PILGRIM v1.0

A thermal rate constant calculator and a chemical kinetics simulator

David Ferro-Costas[1]

Donald G. Truhlar[2]

Antonio Fernández-Ramos[1]

(1) Universidade de Santiago de Compostela (SPAIN)

(2) University of Minnesota (USA)

Contents

III Part Three - Pilgrim input files

IV Part Four - Bibliography

Part One - Introduction to Pilgrim

1. About Pilgrim

1.1 Introduction

Pilgrim v1.0 is an user-friendly program written in Python 2.7. It was designed to use direct-dynamics to calculate thermal rate constants of chemical reactions and to simulate chemical kinetics mechanisms.

For reaction processes with many elementary steps, each of these steps can be calculated using conventional transition state theory (TST) or variational transition state theory (VTST). In this version, Pilgrim can calculate thermal rate constants with the canonical version of the variational transition state theory (CVT), which requires the calculation of the minimum energy path (MEP) associated to each elementary step. Moreover, multi-dimensional quantum effects can be incorporated through the small-curvature tunneling approximation (SCT). The above methodologies are available for reactions involving a single structure and for reactions involving flexible molecules with multiple conformations. Specifically, for systems with many conformers the program can evaluate each of the elementary reactions by multi-path canonical variational transition state theory (MP-CVT) or multi-structural VTST (MS-VTST). Torsional anharmonicity can be also incorporated through the MSTor and Q2DTor programs.

Pilgrim also performs dual-level calculations automatically. First, low-level calculations are carried out for the reaction of interest and second, single-point energy calculations of the reactants, transition state, points along the MEP and products are performed at a higher level. The low-level calculations are corrected with the high-level single point energies using the interpolated single-point energies (ISPE) algorithm.

Once all the rate constants of the chemical processes of interest are known, by means of their calculation using Pilgrim or by using an analytical expression, it is possible to simulate the whole process using kinetic Monte Carlo (KMC). This algorithm allows performing a kinetics simulation and monitoring the evolution of each chemical species with time, as well as providing its chemical yield.

1.2 How to obtain Pilgrim?

In order to get Pilgrim, the user should visit our GitHut page (`https://github.com/daferro/Pilgrim`) and download the latest version.

1.3 Terms of use

Pilgrim is free software under license MIT. See `https://github.com/daferro/Pilgrim` for details about the license.

1.4 About this manual

This manual is organized as follows:

- **Chapter 2** presents an overview about how to use Pilgrim. For users who are familiar with VTST and with electronic structure programs, this chapter contains most of the information they need to run the program. Occasionally, they may need to consult other chapters to find a more detailed explanation about a given particular aspect.
- **Chapters 3-5** describe some worked examples (WEs) that allow the user to explore some capabilities of the program.
- **Chapter 6** describes in detail the input files of Pilgrim.
- **Chapter 7** lists some important references.

1.5 Pilgrim software contents

This version of Pilgrim is provided in a tar.gz file which can be uncompressed using:

```
 tar -zxvf pilgrim_version.tar.gz
```

A new directory is created with the name `Pilgrim_v1.0/`. Inside the directory there are the `README` file and three subdirectories:

- `src`: Contains the source code.
- `docs`: Includes this manual.
- `tests`: Contains two directories:
    - `tests_to_run`: Includes the electronic structure files (ESFILs) with the stationary points needed to run each of the WEs using Pilgrim. The ESFILs are the starting point to run the tests. All the tests were designed to be run at the HF/STO-3G level or at the PM6//HF/STO-3G dual level, so they can be performed in a short period of time. How to run these WEs with Pilgrim is discussed in Chapters 3-5. The tests include the following reactions:
        - Directory `tests/tests_out/FA/`: Isomerization of formic acid.
        - Directory `tests/tests_out/NH3/`: The inversion of ammonia.
        - Directory `tests/tests_out/EtOH/`: Several hydrogen abstraction reactions involving ethanol.

  - tests_out/: Includes the ‘already run’ WEs, so they can be compared with those obtained by running the tests of tests_to_run/.

1.6 Software requirements

Before running Pilgrim the user should have installed the following software:

- Python 2.7 and modules: cmath, fcntl, glob, math, matplotlib, multiprocessing, numpy, os, random, scipy, sys, time. If any of these modules is missing Pilgrim displays an error message indicating which module needs to be installed.
- Electronic structure software (ESSO), in particular, *Gaussian* and/or Orca

In order to interact with *Gaussian* and/or Orca, Pilgrim needs to know the location of some executable files. Such information is obtained from the following environment variables, which have to be defined and exported by the user in their *.bashrc* file:

- Environment variables for Gaussian software:
  - GauExe: with the path to the Gaussian executable
  - GauFchk: with the path to the formchk tool
- Environment variable for Orca software:
  - OrcaExe: with the path to the Orca executable

For example:

.bashrc: exporting environment variables

```
export GauExe="/home/programs/g09/g09"
export GauFchk="/home/programs/g09/formchk"
export OrcaExe="/home/programs/orca_4_0_1_2/orca"
```

1.7 Pilgrim Facebook web page

We recommend to follow our Facebook web page for updates, questions and other program information

- Pilgrim v1.0: https://www.facebook.com/pilgrimSoftware/

1.8 How to cite Pilgrim?

If you are using Pilgrim, please cite it using the following reference:

- D. Ferro-Costas, D.G. Truhlar, A. Fernández-Ramos, *Journal Name* **year**, *vol*, page

For references about key general methods and about specific methods used in Pilgrim see Chapter 7.

2. Getting started with Pilgrim

The execution of Pilgrim can be divided into three different stages:

- Setup.
- Input files preparation.
- Calculations.

In this chapter, we explain how to proceed at each stage.

2.1 Setup

This section describes how to organize the electronic structure files (ESFILs) for the reaction mechanism object of study. At the moment Pilgrim can work with the following electronic structure software (ESSO) : *Gaussian 03*, *Gaussian 09*, *Gaussian 16*, Orca 3.x, Orca 4.0 and Orca 4.1.

2.1.1 Organization of the electronic structure files

Prior to work with Pilgrim, the user should have at hand the ESFILs of all the stationary points relevant to the reaction mechanism. The reaction mechanism may consist of one or several elementary chemical reactions. Each ESFIL should contain the optimized geometry and Hessian of a given stationary point, and they have to be organized as indicated in Figure 2.1 (blue boxes) and outlined below:

- Create a directory (hereafter `MDIR/`) to study your reaction mechanism using Pilgrim.
- Create a subdirectory called `UDATA/` inside `MDIR/`.
- Create a subdirectory inside `UDATA/` for each of the species that participate in the chemical mechanism. Hereafter the subdirectories inside `UDATA/` will have the generic name `spname/`and the `spname` label is the name by which Pilgrim recognizes that stationary point

species.[i] Therefore use meaningful names for those directories.[ii]

- Place the ESFILs in the corresponding `spname/` directory. The allowed formats are listed in the green boxes of Figure 2.1. If a given species has several conformational isomers[iii] they should be included in the same `spname/` directory.
  Which conformers should be included? All the conformers that are distinguishable except in the case of conformational enantiomers that only one of them should be included.[iv]
- Pilgrim can incorporate torsional anharmonicity calculated with MSTor or Q2DTor programs. The output files from these programs should be located in a directory hanging from `MDIR/` and called `ANHAR/`. All the torsional anharmonicity files should be placed inside this directory.[v]

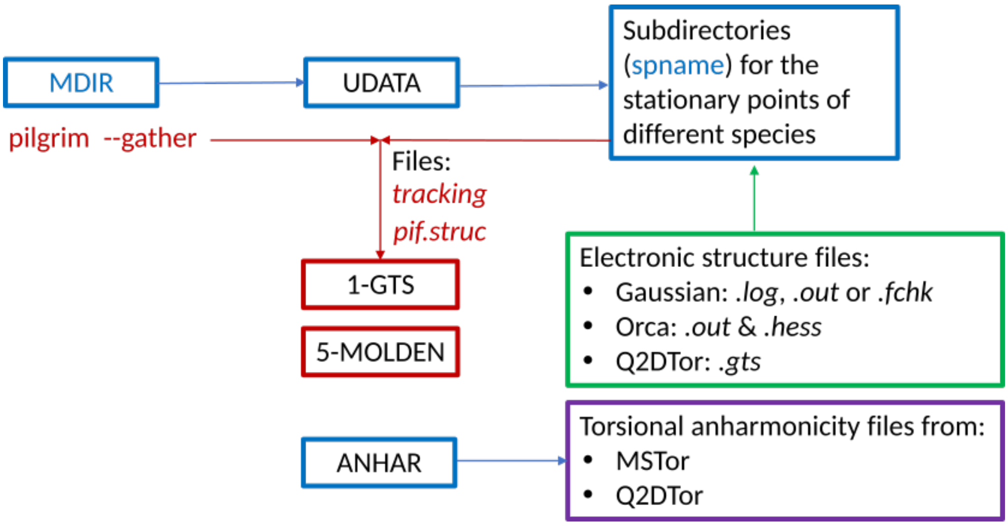
MDIR
UDATA
Subdirectories (spname) for the stationary points of different species
pilgrim --gather
Files:
tracking
pif.struc
1-GTS
5-MOLDEN
Electronic structure files:
• Gaussian: .log, .out or .fchk
• Orca: .out & .hess
• Q2DTor: .gts
ANHAR
Torsional anharmonicity files from:
• MSTor
• Q2DTor

Figure 2.1: This scheme shows the directories tree that should be created by the user (blue boxes) and the directories that are created by the program (red boxes) after execution with `--gather`. At this stage the user can also incorporate the output files of the MSTor or Q2DTor programs into the `ANHAR/` directory.

2.1.2 The `--gather` option

Once the ESFILs are organized as previously described, execute Pilgrim from `MDIR/` using:

```
   pilgrim.py --gather
```

Pilgrim creates two directories inside `MDIR/` : (i) `1-GTS/` contains the *.gts* files (for internal use by Pilgrim) of all electronic structure files of the stationary points included in the subdirectories

[i] For species with a single stationary point the ESFIL may hang directly from UDATA/, but for species with several conformations the creation of a directory is compulsory. For the former the `spname` label is the name of the ESFIL without extension.

[ii] Names should start with a letter; the rest of characters may be underscores (_), letters or numbers.

[iii] Also called rotamers, torsional conformers or symply conformers. Conformational isomers are species that can interconvert among them by internal rotations.

[iv] For instance, ethanol has three conformations, two *gauche* and one *anti*; however, the two gauche conformations are conformational enantiomers and therefore only one of them should be included. Notice that a couple of conformational enantiomers can be describe by a unique ESFIL; therefore, only one of them has to be specified and the other is easily incorporated later on when the input files of Pilgrim are created.

[v] Anharmonicity output files **are not** recognized if placed somewhere else.

of `UDATA/`; (ii) `5-MOLDEN/` includes files for visualizing with Molden. The program also creates the files *tracking* and *pif.struc*. The former correlates the ESFILs of `UDATA/` with the *.gts* files of `1-GTS/` and the latter provides information about each structure. The *.gts* files are labeled as: *spname.idx.gts*, where *spname* indicates the name of the species and *idx* is the numbering of the torsional conformation for that species (starting at 001) and sorted by increasing electronic energy. If there is only one conformation the *.gts* file inside `1-GTS/` will be called *spname.001.gts*

File *pif.struc* is the only one that may need some editing by the user as it contains, for instance, the scaling factor for vibrational frequencies, the weight of a given conformation,[vi]etc. For more information, see Chapter 6.

2.1.3 About labels

The *pif.struc* file is formed by a set of blocks (see Chapter 6), i.e., one block for each of the species defined in `UDATA/` and labeled by its corresponding `spname` (which was previously defined by the user). However, hereafter the generic name `ctcsp` will be used instead of `spname` for the labels. In the same manner, the generic label `ctcts` will be used for blocks associated with transition states.[vii] Notice that the `ctcsp` and `ctcts` labels are also used by Pilgrim as 'targets' during the execution stage.

Pilgrim automatically assigns all the `spname` labels as `ctcsp` labels, but intervention by the user is required to study isotopically substituted reactions. In that case the ESFILs associated with a given `spname` are valid for the root species, as well as for any isotopically modified species, but new blocks with their corresponding labels have to be added to incorporate the new species. In this case the `ctcsp` labels include those of the root species plus the ones of the isotopically substituted species.

For example, to study the reaction EtOH + H→ EtO + H_2 the `spname` labels were named as: `EtOH` for ethanol, `H` for hydrogen atom, `EtO` for the EtO radical, `H2` for the H_2 molecule and `TSc` for the hydrogen abstraction transition state. The `ctcsp` labels used by Pilgrim in *pif.struc* coincided with the `spname` labels. Within the `ctcsp` labels, the only `ctcts` label is `TSc`.

However, to study also the isotopically substituted reaction EtOD + D → EtO + D_2, new blocks and labels should be added to incorporate the isotopic species.[viii] These new `ctcsp` labels are: `EtOD`, `D`, `D2` and `TScOD`.[ix]

2.2 Input files preparation

The generation of Pilgrim input files can be carried out using:

```
pilgrim.py --input
```

which displays an interactive and self-explanatory menu. This menu allows the user to generate all

[vi] Conformational isomers that have another enantiomer, should be added at this stage

[vii] Therefore the label of a transition state is at the same time a `ctcsp` label and a `ctcts` label.

[viii] Notice that the `spname` labels are the same as for the previous reaction.

[ix] There is also a new `ctcts` label: `TScOD`.

Table 2.1: Description of the variables of Pilgrim inside the input menu and the input files modified or generated by each option.

Variable	Information	Pilgrim Input File
struc	Structures	*pif.struc*
isomass	Isotopic masses	*pif.struc*
temp	Temperatures (in K)	*pif.temp*
chem	Chemical reactions	*pif.chem*
path	MEP parameters	*pif.path* & *pif.calcs*
kmc	Chemical kinetics simulation	*pif.kmc*
dlevel	Dual-level calculations	*pif.dlevel* & *pif.calcs*

the input files.[x] The variables inside the menu, their meaning and the generated Pilgrim input files (*pifs*) are displayed in Table 2.1. Although this menu can generate each of the files needed to run Pilgrim, some of them may need further editing outside the menu to adjust some default options. For a detailed description of each input file, the user should turn to Chapter 6.

When the interactive menu is initialized, it displays the status of every possible input file. The first time that we enter the menu it looks like:

```
   Status of input files

      ---------------------------------
             input file       | status
      ---------------------------------
       #1 : pif.struc         |    1
       #2 : pif.temp          |   -1
       #3 : pif.path          |   -1
       #4 : pif.calcs         |   -1
       #5 : pif.chem          |   -1
       #6 : pif.kmc           |   -1
       #7 : pif.dlevel        |   -1
      ---------------------------------
      status = -1 ==> file does not exist
      status =  0 ==> file exists but it is empty
      status =  1 ==> file exists and contains data
```

Notice that none of the files exists with the exception of *pif.struc*. The menu also displays:

```
===========================================================
 ||  EXECUTING PILGRIM WITH --input                      ||
 ===========================================================


  There are several variables ($var) and commands ($cmd) available
  in this interactive menu. The command line should have the
  following syntax:

      > $cmd $var [$values]

  where the square brackets indicate that the $cmd-$var combination
  may require the specification of values ($values)
  For more information, use the 'help' command on each
```

[x] With the exception of *pif.struc* which was generated in the previous stage. However, the interactive menu can modify it.

```
variable inside the interactive menu.

List of commands ($cmd) and variables ($var):

  ----------------------------------------------------------------
  $cmd\$var | struc | isomass | temp | chem | path | kmc | dlevel
  ----------------------------------------------------------------
  help      | x     | x       |  x   |  x   |  x   |  x  |  x
  ls        | x     | x       |  x   |  x   |  x   |  x  |  x
  add       |       | x+      |  x+  |  x+  |  x+  |  x  |  x
  mod       | x+    |         |      |      |  x+  |     |
  rm        | x+    | x       |  x+  |  x+  |  x+  |     |  x
  ----------------------------------------------------------------
   x: the combination $cmd $var is available
   +: the combination $cmd $var requires $values

Information about variables ($var):

  -----------------------------------------------------------------
   $var    | affects the input file| which contains...
  -----------------------------------------------------------------
   struc   | pif.struc             | the structures & isot. masses
   isomass | pif.struc             |          "
   temp    | pif.temp              | the temperatures
   chem    | pif.chem              | the chemical reactions
   path    | pif.path & pif.calcs  | the minimum energy paths
   kmc     | pif.kmc               | the variables in the KMC
   dlevel  | pif.dlevel            | the structures for high-level
  -----------------------------------------------------------------

To go back to an upper level in the menu or to exit, use
one of the next strings: end / .. / exit
```

Through the help command inside the menu, the information about a given variable can be displayed. We highly recommend to read the help message of each of the variables to learn how to proceed.

2.3 Calculations

Pilgrim was designed to be executed in steps, and some of the options of the program follow a given order. Within this step-by-step philosophy, each execution carries out a different task, whose result can be checked by the user before proceeding to another step. The general command line for executing Pilgrim is:

```
pilgrim --option [target[.idx]] [--dlevel] [--software ESSO]
```

2.3.1 Options

The available options in Pilgrim are shown in Table 2.2. The `--software` option can be only combined with `--path` (calculation of the minimum energy path) and with `--hlcalc` (high-level single point calculation). By default, the ESSO is set to *Gaussian* (`--software gaussian`). Therefore, this keyword is only required when using Orca (`--software orca`). The `--dlevel` option should be only specified to consider the dual-level calculations (previous high-level calculations are required).

Table 2.2: Description of the options of Pilgrim; the order of execution of the main options is also indicated.

Order of execution, `--option` and explanation		
1	`--pfn`	Calculates the partition functions at the stationary points
2	`--path`	Calculates the MEP & VTST coefficients
3	`--rcons`	Calculates the thermal rate constants
4	`--kmc`	Performs a kinetic Monte Carlo simulation
Options for special actions		
`--dlevel`		Dual-level calculations and interpolation via ISPE
`--software`		Selects ESSO [`gaussian` (default) or `orca`]; it goes together with `--path or --hlcalc`; if omitted, `gaussian` is selected by default;
Additional options and tools		
`--ics`		Generates internal coordinates
`--hlcalc`		Performs high-level calculations needed for dual-level
`--fit`		Fits the thermal rate constants to prescribed analytical expressions
`--plot`		Generates diverse plots
Information options		
`--help`		Displays help messages; `--help option` can also be used
`--ls`		Lists the species defined in *pif.struc*
`--version`		Displays the version of the program

2.3.2 Targets

When `target` is omitted from the command line, `--option` is applied to *all* possible targets. Different options have different targets as indicated in Table 2.3. For instance, option `--path` can have as target only stationary points associated to transition states (*i.e.* `ctcts`). If a `ctcts` contains more than one conformation, it is possible to calculate the MEP of an individual TS structure using `ctcts.idx`, where `idx` is the numerical index of the conformation (from `001` to `999`) inside a block in *pif.struc*. In the case of the calculation of the rate constants, the option `--rcons` may have as target the name of a elementary chemical reaction that was defined in the input with the variable *chem*, and with the generic label *chemname*.

Table 2.3: List of the targets allowed by the different options. The default (the target is omitted) is also indicated.

`--option`	default	allowed `target`
`--pfn`	*all* `ctcsp`	`ctcsp`
`--path`	*all* `ctcts`	`ctcts[.idx]`
`--rcons`	*all* `chemname`	`chemname`
`--kmc`	—	—
`--ics`	*all* `ctcts`	`ctcts[.idx]`
`--hlcalc`	*all* `ctcsp`	`ctcsp[.idx]`
`--fit`	*all* `chemname`	`chemname`

2.3.3 Output files

A list of all the directories generated by Pilgrim is given in Table 2.4. The output files generated after the execution with a given option are indicated in Table 2.5.

Table 2.4: Directories created by Pilgrim during the execution of the program

Directory	Content
`1-GTS/`	The *.gts* files generated by Pilgrim from the ESSO output files
`2-PLG_DATA/`	Data files (Pilgrim's internal use)
`3-PLG_OUTPUT/`	Output files
`4-PLG_RST/`	Restart files for MEP calculations (Pilgrim's internal use)
`5-MOLDEN/`	Files in Molden and xyz format
`6-PLOTFILES/`	Data for plotting

Table 2.5: Output files generated in the `3-PLG_OUTPUT/` directory when executing Pilgrim with different options according to the selected target. If the options are run together with `--dlevel`, the word *slevel* is replaced by *dlevel* in the output file name.

--option	target	Output files
`--pfn`	*ctcsp*	*pfn.ctcsp.slevel.txt*
`--path`	*ctcts.idx*	*path.ctcts.idx.slevel.txt*
`--rcons`	*chemname*	*rcons.chemname.slevel.txt*
`--kmc`	—	*kmc.slevel.txt*

Part Two - Worked examples

3. Isomerization of formic acid

3.1 About this system

This Chapter describes the isomerization process between the two isomers of formic acid (FA): *Z*-FA and *E*-FA. The concerned reaction is:

$$Z\text{−HCOOH} \rightleftarrows E\text{−HCOOH}$$

and we want to estimate the rate constant of the process at 300, 400 and 500 K.

The four WEs designed for this reaction include the following methodologies:

- WE01: TST
- WE02: CVT/SCT (I)[i]
- WE03: CVT/SCT (II)[ii]
- WE04: CVT/SCT with an isotopic substitution

- ESSO: *Gaussian 09*
- Level of calculation: HF/STO-3G
- Working temperatures (in K): 300, 400 and 500

3.2 WE01: TST

Study the isomerization reaction using conventional TST.

Steps:

- Create the directory `WE01/` inside tests_to_run/FA (hereafter, the working directory for this system).

[i] The projection of the frequencies along the path is carried out in Cartesian coordinates.
[ii] The projection of the frequencies along the path is carried out in internal coordinates.

- Copy the directory UDATA/ from tests_to_run/FA to WE01/. Note that in this case no subdirectories are needed inside UDATA/ because none of the structures has additional conformational isomers (in fact the reaction is a conformational isomerization itself). The ESFILs are: Z.fchk, E.fchk and ZE_TS.fchk.
- Run 'pilgrim --gather'. This creates the files *tracking* and *pif.struc*, as well as the directories 1-GTS/ and 5-MOLDEN/ that contain the *gts* and *molden* files, respectively. The latter can be visualized with Molden.
- Use the interactive menu of Pilgrim by typing 'pilgrim --input' and define the temperatures and the chemical reaction with:

  ```
  > add temp 300 400 500
  > add chem FA_ISO : Z --> ZE_TS --> E
  ```

  After exiting the menu, the files *pif.temp* and *pif.chem* are created. In the input menu we have labeled the reaction as FA_ISO. Notice that the chemical species have the same name as the ESFILs of UDATA/ without the extension. The labels E, Z and ZE_TS are the spname labels of the species.
- Run 'pilgrim --pfn' to calculate the partition functions. Once the program is executed the directory 3-PLG_OUTPUT/ is created. Inside that directory the user can find the files *pfn.E.slevel.txt*, *pfn.Z.slevel.txt* and *pfn.ZE_TS.slevel.txt*, which contain information about the energetics and partition functions of the species labeled as E, Z and ZE_TS, respectively.
- Run 'pilgrim --rcons FA_ISO' or simply 'pilgrim --rcons'. The forward and backward TST rate constants, as well as the equilibrium constants, are calculated and printed in *rcons.FA_ISO.slevel.txt*, inside 3-PLG_OUTPUT/.

3.3 WE02: CVT/SCT (I)

Calculate the CVT/SCT thermal rate constants for the isomerization process. The projection of the frequencies along the path is carried out in Cartesian coordinates. The MEP is initially calculated between -1.0 and 1.0 a_0 with a stepsize of 0.01 a_0 and Hessian updates every 10 steps. We require a convergence of 0.1% for the SCT transmission coefficient at the lowest temperature.[iii]

Steps:

- All the information obtained in WE01 can be used to obtain the CVT/SCT rate constants, so the first step is to copy the directory WE01/ as WE02/.
- Run 'pilgrim --input' inside WE02/ to create the *pif.path*. Change the values of sbw and sfw to -1.0 a_0 and 1.0 a_0, respectively. Include also the keywords sctmns 10 and scterr 0.1; they are needed to obtain converged results within 0.1% for the SCT transmission coefficients.

  ```
  > add path *
  >> sbw = -1.0
  >> sfw = +1.0
  >> sctmns = 10
  >> scterr = 0.1
  >> ..
  ```

[iii] Notice that the keywords sctmns and scterr override the values of sbw and sbw, and the MEP continues or until convergence is achieved or until the number of additional Hessians calculated (after the initial specification) is larger than sctmns.

After exiting the menu, the files *pif.path* and *pif.calcs* are created.

- The file *pif.calcs* does not need further editing because the level by default is HF/STO-3G.
- Run 'pilgrim --path ZE_TS' or simply 'pilgrim --path' to evaluate the variational Γ^{CVT} and tunneling transmission $\kappa^{\text{CVT/SCT}}$ coefficients. The file *path.ZE_TS.001.slevel.txt* is written in 3-PLG_OUTPUT/. A restart file called ZE_TS.001.rst containing the information along the MEP is stored in the 4-PLG_RST/ directory.
- Run 'pilgrim --rcons FA_ISO' or simply 'pilgrim --rcons' to calculate the CVT/SCT thermal rate constants.

3.4 WE03: CVT/SCT (II)

Same as WE02 but in this case the frequencies along the MEP are projected using redundant internal coordinates.

Steps:

- Copy the directory WE02/ as WE03/, so the restart file *ZE_TS.001.rst* inside the 4-PLG_RST/ directory can be used in this new run. Notice that the MEP does not depend on the coordinates defined to project the normal-mode frequencies along the path.
- The redundant internal coordinates are generated automatically by running 'pilgrim --ics' inside the WE03/ folder. The program looks for a suitable set of internal coordinates that correctly describes the TS frequencies. The resulting coordinates are incorporated it into the *pif.struc* file.
- Edit the *pif.path* file and add useics yes. By default the program always use internal coordinates if a set of them is provided in the *pif.struc* file, so this line is not mandatory.
- Run 'pilgrim --path ZE_TS' or simply 'pilgrim --path'. The program restarts the MEP stored in the 4-PLG_RST/ directory. This time the vibrationally adiabatic potential is calculated using projected frequencies obtained from the previously obtained internal coordinates set.
- Run 'pilgrim --rcons FA_ISO' to obtain the CVT/SCT thermal rate constants. They are printed in the file *rcons-FA_ISO.slevel.txt*.

3.5 WE04: CVT/SCT with an isotopic substitution

Same as WE03, but replacing the hydrogen that moves from *Z* to *E* during the isomerization by a deuterium atom. The ESFILs are the same as for previous runs and the user can reuse the files saved previously (for instance the ones from directory WE01/), but the program has to be executed from the beginning to include the isotopic substitution. Notice that the MEP stored in WE02/ or in WE03/ cannot be restarted because the MEP is mass-dependent.

Steps:

- Copy the directory WE01/ as WE04/.

- Initialize the interactive menu with ‘`pilgrim --input`’ and include the isotopic substitutions by typing

  ```
  > mod struc Z
  >> iso = D(4)
  >> ..
  ```

  where we are indicating that the index of the hydrogen atom to be replaced is 4. Do the same for `E` and `ZE_TS`.
- Use the interactive menu to create the files *pif.path* and *pif.calcs*. Set the length of the MEP from $-1.30\ a_0$ to $1.30\ a_0$. Include also the keywords `sctmns 10` and `scterr 0.1`.

  ```
  > add path *
  >> sbw = -1.30
  >> sfw = +1.30
  >> sctmns = 10
  >> scterr = 0.1
  >> ..
  ```
- Run ‘`pilgrim --ics`’ to incorporate the internal coordinates to *pif.struc*.
- Run ‘`pilgrim --pfn`’ to calculate the partition functions. The following files *pfn.E.slevel.txt*, *pfn.Z.slevel.txt* and *pfn.ZE_TS.txt* are created in `3-PLG_OUTPUT/`.
- Run ‘`pilgrim --path ZE_TS`’ to evaluate the variational Γ^{CVT} and tunneling transmission $\kappa^{\mathrm{CVT/SCT}}$ coefficients. The file *path.ZE_TS.001.txt* is written in `3-PLG_OUTPUT/` and the restart file `ZE_TS.001.rst` containing the information along the MEP is stored in the `4-PLG_RST/` directory.
- Run ‘`pilgrim --rcons`’ to obtain the CVT/SCT thermal rate constants. They are printed in the file *rcons.FA_ISO.slevel.txt*.

Notice that we have used the same labels for the deuterated species as for the root species. This is not important in this case because the run with the isotopic substituted species was carried out in a different directory than the one containing the root species. If we desire to run everything in the same directory, new labels for the isotopic species should be provided.

4. The inversion of ammonia

4.1 About this system

In the inversion of ammonia, the pyramidal structure of the molecule passes through a planar transition state by an umbrella-type of motion reaching at the end the same pyramidal structure. This symmetric reaction will be studied at 100, 200 and 300 K for the root and fully deuterated species. Dual-level (DL) calculations are performed using PM6 as the low-level (LL) and HF/STO-3G as the high level calculations. A frequency scale factor of 1.078 is used for the frequencies obtained at the PM6 level. Initially, the MEP is calculated between $-1.00\ a_0$ and $+1.00\ a_0$ with a stepsize of 0.005 a_0 and with Hessian updates every 10 steps. The MEP is extended till the SCT transmission coefficient at the lowest temperature is converged within a 1% The initial step along the path is taken using the normal mode eigenvector of the imaginary frequency at the transition state (default). The frequencies along the MEP are projected using redundant internal coordinates.

The WEs for this system are:

- WE05: LL and DL CVT/SCT
- WE06: LL and DL CVT/SCT calculations for root and deuterated species

- ESSO: *Gaussian 09*
- Level of calculation: PM6//HF/STO-3G
- Working temperatures (in K): 100, 200 and 300

4.2 WE05: LL and DL CVT/SCT

Study the inversion reaction using conventional CVT/SCT.

Steps:

- Create the directory `WE05/` inside tests_to_run/NH3 (hereafter the working directory for this system).
- Copy the directory `UDATA/` from `tests_to_run/NH3` to `WE05/`. Inside `UDATA/`, there are

two subdirectories with the names nh3_pyram/ and nh3_planar/.

- Run ‘pilgrim --gather’. This will create the directories 1-GTS/ and 5-MOLDEN/ with the *gts* and *molden* files, respectively. The latter can be visualized with Molden.
- Run ‘pilgrim --input’ and inside the input menu incorporate the frequency scale factor by typing:

```
> mod struc *
>> freqscal = 1.078
>> ..
```

- Write ‘add nh3inv : nh3_pyram -> nh3_planar -> nh3_pyram’ to define the chemical reaction.
- Generate the rest of Pilgrim input files, including *pif.dlevel*.
- Run ‘pilgrim --ics’ to incorporate the internal coordinates to the *pif.struc* file.
- Run ‘pilgrim --pfn’ to calculate the partition functions of the stationary points with the energies calculated at the low level.
- Edit *pif.calcs* to perform the PM6 calculations along the MEP. Set also the high level to HF/STO-3G in the template.
- Run ‘pilgrim --path’ to build the MEP at the low level.
- Run ‘pilgrim --rcons’ to calculate the CVT/SCT at the low level.
- Run ‘pilgrim --hlcalc to perform the high-level calculations. By default the high-level calculations are preformed at the stationary points plus 3 equidistant points at each side of the MEP. The high-level energies are stored in the file *highlevel.txt* inside the 2-PLG_DATA/ directory.
- Run ‘pilgrim --pfn --dlevel’ to generate the partition functions of the stationary points with the energies calculated at the high level. Notice that for cases as this, i.e., for a molecule without torsional conformers, the high-level partition functions coincide with the low-level partition functions. In spite of that, this step is mandatory.
- Run ‘pilgrim --path --dlevel’ to build the MEP at the dual-level using the ISPE algorithm.
- Run ‘pilgrim --rcons --dlevel’ to calculate the dual-level CVT/SCT thermal rate constants.

4.3 WE06: LL and DL CVT/SCT calculations for root and deuterated species

Same as WE05 but with the substitution of all hydrogen atoms by deuterium. In this case, we would like to have the results of the root and deuterated species in the same directory, so new labels have to be defined for the deuterated reaction. Because the electronic structure information is the same for the reaction inversion of NH_3 and ND_3 no more ESFILs are needed. However, new additions to the *pif.struc*, *pif.chem*, *pif.path* and *pif.dlevel* are needed.

Steps:

- Copy the directory WE05/ as WE06/ (hereafter the working directory for this system).
- Run ‘pilgrim --input’ and inside the input menu add the species, i.e., nd3_pyram and nd3_planar:

```
> mod struc nh3_pyram
>> copywith D(all_H) as nd3_pyram
> mod struc nd3_planar
>> copywith D(all_H) as nd3_planar
```

- Write ‘add nd3inv : nd3_pyram -> nd3_planar -> nd3_pyram’ to define the chemical reaction.
- Add the new MEP with ‘add path nd3_planar’ and modify the variables as for the case of nh3_planar.
- Add the isotopic reaction by typing ‘add dlevel’.
- Edit *pif.calcs* to perform the PM6 calculations along the MEP. Set also the high level to HF/STO-3G in the template.
- Run ‘pilgrim --pfn’ to calculate the partition functions.
- Run ‘pilgrim --path nd3_planar’ to build the MEP for the deuterated reaction at the low level.
- Run ‘pilgrim --rcons nd3inv’ to calculate the thermal rate constants for the deuterated reaction at the low level.
- Run ‘pilgrim --hlcalc to perform the calculations at the high level. For the root species the calculations are already stored, so they are not repeated.
- Run ‘pilgrim --pfn --dlevel’ to calculate the partition functions at the dual level.
- Run ‘pilgrim --path nd3_planar --dlevel’ to build the MEP for the deuterated reaction at the dual level.
- Run ‘pilgrim --rcons nd3inv --dlevel’ to calculate the thermal rate constants for the deuterated reaction at the dual level.

5. Hydrogen abstraction reactions

5.1 About this system

This chapter presents several tests of hydrogen abstraction reaction from ethanol by atomic hydrogen. This process is more complex than the ones of previous examples due to the fact that it involves three different reactions, as the hydrogen can be abstracted from ethanol at three different positions:

$$
\begin{aligned}
&CH_3CH_2OH + H \xrightarrow{\text{TSA}} H_2 + CH_3CHOH \quad &\text{(Ra)}\\
&CH_3CH_2OH + H \xrightarrow{\text{TSB}} H_2 + CH_2CH_2OH \quad &\text{(Rb)}\\
&CH_3CH_2OH + H \xrightarrow{\text{TSC}} H_2 + CH_3CH_2O \quad &\text{(Rc)}
\end{aligned}
$$

Ethanol presents some conformational flexibility. It has an *anti* conformer and two *gauche* conformers (which are conformational enantiomers). Therefore the *anti* and one of the *gauche* conformations have to be supplied, so the program can calculate the MS-HO partition function. All the torsional conformers of the TS for the hydrogen abstraction from the α-C (TSA), methyl group (TSB) and hydroxyl group (TSC) are supplied in `tests_to_run/`. There are 3, 9 and 3 torsional conformers of the transition states corresponding to the reactions Ra, Rb and Rc at the HF/STO-3G level of calculation. The rate constant are calculated using MS-TST and MP-CVT/SCT. In all cases the MEP was calculated with a stepsize of 0.010 a_o and with Hessian update every 10 steps. The initial length of the MEP is -0.5 a_o and $+0.5$ a_o for all cases and the SCT transmission coefficients are calculated using a convergence criterion of 1%. In this case the electronic structure calculations are performed with Orca. Once the rate constants are know a KMC simulation is carried out to obtain the products yield.

In some of the WEs, hindered rotor anharmonicity (calculated with `Q2DTor` and `MSTor` programs is also included. The output files of this software is included in the directories `ANHAR_WE08` and `ANHAR_WE10` inside `tests_to_run/EtOH`.

The following isotopic substitutions are also considered in these tests:

- First isotopic substitution

$$
\begin{aligned}
&CD_3CD_2OH + H \xrightarrow{\text{TSAD1}} HD + CD_3CDOH \quad &\text{(RaD1)}\\
&CD_3CD_2OH + H \xrightarrow{\text{TSBD1}} HD + CD_2CD_2OH \quad &\text{(RbD1)}\\
&CD_3CD_2OH + H \xrightarrow{\text{TSCD1}} H_2 + CD_3CD_2O \quad &\text{(RcD1)}
\end{aligned}
$$

- Second isotopic substitution

$$CH_3CH_2OD + D \xrightarrow{TSAD2} HD + CH_3CHOD \quad (RaD2)$$
$$CH_3CH_2OD + D \xrightarrow{TSBD2} HD + CH_2CH_2OD \quad (RbD2)$$
$$CH_3CH_2OD + D \xrightarrow{TSCD2} D_2 + CH_3CH_2O \quad (RcD2)$$

- Third isotopic substitution

$$CD_3CD_2OD + D \xrightarrow{TSAD3} D_2 + CD_3CDOD \quad (RaD3)$$
$$CD_3CD_2OD + D \xrightarrow{TSBD3} D_2 + CD_2CD_2OD \quad (RbD3)$$
$$CD_3CD_2OD + D \xrightarrow{TSCD3} D_2 + CD_3CD_2O \quad (RcD3)$$

The WEs for this system are:

- WE07: MS-TST and KMC
- WE08: MP-CVT/SCT with torsional anharmonicity and KMC
- WE09: KMC simulation using analytical expressions
- WE10: MP-CVT/SCT with isotopic subtitutions and anharmonicity

- ESSO: Orca
- Level of calculation: HF/STO-3G
- Working temperatures (in K): 250, 300, 350, 400, 450, 500, 700, 1000, 1500, 2000 and 2500

5.2 WE07: MS-TST and KMC

Steps:

- Create the directory `WE07/` inside `tests_to_run/EtOH` (hereafter the working directory for this system).
- Copy the directory `UDATA/` inside `tests_to_run/EtOH` to `WE07/`. For this reaction remove the directories corresponding to products. Notice that TST rate constants are independent of the product species. Therefore, even without having a detailed information about products, it is possible to calculate their yield using KMC.
- Run '`pilgrim --gather`'. This will create the directories `1-GTS/` and `5-MOLDEN/` with the *gts* and the *molden* files, respectively.
- Run '`pilgrim --input`' and add the proper weights for the conformers with enantiomers. In the case of `TSB` and `TSC` there are conformational enantiomers, but in the case of `TSA` the conformers are configurational and should be included when the KMC simulation is carried out. For instance, the weights in the ethanol molecule can be changed by:

```
> mod struc Etoh
>> weight(all) = 2
>> ..
```

  The keyword `weight(all) = 2` changes all the weights of structures with C_1 point group symmetry to 2.
- Add the temperatures and the chemical reactions. Initialize the KMC variables with:

```
> add kmc
>> pop(Etoh)_0 = 1e14
>> pop(H)_0 = 1e6
>> ..
```

- Run ‘pilgrim --pfn’ to generate the MS-HO partition functions for ethanol, TSA, TSB and TSC and the HO partition functions for the rest of species.
- Run ‘pilgrim --rcons’ to calculate the MS-TST thermal rate constants.
- Edit *pif.kmc* and multiply Ra by two. This coefficient indicates that there is another channel identical to Ra with TSs that are configurational isomers to the ones that are being considered.
- Run ‘pilgrim --kmc’ to obtain the products distribution and the time scale of the reaction for the consumption of the limiting reagent. The output file is located in the directory 3-PLG_OUTPUT/ and is called kmc.slevel.txt

5.3 WE08: MP-CVT/SCT with torsional anharmonicity and KMC

Steps:

- Copy the directory WE07/ as WE08/.
- In UDATA/ add the directories with the ESFILs corresponding to products.
- Create a directory inside WE08/ called ANHAR/ and add the Q2DTor output files of the species with hindered rotors located in tests_to_run/EtOH/ANHAR_WE08. In this case we are interested in the species ethanol, TSA, TSB and TSC.
- Specify the output files of Q2DTor using the interactive menu ‘pilgrim --input’. For example:

```
    > mod struc Etoh
    >> anharfile = ethanol.q2dtor
    >> ..
```

- Use the interactive menu to generate the *pif.path* and *pif.calcs* files.
- Run ‘pilgrim --ics’ to generate redundant internal coordinates for each of the TS species.
- Run ‘pilgrim --pfn’. Pilgrim compares the MS-HO partition functions with the one calculated by Q2DTor. If there is an important discrepancy, Pilgrim gives a warning and anharmonicity is not included for that species.
- Run ‘pilgrim --path --software orca’ to start the calculation of the MEP from each of the TSs in a sequential manner. The calculation can be sped up by running several paths at the same time. For instance ‘pilgrim --path TSA --software orca’ runs sequentially the MEPs from each of the conformers of TSA. Simultaneous runs ‘pilgrim --path TSA.001 –software orca’, ‘pilgrim --path TSA.002 --software orca’ and ‘pilgrim --path TSA.003 --software orca’ are also possible. Notice that we do not need to specify sbw and sfw values for each conformation. The keywords sctmns and scterr make the MEP to continue till the SCT transmission coefficient is obtained for each MEP within the specified percentage error.
- Run ‘pilgrim --rcons’ to calculate the MP-CVT/SCT thermal rate constants with and without torsional anharmonicity.
- Modify the pif.kmc file to consider the anharmonic MP-CVT/SCT rate constants. The interactive menu can be used.
- Run ‘pilgrim --kmc’ to obtain the products distribution and the time scale of the reaction for the consumption of the limiting reagent.

5.4 WE09: KMC simulation using analytical expressions

Steps:

- Run ‘pilgrim --fit’ in WE08/ to obtain the fit to the different analytical expressions and take the anharmonic MP-CVT/SCT forward rate constants with the parameters corresponding to the analytic form 4.
- Create the directory WE09/ and copy the files *pif.temp*, *pif.chem* and *pif.kmc* of WE08/ into WE09/ .
- Already in WE09/, modify the *pif.kmc* file by introducing the parameters for the analytical expressions.
- Run ‘pilgrim --kmc’ to obtain the products distribution and the time scale of the reaction for the consumption of the limiting reagent. Results should be very similar to those obtained in WE08.

5.5 WE10: MP-CVT/SCT with isotopic subtitutions and anharmonicity

Steps:

- Copy the directory WE08/ as WE10/ and enter WE10/.
- Run ‘pilgrim --input’ and inside the input menu and create the deuterated species. For instance for ethanol and for TSA:

```
> mod struc Etoh
>> copywith D(4, 5, 6, 7, 8) as Et_iso1
> mod struc Etoh
>> copywith D(9) as Et_iso2
> mod struc Etoh
>> copywith D(all_H) as Et_iso3
> mod struc TSA
>> copywith D(4, 5, 6, 7, 8) as TSA_iso1
> mod struc TSA
>> copywith D(9, 10) as TSA_iso2
> mod struc TSA
>> copywith D(all_H) as TSA_iso3
```

- Specify the output files of MSTor to include torsional anharmonicity. For example:

```
> mod struc Etoh
>> anharfile = ethanol.mstor
> mod struc Et_iso1
>> anharfile = ethanol.iso1.mstor
> mod struc Et_iso2
>> anharfile = ethanol.iso2.mstor
> mod struc Et_iso3
>> anharfile = ethanol.iso3.mstor
```

- Create a directory inside WE10/ called ANHAR/ and add the MSTor output files of the species with hindered rotors located in tests_to_run/EtOH/ANHAR_WE10.Follow the same instructions and run Pilgrim as for WE08.

Part Three - Pilgrim input files

6. Pilgrim input files

This Chapter explains in some detail the format and keywords of each of the seven Pilgrim input files (*pif*s). Along the description of the *pif*s, some extracts from the WEs described in the previous chapters are included.

Before proceeding, we highlight that the first input file, *pif.struc*, is generated when Pilgrim is run with the `--gather` option. Although the other six input files can be generated from scratch, we highly recommend to create and modify them using the interactive menu. This menu can also be used to modify *pif.struc*.

Notice that in all the input files, with exception of *pif.calcs*, the # symbol precedes a comment.

6.1 The *pif.struc* file

The *pif.struc* files gathers all the structural information of the stationary points (`ctcsp`) in blocks. This file is generated when executing:

```
    pilgrim.py --gather
```

Each block, which has a `ctcsp` label, may contain one chemical structure or a cluster of torsional conformers (CTCs). The blocks are limited by the `start_ctc` and `end_ctc` keywords:

```
start_ctc ctcsp
 root      spname
 $keyword $value
 ...
 $keyword $value
end_ctc
```

where `ctcsp` is the name given to the block. Notice that the keyword `root` points towards the corresponding ESFILs and that, under isotopic substitution, the `spname` label should remain unchanged, whereas the `ctcsp` label should be different from the one of the root species. The available keywords are listed in Table 6.1.

Table 6.1: Keywords admitted values in the CTC blocks of the *pif.struc* file. Keywords between [ ] do not have default values and should be provided by the user.

Keyword	**Format**	**Default**	**Comment**
root	str	spname	Label of the species inside UDATA/ (or inside 1-GTS/); compulsory with iso
conformer	int * int	001 * 1	1st integer between 001-999 2nd integer (1 or 2), the weight of the conformer
mformu	str	Provided by Pilgrim	The molecular formula
ch	int	Provided by Pilgrim	The molecular charge
mtp	int	Provided by Pilgrim	The spin multiplicity
type	int	Provided by Pilgrim	0 for minimum, 1 for saddle point
freqscal	float	1.000	Frequency scale factor (also used along the MEP)
elestate	int float	Provided by Pilgrim	Degeneracy and relative E in hartree
[ics[.idx]]	str(s)	Provided by the user or by --ics	Definition of internal coordinates (see Table 6.2)
[iso[.idx]]	str(s)	Provided by the user	Isotopic substitution
[anharfile]	str	Provided by the user	The name of the MSTor/Q2DTor output file placed inside ANHAR/

The program creates as many blocks as species inside UDATA/ (or 1-GTS/). When dealing with isotopic substitutions, it is possible to add additional blocks, and therefore to add additonal species pointing to the same spname label:

WE12: Extract of pif.struc

```
start_ctc H2
   root      H2
   conformer 001  * 1
   ...
end_ctc

start_ctc HD
   root      H2
   conformer 001  * 1
   ...
   iso       D(1)
end_ctc
```

In the example, the molecule of H_2 has H2 as the spname. When one of the hydrogen atoms is substituted by deuterium, a new block with the species labeled HD is added. The root keyword indicates that HD is an isotopic derivative of H2.

6.1.1 Conformers and weights

Each of the conformers inside a block is specified using the syntax:

```
conformer idx * wgtconf
```

where `idx` refers to the index of the conformation in the *.gts* file inside the `1-GTS/` directory, and `wgtconf` is an integer number: 1 if the structure does not have a conformational enantiomer, and 2 otherwise. Therefore, `wgtconf` avoids electronic structure calculations for species that are conformational enantiomers. The program also include additional information (given as comments) as the relative energy of a given conformer and the point group symmetry. The latter can help us to decide if a given structure has conformational enantiomers.

In the case of a molecule with just one equilibrium structure, only one `conformer` line is used:

WE05: Extract of pif.struc

```
start_ctc nh3_pyram
  # conformers
  conformer 001 * 1 #  0.00 kcal/mol, C3v
  ...
end_ctc
```

For a compound with several conformations (for instance ethanol) the block contains one line per conformation:

WE07: Extract of pif.struc

```
start_ctc ethanol
  # conformers
  conformer 001 * 2 #  0.00 kcal/mol, C1
  conformer 002 * 1 #  0.25 kcal/mol, Cs
  ...
end_ctc
```

For the case of ethanol, `conformer 001` corresponds to one of the gauche conformers and, therefore, the weight of that structure is 2, whereas `conformer 002` is the anti conformer of ethanol without conformational enantiomers.

6.1.2 Electronic states

The keyword `elestate` can be used to define the electronic state to be considered in the electronic partition function. This keyword is followed by the degenerancy of the state and its relative energy with regard to the electronic ground state (in hartree). By default, only the ground state is considered. However, more states can be included just by adding more lines, which may be particularly important for atomic species. For example, for the fluorine atom two states should be specified as:

```
        elestate  4  0.0000000000E+00
        elestate  2  0.00184124E+00
```

6.1.3 Internal coordinates

For the case of transition states[i] a set of non-redundant or redundant internal coordinates can be specified with the keyword `ics`. These coordinates can be generated with the option `--ics` (run '`pilgrim --help ics`' for more information) or can be introduced directly by the user. This option is compulsory if the projection of the frequencies along the MEP is performed in internal coordinates. The keyword accepts as many coordinates as desired, as long as they are defined according to the format described in Table 6.2. For instance, for the transition state of the inversion of ammonia the internal coordinates were defined as:

WE05: Extract of pif.struc

```
ics        1-2   1-3   1-4
ics        2-1-3   3-1-4
ics        2-3-4-1
```

In cases with many conformations, it is possible to specify different sets of internal coordinates for different conformers by adding the index of the conformer to the `ics` keyword:

Example of two sets of internal coordinates in pif.struc

```
ics        1-2   1-4   1-5   1-6   2-3
...
ics        1_3_8_2   1_7_8_2   2_4_5_1   4_5_6_1
ics        1-2-3-9   5-1-2-7
ics.002    1-2   1-4   1-5   1-6   2-3
...
ics.002    1_7_8_2   2_4_5_1   3_7_8_2   4_5_6_1
ics.002    3-2-1-6   7-2-3-9
```

Table 6.2: Format for the definition of internal coordinates. Each atom, specified by the integers `m`, `n`, `o`, or `p` is referred using its numbering in the molecule (starting at 1).

Type of internal coordinate	**Format**	**Example**	**Comment**
Stretch	`m-n`	`1-2`	
Linear bend	`m=n=o`	`1=2=3`	
Angular bend	`m-n-o`	`1-2-3`	
Proper torsion	`m-n-o-p`	`1-2-3-4`	`n-o` is the central bond

Improper torsion	`m-n-o-p`
	or
	`m_n_o_p`
		`1-2-3-4`
		or
		`1_2_3_4`
			`p` is the central atom

[i] Identified by a `ctcts` label; i.e., by a `ctcsp` label with `type 1`.

6.1.4 Isotopic substitutions

The *pif.struc* file also includes at the beginning a list of the most common masses (in amu) for isotopic substitutions (`imass` block) where additional atomic masses can be included:

Extract of pif.struc

```
start_isomass
  D      =     2.0141018
  T      =     3.0160493
  ...
  O18    =    17.9991600
end_isomass
```

It is possible to specify different isotopic substitutions with `iso[.idx]`. If `idx` is not specified, the program assumes that the numbering of the atoms is the same for all the conformers. The syntax of the `iso` line is as follows:

```
  iso[.idx]  imass(atidx(s)) ...
```

where:

- `imass` is the name of the isotopic mass defined in the `isomass` block,
- `atidx(s)` are the numerical indexes of the affected atoms; if all the atoms of a type are going to be substituted, `all_X` can be used, `X` being the atomic symbol.

For example, all the hydrogen atoms of ammonia can be replaced by:

WE06: Extract of pif.struc

```
start_isomass
  D     =     2.0141018
end_isomas

start_ctc nd3_pyram
  root   nh3_pyram
  # conformers
  conformer 001 * 1 #   0.00 kcal/mol, C3v
  # common
  mformu     H(3)N
  ch         0
  mtp        1
  type       0
  freqscal   1.000
  elestate   1   0.0000000000E+00
  iso        D(all_H)
end_ctc
```

Equivalent to '`iso D(all_H)`' is:[ii]

```
iso.001 D(all_H)
iso D(2,3,4)
iso.001 D(2,3,4
```

[ii] Notice that labels 2, 3 and 4 refer to the hydrogen atoms.

6.2 The *pif.temp* file

This file contains the list of working temperature in Kelvin. For example:

WE08: pif.temp file

```
    250.00    300.00    350.00    400.00    450.00    500.00    700.00
   1000.00   1500.00   2000.00   2500.00
```

6.3 The *pif.chem* file

In this file, each (elementary) reaction of interest is defined using the following scheme:

```
chemname : reaction_equation
```

where `chemname` is a label given by the user to identify the reaction, and `reaction_equation` contains the name of reactants, transition state and products separated by arrows (`-->`). The names should coincide with the `ctcsp` labels of the blocks defined in *pif.struc*.[iii] The available species can be listed with the `--ls` option. In the case of two reactants (or products), the "+" symbol should be used to separate their names.

For example, in the inversion of ammonia we have:[iv]

WE05: pif.chem

```
    nh3_inv : nh3_pyram --> nh3_planar --> nh3_pyram
```

whereas for the hydrogen abstraction from the ethanol molecule by atomic hydrogen:

WE08: pif.chem

```
Ra : Etoh + H --> TSA --> PA + H2
Rb : Etoh + H --> TSB --> PB + H2
Rc : Etoh + H --> TSC --> PC + H2
```

[iii] Actually, if we are not interested in the backward process (from products to reactants), the products do not have to be defined inside *pif.struc*. In such a case, the user should give a name to each product in the definition of the chemical reaction, even if the ESFILs associated with it does not exist. In this manner, the product(s) can still be considered in a further study by kinetics Monte Carlo.

[iv] The calculation of the thermal rate constants of the reactions defined in *pif.chem* already include the rotational symmetry numbers (internal and external) and the weight of conformations (included in the *pif.struc* file). However, there are special cases, as the the ammonia inversion (WE05), in which the calculated rate constant is half the one that ignores the symmetry numbers. This spurious factor of $1/2$ is always present in reactions for which the reactants and products are indistinguishable. Pilgrim corrects that, by multiplying the calculated rate constants by two. Pilgrim also automatically 'corrects' the rate constants for bimolecular reactions with one reagent.

6.4 The *pif.path* file

The file *pif.path* contains all the keywords needed to control the calculation of the MEP. They are organized inside `mep` blocks, one for each `ctcts` defined by the user. Each block has the following format:

```
start_mep ctcts
  keyword[.idx] value
  ...
end_mep
```

The available keywords are listed in Table 6.3. Every keyword inside a block is common to all torsional conformers of `ctcts`. However, the user can add the index of a given conformer to a keyword to treat it differently. For instance, in the example below, the MEP is extended till $s = -1.00\ a_0$ instead of to $s = -0.75\ a_0$ in the backward direction for conformer `002`.

pif.path

```
start_mep TSA
  sbw      -0.75
  sfw       0.75
  ds        0.01
  hsteps    10
  sbw.002  -1.00
end_mep
```

By default, Pilgrim calculates the MEP using the Page-McIver algorithm, but the Euler algorithm is also available. Notice that the Page-McIver algorithm makes use of Hessian matrices to calculate the MEP and its curvature, whereas the Euler algorithm is exclusively based on the direction of the gradient. However, Hessians along the path are needed anyway to obtain the VTST coefficients. In consequence, the computational effort associated with the Euler and Page-McIver algorithms is similar, but the latter is more accurate. For this reason, we highly recommend using the default algorithm.

6.4.1 Basic keywords

The interactive menu allows modifying the following keywords:

- `sbw`: the extension of the MEP in the backward direction (in bohr).
- `sfw`: the extension of the MEP in the forward direction (in bohr).
- `ds`: the step in the intrinsic reaction coordinate in bohr.
- `hsteps`: the number of steps for the Hessian matrix update.
- `sctmns`: if set to a number smaller than 1, the SCT coefficient will be calculated only using the available MEP. Otherwise, the MEP is expanded automatically up to `sctmns` times, increasing its limits by an amount of `hsteps`×`ds` each time in order to achieve the convergence of the SCT transmission coefficient at the lowest temperature.
- `scterr`: defines the convergence criterium for the SCT coefficient (in percentage). If the relative variation in the coefficient is smaller than the specified percentage, the MEP is not longer increased. This keyword must go together with `sctmns` and lacks of sense if `sctmns` is smaller that 1. Pilgrim will stop when the number of steps is larger than the ones

Table 6.3: Available keywords in the *pif.path* input file. Some keywords are explained with more detail in the text. The 'advanced' keywords cannot be modified in the input menu and the user has to edit the *pif.path* file.

Keyword	Accepted values	Default	Brief description
Basic			
sbw	float	-0.75	Minimum value of s (in a_o)
sfw	float	+0.75	Maximum value of s (in a_o)
ds	float	0.01	Stepsize along the path (in a_o)
hsteps	integer	10	Number of steps for Hessian update
paral	no/yes	no	yes to calculate both sides of the MEP simultaneously
sctmns	integer	0	If *value* $\geq$1 increases the path (see text)
scterr	float	1.0	See text
Advanced keywords related to the MEP			
fwdir	ic (++/--)	Provided by Pilgrim	The internal coordinate increases (++); or decreases (--) in the forward direction
cubic	no/yes/float	no	Use cubic first step
mtype	es/pm	pm	Euler or Page-McIver algorithms to calculate the path
mu	float	1.0	The scaling mass (in *amu*)
epse	float	1e-8	See text (in E_h)
epsg	float	1e-4	See text (in E_h/a_o)
eref	float / auto	auto	Defines the ref. energy for the MEP (in E_h)
Advanced keywords related to the VTST coefficients			
lowfq	integer float [++/--/+-]		*imode*: integer; freq. index; float: freq. in cm^{-1}; ++: $s > 0$; --: $s < 0$; +-: $s < 0$ & $s > 0$
cvt	no/yes	yes	Calculates the Γ^{CVT} coefficient
sct	no/yes	yes	Calculates the $\kappa^{\mathrm{CVT/SCT}}$ transmission coefficient
qrc	integer integer		See text
muintrpl	linear/cubic int	linear 0	Type of interpolation for μ_{eff} at the TS and at the first points along the MEP
e0	float / auto	auto	Specifies the lowest energy for the tunneling calculations (in E_h)
useics	no/yes	yes	The projection of the Hessians along the MEP is performed in ics (specified in *pif.struct*)
v1mode	grad/hess	grad	Indicates how the $B_{m,F}$ terms are calculated

specified in sctmns or when the percentage is smaller than value specified in this keyword. The keywords epse and epsg may override sctmns and scterr, because the program may consider that the MEP is converged.

- paral: to activate the calculation of both sides of the MEP at the same time (yes). By default this option is deactivated (no).

6.4.2 Advanced keywords

In order to modify or include these keywords, the user should edit the *pif.path* file.

- `fwdir`: Defines a forward direction based on the increase (++) or decrease (`--`) of a given internal coordinate. The internal coordinate can be a distance, a bond angle or a dihedral angle. For instance, `fwdir 3-1-2-4 ++` means that the forward direction of the MEP is toward the increase of the dihedral angle containing the `3-1-2-4` atoms. Pilgrim provides a value for this variable based on the eigenvector of the mode with the imaginary frequency.
- `mu`: Defines the scaling mass that transforms Cartesian coordinates to mass-scaled coordinates. Its default value is 1.0 amu.
- `cubic`: if set to `no` (the default), the initial step of the path is taken using the eigenvector associated to the imaginary frequency. If set to `yes`, the program calculates two Hessian matrices at both sides of the path and with a stepsize from the saddle point of $s = \pm 10^{-4} a_\mathrm{o}$. This stepsize is the default in `cubic yes`, but it can be changed (for instance `cubic 2E-4`). This two additional Hessian calculations allow the use of a quadratic Taylor expansion for the first step of the MEP.
- `epse`: Stops the MEP if the variation in energy between two consecutive steps is smaller than a given threshold. The default value for this threshold is $1 \times 10^{-8} E_\mathrm{h}$.
- `epsg`: Stops the MEP if the module of the gradient in Cartesian coordinates ($|\vec{g}|$) is smaller than the value of this keyword. The default value is $1 \times 10^{-4} E_\mathrm{h}/a_\mathrm{o}$
- `eref`: Defines the reference energy for the MEP (in E_h). It can be set to a float number or to `auto`. In the last case, the reactants energy is set as the reference if the saddle point of this path has been defined in *pif.chem* and Pilgrim was previously executed with the `--pfn` option. Otherwise, the energy at `sbw` is used as the reference.
- `lowfq`: Sometimes, the lowest vibrational modes along the MEP may present really small (or even imaginary) frequencies. With this keyword, those frequencies may be substituted by fixed values. It requires two values: (1) the frequency index i and (2) the frequency limit in cm^{-1}, ω_L. It only switches on when the i-th frequency along the MEP, ω_i, is smaller than ω_L, i.e.:

  $$\text{if } \omega_i < \omega_\mathrm{L} \Rightarrow \omega_i \leftarrow \omega_\mathrm{L}$$

  The user can define as many `lowfq` lines as needed. The symbols `++`, `--` and `+-` indicate that the correction is performed for $s > 0$, $s < 0$ and for any sign of s, respectively.
- `qrc`: Considers quantized-reactant-states tunneling calculations. The energy levels are determined using the harmonic approximation. It can only be used with unimolecular reactions. This keyword should be included when studying tunneling effects at very low temperatures. It requires two values: (i) an integer that specifies the index of the normal mode to be quantized (notice that in Pilgrim the normal-mode frequencies are sorted in increasing value of frequency); (2) the maximum energy level at which this quantization is carried out.
- `muintrpl`: The effective mass of tunneling at the TS, $\mu_\mathrm{eff}(s=0)$, cannot be calculated at the transition state. It has to be interpolated using values at the nearest points along the MEP. The default is `linear 0`, which means that only $\mu_\mathrm{eff}(s=0)$ is interpolated, and the value is obtained as the average of $\mu_\mathrm{eff}(s=s_{-1})$ and $\mu_\mathrm{eff}(s=s_1)$, being $s=s_{-1}$ and $s=s_1$ the first points along the backward and forward direction of the MEP at which a Hessian is available. If the user specifies `linear 1` the values of μ_eff at $s=s_{-1}$, $s=0$ and $s=s_1$ are linearly interpolated using the values at $s=s_{-2}$ and $s=s_2$ (the second points along the backward and forward direction of the MEP at which a Hessian is available. If `cubic` is used instead of `linear`, μ_eff is interpolated using a cubic spline.
- `e0`: Defines the lowest energy at which the tunneling probability is evaluated. The user can either define this value (in E_h) or can set it to `auto`. In the last case, this value is taken as

  $$E_0 = \max\{(E + \mathrm{ZPE})_\mathrm{R}, (\mathrm{E} + \mathrm{ZPE})_\mathrm{P}\}$$

  where E is the total energy, ZPE is the vibrational zero-point energy, and R and P represent the reactants and the products, respectively. Notice that this requires the previous execution of Pilgrim with the --pfn option.

- useics: If this keyword is set to yes, the Hessian matrix at each point of the path is projected using internal coordinates. This option requires the definition of a set of internal coordinates in the *pif.ctc* file. If set to no, Cartesian coordinates are used.
- v1mode: If set to grad, the curvature of the path at each point is calculated numerically from the gradient vectors; if set to hess, the curvature is obtained from the Hessian matrix. The default value is grad, which in general provides more stable results than hess.

6.5 The *pif.calcs* file

The *pif.calcs* file contains the templates for the electronic structure calculations needed by Pilgrim in the calculation of the MEP. It also contains a template to perform the high-level calculations.

The templates for the MEP calculations are defined inside meppoint blocks:

pif.tes: example

```
start_meppoint ctcts ESSO
  the template
end_meppoint
```

An example, extracted from WE04, is shown below:

WE04: pif.calcs file

```
start_meppoint nh3_plane gaussian
%nproc=1
%mem=1GB
%chk=[Pilgrim_name].chk
#p pm6
scf=verytight
NoSymm
[Pilgrim_gradhess]

Input file for MEP calculation

0 1
[Pilgrim_geometry]
end_meppoint
```

Notice that the commands in brackets (in blue) are not *Gaussian* commands, but are indicators to Pilgrim to perform different actions when the ESSO is working with a given geometry:

- [Pilgrim_name]: is substituted by the name of the file generated by Pilgrim.
- [Pilgrim_geometry]: is substituted by the Cartesian coordinates of the structure.
- [Pilgrim_gradhess]: is substituted by the corresponding *Gaussian*/Orca keyword for the calculation of the gradient and/or the Hessian matrix.

Unlike in other *pif* files, **'#' cannot be used to start a comment**. The templates for the high-level calculations (including single-point energy calculations for MEP points) are defined inside `highlevel` blocks:

pif.tes: example

```
start_highlevel ctcsp ESSO
  the template
end_highlevel
```

An example is:

WE04: pif.calcs file

```
start_highlevel nh3_plane gaussian
%nproc=1
%mem=1GB
%chk=[Pilgrim_name].chk
#p hf/sto-3g
scf=verytight
NoSymm

Input file for MEP calculation

0 1
[Pilgrim_geometry]

end_highlevel
```

6.6 The *pif.kmc* file

The *pif.kmc* file contains the information to perform a simulation of a chemical kinetics mechanism using the kinetic Monte Carlo algorithm. The file can be created with the aid of the interactive menu but it may need further editing by the user. The general syntax for the selection of a chemical reaction inside the *pif.kmc* file is:

```
k(chemname[.fw/.bw])[*wgtreac] labelrc
```

Every rate constant is defined using the label of the reaction name (`chemname`). On the one hand, reactions that lead to one product are considered reversible (forward and backward reactions). On the other hand, reactions that lead to two products are considered irreversible (only forward reaction). However, the user can modify these defaults using `chemname.fw` and `chemname.bw`, which means that the forward reaction and the backward reaction are being specified, respectively.

The options for `labelrc` are given in Table 6.4. It is also possible to specify rate constants that are given by analytical expressions, instead of being calculated by Pilgrim. In such a case, the user must choose the analytic equation and include the corresponding parameters (see Table 6.5).

Table 6.4: Different options for `labelrc`.

Rate constant	`label`	**Comment**
TST	`tst`	
CVT	`cvt`	
TST/ZCT	`tstzct`	
TST/SCT	`tstsct`	
CVT/ZCT	`cvtzct`	
CVT/SCT	`cvtsct`	
anharmonic TST	`anhtst`	
anharmonic CVT	`anhcvt`	
anharmonic TST/ZCT	`anhtstzct`	
anharmonic TST/SCT	`anhtstsct`	
anharmonic CVT/ZCT	`anhcvtzct`	
anharmonic CVT/SCT	`anhcvtsct`	
Arrhenius	`analytic1`	Requires two parameters
van't Hoff type 1	`analytic2`	Requires three parameters
van't Hoff type 2	`analytic3`	Requires four parameters
Truhlar and co.	`analytic4`	Requires five parameters

For the case of analytical expressions for thermal rate constants, the final part of the file should be similar to the one of WE09, i.e.,

WE09: Extract of the *pif.kmc* file.

```
k(Ra.fw)*2 analytic4  7.774E-15  3.151E+03  3.555E+00  3.000E+02  2.581E+02
k(Rb.fw)   analytic4  9.430E-15  4.349E+03  3.901E+00  3.000E+02  2.379E+02
k(Rc.fw)   analytic4  2.118E-13  1.247E+03  2.459E+00  3.000E+02  1.945E+02
```

Table 6.5: Available analytical expressions for the rate constants to be entered in the *pif.kmc* file. Pilgrim fits the expressions `analytic3` and `analytic4` using $T_r = 300$ K

`label`	**Equation**	**Parameters** (in input order)	**Units**
`analytic1`	$k = Ae^{-B/T}$	A, B	A: molecule cm^{-3} s^{-1} B : K
`analytic2`	$k = AT^n e^{-B/T}$	A, B, n	AT^n: molecule cm^{-3} s^{-1} B: K
`analytic3`	$k = A\left(\frac{T}{T_r}\right)^n e^{-B/T}$	A, B, n, T_r	A: molecule cm^{-3} s^{-1} B and T_r: K
`analytic4`	$k = A\left(\frac{T}{T_r}\right)^n e^{-B(T+T_0)/(T^2+T_0^2)}$	A, B, n, T_r, T_0	A: molecule cm^{-3} s^{-1} B, T_r and T_0: K

Table 6.6: Available keywords in the *pif.kmc* input file.

Keyword	**Type**	**Default**	**Brief description**
`psteps`	integer	1000	Prints the number of molecules of each species every given number of steps.
`volume`	float	1	Volume of the reaction vessel (in cm^3).
`timeunits`	str	`ps`	Time evolution units: `fs, ps, mcs, ms, s, min, hr` femtoseconds, picoseconds, microseconds, miliseconds, seconds, minutes and hours, respectively.
`pop(ctcsp)_0`	float	0	Initial number of particles of the *ctcsp* species.

Besides of the definition of the thermal rate constants, Table 6.6 shows a list of additional keywords that can be specified.. The thermal rate constant can be multiplied by a weight (integer), specified in `wgtreac`. Notice that the chemical reactions are defined in *pif.chem*.

WE08: pif.chem

```
Ra : Etoh + H --> TSA --> PA + H2
Rb : Etoh + H --> TSB --> PB + H2
Rc : Etoh + H --> TSC --> PC + H2
```

In the example, the rate constant for the abstraction of one of the hydrogen atoms from the α-C of ethanol (reaction `Ra`) should be multiplied by two because it is possible to abstract another equivalent hydrogen from the α-C . The transition states for the abstraction of one of the hydrogen atoms are configurational isomers of the transition states resulting from the abstraction of the other hydrogen.[v]

[v] Notice that the weight of a given configuration in the *pif.struc* (`wgtconf`) file only applies to conformational isomers, but not to configurational isomers. This situation always occurs when there is a hydrogen abstraction from a carbon atom of the type R1-CH_2-R2, where the two hydrogen atoms are equivalent and R1 and R2 are two different non-hydrogen substituents. This is because the abstraction of each of the hydrogen atoms leads to two configurational isomers with two different asymmetric carbon atoms. None of the structures generated by the abstraction of one hydrogen atom can interconvert with the structures generated by the abstraction of the other hydrogen atoms, leading to two chiral reaction

The following example shows a case in which KMC will use CVT/SCT thermal rate constants that were previously calculated with Pilgrim:

WE08: *pif.kmc* file

```
# KMC Parameters
psteps    1000   # print data each nstp steps
volume     1.0   # simulation volume (mL)
timeunits s      #time in seconds

# Initial (non-zero) populations (number of molecules)
pop(Etoh)_0    1.00e+14
pop(H)_0       1.00e+06
pop(H2)_0      0.00e+00
pop(PA)_0      0.00e+00
pop(PB)_0      0.00e+00
pop(PC)_0      0.00e+00

# Selection of the rate constant to use
k(Ra)*2 anhcvtsct
k(Rb)   anhcvtsct
k(Rc)   anhcvtsct
```

6.7 The *pif.dlevel* file

The *pif.dlevel* contains the target structures to be calculated at high-level. The syntax of the file is:

```
spname1
spname2
...
ctcts1[.idx] {auto_nptbw_nptfw}
ctcts2[.idx] {auto_nptbw_nptfw}
...
```

The high-level single-point energies of the stationary points do not depend upon the isotopic substitution, therefore the `spname` labels are used. However, the MEP is mass-dependent, and the points to be calculated along the MEP (given between curly brackets) are preceded by a `ctcts` label. This allows us to specify which MEP is being calculated when there are reactions with isotopic substitutions. The integers `nptbw` and `nptfw` indicate the number of (equally spaced) points to be calculated in the backward and forward directions of the MEP, respectively. For instance:

WE06: pif.dlevel

```
nh3_planar {auto_3_3}
nh3_pyram
```

calculates single-point high energies for `nh3_pyram` (reactant), `nh3_planar` transition state and three equally spaced points along the backward and forward directions of the MEP. The first point to be calculated corresponds to the geometry with the smallest value of s, whereas the last point corresponds to the geometry with the highest value of s.

channels. The weight `wgtreac` is the unity for the hydrogen abstractions from the methyl and hydroxyl groups.

It is also possible to specify the values of *s* along the reaction path at which the high-level single-point calculations are going to be performed. The syntax is:

```
ctcts1[.idx] {svalue1,svalue2,...,svalueN}
```

Notice that the level of the electronic structure calculations is specified in the *pif.calcs* file. Only when the two files are properly modified (*pif.dlevel* and *pif.calcs*), the high-level calculations can be carried out. This is achieved by executing Pilgrim with the `--hlcalc` option. High-level energies will be stored in `2-PLG_DATA/highlevel.txt`.

Once the high-level calculations are performed, the dual-level methodology is applied using the interpolated single-point energies (ISPE) algorithm. The MEP is calculated at a low-level, but it is corrected with high-level single-point calculations at some locations along the path. The high-level energies are interpolated using a cubic spline. For a reasonably good interpolation, it is important to provide, apart from the stationary points, the high-level energies of the smallest and highest values of *s*, and some additional points along the MEP. The program automatically incorporates those points with 2 additional points at each side of the MEP. Although the *pif.dlevel* file that is generated using the interactive menu can be edited, we recommend to use the default values.

IV

Part Four - Bibliography

7. Bibliography

7.1 General references about methods in Pilgrim.

In this section we list some key references related to the algorithms, procedures and methodologies implemented in Pilgrim.

- General Reviews about Variational Transition State Theory (VTST):
  - D. G. Truhlar, A. D. Isaacson and B. C. Garrett, in *Theory of Chemical Reaction Dynamics*, Vol. 3, M. Baer, ed., CRC Press, Boca Raton, FL, 1985, pp. 65. Generalized Transition State Theory.
  - A. Fernández-Ramos, B. A. Ellingson, B. C. Garrett and Donald G. Truhlar, in *Reviews in Computational Chemistry*, Vol. 23, ed., John Wiley & Sons, Inc., Hoboken, NJ, 2007. Variational Transition State Theory with Multidimensional Tunneling.
  - J. L. Bao and D. G. Truhlar, *Chem. Soc. Rev.*, **46**, 7548 (2017). Variational Transition State Theory: Theoretical Framework and Recent Developments.

- Other Software for VTST calculations:
  - J. Zheng, J. L. Bao, R. Meana- Pañeda, S. Zhang, B. J. Lynch, J. C. Corchado, Y.-Y. Chuang, P. L. Fast, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernández-Ramos, B. A. Ellingson, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, A. Ratkiewicz, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Polyrate 17-C: Computer Program for the Calculation of Chemical Reaction Rates for Polyatomics, University of Minnesota, Minneapolis, MN, 2017.

- Direct dynamics:
  - K. K. Baldridge, M. S. Gordon, R. Steckler, and D. G. Truhlar, *J. Phys. Chem. A*, **93**, 5107 (1989). Ab Initio Reaction Paths and Direct Dynamics Calculations.
  - A. González-Lafont, T. N. Truong and D. G. Truhlar, *J. Phys. Chem.*, **95**, 4618 (1991). Direct Dynamics Calculations with NDDO (Neglect of Diatomic Differential Overlap) Molecular Orbital Theory with Specific Reaction Parameters.

- Minimum Energy Path (MEP):
  - D. G. Truhlar and A. Kuppermann, *J. Am. Chem. Soc.*, **93**, 1840 (1971). Exact Tunneling Calculations.

  - K. Fukui, S. Kato and H. Fujimoto, *J. Am. Chem. Soc.*, **97**, 1 (1975). Constituent Analysis of the Potential Gradient along a Reaction Coordinate. Method and Application to Methane + Tritium Reaction.
  - W. H. Miller, N. C. Handy and J. E. Adams, *J. Chem. Phys.*, **73**, 99 (1980). Reaction Path Hamiltonian for Polyatomic Molecules.
  - M. Page and J. W. McIver, Jr., *J. Chem. Phys.*, **88**, 922 (1988). On Evaluating the Reaction Path Hamiltonian.
  - V. S. Melissas, D. G. Truhlar and B. C. Garrett, *J. Chem. Phys.*, **96**, 5758 (1992). Optimized Calculation of Reaction Paths and Reaction-Path Functions for Chemical Reactions.

- Canonical Variational Transition State Theory (CVT):
  - B. C. Garrett and D. G. Truhlar, *J. Chem. Phys.*, **70**, 1593 (1979). Criterion of Minimum State Density in the Transition State Theory of Bimolecular Reactions.
  - B. C. Garrett and D. G. Truhlar, *J. Phys. Chem.*, **83**, 1072 (1979), (E) 3058 (1979). Generalized Transition State Theory. Classical Mechanical Theory and Applications to Collinear Reactions of Hydrogen Molecules.
  - B. C. Garrett, D. G. Truhlar, R. S. Grev and A. W. Magnuson, *J. Phys. Chem.*, **84**, 1730 (1980). Improved Treatment of Threshold Contributions in Variational Transition-State Theory

- Small-Curvature Tunneling (SCT):
  - R. T. Skodje, D. G. Truhlar and B.C. Garrett, *J. Phys. Chem.*, **85**, 3019 (1981).
  - D.-h. Lu, T. N. Truong, V. S. Melissas, G. C. Lynch, Y.-P. Liu, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph and D. G. Truhlar, *Comput. Phys. Commun.*, **71**, 235 (1992). POLYRATE 4: A New Version of a Computer Program for the Calculation of Chemical Reaction Rates for Polyatomics.
  - Y.-P. Liu, G. C. Lynch, T. N. Truong, D.-h. Lu and D. G. Truhlar, *J. Am. Chem. Soc.* **115**, 2408 (1993). Molecular Modeling of the Kinetic Isotope Effect for the [1,5]-Sigmatropic Rearrangement of cis-1,3-Pentadiene.

- Multi-Structural Variational Transition State Theory (MS-VTST):
  - T. Yu, J. Zheng and D. G. Truhlar, *Chem. Sci.*, **2** 2199 (2011). Multi-structural Variational Transition State Theory. Kinetics of the 1,4-Hydrogen Shift Isomerization of the Pentyl Radical with Torsional Anharmonicity.
  - X. Xu, T. Yu, E. Papajak and D. G. Truhlar, *J. Phys. Chem. A*, **116**, 10480 (2012). Multistructural Variational Transition State Theory: Kinetics of the Hydrogen Abstraction from Carbon-2 of 2-Methyl-1-Propanol by Hydroperoxyl Radical Including all Structures and Torsional Anharmonicity.

- Multi-Path Variational Transition State Theory (MP-VTST):
  - T. Yu, J. Zheng, and D. G. Truhlar, *J. Phys. Chem. A*, **116**, 297 (2012). Multipath Variational Transition State Theory. Rate Constant of the 1,4-Hydrogen Shift Isomerization of the 2-Cyclohexylethyl Radical.
  - R. Meana-Pañeda, A. Fernández-Ramos, *J. Am. Chem. Soc.*, **134**, 346 (2012) (E) 7193 (2012). Tunneling and Conformational Flexibility Play Critical Roles in the Isomerization Mechanism of Vitamin D.
  - J. Zheng and D. G. Truhlar, *Faraday Discuss.*, **157**, 59 (2012). Multi-Path Variational

Transition State Theory for Chemical Reaction Rates of Complex Polyatomic Species: Ethanol + OH Reactions.
  - R. Meana-Pañeda and A. Fernández-Ramos, *J. Chem. Phys.*, **140**, 174303 (2014). Accounting for Conformational Flexibility and Torsional Anharmonicity in the H + CH_3CH_2OH Hydrogen Abstraction Reactions: A Multi-Path Variational Transition State Theory Study
  - J. L. Bao, R. Meana-Pañeda and D. G. Truhlar, *Chem. Sci.*, **6**, 5866 (2015). Multi-Path Variational Transition State Theory for Chiral Molecules: The Site-Dependent Kinetics for Abstraction of Hydrogen from 2-Butanol by Hydroperoxyl Radical, Analysis of Hydrogen Bonding in the Transition State, and Dramatic Temperature Dependence of the Activation Energy.
  - L. Simón-Carballido, T. V. Alves, A. Dybala-Defratyka and A. Fernández-Ramos, *J. Phys. Chem. B*, **120**, 1911 (2016). Kinetic Isotope Effects in Multipath VTST: Application to a Hydrogen Abstraction Reaction.

- Kinetic Monte Carlo (KMC):
  - D. T. Gillespie, *J. Comput. Phys.*, **22**, 403 (1976). A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions.
  - D. T. Gillespie *Annu. Rev. Phys. Chem.*, **58**, 35 (2007). Stochastic Simulation of Chemical Kinetics.

- Dual-level Calculations by Interpolated Single-Point Energies (ISPE):
  - Y.-Y. Chuang, J. C. Corchado and D. G. Truhlar, *J. Phys. Chem. A*, **103**, 1140 (1999). Mapped Interpolation Scheme for Single-Point Energy Corrections in Reaction Rate Calculations and Critical Evaluation of Dual-Level Reaction Path Dynamics Methods.

- Internal Coordinates:
  - E. B. Wilson, Jr., J. C. Decius and P. C. Cross, *Molecular Vibrations*, Dover Publications, Inc., NY, 1955.
  - M. A. Pariseau, I. Suzuki and J. Overend, *J. Chem. Phys.*, **42**, 2335 (1965). Least-Squares Adjustment of Anharmonic Potential Constants: Application to $^{12}CO_2$ and $^{12}CO_2$.
  - D. F. McIntosh, K. H. Michaelian and M. R. Peterson, Can. J. Chem. **56**, 1289 (1978). A Consistent Derivation of the Wilson-Decius s Vectors, Including New Out-of-plane Wag Formulae.
  - M. Challacombe and J. Cioslowski, *J. Chem. Phys.*, **95**, 1064 (1991). Coordinate Transformations of Cubic Force Constants and Transferability of Anharmonic Force Constants in Internal Coordinates.
  - P. Pulay and G. Fogarasi, *J. Chem. Phys.*, **96**, 2856 (1992). Geometry Optimization in Redundant Internal Coordinates.
  - C. F. Jackels, Z. Gu and D. G. Truhlar, *J. Chem. Phys.*, **102**, 3188 (1995). Reaction-Path Potential and Vibrational Frequencies in Terms of Curvilinear Internal Coordinates.
  - Y.-Y. Chuang and D. G. Truhlar, *J. Phys. Chem. A*, **102**, 242 (1998). Reaction-Path Dynamics in Redundant Internal Coordinates.
  - V. Bakken and T. Helgaker, *J. Chem. Phys.*, **117**, 9160 (2002). The Efficient Optimization of Molecular Geometries Using Redundant Internal Coordinates
  - D. F. McIntosh, *Theor. Chem. Acc.*, *125*, 177 (2010). The Determination of Wilson-Decius F Matrix Elements from Cartesian Force Constants.

- Symmetry Numbers:
  - A. Fernández-Ramos, B. A. Ellingson, R. Meana-Pañeda, J. M. C. Marques and D. G. Truhlar, *Theor. Chem. Acc.*, *118*, 813 (2007). Symmetry Numbers and Chemical Reaction Rates.

- Molecular Symmetry:
  - O. Beruski and L. N. Vidal, *J. Comput. Chem.* **35**, 290 (2014). Algorithms for Computer Detection of Symmetry Elements in Molecular Systems.

- Anharmonicity of Coupled Torsions:
  - A. Fernández-Ramos, *J. Chem. Phys.*, **138**, 134112 (2013). Accurate Treatment of Two-Dimensional Non-separable Hindered Internal Rotors.
  - J. Zheng and D. G. Truhlar, *J. Chem. Theory Comput.*, **9**, 1356 (2013) Quantum Thermochemistry: Multistructural Method with Torsional Anharmonicity Based on a Coupled Torsional Potential.
  - J. Zheng, S. L. Mielke, K. L. Clarkson, and D. G. Truhlar, *Comput. Phys. Commun.*, **183**, 1803 (2012). MSTor: A Program for Calculating Partition Functions, Free Energies, Enthalpies, Entropies, and Heat Capacities of Complex Molecules Including Torsional Anharmonicity.
  - J. Zheng, R. Meana-Pañeda and D. G. Truhlar, *Comput. Phys. Commun.*, **184**, 2032 (2013). MSTor Version 2013: A New Version of the Computer Code for the Multistructural Torsional Anharmonicity with a Coupled Torsional Potential.
  - J. Zheng, S. L. Mielke, J. L. Bao, R. Meana-Pañeda, K. L. Clarkson and D. G. Truhlar, MSTor Computer Program, Version 2017, University of Minnesota, Minneapolis, MN, 2017.
  - L. Simón-Carballido, J. L. Bao, T. V. Alves, R. Meana-Pañeda, D. G. Truhlar and A. Fernández-Ramos, *J. Chem. Theory Comput.* **13**, 3478 (2017). Anharmonicity of Coupled Torsions: The Extended Two-Dimensional Torsion Method and Its Use To Assess More Approximate Methods.
  - D. Ferro-Costas, M. N. D.S. Cordeiro, D. G. Truhlar and A. Fernández-Ramos, *Comput. Phys. Commun.*, **232**, 190 (2018). Q2DTor: A Program to Treat Torsional Anharmonicity Through Coupled Pair Torsions in Flexible Molecules.

- Electronic Structure Software (ESSO):
  - M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, J. A. Pople, *Gaussian* 03, Revision C.02 Gaussian, Inc., Wallingford CT, (2004).
  - M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman,

G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, D. J. Fox, *Gaussian* 09, Revision B.01, Gaussian, Inc., Wallingford CT, (2010).

  - M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, D. J. Fox, *Gaussian* 16, Revision A-03, Gaussian, Inc., Wallingford CT, (2016).
  - F. Neese Wiley Interdiscip. Rev. Comput. Mol. Sci. **2**, 73 (2012). The ORCA Program System.
  - F. Neese Wiley Interdiscip. Rev. Comput. Mol. Sci. **8**, 1 (2018). Software Update: The ORCA Program System, Version 4.0.

7.2 Specific references for specific methods in Pilgrim.

In this section we give specific suggestions of citations for methods used.

- Conventional Transition State Theory
  - H. Eyring, *J. Chem. Phys.*, **3**, 107 (1935). The Activated Complex in Chemical Reactions.

- Canonical Variational Transition State Theory
  - B. C. Garrett and D. G. Truhlar, *J. Chem. Phys.*, **70**, 1593 (1979). Criterion of Minimum State Density in the Transition State Theory of Bimolecular Reactions.

- Minimum Energy Path (MEP) in Isoinertial Coordinates
  - D. G. Truhlar and A. Kuppermann, *J. Am. Chem. Soc.*, **93**, 1840 (1971). Exact Tunneling Calculations.

- Generalized Normal Modes in Nonredundant Internal Coordinates Along a Reaction Path
  - C. F. Jackels, Z. Gu and D. G. Truhlar, *J. Chem. Phys.*, **102**, 3188 (1995). Reaction-

Path Potential and Vibrational Frequencies in Terms of Curvilinear Internal Coordinates.

- Generalized Normal Modes in Redundant Internal Coordinates Along a Reaction Path
  - Y.-Y. Chuang and D. G. Truhlar, *J. Phys. Chem. A*, **102**, 242 (1998). Reaction-Path Dynamics in Redundant Internal Coordinates.

- Quantized-reactant-states tunneling
  - S. E. Wonchoba, W.-P. Hu, D. G. Truhlar in *Theoretical and Computational Approaches to Interface Phenomena* (Eds.: H. L. Sellers, J. T. Golab), Plenum, NY 1994, pp. 1. Reaction Path Approach to Dynamics at a Gas-Solid Interface: Quantum Tunneling Effects for an Adatom on a Non-Rigid Metallic Surface

- Small Curvature Tunneling
  - Y.-P. Liu, G. C. Lynch, T. N. Truong, D.-h. Lu and D. G. Truhlar, *J. Am. Chem. Soc.* **115**, 2408 (1993). Molecular Modeling of the Kinetic Isotope Effect for the [1,5]-Sigmatropic Rearrangement of cis-1,3-Pentadiene.

- Multi-Structural VTST (MS-VTST)
  - T. Yu, J. Zheng, and D. G. Truhlar, *J. Phys. Chem. A*, **116**, 297 (2012). Multipath Variational Transition State Theory. Rate Constant of the 1,4-Hydrogen Shift Isomerization of the 2-Cyclohexylethyl Radical.

- Multi-path VTST (MP-VTST)
  - T. Yu, J. Zheng, and D. G. Truhlar, *J. Phys. Chem. A*, **116**, 297 (2012). Multipath Variational Transition State Theory. Rate Constant of the 1,4-Hydrogen Shift Isomerization of the 2-Cyclohexylethyl Radical.

- MS-T Method for Torsional Anharmonicity
  - J. Zheng and D. G. Truhlar, *J. Chem. Theory Comput.*, **9**, 1356 (2013) Quantum Thermochemistry: Multistructural Method with Torsional Anharmonicity Based on a Coupled Torsional Potential.

- Q2DTor Method for Torsional Anharmonicity
  - D. Ferro-Costas, M. N. D.S. Cordeiro, D. G. Truhlar and A. Fernández-Ramos, *Comput. Phys. Commun.*, **232**, 190 (2018). Q2DTor: A Program to Treat Torsional Anharmonicity Through Coupled Pair Torsions in Flexible Molecules.