

# The chemkin User Manual

Jane Huang, Kimia Mavon, Weidong Xu, Zeyu Zhao

## 1 Introduction

`chemkin` is a Python 3 library that computes the reaction rates of all species participating in a system of elementary, irreversible reactions.

### 1.1 Key chemical concepts and terminology

A system consisting of  $M$  elementary reactions involving  $N$  species has the general form

$$\sum_{i=1}^N \nu'_{ij} \mathcal{S}_i \longrightarrow \sum_{i=1}^N \nu''_{ij} \mathcal{S}_i, \quad j = 1, \dots, M. \quad (1)$$

$\mathcal{S}_i$  is the  $i$ th specie in the system,  $\nu'_{ij}$  is its stoichiometric coefficient (dimensionless) on the reactants side of the  $j$ th reaction, and  $\nu''_{ij}$  is its stoichiometric coefficient (dimensionless) on the product side for the  $j$ th reaction.

Each specie is characterized by a concentration  $x_i$ , in units of [mol/vol]. The reaction rate of each specie is the time rate of change of its concentration,  $\frac{dx_i}{dt}$ . The reaction rate is usually represented by the symbol  $f_i$ , such that

$$f_i = \sum_{j=1}^M (\nu''_{ij} - \nu'_{ij}) \omega_j = \sum_{j=1}^M \nu_{ij} \omega_j, \quad i = 1, \dots, N. \quad (2)$$

$\omega_j$  is the progress rate of the  $j$ th reaction,

$$\omega_j = k_j \prod_{i=1}^N x_i^{\nu'_{ij}}, \quad j = 1, \dots, M. \quad (3)$$

The forward reaction rate coefficient  $k_j$  for each reaction is assumed to take one of three possible forms:

1.  $k = \text{constant}$
2. Arrhenius:  $k = A \exp(-\frac{E}{RT})$ , where  $A$  is the pre-factor,  $E$  is the activation energy,  $R$  is the universal gas constant, and  $T$  is the temperature.
3. Modified Arrhenius:  $k = AT^b \exp(-\frac{E}{RT})$ , where  $A$  is the pre-factor,  $E$  is the activation energy,  $R$  is the universal gas constant,  $T$  is the temperature, and  $b$  is the temperature scaling parameter.

## 1.2 Features

The package can solve for the reaction rates of a system with an arbitrary number of species and elementary, irreversible reactions. Reaction system can be either read from an xml file or input as NumPy arrays. For each system of reactions, the user supplies the species participating in the reactions, the chemical equations, the stoichiometric coefficients for the reactants and products, and the rate coefficient parameters (e.g.,  $E$  and  $A$  for Arrhenius rates). For a given system, the user can then specify a temperature and a vector of species concentrations in order to return the reaction rates in the form of a NumPy array.

## 2 Installation

### 2.1 Where to find and download the code

The package is freely available at <https://github.com/cs207group4/cs207-FinalProject>. If you have a GitHub account, you can simply open your terminal and type

```
git clone git@github.com:cs207group4/cs207-FinalProject.git
```

Otherwise, you can download the package by clicking on the green button in the upper right corner of the page that says "Clone or download," then click "Download ZIP" to download the entire repository as a ZIP file.

If you just want to use the library, you can download the single library file at <https://raw.githubusercontent.com/cs207group4/cs207-FinalProject/master/chemkin.py>

### 2.2 Dependencies

This package has several dependencies. These usually come standard with the Anaconda distribution, but are straightforward to install if you do not have them in your Python distribution. (Click on the individual package names to be directed to their website and installation instructions).

- [NumPy](#) v.1.11+

### 2.3 Running the test suite

In order to run the test suite, you need to have [pytest](#) v. 3.00+ and [pytest-cov](#) v.2.5+ installed. Once you download the contents of the repository from GitHub, enter the directory and type `pytest` into the terminal. The results of the test code will be printed out to the terminal.

### 3 Basic Usage and Examples

As an example of basic code usage, we consider the following system of elementary, irreversible reactions:

1.  $\text{H} + \text{O}_2 \xrightarrow{k_1} \text{OH} + \text{O}$
2.  $\text{H}_2 + \text{O} \xrightarrow{k_2} \text{OH} + \text{H}$
3.  $\text{H}_2 + \text{OH} \xrightarrow{k_3} \text{H}_2\text{O} + \text{H}$

Reaction 1 has an Arrhenius rate coefficient with  $A = 3.52 \times 10^{10}$  and  $E = 7.14 \times 10^4$ . Reaction 2 has a modified Arrhenius rate coefficient with  $A = 5.06 \times 10^{-2}$ ,  $b = 2.7$ , and  $E = 2.63 \times 10^4$ . Finally, reaction 3 has a constant rate coefficient of  $k_3 = 10^3$ .

#### 3.1 Input format

In an xml input file, the user provides the species participating in the reactions, the chemical equations, the stoichiometric coefficients, and the rate coefficient parameters. See `rxns.xml` in the repository folder for an example of how to format the input file. To read the xml file and build a `chemkin` object based on it, the user simply call `chemkin.from_xml(file_name)`. For example:

```
reactions = chemkin.from_xml('rxns.xml')
```

#### 3.2 Computing reaction rates

An initialized `chemkin` object can be called to calculate reaction rates given temperature and concentration. For example:

```
reactions.reaction_rate_T([[1],[1],[1],[1],[1],[1]],1000)
```

User could also set a temperature and calculate reaction rates multiple times on different concentrations. For example:

```
reactions.set_rc_params(T=1000)
reactions.reaction_rate([[1],[1],[1],[1],[1],[1]])
```

#### 3.3 Full example

The following example shows how to read reaction systems from an xml and calculate the reaction rate under  $T = 1000$  and all concentration is 1.

```
from chemkin import *
reactions = chemkin.from_xml('rxns.xml')
reactions.reaction_rate_T([[1],[1],[1],[1],[1],[1]],1000)
```

There are more use cases of our library, please refer to `demo.ipynb`