Available online at www.sciencedirect.com

ScienceDirect

Proceedings of the Combustion Institute 38 (2021) 375–384

Proceedings of the Combustion Institute

www.elsevier.com/locate/proci

Automated theoretical chemical kinetics: Predicting the kinetics for the initial stages of pyrolysis

Sarah N. Elliott [a], Kevin B. Moore III [b], Andreas V. Copan [c], Murat Keçeli [d], Carlo Cavallotti [e], Yuri Georgievskii [b], Henry F. Schaefer III [a], Stephen J. Klippenstein [b,*]

[a] *Center for Computational Quantum Chemistry, University of Georgia, Athens, GA, 30602, United States*
[b] *Chemical Sciences and Engineering Division, Argonne National Laboratory, Lemont, IL, 60439, United States*
[c] *Emmanuel College, Natural Sciences Department, Franklin Springs, GA, 30639, United States*
[d] *Computational Science Division, Argonne National Laboratory, Lemont, IL, 60439, United States*
[e] *Department of Chemistry, Materials and Chemical Engineering "G. Natta", Politecnico di Milano, Milan, Italy*

Received 8 November 2019; accepted 28 June 2020
Available online 13 August 2020

Abstract

Large scale implementation of high level computational theoretical chemical kinetics offers the prospect for dramatically improving the fidelity of combustion chemical modeling. To facilitate such efforts, we have developed a suite of codes, collectively referred to as AutoMech, that allow for the automatic prediction of the kinetics for large sets of reactions via *ab initio* transition-state-theory based master-equation calculations. The primary input is simply the mechanism, a dictionary relating chemically identifiable species descriptors (e.g., SMILES or InChIs) to species labels in the mechanism, and a specification of the electronic structure and transition state theory models to be implemented. Here we illustrate the current utility of AutoMech through a study of the initial stages of pyrolysis for 3 sets of fuels: sequences of alkanes, alcohols, and aldehydes. For simplicity, the analysis focuses on abstractions from the fuel by H, CH_3, and OH, and the decomposition of the resulting radicals. Altogether, there are a total of 166 input channels in these sets (more than 363 forward reactions when expanded to the full set of elementary reactions). The code successfully produces high quality rate estimates (with apparent uncertainties less than a factor of two in limited comparisons with experiment) for >95% of these. For the radical decomposition reactions, the analysis includes predictions for the pressure dependence of the kinetics. This wide-ranging exploration illustrates (i) the effect of different levels of prediction on the expected accuracy, (ii) the branching between abstractions at different sites for different abstractors, (iii) the dependence of the rates on the chemical structure, and (iv) the variation in radical stabilities across chemical families. These results, as well as the demonstrated feasibility of the methodology, should find further utility in the development of accurate rate expressions for arbitrary fuels.

Keywords: Computational chemistry; Ab initio kinetics; Workflow; Pyrolysis

* Corresponding author.
E-mail address: sjk@anl.gov (S.J. Klippenstein).

https://doi.org/10.1016/j.proci.2020.06.019
1540-7489

1. Introduction

The compilation of a detailed chemical kinetic mechanism for fuel combustion is a challenging task due to the vast amount of data required to represent the chemical transformation from fuel to products [1]. The decomposition and ultimate conversion to combustion end products (e.g., CO_2 and H_2O) and pollutants (e.g., NOx, oxygenated hydrocarbons, and soot) occurs through a bewildering array of pathways. The complete description of these transformations commonly requires the representation of the temperature and pressure dependence of rate coefficients for 1000 to 10,000 reactions, as well as the thermochemical properties for 100 to 1000 species. Needless to say, it is challenging to obtain well validated values for all of these quantities.

Ab initio theoretical kinetics has proven to be a valuable supplement to the combustion modeling enterprise [2,3]. Careful studies can now yield reaction rate predictions with accuracies equivalent to those of many experiments. As such, it is now fairly commonplace for modeling studies to include *ab initio* kinetics predictions for a limited number of the most important and/or least studied reactions.

The ever-advancing capabilities of modern computational facilities provides a new opportunity for utilizing such *ab initio* kinetics. In particular, there is now enough CPU power to make meaningful rate predictions for a large fraction of the reactions in a given mechanism. Ideally, one would make rate predictions for every reaction in a mechanism, thereby reducing the effect of uncertainties in the more crudely estimated reaction rates. However, reaching this ideal is severely limited by the amount of human effort involved in making a single *a priori* rate prediction.

In recent work, we began to address this issue by developing a procedure (EStokTP) for automatically predicting the rate constant for a single reaction from simple specifications of the molecular structure, as well as the electronic structure and transition state models [4]. The initial release of the EStokTP code was limited to a modest set of reaction types: additions, abstractions, and migrations, which nevertheless allows for the treatment of a large fraction of the combustion relevant reactions. Other researchers have focused on the implementation of advanced protocols for mapping out potential energy surfaces [5-8], which ultimately governs the reaction kinetics, or automatically predicting pressure independent rate constants [9,10]. The present effort is driven by the desire to extend EStokTP to the automated (i) treatment of multiple-well multiple-product potential energy surfaces in order to predict physically meaningful reaction rates for arbitrarily complex pressure dependent reactions, (ii) consideration of large sets of such reactions, and (iii) comparison of predictions provided by alternative theoretical methods.

Such extensions to large numbers of complex reactions with optional recoverable theoretical methods require several aspects of chemical informatics. The ability to uniquely identify species and their structural characteristics using identifiers such as SMILES [11] or InChI [12] is required in order to reproducibly and automatically manipulate species. Meanwhile, the implementation of sophisticated data structures and filesystems is required in order to efficiently generate, save, and use chemical data (e.g., electronic structure, kinetic, and thermochemical) through the multiple phases of automated kinetics procedures. The use of chemical informatics is a central aspect of advanced combustion modeling programs such as RMG [13]. As a first step in automatically considering large sets of species, we developed a code (QTC) for predicting thermochemical properties for a large set of species [14]. In essence, this code was designed to couple chemical informatics-based representations and filesystems with the EStokTP codebase.

In this work, we describe our continuing efforts to completely automate the prediction of the thermochemical kinetics for a full combustion mechanism. This effort revolves around the development of a completely refactored codebase (AutoMech [15]) that takes as primary input an arbitrary chemical mechanism and provides as primary output an updated ChemKin formatted chemical mechanism. In its current state, AutoMech is essentially fully debugged for application of conventional transition state theory with one-dimensional hindered rotor corrections and arbitrary composite electronic structure models to abstractions, additions, and hydrogen migrations. Many other capabilities are in the process of being added.

The current set of capabilities for AutoMech allows for a ready examination of the initial stages of pyrolysis. In particular, the primary decomposition generally involves H-atom abstraction reactions, and the next step involves beta-scission decompositions, which may be preceded by hydrogen migrations. To test the code, and to demonstrate its utility we have applied it to the corresponding set of reactions for an alkane series (CH_4, C_2H_6, C_3H_8, and n-C_4H_{10}), an alcohol series (CH_3OH, C_2H_5OH, i-C_3H_7OH, n-C_3H_7OH, and n-C_4H_9OH), and an aldehyde/ketone series [H_2CO, CH_3CHO, and $(CH_3)_2CO$], with H, CH_3, and OH as the abstractors. The latter two series are motivated by ongoing collaborations with Wooldridge (Michigan), where the pyrolysis of i-C_3H_7OH is being examined in a rapid compression machine, as well as with Prozument (Argonne), where the pyrolysis of $(CH_3)_2CO$ is being examined with chirped pulse microwave spectroscopy in a microreactor. More generally, the data is expected to prove useful in exploring the chemical dependence of kinetics (total rate constants, branching fractions, and radical stabilities) and the method dependence of theoretical kinetics.

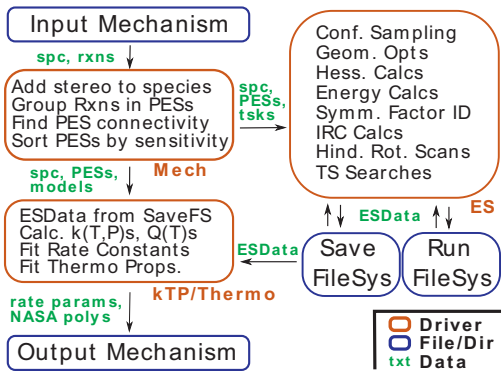

Fig. 1. Overview of the AutoMech workflow governed by MechDriver.

2. Automated workflow code

2.1. Overview

The "AutoMech" suite [15] utilized in the present study is designed to fully automate all of the first-principles electronic-structure and rate theory calculations necessary to predict the kinetics and thermochemistry of gas-phase reaction mechanisms. AutoMech builds from the concepts previously implemented in the EStokTP and QTC codes and refactors them into a more modular, functional format. Its full implementation in Python provides a robust range of I/O capabilities that facilitate the movement of data throughout the workflow. AutoMech is designed to take as input an arbitrary chemical mechanism in ChemKin or JSON format, along with a file relating the species labels in the mechanism to chemical identifiers (currently SMILES [11] or InChI [12] strings). Additional input involves a specification of the electronic structure methods and transition state theory models to be used.

The workflow in AutoMech is managed by the MechDriver code, which serves as a foundation stone for the AutoMech suite. MechDriver consists of high-level drivers and routines that execute series of individual tasks by calling libraries of simple, low-level functions. This approach facilitates adaptation to new situations and methodologies, while preventing the code from becoming overly monolithic. Currently, it is used to (1) calculate electronic structure, thermochemical, and kinetic data for arbitrary numbers of species on multichannel potential energy surfaces (PESs) and (2) implement novel methods to generate partition functions and rate constants using transition state theory.

A broad view of the automated workflow implemented in MechDriver is provided in Fig. 1. It functions by parsing the input from the user and passes this information to the drivers that handle electronic structure (ESDriver), kinetics (kTPDriver), and thermochemistry (ThermoDriver) calculations for all the requested reactions and species. Upon launch, each driver will cycle through the required tasks to generate the user-requested data.

Importantly, MechDriver has been built so that it can launch these drivers either separately or in sequence. Moreover, it is possible to run only a single task of one driver or every task of every driver. This approach is a part of broader goal that the use of MechDriver is clean and consistent for both very simple and very complex sets of calculations. Minimal changes to the input to MechDriver allow the user to go from a simple task such as optimizing the geometry of a single species to calculating the rate of a single reactions, to heretofore impossible tasks such as generating the thermochemistry and kinetics for 10 multichannel PESs at four levels of theory (as in the present work).

While functionally independent, ESDriver, kTPDriver, and ThermoDriver are tied together by two main factors: (1) their common input file structure and internal data representations described later, and (2) their interaction with the *run-save* filesystem framework, which is integral to the functioning of the workflow routine. The *run* filesystem contains the working directories where all the electronic structure, thermochemistry, and kinetics calculations are performed. The *save* filesystem contains all of the archival quality electronic structure data obtained from validated calculations. It also contains info files that summarize the input files used to generate the data. The combination of info files and save filesystem data storage allows for the safe deletion of the often very large electronic structure log files once the user is satisfied with the results. Meanwhile, the data stored in the *save* filesystem, can be used as a *de facto* local database.

The structure of the *run-save* filesystem, and the data contained within it, is enforced by our autofile library, whose functions also handle the creation and modification of the filesystem via calls from the workflow routine. The *run* and *save* filesystems are mirrors of one another, which facilitates user navigation and code development. Each filesystem involves a series of layers that are comprised of several identifiers to uniquely identify the type and location of a piece of information, including, but not limited to: an InChI key, charge, multiplicity, level of theory for the geometry, ID of a conformer of the geometry.

The interaction with the *run-save* filesystem is principally simple for each driver. Take as an example a request to ESDriver to obtain a B2PLYP-D3/cc-pVTZ optimized geometry. ESDriver will first check in the *save* filesystem for this geometry. If the geometry exists, it will read it from the storage file and move to the next task. If not, it will launch the geometry optimization in the *run* filesystem. If the job is successful, the code will parse the

output from the *run* filesystem and save the geometry in the appropriate storage file in the *save* filesystem. This geometry is then available for ESDriver to run B2PLYP-D3/cc-pVTZ hindered rotor scans or for kTPDriver to build a master equation input file. This storage and reuse of previously calculated data dramatically reduces CPU usage, and greatly facilitates method comparisons.

Lastly, a variety of tools in AutoMech help analyze and produce information from the ChemKin mechanism files output by MechDriver. These tools include (1) calculations of the branching fractions for all reactions in a mechanism that share common reactants, (2) summaries of the radical stability temperature range, and (3) comparison plots of pressure-dependent rate constants and thermochemical parameters from two mechanism files. We expect to extend the breadth and capability of such tools in coming months.

2.2. Description of mechdriver workflow routines

The following sections provide a more detailed, but still brief, overview of the input and each of the drivers in the workflow. The interested reader may refer to the more detailed descriptions provided in the closely related EStokTP effort [4].

2.2.1. Central input to mechdriver

AutoMech currently takes as input a list of reactions in a ChemKin- or JSON-formatted file accompanied by a CSV file that relates species names in the reaction file to a unique structural identifier, such as InChI or SMILES strings. Internally, each of the species are indexed by InChI strings because this identifier allows for greater specificity, particularly with regard to the consideration of stereochemistry. Various further species (or transition state) specific parameters, (e.g., the number of conformational samples desired, symmetry factor, angle increments for hindered rotor scans, specific hindered rotor to be scanned, etc.) can be provided in auxiliary files.

The theoretical treatment to be employed is specified through two additional files. One file defines "models" comprised of (1) methods and approximations used for constructing partition functions, and (2) levels of electronic structure theory used to generate the data required to construct the partition function. We currently support many different model treatments, with many fully debugged while others are still under the final stages of development. For rotational and vibrational partition functions: rigid-rotor harmonic oscillator, one-dimensional and multi-dimensional grid based hindered rotor treatments, and a new path integral based multidimensional rotor analysis. The vibrational modes may also be treated with second-order vibrational perturbation theory. For transition state partition functions, we provide conventional, variational, phase space theory, and variable-reaction coordinate transition state theory models. For tunneling corrections, Eckart or small curvature tunneling models are available.

The electronic structure models are simplified through the definition of various electronic structure "levels" in the second file, which are then called by the level label in the first file. Within each level, all of the necessary information is provided to launch a calculation for a given electronic structure method including method, basis, convergence options, as well as job parameters such as program, number of processes, memory, etc. In addition, there are various option keywords that allow users to input job parameters that are highly specific to a given electronic structure package. The user may also employ arbitrary composite methods through weighted sums of multiple electronic structure methods.

Notably, for all of these files, there is no limit to the number of things a user may define. The mechanism and species files can include any number of reactions and species, respectively. In addition, the user can define any number of "models" and "levels" within the partition function models file and electronic structure theory file, respectively. MechDriver will only pull from these files what the user provides in a final run input file. Herein, the user simply lists the PES or species labels that pertain to their chemistry of interest alongside the names they have given to their defined "models" and "levels" to define their desired level of treatment. This approach provides an elegant way of providing enormous control to the user while maintaining a consistent approach to calculations of any complexity.

2.2.2. MechDriver

As described previously, MechDriver is the principal/central workflow driver in AutoMech. It parses the input and launches its children drivers, ESDriver, kTPDriver, and ThermoDriver, in order to execute the user's requested tasks. One of the main initial functions of MechDriver is to preprocess the input mechanism in order to produce representations of the species and potential energy surfaces suitable for theoretical calculations. These preprocessing utilities of MechDriver are key to the extension to treating complex multi-channel multi-well reactions.

In this preprocessing, MechDriver first determines all the unique potential energy surfaces (PESs) by examining the overall stoichiometry of each reaction. Subsequently, MechDriver analyzes the connectivity of the reactions within each PES, i.e., it seeks to identify sets of reagents on the PES that are connected by a contiguous series of reaction channels. This process yields separate sub-PESs that share an overall stoichiometry, but that are not connected by the chemistry of the mechanism. Each sub-PES of completely connected reaction channels then serves as a proper distinct

model for determining the phenomenological rate coefficients. When kTPDriver is called, it loops over each sub-PES to generate the input for master equation calculations, which are all performed with the MESS code [16], as are all the partition function evaluations. The advanced solver capabilities in MESS directly provide the full set of phenomenological rate coefficients for arbitrary PESs.

For the reactants and products of each reaction on the requested PESs, AutoMech is able to convert from the structural identifiers to molecular graphs, the relationships of which are exploited by AutoMech to automatically identify the reaction type. Here, we obtained results for abstractions, additions, and isomerizations, whose searching algorithms closely match those in the EStokTP code. More recently, we added algorithms for substitution, insertion, and unimolecular elimination reactions. We have also introduced sophisticated TST algorithms for radical-radical abstractions and additions, which require VTST or VRC-TST treatments based on multireference electronic structure calculations; such reaction types are often crudely treated in most theoretical studies, even without consideration of automated approaches. In keeping with our overall design philosophy, we have made it easy to add in new reaction classes and searching algorithms.

2.2.3. ESDriver

ESDriver serves as the workhorse of the automated workflow, responsible for handling all the electronic structure theory calculations, which are the most time- and CPU-intensive processes. The driver cycles through a set of tasks for specified species and reactions. The type, number, and order of these tasks can be either directly provided by the user or set automatically by MechDriver based on what desired kinetics or thermochemistry results the user requests. For each task, ESDriver makes calls to a generalized interface to all electronic structure programs, which handles writing input files and parsing output files for the specified calculation and program. In conjunction with this, the driver makes calls to autofile to read and write the resultant electronic structure data into the filesystems for use in either subsequent electronic structure tasks or kinetics and thermo calculations.

The generalized electronic structure interface can readily be extended to new electronic structure packages or new algorithms in supported codes. We currently have some level of support for the following programs: CFour, Gaussian, Molpro, MRCC, NWChem, Orca, and Psi4. Broadly, for each new electronic structure package, we write procedures for writing and reading the files for the basic electronic structure tasks, including single-point energy calculations, gradient calculations, geometry optimizations, and Hessian calculations. We are also capable of performing intrinsic reaction coordinate calculations and second-order vibrational perturbation theory for codes that support these options. Built from these basic tasks are more involved routines, such as scans along hindered rotors and reaction modes, and Monte Carlo sampling over torsional modes. The ability of AutoMech to employ any electronic structure package and method allows a multitude of users the ability to use the code, since many users preferred electronic structure packages are neither free nor open-source.

Another important novel aspect of the AutoMech workflow involves robust procedures for assessing and resolving failures common to electronic structure calculations, such as failed SCF convergence or geometry convergence. Upon detecting such failures, the code can automatically proceed through a sequence of subsequent calculations with modified input until a successful result is obtained. For instance, given a geometry optimization failure, we first launch a second job with more iterations, then, if needed, launch jobs with Hessians computed at various steps to help the optimizer. In the case of SCF failures, we can modify the guess for the wavefunction or electron density by adding level shifts or damping values, or build an initial guess using a different method for the desired SCF method (e.g. use a UHF or CASSCF wavefunction for the more troublesome ROHF wavefunction). We have also built in procedures to check for failures in sampling or scanning tasks.

2.2.4. kTP and thermo driver

From the partition function and electronic structure models provided by the user, these drivers proceed to search the *save* filesystem for all of the required electronic structure information. This information is then collated into a MESS [16] input file, which is then run to obtain the rate constants and/or partition functions, as appropriate. These drivers then take output from MESS and generate the data needed to build a new mechanism in ChemKin format.

In the case of kTPDriver, the rate constants are fit to a functional form of the user's choice. Currently, we have support for fitting to the following expressions: modified Arrhenius, PLog, Lindemann, and Troe. The rate constants are probed to see whether their pressure variation is enough to warrant a fit to a pressure-dependent functional form. We also examine the adequacy of single versus double modified Arrhenius representations, with the thresholds for such checks set by the user. Meanwhile, for ThermoDriver, partition functions are used to calculate thermochemical properties which are ultimately fit to 7-coefficient NASA polynomials.

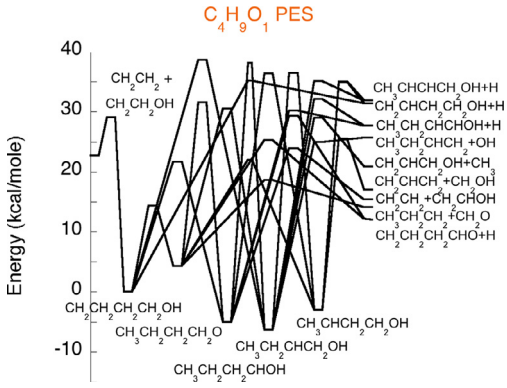

Fig. 2. Schematic plot of the $C_4H_9O_1$ PES with L4.

3. Theoretical methods

A 166-reaction channel starting mechanism was generated by considering the H-abstractions by H, CH_3, and OH from methane, ethane, propane, n-butane, formaldehyde, acetaldehyde, acetone, methanol, ethanol, i- and n-propanol, and n-butanol, and the decomposition of the corresponding fuel radicals. For acetone, we also considered the reactions arising from the addition of H atoms. The complete mechanism so generated is provided in the supplemental material.

For every PES of interest, multichannel, pressure-dependent rate constants [$k(T,P)$] were calculated via the Master Equation (ME) using the MESS program [16]. Fig. 2 demonstrates the complexity of the PES obtained for the fuel radicals arising from n-butanol. Corresponding PESs are provided for the other radicals in the Supplementary material. Notably, on this PES alone, we generate 237 rate constants for each temperature and pressure.

The microcanonical rate constants [$k(E)$] required for the ME treatment were obtained using conventional transition state theory. For all species, nuclear degrees-of-freedom are treated within the rigid-rotor, harmonic oscillator approximation, with the exception of internal rotations, which are automatically identified from the valence structure, and are modeled as one-dimensional hindered rotors. These rotors are projected out of the Hessian using the ProjRot code [4] and the projected harmonic vibrational frequencies are used in the master equation simulation. Electronic partition functions are obtained from a user specified dictionary or simply taken as the multiplicity. Tunneling effects are treated using a simple Eckart model. For simplicity, generic Lennard-Jones parameters are used to represent the collision rates and energy transfer probabilities are calculated using the exponential down model.

Four separate sets of mechanism data were generated via the master equation using four different levels of electronic structure theory. This series allow us to examine the variation in the predicted rate constant with electronic structure methods of varying costs. In terms of increasing expected quality of the data, these levels are:

(L1) CCSD(T)-F12/cc-pVDZ-F12//ωB97X-D/6–31G*
(L2) CCSD(T)-F12/cc-pVTZ-F12//ωB97X-D/cc-pVTZ
(L3) CCSD(T)-F12/cc-pVQZ-F12//B2PLYP-D3/cc-pVTZ
(L4) CCSD(T)-F12/CBS(QZ,TZ)//B2PLYP-D3/cc-pVTZ

For each method listed, the given density functional was used to obtain all of the optimized geometries, harmonic vibrational frequencies, and hindered rotor potentials. The given coupled cluster method was used for all the single-point energy computations. For all of the approaches listed above, initial conformational sampling and symmetry factor determination was done at the ωB97X-D/6–31G* level of theory. The ωB97X-D and B2PLYP-D3 calculations were all performed using Gaussian09 [17], whereas CCSD(T)-F12 calculations were done using Molpro2015 [18]. Note that for certain cases where B2PLYP-D3 torsional potentials couldn't be converged, we employed ωB97X-D/cc-pVTZ torsional potentials in the rate calculations. Results from this approach are labeled L4'.

4. Results and discussion

The full set of rate constants evaluated at the L4 level are reported in the Supplementary material through provision of a ChemKin style mechanism file, while the values from the L1, L2, and L3 methods are comprehensively compared with the L4 values for the full set of reactions. These comparisons indicate that the basis set extrapolation typically has at most a few percent effect on the rate predictions. Meanwhile, near 1000 K, the difference between the L2 and L4 and L1 and L4 methods, are generally less than about 30% and a factor of 2, respectively. Although not explored here, the full set of data can provide useful consistent correlations of value in refining groups and other chemistry-based rate rules.

Overall, we had > 95% success in locating the appropriate transition state with the current algorithms and a similar level of success in predicting meaningful rate coefficients from those transition states. Space limitations preclude detailed discussions of the pratfalls in the process. One simple observation though is that the greatest difficulty is maintaining chemically correct structures at the TS for torsional mappings and samplings. Employing

Table 1
Radical stability temperatures (K).[a]

Radical	Pressure (atm)			
	0.1	1	10	100
HCO	1650	1700	1750	1800
CH_2OH	>2000	>2000	>2000	>2000
CH_3O	1400	1450	1550	1650
CHCHOH	1950	>2000	>2000	>2000
CH_2CHO	1850	>2000	>2000	>2000
CH_2COH	1500	1650	1800	>2000
CH_3CO	950	1000	1050	1150
CH_3CHOH	1500	1600	1800	>2000
CH_2CH_2OH	1200	1300	1500	1700
CH_3CH_2O	850	900	1000	1100
CH_3CH_2	1700	1800	1950	>2000
$CH_3C(O)CH_2$	1550	1700	1900	>2000
$CH_2C(OH)CH_2$	1950	>2000	>2000	>2000
CH_3CH_2CHOH	1250	1350	1500	1750
CH_3CHCH_2OH	1100	1250	1400	1650
$CH_2CH_2CH_2OH$	1200	1350	1550	1850
$CH_3CH_2CH_2O$	650	700	800	900
$CH_3C(OH)CH_3$	1300	1450	1600	1850
$CH_3CH(OH)CH_2$	1100	1250	1400	1650
$CH_3CH(O)CH_3$	650	700	750	900
CH_3CHCH_3	1450	1550	1700	1950
$CH_3CH_2CH_2$	1250	1350	1500	1750
$CH_3CH_2CHCH_2OH$	1000	1100	1250	1450
$CH_3CH_2CH_2CHOH$	1100	1250	1450	1700
$CH_3CHCH_2CH_2OH$	1100	1250	1450	1750
$CH_2CH_2CH_2CH_2OH$	950	1200	1450	1800
$CH_3CH_2CHCH_3$	1200	1300	1500	1750
$CH_3CH_2CH_2CH_2$	1100	1250	1400	1650
$CH_3CH_2CH_2CH_2O$	N/A	650	750	900

[a] Determined on a 50 K temperature grid with L4.

smaller torsional grid steps (e.g. 10°) helps ameliorate the latter, while robust structure checking was implemented for the sampling. Cubic spline based interpolations are implemented to provide acceptable remedies for problematic torsional angles in hindered rotor scans. Such difficulties were exacerbated for the abstractions in alcohols by the OH group, where the barrier can be quite deeply submerged. For the pressure dependent reactions, some difficulties were encountered for the OH addition to molecules with pi-bonds, which may or may not have a short-range saddle point depending on the theoretical method [3]. The VTST methods under development will improve our treatment of those channels.

One novel feature of the MESS master equation code is its automatic recognition of when the timescale for chemical transformation is shorter than that for internal energy relaxation. When this occurs, the species is no longer stable, and should be considered as effectively merged with the species obtained from the chemical transformation. Notably, this merging is directly related to the observation from the 1980s, that a lumping of reactions together yields an effective treatment of the decay process at high temperatures [19–21]. Wang and coworkers have recently followed up on this concept with their HyChem model [22].

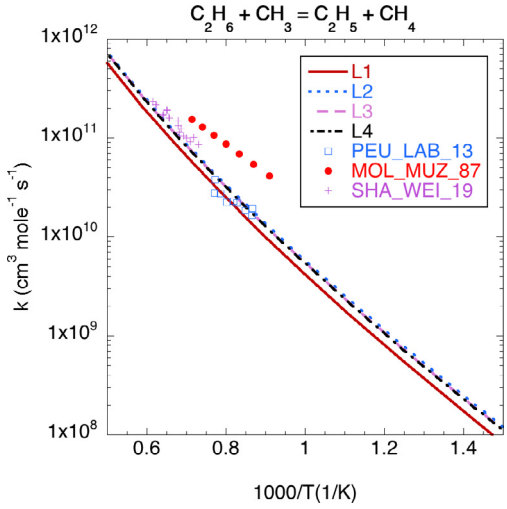

Fig. 3. Comparison of L1, L2, L3, and L4 predictions with experiment [23–25] for the $C_2H_6 + CH_3$ abstraction reaction.

Table 2
Branching fractions for molecule-radical abstraction reactions at 1 atm.[a]

Molecule	Product	Temperature (K)		
		500	1000	2000
H Abstractor				
C_2H_5OH	CH_2CH_2OH	0.01	0.14	0.33
C_2H_5OH	CH_3CH_2O	0.00	0.02	0.06
C_2H_5OH	CH_3CHOH	0.98	0.85	0.61
C_3H_8	$CH_2CH_2CH_3$	0.15	0.40	0.57
C_3H_8	CH_3CHCH_3	0.85	0.60	0.43
C_4H_{10}	$CH_2CH_2CH_2CH_3$	0.09	0.27	0.43
C_4H_{10}	$CH_3CHCH_2CH_3$	0.91	0.73	0.57
$CH_2C(OH)CH_3$	$CH_2C(O)CH_3$	0.05	0.11	0.14
$CH_2C(OH)CH_3$	$CH_2C(OH)CH_2$	0.95	0.87	0.71
$CH_2C(OH)CH_3$	$CHC(OH)CH_3$	0.00	0.02	0.16
CH_3CHO	CH_2CHO	0.01	0.08	0.28
CH_3CHO	CH_3CO	0.99	0.92	0.72
CH_3OH	CH_2OH	1.00	0.99	0.96
CH_3OH	CH_3O	0.00	0.01	0.04
iC_3H_7OH	$CH_3C(OH)CH_3$	0.99	0.80	0.44
iC_3H_7OH	$CH_3CH(O)CH_3$	0.00	0.01	0.05
iC_3H_7OH	$CH_3CH(OH)CH_2$	0.01	0.19	0.50
nC_3H_7OH	$CH_2CH_2CH_2OH$	0.02	0.14	0.27
nC_3H_7OH	$CH_3CH_2CH_2O$	0.00	0.01	0.04
nC_3H_7OH	CH_3CH_2CHOH	0.88	0.62	0.41
nC_3H_7OH	CH_3CHCH_2OH	0.09	0.23	0.28
CH_3 Abstractor				
C_2H_5OH	CH_2CH_2OH	0.05	0.17	0.31
C_2H_5OH	CH_3CH_2O	0.11	0.13	0.13
C_2H_5OH	CH_3CHOH	0.84	0.70	0.57
C_3H_8	$CH_2CH_2CH_3$	0.24	0.44	0.57
C_3H_8	CH_3CHCH_3	0.76	0.56	0.43
C_4H_{10}	$CH_2CH_2CH_2CH_3$	0.14	0.32	0.45
C_4H_{10}	$CH_3CHCH_2CH_3$	0.86	0.68	0.55
CH_3CHO	CH_2CHO	0.03	0.11	0.24
CH_3CHO	CH_3CO	0.97	0.89	0.76
CH_3OH	CH_2OH	0.65	0.74	0.79
CH_3OH	CH_3O	0.35	0.26	0.21
OH Abstractor				
C_2H_5OH	CH_2CH_2OH	0.31	0.39	0.54
C_2H_5OH	CH_3CH_2O	0.69	0.61	0.46
C_3H_8	$CH_2CH_2CH_3$	0.29	0.45	0.55
C_3H_8	CH_3CHCH_3	0.71	0.55	0.45
$CH_2C(OH)CH_3$	$CH_2C(O)CH_3$	1.00	0.97	0.72
$CH_2C(OH)CH_3$	$CHC(OH)CH_3$	0.00	0.03	0.28
CH_3CHO	CH_2CHO	0.06	0.19	0.38
CH_3CHO	CH_3CO	0.94	0.81	0.62
iC_3H_7OH	*$CH_3C(OH)CH_3$*	*0.87*	*0.64*	*0.41*
iC_3H_7OH	$CH_3CH(OH)CH_2$	0.13	0.36	0.59

[a] Evaluated at L4 (or at L4′ when a row is in italics).

In Table 1, we list this merging temperature for each of the radicals studied here. One can observe strong correlations in this temperature with chemical structure. Interestingly, the merging temperature is fairly strongly pressure dependent, suggesting that one should use caution in validating the appropriateness of a lumping scheme at low pressures and then employing it at the higher pressures of relevance to real devices.

It is often difficult to experimentally measure the branching between different possible product channels. The L4 predicted branching fractions for the abstraction reactions are reported in Table 2.

Performing detailed comparisons with experiment for as many reactions as were studied here was considered beyond the scope of this work. Nevertheless, extensive comparisons with experiment

are provided in the Supplementary material. These comparisons suggest that the present L4 predictions are generally within a factor of about 1.3 of the experimental predictions near 1000 K. For illustrative purposes we provide one sample comparison for the $C_2H_6 + CH_3$ reaction in Fig. 3, where the theoretical predictions are seen to split the experimental observations from the Michael [23] and Hanson [24] groups.

5. Conclusion

The automatic prediction of rate constants has been demonstrated for a set of reactions of relevance to the initial stages of pyrolysis in series of alkane, alcohol, and aldehyde fuels. The results allow for useful exploration of correlations in radical stability, chemical branching, total rate constants, and pressure dependence rate constants. ChemKin style mechanism files are used to report the full set of highest quality predictions obtained at level L4.

Declaration of Competing Interest

None.

Acknowledgements

This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two U.S. Department of Energy (DOE) organizations, the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem including software, applications, hardware, advanced system engineering, and early test bed platforms to support the nation's exascale computing imperative. This research was conducted using the Blues and Bebop computing resources, two high-performance computing clusters operated by the Laboratory Computing Resource Center at Argonne National Laboratory. This material is based on work supported by the DOE, Office of Science, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences at Argonne under Contract No. DE-AC02–06CH11357. Several authors of this research also wish to acknowledge additional support. Kevin Moore acknowledges support from the U. S. Army Research Office under award number W911NF1710531. Lastly, Sarah Elliott gratefully acknowledges support from a DOE Computational Sciences Graduate Fellowship through grant number DE-FG02–97ER25308.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.proci.2020.06.019.

References

[1] H.J. Curran, *Proc. Combust. Inst.* 37 (2019) 4857–4864.
[2] S.J. Klippenstein, *Proc. Combust. Inst.* 36 (2017) 77–111.
[3] S.J. Klippenstein, C. Cavallotti, in: T. Faravelli, F Manenti, E. Ranzi (Eds.), Computer-Aided Chem. Eng., 45, Elsevier, Cambridge, 2019.
[4] C. Cavallotti, M. Pelucchi, Y. Georgievskii, S.J. Klippenstein, *J. Chem. Theory Comp.* 15 (2019) 1122–1145.
[5] C.A. Grambow, A. Jamal, Y.P. Li, W.H. Green, J. Zador, Y.V. Suleimanov, *J. Am. Chem. Soc.* 140 (2018) 1035–1048.
[6] G.N. Simm, A.C. Vaucher, M. Reiher, *J. Phys. Chem. A* 123 (2019) 385–399.
[7] S. Maeda, Y. Harabuchi, *J. Chem. Theory Chem. Comput.* 15 (2019) 2111–2115.
[8] R. Van de Vivjer, J. Zador, KinBot, *Comp. Phys. Comm.* 248 (2020) 106947.
[9] R. Van de Vivjer, K.M. Van Geem, G.B. Marin, *Proc. Combust. Inst* 37 (2019) 283–290.
[10] P.L. Bhoorasingh, B.L. Slakman, F.S. Khanshan, J.Y. Cain, R.H. West, *J. Phys. Chem. A* 121 (2017) 6896–6904.
[11] The Open Babel Package, version 2.3.1, http://openbabel.org N.M. O'Boyle, M. Banck, C.A. James, C. Morley, T. Vandermeersch, G.R. HutchisonSimplified Molecular Line Entry System, *J. Cheminf.* 3 (2011) 33. https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system.
[12] International Chemical Identifier; https://en.wikipedia.org/wiki/International_Chemical_Identifier; https://web.archive.org/web/20120527162256/http://www.iupac.org/home/projects/project-db/project-details.html?tx_wfqbe_pi1%5Bproject_nr%5D=2000-025-1-800.
[13] C.W. Gao, J.W. Allen, W.H. Green, R.H. West, *Comp. Phys. Comm* 203 (2016) 212–225.
[14] M. Keceli, S.N. Elliott, Y.P. Li, M.S. Johnson, C. Cavallotti, Y. Georgievskii, W.H. Green, M. Pelucchi, J.M. Wozniak, A.W. Jasper, S.J. Klippenstein, *Proc. Combust. Inst.* 37 (2019) 363–371.
[15] K.B. Moore III, A.V. Copan, S.N. Elliott, Y. Georgievskii, C. Cavollotti, M. Keceli, and S.J. Klippenstein. AutoMech. https://github.com/Auto-Mech, 2020.
[16] Y. Georgievskii, S.J. Klippenstein, *J. Phys. Chem. A* 117 (2013) 12146–12154. http://tcg.cse.anl.gov/papr/codes/mess.html.
[17] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G.A. Petersson, et al., *Gaussian 09, Revision D.1*, Gaussian, Inc., Wallingford, CT, 2009.
[18] MOLPRO, version 2015. H.-.J. Werner, P.J. Knowles, G. Knizia, F.R. Manby, M. Schutz, and others, see http://www.molpro.net.

[19] E. Ranzi, M. Dente, S. Pierucci, G. Biardi, *Ind. Eng. Chem. Fundam.* 22 (1983) 132–139.
[20] J. Warnatz, *Proc. Combust. Inst.* 20 (1984) 845–856.
[21] T.J. Held, A.J. Marchese, F.L. Dryer, *Combust. Sci. Technol.* 123 (1997) 107–146.
[22] R. Xu, K. Wang, S. Banerjee, et al., *Combust. Flame* 193 (2018) 520–537.
[23] S.L. Peukert, N.J. Labbe, R. Sivaramakrishnan, J.V. Michael, *J. Phys. Chem. A* 117 (2013) 10228–10238.
[24] J. Shao, W. Wei, R. Choudhary, D.F. Davidson, R.K. Hanson, *J. Phys. Chem. A* 123 (2019) 9096–9101.
[25] W. Moller, E. Mozzhukhin, H. Gg. Wagner, *Ber. Bunsenges. Phys. Chem.* 91 (1987) 660–666.