.2.1 and the ts_geom block:

- **ts_geom**: this block, read by module opt_TS_1, is optionally used to request for level1 calculations the use of the geometry of a transition state different from that of the minimum energy conformer found in the stochastic search performed at level 0 of theory by the 1dTau_TS module. It is made by the keyword line followed by an integer number iTS indicating the structure to use for level1, and thus successive, TS calculations. When invoked, the geometry of conformer iTs saved in the **./output** subdirectory as ts_opt_iTS.out is copied to file ./output/ts_opt.out, which is thus overwritten.
  Example (uses geometry, in particular z-matrix parameters, saved in ./output/ts_opt_4.out to perform level1 and successive calculations, as the ./output/ts_opt.out file is overwritten)

  ts_geom
  4

- **reac_geom**: this block, reported in file ts.dat, is optionally used to request for level0 calculations of isomerization reactions the use of a specific isomer of the reactant found in the torsional scan, rather than the one at which the distance between reacting atoms is minimal. When invoked, the geometry of conformer iR saved in the **./output** subdirectory as react_opt_iR.out is used to generate the Z-matrix for the isomerization scan.
  Example (uses geometry, in particular Z-matrix parameters, saved in ./output/reac1_opt_02.out to generate the Z-matrix for isomerization)

  reac_geom
  2

3.3. Electronic Structure and Master Equation Inputs

EstokTP relies on external codes to perform electronic structure and master equation calculations. At present the supported electronic structure codes are Gaussian 09 and molpro. Extension to Nwchem and Gamess is planned within the next year. The supported master equation code is Mess,

which is part of the Papr distribution (http://tcg.cse.anl.gov/papr/). Sections 3.3.1 and 3.3.2 describe how to select the level of theory to use for electronic structure calculations and the input for master equation calculations performed with Mess.

3.3.1. Theory files

The electronic structure code to use for each calculation module is defined in the theory.dat file. The level of theory to be used for each module is also defined in theory.dat when Gaussian 09 is used and in a separate set of files for molpro.

3.3.1.1. Theory.dat

The theory.dat file is organized in blocks, each one identified by a keyword. Each block defines the electronic structure code to use for a set of modules through a string reported after the keyword. If the code of choice is g09, the level of theory to use in the calculations is reported in the lines that follow the keyword, which are then used as the route section for Gaussian calculations. All optimizations should be performed using internal coordinates, thus any opt g09 call should include the internal option. Also it is suggested to suppress the use of symmetry, and thus include the 'nosym' option in the g09 route section. If the code is molpro one or more separate files must be prepared, as described in section 3.3.1.2.

- **level0**: level of theory for modules **Opt_Reac1**, **Opt_Reac2**, **Opt_Prod1**, **Opt_Prod2**, **Opt_WellR**, **Opt_WellP**, **Grid_Opt_TS.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
  Example (performs optimization using g09 at the b3lyp/6-31+g(d,p) level of theory):

  level0 g09
  b3lyp/6-31+g(d,p) opt=internal
  int=ultrafine nosym

- **level0_ts:** level of theory for modules **Opt_TS_0** and **TauO_TS.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
  Example (performs ts search at using g09 at the b3lyp/6-31+g(d,p) level of theory):

  level0_ts g09
  b3lyp/6-31+g(d,p) opt=(ts,calcfc,noeig,intern,maxcyc=50)
  int=ultrafine nosym

- **level1**: level of theory for modules **Opt_Reac1_1, Opt_Reac2_1, Opt_Prod1_1, Opt_Prod2_1, Opt_WellR_1, Opt_WellP_1.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
  Example (performs optimization using g09 at the m062x/6-311+g(d,p) level of theory):

  level1 g09
  m062x/6-311+g(d,p) opt=internal
  int=ultrafine nosym freq

- **level1_ts:** level of theory for module **Opt_TS_1.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
Example (performs transition state search using g09 at the m062x/6-311+g(d,p) level of theory):

    level1_ts g09
    m062x/6-311+g(d,p) opt=(ts,calcfc,noeig,intern,maxcyc=50)
    int=ultrafine nosym freq

- **hind_rotor:** level of theory for modules **1dTau_Reac1, 1dTau_Reac2, 1dTau_Prod1, 1dTau_Prod2, 1dTau_WellR, 1dTau_WellP, MdTau_Reac1, MdTau_Reac2, MdTau_Prod1, MdTau_Prod2, MdTau_WellR, MdTau_WellP.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword. If the mdtau module is called, it searches for an additional line for the 'mhr_freqs' keyword. If present, frequencies are computed for each point along the hindered rotor path.
Example (performs hindered rotor scans using g09 at the b3lyp/6-31+g(d,p) level of theory, when called from an mdtau module, performs a frequency analysis for each point of the hindered rotor PES).

    hind_rotor g09
    b3lyp/6-31+g(d,p) opt=internal
    int=ultrafine nosym
    mhr_freqs

- **hind_rotor_ts:** level of theory for modules **1dTau_TS** and **MdTau_TS.** If the g09 code is used two different g09 route sections are reported in four separate lines after the keyword. The first two lines are used for the first attempt at determining the torsional PES, while the other two are used as fallback option if the first attempt fails. It is assumed that in the first attempt the search is performed for a constrained TS, that is a saddle point where one (or more) dihedral coordinates is kept fixed. If this fails, the hindered rotor analysis is performed fixing the length of the bond that is being formed/broken to the TS value. If the mdtau_ts module is called, it searches for an additional line for the 'mhr_freqs' keyword. If present, frequencies are computed for each point along the hindered rotor path.
Example (performs hindered rotor scans using g09 at the b3lyp/6-31+g(d,p) level of theory first searching for a constrained TS and then performing a constrained optimization if this fails).

    Hind_rotor_ts g09
    b3lyp/6-31+g(d,p) opt=(ts,calcfc,noeig,intern,maxcyc=50)
    int=ultrafine nosym
    b3lyp/6-31+g(d,p) opt=internal
    int=ultrafine nosym

- **symmetry:** level of theory for modules **Symm_Reac1, Symm_Reac2, Symm_Prod1, Symm_Prod2, Symm_WellR, Symm_WellP.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
Example (if requested by the symmetry algorithm, performs optimization using g09 at the b3lyp/6-31+g(d,p) level of theory):

```
symmetry g09
b3lyp/6-31+g(d,p) opt=internal
int=ultrafine nosym
```

- **symmetry_ts:** level of theory for module **Symm_TS**. If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
  Example (if needed, performs optimization using g09 at the b3lyp/6-31+g(d,p) level of theory):

```
symmetry_ts g09
b3lyp/6-31+g(d,p) opt=(ts,calcfc,noeig,intern,maxcyc=50)
int=ultrafine nosym
```

- **hlevel**: level of theory for modules **HL_Reac1, HL_Reac2, HL_Prod1, HL_Prod2, HL_WellR, HL_WellP.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
  Examples (searches for a molpro or composite input file):

```
hlevel molpro
hlevel composite
```

- **hlevel_ts**: level of theory for module **HL_TS.** If the g09 code is used the g09 route section is reported in two separate lines after the keyword.
  Examples (searches for a molpro or composite input file):

```
hlevel_ts molpro
hlevel_ts composite
```

- **irc**: level of theory for module **IRC.** At present only g09 is supported. It requires that four lines are reported after the keyword. The first two contain the g09 route section for the forward IRC scan, while the second two contain the route section for the backward scan. An additional line is also searched by the IRC module to check if the HRcc option is active. It is the HRcc keyword followed two numbers, ifor and iback. This option is used to rescale the hindered rotor potentials along the IRC path by performing ifor +1 and iback+1 times the hindered rotor analysis at equally spaced points along the IRC path. To produce consistent results, the level of theory should be the same used to determine the level1 geometry and frequencies. This is not enforced at present and it is left to the user to check that this condition is respected.
  Example (performs an IRC scan at the m062x/6-311+g(d,p) level taking 10 points in the forward direction and 10 in the backward using a step of 0.03 Bohr for a total number of 21 points. The hindered rotor potential is re-evaluated 4 times at points 1, 5, 16, and 21. Point 11 is the transition state and it is assumed that its potential is already available):

```
irc g09
m062x/6-311+g(d,p) irc(forward,calcall,stepsize=3,maxpoints=10)
int=ultrafine  nosym  iop(7/33=1)
m062x/6-311+g(d,p) irc(reverse,calcall,stepsize=3,maxpoints=10)
int=ultrafine  nosym  iop(7/33=1)
HRcc 1 1
```

- **scalefreq**: this keyword allows to specify scaling factor for frequencies, which should thus be specified immediately after the keyword. It will act on calculations performed at level 1 of theory. For possible values see Alecu, I.M.; Zheng, J.; Zhao, Y.; Truhlar, D. G. Computational Thermochemistry: Scale Factor Databases and Scale Factors for Vibrational Frequencies Obtained from Electronic Model Chemistries. J. Chem. Theory Comput. 2010, 6, 2872-2887. Scaling factors can be assigned specifically for each species using the keywords: **scalefreq_ts, scalefreq_reac1, scalefreq_reac2, scalefreq_prod1, scalefreq_prod2, scalefreq_wellr, scalefreq_wellp.**

All keywords may have additional options, some of which are keyword specific. In particular if the word after the electronic structure code is 'restart' it means that in case of failure the electronic structure calculation is restarted for the number of times reported after the 'restart' keyword. Example (restarts 3 times the calculation from the last available geometry if it fails):

level0 g09 restart 3

It is possible to specify module specific theory levels by introducing keyword whose first part is the general module keyword and the second part, separated by an underscore from the first part, is an integer number that identifies each species (1 for reac1, 2 for reac2, 3 for prod1, 4 for prod2, 5 for wellr, 51 for wellr searched using the findgeom algorithm, 6 for wellr, 61 for wellr searched using the findgeom algorithm).
Example (level 0 of theory used only for product 2, supersedes what was written in the level0 keyword):

level0_4 g09
b3lyp/6-31+g(d,p) opt(internal,calcall)
int=ultrafine nosym

3.3.1.2.   Molpro theory files

If the molpro option is specified as the code to use for a specific module, a dedicated file must be written. The file is formatted so that a molpro input file can be easily reconstructed. The file should contain three lines: 'End1', 'End2', and 'End3'. All that is reported above line End1 will be inserted in the molpro input above the z-matrix, which will be inserted by EstokTP, all that is reported between lines 'End1' and 'End2' will be inserted after the z-matrix for closed shell species, while all that is reported between lines 'End2' and 'End3' will be inserted after the z-matrix for open shell species. Additionally, two lines must be present both in the open shell and in the closed shell species. One line that saves the molpro output in molden format: 'put,molden,molpro.molden', and one line where the 'CBSen' variable is initialized to the desired energy value. The CBSen variable defines the high level energy that will be employed in the master equation analysis.
The correspondence between molpro file names and theory.dat keywords is the following:

Level0 molpro → level0_molpro.dat

Level0_ts molpro → level0_ts_molpro.dat
Level1 molpro → level1_molpro.dat
Level1_ts molpro → level1_ts_molpro.dat
hindrotor molpro → onedtau_molpro.dat
hindrotor_ts molpro → onedtau_ts_molpro.dat
symmetry molpro → symm_molpro.dat
symmetry_ts molpro → symm_ts_molpro.dat
hlevel molpro → hl_molpro.dat
hlevel_ts molpro → hl_ts_molpro.dat

Additionally, if a specific theory file is requested for a particular species, this is searched by appending the species number to the molpro file as defined above (see the bottom of section 3.3.1.1 for details about the correspondence between number and species) warning: the general *level0 molpro* statement must still be specified alongside the following:

Level0_1 molpro → level0_reac1_molpro.dat
Level0_2 molpro → level0_reac2_molpro.dat
Level0_3 molpro → level0_prod1_molpro.dat
Level0_4 molpro → level0_prod2_molpro.dat
Level0_5 molpro → level0_wellr_molpro.dat
Level0_51 molpro → level0_wellr_molpro.dat
Level0_6 molpro → level0_wellp_molpro.dat
Level0_61 molpro → level0_wellp_molpro.dat
and so on

Example (file to read when the 'hlevel molpro' keyword is present in the theory.dat file. It performs CCSD(T) calculations followed by MP2 corrections for basis set size for the reac1, reac2, prod1, prod2, wellr, and wellp species if called from the hl_reac1, hl_reac2 and so on modules):

hl_molpro.dat

nosym
End1
!closed shell input
basis=aug-cc-pvtz
hf
ccsd(t)
en_cc=energy
gdirect
basis=aug-cc-pvtz
hf
df-mp2
en_mp2_tz=energy
basis=aug-cc-pvqz
hf
df-mp2
en_mp2_qz=energy
! these lines must be always included in molpro input

```
! CBSen should be defined as desired
! the molden line should be left as it is
put,molden,molpro.molden
CBSen = en_cc+en_mp2_qz-en_mp2_tz
```
```
End2
!open shell input
basis=cc-pvdz
uhf
basis=cc-pvtz
uhf
rhf
basis=aug-cc-pvtz
rhf
uccsd(t)
en_cc = energy
gdirect
basis=aug-cc-pvtz
rhf
df-rmp2
en_mp2_tz=energy
basis=aug-cc-pvqz
rhf
df-rmp2
en_mp2_qz=energy
!these lines must be always included in molpro input
! CBSen should be defined as desired
! the molden line should be left as it is
put,molden,molpro.molden
CBSen = en_cc+en_mp2_qz-en_mp2_tz
```
```
End3
```

3.3.1.3.   Composite theory files

If the composite option is specified for the high level module, a dedicated file must be written. This option allows any number of single point energies be calculated in series, and used to compute/extrapolate the final high level energy using user-provided algebraic expressions. Supported electronic structure codes include Gaussian 09, Gaussian 16, MOLPRO, and MRCC. A detailed log file **./output/hl_composite.log** is produced, and renamed according to the species after termination, such as **./output/reac1_hlcomp.log** for Reac1 and so on. Within this log file, ERROR messages refer to erroneous input in hl_composite.dat and will cause the immediate termination of the module. WARNING messages refer to shortcomings in the execution of a specific step, and will prompt the module to skip the remainder of the current step and proceed to the next task. A job containing any number of warnings will unavoidably terminate with error as the calculation of the final high level energy will not be possible. For any successful step, an energy is appended to a species-specific restart file, such as **./output/reac1_hlcomp.restart** for Reac1 and so on. If this file is present at the beginning of a new composite job on the same species, new

calculations will be avoided if possible, and energies matching nemes read and stored from the restart file itself.

The input file (not case-sensitive) consists of delimited blocks (no specific order required). Each $energy block identifies a single step, which performs a single-point energy calculation.

```
$energy software[,procs,mem] en_name
  keywords to be put before geometry specification, CLOSED SHELL
$end0
  keywords to be put before geometry specification, OPEN SHELL
$end1
  keywords to be put after geometry specification, CLOSED SHELL
$end2
  keywords to be put after geometry specification, OPEN SHELL
$end3
```

Where:
  software - name of the code to use: g09, g16, molpro, or mrcc.
  procs - optional (but mandatory if 'mem' is present), comma-separated, number of processors for this specific step, will replace defaults values.
  mem - optional (but mandatory if 'procs' is present), comma-separated, memory to allocate for the job, will replace default values, units are mandatory; software-specific caveats as per total-memory vs memory-per-process apply.
  en_name - name of the energy generated at this step, to be used in any equation. must be unique; allowed characters include upper- and lowercase letters, numbers, and underscore '_' as the only symbol.
Software-specific keywords not required (ignored if present):
  g09,g16: chk, mem, nprocshared, cpu
  molpro: memory, put
  mrcc: mem, unit, geom

Energies returned by $energy calculation blocks can be combined in arbitrary algebraic expressions to obtain the final high level energy. This is specified by a $solve block, terminated by $end. Any number of explicit expressions can be provided in any order, whereas the final energy must be named just 'e'. Space-separation of operands is not mandatory, supported operations include sum '+', difference '-', product '*', ratio '/', and exponentiation '**'. Dot '.' is used as decimal symbol. Example:

```
$solve
name1 = en5+( en1 + en2 ) / 2
euler= 2.718281828459
name2=name1-0.667*(en3-en4)
e= euler**(-2*name2/600)
$end
```

Software-specific processors and memory details can be provided once by a $param block. The same restrictions described for procs and mem in the $energy block documentation apply. These values will be replaced by step-specific ones if present. Example:

$param software procs mem

Additionally, if a specific theory file is requested for a particular species, this is searched by appending the species number to the molpro file as defined above (see the bottom of section 3.3.1.1 for details about the correspondence between number and species) warning: the general *hlevel composite* statement must still be specified alongside the following:

hlevel_1 composite → hl_reac1_composite.dat
hlevel_2 composite → hl_reac2_composite.dat
hlevel_3 composite → hl_prod1_composite.dat
hlevel_4 composite → hl_prod2_composite.dat
hlevel_5 composite → hl_wellr_composite.dat
hlevel_51 composite → hl_wellr_composite.dat
hlevel_6 composite → hl_wellp_composite.dat
hlevel_61 composite → hl_wellp_composite.dat

3.3.2. Master Equation files

3.3.2.1.   me_head.dat

The input for master equation simulations is reported in the me_head.dat file, which should always be in the **./data** subdirectory. Please look at EstokTP examples for examples on how it should be formatted and at the Mess manual for an explanation of its syntax (http://tcg.cse.anl.gov/papr/codes/mess.html). It is in many ways self – explanatory, so it is suggested to copy it from the EstokTP example subdirectory and modify it according to needs.

4. Setup of an EstokTP job

This section reports a list of the files that must be edited and of the fields that must be filled in in order to set up an EstokTP job.

4.1.1. Abstraction

4.1.2. Addition

4.1.3. Beta-scission

4.1.4. Isomerization

4.1.5. MECP

4.1.6. Barrierless

4.1.6.1. The ref/data directory

This section describes the organization of input files in ref/data directories specific for VRC-TST jobs. These files are eventually used to generate automatically the input for rotd code, described more in detail in sections 4.1.6.2 and 4.1.6.3.

The **ref** directory contains all the information required to setup the calculations properly, such as reactants and products, levels of theory for the calculations, type of recombination sites (symmetry, number of sites..), reference distance for the calculations.
Warnings: NAMES OF THE COORDINATES: the two products must have non-overlapping names in both the atoms and their internal coordinates. For instance, if prod1 has three carbon atoms C1, C2, C3, the carbon atoms of prod2 cannot be indicated by the same names. *Furthermore, the site to which the radical is added should have the same atom number in both reac1 and prod1. Finally, it is advised that the last atoms of reac1 must correspond to those of prod2 and appear in the same order /this should not be necessary).*

The main files in the **ref/data** directory that must be set up properly for a VRC-TST calculations are the following:

ts.dat:

nhind 0	Hindered rotors are useless for VRC-TST (different configurations accounted for by multifaceted spheres). They may be used though for VTST calculations.
isite ji ki 1 2 3	ijk sites: atoms of **prod1** Zmat defining the connectivity of the first atom of prod2 (ibond) in the Zmat of the ts. It is assumed that the first atom of prod2 is the one that will bind prod1.
ibond 5 DISTN	ibond: bond that is broken in the Zmat of reac1/ts. When the TS is created the bond distance is renamed RTS.
jkbond 6 7	jkbond: II and III atom of **prod2** that correspond to those in the Zmat of reac1/ts.
multireference level1 hlevel	Below "multireference" keyword: at which level are multireference calculations to be performed? In this case, for level1 and high level calculations for the MEP. In addition: may be hindrot. Corresponding levels of theory specified in theory.dat.
vrc_info pivrefgeom dist_2.4 npivpoints 1 1 addpivpoints 2 !fragment,atom,refat1,refat2 1 3 1 7 1 9 5 7	Pivrefgeom is the RTS distance (in Angstrom) computed as a reference for the rest of the calculations. The number of pivot points (1 1 in this case) along the breaking bond (two numbers one for each fragment). If 0 is given two pivot points on each side of the fragment, displaced along the pivrefgeom bond, are added by default. The pivot points may be increased with addpivpoints specifying first the fragment and then the connectivity of the pivot point (in this case: 2 pivot points added on fragment 1, atoms 3, 9 numbered as in the Zmat of prod1).
vrc_symm frag1 2 frag2 2	Vrc_symm is the "face" symmetry of each fragment: N of equivalent points to which I can attach the other fragment (ex. HO2 and CH3: both have face symmetry of 2: the bond can be formed on both sides of each fragment)
vrc_dyn_corr 0.9	VRC-TST "dynamic" correction: lowers the final flux to account for potential recrossing.

Symmetryfactor 1	It is recommended to keep the symmetry factor to 1 (external symmetry accounted for automatically at a later stage in vrctst)

theory.dat: besides usual specifications of levels of theory: contains active space specifications for calculations along the MEP. **Warning:** for VRC-TST calculations it is recommended to set **unrestricted** DFT calculations (these are used for a first rough optimization of the points on the MEP) and use the **restart** keyword for gaussian calculations.

``` Level1_31 molpro level1_41 molpro ```	To be introduced only for opt_pr1_vrc and opt_pr2_vrc. Refers to files **level1_prod1vrc_molpro.dat** and **level1_prod2vrc_molpro.dat** and must be consistent with as_level1. Thus if DFT geometries are used for level1, the same chouce must be made here.
``` as_level1 num_bonds 3 num_lps 0 num_states 1 as_hindrot num_bonds 3 num_lps 0 num_states 1 as_hlevel num_bonds 3 num_lps 0 num_states 1 as_vrc num_bonds 1 num_lps 0 num_states 1 ```	Active spaces to be used for level1, hindered rotors, high level, and vrc calculations. Specifications are the number of bonds to include, the number of lone pairs, and the number of states for casscf and rs2 calculations. Level1 includes geometry optimization and frequency calculations for points along the MEP (fixed RTS); hindrot includes only geometry optimization; hlevel and vrc only include energy calculations. As_vrc is normally 2e2o.

Machines: contains the nodes where calculations will be performed at the last step (see rotd setup above).

level1_ts_molpro.dat: read for level1 calculations for points along the MEP. Geometry optimization occurs in two steps so as to minimize computational time. In the first step, the transitional modes between the two fragments are frozen. Their names must therefore be changed according to the Zmat of reac1.

vrc_as.inp: contains instructions for energy corrections. This is copied to the final vrctst calculations directory **vrctst/vrc_tst/sr, vrctst/vrc_tst/lr**.

``` basis=cc-pvdz active space ENERGY {rs2c,ipea=0.25;options,nstati=2} ```	

``
if (iterations.ge.0) then
{rs2c,ipea=0.25;options,nstati=2}
   ENCORR
else
   molpro_energy = 10.0
endif
--- 	
End
``
	Active space: active space specification for vrctst calculations. May be as_level1/as_hlevel/as_vrc as specified in theory.dat, or other manual specifications.
	
	The type of energy calculations (rs2c etc.) must be specified manually.

vrc_tst.inp: input for final vrctst calculations. Contains, among other specifications: temperature and energy grids for the calculations, minimum and maximum number of sampling points to use, face symmetry.

vrc_molpro.inp:

``` MolproExecFile /software/cinetica/molpro2020.1/bin/molpro GeometryPattern GEOMETRY_HERE EnergyPattern molpro_energy BaseFileName vrc_molpro DebugLevel 5 PotentialCorrectionTable /software/cinetica/rotd/pot_corr/libcorrpot.so estoktp_vrc_corr_ 1 ParameterInteger 2 dummy_corr_ 1 End ```	Path to molpto Table with the pattern of the correction potential ParameterInteger: 1 (no correction potential), 2 (with correction potential), 3 (correction potential including penalties, i.e. repulsion on some pivot points, see **pot.dat** in section 4.1.6.3)

4.1.6.2.   Rotd setup and running

Make sure you add to the .bashrc the paths to mpi libraries and rotd program. Something like this should be added:

export PATH=/*path_to_openmpi_libraries*/openmpi-1.10.0-gcc-4.8.2/bin/:$PATH
export LD_LIBRARY_PATH=/*path_to_openmpi_libraries*/openmpi-1.10.0-gcc-4.8.2/lib/:$LD_LIBRARY_PATH
export PATH=/*path_to_rotd_program*/rotd:$PATH

In the file **machines** (placed in the directory of rotd calculations; for automatic estoktp jobs, copied from **data/machines**): list of the nodes where the rotd calculation will be performed. The first node in the list is the one where calculations are organized. All the others are used to run calculations. The present configuration automatically runs molpro calculations on 4 processors, therefore the name of the node shall be repeated as many times as possible.
Example: nodes with 12 processors
node-5-1

```
node-5-1
node-5-1
node-5-1
node-5-10
node-5-10
node-5-10
node-5-15
node-5-15
node-5-15
```

in this case, all the 12 processors of each node are used. Node-5-1 is also used to arrange calculations, therefore it is listed 4 times.

To perform calculations: the program is called from with the command
rotd –f machines tst.inp
To set the process in background and read the output, the command may be modified to
rotd –f machines tst.inp >& stdout.log &

A brief description of the input files to rotd calculations is provided below.

4.1.6.3. Vrc_tst directory

In the **vrctst/vrc_tst/sr** and **vrctst/vrc_tst/lr** directorys, vrctst calculations are run using **rotd** program. The main input files **machines, tst.inp, molpro.inp** are copied from the directory **vrctst/data**, however they can be also modified manually. A brief description of the main files is provided below.

divsurf.inp

| ```
PivotPoints                  6
 Frame 0 0 0 0
 #           ^ ^ ^ ^
 # origin: 1
 # orientation:
 #  3
 #  ^
 #  |y
 #  |  x
 #  4--->2

x1 = -3.45 +d1*sin(p1)*cos(a1)
y1 =  0.00+d1*sin(p1)*sin(a1)
z1 =  0.00 +d1*cos(p1)
x2 = -3.45 -d1*sin(p1)*cos(a1)
y2 =  0.00-d1*sin(p1)*sin(a1)
z2 =  0.00 -d1*cos(p1)
x3 = -2.13 +d1*sin(p2)*cos(a2)
y3 = -2.31+d1*sin(p2)*sin(a2)
z3 = -0.00 +d1*cos(p2)
x4 = -2.13 -d1*sin(p2)*cos(a2)
y4 = -2.31-d1*sin(p2)*sin(a2)
z4 = -0.00 -d1*cos(p2)
x5 = -2.13 +d1*sin(p3)*cos(a3)
y5 =  2.31+d1*sin(p3)*sin(a3)
``` | File automatically generated in Step5 of the calculations described in 1.6.3. It describes the positions of the pivot points in vrctst calculations. **vrctst/vrc_tst/sr/divsurf.inp** considers the sets of pivot points as described in **vrctst/data/ts.dat**, i.e. they are placed close to specific atoms of the associating fragments. In **vrctst/vrc_tst/lr/divsurf.inp** instead the pivot points are placed in the centre of mass of the associating fragments. |

```
z5 =  0.00 +d1*cos(p3)
x6 = -2.13 -d1*sin(p3)*cos(a3)
y6 =  2.31-d1*sin(p3)*sin(a3)
z6 =  0.00 -d1*cos(p3)

 PivotPoints      1
 Frame 0 0 0 0

x7 =  0.00 y7 =  0.00 z7 =  0.00

 # interpointal distances
 #

 Distances

r11 = r-d1/2
r21 = r-d1/2
r31 = r-d1/2
r41 = r-d1/2
r51 = r-d1/2
r61 = r-d1/2

 Conditions 0

 Cycles           8

 r = (8.5, 8.0, 7.5, 7.0, 6.5, 6.0,
5.75, 5.5, 5.0, 4.75, 4.5, 4.3,  4.1)
 d1 = (0.01, 0.3, 0.5)
p1 = ( 168.50)
a1 = ( 176.78)
p2 = ( 180.00)
a2 = (   0.01)
p3 = (   0.00)
a3 = (   0.01)

 EndSurface
```

Warning: when there are more than 9 total pivot points, estoktp writes them automatically as e.g. x1*, .. so the numbers must be fixed manyally

Vrctst calculations are performed varying the distance of the sets of pivot points. In this examples, 8 cycles are run with varying distances r, d1 (units are in bohr) and fixed angles p1, a1, p2, a2, p3, a3 (angles are in degrees).

"Endsurface" marks the end of one surface; it may be followed by another section describing another multifaceted surface.

pot.dat:

```
       7           13
        11
  2.1000000000000001        -14.240182545060520
  2.2000000000000002        -11.357120617488334
  2.3999999999999999         -6.7693068500432219
  2.6000000000000001         -3.4343407575374840
  2.7999999999999998         -1.5224976025395165
  3.0000000000000000         -1.1474333225031330
  3.2000000000000002         -0.32440118499550863
  3.3999999999999999         -9.7584407524067274E-002
  3.6000000000000001          0.38310631995273070
  3.7999999999999998          0.56537373496780674
  4.0000000000000000          0.69328270749963394
2 ! sites with rep pot on frag 1
3 ! I site
9 ! II site
```

File automatically generated in Step5 of the calculations described in 1.6.3.

Correction potential with respect to vrctst energy computed according to the level of theory of **vrc_molpro.tml**. The correction is in kcal/mol and is indicated at different distances (in Angstrom).

Sites with additional repulsive potential are indicated at the end of the file. These must be added

	manually and correspond to those indicated in **vrctst/data/ts.dat**

mc_flux.inp: specification of input files to assemble the final flux.

structure.inp: geometries of prod1 and prod2 (the two associating fragments). The order of the atoms in each fragment is that specified in prod1 and prod2 zmatrix in **data/prod1.dat, data/prod2.dat**.

vrc_molpro.tml: input file containing the level of theory for the energy calculations. the level of theory is copied from **data/vrc_as.inp**.

flux.out: file containing the temperature-dependent fluxes obtained from vrctst calculations. The flux is provided for each pivot-points configuration specified in **divsurf.inp**. For each surface, the minimal energy is provided with and without the correction potential indicated in **pot.dat**. The structures corresponding to the minimum flux for each surface are found at the end of the file.

5. Output

The output of EStokTP calculations is stored in several files and subdirectories. The log files of an EStokTP calculation (the ones to control to check for error or warning messages) are the ./output/estoktp.out file and several standard output messages (screen). It is suggested to re-direct the standard output to a log file (we usually call this file estoktp.log) when running EStokTP, which should therefore be called as 'estoktp.x > estoktp.log). Estoktp.log reports all the unix shell commands called by EStokTP together with some error messages. The output/estoktp.out reports several calculation log messages, which may be useful to track the progression of the code and should be checked in case of an unexpected termination.

5.1. EStokTP Outputs
EStokTP saves as much as possible of the data produced from the electronic structure calculations. Briefly, calculation logs of each module are reported in the **./output** subdirectory, geometries in the **./geoms** subdirectory, files used to create the master equation input in the **./me_files** subdirectory, and logs of high level calculations in the **./hl_logs** subdirectory. More details are reported in dedicated subsections.

5.1.1. Geometries
Optimized geometries are reported in the **./geoms**, **./ouput**, and **./me_files** subdirectories. Level 0 geometries are reported in the **./geoms** subdirectory, where they are named $species_$numb.xyz, where $species may be reac1, reac2, prod1, prod2, wellr, wellp, or ts, and $numb refers to the progressive number of the structure optimized using the Monte Carlo protocol. The optimized geometries are numbered following the order on which they are found, not the energy. The structures are reported in xyz format, with the level0 energy of the structure reported in the second line. The EStokTP log file of a level0 geometry optimization is reported in the **./output**

subdirectory as $species.out, which describes the progress of the Monte Carlo sampling procedure and reports which structure has been selected for successive level1 calculations. The z-matrix parameters of each structure are reported in the ./output subdirectory as $species_opt_$numb.out. The minimum energy structure z-matrix parameters are reported in the **./output** subdirectory as $species_opt.out. This file is the reference file for all geometry calculations, so that it is overwritten by level1 calculations and used in high level calculations.

Level1 optimized structures are reported in the **./geoms** subdirectory as $species_l1.xyz, where $species may be reac1, reac2, prod1, prod2, wellr, wellp, or tsgta (for the transition state structure). The level1 module generates also the geometry structures that will be used by the master equation solver to compute inertia moments. These structures are saved in the **./me_files** subdirectory as $species_ge.me files ($species may be reac1, reac2, prod1, prod2, wellr, wellp, or ts).

Log files produced by the electronic structure code for level1 and level0 calculations are saved as well, with the same name as that of the optimized xyz structure, but with the .log extension ($species_$numb.log and $species_l1.log). In case molpro is used, the molden file generated is also saved in the **./geoms** subdirectory using the molden extension ($species_$numb.molden and $species_l1.molden).

5.1.2. Frequencies

Frequencies are computed in the level 1 module and saved in the **./me_files** subdirectory as $species_fr.me. In addition, level1 calculations by default generate a log file with the Cartesian Hessian, which is stored in the **./output** subdirectory as $species_fcmat.log (where $species may be reac1, reac2, prod1, prod2, wellr, wellp, or ts). The list of frequencies to be used for master equation calculations is updated after a call to the hindered rotor module if hindered rotor treatment has been requested in the species input file. This is done by projecting out the torsional motions from the Hessian, orthonormalizing and diagonalizing it as suggested by Green et al. (Sharma et al., J. Phys. Chem. A 2010, 114, 5689–5701). After projection, the updated list of frequencies is stored in the **./me_files** subdirectory as $species_fr.me, while the unprojected frequencies are saved as $species_unpfr.me. In case of need, an animation of the frequencies is possible post-processing the electronic structure calculation log file (the .molden file if molpro was used) reported in the **./geoms** subdirectory, using a visualization program such as, for example, molden.

5.1.3. Hindered Rotor

The log file of hindered rotor calculations is reported in the **./output** subdirectory as $species_hr.out. It describes the progress of the calculation and the adopted algorithm. The hindered rotor potential together with details on the rotating top is reported in the **./me_files** subdirectory as $species_hr.me. If a multi dimensional hindered rotor treatment is requested, the multi dimensional hindered rotor PES and the frequencies at each point along the PES (if requested) are stored in the **./me_files** subdirectory in the $species_$dimension_$number.dat file, with $dimension being 1dhr, 2dhr or 3dhr and $number being a progressive number starting from 01 (at present only one multidimensional hindered rotor is supported). For 1D hindered rotors geometries of point taken on the PES are stored in the **./hr_geoms** subdirectory. Structures are saved in xyz format as geom_isp$numb1_hr$numb2_hpt$numb3.xyz, where $numb1 refers to the

species (01 for reac1, 02 for reac2, 03 for prod1, 04 for prod2, 05 for wellr, 06 for wellp, 00 for transition state), $numb2 refers to the hindered rotor progressive number (as numbered in the specie input file) and $numb3 is the number of the point taken along the hindered rotor PES. These files are used by the symmetry module to determine the optical symmetry number for internal rotations.

5.1.4. IRC

Files used in internal reaction coordinate calculations (for now supported only for Gaussian) are stored in the **./irc_files** subdirectory. In particular irc_g09f.log and irc_g09b.log are the log files of the Gaussian forward and backward IRC scan. Input files for projection of hindered rotor frequencies are stored as well in this subdirectory as input$numb. The main output of IRC calculations is reported in the variational.me file, which is stored in the **./me_files** subdirectory. The log file of irc calculations is irc.out and is reported in the **./ouptut** subdirectory.

5.1.5. Energies

The electronic structure log files of high level calculations are stored in the **./hl_logs** subdirectory as $species_$code.out, where $code could be either molpro or g09. The energy (In Hartree) determined through high level calculations is saved in the **./me_files** subdirectory as $species_en.me. The EStokTP log file of high level calculations is very terse, as the electronic structure log files contain most of the information necessary to check the progress and success of these calculations.

5.1.6. Multi Dimensional Tunneling

File used to perform multidimensional tunneling calculations are stored in the **./md_tunn** subdirectory. They are produced whenever the IRC module is called. The trajec.xyz file reports all the structures along the irc in xyz format and can be viewed using a structure visualization software, such as molden or vmd. The reduced mass computed through the SCT formalism is stored in the mueff.txt file (last column) in atomic units, while the vibrationally adiabatic energy is reported in the VaG.txt file. The SCT tunneling coefficients are reported in the last rows of the mdtunn.out file, in the kSCT column. The imaginary action integral file that is used by the master equation solver (Mess) to determine tunneling contributions is stored in the imactint.dat file, which is saved in the main directory of the EStokTP calculation and in the **./me_files** subdirectory.

5.2. Rate Output

The master equation input file is created by the kTP module, which then runs the master equation calculations calling the Mess program. The Mess output is composed of two parts, the rate.out file, which contains all the temperature and pressure dependent rate constants, and the me_ktp.log file, which is the log file of the Mess code. A detailed description of the Mess output is reported in the Mess manual at http://tcg.cse.anl.gov/papr/codes/mess.html.

6. Examples

Several examples of uses of EStokTP are included in the distribution in the **./examples** subdirectory. Only the input data files are reported in the distribution, while the log files for all the examples are distributed as a separate tar file. With reference to the subdirectory path, the examples are:
- Wells/C3H8/c3h8_1dhr: minimum energy search for propane with 1D hindered rotor treatment up to the production of the Mess block.
- Wells/C3H8/c3h8_2dhr: minimum energy search for propane with 2D hindered rotor treatment up to the production of the Mess block.
- Wells/C3H8/c3h8_2dhr_fr: minimum energy search for propane with 2D hindered rotor treatment and evaluation of frequencies at each point of the hindered rotor PES up to the production of the Mess block.
- Abstraction/ c3h8_h_n3h7_h2_1TS: rate constant evaluation using the 1TS model for H abstraction from propane (see section 1.3.2).
- Abstraction/ c3h8_h_n3h7_h2_3TS: rate constant evaluation using the 3TS model for H abstraction from propane (see section 1.3.2).
- Addition/c2h4_h_1TS: rate constant for addition of H to ethylene using the 1TS model (see section 1.3.3).
- Addition/c2h4_h_2TS: rate constant for addition of H to ethylene using the 2TS model (see section 1.3.3).
- Betascission/nc3h7_c3h6_h; rate constant calculation for betascission of n-propyl to propene and H (see section 1.3.4).
- Isomerization/nc3h7_nc3h7: rate constant calculation for n-propyl isomerization (see section 1.3.5). The product well (nc3h7) is determined using the findgeom keyword, so that no z-matrix has to be specified in the input.
- Isomerization/nc3h7_nc3h7_well5: rate constant calculation for n-propyl isomerization (see section 1.3.5). The product well (nc3h7) is specified explicitly in the input as wellp.

7. Advices/Suggestions/Troubleshooting

- Highly endothermic reactions must be studied from the endothermic side.
- Search of van der Waals wells with H atoms may be problematic, it may be skipped without loosing much accuracy.

8. Index