

# Estimating Reaction Rate Constants with Neural Networks

Benedek Kovács, János Tóth

**Abstract**—Solutions are proposed for the central problem of estimating the reaction rate coefficients in homogeneous kinetics. The first is based upon the fact that the right hand side of a kinetic differential equation is linear in the rate constants, whereas the second one uses the technique of neural networks. This second one is discussed deeply and its advantages, disadvantages and conditions of applicability are analyzed in the mirror of the first one. Numerical analysis carried out on practical models using simulated data, and our programs written in *Mathematica*.

**Keywords**—neural networks, parameter estimation, linear regression, kinetic models, reaction rate coefficients

## I. INTRODUCTION

A fundamental problem of (homogeneous) chemical kinetics is to elucidate the mechanism of complex chemical reactions. This process consists of collecting all the possible species taking part in a reaction, enumerating the elementary steps (perhaps by systematically decomposing an overall reaction [9]), and finally, determining the reaction rate coefficient of all the elementary steps in such a way that solving the kinetic differential equations of the complex chemical reaction consisting of the elementary steps with the calculated rate constants have the same concentration versus time curves as those measured in the experiment. The problem is a kind of an estimation problem and the methods presented may be applied to models from various sciences too.

The first steps to the usual approach of this problem are almost trivial. Suppose we have  $N_c$  chemical species and  $N_k$  elementary reaction steps, and the kinetic differential equation is:

$$c(t) = f(c(t), k) \quad c(0) = c^0, \quad (1)$$

where  $c(t), c_0 \in \mathbb{R}^{N_c}, k \in \mathbb{R}^{N_k}$ . Then one defines the error as the integral of the difference between the calculated and measured concentration versus time functions:

$$Q(k) = \int_0^T \|c_{\text{calculated}}^k(t) - c_{\text{measured}}(t)\|^2 dt, \quad (2)$$

and tries to find the minimum of this function.

Let us mention a few problems of the basic approaches [2],[15].

- 1) Not all the concentrations are measured.

Manuscript received February 28, 2007. The present work has partially been supported by the National Science Foundation, Hungary (T047132, K2N: 63066).

B. Kovács is with the Department of Analysis, Institute of Mathematics, Budapest University of Technology and Economics, Budapest, Egrý J. u. 1. H-1521 HUNGARY (e-mail: kovacsbe@math.bme.hu).

The corresponding author J. Tóth is with the Department of Analysis, Institute of Mathematics, Budapest University of Technology and Economics, Budapest, Egrý J. u. 1. H-1521 HUNGARY (phone: 361-463-2314; fax: 361-463-3172; e-mail: jtoth@math.bme.hu).

- 2) Measurements are not taken continuously, the discrete time points of measurement may even differ for the different concentrations.
- 3) The concentrations are measured with error.
- 4) There are dependencies between the parameters, only some functions of the parameters can be estimated from the data.
- 5) The function  $Q$  usually has a number of local minima.

Therefore, estimating the reaction rate constants even with today's computational capacities is a kind of art.

In this paper we present two novel methods to solve the problem above. Each one of them might be useful under certain circumstances. The first method utilizes the fact, that the right-hand-side of the kinetic differential equations is almost always (e.g. in the case of mass action kinetics) a linear function of the parameters to be estimated. This method has been proposed in [7], here we elaborate the method in more detail, and apply them to realistic models and use it as a reference to analyze our new approach. The second method uses neural networks which has turned out to be really useful in many areas such as the calculation of models as e.g. turbulent combustion or potential energy surfaces [1],[5],[16].

## II. ESTIMATION METHODS

### A. Theory of the Matrix Inversion Method

The basic idea comes from the standard problem of estimating the parameters in linear models [13]. Let us consider the complex chemical reaction consisting of the reaction steps

$$\sum_{m=1}^{N_c} \alpha(m, r) X(m) \rightarrow \sum_{m=1}^{N_c} \beta(m, r) X(m). \quad (3)$$

The usual mass action type model of this reaction is the Cauchy problem:

$$\begin{aligned} c_m(t) &= \sum_{m=1}^{N_c} (\beta(m, r) - \alpha(m, r)) k(r) \\ &\quad \prod_{p=1}^{N_c} c_p^{\alpha(p, r)}(t) \\ c_m(0) &= c_m^0 \quad (m = 1, 2, \dots, N_c), \end{aligned} \quad (4)$$

where  $c_m(t) = X(m)(t)$  ( $m = 1, 2, \dots, N_c$ ) is the concentration of the  $m$ th species, and  $k_r$  ( $r = 1, 2, \dots, N_k$ ) is the reaction rate coefficient of step (3). Obviously, (4) is of the form

$$c(t) = F(c(t))k \quad c(0) = c^0 \quad (5)$$

with the linear operator on any function of  $c$ ,  $F(c(t)) \in \mathbb{R}^{N_k} \rightarrow \mathbb{R}^{N_c}$  compressing the structure of the complex chemical reaction in a certain way.

Mass action kinetics is enough to ensure the linear dependence of the right hand side on the parameters to be

estimated. However, it is not fully necessary: the condition underpinning our method may be satisfied in other cases, as well. Furthermore, if the right hand side is an inhomogeneous linear function of the parameters to be estimated, our method works still with a slight modification. The generalization of Eq. (5) can then be rewritten as

$$c(t_n) - c^0 = \int_0^{t_n} F(c(s)) \cdot s \cdot k + \int_0^{t_n} F_r(c(s)) \cdot s, \quad (6)$$

or,

$$\begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ (c(t_n) - c^0) - \int_0^{t_n} F_r(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}^\top \begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix} \cdot k. \quad (7)$$

If the matrix  $\begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}^\top \begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}$  is invertible (a pseudoinverse of  $\begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}$  can always be calculated), an estimate of  $k$  can be obtained:

$$k = \left( \begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}^\top \begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix} \right)^{-1} \begin{pmatrix} \int_0^{t_n} F(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}^\top \begin{pmatrix} (c(t_n) - c^0) - \int_0^{t_n} F_r(c(s)) \cdot s \\ \int_0^{t_n} F(c(s)) \cdot s \end{pmatrix}. \quad (8)$$

We have an exact formula to obtain  $k$  that means if there is no measurement error and no numerical error in the integration, the result would equal to the real value of  $k$  with no estimation needed. In the cases when this simple method does not work, we can calculate the integrals at different  $t_n$  values thus get a large set of linear equation systems and become able to run multivariate regression on it. (Having calculated the integrals numerically, the method already requested many measurements, and a simultaneous measurement of the concentrations at every  $t_n$  so these can be used for the regression analysis too.)

### B. Theory of the Neural Networks Method

A neural network, a network of formal (McCulloch-Pitts type) neurons can be used to estimate parameters of a given model. Our method and its main properties will be explained in this section.

If we use an Artificial Neural Network (ANN) to estimate the rate constants, we need some a priori conditions or distribution functions for the values of the unknown parameters  $\kappa$ , which is a vector with  $N_\kappa$  components of  $k$ . This may come from biological, chemical, or mathematical theories.

Estimating the rate constants means to find out what their values are using measured data of each state variable:  $c_m$ . Let the function  $I$  realize this:

$$I \cdot (c_m(t_1), c_m(t_2), \dots, c_m(t_T)) \mapsto k \quad (9)$$

The estimated parameters ( $\theta$ ) are some components of  $k$ . We will use a regular feedforward neural network with a backpropagation training algorithm to simply approximate the

function  $I$ . The main idea is that we can build the training set with the an "inverse"-like function of  $I$ ! The inverse function can not be determined since even if we knew the estimated parameters, the measurement error could not be determined exactly. So our "inverse"-like function becomes:

$$G \cdot \kappa \mapsto (c_\kappa(t_1), c_\kappa(t_2), \dots, c_\kappa(t_{N_t})). \quad (10)$$

We numerically evaluate this function at some  $\kappa$  values: *locations* determined by its PDFs. The exact methods use are discussed in Section III-B.

We use a neural network to interpolate  $I$  since it is usually a very complex function mapping  $\mathbb{R}^{N_t \cdot N_\kappa} \rightarrow \mathbb{R}^{N_\theta}$  where  $1 \leq N_\theta \leq N_\kappa$ . (The numerical value depends on many parameters in the method.) The input data of the neural network(s) is a set of concentrations calculated with the  $G$  function on different  $\kappa$  parameters and the output is some of its components, the  $\theta$  set of parameters, to be estimated. This is the training phase which is the most sensitive part and determines the success.

Provided a trained network we easily get the estimated rate constants as

$$\theta = \text{ANN}(c_m(t_1), c_m(t_2), \dots, c_m(t_T)), \quad (11)$$

where  $c(t_i)$  denotes the measured value of the parameter vector at the  $i$ th moment.

As a last step, the estimated parameters have to be verified with evaluating the model numerically using the estimated values of the parameters. The result (the value of the estimated concentrations) can be compared to the measured ones applying the definition of the error function Eq. (2).

## III. NEURAL NETWORK METHOD ISSUES

### A. The parameters used for the supervised training

Supervised training means means to use the output of the neurons to determine the error and modify the synaptic strengths of the network with some strategy. At the output side there can be either all or a subset of the parameters examined. Also a different subset can be the estimated ones.

Since we work with the supervised backpropagation training algorithm the constructed network might highly depend on the number of unknown variables examined at the output of the network. However, examining all the parameters needs significantly more computation when the network is trained and often makes the examination more exact (in the case if we do not overtrain the network). We suggest that for large systems only a limited number of parameters should be estimated with one network because the accuracy of the estimation is not too much worse while the training becomes easier.

### B. The problem of the many unknown parameters

For the *Neural network method* theoretically, there can be an unlimited number of unknown parameters in the model under examination. Of course there are limitations for both this number and the number of rate constants can be estimated.

The input of the network is roughly the numerical solution of the differential equation system. To obtain this we need

numerical values for the unknown parameter vectors, (the selected  $\kappa$ s). If there are too many components in  $\kappa$  to deal with than having every possible parameter vector even if there are only two possible values for each parameter would end up in an exponentially large training set:  $2^k$ . This is a problem one has to think about when tries to build a neural network with too many unknown parameters.

There are many approaches how to select the parameter vectors [6]. We select a couple of them to apply:

- The first fixes a set of parameters which are not included to the supervision algorithm. Even if these parameters are fixed to a wrong but not impossible value the training of a network succeeds and the estimation can be done with an accuracy good enough. This method cannot be used when there are too many parameters to fix or if the model is sensitive to the accuracy of the fixed values.
- The second method selects a set of values of each parameter and a set of the possible  $\kappa$  vectors. This turned out to perform quite the same way and have the feature of disregarding which is the target parameter to be estimated.
- There can be Monte–Carlo–like solutions for this problem. It is shown in [8] that this works better than selecting the parameter vectors randomly.
- The Latin Hypercube is a widely used method for such a problem. The basic concepts of it is that it divides the density function of a parameter into areas with equivalent probability. Then the value of a parameter is taken randomly from one of these regions. A supervised selection of the possible  $\kappa$  vectors is used to avoid that they are focused around a certain point or points in the parameter space.

### C. The problem of measurement error

According to the neural networks theory it is useful to add a very small noise to the numerically computed training set. This makes the function to be interpolated more smooth and better. Theoretically the function what can be interpolated with a neural network depends on the number of neurons it has and on the structure of it. Usually Sigmoid function is used as a threshold function which is monotonously increasing. This limits the number of monotonous regions of the function can be interpolated [16].

### D. The problem of the number of training set elements

For a given network it is easy to calculate the number of the synaptic strengths we have ( $w$ ) which has to be smaller than the elements in the training set  $s$ . There are no exact solutions for this problem also but we say that we should have at least  $\rho \geq 1$  times more training set elements than weights:  $s \geq w \cdot \rho$ . We can calculate  $w$  from the network structure having

- $m$ : the number of the measurement moments;
- $v$ : the number of the measured variables in each moment;
- $o$ : the number of output (supervised) parameters;
- $i$ : the number of the hidden layers;
- $h$ : the number of hidden neurons in a hidden layer.

It is known that the training of the network is more efficient if there are the same number of neurons in the layers. The neural network method can also be used even if the variables are measured in different moments and different times but now we focus on the simplest case where all the measured variables are measured at the same moments. Now we can calculate  $w$  and then:

$$s \geq ((m \cdot v + o) \cdot h + (i - 1) \cdot h^2) \cdot \rho. \quad (12)$$

There are limitations for  $s$  in practice we can say that it is necessary to keep it small. As it was already mentioned there is a minimum threshold for the numbers of neurons in the network to interpolate the function in question. It is easy to consider that more measurements should make the estimation better but on the other hand makes the function to be interpolated less smooth. So basically we can say that the more neurons needed implies  $s$  to be bigger. Since our aim is to keep  $s$  rather small to avoid exponential computational complexity it is likely that we should use a minimum number of measurements that still makes the estimation good enough.

If one is having a rather large equation system there might be rate constants and concentrations depending on each other. Sometimes not all the concentrations can be measured too. Regarding to these facts the question of dismissing some of the concentrations from the estimation method comes up. We expect that it would make the estimation more efficient without heavily decreasing its accuracy.

There are a couple of parameters more that can be influenced in the equation above (12). We can agree that  $\rho$  is a constant although there is no theory yet to define its exact value. The number of output neurons  $o$  for the supervised parameters is already discussed in Section III-B.

## IV. EXAMPLES

### A. The triangle reaction

A reaction often studied from the theoretical point of view is the triangle reaction (see e.g. [12]) with the kinetic differential equation:

$$\begin{aligned} c_1 &= -k_1 c_1 + k_2 c_2 \\ c_2 &= -k_2 c_2 + k_3 c_3 \\ c_3 &= -k_3 c_3 + k_1 c_1. \end{aligned} \quad (13)$$

For our purposes we rewrite the differential equation (13) of the triangle reaction into the following form:

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \begin{pmatrix} -c_1 & 0 & c_2 \\ c_1 & -c_2 & 0 \\ 0 & c_2 & -c_3 \end{pmatrix} \cdot \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}. \quad (14)$$

Originally there are three rate constants to be estimated. To use the inverse method the right hand side should linearly depend on the rate constants, this is true. However the derived matrix of (13) can not be inverted. We have to reduce the problem and estimate only two of the rate constants. As one of the concentrations can be expressed with the two others there cannot be more than two parameters estimated. Also

$c_1 + c_2 + c_3$  is constant so using  $c_1(0) = 1, c_2(0) = 0, c_3(0) = 0$  as initial conditions, the equation can be rewritten:

$$\begin{aligned} c_1 &= -k_1 c_1 + k_3(1 - c_1 - c_2) \\ c_2 &= -k_2 c_2 + k_1 c_1 \end{aligned} \quad (15)$$

Adapting the neural networks method to the triangle reaction does not need any extra operation. However, estimations with better accuracy are achieved when one of the three state variables is substituted.

### B. A model to describe the transport of butadiene

A simplified model to describe butadiene transport is

$$\begin{aligned} c_1 &= -(k_1 + k_2 + k_3 + k_4 + k_5)c_1 \\ &\quad -k_6 c_2 - k_7 c_3 - k_8 c_4 - k_9 c_5 - k_{10} c_6 \\ c_2 &= k_1 c_1 - k_6 c_2 \\ c_3 &= k_2 c_1 - (k_7 + k_{11})c_3 \\ c_4 &= k_3 c_1 - (k_8 + k_{12} + k_{13})c_4 + k_{14}u \\ c_5 &= k_4 c_1 - k_9 c_5 \\ c_6 &= k_5 c_1 - (k_{10} + k_{12})c_6 \end{aligned} \quad (16)$$

with the initial condition  $c_1(0) = c_2(0) = c_3(0) = c_4(0) = c_5(0) = c_6(0) = 0$ . Here the function  $u$  describes the input, the butadiene inhaled by the investigated living being and  $c_1 \cdot Q_B, c_2 \cdot Q_{Fa}, c_3 \cdot Q_{Mu}, c_4 \cdot Q_{Li}, c_5 \cdot Q_{Lu}, c_6 \cdot Q_{Vi}$ , with  $Q_X$  denoting the quantity of butadiene in the Xth compartment; (X=B(lood), Fa(t), Mu(scle), Li(ver), Lu(ngs), Vi(scera)). The interested reader may find the biological details of the model in References [4],[14]. Decomposing the model with our notation above it turns out that only a subset of the parameters can be estimated with the inverse method and the neural network method still does not need any restrictions but a couple of preconditions for the parameter values. These conditions, especially if they are simple bounds, are easy to create since we have our application from a real-world problem.

### V. PARAMETER ESTIMATIONS WITH NEURAL NETWORKS METHOD

Since all the other methods have a lot of restrictions on the model to be estimated it is not really fair to compare them with the *Neural Networks Method*. Thus, we focus on the method and its problem itself. The numerical analysis is done with a program written in *Mathematica* [17] that implements both the methods, training set builder strategies, special training strategies and uses the *Neural Networks Package* of *Mathematica*.

1) *Examination of a triangle reaction:* We simulated our sample data with  $p_1 = 1, p_2 = 10, p_3 = 0.1$ . We trained several networks on various kinds of the parameter vector chosen. Here we present the result gained by a couple of successful ones by means of small error in the error function defined with Equation 2. It can be seen that small error is obtained with various results of the parameters. We can see that there can be problems with the verification what will be discussed in an upcoming work.

TABLE I  
ESTIMATED PARAMETERS FOR THE TRIANGLE REACTION WITH OUR METHOD OF NEURAL NETWORKS.

$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	SE( $\hat{\theta}$ )
0.001	4.11248	50	0.0804988
0.001	2.25639	50	0.0801564
0.001	0.72205	0.1	0.0760324
0.001	0.0151956	50	0.075681

2) *Results for the Butadiene transport:* Here, separate networks are used for each variable (that also implies that only one unknown parameter was supervised at the output of the network), and typically a rather small network of one hidden layer consisting of 5 neurons was used. The estimation results with two error functions are presented on Table II, one for the quadratic errors  $E_Q(i) = (\theta_i - \hat{\theta}_i)^2$ , another for the per cent error for each parameter  $E_P(i) = \frac{|\hat{\theta}_i - \theta_i|}{\theta_i/100}$ . Table III shows the error of the concentrations.

TABLE II  
ERROR OF THE BUTADIENE REACTION PARAMETERS

param.	$\theta_i$	$\hat{\theta}_i$	$E_Q(i)$	$E_P(i)$
$p_{12}$	0.01994	0.00992	0.0001004	50.2553
$p_{21}$	4.40816	4.02575	0.146237	8.67504
$p_{13}$	10.9914	9.86131	1.27712	10.2817
$p_{31}$	19.5918	23.5071	15.3292	19.9841
$p_{33}$	199.728	219.185	378.588	9.74194
$p_{14}$	160.514	148.071	154.826	7.75194
$p_{41}$	122.449	129.802	54.068	6.00503
$p_{40}$	437.764	500.568	3944.27	14.3464
$p_{400}$	1200	1114.06	7385.95	7.1618
$p_{4466}$	13.5	15.4851	3.94046	14.7042
$p_{15}$	1.18299	1.19148	0.000072	0.718041
$p_{51}$	44.2041	60.3704	261.349	36.5719
$p_{16}$	30.2877	43.7812	182.075	44.5511
$p_{61}$	54.6122	50.8239	14.3516	6.93682

TABLE III  
ERROR OF THE ESTIMATED BUTADIENE CONCENTRATIONS

St. var.	SE $_{Q,F9}(\hat{\theta})$	SE $_{AP,F9}(\hat{\theta})$	SE $_Q(\hat{\theta})$	SE $_{AP}(\hat{\theta})$
$c_1$	8.87101	23.3943	12.6193	57.8262
$c_2$	72.3346	32.9736	1369.24	99.1937
$c_3$	0.59479	15.7107	0.0576155	56.68
$c_4$	8.08449	12.5928	10.6377	51.5965
$c_5$	15.5919	1.59616	51.6394	59.0677
$c_6$	20.3892	47.3815	65.1695	64.4624

### VI. CONCLUSIONS ON ADVANTAGES AND DISADVANTAGES OF THE METHODS

We have presented a method with some extra properties that might be useful in applications and what classical methods do

not have. The basic problems are discussed and some principles and guidelines are laid down about the selection of the training set and the network what is the most difficult question when one uses Neural Networks as a function approximator. Let us explain a couple of properties of the *Neural Network Method* that no other methods have:

- Not all the concentrations must be measured and there can be almost any number of parameters to be estimated.
- The training takes time but *needs no measurements* and once a network is trained for a model it can do the estimation even if the parameters vary with the measurements.
- There is no need to measure all the variables in the same moments what is often very useful and we neglect the inaccuracy of simultaneous measurements.
- There can be any kind of input output model that can be evaluated at certain parameter vectors, this method is not restricted on any specific application.

It takes a large work to train a neural network for a given structure but once it is done the estimation is easy and can be applied to the same reaction under different circumstances as well (different initial concentrations, temperature, pressure etc.).

#### ACKNOWLEDGMENT

We learned the idea of using neural networks to estimate parameters of differential equations from Dr. B. Paláncz.

#### REFERENCES

- [1] J. S. Almeida, E. O. Voit, "Neural-Network-Based Parameter Estimation in S-System Models of Biological Networks". *Genome Information*. vol. 14, 2003, pp 114–123.
- [2] Y. Bard, "Nonlinear Parameter Estimation" New York, Academic Press, 1974.
- [3] S.M. Blower and H. Downlatabadi, "Sensitivity and Uncertainty Analysis of Complex Models of Disease Transmission: an HIV Model, as an Example", *International Statistical Review*, vol. 62, no. 2, pp 29–243, 1994.
- [4] C. Brochot, J. Tóth, F. Y. Bois, "Lumping in pharmacokinetics", *Journal of pharmacokinetics and Pharmacodynamics* vol. 32 no. 5-6, pp 719–736, Dec. 2005.
- [5] F. C. Christo, A. R. Masri, E. M. Nebot, "Utilising artificial neural network and repro-modelling in turbulent combustion", *Proceedings of the IEEE International Conference on Neural Networks*, Perth, 27th November-1st December 1995, vol. 1, pp 911–916, 1995.
- [6] M. Grindal, J. Offutt and S. F. Andler, "Combination Testing Strategies: A Survey", *GMU Technical Report ISE-TR-04-05*, vol. 15, no. 3, July 2004, pp 167–199.
- [7] K. M. Hangos, J. Tóth, "Maximum likelihood estimation of reaction-rate constants", *Computers and Chemical Engineering*, vol. 12, pp. 135–139, 1998.
- [8] K. M. Hanson, "Half-toning and Quasi-Monte Carlo", *Los Alamos National Library*, pp. 430–442, 2005.
- [9] K. Kovács, B. Vizvári, M. Riedel, J. Tóth, "Decomposition of the permanganate/oxalic acid overall reaction to elementary steps based on integer programming theory", *Physical Chemistry, Chemical Physics*, vol. 6, 1236–1242, 2004.
- [10] A. Liaqat, M. Fukuhara, T. Takeda, "Optimal estimation of parameters of dynamical systems by neural network collocation method", *Computer Physics Communications*, vol. 150, pp 215–234, 2003.
- [11] M. T. Manry, S. J. Apollo, Y. Qiang, "Minimum mean square estimation and neural networks", *Neurocomputing*, vol. 13, pp 59–74, 1996.
- [12] I. Nagy, Gy. Póta, J. Tóth, "Detailed balanced and unbalanced triangle reactions", *Journal of Chemical Education*, submitted for publication
- [13] C. R. Rao, *Linear Statistical Inference and Its Applications*, John Wiley & Sons, Inc.: New York, London, Sydney, 1965.
- [14] T. Smith, Y.-S. Lin, L. Mezzetti, F. Y. Bois, K. Kelsey, J. Ibrahim, "Genetic and dietary factors affecting human metabolism of 1,3-butadiene", *Chemico-Biological Interactions*, vol. 132, no. 0, pp 407–428, 2001.
- [15] A. S. Tomlin, T. Turányi, M. J. Pilling, "Mathematical tools for the construction, investigation and reduction of combustion mechanisms", in *Low temperature combustion and autoignition*, J. Peters, Ed. Amsterdam: Elsevier, 1997, pp 293–437.
- [16] J. B. Witkoskie, D. J. Doren, "Neural Network Models of Potential Energy Surfaces: Prototypical Examples", *Journal of Chemical Theory and Computation*, vol. 1, pp 14–23, 2005.
- [17] <http://www.wolfram.com/>