

# SANDIA REPORT

SAND2012-8095

Unlimited Release

Printed September 2012

## Automated exploration of the mechanism of elementary reactions

Judit Zádor and Habib N. Najm

Prepared by  
Sandia National Laboratories  
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831

Telephone: (865) 576-8401  
Facsimile: (865) 576-5728  
E-Mail: [reports@adonis.osti.gov](mailto:reports@adonis.osti.gov)  
Online ordering: <http://www.osti.gov/bridge>

Available to the public from

U.S. Department of Commerce  
National Technical Information Service  
5285 Port Royal Rd.  
Springfield, VA 22161

Telephone: (800) 553-6847  
Facsimile: (703) 605-6900  
E-Mail: [orders@ntis.fedworld.gov](mailto:orders@ntis.fedworld.gov)  
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2012-8095  
Unlimited Release  
Printed September 2012

# Automated exploration of the mechanism of elementary reactions

Judit Zádor  
Combustion Chemistry Department

Habib N. Najm  
Reacting Flow Department

Sandia National Laboratories  
P.O. Box 969, MS 9055  
Livermore, California 94550

## Abstract

Optimization of new transportation fuels and engine technologies requires the characterization of the combustion chemistry of a wide range of fuel classes. Theoretical studies of elementary reactions — the building blocks of complex reaction mechanisms — are essential to accurately predict important combustion processes such as autoignition of biofuels. The current bottleneck for these calculations is a user-intensive exploration of the underlying potential energy surface (PES), which relies on the “chemical intuition” of the scientist to propose initial guesses for the relevant chemical configurations. For newly emerging fuels, this approach cripples the rate of progress because of the system size and complexity. The KinBot program package aims to accelerate the detailed chemical kinetic description of combustion, and enables large-scale systematic studies on the sub-mechanism level.



## **ACKNOWLEDGMENTS**

We thank Dr. Oliver Welz (8353) for his helpful comments and contributions to the testing of the code.

## CONTENTS

1. Introduction.....	8
2. KINBOT .....	10
2.1. Program structure.....	11
2.1.1. Structural analysis .....	12
2.1.2. Reaction path search.....	16
Exploration of the dihedral space.....	22
2.2. Input and keywords.....	24
2.3. Running KinBot and reading the output.....	27
3. Further development plans .....	28
4. References.....	29
Distribution.....	31

## FIGURES

Figure 1. Schematics of the reaction pathways during low-temperature autoignition. ....	8
Figure 2. Currently KinBot covers reaction search only up to level one. ....	10
Figure 3. Energy along the bond scanned.....	18
Figure 4. Searching for beta scission TS. ....	18
Figure 5. Searching for internal hydrogen transfer TS. ....	20
Figure 6. Searching for direct HO <sub>2</sub> -elimination TS.....	21
Figure 7. Searching for cyclic ether forming TS. ....	22

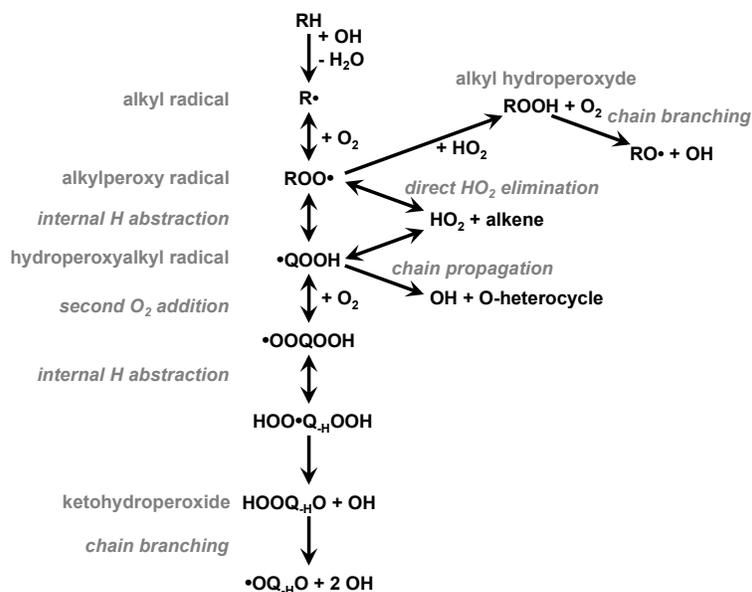
## NOMENCLATURE

PES	Potential Energy Surface
TS	Transition State
DOF	Degrees of Freedom

# 1. INTRODUCTION

With the great advancement of electronic structure calculations and the refinement of statistical theories to describe chemical reactions, theoretical chemical kinetics became an indispensable tool to predict accurate rate coefficients and branching fractions that are relevant in gas-phase systems, such as in combustion and atmospheric processes, or in the interstellar media (see e.g. [1]).

There are several factors that limit our current predictive power in these calculations, and systematic research in exploring the limitations as well as the prospects of model refinement has just begun recently [2,3]. The sources of uncertainties in such calculations can be categorized as (a) the limited ability to calculate accurate *ab initio* parameters (energies, frequencies and geometries), and (b) the simplifications we employ in the dynamics part of the calculation (i.e. transition state theory and energy transfer model). However, there is a third, but much less emphasized aspect to the difficulties: the complexity of mapping the possible reaction channels. Even for moderately large systems, that is, structures containing 4-5 heavy atoms, the number of reaction channels can be so large that most of the human effort to achieve a reliable characterization goes into the exploration of the pathways. As an example, in the propene + OH reaction (4 heavy and 7 hydrogen atoms) we found 7 intermediate wells, 39 saddle points and 20 bimolecular products [4]. Moreover, not only the chemical pathways, but the conformational space of the participating molecules and radicals has to be explored as well.



**Figure 1. Schematics of the reaction pathways during low-temperature autoignition.**

There is a large literature on strategies to explore chemical pathways (for example [5-11]), but there are no readily available programs that can be used to perform this task in a way that is most useful and efficient to the gas-phase chemical kinetics community. The goal of this project was

to create a computer code that provides an efficient way to explore chemical pathways automatically for reactions that are relevant in gas-phase chemical problems. The code is designed in a modular manner that allows future extensions to incorporate more reaction types. The code heavily builds on the incredibly well working “chemical intuition” that gas-phase chemical kineticists have developed in the past ~100 years. This intuition has been and is constantly checked against both experimental and theoretical evidence, and as such, should rather be called rigorously proved scientific fact than intuition and provide guidance in the high dimensional space that represents a many-atom system.

For the current, 1.0 version of the code, we focused on the chemistry that occurs during low-temperature autoignition processes [12], illustrated in Figure 1. This reaction system represents a wide variety of chemical transformations, which play a role also in atmospheric chemistry. The code will be continuously extended by new formal reaction types, but already covers a wide variety of chemistry as shown later.



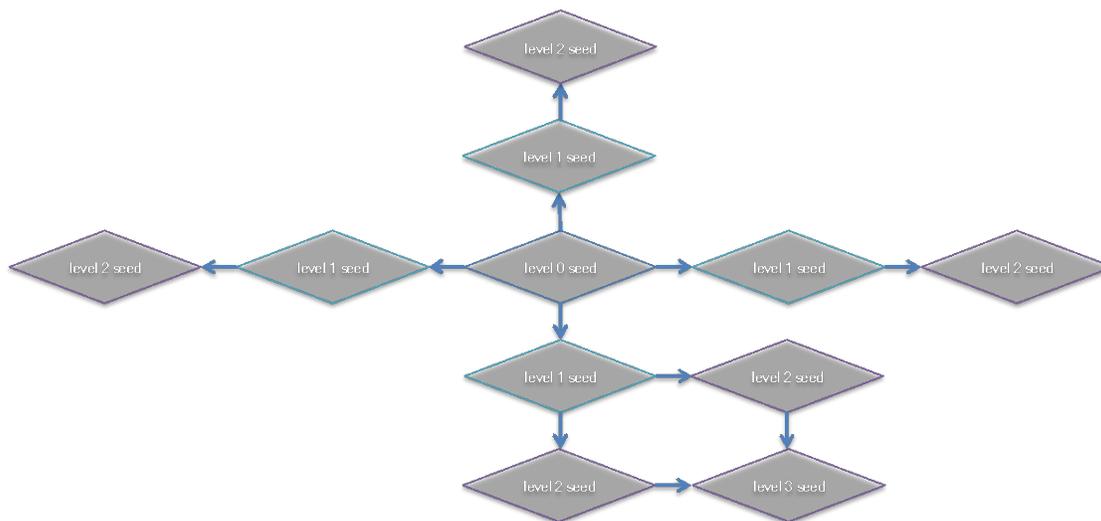
## 2. KINBOT

KinBot is a parallel C code that enables the automatic exploration of reaction channels and the conformer space that is relevant in combustion and atmospheric chemistry. It combines a growing network approach with a chemistry-based method. The name KinBot derives from the contraction of the words “kinetic“ and “bot“, where bot is used in the sense that it is an autonomous computer program that performs time-consuming tasks. Bots are often used on the world wide web to explore the network systematically and automatically.

In the growing-network approach the network of reaction pathways is grown starting from a structure, which will be referred to as seed throughout this document. In this way the connectivity in this network is inherently obtained during the search, and it makes it easy to leave off unimportant branches as the search goes. However, the method requires an efficient way to crawl in the network and on the PES.

Crawling on the PES in our code is based on the enormous amount of chemical knowledge that has been accumulated about gas-phase reactions. Many reaction types that are called by different names, in fact involve the same morphological changes in a molecule, therefore, it is likely that with a relatively small number of rules a large fraction of chemical reactions can be covered.

The current version of KinBot does not crawl in the network of reactions; it only goes one level away from the seed, but in every possible direction. This is shown schematically in Figure 2. In KinBot 1.0 the user has to resubmit the product structures manually using the level 1 seed(s) as the new level 0 seed(s) to discover further regions on the reaction map. Mapping the whole network will be completed for the next version of this code.



**Figure 2. Currently KinBot covers reaction search only up to level one.**

The code is developed to work on shared memory Unix clusters and works in conjunction with the Gaussian09 quantum chemical package [13]. The code is compiled with the gcc compiler and uses the lgsl, lgslcblas, lm and pthread standard C libraries. The pthread library enables

parallelism on the shared memory architecture. The program currently allows studying compounds containing C, H and O atoms only.

## 2.1. Program structure

The following meta code shows the overall structure of the KinBot code. The code creates inputs and calls the Gaussian 09 program package [13] at several points, which are marked by /G09 in the meta code. All of the Gaussian calls are done via writing regular text-based input files, which are supplied to Gaussian calls from within the code, in parallel fashion, whenever possible. At each step, the results are tested for errors. If they occur, the program terminates with an error message, but these branches are not shown in this meta code for the sake of clarity.

Parallelism in Kinbot is achieved on two levels to maximize processor usage:

1. When more than one independent calculation is done (e.g. conformational sampling), the jobs are submitted using the pthread library.
2. Whenever there are more cores available than the number of jobs, the %NProcshared=<np> Gaussian keyword is used, where np is the number of cores per a specific job, which is determined automatically.

```
*** OVERALL PROGRAM STRUCTURE ***

read keywords and seed geometry

analyze seed
  make distance matrix
  make bond matrix
  determine if structure is unimolecular or bimolecular
  find radical center
  determine whether it is TS
  find cyclic substructures in seed
  find all dihedral angles in seed

optimize seed /G09
  read optimized seed geometry and analyze optimized seed as
  above

perform reaction pathway search
  beta scission pathways /G09

  all other (non-beta) scission pathways /G09

  internal hydrogen abstraction pathways /G09

  direct H2O2 elimination pathway /G09

  cyclic ether formation pathway /G09
```

```
output optimized geometries, frequencies and rotational
constants of both products and transition states for all
pathways found
```

```
sample randomly the conformational space of seed
optimize structures /G09
analyze results
```

```
create 1-D hindered rotor scan for the lowest energy conformers
perform 1-D scans /G09
```

In the following subsections, more details about the above meta code are presented.

### 2.1.1. *Structural analysis*

The structural analysis starts with the construction of the  $N$ -dimensional symmetric square distance matrix:

```
*** DISTANCE MATRIX ***

for all atoms i:
  for all atoms j:
    d_ij = Euclidian distance between atom i and j
```

It is followed by the construction of the also  $N$ -dimensional bond matrix, where the interatomic distances are compared to some standard values with a large (50%) additional margin in order to be able to detect bonds that are parts of TS's or other loosely associated parts of structures. A and B in the meta code mean specific atom types, i.e. carbon, hydrogen or oxygen.

```
*** BOND MATRIX ***

if i is atom type A and j is atom type B and d_ij < d_AB * 1.5
  i and j are bonded (b_ij = 1)
else
  i and j are not bonded (b_ij = 0)

standard d_AB values in Angstrom:
d_CC = 1.5
d_CO = 1.4
d_CH = 1.1
d_OO = 1.5
d_OH = .95
```

It is possible to provide a seed to the code that has two fragments at relatively large distances, which could result in two separate fragments based even on the 50% extra length on the standard

interatomic distances. This is the case for instance for abstraction reactions. To avoid this, the code checks whether the structure is in one piece or not as represented in the initial bond matrix. This is done by the following depth-first walking algorithm:

```
*** TEST FOR NUMBER OF FRAGMENTS ***

select arbitrary atom

(1) if the current atom has a yet unvisited bond:
    walk along it
    arrive at atom on the other end of the bond
    mark this atom as visited
    goto (1)
else:
    if at initial atom and all its bonds have been explored:
        walking is done
    else:
        retract to previous atom
        goto (1)

if not all atoms were visited during the walk:
    the structure has unconnected parts
    create an extra bond between the closest atoms of the two
    unconnected parts
else:
    the structure is in one piece
```

The next step is the determination of the multiple bonds, if any, for neutral species. For charged species it is not straightforward how to determine the location of double or triple bonds unambiguously, as it would be required to know where the positive or negative charge sits. In future versions of KinBot the location of the charge will be determined based on Mulliken or other partial charge information, but it is currently not implemented.

```
*** MULTIPLE BONDS ***

for all atoms i:
    nb_i = the number of bonds of atom i
if the structure has no charge:
    for all atoms i:
        if atom i is type A and nb_i < nb_A:
            there are nb_i - nb_A excess electrons on atom i
            create multiple bonds from excess electrons on neighboring
            atoms by setting b_ij to 2 or 3 (double or triple bond)
else:
    do not look for multiple bonds

standard nb_A values:
```

```
nb_C = 4
nb_O = 2
nb_H = 1
```

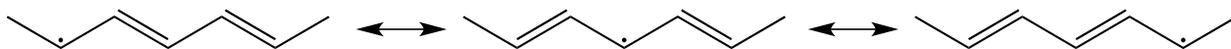
To determine the spin multiplicity of a structure and whether the structure is a radical, with special emphasis on resonance stabilized structures the algorithm below is used. Again, charged structures will be dealt with in a later version of the code, as the determination of the radical site would require the location of the charge in the structure. It is possible to define the multiplicity in the input as well, which overrides the calculated multiplicity, unless there is a conflict.

```
*** RADICAL CENTER ***

if the structure has no charge:
  if after multiple bond creation, excess electrons exist:
    if number of excess electrons = 1:
      if double bond in beta position to the radical center:
        it is a resonance stabilized structure
        multiple possible radical centers are registered
      else:
        doublet radical, with radical center on where excess
        electron is
    if number of excess electrons > 1:
      currently only doublet radicals are supported for reaction
      search, only conformational search will be performed
  else:
    singlet closed shell molecule
else:
  multiplicity is determined only based on total number of
  electrons

if user defined multiplicity is provided and is consistent with
the number of electrons in the system:
  use user defined multiplicity
```

KinBot treats for instance the below radical as a structure with three possible radical sites:



and accordingly, searches for reaction pathways assuming three different radical site locations.

As describe earlier, in the determination of the bonds a generous 50% cutoff was applied to the standard interatomic distances. However, a 50% elongation in the bond lengths is not realistic for stable species, and is signifying that the bond is about to break, i.e. the structure is likely to be in a transition state. The following rule can be applied to assign a bond to be in transition state:

```
*** TS BASED ON BOND LENGTH ***
```

```
if d_ij > 1.15 * d_AB and d_ij < 1.5 * d_AB:
    mark the structure as TS
```

Another possibility to detect transition states is the excess number of bonds relative to the nb\_A values:

```
*** TS BASED ON TOO MANY BONDS ***

if nb_i > nb_A:
    mark the structure as TS
```

Both tests are used in KinBot to identify transition states.

Cyclic structures are notoriously hard to deal with and usually require special treatment in various parts of the calculations. The detection of the cycles are also done with a depth-first walking algorithm. Currently only one cycle is supported.

```
*** CYCLE SEARCH ***

select arbitrary atom

(1) if the current atom has a yet unvisited bond:
    walk along it
    if the atom where arriving was visited previously and the
    current step is not part of a retraction:
        the structure contains a cycle
        keep track of the steps of the walk
        goto (1)
else:
    if we are not at the initial atom:
        retract to previous atom
        goto (1)
    else:
        walking is done

if the number of cycles found = 1:
    register atoms in the cycle
if number of cycles found > 1:
    terminate program
```

Finally, the bonds around which torsional rotation is possible have to be found.

```
*** ROTATABLE BOND SEARCH ***

for all single bonds in structure:
```

```
if bond is not part of a cycle and has at least one atom
attached to both ends:
    mark it as a rotatable bond
```

### 2.1.2. Reaction path search

There are several archetypical reaction searches that are implemented in KinBot, although the list is far from complete. However, we have shown that despite the variability of real chemical systems, relatively few formal reaction rules can cover a wide range of important reactions, and that it is possible to design robust and efficient geometrical manipulation algorithms using redundant internal coordinates and stepwise relaxation.

All reaction search algorithms start with the identification of a key motif in the structure. If this motif is present, the specific reaction can take place. The search algorithm has to identify all possible motifs, and exclude the ones that are identical from a chemical point of view. An example of a motif is H-.....X\*, which means that the code has to search the structure for all substructures in the molecule that start with a hydrogen atom and end with an atom that is a radical (\*), but whose atom type is irrelevant (X). One can think of a more well specified motif search, e.g. H-X-X-X-X\*, which is the same, except it is prescribed that H and X\* are separated by exactly four bonds. And of course, it should be also possible to search for a motif such as H-C-C-O-O\*.

We created a general motif-finding algorithm, which takes wildcards as well as direct atom type specifications as input. The algorithm is recursive and can do either an overall motif search or a motif search just starting from a given atom (e.g. starting from the radical). In the latter case the initial outer for loop is omitted.

```
*** MOTIF SEARCH ***

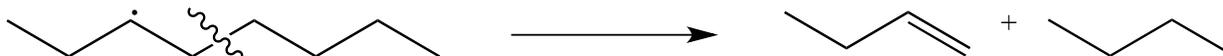
motif search (motif specified as string):
  if the search just started:
    for all atoms i:
      for all atoms j:
        motif search
  else:
    if atom i not yet visited:
      if atom i matches the current motif element:
        if atom j is connected to atom i:
          if atom j matches the next motif element:
            if not at the last motif element:
              i = j
              for all atoms j:
                motif search
            else:
              return (success)
          else:
            return (failure)
```

```

        return
    else:
        return
    else:
        return
else:
    return

```

A general beta scission pathway is when a radical is dissociating into two fragments along a bond that is in a beta position relative to the radical center, and forms a double bond and another, but smaller radical. An example is shown on the below scheme:



There are several reaction types that fall under the general beta scission pathway:

- simple C-C bond cleavage
- H-atom elimination
- ROO dissociation to R and O<sub>2</sub>
- QOOH dissociation to HO<sub>2</sub> + alkene
- aldehyde/ketone formation from alkoxy radicals
- etc.

Since the bonding pattern of the seed is known from the structural analysis, finding beta scission pathways is straightforward. Note that only one beta scission pathway is investigated when the leaving fragment is a hydrogen atom and there are several hydrogen atoms are connected to the same atom (e.g. in case of a methyl group only one, arbitrarily selected hydrogens are removed).

```

*** BETA SCISSION ***

```

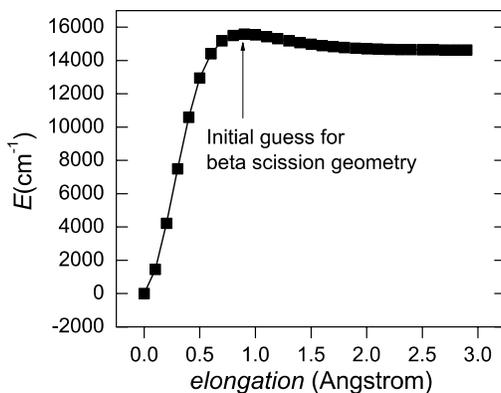
```

for all X1---X2---X3* motifs:
    if X1 is hydrogen and a hydrogen connected to X2 was already
        looked at:
        next
    create input to scan energies along X1---X2 bond
    calculate energies along this coordinate /G09
    read in energies from the scan
    if the energy has a maximum along the scan coordinate:
        create an initial guess with the bond distance that belonged
            to the maximum energy
        optimize initial guess /G09
    else:
        this is a barrierless reaction
        do not try to find saddle point

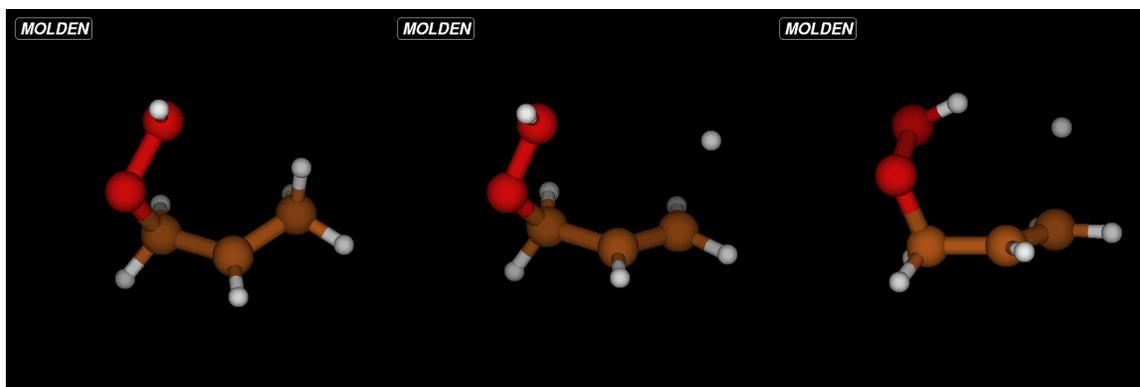
```

An example of a beta scission scan is shown in Figure 3. The potential maximum was found when the bond was elongated by 0.9 Angstrom, and this geometry was used as an initial guess

for the transition state search, see Figure 4, which shows the structure of the seed, the structure at the maximum of the scan and the geometry of the saddle point.



**Figure 3. Energy along the bond scanned.**



**Figure 4. Searching for beta scission TS.**

All non-beta scission bond-breaking transition states are searched for as well in a similar fashion. In principle, these searches do not have to be treated separately from the beta scission, but when treated separately, the user has the option not to search for these transition states, and they can be numerous and are generally higher in energy than the beta scission ones. The separate treatment does not add complexity or extra computational time to the code.

```

*** NON-BETA SCISSION ***

for all bonds:
  if bond is in beta position to radical center:
    next
  if one of the atoms along the bond is a hydrogen it was
  already looked at:
    next

```

```

create input to scan energies along bond
calculate energies along this coordinate /G09
read in energies from the scan
if the energy has a maximum along the scan coordinate:
    create an initial guess with the bond distance that belonged
    to the maximum energy
    optimize initial guess /G09
else:
    this is a barrierless reaction
    do not try to find saddle point

```

Internal hydrogen abstraction is when a hydrogen atom is transferred from one heavy atom to another within a molecule. Examples of internal H-abstraction pathways are the isomerization of alkyl radicals and the ROO/QOOH isomerization that is a key process in low-temperature autoignition, see Figure 1. Depending on the distance between the initial and final location of the H-atom, we can talk about three-, four-, five-, etc-membered transition states that connect the initial and final structures.

KinBot uses a robust geometry manipulation algorithm to create a good initial guess for the TS, and uses a unified algorithm for the four- to eight-membered ring versions of this TS, outlined in the following meta code. In principle the same method could be used for larger rings as well, but larger rings have typically no relevance to gas-phase kinetics.

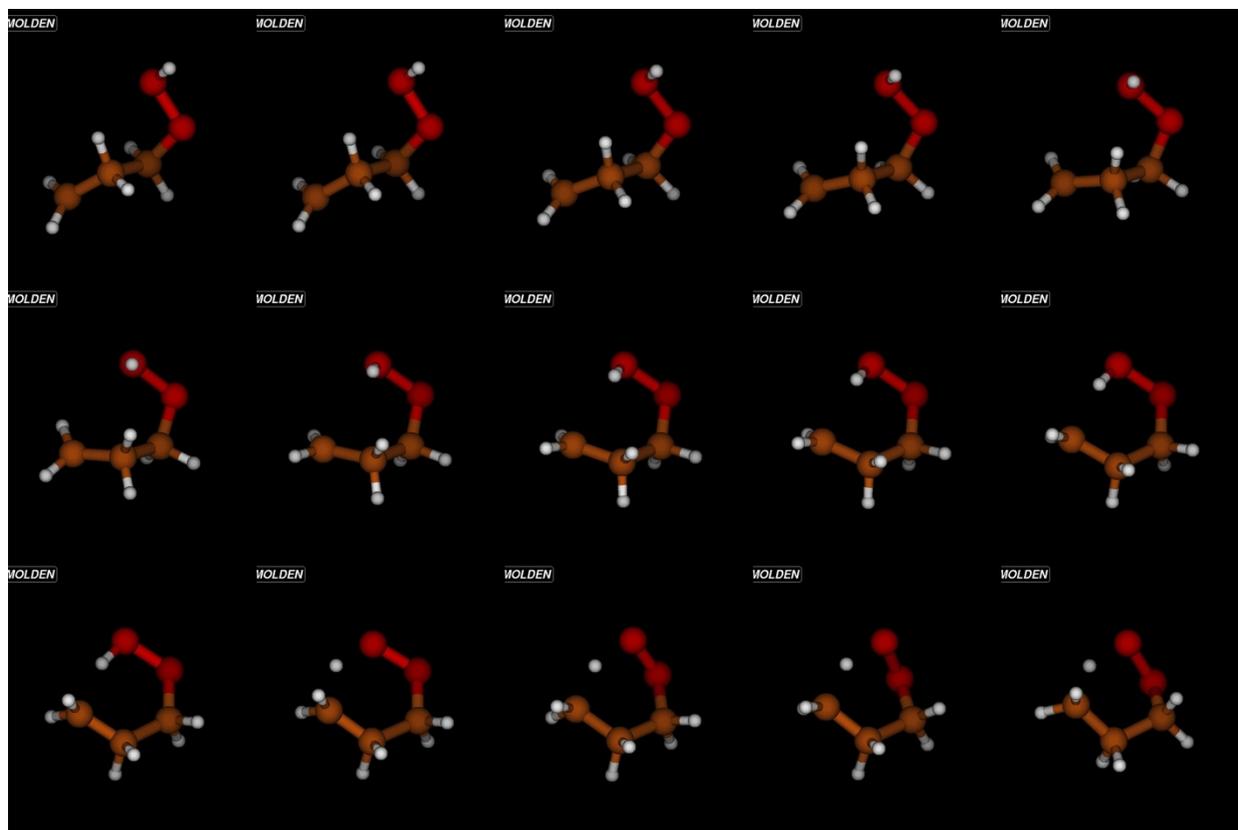
```

*** INTERNAL H-ABSTRACTION FOR 4- TO 8-MEMBERED RINGS ***

for ring size 4 to 8:
  for all H---X1---...---X(ring size-2)---X* motifs:
    calculate current dihedrals in motif
    create input for Gaussian prescribing these steps:
      //create a flat ring from H, X1, ..., X* in 10 steps, while
      relaxing the other DOF//
      //set angles in the ring to that of an equiangular
      polygon, while relaxing the other DOF except the already
      flat dihedrals//
      //fix H---X1 and H---X* distances at 1.35 A, while
      relaxing all other DOF//
      //full optimization to search for TS//
    obtain TS structure /G09

```

An example of this algorithm is visualized in Figure 5 for a reaction that connects the 3-propyl-1-hydroperoxy radical to the *n*-propylperoxy radical. The steps in which an internal H-abstraction pathway transition state is found: the first 11 frames show the closing of the ring, the next shows the setting of the angles, followed by two frames for ring relaxation. The last frame is the optimized transition state structure.



**Figure 5. Searching for internal hydrogen transfer TS.**

For three-membered ring structures the algorithm is simpler:

```

*** INTERNAL H-ABSTRACTION FOR 3-MEMBERED RINGS ***

for all H---X1---X* motifs:
  create input for Gaussian prescribing these steps:
    //place H atom 1.35 A from X1 and X*//
    //full optimization to search for TS//
  obtain TS structure /G09

```

A less general, but also very important pathway under low-temperature conditions is the so-called direct HO<sub>2</sub>-elimination pathway. In this reaction the slightly modified version of the internal H-abstraction algorithms was used.

```

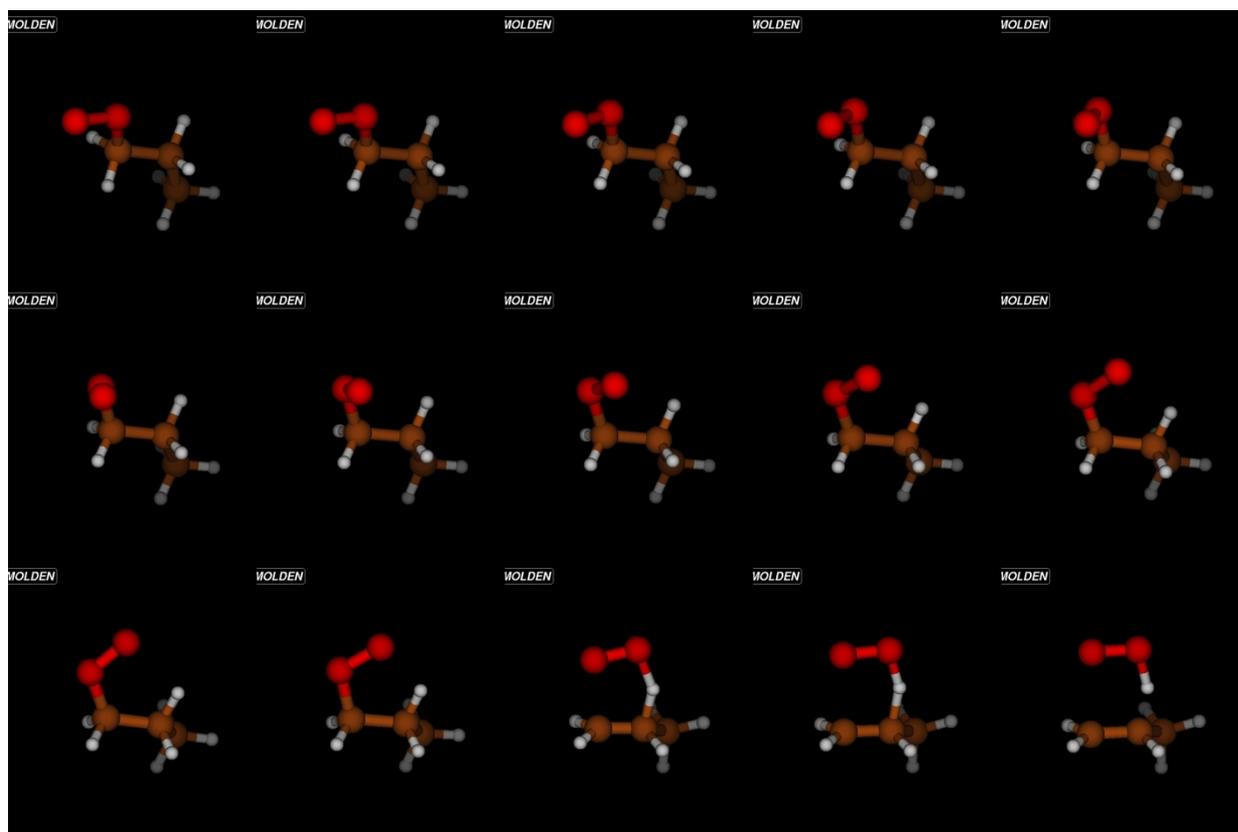
*** DIRECT HO2 ABSTRACTION ***

for all H---X1---X2---O---O* motifs:
  calculate current dihedrals in motif
  create input for Gaussian prescribing these steps:

```

```
//create a flat ring of H, X1, X2, O, O* in 10 steps,  
optimizing the other DOF//  
//set the bond length in the motif to that of a typical HO2  
elimination TS, while optimizing the other DOF //  
//full optimization to search for TS//  
obtain TS structure /G09
```

The procedure is visualized in Figure 6 using the example of the *n*-propylperoxy radical. The steps in which a direct HO<sub>2</sub>-elimination pathway transition state is found: the first 12 frames show the creation of a planar H-C-C-O-O structure, the next three show the setting of the angles and bond distances to the prescribed values. The last frame is the optimized transition state structure.



**Figure 6. Searching for direct HO<sub>2</sub>-elimination TS.**

Cyclic ether formation is another important reaction during low-temperature autoignition. The transition states for cycle sizes three to eight are searched for with the same, robust algorithm.

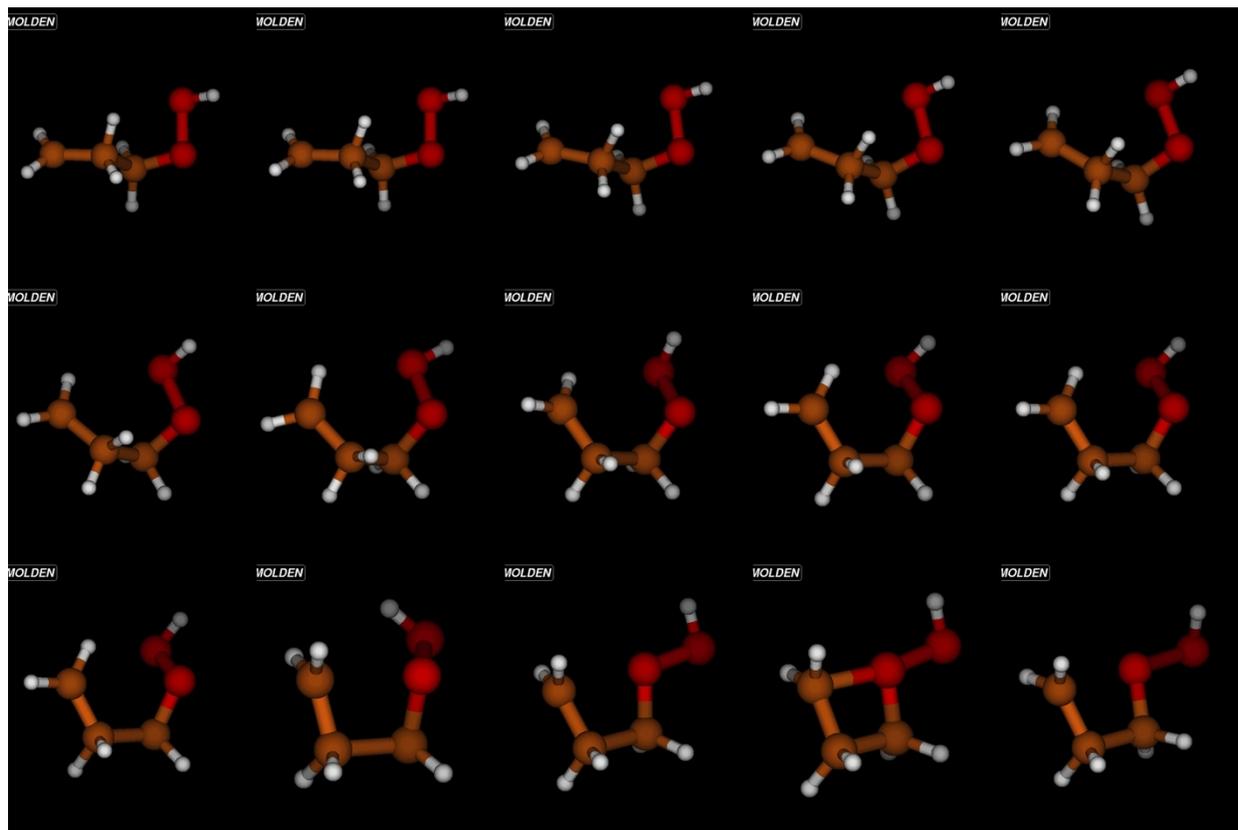
```
*** CYCLIC ETHER FORMATION ***  
  
for ring size 3 to 8:
```

```

for all H---O1---O2---X1---...---X(ring size-2)---C* motifs:
  calculate current dihedrals in motif
  create input for Gaussian prescribing these steps:
    //create a flat ring of H, O1, O2, ..., C* optimizing the
    other DOF//
    //set angles in the ring to that of an equiangular
    polygon, optimizing the other DOF//
    //fix the C*-O2 distance, optimizing the other DOF//
    //full optimization to search for TS//
  obtain TS structure /G09

```

The procedure is visualized in Figure 7 on the example of 3-propyl-1-hydroperoxy radical going to a transition state that will eventually lead to an oxetane molecule and an OH radical. The steps in which a transition state leading to cyclic ether is found: the first 12 frames show the creation of a planar H-C-C-O-O structure, the next three show the setting of the angles and bond distances to the prescribed values. The last frame is the optimized transition state structure.



**Figure 7. Searching for cyclic ether forming TS.**

### *Exploration of the dihedral space*

For the calculation of rate coefficients it is very important to explore the conformational space of each stable species and transition states. This exploration serves two purposes:

1. To find the lowest energy conformer
2. To find all (or most) of the conformers to accurately account for anharmonicities and obtain accurate partition functions.

KiBot can explore the dihedral space of both stable species and transition states in an automated fashion. Currently there is only a simple Monte Carlo-type sampling algorithm implemented. It can be easily modified either to accommodate more efficient random sampling strategies, e.g. Latin hypercube sampling or other, directed sampling methods.

In KinBot, rotation is allowed only around single bonds that are not part of a cycle. If the structures are part of a cycle, a slightly different strategy is employed to explore the conformational space of the cycle parallel to the conformational space of the non-cyclic components. There are various strategies in the literature to generate cyclic structures in different conformers, which is not reviewed in this report (e.g. [14-16]). However, it has to be noted that most of the methods are oriented towards larger cycles than the ones typically occur in gas-phase chemistry. Initial tests have shown that the relatively simple strategy based on Ref. [15] provides a sufficient sampling of this space for our purposes. Further algorithms can be built into KinBot in the future if necessary.

The following meta code presents the combined non-cyclic and cyclic sampling strategy as implemented in KinBot. The calculations are run in batches of  $2^n$ , where  $n$  is defined as shown below, and the total number of samples is maximized at  $3^n$ . Our tests have shown that such a sample size is sufficient for most structures to explore the conformational space. Also, we have found that the convergence of the Gaussian calculations is both faster and less prone to errors if the dihedrals are modified not at once but in steps. We found that 6 steps are sufficient.

```

*** RANDOM CONFORMATIONAL SAMPLING ALGORITHM ***

if the structure contains no cycle:
  n = dihedrals around single bonds that are not in cycle
else:
  n = dihedrals around single bonds that are not in cycle + 3

(1) for 2^n samples:
  for all dihedrals i defined around single bonds that are not
  part of a cycle:
    generate a random number between -Pi and Pi, -> d_i
  if the structure contains a cycle:
    calculate the average value of the absolute value of the
    dihedrals along the cycle -> a
    generate a random integer between 0 and 3 -> r
    for r randomly selected bonds j in the cycle:
      generate a random number between -2*a and 2*a, -> c_j
  create Gaussian input:
  //in 6 steps modify the all non-cyclic dihedrals by d_i
  and the randomly selected cycle dihedrals by c_j//
  //full optimization//

```

```

execute 2^n calculations /G09

select lowest energy conformer from current 2^n calculations
if already more than 2^n calculations were done:
  if the current lowest energy conformer is not lower in
  energy than the one from the previous calculations
    lowest energy conformer found
  if sample size reached 3^n:
    the user is prompted to check the problem
  else: goto (1)
summarize the results of the sampling procedure

```

The dihedral space can be further characterized by scanning the dihedral angles of the lowest energy conformer one by one. This procedure is necessary to certain parts of a statistical rate calculation, and KinBot can perform these scans automatically as well. Part of the strategy is to build a Z-matrix, in which the atoms that are in fragments attached to one end of the rotating bond do not have coordinate dependencies on the ones that are in fragments attached to the other end of the bond, and vice versa. Dihedral scans are only allowed for bonds that are not part of a ring.

```

*** 1-D DIHEDRAL SCANS ***

for all dihedrals around single bonds and not in cycle:
  build a Z-matrix representation of the structure, in which
  there are no dependencies across the rotating bond
  create Gaussian input:
    //scan around dihedral angles using Z-matrix//
  execute scan /G09
  read in energies along the scan
  fit Fourier series to the obtained energy vs. angle values

```

## 2.2. Input and keywords

KinBot requires two input files. One (will be referred to as seedfile) contains the structure of the seed in Cartesian coordinates in Angstrom units in the following format, and can be named arbitrarily, but without an extension (e.g. ethyl in this case). The file takes the format below, containing the number of atoms and a title followed by the Cartesian coordinates:

```

7
ethyl radical
C    0.000000    0.000000    0.000000
H    0.000000    0.000000    1.089000
C    1.367075    0.000000   -0.483328
H   -0.513360   -0.889165   -0.363000
H   -0.513360    0.889165   -0.363000

```

H	1.367079	0.000000	-1.572328
H	1.880433	0.889165	-0.120326

The second input file (will be referred to as keywordfile), which can also be named arbitrarily, contains the keywords governing the calculations. The currently available options are listed in the following. Note that most keywords have sensible defaults and the order of the keywords and their capitalization is not important. However, currently no comments are allowed in the input. The keywords that do take arguments must be followed by the values on the very next line. The type of the argument is shown here in brackets.

`Title` (string): Compulsory keyword, which has to be followed by the name of the seedfile.

`Low` (string): The low level of theory (method/basis set) that is used in intermediate calculations, e.g. HF/STO-3G. Does not affect the precision of the final results, however, if chosen poorly, it can lead to convergence problems in the QC calculations. On the other hand, a sensible lower level theory can speed up the search. The string will be directly copied into the Gaussian files. Check the Gaussian manual for further details. The default is B3LYP/6-31G.

`OptL` (string): Additional keywords for the intermediate calculations that are to be applied to the `Opt` keyword of a Gaussian calculation, e.g. `MaxCycle=300`. `Tight` is not allowed, and if more than one keyword is desired, they should be listed separated by a comma without spaces. The default is none.

`ExtL` (string): Additional keywords for the intermediate calculations that are added to the header for the Gaussian input, for instance `SCF=QC`. `Freq` is not allowed, and if more than one keyword is desired, they should be listed separated by a comma without spaces. The default is none.

`PostL` (string): Additional keywords that are to be placed after the geometry specification, such as the location of a locally stored basis set. The default is none.

`High` (string): The (high) level of theory (method/basis set) that is used to obtain the actual results. The string will be directly copied into the Gaussian files, so check the Gaussian manual for further details. The default is B3LYP/6-31G.

`OptH` (string): The equivalent of `OptL` for the high level of theory. `Tight` is allowed, and the default is none.

`ExtH` (string): The equivalent of `ExtL` for the high level of theory. `Freq` is not allowed, as frequency calculations are automatically invoked at the end of all calculations.

`PostH` (string): The equivalent of `PostL` for the high level of theory.

`Ncpumax` (integer): The number of cores that are available for calculation on the machine. The default is 4.

`MemGB` (floating point): The total available shared memory on the system in GB units. The default is 4 GB. KinBot utilizes maximum 90% of the total memory available in the system to avoid crashes.

`MaxJob` (integer): The maximum number of parallel jobs that the user wants to allow. If this keyword is not provided or is larger than `Ncpumax`, the number of parallel jobs will be limited by the number of cores (`Ncpumax`) and/or by the value of `FixMem`, if provided.

`FixMem` (floating point): It can be desirable to specify the memory allocated to each Gaussian job. `FixMem` has to be followed by the amount of memory per job in GB units. The default is such that the total memory (`MemGB`) determines the memory per job.

`Charge` (integer): The charge of the seed particle. The default is zero.

`Rottol` (floating point): The smallest energy difference that is considered larger than the numerical accuracy of a Gaussian calculation when comparing the energies of different structures. The unit is  $\text{cm}^{-1}$ , and the default is  $35 \text{ cm}^{-1}$  ( $\sim 0.1 \text{ kcal mol}^{-1}$ ).

`MaxTime` (integer): The approximate maximum wall clock time in seconds that is allowed to finish one set of parallel job execution. When the time is passed, all currently running Gaussian jobs are killed, and KinBot moves to the next task, if possible. The default is 8640000 s (100 days).

`Restart`: KinBot determines if the prescribed jobs are finished, and if yes, the finished jobs are not resubmitted, but their results are used. Care must be taken: if for instance the basis set is changed from a previous calculation, the `Restart` option will not recognize that the finished jobs were done at a different level of theory, which will lead to false results and/or crash.

`NRot` (integer): The number of steps requested when a 1-D scan is done along the dihedral angles. The default is 12.

`MC / NoMC`: Request / do not request a MC-type conformational search. The default is `MC`.

`Scission / NoScission`: Request / do not request reaction path search for non-beta scissions. The default is `Scission`.

`Relax / NoRelax`: In the 1-D hindered rotor scans of transition states (if scans are requested) do a relaxed / non relaxed scan. The non-relaxed scan means that the TS region of the structure is kept fixed. The default is `Relax`.

`Fitorder` (integer): The order of the Fourier series that is fitted to the results of the successful 1-D scans (if requested). The default is 5.

Rot / NoRot: Request / do not request a 1-D dihedral scan along the dihedral angles. The default is Rot.

Multip (integer): User-defined multiplicity. If not provided, the multiplicity is determined internally. Note, however, that reaction search is currently only available for singlet and doublet structures.

The values of the input and the applied default values are repeated at the beginning of the main output file.

### 2.3. Running KinBot and reading the output

The executable of the code is called `kinbot.run`, and has to be in the same directory where the two input files are. To run the code:

```
./kinbot.run <keywordfile> &
```

The program will create a directory called `<seedfile>_dir`, and all outputs will be created in that directory.

The main output file is called `<seedfile>.out`. The output is verbose, to facilitate easy understanding of the results and what has been done by the code. It has the following structure:

1. Input parameters and the geometry of the seed.
2. Structural characterization of the seed (bond matrix, bonds, position of the radical center, whether it is a transition state or not, number of cycles, number of pieces etc.). If any of the determined characteristics is erroneous for some reason, or their values lie outside the current applicability of the code (e.g. more than one cycle is present), the code quits with an error message, displayed in this output.
3. Information on the reaction pathway searches, such as how many and what kind of pathways were searched for. Upon successful completion, the geometries, frequencies and rotational constants of the corresponding transition states and product channels are printed as well.
4. The number of valid dihedral angles in the original, optimized structure, and the results of the MC sampling. The structure of the lowest energy conformer is displayed, as well as the number of unsuccessful samplings.

### 3. FURTHER DEVELOPMENT PLANS

We have demonstrated that it is possible to create a practical and robust code that greatly accelerates the characterization of chemical pathways in gas-phase chemistry. The code that we created, however, can be extended further. In the following, the possible extensions are listed.

Developments to extend the range of applicability:

- Provide means to treat multiple cyclic substructures.
- Allow reaction search for triplet structures, when the two unpaired electrons are not on neighboring atoms.
- Automatically determine the location of the charge, and/or allow user input to define the location of the charge.
- Detect symmetry and equivalent functional groups beyond equivalent H atoms.

Reaction types that can be added to the already existing ones:

- Generic 1,*n* elimination pathways. This reaction type covers simple 1,2 elimination, e.g. water elimination from alcohols, but also more complex elimination channels that are expected to play a role in oxygenates.
- Internal radical addition to double bonds.
- Simple (bimolecular) abstraction reactions.
- Internal alkyl group abstraction (or migration).
- Ring-closure of triplet biradicals.

The code will be evolved to include an overall recursive loop that will allow the determination of reaction pathways beyond level 1 (see Figure 2). This will also include algorithms that can detect the identity of two chemical structures, even if their conformations are different, in order to avoid exploring the same region of the PES twice.

Future versions will include calculation of partition functions using the Monte Carlo sampling of the conformers as well as different strategies to find regions of the reaction path network that were not discovered by the chemistry-based method implemented currently in KinBot.



## 4. REFERENCES

- [1] A. Fernandez-Ramos, J.A. Miller, S.J. Klippenstein, D.G. Truhlar, *Chem. Rev.* 106 (11) (2006) 4518-4584.
- [2] C.F. Goldsmith, A.S. Tomlin, S.J. Klippenstein, *Proc. Combust. Inst.* accepted (2012)
- [3] J. Prager, N.H. Najm, J. Zádor, *Proc. Combust. Inst.* accepted (2012)
- [4] J. Zádor, A.W. Jasper, J.A. Miller, *Phys. Chem. Chem. Phys.* 11 (46) (2009) 11040-11053.
- [5] T. Wieland, A. Kerber, R. Laue, *J. Chem. Inf. Comp. Sci.* 36 (3) (1996) 413-419.
- [6] S. Maeda, K. Ohno, *Chem. Phys. Lett.* 381 (1-2) (2003) 177-186.
- [7] S. Maeda, K. Ohno, K. Morokuma, *J. Chem. Theory Comp.* 5 (10) (2009) 2734-2743.
- [8] C.J. Cerjan, W.H. Miller, *J. Chem. Phys.* 75 (6) (1981) 2800-2806.
- [9] N. Gonzalez-Garcia, J.Z. Pu, A. Gonzalez-Lafont, J.M. Lluch, D.G. Truhlar, *J. Chem. Theory Comp.* 2 (4) (2006) 895-904.
- [10] C.Y. Peng, H.B. Schlegel, *Isr. J. Chem.* 33 (4) (1993) 449-454.
- [11] I.V. Ionova, E.A. Carter, *J. Chem. Phys.* 98 (8) (1993) 6377-6386.
- [12] O. Welz, J. Zádor, J.D. Savee, M.Y. Ng, G. Meloni, R.X. Fernandes, L. Sheps, B.A. Simmons, T.S. Lee, D.L. Osborn, C.A. Taatjes, *Phys. Chem. Chem. Phys.* 14 (9) (2012) 3112-3127.
- [13] M.J. Frisch, G.W. Trucks, H.B. Schlegel, et al. *Gaussian 09, Revision A.02*; Gaussian, Inc.: Wallingford CT, 2009.
- [14] I. Kolossvary, W.C. Guida, *J. Am. Chem. Soc.* 118 (21) (1996) 5011-5019.
- [15] I. Kolossvary, W.C. Guida, *J. Comput. Chem.* 14 (6) (1993) 691-698.
- [16] N. Weinberg, S. Wolfe, *J. Am. Chem. Soc.* 116 (22) (1994) 9860-9868.



## DISTRIBUTION

[List external recipient names and addresses]

4 Lawrence Livermore National Laboratory  
Attn: N. Dunipace (1)  
P.O. Box 808, MS L-795  
Livermore, CA 94551-0808

[List in order of lower to higher Mail Stop numbers.]

1	MSXXXX	Name of Person	Org. Number
1	MSXXXX	Name of Person	Org. Number

[The housekeeping entries are required for all SAND reports.]

1	MS0899	Technical Library	9536 (electronic copy)
---	--------	-------------------	------------------------

For LDRD reports, add:

1	MS0359	D. Chavez, LDRD Office	1911
---	--------	------------------------	------

For CRADA reports add:

1	MS0115	OFA/NFE Agreements	10012
---	--------	--------------------	-------

For Patent Caution reports, add:

1	MS0161	Legal Technology Transfer Center	11500
---	--------	----------------------------------	-------

