Transitivity User's Manual

Version 1.0.0

Transitivity: A Code to Computing Kinetic and

Thermodynamic Parameters in Chemical and

Physical Processes

SUMMARY

1. Getting Started

1.1 How to obtain Transitivity

The code Transitivity in python language is available for download in: https://www.vhcsgroup.com/transitivity

1.2 Citations

If you use *Transitivity* for your research or for the preparation of publications, please cite the next reference:

Hugo G. Machado, Flávio O. Sanches-Neto, Nayara D. Coutinho, Kleber C. Mundim and Vincenzo Aquilanti, Valter H. Carvalho-Silva. Transitivity: A Code to Computing Kinetic and Thermodynamic Parameters in Chemical and Physical Processes. *Submitted*. 2019.

1.3 Bug Reports

If you notice any errors or behaviors that are unfamiliar with your results, please report this to our research group. Describe your problem in detail and attach the results (output files, graphics, or other related files). You can also send questions and suggestions to improve the performance of *Transitivity* for:

- Valter Carvalho-Silva: fatioleg@gmail.com
- Hugo Machado: hugogontijomachad@gmail.com

1.4 Requirements

Transitivity can be run on operational platforms (Windows, Linux, Mac,…) with python3 installed (version 3.5 or higher). The python3 can be download from https://www.python.org/downloads/ for several platforms (the fitting function works only on the Windows platform). See attached installation video.

After downloading and installing python3 for your platform, two libraries need to be installed before running Transitivity: Matplotlib (https://matplotlib.org/) and NumPy (http://www.numpy.org/). Open the terminal of your operational platforms, and enter with the following commands:

```
py -m install matplotlib
py -m install numpy
```

Loading the previous libraries the *Transitivity* is ready to be executed. Transitivity need the output files of the Gaussian program (* .out, * .log). For more information about Gaussian: https://www.gaussian.com.

SEE INSTALLATION VIDEO IN HOMEPAGE.

1.5 How to install

With python3, the matplotlib, numpy installed, Transitivity code can be downloaded at www.vhcsgroup.com/transitivity and run the *Transitivity.py* file.

2. Using Transitivity Interface

When you run the *Transitivity* code through the Transitivity.py file you will be presented at a window with three tabs: *Rate Constant, Molecular Dynamic and Fitting*.

Rate Constant: Calculation of the reaction rate constant with Transition-State Theory (TST) for reactions in gas and solution phases.

Molecular Dynamic: Generation of input files for *ab initio* molecular dynamics simulations in the CPMD code. For more information about CPMD: www.cpmd.org/

Fitting: Fit of temperature dependence of the reaction rate constant using formulas predefined.

2.1 Rate Constant

In the *Rate Constant* tab (see Fig. 1) select the Gaussian output file (* .out or * .log) of each of the species involved in the reaction by clicking in the *Open File* button. At least 1 reagent, 1 transition state and 1 product must be selected.

After selecting the output files and, if necessary, changing the available options, click in the *Calculate* button and the following buttons will be activated:

➢ *Species Properties*

➢ *Reaction Properties*
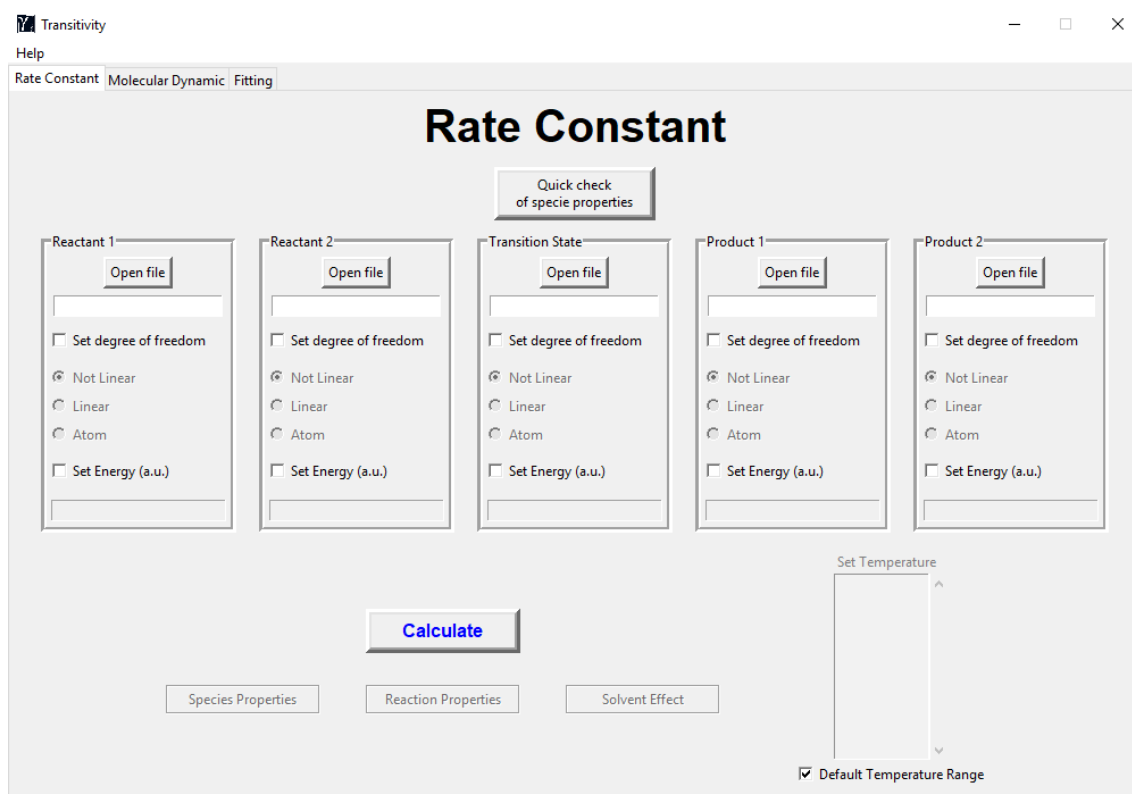
➢ *Solvent Properties*

Figure 1. *Rate Constant* window.

To activate the *Species Properties* and *Reaction Properties* buttons, the only requirement for Gaussian output files is the frequency calculation (atomic species exempt this requirement). To activate the *Solvent Properties* button, it is necessary a calculation in Gaussian program with the VOLUME keyword.

2.1.1 Options

2.1.1.1 Quick check of specie properties

If you want to quickly check the properties of a specific species, click *Quick check of specie properties* and select the output file desired.

2.1.1.2 Geometry and Energy

All the necessary informations for the calculations are extracted directly from the Gaussian output file of each species. However, there is external options: 1) geometry type and 2) the electronic energy of the species in question.

1) Declare the geometry type, select the *Set degree of freedom* option of the species and choose one of the available options: *Not Linear, Linear, Atom*.

2) If you want to declare the electronic energy, select *Set Energy (a.u.)* of the species in question and write the energy value in atomic units in the field below. This option is very useful in case of composite methods where it is possible to estimate reaction rate constant using the geometry from a low-level with energy from a high-level.

When *Set Energy (a.u.)* option is not selected, the code will extract this information directly from the output file.

2.1.1.3 Temperature Range

The reaction rate calculation will be performed for any temperature range. The default temperature range is from 273.15 K to 4000.00 K:

273.15; 298.15; 300.0; 400.0; 500.0; 600.0; 700.0; 800.0; 900.0; 1000.0; 1200.0; 1400.0; 1600.0; 1800.0; 2000.0; 2200.0; 2400.0; 2600.0; 2800.0; 3000.0; 3200.0; 3400.0; 3600.0; 3800.0 e 4000.0.

If you want to declare a specific temperature range, deselect the *Default Temperature Range* and type the desired temperatures in the *Set Temperature* field with each temperature per line.

2.1.2   Species Properties

The *Species Properties* window is the part of the program which analyzes the properties of each individual species (Figure 2).
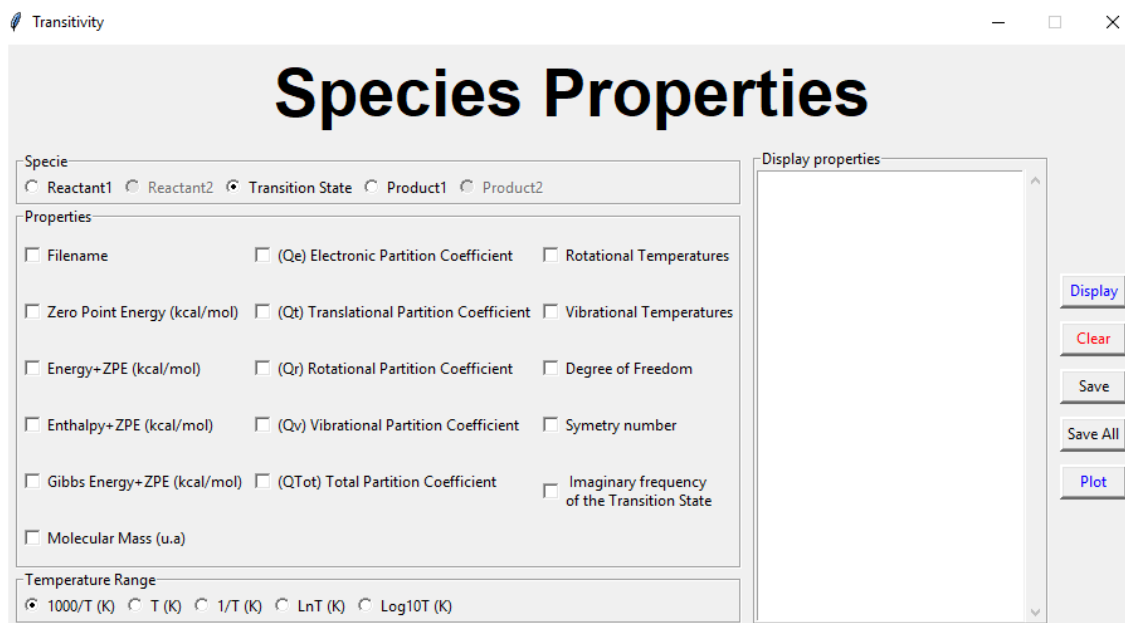

Figure 2. *Species Properties* window

Select in the *Specie* panel the desired species and select in the *Properties* panel the property which should be evaluated. If the desired property varies with temperature it is possible to select different temperature formats in the *Temperature Range* panel. Selecting the species and properties it is possible to: 1) view the values on the screen, 2) save a text file and 3) generate a graph with temperature dependence:

1) Click in *Display* button and the values will be shown in the *Display of Properties* box. If you want to clear the board to enter new properties, click in the *Clear* button.
2) Save a text file selecting the *Save* button. Alternatively, you can also use the *Save All* button to save all available properties.
3) Generate a graph selecting the *Plot* button. Properties with temperature dependence are: (Qt) Translational Partition function, (Qr) Rotational Partition function, (Qv) Vibrational Partition function, and (QTot) Total Partition function. Zero Point Energy, electronic energy, Enthalpy and Gibbs Energy properties are calculated at 298.15 K.

2.1.3 Reaction Properties

The *Reactions Properties* window analyzes the properties of the gas-phase reactions (Figure 3). In this window, it is possible to estimate reaction rate constant with TST and several one-dimensional tunneling corrections, thermodynamic (ΔE, ΔH and ΔG) and kinetic (E_o, d, $v^{\ddagger}$, and α from ST model) properties of the reaction. $\beta = 1/k_B T$, where k_B is the Boltzmann's constant.
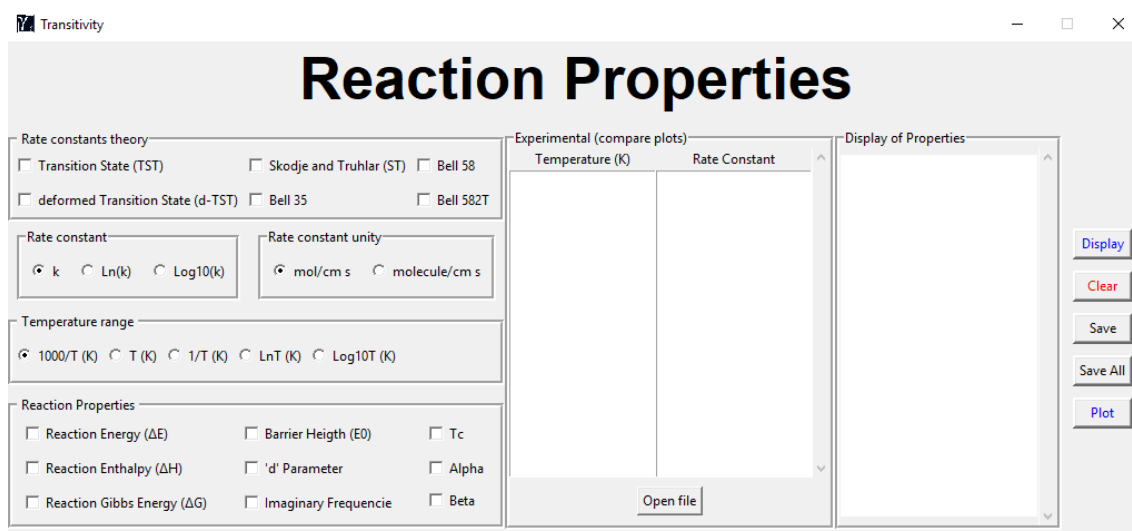

Figure 3. *Reaction Properties* window

It is possible 1) to view the values on the screen, 2) save in a text file and (3) generate a graph with temperature dependence using the same procedure described in Sec. 2.1.2.

2.1.3.1 Reaction Rate Constant

It is possible choose several formats to write the reaction rate constant (k): in the *Rate constant* panel choode k, ln (k) or log10 (k) and in *Temperature range* panel choose the temperature format. For unimolecular reactions, the reaction rate constant unit is s^{-1}. For bimolecular reactions, it is possible to choose mol.cm^{-3}.s^{-1} or molecule.cm^{-3}.s^{-1}.

In the *Experimental (compare plots)* box, it is possible to insert external data of reaction rate constant (with the units above) and temperature (K) to be compared with that estimated theoretical by the *Transitivity* code (comparison is shown in graph using the *Plot* button). Using the *Open file* button, you can import external values from a text file instead of typing them. The file should contain only two columns with temperature (first column) and reaction rate constant (second column). These values must be separated by space ( ), comma (,) or semicolon (;).

2.1.4 Solvent Properties

The *Solvent Properties* window analyzes the properties of reactions in solvent-phase (Figure 4). It is possible to estimate the reaction rate constant with Kramers and Collins-Kimball theories to account for solvent effect, and several properties related to these two theories.
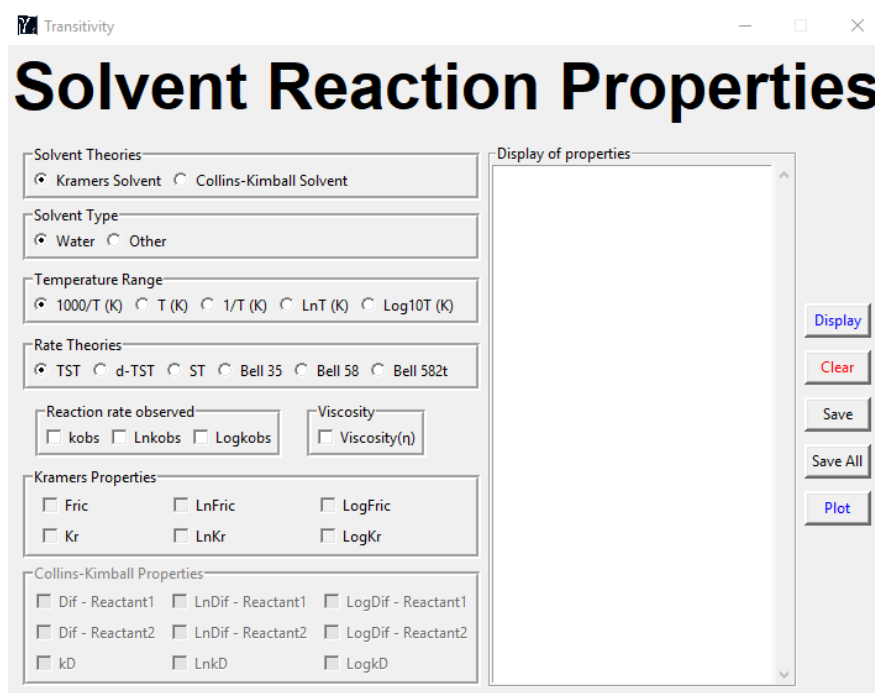

Figure 4. *Solvent Properties* window

When *Kramers Solvent* or *Collins-Kimball Solvent* are selected, a panel contained parameters related to their theories will become active. Collins-Kimball theory can only

be applied in bimolecular reactions, for unimolecular reactions this option will not be available. Kramers procedure can be used to unimolecular and bimolecular reactions.

In the *Solvent Type* panel, water will be selected as default solvent. Another solvent can be inserted, when selecting the *Other* option. In a new window, user can insert temperature dependence of the viscosity with Aquilanti-Mundim formula [$\eta = \eta_o(1 - d\varepsilon/RT)^{1/d}$].

It is possible 1) to view the values on the screen, 2) save in a text file and (3) generate a graph with temperature dependence using the same procedure described in previous sections.

2.2   Molecular Dynamic

In this window, the user can generate CPMD input files for several ab initio molecular dynamics using Gaussian input or output files (Figure 5).
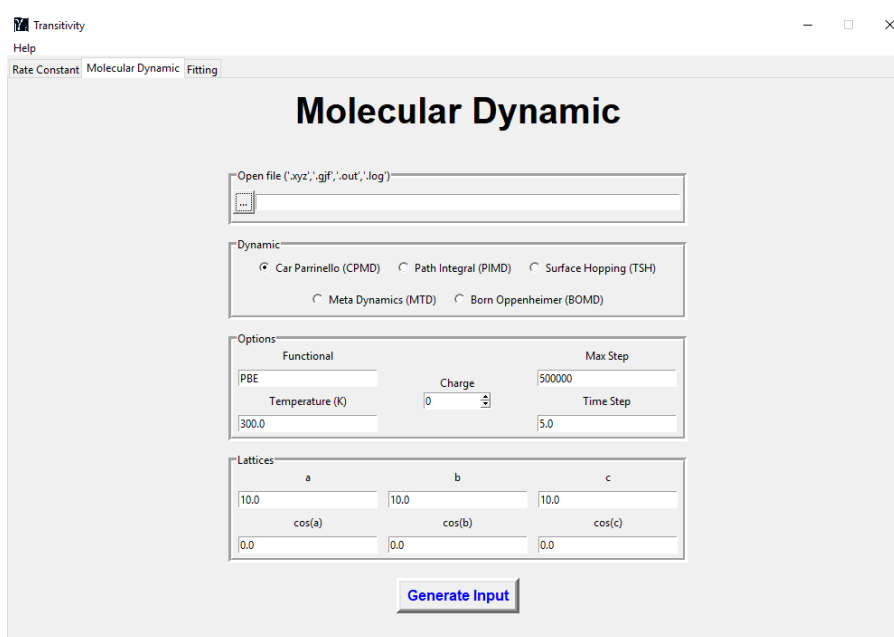

Figure 5. *Molecular Dynamic* window

Using *Open File* panel is possible selected a specific molecular geometry in .xyz, .gjf, .out and .log formats. In the *Dynamic* panel, Car Parrinello (CPMD), Path Integral (PIMD), Surface Hopping (TSH), MetaDynamics (MTD) and Born-Oppenheimer (BOMD) molecular dynamics can be selected. Additional parameters can be controlled in the *Options* and *Lattice* panels.

Selecting *Generate Input* button, input files (.inp) will be created in the same directory where the file with the molecular geometry is located.

2.3 Fitting

Here (Figure 6), user can fit parameters of the reaction rate constant with 5 formulas (Arrhenius, Aquilanti-Mundim, SATO, VFT and ASCC) using GSA algorithm (Force 2.0 compiler - force.lepsch.com/p/download.html). External GSA code in fortran language is located in main directory. This option is only available for Windows plataform.
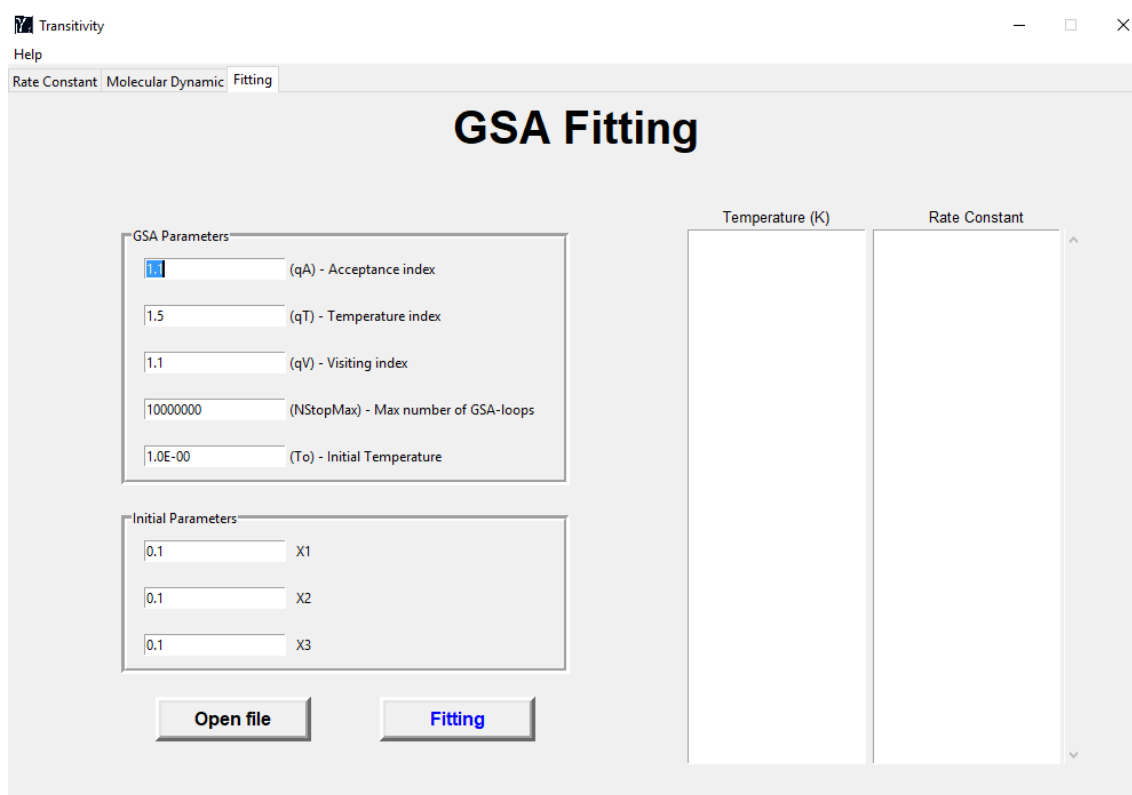

Figure 6. *Fitting* window.

Insert the temperature (K) and reaction rate constants in the *Temperature (K)* and *Rate Constant* panel. Using the *Open file* button, you can import experimental values from a text file instead of typing them. The file should only contain two columns containing respectively the values of temperature (first column) and reaction rate constant (second column). Values should be separated by space ( ), comma (,), or semicolon (;). In the X1, X2 and X3 fields select the initial value that will be assigned to the parameters of the reaction rate constant choosen (see next Section).

Start the procedure with *Fitting* button. Choose the formula to be used and wait the notification "*The process is finished*". Text file containing the fitting parameters will be created in same directory where the file with input data is located. This file contain the formula and parameters fitted with specific units; and Chi-Square value. Five columns are presented with: Temperature (T), Reciprocal of Temperature (1000/T), input reaction rate constant (k-exp), logarithm form of input reaction rate constant (ln k -exp) and logarithm form of reaction rate constant fitted (ln k -Fit). It is possible change GSA parameters in the *GSA Parameters* panel (see details in www.cursosvirtuais.pro.br/gsa/)

2.3.1 Formulas Available in Code

1. **Arrhenius**

$$\ln(k) = \ln(A) - \left(\frac{E_0}{RT}\right)$$

$X1 = \ln(A)$

$X2 = E_0$

2. **Aquilanti-Mundim (*d-Arrhenius*)**

$$\ln(k) = \ln(A) + \left(\frac{1}{d}\right) * \ln\left[1 - \left(\frac{dE_0}{RT}\right)\right]$$

$X1 = \ln(A)$

$X2 = E_0$

$X3 = d$

3. *SATO*

$$\ln(k) = \ln(A) - \frac{E_0}{R\sqrt{T^2 - T_0{}^2}}$$

$X1 = \ln(A)$

$X2 = E_0$

$X3 = T_0$

4. VFT

$$\ln(k) = \ln(A) + \left(\frac{B}{T - T_0}\right)$$

X1 = ln(A)

X2 = B

X3 = T_0

5. ASCC

$$d = \left(-\frac{1}{3}\right) * \left(\frac{E_v}{2E_0}\right)$$

$$\ln(k) = \ln(A) + \left(\frac{1}{d}\right) * \ln\left(1 - \frac{dE_0}{RT + E_v}\right)$$

X1 = ln(A)

X2 = E_0

X3 = E_v

3. Units

Imaginary Frequency	cm^{-1}
Translational Partition Function	$mol^{-1}.m^{-3}$
Viscosity	poise
k_{Obs}	$cm^{-3}.mol^{-1}.s^{-1}$
Friction	s^{-1}
Diffusion	$cm^2.s^{-1}$
E_0	$cal.mol^{-1}$
E_v	$cal.mol^{-1}$
T_0	K
B	K