

MANUAL

TSSMM: A Program for the Two-State Spin-Mixing Model

version 2017

Bo Yang, Laura Gagliardi, and Donald G. Truhlar

*Department of Chemistry, Chemical Theory Center, and Minnesota Supercomputing Institute
University of Minnesota, Minneapolis, Minnesota 55455*

Program: version date: July 28, 2017

Most recent update of manual: July 6, 2021

Abstract:

TSSMM is a Perl script that interfaces with the *Gaussian 09* program or the *Gaussian 16* program for the purpose of locating the stationary points on the lowest spin-orbit coupled (mixed-spin) potential energy surface (PES) using the electronic structure methods that are available in the *Gaussian* program. The program uses the two-state spin-mixing model (TSSMM). The interface is based on the *Gaussian* `External` keywords. The spin-orbit coupling between two different spin states of the same geometry is incorporated into the geometry optimization by using a pre-defined coupling constant. The coupling constant can be chosen semiempirically to reproduce the multiplet splitting in the atom or monatomic ion, or it can be determined by any other means, such as by a time-dependent density-functional theory (TDDFT) or a CASSI calculation at a representative geometry along the reaction coordinate where the low- and high-spin surfaces are close in energy.

Table of Contents

1.	USER AGREEMENT and required reference.....	3
2.	INTRODUCTION	4
3.	Recommended citation.....	6
4.	DISTRIBUTION.....	7
5.	PROGRAM DESCRIPTION.....	8
5.1.	File usage	8
5.2.	Perform a <i>TSSMM</i> calculation	9
5.2.1.	<i>Gaussian</i> input	9
5.2.2.	<code>TSSMM_main.pl</code>	9
5.2.3.	External basis sets	11
5.2.4.	<code>\$PATH</code> variable.....	11
5.3.	Notes on keywords.....	11
6.	EXAMPLE OF CALCULATING THE SPIN-ORBIT COUPLING CONSTANT USING THE <i>ADF</i> PROGRAM	13
7.	EXAMPLE OF A <i>TSSMM</i> CALCULATION	15
7.1.	Tests performed	15
7.2.	Test suite	15
7.2.1.	Example <code>feco4</code>	15
7.2.2.	Example <code>ts_feco4_co</code>	16
7.2.3.	Example <code>IRC_feco4_co</code>	16
8.	ACKNOWLEDGMENTS	17

1. USER AGREEMENT AND REQUIRED REFERENCE

The use of the *TSSMM* script implies acceptance of the terms of the license, which are repeated here for convenience:

The software is licensed under the Apache License, Version 2.0.
The manual is licensed under CC-BY-4.0.

No guarantee is made that this script is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within.

No consulting or maintenance services are guaranteed or implied.

The *Gaussian 09* and *Gaussian 16* code, which are required to use *TSSMM*, are covered by separate licenses. Using *TSSMM* presupposes that the user has *Gaussian 09* or *Gaussian 16* working properly before attempting to run *TSSMM*. No part of *Gaussian* is contained in *TSSMM* or needs to be modified for running this program. This *TSSMM* manual does not cover materials that are in the *Gaussian* manual.

Publications based on results obtained with the *TSSMM* script should include the following reference:

B. Yang, L. Gagliardi, and D. G. Truhlar, *TSSMM* – version 2017, University of Minnesota, Minneapolis, MN, 2017. <https://comp.chem.umn.edu/tssmm>

2. INTRODUCTION

TSSMM is a Perl script that interfaces with either the *Gaussian 09* program or the *Gaussian 16* program to locate the stationary points on the lowest mixed-spin potential energy surface (PES) using the electronic structure methods available in the *Gaussian* program to calculate the potential energy surfaces. The interface is based on the *Gaussian External* keyword. The spin-orbit coupling between two different spin states is incorporated into the geometry optimization by using a pre-defined coupling constant. The coupling constant can be chosen semiempirically to reproduce the multiplet splitting in the atom or monatomic ion, but one can also determine the constant by a time-dependent density-functional theory (TDDFT) or CASSI calculation at a representative geometry along the reaction coordinate where the low- and high-spin surfaces are close in energy.

The optimization of the local minima and transition state geometries can be done by using any *Gaussian* optimization keywords provided that the chosen keyword is compatible with the *External* keyword so that *Gaussian* can correctly call the external Perl script to acquire the needed energies and corresponding derivatives. The geometry optimization is performed by iterating of three general steps until converging criteria are met:

(a) For a given geometry (which will include the initial and optimized geometries for all reactants, transition structures, and intermediates that the user wishes to study and partially optimized geometries during the optimization processes), the user performs two single-point density functional calculations (one for each spin state) for the given geometry (which will include the initial and optimized geometries as well as all intermediates that the user wishes to study) at both low- and high-spin states using the *Gaussian* programs. The resulting energies and gradients of both spin states are provided to a Perl script at the end of the *Gaussian* calculation. As in all density functional calculations, the choice of functional and basis set will determine the accuracy of the result.

(b) The *TSSMM* script was written to read the quantities calculated in step (a) and to update those quantities following equations (1-3):

$$E_{mixed}^0 = \min \left\{ \text{eigenvalue} \begin{pmatrix} E_1 & \chi \\ \chi & E_2 \end{pmatrix} \right\} \quad (1)$$

$$G_i = \frac{\partial E_{mixed}^0}{\partial q_i} = \frac{(1-A)}{2} \frac{\partial E_1}{\partial q_i} + \frac{(1+A)}{2} \frac{\partial E_2}{\partial q_i} \quad (2)$$

$$A = \frac{(E_1 - E_2)}{\sqrt{4\chi^2 + (E_1 - E_2)^2}}$$

$$H_{i,j} = \frac{\partial^2 E_{mixed}^0}{\partial q_i \partial q_j} = B_{i,j} + \frac{(1-A)}{2} \frac{\partial^2 E_1}{\partial q_i \partial q_j} + \frac{(1+A)}{2} \frac{\partial^2 E_2}{\partial q_i \partial q_j} \quad (3)$$

$$B_{i,j} = \frac{(A^2 - 1) \left(\frac{\partial E_1}{\partial q_i} - \frac{\partial E_2}{\partial q_i} \right) \left(\frac{\partial E_1}{\partial q_j} - \frac{\partial E_2}{\partial q_j} \right)}{2\sqrt{4\chi^2 + (E_1 - E_2)^2}}$$

Equation (1) defines the energy of the species that lies on the lowest mixed-spin PES (E_{mixed}^0). Equations (2) and (3) are the gradients and Hessian matrix elements on the lowest spin-coupled PES;

(c) One step of geometry optimization using the chosen *Gaussian* optimizer based on the energies and derivatives provided by *TSSMM* in step (b). The updated geometry is then provided to *TSSMM*, which in turn calls another set of two single-point density functional calculations for both low- and high-spin states, i.e., one repeats steps (a), (b), and (c) until convergence of the optimized geometry is obtained.

3. RECOMMENDED CITATION

Citation for the program:

B. Yang, L. Gagliardi, and D. G. Truhlar, *TSSMM* – version 2017, University of Minnesota, Minneapolis, MN, 2017. <https://comp.chem.umn.edu/tssmm>

Citation for the method:

Transition States of Spin-Forbidden Reactions,” B. Yang, L. Gagliardi, and D. G. Truhlar, *Physical Chemistry Chemical Physics* **20**, 4129-4136 (2018). doi.org/10.1039/C7CP07227A

If *Gaussian* is used, the user should also cite the version that was used:

Gaussian 09, by 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, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. J. Montgomery, 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, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski and D. J. Fox, *Journal*, 2010, Revision E.01.

Gaussian 16, by 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, Williams, 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 and D. J. Fox, *Journal*, 2016, Revision A.03.

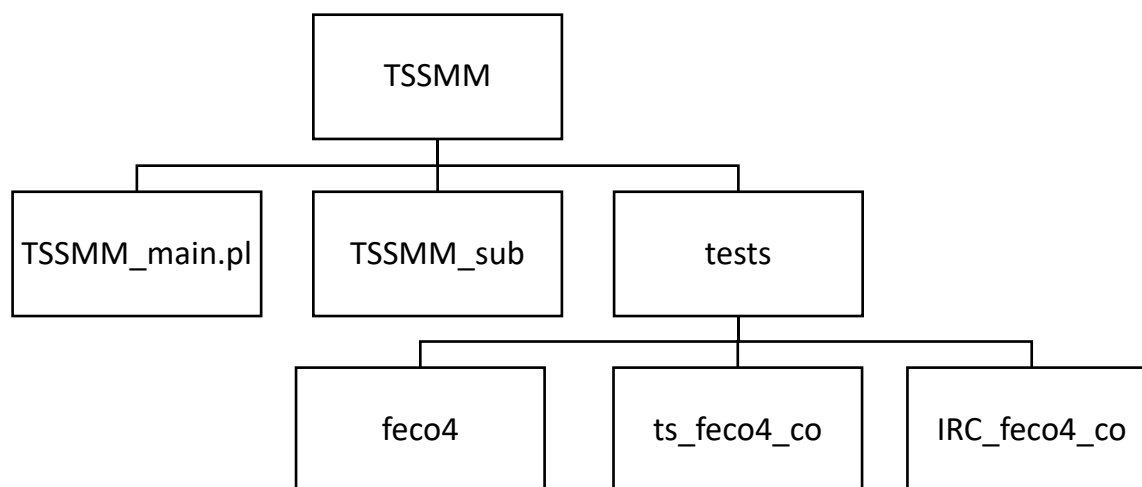
4. DISTRIBUTION

TSSMM is a Perl script that interfaces with a *Gaussian* program. The prospective user of *TSSMM* needs to obtain a licensed copy of *Gaussian 09* or *Gaussian 16* before proceeding with *TSSMM*. *Gaussian 09* and *Gaussian 16* are licensed codes available from *Gaussian Inc.* For more information see <http://www.gaussian.com>.

The *TSSMM* program is distributed as a compressed tar file named `TSSMM.tar.gz`. The distributed files should be uncompressed and extracted in the desired directory. To uncompress and extract the tar file, enter:

```
tar -xvf TSSMM.tar.gz
```

After extracting the files from the tar file, a new directory, `TSSMM`, is created. This directory contains all the files included in the distribution package. If these instructions are followed correctly, the `TSSMM` directory structure should contain the following files and subdirectories:



The `TSSMM_sub` script can be left in the same directory or moved to any directory that user prefers. The `TSSMM_main.pl` file needs to be copied to the desired working directory. Detailed description about running the *TSSMM* calculation will be given in Section 4.

5. PROGRAM DESCRIPTION

TSSMM is written in the PERL language and consists of two scripts which generate an external script (i.e. [job_name]_TSSMM) that can be called by the *Gaussian* program. The overall structure of the calculation is:

primary GAUSSIAN09/16 ↔ [job_name]_TSSMM ↔ secondary GAUSSIAN09/16

where *Gaussian 09/16* denotes the *Gaussian 09* or the *Gaussian 16* program, [job_name] is the *Gaussian* input file name for the primary *Gaussian 09/16* calculation.

The *TSSMM* calculation starts by running a primary *Gaussian 09/16* calculation, which will call the external script ([job_name]_TSSMM) to provide a geometry for further calculations. The [job_name]_TSSMM script, in turn, will write the *Gaussian* input files and call a set of two secondary *Gaussian* single point calculations to acquire the corresponding energies and derivatives at two different spin states as defined by the user. The [job_name]_TSSMM script will extract the results from the formatted checkpoint files of the secondary *Gaussian* calculations and calculate the spin-orbit coupled energies and derivatives and pass them to the primary *Gaussian 09/16* calculations.

The optimization of the local minima and transition state geometries can be done by using any *Gaussian* optimization keywords provided that the chosen keyword is compatible with the `External` keyword so that *Gaussian* can correctly call the external Perl script to acquire the needed energies and corresponding derivatives.

5.1. File usage

In addition to the *Gaussian* input files, *TSSMM* uses two files of its own for data input and output, and generating an external script for the *Gaussian* program:

TSSMM_main.pl	Generates an external script that can be called by the <i>Gaussian</i> program.
TSSMM_sub	Contains all the subroutines for data input and output, updating energies and gradients provided by <i>Gaussian</i> program, and creating secondary <i>Gaussian</i> calculations. More information with regard to each subroutine is provided in the script.

Note:

- (a) The user needs to have a copy of the `TSSMM_main.pl` script and a standard *Gaussian* input file for each *TSSMM* calculation. The `TSSMM_main.pl` script and the input file for the *Gaussian* calculation should always be in the same directory.
- (b) Only one *TSSMM* calculation can be carried out in one directory. Other *TSSMM* or *Gaussian* calculations in the same directory might cause the calculation to crash due to possible overwriting of the temporary files.
- (c) The `TSSMM_sub` script can be put in any directory as preferred by the user. The pathway to the `TSSMM_sub` script needs to be defined in the `TSSMM_main.pl` script as will be mentioned later.

5.2. Perform a *TSSMM* calculation

The *TSSMM* calculation starts by submitting the primary *Gaussian* calculation. In general, a total of three files need to be prepared before the submission, including:

- 4.2.1 *Gaussian* input
- 4.2.2 `TSSMM_main.pl`
- 4.2.3 external basis sets

5.2.1. *Gaussian* input

To start a *TSSMM* calculation, one first needs to provide a *Gaussian 09/16* input file (file with the `.com` or the `.gjf` extension) with the following keyword in the route section in addition to other keywords such as `Opt` or `Freq`:

```
External=[job_name]_TSSMM.
```

Here, `[job_name]` should be replaced by the *Gaussian* input file name of the specific calculation. The `External` keyword will enable the *Gaussian* program to fetch the required energies and derivatives from the external `[job_name]_TSSMM` script for each calculation.

5.2.2. `TSSMM_main.pl`

In addition to the *Gaussian* input file, one also needs to have the `TSSMM_main.pl` script in the same working directory as the *Gaussian* input file. The following parameters need to be specified in the `TSSMM_main.pl` prior to the calculation:

(The `TSSMM_main.pl` file can be edited using any text editor of user's choice.)

Directory related:

```
$GAUSSIAN ' = pathway to the GAUSSIAN executable'. One can usually
acquire this information through the which command. For example, the
```

pathway to the *Gaussian 09* executable “g09” can be acquired by using the command: `which g09`.

- `$TSSMM` ' = pathway to the directory which contains the TSSMM_sub script'. The TSSMM_sub script can be put in any directory of user's choice.
- `$JOB` ' = pathway to the directory which contains the GAUSSIAN input file'. The TSSMM_main.pl script should also be in this directory.
- `$SCRATCH` ' = pathway to a user pre-defined scratch directory'. A scratch directory for the TSSMM calculation defined by the user is preferred. One can use the default scratch directory used by the *Gaussian* program, which is sometimes defined using the following format:
`export GAUSS_SCRDIR=/username/scratch_dir`

Calculation settings:

(Settings between the ' ' signs are only shown as examples)

- `$nproc` ' = %nproc=12 '. Defines the number of processors one would like to use for the calculation.
- `$mem` ' = %mem=36000mb '. Defines the amount of memory one would like to use for the calculation.
- `$functional` ' = uM06L '. Defines the DFT functional one would like to use for the calculation.
- `$option` ' = scf=(xqc) '. Other options one would like to include for the secondary *Gaussian* calculations. For more options <http://www.gaussian.com>
- `$const` ' = # '. Defines the spin-orbit coupling constant for the calculation, where # should be replaced by a number.
- `@multi` ' = (1, 3) '. Defines the two spin multiplicity numbers one would like to consider in the calculation. The example shown refers to a calculation considering the coupling between singlet and triplet state of the geometry.

After one specifies all the parameters as listed above, the users need to run the TSSMM_main.pl script once using the following command:

```
Perl TSSMM_main.pl.
```

This will generate the `[job_name]_TSSMM` external file that can be called by the *Gaussian* program.

5.2.3. External basis sets

The user needs to prepare an external basis set file with the `.gbs` (gaussian basis set) extension to specify the choice of basis set for the DFT calculation. The pathway to the `.gbs` file needs to be included by the end of the *Gaussian* input file manually in the following format:

```
@/home/username/working_directory/basis_set.gbs/N
```

The *Gaussian* keyword `GenECP` should be included in the route section for reading the external basis set file. Note, if one does not specify an external basis set, the STO-3G basis set will be used by default for the *Gaussian* calculation.

5.2.4. \$PATH variable

Before submitting the primary *Gaussian* calculation, the user should make sure that the current working directory (the directory where the *Gaussian* input and the `TSSMM_main.pl` script are stored) is present in the system `$PATH` variable. If not, the directory can be added through the following command:

```
export PATH=/home/username/working_directory:$PATH
```

If the user uses a `.pbs` script for *Gaussian* job submission, the above command can be included in the `.pbs` script as well and will be executed when submitting the `.pbs` script.

5.3. Notes on keywords

We will assume that the user knows how to use *Gaussian 09/16* and is familiar with their keywords. Here, we will only discuss some of the keywords that are necessary for the *TSSMM* calculation or that could cause *TSSMM* to malfunction.

<code>External=[job_name]_TSSMM</code>	Always required in the route section of the corresponding primary <i>Gaussian</i> input file where <code>[job_name]</code> should be replaced by the corresponding <i>Gaussian</i> input file name.
--	---

Opt=(NoMicro)

Mandatory for all types of geometry optimization and internal reaction coordinate (IRC) scans.

Freq

Cannot be used for initiating frequency analysis by the end of a geometry optimization. (In other words, keyword `Freq` does not working properly when keyword `Opt` is specified in the same route section of a *Gaussian* input file.) A separate frequency calculation using a new *Gaussian* input file is necessary for the frequency analysis at a converged geometry.

GenECP

Required for reading external basis set for the *Gaussian* calculation and is automatically applied for all of the secondary *Gaussian* calculations.

6. EXAMPLE OF CALCULATING THE SPIN-ORBIT COUPLING CONSTANT USING THE *ADF* PROGRAM

This section shows an example of how to obtain the spin-orbit coupling constant through a TD-DFT calculation as implemented in the *ADF* program. The *ADF* program is covered by a separate license and is not included in the *TSSMM* package; for more information about the *ADF* program see: <https://www.scm.com/>. The example considers the coupling between the singlet and the triplet spin states of the $\text{Fe}(\text{CO})_n$ system. The coupling constant χ was determined by following the procedure:

(a) Initial saddle point geometry optimization is performed by employing the Perl script along with the *Gaussian* program as described above. An initial guess for the coupling constant χ is provided; this can be determined based on parametrization to an atom or monatomic ion;

(b) A reaction coordinate scan is performed starting with the optimized saddle point geometry from step (a) to locate a representative geometry along the reaction coordinates where the low- and high-spin surfaces are close in energy;

(c) Singlet and triplet energies at the representative geometry are computed using the *ADF2016.103* program package. The effect of spin-orbit coupling is considered by employing the spin-orbit ZORA Hamiltonian. The example uses the M06-L exchange-correlation functional, and a Slater-type ZORA/QZ4P basis set is used for all *ADF* calculations in this study.

(d) The singlet-triplet splitting of the representative geometry is determined by averaging the corresponding singlet and triplet energies as calculated in step (c). The value of the coupling constant χ is updated accordingly such that the difference of the eigenvalues of the following 2×2 matrix reproduces the singlet-triplet splitting.

$$\begin{pmatrix} E_1 & \chi \\ \chi & E_2 \end{pmatrix}$$

Here E_1 and E_2 are the energies of low- and high-spin states computed from the single-point calculations using *Gaussian* for the representative geometry.

(e) Steps (a) to (d) can be repeated until the value of the coupling constant χ converges, but considering the simple nature of the present model, and to reduce the computational cost, one or two iterations should be enough to provide a reasonable value for the constant χ .

Table 1: Calculated single point energies, singlet-triplet excitation energies, and coupling constant χ for the $\text{Fe}(\text{CO})_n$ representative geometry at the final iteration of the aforementioned procedure. All energies are given in hartrees.

	$\text{Fe}(\text{CO})_n$

E_1	-377.17345
E_2	-377.17352
	0.58562
$E_{\text{excitation}}$	0.59541
	0.63180
Coupling constant χ	0.011103

7. EXAMPLE OF A *TSSMM* CALCULATION

7.1. Tests performed

We tested the *TSSMM* script using the revisions D01, and E01 of *Gaussian 09* and revision A.01 of *Gaussian 16* on an HP c7000 Linux Cluster with Intel Haswell E5-2680v3 processors (CentOS Linux).

7.2. Test suite

We included a test suite to provide some examples of the way *TSSMM* works. The tests are in the `tests` directory that contains three subdirectories. Each subdirectory contains one or more tests based on the following reactions:



The jobs are run in a subdirectory created for each run. This way, problems that could be caused when running other *Gaussian* or *TSSMM* calculations in the same directory that *TSSMM* is using are avoided.

7.2.1. Example feco4

Functional: M06-L

Basis set: Def2-TZVPP

This example includes the ground state geometry optimization and frequency analysis of the $\text{Fe}(\text{CO})_4$ molecule on the lowest mixed-spin potential energy surface.

Inputs and outputs for the example run

<code>feco4_TSSMM</code>	<i>TSSMM</i> script generated to interface with <i>Gaussian</i> program for the ground state geometry optimization
<code>feco4.gjf</code>	Gaussian input for geometry optimization using the <i>TSSMM</i> script
<code>feco4.out</code>	Gaussian output for geometry optimization using the <i>TSSMM</i> script
<code>feco4_freq_TSSMM</code>	<i>TSSMM</i> script generated to interface with <i>Gaussian</i> program for frequency analysis
<code>feco4_freq.gjf</code>	Gaussian input for frequency analysis of the optimized geometry using the <i>TSSMM</i> script
<code>feco4_freq.out</code>	Gaussian output for frequency analysis of the optimized geometry using the <i>TSSMM</i> script

7.2.2. Example ts_feco4_co

Functional: M06-L

Basis set: Def2-TZVPP

This example includes the transition state geometry optimization and frequency analysis for the transition state intermediate considering the reaction: $\text{Fe}(\text{CO})_4 + \text{CO} \rightarrow \text{Fe}(\text{CO})_5$. The optimization was performed using the eigenmode following (*Gaussian* keyword TS) methods.

Inputs and outputs for the example run

ts_feco4_co_TSSMM	<i>TSSMM</i> script generated to interface with <i>Gaussian</i> program for geometry optimization and frequency analysis
ts_feco4_co.gjf	Gaussian input for geometry optimization and frequency analysis using the <i>TSSMM</i> script
ts_feco4_co.out	Gaussian output for geometry optimization and frequency analysis using the <i>TSSMM</i> script

7.2.3. Example IRC_feco4_co

Functional: M06-L

Basis set: Def2-TZVPP

This example includes the transition state geometry optimization and frequency analysis for the transition state intermediate considering the reaction: $\text{Fe}(\text{CO})_4 + \text{CO} \rightarrow \text{Fe}(\text{CO})_5$. The optimization was performed using the eigenmode following (*Gaussian* keyword TS) methods.

Inputs and outputs for the example run

ts_feco4_co_TSSMM	<i>TSSMM</i> script generated to interface with <i>Gaussian</i> program for geometry optimization and frequency analysis
ts_feco4_co.gjf	Gaussian input for geometry optimization and frequency analysis using the <i>TSSMM</i> script
ts_feco4_co.out	Gaussian output for geometry optimization and frequency analysis using the <i>TSSMM</i> script

8. ACKNOWLEDGMENTS

The authors are grateful to Kamal Sharkas and Zoltán Varga for helpful discussions. This work was supported as part of the Inorganometallic Catalysis Design Center, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under Award DE-SC0012702.