MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **79** (2018) 619-644

ISSN 0340 - 6253

Application of a Robust Hybrid Algorithm (Neural Networks-AGDC) for the Determination of Kinetic Parameters and Discrimination among Reaction Mechanisms

J. L. González-Hernández, M. Mar Canedo*, Sonsoles Encinar
Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca. E-37008 Salamanca. (Spain)

(Received October 13, 2017)

Abstract

In this paper a Hybrid Algorithm (HA) is applied to determine the kinetic parameters and the discrimination between mechanisms responsible for the development of a chemical reaction. The HA used is formed by a combination of two complementary algorithms that are applied sequentially: the method "*soft-modelling"* of Artificial Neural Networks (ANN) and the Mathematical Optimization Algorithm, AGDC. The consecutively application of these methods means a great advantage due to the ANN methodology which is a treatment that does not (need to) use initial estimates of the parameters to determine. Initially the *soft-modelling* ANN methodology is applied and the obtained results are used as initial estimates in the second method (AGDC) that uses these values because it is a gradient optimization method.

The Hybrid Algorithm (ANN-AGDC) is applied to determine the individual rate constants that correspond to three different reaction models in which the several different species, the reactions between the species and the rate constants are involved. First, the "*soft-modelling"* ANN methodology is applied because it is not necessary to have initial estimates

(*) ***Author for correspondence***
e-mail: mcanedo@usal.es

of the rate constants and after, the values obtained for the kinetic constants, are used as initial estimates for the application of the second method of the HA, the AGDC algorithm.

The results of the application of the HA let us to establish the most probable reaction mechanism responsible of the experimental kinetic data, since the methodology has the capacity of discrimination between the different models that are theoretically applicable to the chemical reaction. This robust algorithm HA, provides the rate constants of the stages reactions of the global mechanism and further, it is able to discriminate between several different possible models that represent a great very valuable advantage in the field of Modelling in Chemical Kinetics.

1. Introduction

One of the main objectives of Chemical Kinetics has been the determination of kinetic parameters and the discrimination between reaction mechanisms, the development and application of diverse methodologies and computational techniques for the treatment of kinetic data, it has allowed a great advance in the achievement of this objectives and in the development of the Computational Chemical Kinetics.

In the literature is possible to find numerous computational methods to determine kinetic, thermodynamic, analytical parameters… as the treatments *Hard-* and *Soft-modelling*, based both on non-linear regression techniques and other computational methodologies. The traditional methods of curve fitting [1-7] utilize different optimization algorithms to determine chemical kinetics parameters of interest and they are used for the study of different kinetic systems. We have developed and used the Mathematical Optimization Algorithm AGDC implemented in diverse computational applications [4-7], which has allowed the calculation of different kinetic, thermodynamic and analytical parameters.

The treatment of kinetic data and the calculation of the parameters is also performed by other techniques such as: Classic Curve Resolution techniques and their modifications, Classical Curve Resolution *Hard-Modelling* (CCR-HM) [8], Classical Curve Resolution *Soft-Modelling* (CCR-SM) [15] or a combination of both, Classical Curve Resolution Combining *Hard-* and *Soft-Modelling* (CCR-CHSM) [9].

An alternative for these techniques of kinetic treatment are the computational methodologies based on the Artificial Neural Networks (ANN). It is a powerful *soft-modelling* tool for the treatment of multivariable and multi-response data that is applied nowadays in numerous Chemistry fields and in particular in Chemical Kinetics. In the literature it is possible

to find different ANN based methods that are applied in the study of certain reaction mechanisms to determine diverse kinetic parameters as the rate constants [10,11].

The application of ANN with quantitative purposes in Chemical Kinetics involves performing numerous *training*s of the Neural Network to obtain its optimal architectures, although the model is simple, this makes the method to proceed slow and tedious and sometimes it will not produce acceptable results. These aspects are analyzed in a previous paper [12] in which we obtain the first results to perform the computational treatment with the application of a new computational methodology ED-ANN that is successfully applied to kinetic data of concentration, obtaining excellent results. In this same way [13] it is performed the determination of rate constants, developing a procedure in which first an Experimental Design of the type *Star Composite Experimental Design* (CSCED) is performed and later the *prediction* of the rate constants is carried out through ANN. In this paper, the software provided by MATLAB [14] was used for the design of the Network's architecture, as well as for the process of *training* and *prediction*. The procedure is applied to determine the rate constants of the reaction that takes place between 2-mercaptoethanol with Carbonyl Cyanide 3-Chlorophenylhydraone ((3-Cl)-PHPD).

The simultaneous determination of the reaction's mechanism and the estimate of the rate constants, is an important aspect in Chemical Kinetics and in which diverse methodologies have been used. The evaluation of the mechanism of a chemical reaction implies investigating all possible species involved in the reaction, enumerate the elementary steps (*modelling*) and determine the rate constants of reaction of all elementary steps. When the mechanism of the reaction is complex several chemical species can be involved and take place between them a considerable number of elementary reactions that will make the *modelling* process particularly complex. In the kinetic investigation of complex mechanisms different methodologies have been used [15], between them we could find techniques of mathematical optimization [5], it is also interesting the application of methodologies based on Artificial Neural Networks [16,17]. The application of ANN based methods for the treatment of experimental data from different types of chemical reactions has revealed that it is an efficient method to solve *modelling* problems in catalytic reactions or for the treatment of enzymatic systems.

In this paper we have developed a Hybrid Algorithm (HA) to determine the individual rate constants (k_r) of diverse reaction mechanisms, besides this algorithm allows the discrimination between mechanisms that could be responsible of the development of the reaction. The HA developed is formed by a combination of two algorithms that are based in different mathematical principles and that are applied sequentially: the *soft-modelling* method of the

Artificial Neural Network (ANN) and the Mathematical Optimization Algorithm AGDC (General Algorithm of Controlled Descent). The treatment of kinetic data through HA means in the first stage applying the ANN methodology to determine the values of kinetic parameters, the values of the parameters obtained through the application of ANN are used as initial estimates of the optimization algorithm (AGDC) applied in a second stage with the objective of improving the final values of the parameters.

The application of the ANN methodology, offers a great advantage from other parameter determination methods because it is not necessary to start with initial estimates of them, unlike the mathematical optimization methods that need estimated values of the parameters to determine, further it is convenient that the estimates are close to the real values of the parameters to guarantee the successful optimization process. The HA developed throughout this paper determine, in the first place, values of the individual rate constants (k_r) by means of the application of ANN and these values are used as initial estimates in the mathematical optimization algorithm AGDC.

The first version of this Hybrid Algorithm has been applied to determine the Activation Thermodynamic Parameters (ATP) in different reaction mechanisms [18-20] through the treatment of kinetic data of non-isothermal kinetics. First the application of the algorithm was tested through the treatment of synthetics kinetic data endowed with a random error of experimental order [18,19], subsequently the treatment of experimental kinetic data was carried out [20]. In these papers we checked the application of the HA (ANN-AGDC) to non-isothermal kinetics that allows to determine ATP, directly, with no necessity to determine the kinetic constants in a previous step. In the case of non-isothermal experiments a large group of experimental data through a single replicated kinetic are obtained, so that the treatment of data is complex. If a classic treatment is executed it is necessary to carry out mathematical transformations of the equations and the experimental data that produce a modification, both the intrinsic errors as well as errors and accuracies in the values of the parameters obtained. This problem does not exists when HA (ANN-AGDC) is used, the results of these papers prove that HA can be perfectly applied because it is a rigorous and robust method that has a great advantage which is a procedure that does not need initial estimates of the parameters to determine.

2. Theoretical aspects

2.1. Chemical kinetics aspects

A chemical system formed by n_r chemical elementary reactions where n_s chemical species involved, can be represented [21,22]:

$0 = \nu_{1,1}\mathrm{B}_1 + \nu_{2,1}B_2 + \nu_{3,1}B_3 + \cdots\cdots + \nu_{\mathrm{n_s},1}B_{\mathrm{n_s}}$

$0 = \nu_{1,2}B_1 + \nu_{2,2}B_2 + \nu_{3,2}B_3 + \cdots\cdots + \nu_{\mathrm{n_s},2}B_{\mathrm{n_s}}$

$0 = \nu_{1,3}B_1 + \nu_{2,3}B_2 + \nu_{3,3}B_3 + \cdots\cdots + \nu_{\mathrm{n_s},3}B_{\mathrm{n_s}}$

.....

$0 = \nu_{1,\mathrm{n_r}}B_1 + \nu_{2,\mathrm{n_r}}B_2 + \nu_{3,\mathrm{n_r}}B_3 + \cdots\cdots + \nu_{\mathrm{n_s},\mathrm{n_r}}B_{\mathrm{n_s}}$

It can be expressed for the *r-th* reaction with the generic equation:

$$0 = \sum_{j=1}^{n_s} \nu_{j,r} B_j \qquad (1)$$

B_j= chemical species involved in the system of reactions.

r= (1,…, $\mathrm{n_r}$) number of chemical reactions.

j= (1,…, $\mathrm{n_s}$) number of chemical species.

k_r= kinetic rate constant of the *r-th* reaction.

$\nu_{j,r}$= stoichiometric coefficient of the species B_j in the *r-th* reaction.

The variation in the concentration of each species B_j with respect to the time *i* is given by the general differential equation:

$$\frac{\mathrm{d[B_j]}}{\mathrm{dt}} = \sum_{\mathrm{r=1}}^{\mathrm{n_r}} \mathrm{k_r \nu_{j,r}} \prod_{\mathrm{l=1}}^{\mathrm{n_s}} [\mathrm{B_l}]^{|\nu_{\mathrm{l,r}}|} \qquad (2)$$

where: $\mathrm{k_r}$ = rate constant of the reaction *r*; $[\mathrm{B_l}]$ = concentration of the species acting as reagents in the reaction *r* ($\nu_{l,r} < 0$); $\nu_{l,r}$ = stoichiometric coefficient of the species B_l in the reaction *r*.

Each of the species involved in the mechanism of the reaction provides a differential equation and therefore we have a system of differential equations (ODEs) whose resolution provides the concentration of each of species with respect to time ($[B_j]_{t_i}$).

In this work we have studied three chemical systems formed by 3 first order reactions that involve 3 chemical species:

Model I

$$\boldsymbol{\nu_{1,1}B_1 \xrightarrow{k_{12}} \nu_{2,1}B_2}$$

$$\boldsymbol{\nu_{2,2}B_2 \xrightarrow{k_{21}} \nu_{1,2}B_1}$$

$$\boldsymbol{\nu_{2,3}B_2 \xrightarrow{k_{23}} \nu_{3,3}B_3}$$

Considering $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,2} = -1, \nu_{1,2} = 1, \nu_{2,3} = -1, \nu_{3,3} = 1$ we have:

$$B_1 \underset{k_{21}}{\overset{k_{12}}{\rightleftarrows}} B_2 \xrightarrow{k_{23}} B_3$$

The system of ODEs is as follows:

$$\frac{d[B_1]}{dt} = -k_{12}[B_1] + k_{21}[B_2]$$

$$\frac{d[B_2]}{dt} = k_{12}[B_1] - k_{21}[B_2] - k_{23}[B_2]$$

$$\frac{d[B_3]}{dt} = k_{23}[B_2] \tag{3}$$

Model II

$$\boldsymbol{\nu_{1,1}B_1 \xrightarrow{k_{12}} \nu_{2,1}B_2}$$

$$\boldsymbol{\nu_{2,3}B_2 \xrightarrow{k_{23}} \nu_{3,3}B_3}$$

$$\boldsymbol{\nu_{3,2}B_3 \xrightarrow{k_{32}} \nu_{2,2}B_2}$$

Considering $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,3} = -1, \nu_{3,3} = 1$, $\nu_{3,2} = -1, \nu_{2,2} = 1$, we have:

$$B_1 \xrightarrow{k_{12}} B_2 \underset{k_{32}}{\overset{k_{23}}{\rightleftarrows}} B_3$$

For this mechanism the ODEs is as follow:

$$\frac{d[B_1]}{dt} = -k_{12}[B_1]$$

$$\frac{d[B_2]}{dt} = k_{12}[B_1] - k_{23}[B_2] + k_{32}[B_3]$$

$$\frac{d[B_3]}{dt} = k_{23}[B_2] - k_{32}[B_3] \tag{4}$$

Model III

$$\boldsymbol{\nu_{1,1}B_1 \xrightarrow{k_{12}} \nu_{2,1}B_2}$$

$$\boldsymbol{\nu_{2,2}B_2 \xrightarrow{k_{23}} \nu_{2,3}B_3}$$

$$\boldsymbol{\nu_{3,3}B_3 \xrightarrow{k_{31}} \nu_{1,3}B_2}$$

Considering $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,2} = -1, \nu_{2,3} = 1$, $\nu_{3,3} = -1, \nu_{1,3} = 1$, we have:

$$B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3 \xrightarrow{k_{31}} B_1$$

In this case we have the following ODEs:

$$\frac{d[B_1]}{dt} = -k_{12}[B_1] + k_{31}[B_3]$$

$$\frac{d[B_2]}{dt} = k_{12}[B_1] - k_{23}[B_2]$$

$$\frac{d[B_3]}{dt} = k_{23}[B_2] - k_{31}[B_3] \tag{5}$$

To determine the concentration of all the species in the period of time considered, ($[B_j]_{t_i}$), the numerical solution of the ODEs is performed using different methods (Gear algorithm, Runge-Kutta, etc,..). If the experimental data are expressed in absorbance, we have to consider the Beer-Lambert-Bouguer law:

$$A^{\lambda}_{j,t_i} = \varepsilon^{\lambda}_j \cdot l \cdot [B_j]_{t_i} \tag{6}$$

where A^{λ}_{j,t_i} is the absorbance of the species B_j at time t_i and at a wavelength λ, ε^{λ}_j is the molar absorption coefficient of the species j at a wavelength λ, l=1 cm is the path length and $[B_j]_{t_i}$ is the molar concentration of the species j at time t_i. The total absorbance of a mixture formed by n_s chemical species, A^{λ}_{T,t_i}, at a wavelength λ and a time (t_i), must be the sum of the contribution of all the j species showing absorption al the wavelength, would be:

$$A^{\lambda}_{T,t_i} = \sum_{j=1}^{n_s} A^{\lambda}_{j,t_i} = \sum_{j=1}^{n_s} \varepsilon^{\lambda}_j \cdot [B_j]_{t_i} \tag{7}$$

2.2. Hybrid algorithm (ANN-AGDC)

A Hybrid Algorithm (HA) is one in which two or more different algorithms are combined in order to solve the same mathematical problem. It is a combination of algorithms that solve the same problem, but differ in their characteristics, in particular in efficiency on its application. The HA proposed in this paper is able to incorporate different techniques resulting in an algorithm that is a combination of all procedures and it provides better results that individual algorithms. The final aim is to reach the most suitable combination of the characteristics from each one, so that the general global HA can function better than each of the methods individually.

In this paper we utilized a Hybrid Algorithm designed in our laboratory to determine kinetic parameters in several reaction mechanisms, the application of the HA allows the discrimination between the possible mechanism that can be responsible of the course of the reaction. The HA [18-20], utilizes two methods based in different mathematical principles and it is applied sequentially in two steps. First the methodology *soft-modelling* based on Artificial Neural Networks (ANN) and later the AGDC mathematical optimization algorithm is applied.

The ANN methodology has a great advantage, it is no necessary to provide initial estimates of the parameters to ~~be~~ determine. The results obtained in the output matrix are the initial estimates to start the optimization process with the AGDC optimization algorithm, continuing with the second step in the treatment. The mathematical optimization methods need estimated values from the parameters to determine, in addition it is suitable that these estimations are close to the actual values of the parameters to assure the success of the optimization process. The HA (ANN-AGDC) developed along this paper first determines the parameter values by applying ANN and these values are used as initial estimates of the mathematical optimization algorithm AGDC. This algorithm is a numerical method for mathematical optimization gradient of second order, efficient and robust, capable of reaching the global minimum desired ensuring the success of the final optimization of the parameters.

2.2.1. Artificial neural networks (ANN)

An Artificial Neural Network (ANN) is a systematic data processing method, consisting of a large number of simple and highly interconnected elements. It is not based on an explicit algebraic model, but on a set of activation units called nodes or artificial neurons. The neurons are connected to each other through a network and are structured in layers [23]. The multilayer neural network uses sets of input data and parameters (called *Targets)*, distributed in 2 input matrices when MATLAB is applied. The elements of the input matrix are the data, where one row contains a single curve and all the curves thus obtained (n_c) are grouped in an *Input data* matrix. The *Target* matrix is formed by the sets of parameters (n_p). In our case, the input data matrix contained the kinetic data of all curves expressed in A_T (total absorbance) and the *Target* matrix ($n_c x n_p$) contained the set of kinetic rate constants (k_{mn}). Formally, a multilayer neural network is an oriented graph in which the nodes represent a set of processing units, called neurons, and the connections represent the information flow channels. Each connection between two neurons has an associated value called *"weight"* which specifies the strength of the connection between neurons. Positive and negative values determine excitatory and inhibitory connections, respectively. The choice of a specific class of networks for the approximation of a nonlinear map depends on a variety of *factors* dictated by the context and is related to the desired accuracy and the prior information available concerning the input-output pairs.

The neurons that form the first layer of a multilayer neural network receive the input data values from the elements of the input data matrices. This information is transmitted from the i-*th* neuron of a layer to the j-*th* neuron of the subsequent one, with a weight w_{ji}. A neuron

parameter (*"bias")* is summed with the weighted inputs of the neurons and passed through the transfer function to generate the output of the neurons. The following layer is the *Hidden*, in each neuron of this layer the weighed inputs coming from the previous one are summed with each other and added to a *"bias"*. The result is then transformed by means of a suitable mathematical function to obtain an output called "*activation of the neuron*", which is transferred to the neurons in the next layer after another weighing step. The output parameters values are calculated in the last layer *(Output* layer) by means of a suitable transformation function**.** This process is called *training* or *learning* of the multilayer neural network and constitutes an iterative method where after each iteration (*"epochs"*), the calculated values of the parameters are grouped in the *Output* matrix (b_{ij}^{output}) and they are compared with those of the corresponding curve in the *Target* matrix (b_{ij}^{target}).

If n_c is the number of curves, n_p is the number of parameters and $n_c x n_p$ are the dimensions of both matrices (*Output* matrix and *Target* matrix), the value of the Mean Squared Error (MSE), expressed in absolute value, is calculated according to the expression:

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{output} - b_{ij}^{target})^2}{n_p . n_c} \right)^{1/2} \quad (8)$$

The values of *"weights"* and *"bias"* are modified during the process of *training*, by means of suitable mathematical optimization algorithms in order to minimize the calculated values of MSE in each *epoch*. In the present work, the *back-propagation* algorithm was used. The iterative process finishes when the minimum value of MSE is reached, after which the *training* process can be considered to be completed.

The optimal architecture and topology of the multilayer neural network is determined ~~in~~ to obtain the best results when ANN is applied to the system under study. We have used a method of *"trial and error"* by minimizing the MSE values obtained for the different possible configurations of the same number of *Hidden* layer/s. It must to determine the minimum value (optimum) of the MSE for all possible configurations for the *Hidden* layer/s chosen. For each *Hidden* layer, a graph of MSE values *vs.* the number of neurons *s*hows that initially, for the lower configuration, the value of the MSE decreases rapidly when the number of neurons increases, but after a constant value a poor improvement is obtained. The optimum number of neurons in that *Hidden* layer is given by the point of intersection of the two branches of the graph. Sometimes, a small minimum appears near this intersection point. The architecture of the Neural Network can be written in abbreviated notation as (n_{inp}, n_{hid}, n_{out}), where n_{inp} is the number of neurons in the *Input* layer, n_{hid} in the *Hidden* layer and n_{out} in the *Output* layer.

When the processes of *"validation"* and *"testing"* reach satisfactory results the Neural Network *training* is completed. These are 2 control and verification processes of the iterative minimization method between the elements of the *Output* and *Target* matrices. Among the different curves comprising the *Input* matrix, random choice is made of a percentage of the total, established previously (5%,10%...), which gives rise to a "*sub-matrix*" of input curves that are subjected to iterative optimization until a minimum MSE value is reached. It is thus possible to verify the validity of the *training* process by ensuring that it is convergent, that it has an appropriate termination, and that there has not been any "*overfitting"*, since any possible *"overtraining"* has been taken into account. Validation is completed when in a given number (≥6) of consecutive *"epochs"* the MSE remains constant or shows a slight tendency to increase. The *"testing"* process is similar, except that the control of the process is performed by controlling the computation time instead of the number of *"epochs"*.

Once the network *training* is completed the process of *prediction* is carried out. This process consists in the determination of the unknown parameters from a set of experimental data after application of the optimal and trained neural network. Obviously, the elements of the *"target"* matrix are unknown for this *"prediction"* process, and only the *Input data* matrix is provided to the Neural Network. In our case, the elements of the *Input data* matrix in the process of *prediction* are experimental kinetic values (A_T, A_j, $[B_j]$, α_j, etc), acquired from a system of reactions developed at the laboratory.

2.2.2. AGDC algorithm

The second method that forms the Hybrid Algorithm (ANN-AGDC) is the AGDC mathematical optimization algorithm. This method allows the determination of different parameters by means of a second-order gradient method that minimizes the numerical function (SQD) given by:

$$\mathrm{SQD}(\mathbf{X}) = \sum_{i=1}^{N_d} \left(\left(A_{T,t_i}^{\lambda}\right)_C - \left(A_{T,t_i}^{\lambda}\right)_E \right)^2 \quad (9)$$

where $\mathbf{X}$ is the vector that contains the parameters to be optimized $X_1, X_2, \ldots, X_p$, in this case the kinetic rate constants (k_r); N_d = Number of experimental data pairs; $\left(A_{T,t_i}^{\lambda}\right)_E$ = Total absorbance value obtained at a wavelength λ; $\left(A_{T,t_i}^{\lambda}\right)_C$ = Total absorbance value calculated (Eq.9).

The minimization of the SQD function (Eq. 9) was carried out by means of an iterative process that first uses as the movement vector ($\boldsymbol{p}^{(m)}$) the one indicated by the Gauss-Newton method [5-7]:

$$\boldsymbol{p}^{(m)} = -\boldsymbol{g}^{(m)}\left[\boldsymbol{H}^{(m)}\right]^{-1} \tag{10}$$

where $\boldsymbol{g}^{(m)}$ and $\left[\boldsymbol{H}^{(m)}\right]^{-1}$ are respectively the gradient vector and the inverse of the Hessian matrix of the iteration *m*, whose terms are derived from the function to be minimized (SQD) with respect to each of the parameters to be determined (**X**). The identity of the components of the Gradient vector and Hessian matrix is different depending on the parameters to optimize.

If the parameters to determine are the kinetic constants (k_r) then $\boldsymbol{g}^{(m)}$ and $\boldsymbol{H}^{(m)}$ are given by:

$$\boldsymbol{g} = 2\begin{bmatrix} \sum_1^{N_d} \mathrm{RES_i} \frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_1} \\ \sum_1^{N_d} \mathrm{RES_i} \frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_2} \\ \sum_1^{N_d} \mathrm{RES_i} \frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_3} \\ \vdots \\ \sum_1^{N_d} \mathrm{RES_i} \frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_r} \end{bmatrix} \tag{11}$$

$$\boldsymbol{H} = 2\begin{bmatrix} \sum_1^{N_d}\left(\frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_1}\right)^2 & \dots & \sum_1^{N_d}\left(\frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_1}\frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_r}\right) \\ \cdots & \dots & \dots \\ \sum_1^{N_d}\left(\frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_r}\frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_1}\right) & \cdots & \sum_1^{N_d}\left(\frac{\partial\left(A^{\lambda}_{T,t_i}\right)_C}{\partial k_r}\right)^2 \end{bmatrix} \tag{12}$$

The residuals *RES* are given by:

$$RES_i = \left(\left(A^{\lambda}_{T,t_i}\right)_C - \left(A^{\lambda}_{T,t_i}\right)_E\right) \tag{13}$$

The AGDC algorithm, performs a rigorous analysis and control of the movement vector and of each of its terms, and suitable modifications can be made if any errors are detected, thereby ensuring successful optimization. Once the optimization process has been achieved, the program determines the errors of the optimized parameters and performs an exhaustive analysis of the residuals thus allowing the goodness of fit to be checked. The statistical calculated parameters are the following ones: arithmetic mean, variance, standard deviation, square error, statistical measure of adjustment and Pearson function (χ^2).

The procedure followed is indicated of schematic and reduced form:

1. Start the optimization process. m = 0 (Iteration number).

1.1. Select the mechanism of reaction to be studied.

1.2. Select the parameters to be optimized $\boldsymbol{X}$ (k_r).

1.3. Input data: Initial estimates of the unknown parameters $\boldsymbol{X}^{(0)}$ (Values of *outputs* from ANN), Experimental data of absorbance/time $\left(A^{\lambda}_{T,t_i}\right)_E$, Initial concentrations $[B_j]_0$, convergence criteria *CC*.

2. Determinate the $SQD^{(0)}$ function (Eq. 9).

2.1. Calculate of concentrations of species $[B_j]_{t_i}$: a) Establish the rate differential equations system (Eq. 2), b) Numerical solution of the rate differential equations system.

2.2. Calculate the total absorbance at each wavelength $\left(A^{\lambda}_{T,t_i}\right)_C$ (Eq. 9).

3. AGDC OPTIMIZATION ALGORITHM

3.1. Calculate the vector of movement $\boldsymbol{p}^{(m)}$ (Eq. 10). Compute partial numerical derivatives of $\left(A^{\lambda}_{T,t_i}\right)_C$ with respect to the parameters to be determined $\boldsymbol{X}^{(m)}$ by the central difference method [24]. Compute Gradient vector and Hessian Matrix ($\boldsymbol{g}^{(m)}$ and $\boldsymbol{H}^{(m)}$). Compute $(\boldsymbol{H}^{(m)})^{-1}$.

3.2. Control and correction of the direction of the vector of movement $\boldsymbol{p}^{(m)}$

3.2.1. If $\boldsymbol{H}^{(m)}$ is singular, $\boldsymbol{p}^{(m)} = -\boldsymbol{g}^{(m)}$ go to 3.3.

3.2.2. If $\boldsymbol{p}^{(m)}\boldsymbol{g}^{(m)} < \varepsilon$ (ε = scalar close to zero), $\boldsymbol{p}^{(m)} = -\boldsymbol{g}^{(m)}$ and go to 3.3.

3.2.3. If $\boldsymbol{p}^{(m)}\boldsymbol{g}^{(m)} > 0$, $\boldsymbol{p}^{(m)} = -\boldsymbol{p}^{(m)}$.

3.3. Control the length of the vector of movement $\boldsymbol{p}^{(m)}$

3.3.1. Compute the scalar ($\alpha^{(m)}$) by the method of Hartley [25].

3.3.2. $\boldsymbol{X}^{(m+1)} = \boldsymbol{X}^{(m)} + \alpha^{(m)}\boldsymbol{p}^{(m)}$

3.3.3. Determine the $SQD^{(m+1)}$ function (Eq. 9).

3.3.4. If the Goldstein-Armijo criterium [26] is satisfied go to 3.4.

3.3.5. $\alpha^{(m)} = \alpha^{(m)} / 2$ go to 3.3.2.

3.4. Calculate $CON = \left|\frac{SQD^{(m+1)} - SQD^{(m)}}{SQD^{(m)}}\right|$

3.5. If convergence is not attained ($CON > CC$), set $m = m + 1$ and go to 3.1.

4. $\mathbf{X}^{(m+1)}$ = Optimized Parameters. Calculation of the errors of the parameters.

5. Statistical analysis of residuals.

6. END.

3. Computational aspects

The computational treatment of ANN has been performed by means of the application of Matlab *"Neural Networks Toolbox"* [14] with the creation of user´s interfaces *(GUI)* including the appropriate analysis of Residuals and errors (MSE, SD, etc). We design and perform~~ing~~ specific computational executable programs (##.m type) in the MATLAB environment using "M" language, functions and applications for obtaining synthetic data.

The mathematical optimization process was carried out with the AGDC algorithm by means of the computational program KINMODEL(AGDC) [5,6]. The program is formed by a main program and a series of subprograms, in which the different treatments and calculations necessary for the optimization process of AGDC are carried out. It has been structured in the following parts:

Main program, **KINMODEL** performs the following functions:

- Input data: Initial estimates of the unknown parameters (Values of *outputs* from ANN), Experimental data of absorbance/time Initial concentrations, convergence criteria *CC.*
- Generates the model to study taking into account the data entered by the user.

Subprograms:

- OPTIMAGDC, collection of subroutines that perform the optimization process of the different parameters by applying the AGDC algorithm.

- GEARAGDC, a package of subroutines that generates and solves the set of differential rate equations according to the model considered and therefore determines the concentration of each species in the time interval considered.

- DERIVAGDC, a set of subroutines that calculate the numerical partial derivatives of the total absorbance of the sample with respect to the parameters to be optimized.

- INVERAGDC, calculates the determinant and performs the inversion of the Hessian matrix.

- ESTADAGDC, subprogram that determines the errors of the parameters and performs statistical analysis of the residuals.

4. Results and discussion

In this paper we have studied the applicability of Hybrid Algorithm (ANN-AGDC) for the determination (*prediction*) of the individual rate constants and discrimination between reaction mechanisms that could be responsible development of it. To achieve this objective three different reaction models are analyzed, in these models the same number of species involved, the same number of reactions take place between them and therefore consist of the

same number of rate constants. Through the application of HA the rate constants for each model are determined and in view of the results obtained after the application of HA, the possibility of discrimination between the mechanisms is analyzed. The models are different in regards to the architecture (lineal or cyclic) or in regards to the distribution of the individual reactions or kinetic constants.

***Model I*)** $B_1 \underset{k_3}{\overset{k_1}{\rightleftarrows}} B_2 \overset{k_2}{\rightarrow} B_3$ (Lineal Structure)

***Model II*)** $B_1 \overset{k_1}{\rightarrow} B_2 \underset{k_3}{\overset{k_2}{\rightleftarrows}} B_3$ (Lineal Structure)

***Model III*)** $B_1 \overset{k_1}{\rightarrow} B_2 \overset{k_2}{\rightarrow} B_3 \overset{k_3}{\rightarrow} B_1$ (Cyclic Structure)

The determination process (*prediction*) of rate constants and the discrimination between different models implies the following steps:

- Generate kinetic data corresponding to each of the models (I, II and III).
- Treatment of these data with HA (ANN-AGDC) assuming applicable in all cases the models I, II and III and *prediction* of the individual rate constants corresponding to each one of the models, through the sequential application of the two methods that form HA: ANN methodology and the Mathematical Optimization Algorithm AGDC.
- Discrimination: in regards to the results obtained after the application of HA deduce the model that is responsible for the course of the reaction.

4.1. Experimental design

The first step for the application of the ANN methodology is to generate a series of kinetic curves through the numeric resolution of the rate differential equation system (ODE). For that, we use groups of three rate constants (*Targets* for *training*), organized according to an Experimental Design (ED) type *Central Star Composite Experimental Design*, (CSCED). The set of 3 rate constants selected for ED's central point is the following: $k_1 = 0.07\ min^{-1}, k_2 = 0.06\ min^{-1}, k_3 = 0.05\ min^{-1}$. The molar absorption coefficients of each of the species involved in the reaction are: $\varepsilon_1 = 20000,\ \varepsilon_2 = 18000,\ \varepsilon_3 = 15000$. In this precise case the ED's space is a 3-D model, because we have three parameters to determine, in other words, each of the rate constants of the reaction mechanism considered (Figure 1).

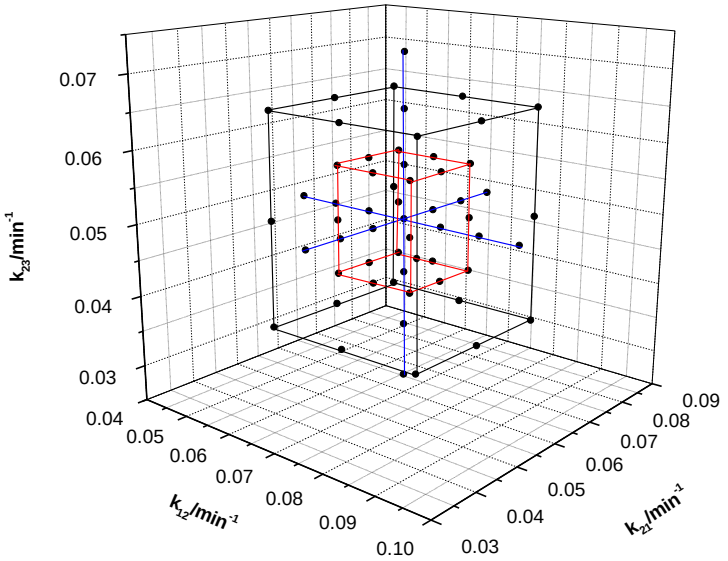

Figure 1. Experimental Design (CSCED), 3 *factors* (k_1, k_2 and k_3), 59 groups of three rate constants (*Targets* for *training*) with which kinetic curves for Neural Network *training* are generated (*Inputs* for *training*).

4.2. Generation of kinetic data: *Inputs* for *training* and *Inputs* for *prediction*

From the sets of constants obtained in the corresponding ED (*Targets* for *training*), we have generated, for each one of the models (I, II and III), a set of 59 kinetic curves. These kinetic curves are formed by total absorbance data generated by the application of equation 9 after the numerical solution of the corresponding system of differential rate equations (ODE) (Eq. 3, 5, 7). The absorbance kinetic data is assigned a random error (*noise*) of an order of equal magnitude to the experimental ($\pm 1.10^{-4}$). These data forms the matrix of *Inputs* for *training* (dimensions 59x50) and in conjunction with the matrix of *Targets* for *training* (59x3) are supplied in the Neural Network to proceed with the *training*.

Subsequently we generate for each one of the three models, eight kinetic curves with fifty absorbance values for each one. These kinetic curves are generated from groups of 3 kinetic constants (Table 1), different to the ones used to generate the synthetic data with which it has carried out the *training*. We impose a value of noise to the kinetic data and distribute

them in three matrices of *Inputs* (*Inputs* for *prediction*), with dimensions 8x50, corresponding to each one of the three models, this matrices are the ones later used to perform the *prediction* process.

Table 1

k_1/min^{-1}	k_2/min^{-1}	k_3/min^{-1}
0.058750	0.048750	0.061250
0.066250	0.056250	0.053750
0.073750	0.063750	0.046250
0.081250	0.071250	0.038750
0.081250	0.071250	0.061250
0.073750	0.063750	0.053750
0.066250	0.056250	0.046250
0.058750	0.048750	0.038750

Table 1. Values of k_1, k_2 and k_3 used to generate the kinetic curves that form the matrices *Inputs* for *prediction* for the *Models I*, *II* and *III*.

4.3. Determination (*prediction*) of rate constants and discrimination of reaction mechanisms.

4.3.1. Discrimination of Model I from Models II and III.

From the data that form the *Inputs* matrix (size 59x50) (*Inputs* for *training*) and the *Targets* matrix (59x3) *(Targets* for *training)* generated for the *Model I*, the *training* of the Neural Network is performed with the purpose of determining the optimal architecture. To reach our purpose the influence of different variables is evaluated in the process of *training*:

- Number of *Inputs* data *(n_d)*.
- Configuration or number of neurons of the *Input* layer.
- Number of synthetic curves *(n_c)* that are supplied to the network.
- Variation of the reaction percentage (conversion) in each kinetic curve.
- Relative values of optimal distribution percentages of the process of *Training (TR) Validation (VL), Testing (TS*), of the global process of *Training*.
- Different relationship of the values $\varepsilon_1/\varepsilon_2/\varepsilon_3$
- Optimal number of *Hidden* layers and optimal number of neurons per layer.
- *Output* layer: Influence of the application of diverse algorithms of optimization.

Considering all these aspects, a large number of processes in the ANN *training* are carried out and performed with different structures (number of layers) and configurations (ideal

number of nodes). In this case, the optimal architecture has a configuration of three *Hidden* layers and the number of nodes in each layer is 9/10/10.

Once the Neural Network has been *trained* for the *Model I*, we provide to the Neural Network the three *Inputs* matrices (*Inputs* for *Prediction*) that correspond to each one of the three models, given in each case to the corresponding *Outputs* matrices ***[Y]*** (size 59x3), that have the values of kinetic constants that the ANN predicts for each model. The results are shown in Table 2:

2A) *Prediction* of the rate constants from the data corresponding to *Model I* performed with the optimal network trained for *Model I*.

2B) *Prediction* of the rate constants from the data corresponding to *Model II* performed with the optimal network trained for *Model I*.

2C) *Prediction* of the rate constants from the data corresponding to *Model III* performed with the optimal network trained for *Model I*.

Analyzing the three *Outputs* ***[Y]*** matrices, we observe that exclusively the one that corresponds to the *prediction* process of the rate constants from data corresponding to *Model I*, there exists a coincidence between the values of the kinetic constants of *Outputs* with those who served to generate the eight curves. However, in the other two *Outputs* ***[Y]*** matrices corresponding to *Models II* and *III*, besides the non-existing conformity, absurd values are obtained (even negative values of the kinetics constants) that show the failure in the *prediction* of the values of kinetic constants for these two models with the Network *trained* exclusively for *Model I*. Therefore, the Neural Network is able to distinguish between different models in the process of *prediction* taking only the right values of the rate constants when they are supplied the *Input* kinetic data which belongs to the model that was used for *training*. Otherwise, when it does not belong to the model mentioned, the process of *prediction* fails, obtaining ridiculous values and ultimately, unacceptable.

The second step of HA consist on the process of mathematical optimization through the application of the algorithm AGDC using the program KINMODEL(AGDC) [5,6] for the determination of the rate constants of the stages that form the reaction *Models I, II* and *III*. To perform the optimization process we have used as initial estimates the values of the constants that have been obtained through the process of *prediction* executed with ANN. The analysis of the results and the statistical parameters obtained (deviation of each one of the constants) have allowed to decide which one is the mechanism that better represents the absorbance data [5], in this case it corresponds to *Model I*.

Table 2

2A	ANN			Deviation		
	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev % k_2)	Dev %(k_3)
Model I	0.057558	0.056723	0.068536	2.0398	-16.355	-11.895
	0.066254	0.056252	0.053750	$-5.7021.10^{-3}$	$-3.6202.10^{-3}$	$-6.9173.10^{-4}$
	0.073752	0.063751	0.046243	$-2.6278.10^{-3}$	$-1.9126.10^{-3}$	$1.5881.10^{-2}$
	0.081336	0.071349	0.038739	$-1.0617.10^{-1}$	$-1.3852.10^{-1}$	$2.6143.10^{-2}$
	0.081259	0.071102	0.061205	$-1.0822.10^{-2}$	$2.0741.10^{-1}$	$7.3227.10^{-2}$
	0.073750	0.063775	0.053756	$-2.7750.10^{-4}$	$-3.9645.10^{-2}$	$-1.1588.10^{-2}$
	0.066252	0.056249	0.046253	$-3.5211.10^{-3}$	$-8.6538.10^{-3}$	$-7.1884.10^{-3}$
	0.058739	0.048704	0.038729	$1.8037.10^{-2}$	$9.5342.10^{-2}$	$5.2121.10^{-2}$
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev %(k_2)	Dev %(k_3)
	0.058727	0.049129	0.061546	$3.8701.10^{-2}$	$-7.7703.10^{-1}$	$-4.8398.10^{-1}$
	0.066253	0.056267	0.053759	$-4.1298.10^{-3}$	$-3.1035.10^{-2}$	$-1.6287.10^{-2}$
	0.073754	0.063773	0.046255	$-4.8732.10^{-3}$	$-3.6619.10^{-2}$	$-1.0104.10^{-2}$
	0.081299	0.071332	0.038753	$-6.0656.10^{-2}$	$-1.1496.10^{-1}$	$-8.6065.10^{-3}$
	0.081259	0.071294	0.061275	$-1.0671.10^{-2}$	$-6.2310.10^{-2}$	$-4.0899.10^{-2}$
	0.073752	0.063789	0.053765	$-3.2353.10^{-3}$	$-6.1994.10^{-2}$	$-2.7847.10^{-2}$
	0.066252	0.056259	0.046255	$-3.0596.10^{-3}$	$-1.7102.10^{-2}$	$-1.0054.10^{-2}$
	0.058745	0.048732	0.038742	$8.2877.10^{-3}$	$3.6864.10^{-2}$	$2.0385.10^{-2}$

2A. *Prediction* of the rate constants from the data corresponding to *Model I* performed with the optimal network trained for *Model I*.

2B	ANN			Deviation		
	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev % k_2)	Dev %(k_3)
Model II	0.089427	-0.15925	-0.073279	-52.217	426.77	219.64
	0.10024	-0.16775	-0.072526	-51.304	398.22	234.93
	0.10707	-0.17533	-0.071168	-45.183	375.03	253.88
	0.11084	-0.18149	-0.069773	-36.421	354.72	280.06
	0.11688	-0.17155	-0.060305	-43.855	340.76	198.46
	0.11065	-0.17813	-0.067629	-50.030	379.41	225.82
	0.098805	-0.17031	-0.069111	-49.139	402.76	249.43
	0.086149	-0.16612	-0.065426	-46.636	440.76	268.84
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev %(k_2)	Dev %(k_3)
	0.059328	0.001136	0.026483	-0.98302	97.670	56.763
	0.064869	-0.002508	0.020803	2.0839	104.46	61.296
	0.069575	-0.007331	0.015470	5.6611	111.50	66.550
	0.073259	-0.014308	0.010470	9.8357	120.08	72.980
	0.077989	-0.003247	0.022999	4.0128	104.56	62.451
	0.070884	-0.004149	0.019745	3.8854	106.51	63.266
	0.063785	-0.005405	0.016441	3.7208	109.61	64.451
	0.056686	-0.007231	0.013086	3.5127	114.8320	66.229

2*B.* *Prediction* of the rate constants from the data corresponding to *Model II* performed with the optimal network trained for *Model I.*

2C	ANN			Deviation		
	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev % k_2)	Dev %(k_3)
Model III	0.093433	-0.082678	-0.027856	-59.034	269.60	145.58
	0.10915	-0.085810	-0.025553	-64.753	252.55	147.54
	0.11904	-0.086761	-0.023484	-61.415	236.10	150.78
	0.12469	-0.093002	-0.024759	-53.474	230.53	163.89
	0.12219	-0.060237	-0.0091367	-50.395	184.54	114.92
	0.11869	-0.074851	-0.017442	-60.930	217.41	132.45
	0.10853	-0.10044	-0.033397	-63.824	278.56	172.21
	0.090642	-0.13174	-0.055338	-54.284	370.25	242.81
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev %(k_2)	Dev %(k_3)
	0.059330	0.000662	0.026478	-0.98797	98.642	56.771
	0.064882	-0.001620	0.020772	2.0656	102.88	61.355
	0.069650	-0.005254	0.015315	5.5591	108.24	66.886
	0.073451	-0.011363	0.010145	9.5989	115.95	73.820
	0.078026	-0.002167	0.022931	3.9679	103.04	62.561
	0.070922	-0.002804	0.019666	3.8343	104.40	63.413
	0.063829	-0.003735	0.016335	3.6531	106.64	64.682
	0.056738	-0.005115	0.012956	3.4241	110.49	66.566

2C. *Prediction* of the rate constants from the data corresponding to *Model III* performed with the optimal network trained for *Model I*.

4.3.2. Discrimination of Model II from Models I and III.

The same way as the previous case from the data containing the *Inputs* matrix for *training* and the *Targets* matrix for *training* generated from *Model II*, the *training* of the Neural Network is carried out, resulting in this case that the optimal architecture has a configuration of three *Hidden* layers and the number of nodes in each layer is 10/10/13.

Subsequently each one of the *Inputs* matrices for *prediction* is submitted to a *prediction* process with the Neural Network exclusively *trained* to the *Model II*, giving in each case the *Outputs* ***[Y]*** (size 59x3), that have the values of the kinetic constants that the ANN predicts for the three models. The results of the *prediction* process performed with the trained ANN for the *Model II* are shown in Table 3:

3A) *Prediction* of the rate constants from the data corresponding to *Model II* performed with the optimal network trained for *Model II*.

3B) *Prediction* of the rate constants from the data corresponding to *Model I* performed with the optimal network trained for *Model II*.

3C) *Prediction* of the rate constants from the data corresponding to *Model III* performed with the optimal network trained for *Model II*.

Only the *Outputs* ***[Y]*** matrix that corresponds to the *prediction* process of *Model II* provides the values of the kinetic constants with which the eight curves have been generated.

The results obtained after the optimization process with the program KINMODEL (AGDC), show that the mechanism that best represents the absorbance kinetic data corresponds to *Model II*.

Table 3

	ANN			**Deviation**		
3A	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	**Dev %(k_1)**	**Dev % k_2)**	**Dev %(k_3)**
Model II	0.058750	0.048746	0.061248	$-5.7661.10^{-4}$	$7.5270.10^{-3}$	$2.9185.10^{-3}$
	0.066250	0.056249	0.053750	$-7.3825.10^{-5}$	$5.3688.10^{-4}$	$-8.7737.10^{-4}$
	0.073749	0.063751	0.046251	$3.3055.10^{-4}$	$-2.1329.10^{-3}$	$-1.7196.10^{-3}$
	0.081250	0.071255	0.038749	$-2.0770.10^{-4}$	$-7.0052.10^{-4}$	$1.4365.10^{-3}$
	0.081249	0.071253	0.061255	$1.4159.10^{-3}$	$-3.5546.10^{-3}$	$-7.5865.10^{-3}$
	0.073750	0.063750	0.053749	$-3.3204.10^{-4}$	$-5.6038.10^{-4}$	$9.5832.10^{-4}$
	0.066249	0.056254	0.046245	$4.0244.10^{-4}$	$-6.8206.10^{-4}$	$1.0641.10^{-2}$
	0.058748	0.048727	0.038748	$3.0051.10^{-3}$	$4.6896.10^{-2}$	$5.8289.10^{-3}$
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	**Dev %(k_1)**	**Dev %(k_2)**	**Dev %(k_3)**
	0.058750	0.048748	0.061249	$-4.8000.10^{-4}$	$4.8903.10^{-3}$	$2.4457.10^{-3}$
	0.066250	0.056249	0.053749	$-3.2151.10^{-4}$	$1.6907.10^{-3}$	$1.8977.10^{-4}$
	0.073750	0.063749	0.046249	$-5.3288.10^{-4}$	$2.0282.10^{-3}$	$2.1557.10^{-3}$
	0.081251	0.071251	0.038749	$-7.1262.10^{-4}$	$-7.7474.10^{-4}$	$3.1458.10^{-3}$
	0.081249	0.071252	0.061254	$1.0646.10^{-3}$	$-2.2540.10^{-3}$	$-5.9102.10^{-3}$
	0.073751	0.063745	0.053746	$-1.6393.10^{-3}$	$7.3663.10^{-3}$	$7.0847.10^{-3}$
	0.066250	0.056251	0.046247	$-2.9736.10^{-4}$	$-9.9378.10^{-4}$	$7.3492.10^{-3}$
	0.058749	0.048738	0.038749	$1.3396.10^{-3}$	$2.4441.10^{-2}$	$2.5884.10^{-3}$

3A. *Prediction* of the rate constants from the data corresponding to *Model II* performed with the optimal network trained for *Model II*.

	ANN			**Deviation**		
3B	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	**Dev %(k_1)**	**Dev % k_2)**	**Dev %(k_3)**
Model I	0.058532	0.030347	0.0036363	0.37049	37.750	94.063
	0.063845	0.018147	-0.0017041	3.6309	67.739	103.17
	0.067102	0.0067477	-0.0092028	9.0139	89.415	119.90
	0.068089	-0.0042771	-0.018818	16.198	106.00	148.56
	0.074876	0.018342	-0.0013348	7.8454	74.257	102.18
	0.068746	0.014279	-0.0039868	6.7857	77.602	107.42
	0.062875	0.010582	-0.0062787	5.0950	81.188	113.58

	0.057543	0.0073420	-0.0081421	2.0537	84.939	121.01
	AGDC			**Deviation**		
	k_1/min^{-1}	**k_2/min^{-1}**	**k_3/ min^{-1}**	**Dev %(k_1)**	**Dev %(k_2)**	**Dev %(k_3)**
	-	-	-	-	-	-
	0.077173	-0.068946	-0.017217	-16.488	222.57	132.03
	0.087471	-0.079599	-0.019675	-18.605	224.86	142.54
	-	-	-	-	-	-
	0.097483	-0.072869	-0.013141	-19.979	202.27	121.45
	0.087330	-0.073128	-0.016224	-18.414	214.71	130.18
	0.077299	-0.075787	-0.020976	-16.679	234.73	145.35
	-	-	-	-	-	-

3*B.* *Prediction* of the rate constants from the data corresponding to *Model I* performed with the optimal network trained for *Model II*.

	ANN			**Deviation**		
3C	**k_1/min^{-1}**	**k_2/ min^{-1}**	**k_3/ min^{-1}**	**Dev %(k_1)**	**Dev % k_2)**	**Dev %(k_3)**
Model III	0.049615	0.081397	0.20273	15.549	-66.968	-230.99
	0.054558	0.088577	0.20041	17.648	-57.470	-272.86
	0.060618	0.094868	0.19206	17.806	-48.813	-315.26
	0.068455	0.097727	0.17023	15.748	-37.161	-339.30
	0.065028	0.10060	0.20697	19.966	-41.197	-237.91
	0.059957	0.095843	0.20242	18.702	-50.342	-276.59
	0.055063	0.085858	0.18869	16.886	-52.636	-307.99
	0.051452	0.065286	0.15579	12.421	-33.920	-302.03
	AGDC			**Deviation**		
	k_1/min^{-1}	**k_2/min^{-1}**	**k_3/ min^{-1}**	**Dev %(k_1)**	**Dev %(k_2)**	**Dev %(k_3)**
	0.061224	0.037857	0.021022	-4.2109	22.344	65.678
	0.068966	0.045285	0.018975	-4.1003	19.494	64.698
	0.076563	0.053156	0.016763	-3.8144	16.618	63.755
	0.084043	0.061369	0.014379	-3.4370	13.868	62.892
	0.085616	0.055466	0.020851	-5.3739	22.153	65.957
	0.077036	0.051311	0.018888	-4.4560	19.513	64.860
	0.068575	0.046943	0.016873	-3.5087	16.546	63.519
	0.060288	0.042165	0.014871	-2.6174	13.507	61.623

3C. *Prediction* of the rate constants from the data corresponding to *Model III* performed with the optimal network trained for *Model II*.

4.3.3. Discrimination of Model III from Models I and II.

Proceeding in the same way as the previous cases, from data that forms the *Inputs* and the *Targets* matrices for *training* generated for *Model III*, the *training* of ANN is performed, in this case the optimal architecture has a configuration of three *Hidden* layers and the number of nodes in each layer is 10/10/13.

Next each one of the *Inputs* matrices for *prediction* corresponding to *Models I, II* and *III*, it is submitted to a *prediction* process with the Neural Network exclusively trained to the *Model III*, giving in each case the *Outputs* ***[Y]*** (dimensions 59x3). These matrices hold the values of the kinetic constants which the Neural Network predicts for each model, corresponding to each one of the processes of individual *prediction* (Table 4). The *Outputs* ***[Y]*** matrix that corresponds to the *prediction* process of *Model III* gives the correct values of the kinetic constants.

4A) *Prediction* of the rate constants from the data corresponding to *Model III* performed with the optimal network trained for *Model III*.

4B) *Prediction* of the rate constants from the data corresponding to *Model I* performed with the optimal network trained for *Model III*.

4C) *Prediction* of the rate constants from the data corresponding to *Model II* performed with the optimal network trained for *Model III*.

Lastly, the treatment of kinetic data with the program KINMODEL(AGDC) is carried out and after performing an analogous process to the previous cases we conclude that the mechanism that better represents the kinetic data is *Model III*.

Table 4

	ANN			**Deviation**		
4A	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	**Dev %(k_1)**	**Dev % k_2)**	**Dev %(k_3)**
Model III	0.058749	0.048749	0.061246	$7.9226.10^{-4}$	$7.2854.10^{-4}$	$6.3928.10^{-3}$
	0.066250	0.056249	0.053751	$-2.0241.10^{-1}$	$3.9991.10^{-5}$	$-1.2650.10^{-3}$
	0.073750	0.063749	0.046249	$-7.6214.10^{-5}$	$2.0942.10^{-4}$	$3.5496.10^{-4}$
	0.081255	0.071247	0.038732	$-5.9079.10^{-3}$	$4.1696.10^{-3}$	$4.7718.10^{-2}$
	0.081252	0.071253	0.061252	$-2.3194.10^{-3}$	$-5.3504.10^{-3}$	$-3.9707.10^{-3}$
	0.073750	0.063749	0.053750	$-2.7557.10^{-4}$	$4.1710.10^{-4}$	$-2.5992.10^{-4}$
	0.066250	0.056250	0.046249	$-2.3049.10^{-4}$	$-7.6780.10^{-4}$	$8.1915.10^{-4}$
	0.058749	0.048749	0.038752	$7.9246.10^{-4}$	$2.0235.10^{-3}$	$-4.5959.10^{-3}$
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	**Dev %(k_1)**	**Dev %(k_2)**	**Dev %(k_3)**
	0.058748	0.048754	0.061248	$2.8800.10^{-3}$	$-7.9097.10^{-3}$	$3.7894.10^{-3}$
	0.066247	0.056258	0.053750	$3.9109.10^{-3}$	$-1.3851.10^{-2}$	$-4.2791.10^{-5}$
	0.073747	0.063758	0.046250	$4.3485.10^{-3}$	$-1.3062.10^{-2}$	$-8.4324.10^{-5}$
	0.081242	0.071269	0.038750	$1.0055.10^{-2}$	$-2.6187.10^{-2}$	$-9.6516.10^{-4}$
	0.081233	0.071297	0.061248	$2.1420.10^{-2}$	$-6.5593.10^{-2}$	$3.5233.10^{-3}$
	0.073746	0.063759	0.053749	$4.9912.10^{-3}$	$-1.5187.10^{-2}$	$3.3860.10^{-4}$
	0.066247	0.056257	0.046250	$4.4362.10^{-3}$	$-1.2668.10^{-2}$	$-1.3622.10^{-4}$
	0.058749	0.048749	0.038751	$1.0757.10^{-3}$	$5.1282.10^{-4}$	$-3.7032.10^{-3}$

4A. *Prediction* of the rate constants from the data corresponding to *Model III* performed with the optimal network trained for *Model III*.

4B	ANN			Deviation		
	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev % k_2)	Dev %(k_3)
Model I	0.058532	0.030347	0.0036363	0.37049	37.750	94.063
	0.063845	0.018147	-0.0017041	3.6309	67.739	103.17
	0.067102	0.0067477	-0.0092028	9.0139	89.415	119.90
	0.068089	-0.0042771	-0.018818	16.198	106.00	148.56
	0.074876	0.018342	-0.0013348	7.8454	74.257	102.18
	0.068746	0.014279	-0.0039868	6.7857	77.602	107.42
	0.062875	0.010582	-0.0062787	5.0950	81.188	113.58
	0.058532	0.030347	0.0036363	0.37049	37.750	94.063
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev %(k_2)	Dev %(k_3)
	-	-	-	-	-	-
	-	-	-	-	-	-
	0.090043	-0.052468	-0.0088579	-22.092	182.30	119.15
	0.10026	-0.062587	-0.0098174	-23.400	187.84	125.34
	-	-	-	-	-	-
	0.90517	-0.047351	-0.0076605	-1127.4	174.28	114.25
	0.079439	-0.048575	-0.0095451	-19.908	186.36	120.64
	0.068685	-0.050689	-0.012203	-16.911	203.98	131.49

4B. *Prediction* of the rate constants from the data corresponding to *Model I* performed with the optimal network trained for *Model III.*

4C	ANN			Deviation		
	k_1/min^{-1}	k_2/ min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev % k_2)	Dev %(k_3)
Model II	0.058552	0.046232	0.021497	0.33689	5.1649	64.902
	0.066059	0.053707	0.019136	0.28820	4.5215	64.399
	0.073704	0.061493	0.016664	0.062049	3.5402	63.969
	0.081586	0.069527	0.013964	-0.41298	2.4187	63.964
	0.081465	0.067412	0.020655	-0.26417	5.3866	66.278
	0.073696	0.060922	0.018846	0.073008	4.4365	64.937
	0.066059	0.054141	0.016959	0.28698	3.7498	63.332
	0.058642	0.047189	0.014808	0.18318	3.2000	61.786
	AGDC			**Deviation**		
	k_1/min^{-1}	k_2/min^{-1}	k_3/ min^{-1}	Dev %(k_1)	Dev %(k_2)	Dev %(k_3)
	0.068553	0.031193	0.18401	-16.686	36.015	-200.43
	0.075719	0.044353	0.16349	-14.293	21.150	-204.18
	0.081543	0.060777	0.14852	-10.567	4.6639	-221.13
	0.086195	0.076649	0.12672	-6.0857	-7.5787	-227.02
	0.095526	0.053636	0.18593	-17.571	24.721	-203.57
	0.084089	0.053848	0.16955	-14.018	15.533	-215.44
	0.072983	0.053367	0.14975	-10.164	5.1252	-223.79
	0.061976	0.052649	0.12823	-5.4902	-7.9996	-230.92

4C. *Prediction* of the rate constants from the data corresponding to *Model II* performed with the optimal network trained for *Model III*.

5. Conclusions

- The three kinetic models analyzed are different in structure (lineal or cyclic) or in the distribution of the individual reactions or kinetic constants, (3 species and 3 kinetic constants) but in all of them we were able to carry out the treatment with the same ED obtaining excellent results.

- The Hybrid Algorithm (ANN-AGDC) applied in the first place, the *soft-modelling* ANN methodology in which it is not necessary to have initial estimates of the rate constants. The HA's second methods is the mathematical optimization algorithm AGDC in which initial estimates of the rate constants are necessary. The HA used the values obtained for the rate constants after the application of ANN, as well as initial estimates for the application of the algorithm AGDC.

- The HA has the ability to discriminate or distinguish between several models or reaction mechanisms, although very similar, that could be responsible for the progress of the chemical reaction. The *prediction* process leads to the determination of the right values of the individual rate constants in each stage, only when the treatment of the kinetic data of a given model is performed by the ANN trained for this model. Otherwise, the *prediction* process fails, obtaining absurd values and unacceptable. This fact allows us to decide which one of all the models could be responsible for the course of the chemical reaction to study, which means a great advantage in the field of *Modelling* in Chemical Kinetics.

References

[1] F. Pérez Pla, J. F. Baeza Redón, R.Valero, A new algorithm for the kinetic data analysis, *Chemom. Intell. Lab. Syst.* **53** (2000) 1–19.

[2] S. Bijlsma, H. F. Boelens, H. C. Hoefsloot, A. K. Smilde, Constrained least squares methods for estimating reaction rate constants from spectroscopic data, *J. Chemom.* **16** (2002) 28–40.

[3] B. Svir, O. V. Klymenco, M. S. Platz, KITFITSIM, a software to fit kinetic data to a user selected mechanism, *Comput. & Chem.* **26** (2002) 379–386.

[4] J. L. González, M. M. Canedo, C. Grande, Classic and multivariate modeling treatment of the kinetics and mechanism of isomerization of 5-cholesten-3-one catalyzed by sodium ethoxide, *Int. J. Chem. Kinet.* **38** (2006) 38–47.

[5] M. M. Canedo, J. L. González–Hernández, KINMODEL(AGDC): a multipurpose computational method for kinetic treatment, *J. Math. Chem.* **49** (2011) 163–184.

[6] M. M. Canedo, J. L. González–Hernández, S. Encinar del Dedo, Application de computational method KINMODEL(AGDC) to the simultaneous determination of kinetic and analitycal parameters, *App. Math. Comp.* **219** (2013) 7089–7101.

[7] J. L. González–Hernández, M. M. Canedo, C. Grande, Combining different mathematical optimization methods: a new "hard-modelling" approach for chemical kinetic, *MATCH Commun. Math. Comput. Chem.* **70** (2013) 951–970.

[8] N. C. Imlinger, C. Blattner, M. Krell, M. R. Buchmeiser, Hard-modeling of reaction kinetics by combining online spectroscopy and calorimetry, *J. Chemom.* **22** (2008) 758–767.

[9] J. Diewok, A. de Juan, M. Maeder, R. Tauler, B. Lendl, Application of a combination of hard and soft modeling for equilibrium systems to the quantitative analysis of pH-modulated mixture samples, *Anal. Chem.* **75** (2003) 641–647.

[10] B. Kovacs, J. Tóth, Estimating reaction rates constants with neural networks, *Int, J. Appl. Math. Comput. Sci.* **4** (2007) 7–11.

[11] N. H. T. Lemes, E. Borges, J. P. Braga, Rate constants and absorption coefficients from experimental data: an inversion procedure based on recursive neural networks, *Chemom. Intell. Lab. Syst.* **96** (2009) 84–87.

[12] F. Amato, J. L. González Hernández, J. Havel, Artificial neural networks combined with experimental design: A "soft" approach for chemical kinetics. *Talanta* **93** (2012) 72–78.

[13] M. M. Canedo, J. L. González–Hernández, S. Encinar del Dedo, Combining artificial neural networks and experimental design to prediction of kinetic rate constants, *J. Math. Chem.* **51** (2013) 1634–1653.

[14] MATLAB & SIMULTLINK, © COPYRIGHT by MathWorks, 1984–2016.

[15] M. Maeder, Y. M. Neuhold, G. Puxty, P. Gemperline, Advances in the modelling and analysis of complex and industrial processes, *Chemom. Intell. Lab. Syst.* **82** (2006) 75–82.

[16] E. J. Molga, B. A. A. Woezik, K. R. Westerterp, Neural networks for modelling of chemical reaction systems with kinetics: oxidation of 2-octanol with nitric acid, *Chem. Engin. Proces.* **39** (2000) 323–334.

[17] Y. Ni, D. Cao, S. Kokot, Simultaneous enzymatic kinetic determination of pesticides, carbaryl and phoxim, with the aid of chemometrics, *Anal. Chim. Acta* **588** (2007) 131–139.

[18] S. Encinar del Dedo, J. L. González–Hernández, M. M. Canedo, A new computational treatment in non-isothermal chemical kinetics by application of a robust hybrid algorithm, *MATCH Commun. Math.Comput. Chem.* **72** (2014) 427–450.

[19] S. Encinar del Dedo, J. L. González–Hernández, M. M. Canedo, D. Juanes, A robust hybrid algorithm (Neural Networks AGDC) applied to non-isothermal kinetics of consecutive chemical reactions, *J. Math. Chem.* **53** (2015) 1080–1104.

[20] S. Encinar del Dedo, Tesis Doctoral, Departamento de Química Física, Universidad de Salamanca, 2015.

[21] K. J. Laidler, Symbolism and terminology in chemical kinetics, *Pure Appl. Chem.* **53** (1981) 753–771.

[22] K. J. Laidler, A glossary of terms used in chemical kinetics, including reaction dynamics, *Pure Appl. Chem.* **68** (1996) 149–192.

[23] T. Kohonen, An introduction to neural computing, *Neural Networks* **1** (1988) 3–16.

[24] C. F. Gerald, P. O. Wheatley, *Applied Numerical Analysis*, Adinson-Wesley, Reading, 1984.

[25] M. A. Wolfe, *Numerical Methods for Unconstrained Optimization*, Van Nostrand, Berkshire, 1978.

[26] P. Gill, W. Murray, M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.