MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **72** (2014) 427-450

ISSN 0340 - 6253

A New Computational Treatment in Non-Isothermal Chemical Kinetics by Application of a Robust Hybrid Algorithm

S. Encinar, J. L. González-Hernández*, M. Mar Canedo

Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca, E-37008 Salamanca, Spain

(Received January 13, 2014)

Abstract.

In the present work we designed and successfully applied a new Hybrid Algorithm (HA) able to optimize Activation Thermodynamic Parameters (ATP) directly from the computational treatment of non-isothermal kinetic data. It comprises two methods based in different mathematical principles whose sequential application is carried out. Firstly it uses a *"soft modeling"* method of Artificial Neural Networks (ANN) that does not need to use the *initial estimates* of the parameters and determine the ATPs values in the neighborhood of the minimum global. These values are improved later after application of a mathematical optimization method based in a second order gradient algorithm (AGDC) able to reach the desired global optimum.

The application of HA that we propose provided good and satisfactory results for the optimized values of the ATPs. HA offers several advantages with respect to the isothermal classic methods since it allows to optimize ATPs directly without the need to determine the kinetic constants previously. In addition for the case of non-isothermal experiments, a single replicated kinetic experiment is enough since it allows the computation of a larger number of kinetic data that implies a considerable saving of reagents and laboratory time.

(*) *Author for the correspondence*
e-mail: jlgh93@usal.es

1. Introduction

Determination of the parameters involved in the mathematical functions by means of the application of computational treatments is usually carried out by means of mathematical optimization methods applying gradient algorithms. These non-linear iterative fitting techniques are often used in the various fields of Chemistry, in particular, and of Science in general. However, a major drawback of these optimization methods is their high sensitivity to the *initial estimates* of the parameters to be supplied. Only if these values are very close to the global minimum can a rapid and reliable convergence of the iterative process be expected, guaranteeing success of the parameter optimization. If the *initial estimates* are far from the global minimum the process may become divergent or may reach a singular point (*local minimum*, *saddle point*, etc…), leading the optimization process to fail. This often happens in the treatment of kinetic models. Additionally there is a series of problems involving *identifiability* and *distinguishability*, which lead to different types of ambiguity in the solutions to the *stiff* systems of ordinary differential equations (ODE). In light of this, it would seem appropriate to design and apply a new method that, initially, would provide an approach to the global optimum and then use such results as a starting point to apply a robust gradient method that will guarantee the success of the mathematical optimization of parameters.

In the present work we designed and successfully applied a new Hybrid Algorithm (HA) able to optimize Activation Thermodynamic Parameters (ATP) -Pre-exponential factor (A) and the Activation Energy (Ea)- directly from non-isothermal kinetic data. It comprises two methods based in different mathematical principles whose sequential application is carried out. Firstly we applied a *"soft modeling"* method using Artificial Neural Networks (ANN) that does not need to use the *initial estimates* of the parameters and its goal is the determination of the parameter values (*outputs*) in the neighborhood of the optimum global. These values will be the *initial estimates* of a robust and efficient second order gradient algorithm (AGDC) [1] able to reach the desired global optimum that will guarantee the success of the final optimization of the values of the parameters. The classic method used to determine ATPs consists of applying *"hard-modelling"* techniques to series of data of isothermal kinetic experiments acquired at different temperatures to initially obtain the rate constants of each kinetic experiment. With the values of these, the ATPs are obtained in a second step from the linearized Arrhenius equation.

The application of this HA that we propose, provided several advantages since it became possible to optimize those parameters directly, without the need to determine the

kinetic constants in a previous step. To be able to apply this treatment, it was necessary to acquire the kinetic data from non-isothermal kinetic experiments [2], imposing a controlled variation of temperature along the reaction kinetic. The number of the set of kinetic data computed in the classic isothermal procedure (T=constant) is very small since it is limited by the experimental requirements when dealing with a low number of isothermal experiments (normally, 10-12), each of them performed at a different temperature. In the case of non-isothermal experiments, a single replicate kinetic experiment is enough since it allows the computation of a huge set of kinetic data which, bearing in mind the laboratory time and reagents saved, is a great advantage.

The literature contains several recent references in which the authors have applied Hybrid Algorithms (HA) in the field of Chemistry and that essentially comprise Genetic Algorithms (GA). Their design and application are currently able to solve problems pending solution and reveal the high degree of reliability and precision in the results obtained. Of interest is one review [3] in which the authors consider a broad range of applications of different types of GA in Chemometrics: M. Maeder *et col*. [4] determine the rate and equilibrium constants of reaction mechanisms by application of a Hybrid Algorithm based on a Genetic Algorithm. C. Hervás *et col.* [5] use GA and pruning computational neural networks for selecting the number of inputs required to correct temperature variations in kinetic-based determinations. Artificial Neural Networks (ANN) offer a versatile *"soft modeling"* method that can be applied in diverse fields with acceptable results [6]**.** The method is applied for quantitative purposes, among others, in the so-called *Principal Component Analysis (PCA)* method, in which there are no explicit functions of multivariate correlation or, if there are, they are extraordinarily complex. The results of applying ANN for quantitative purposes in chemical kinetics in simple models [7,8] are in some cases acceptable. However, it is necessary to perform an exhaustive process of *training* of the neural network in order to obtain its optimal architectures, meaning that the method is time-consuming and tedious and that it cannot be used individually in all cases. Nevertheless, it is an ideal method for carrying out the approximation of the ATPs values to those of the global minimum, without previously knowing anything about the magnitude and sign of the parameters. The AGDC algorithm was designed and successfully implemented at our laboratory in the treatment of many fields within the area of Physical Chemistry [9-13]**.** It is a second-order gradient algorithm in which continuous control of the direction and sense of the movement vector is carried out, evaluating each step performed in the iterative process. It is endowed with corrective devices that, when necessary, are able to correct and modify the movement vector automatically in

order to ensure the decrease to reach the minimum global. Additionally, it is refractive to the presence of local minima and *ill-conditioned* response surfaces and includes different graphic representation methods (3D, contour maps, etc.) that allow both the graphic follow-up of the global minimum and a detailed exploration of the region of the global minimum to detect the existence of singular points.

2. Theoretical and Computational Aspects

2.1. Chemical Kinetic Aspects

Let us consider in general a chemical system formed by n_r chemical elementary (or concerted) reactions where n_s chemical species can be involved. According to IUPAC's norms [14] the r-*th* chemical reaction can be expressed for the generic equation

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

where, B_j is the chemical species involved in the system of reactions; r=(1,…, n_r), the number of chemical reactions; j=(1,…,n_s), the number of chemical species; $\nu_{j,r}$, the stoichiometric coefficient of the species B_j in the r-*th* reaction; $\nu_{j,r} < 0$ when B_j plays only the role of reactant in the r-*th* reaction and $\nu_{j,r} > 0$ when B_j plays only the role of product in the r-*th* reaction.

When the reaction is an elementary or concerted one, the absolute values of the kinetic order ($z_{l,r}$) and stoichiometric coefficient of B_j coincide, that is $|\nu_{l,r}| = |z_{l,r}|$. The rate differential equation of the chemical species B_j in the r-*th* is given by

$$d\left[B_j\right]/dt = k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \qquad (2)$$

where B_l are the species playing only the role of reactants in the r-th reaction ($\nu_{l,r} < 0$) and k_r is the kinetic rate constant of the r-*th* reaction. Each chemical species can take part in several reactions and the rate differential equations will be the sum extended over those reactions where the reactant B_l appears, obtaining a system of ordinary differential equations (ODE) according to the generic equation,

$$d\left[B_j\right]/dt = \sum_{r=1}^{n_r} k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \qquad (3)$$

The general solution of the system of rate ODE give the explicit function of the concentrations of the all species with the time ($[B_j]_{t_i}$). If the experimental data are expressed in absorbance, we have to consider the Lambert-Beer-Bouguer law:

$$A^{\lambda}_{j,t_i} = \varepsilon^{\lambda}_{j} \cdot [B_j]_{t_i} \qquad (4)$$

where A^{λ}_{j,t_i} is the absorbance of the species B_j at the, time t_i and path length 1 cm and $\varepsilon^{\lambda}_{j}$ is the molar absorption coefficient of Bj at the wavelength λ. The absorbance of the mixture (A^{λ}_{T,t_i}) measured at wavelength λ, time t_i and temperature T and, can be expressed as:

$$A^{\lambda}_{T,t_i} = \sum_{j=1}^{n_s} A^{\lambda}_{j,t_i} = \sum_{j=1}^{n_s} \varepsilon^{\lambda}_{j} \cdot [B_j]_{t_i} \qquad (5)$$

2.2. Thermodynamic Aspects in Non-Isothermal Kinetics

Let us consider the simple chemical reaction:

$$\nu_1 B_1 \rightarrow \text{Products} \qquad (6)$$

The rate differential equation (3) corresponding to this reaction, expressed in function of the *extent of reaction* variable in units of molar concentration (ξ') when the reactant B_1 is of kinetic first order, can be written as:

$$d\xi'/dt = k(T)([B_1]_0 - |\nu_1|\xi') \qquad (7)$$

Separating variables and later integration of the first member of the equation, we have

$$-(1/|\nu_1|)\ln\{([B_1]_0 - |\nu_1|\xi')/[B_1]_0)\} = \int_0^t k(T)dt \qquad (8)$$

The Activation Energy, (Ea) and the Pre-exponential Factor (A) can be determined substituting *k(T)* in (8) according the Arrhenius equation, where α is the remaining molar fraction of the reactant B_1.

$$-(1/|\nu_1|) \ln \alpha = \int_0^t Ae^{-Ea/RT}dt \qquad (9)$$

In non-isothermal conditions, the second member of (9) cannot be directly integrated since there are two dependent variables (T and t). It will be crucial to establish the identity of this function because the mathematical method of resolution of the equation (9) will be different. The function must be monotonically increasing, since is this way it will be possible to minimize the great differences in the reaction rate existing between the beginning and the end of the reaction; the rate of heating must be suitable for the interval of time studied and its profile must be reproducible in the laboratory.

We consider several possibilities for the function T = f(t)

1.- Inverse function of the temperature corresponding to a hyperbolic branch

$$1/T = (1/T_0) - mt \qquad (10)$$

Substituted in (9), we have

$$-(1/|\nu_1|) \ln \alpha = \int_0^t Ae^{-Ea/R\,[(1/T_0) - m\,t]}dt \qquad (11)$$

This function has a great advantage since the integral of (9) has a known primitive function and therefore, mathematical exact solution with the following expression:

$$\alpha = e^{-[(|\nu_1| A R/m\,Ea)(e^{-Ea/RT_0})(e^{(m\,Ea/R)\,t}-1)]} \qquad (12)$$

We have to take in account the following important considerations:

a) the explicit function of α depending on the time (12) is not simple and there is necessary the application of a method of sufficiently robust treatment for the determination of the ATPs. In addition it is necessary to consider the great difference in the order of magnitude of both parameters that complicates extraordinarily the success in the application of the method of treatment.

b) the experimental points (T/t) must belong to this inverse hyperbolic function and therefore, it is necessary to reproduce the profile of the curve of the function (T/t) (10) with the points obtained in the laboratory.

2.- Function T = f(t) of polynomial type of n-*th* degree

$$T = \sum_{j=0}^{j=n} a_j\, t^j \qquad (13)$$

substituting in (8) according the Arrhenius equation,

$$-(1/|\nu_1|)\ln\alpha = \int_0^t A e^{-Ea/R\,[(1/\sum_{j=0}^{j=n} a_j t^j)]} dt \qquad (14)$$

This equation does not have mathematical exact solution since the integration cannot be performed. We have 2 options of treatment:

2.1- Numerical integration of the equation (14) using appropriate "*quadrature formulas"* corresponding to a suitable numerical algorithms of resolution of integrals (Simpson, Lobatto, Gauss-Kronrod, Vectorized, etc.)

2.2- Numerical resolution directly from the beginning of the ordinary differential equation (eq. [7]) expressed in terms of Arrhenius's equation, that is:

$$d[B_1] = -[B_1] A\, e^{-Ea/RT} dt \qquad (15)$$

The solution of this differential equation (15) can be performed by means of the application of numerical suitable methods that they must be suitable for the treatment of *"stiff"* systems (Runge-Kutta methods, Gear's method (BDFs), Rosenbrock formula, trapezoidal rule with "free" interpolant….etc).

2.3. Mathematical Aspects. Hybrid Algorithm (HA)

The Hybrid Algorithm (HA) designed in our laboratory comprises two methods based in different mathematical principles and it is applied sequentially in two steps. In the first, a method of

Artificial Neural Network (ANN) is applied. Its most important feature, and indeed a great advantage, is that it is not necessary to supply *initial estimates* values for the determination of ATPs (A and Ea). The results obtained in the output matrix are near to those that belong to the global optimum sought. These values are precisely the *initial estimates* for starting the process of ATPs optimization via application of the AGDC algorithm, constituting the second step in the treatment. This is an improvement of the values obtained in the output matrix after the application of the ANN, thus reaching the global minimum that will guarantee the determination of the optimized values of the ATPs.

2.3.1. Artificial Neural Networks (ANN)

Artificial Neural Networks are parallel interconnected networks of simple computational elements called neurons and are structured in layers that are intended to interact with the objects of the real world in a similar way to the biological nervous systems [15]. Parallel processing is the ability of the brain to simultaneously process incoming stimuli of differing quality. The multilayer neural network uses sets of input data and parameters (called *targets)*, distributed in 2 input matrices when Matlab [16] is applied. The elements of the input matrix are the calculated synthetic values, where one row contains a single curve of the data 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 $(\alpha_j, A_T, A_j, [B_j])$ 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 first layer of a multilayer neural network contains neurons that 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 layer following the input one is called *hidden.* In each neuron of a *hidden* 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**.**

The process described is called to as the *training* or *learning* of the multilayer neural network and constitutes an iterative method where the iterations are called *epochs*. After each *epoch,* 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}) and the optimum value of the Mean Squared Error (MSE), expressed in absolute value, is calculated according the following equation:

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{output} - b_{ij}^{target})^2}{n_p . n_c} \right)^{½} \qquad (16)$$

where n_c is the input number of curves and n_p is the number of parameters, $n_c x n_p$ being the dimensions of both matrices (*output matrix* and *target matrix*).

During the process of *training, weights* and *bias* values are modified with 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.

It is necessary to know the optimal architecture and topology of the multilayer neural network in order 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 configurations, the value of the MSE decreases rapidly when the number of neurons increases, but after a constant value or a poor improvement is obtained. The optimum number of neurons *(configuration)* 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.

Neural network *training* is completed with the processes of *validation* and *testing*. 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 not been any *"overfitting"*, since any possible *"overtraining"* has taken been 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 for terminating the process is performed by controlling the computation time instead of the number of *epochs*.

The process of *prediction* consists of 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 *(α_j, A_T, A_j, $[B_i]$, etc.)*, acquired from a real system of reactions developed at the laboratory.

The computational work of ANN has been performed using Matlab environment [16] by means of the application of *"Neural Networks Toolbox"* with the creation of user´s interfaces (GUI) including the appropriate individual analysis of Residuals and errors (MSE, SD, etc) and special plotting programs for visualizations 3D of SQD functions.

2.3.2. AGDC Algorithm

Mathematical optimization is a computational method that allows the determination of different parameters by means of the maximization or minimization of the objective function. Usually performed objective function minimization using different methods: Search algorithms (Simplex, Fibonacci,..etc) or Gradient algorithms (Steepest Descent, Gauss-Newton, Levenberg-Marquardt, AGDC,…etc) [17]. In this paper we have used the AGDC mathematical optimization algorithm [13,18,19], this is a second-order gradient method that minimizes the numerical function Sum of Quadratic Deviations (SQD) and whose *initial estimates* are the *outputs* values obtained by ANN.

The procedure followed by AGDC, step by step, is the following:

1. Select the parameters to be optimized **X** (Ea/A)

2. m=0 (m= iteration number). Input data:

2.1. Experimental data of $(\alpha_i)_{exp}/t$, initial concentrations, Convergence Criterion (CC), etc.

2.2. *Initial estimates* of the unknown parameters $\mathbf{X}^{(0)}$ (Values of *outputs* from ANN).

3. Determinate the $SQD^{(0)}$ function.

3.1. Calculate of concentrations $[B_1]_i$ by ***a)*** Mathematical exact solution [eq. (12)]; ***b)*** Numerical integration [eq.(14)] ***c)*** ODE [eq. (15)].

3.2. Calculate $(\alpha_i)_{calc}$ $\quad (\alpha_i)_{calc} = [B_1]_i/[B_1]_0$

3.3. Calculate $SQD^{(0)}$ by the equation: $SQD(\mathbf{X}) = \sum_{i=1}^{N_d} \left((\alpha_i)_{calc} - (\alpha_i)_{exp} \right)^2$

4. AGDC ALGORITHM

4.1. Calculate the vector of movement $\mathbf{p}^{(m)}$.

4.1.1. Compute partial numerical derivatives of $(\alpha_i)_{calc}$ with respect to the parameters to be determined $\mathbf{X}^{(m)}$, $\left(\partial(\alpha_i)/\partial X_p\right)$

4.1.2. Compute Gradient vector and Hessian Matrix ($\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$).

4.1.3. Compute $(\mathbf{H}^{(m)})^{-1}$

4.1.4. Calculate the components of the vector of movement

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

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

4.2.1. If $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$, go to 4.3.

4.2.2. If $\mathbf{p}^{(m)}\,\mathbf{g}^{(m)} < \varepsilon$ (ε = scalar close to zero), $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ and go to 4.3

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

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

4.3.1. Compute the scalar ($\lambda^{(m)}$) by the method of Hartley [17]

4.3.2. $\boldsymbol{X}^{(m+1)} = \boldsymbol{X}^{(m)} + \lambda^{(m)}\boldsymbol{p}^{(m)}$

4.3.3. Determinate the $SQD^{(m+1)}$ function.

4.3.4. If the Goldstein-Armijo criterium [20] is satisfied go to 4.4.

4.3.5. $\lambda^{(m)} = \lambda^{(m)}/2$ go to 4.3.2.

4.4. Calculate

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

4.5. If convergence is not attained (CON > CC), set m = m + 1 and go to 4.1.

5. Final of Optimization. $\mathbf{X}^{(m+1)}$ = Optimized Parameters.

5.1 Calculation of the errors of the parameters.

5.2 Statistical analysis of residuals: arithmetic mean, variance, standard deviation, square error, statistical measure of adjustment and Pearson function (χ^2) [21].

6. End of AGDC

7.- End of HA.

The elements constituting the Hessian matrices and gradient vectors are detailed in the following expressions:

$$\mathbf{g} = 2\begin{bmatrix} \sum_{i=1}^{N_d} RES_i \frac{\partial\, \alpha_i}{\partial E_a} \\ \sum_{i=1}^{N_d} RES_i \frac{\partial\, \alpha_i}{\partial A} \end{bmatrix} \quad \text{where,} \quad RES_i = \left[(\alpha_i)_{calc} - (\alpha_i)_{exp}\right]$$

$$\mathbf{H} = 2\begin{bmatrix} \sum_{i=1}^{N_d} \left(\frac{\partial\, \alpha_i}{\partial E_a}\right)^2 & \sum_{i=1}^{N_d} \left(\frac{\partial\, \alpha_i}{\partial E_a}\right)\left(\frac{\partial\, \alpha_i}{\partial A}\right) \\ \sum_{i=1}^{N_d} \left(\frac{\partial\, \alpha_i}{\partial A}\right)\left(\frac{\partial\, \alpha_i}{\partial E_a}\right) & \sum_{i=1}^{N_d} \left(\frac{\partial\, \alpha_i}{\partial A}\right)^2 \end{bmatrix}$$

The computational program has been designed and performed in our laboratory using a computational executable code programs (##.m type), in the Matlab environment [16], using "m" language and it is constituted by a Main program and several Functions and/or Subroutines.

3. Kinetic Data. Input Curve Base

3.1. Study of the Functions T/t and α/t

The function T/t has to fulfill the following requirements: *a)* the function must be monotonically increasing, since is this way it will be possible to minimize the great differences in the reaction rate existing between the beginning and the end of the reaction when isothermal conditions are considered, *b)* the rate of heating must be suitable for reaching a convenient extent of reaction in the interval of time in which the non-isothermal

kinetic data will be acquired, and *c)* the function must be accessible to experimentation and accurately reproducible in the laboratory. The type of functions that we have chosen are:

1.- inverse function of the temperature corresponding to a hyperbolic branch [eq. (10)] that permits to reach an exact mathematical solution of the rate differential equation;

2.- function of polynomial type of n-*th* degree according equation (13), without of exact mathematical solution for the rate differential equation.

It is seen that as the value of *"m"* of equation (10) decreases a smoothing of the concavity of the hyperbolic branch is observed, tending towards linearity. The suitable value, and the one that best satisfies the 3 requisites considered above, is 4.2 10^{-6} K^{-1} min^{-1}. We fitted the T/t points to a first-degree polynomial function and then progressively to polynomials from 2nd to 5th degree, observing that the coefficients of the independent variable had negligible values with respect to the value of a_1 from the quadratic term. The coefficients $(a_0, a_1, ...a_5)$ of the polynomials of 1st, 3rd and 5th degree obtained in the fittings are shown in Table 1.

Polynomial grade	a_0	a_1	a_2	a_3	a_4	a_5
1	298.15	0.3734	-	-	-	-
3	298.15	0.3734	5.0E-4	6.0E-7	-	-
5	298.15	0.3734	5.0E-4	6.0E-7	7.0E-10	1.0E-12

Table 1. Values of the coefficients of the polynomials functions of 1st, 3rd and 5th degree obtained in the fittings according equation (13).

The values of the Statistical Analysis of Residuals show that the linear fitting is correct. This means that the T/t data pairs generated with the inverse hyperbolic function with a value of m=4.20 10^{-6} K^{-1} min^{-1} can be satisfactorily fitted to a linear function. At the laboratory, this allowed us to impose a linear heating rate (slope of 0.3734 K min^{-1}), starting (t=0) at T= 298.15 K. Thus, it was guaranteed certain that the T/t experimental points would coincide with the exact solution of the differential rate equation (12). That is, it ensures satisfactory agreement between the kinetic values of the linear functions and those of the inverse hyperbolic branch.

The profiles of the non-isothermal kinetic curves (α/t) sometimes show concave down segments while the isothermal curves are always concave up. This can be explained in terms of the notion that in non-isothermal curves there are two opposing phenomena that affect the reaction rate: the increase in temperature with time increases the reaction rate, and the logical

decrease in the concentration of reagents as they are consumed reduces it. Depending on which of the two phenomena predominates, one concavity or the other will be observed and even when both effects are balanced quasi-linear profiles appear. This can be seen in Figure 1, where the curve (I) is a sigmoid corresponding to non-isothermal kinetics in which three behaviours are clearly visible. Segment (a) is concave down since the effect of temperature predominates and segment (c) is concave up since the effect of the temperature decrease and the concentration influence predominates. The intermediate segment (b) is linear because both effects are balanced. Curve (II) has a linear profile since both effects are equilibrated almost throughout the kinetics and the whole of curve (III) is concave up, as corresponds to an isothermal kinetics in which exclusively of the effect of concentration exists.

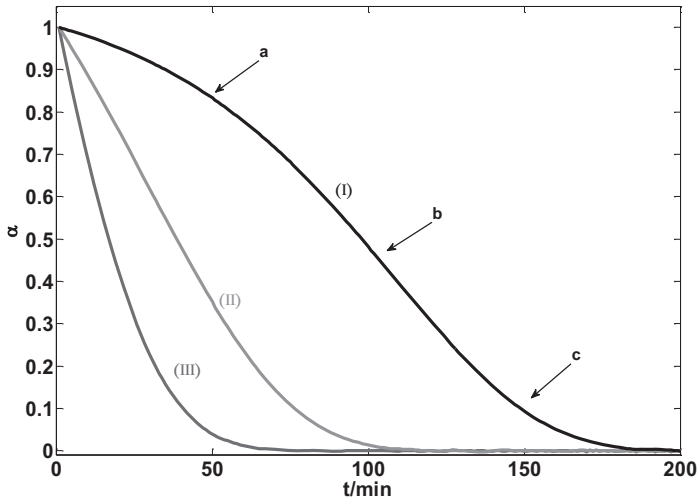

Figure 1. Several profiles of non-isothermal and isothermal kinetic curves (α/t). Curve (I) is a sigmoid corresponding to non-isothermal kinetic in which three behaviours are clearly visible: segment (a) is concave down since the effect of temperature predominates; segment (c) is concave up since the effect of the temperature decrease and the concentration influence predominates; segment (b) is linear because both effects are balanced. Curve (II) has a linear profile since both effects are equilibrated. Curve (III) is concave up, as corresponds to an isothermal kinetic.

3.2. Experimental Design (ED)

The design and implementation of a suitable Experimental Design (ED) is crucial to ensure the success of the *training* process of neural networks. We consider 2 *factors* (A and Ea) whose *responses* are the non-isothermal kinetic data of the base of the input curves (α/t). It is necessary to consider 2 variables: *a)* the extreme values of both *factors* that configure the *experimental domain* and *b)* their relative values. Both variables must ensure that the binary combinations of both *factors* will generate a set of kinetic curves that will have sufficient information to ensure an optimal *training* process of the neural network. In addition, the number of *levels* of the *factors* of the ED must be suitable if one is to avoid useless computational work and avoid large differences in the spacing of the values of the *responses*. Accordingly, to optimize the *training* process the kinetic curves of the *input matrix* must have efficient kinetic information and must be correctly distributed according to the choice of a suitable experimental design and an appropriate *experimental domain.* In agreement to the results of the study of the functions T/t, we have generated the non-isothermal kinetic data (α/t) in order to obtain the curves of the *input* matrix to perform the *training* process of the neural network from the ATPs organized according a suitable ED distribution. Bearing in mind the characteristics of the ANN computation of the non-isothermal kinetic system under study, we have chosen as the optimum one, the ED corresponding to the *Central Star Composite Experimental Design (CSCED)* distribution represented in Figure 2.

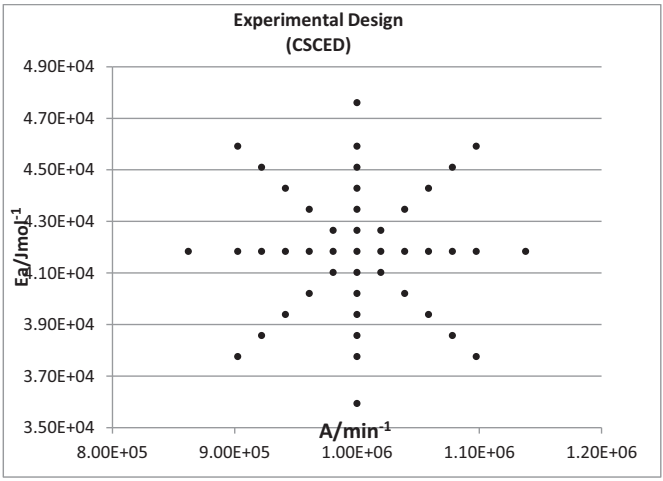

Figure 2. Experimental Design used with a distribution of the experiments corresponding to the named *Central Star Composite Experimental Design (CSCED)*.

3.3. Analysis of the SQD Minimum

When a mathematical optimization algorithm is applied, it is appropriate to analyze graphically *a priori* the profiles of the minima of the SQD, a numerical function dependent upon the ATPs, in order to know whether singular points (*local maxima* and/or *minima*, *saddle points*, etc.) are present in the neighborhood that will make the gradient vector equal to zero, leading the optimization process to fail. It should be noted that to determine the values of ATPs the mathematical expression depicted in equation (12) is fairly complex because it includes several exponential functions with different levels of exponents. Since SQD is a numerical function, we plotted a 3D graphic, generating a large set of non-isothermal curves from pairs of values of the ATPs parameters distributed according to a new CSCED, now in the neighborhood of the minimum that coincides with the *central point* (c_{min}). Later, the *SQD(c)* values are determined for each curve *(c)* according equation (17):

$$SQD(c) = \sum_{t=0}^{t_f} [\alpha_t\,(c) - \alpha_t(c_{min})]^2 \qquad (17)$$

The α_t values with a hyperbolic T/t variation were generated with the equation (12) from pairs of Ea/A values. There were three matrices corresponding to the three coordinate axes (*Ea*, *A* and *SQD*) whose graphic plot (Figure 3) shows the minimum of the *SQD(c)* function for both parameters. After the axes had been rotated suitably to obtain the best perspective for visualization, no singular points were observed in the neighborhoods of the minima.

To better confirm this, a "*cut*" of the 3D figure was made by means of a bisecting plane perpendicular to the XY plane, thereby obtaining a 2D representation (Figure 4) that clearly showed the minimum corresponding to the values of A= 10^6 min^{-1} and Ea= 4.184 10^4 J mol^{-1}. It can be observed the absence of any singular points that might lead the optimization process to fail when the AGDC algorithm was applied.

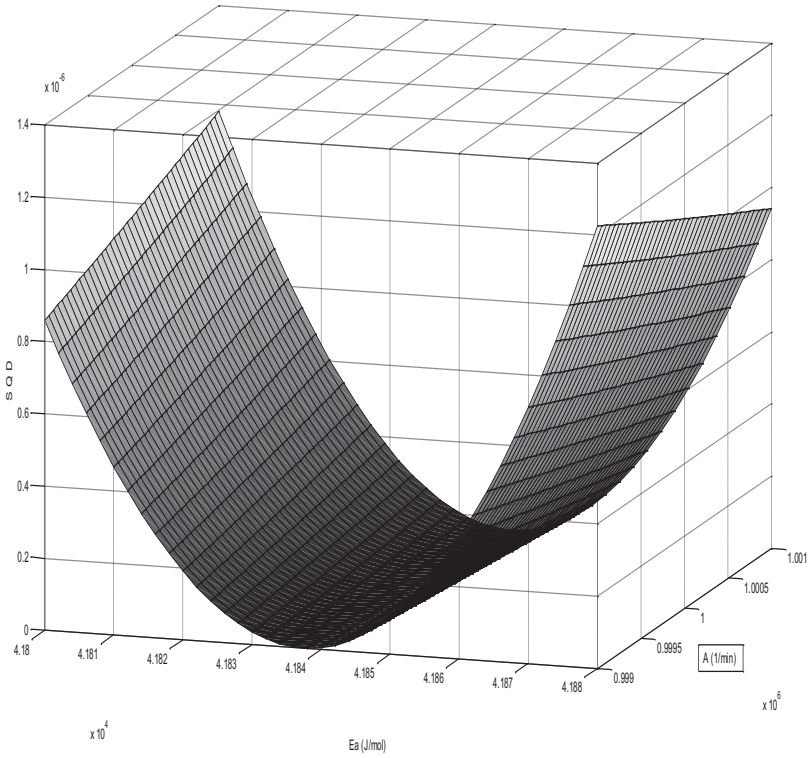

Figure 3. Graphic 3D Plot of the numerical function SQD *vs.* A and Ea showing the minimum of the function for both parameters.

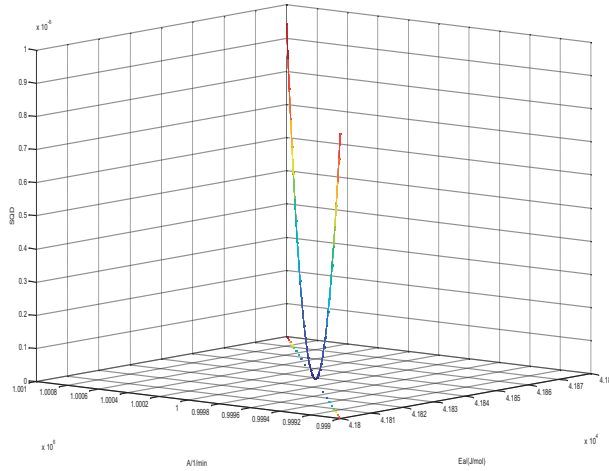

Figure 4. Plot of the *"cut"* of the 3D figure (Fig. 3) obtained by means of a bisecting plane perpendicular to the XY plane where the minimum with respect to both parameters (A and Ea) is clearly showed.

4. Results and Discussion

ANN treatment involves performing a rigorous process of *training* of the neural network. For this, we carried out a previous study to determine the ideal conditions for application. We generated 45 non-isothermal kinetic curves from eq. (12), considering an inverse hyperbolic variation of T/t according eq. (10), with a value of m=4.2 10^{-6} K^{-1} min^{-1} in a period of time sufficient for α values of at least 1.5 *half-lives* to be attained. Each curve had 50 data on which random *noise* was imposed with a magnitude of the order of the experimental error of the data ($\pm 1 \cdot 10^{-4}$). For the *back-propagation* algorithm we chose the Levenberg-Marquardt algorithm as the most ideal for performing the optimization of the elements of the *output* and *target* matrices, in all cases using a percentage ratio of 80/10/10 for the *training/validation/testing* processes. To determine the optimum architecture of the neural network, we performed the treatment varying the number and configuration of the *hidden* layer/s systematically, considering all cases possible. Thus, in the case of a neural network (50,1,2) and a variable configuration of the only one *hidden* layer (3-25 neurons), for the processes of *training*, *validation* and *testing* we obtained *output/target* regression lines with strong dispersions, very poor fitting parameters and unacceptable MSE values, number of *epochs*, μ values, *gradient,* etc.

In the case of the architecture of the neural network with 2 and 3 *hidden* layers (50,2,2) and (50,3,2) and different configurations, we used the same *training* conditions as in the case of 1 *hidden* layer. However, one aspect that merits detailed comment. In the *output/target* regression lines relative to the set of both parameters, the points of the dispersions corresponding to each parameter appear very separated (Figure 5) owing to the great difference in the order of magnitude of Ea (10^4) and A (10^6). In the case of the deviations of A, there is apparently good linearity and in the case of Ea all the dispersion values are superposed and produce a single point, due to the broadness of the scales (10^4-10^6) to be considered on both coordinate axes. This strongly masks the true effects of the dispersion, making the statistical fitting parameters of the regression lines obtained for both parameters considered jointly apparently acceptable. However, on considering the *output/target* regression lines of each of the parameters separately, where the appropriate scales are used, a great dispersion is seen in the case of the Ea (Figure 6), corresponding to the dispersion of points observed in Figure 5 values of R^2 (0.1058), slope (0.0886) and ordinate at the origin (0.0383) are obtained that show a very poor correlation between the *output* and *target* values, which make these parameter values determined by ANN unacceptable as final values. Accordingly, it is necessary to further improve the parameter values, which is done by application of the second part of the HA by means of the AGDC gradient method, observing the definitive values of the optimized parameters.

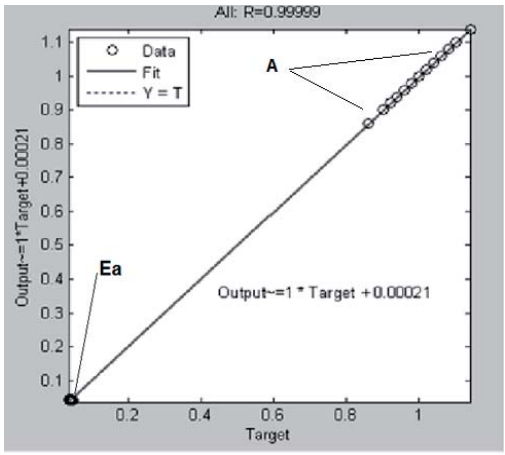

Figure 5. Regression line of the data from the elements of the *Output vs. Target* matrices when both parameters (A and Ea) are represented together for the process of *training* of the ANN in the case of the architecture of the neural network with 3 *hidden* layers (50,3,2).

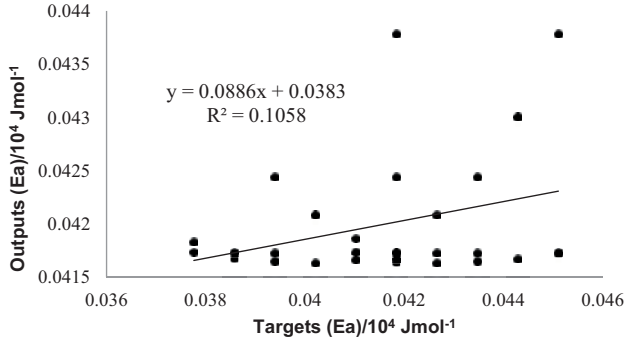

Figure 6. Regression line of the data from the elements of the *Outputs vs. Targets* values in the case of exclusively Ea for the process of *training* of the ANN when the architecture of the neural network has 3 *hidden* layers (50,3,2).

It is necessary to quantify the values of these individual *errors* and *Standard Deviations* of each parameter (*SD(A)* and *SD(Ea)*) obtained with the application of the ANN, with relative *output* and *target* values of the set of 45 curves used in the following general expression for *SD* :

$$SD(b_{ij}) = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{nc} [(b_{ij}^{Outputs} - b_{ij}^{Targets})/(b_{ij}^{Targets})]^2}{n_p . n_c} \right)^{1/2} \qquad (18)$$

where n_c is the number of curves; n_p is the number of parameters per curve (n_p=1*)* and b_{ij} are the ATPs parameters (Ea and A). The values obtained range between are unacceptable such that it is necessary to improve the results after application of the complete HA, whose results obtained are shown in Table 2.

We performed the *training* of a large set of neural networks considering all the possible configurations for architectures with 2 and 3 *hidden* layers and discarding those of a single layer. In light of the large number of cases assayed, we only show the results of the most significant configurations of the curve corresponding to the central point, because it is the most important one of the 45 comprising the ED. The *output* values of Ea and A and of the deviations in % (*Dev %*) with respect to those that served to generate the data are shown in the first part of Table 2, while the second part shows the values of Ea and A optimized with AGDC together with the deviations in % of the real values (*Dev %*). It is possible to note the clear improvement achieved with application of the complete HA on comparing the values of

the % of deviation (columns 8 and 9) of both parameters, with the corresponding values after the application of ANN (columns 4 and 5).

	ANN				HA = (ANN + AGDC)			
Config.	**A /10^6 min^{-1}**	**Ea /10^4 Jmol^{-1}**	**Dev. % (A)**	**Dev. % (Ea)**	**A /10^6 min^{-1}**	**Ea /10^4 Jmol^{-1}**	**Dev. % (A)**	**Dev. % (Ea)**
14/10	0.9800	4.1933	1.9990	-0.2233	1.0070	4.1859	-0.7038	-0.0453
15/10	0.9801	4.1871	1.9908	-0.0746	0.9955	4.1829	0.4451	0.0253
7/10/10	0.9800	4.1871	1.9965	-0.0738	0.9955	4.1829	0.4548	0.0259
9/10/10	0.9801	4.1950	1.9995	-0.2631	1.0101	4.1867	-1.0104	-0.0640
10/9/10	0.9800	4.1961	1.9995	-0.2885	1.0120	4.1872	-1.2051	-0.0758
10/10/7	0.9804	4.1917	1.9628	-0.1847	1.0043	4.1852	-0.4260	-0.0283
10/10/13	0.9795	4.1938	2.0491	-0.2331	1.0075	4.1860	-0.7521	-0.0483
10/12/10	0.9799	4.1885	2.0002	-0.1082	0.9981	4.1836	0.1880	0.0094
10/20/10	0.9806	4.1893	1.9438	-0.1273	0.9999	4.1841	0.0094	-0.0015
11/10/10	0.9800	4.1923	1.9955	-0.1993	1.0052	4.1854	-0.5212	0.0341
20/10/10	0.9795	4.1935	2.0514	-0.2267	1.0500	4.1866	-0.7017	-0.0452

Table 2. Values of the ATPs, Errors and Deviations obtained after application of ANN and the complete algorithm HA for the *training* processes of a large set of neural networks considering all the possible configurations for architectures with 2 and 3 *hidden* layers

The improvement in the final values of the ATPs is clearly seen on performing a joint plot (Figure 7) of the *Residuals* of α calculated with the values of the parameters obtained by applying the ANN [eq. (19)] and those obtained after the application of the complete algorithm HA [eq. (20)], observing improvements of up to 11-fold (0.023 for HA and 0.25 for ANN) in the time interval of 10-20 minutes.

$$[Res(\alpha)_{ANN} = \alpha_{synt} - \alpha_{cal}(ANN)] \quad (19)$$

$$[Res(\alpha)_{HA} = \alpha_{synt} - \alpha_{cal}(HA)] \quad (20)$$

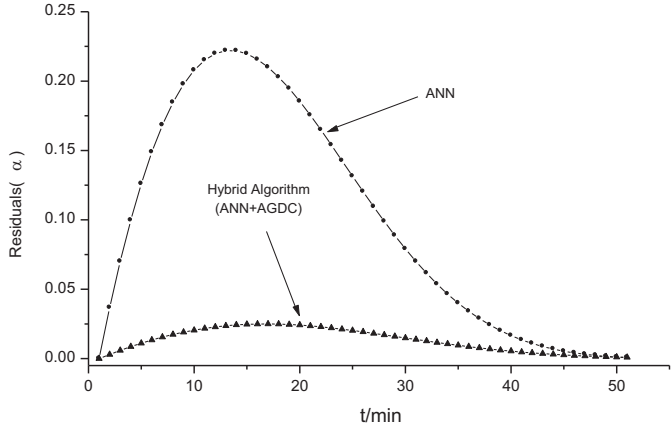

Figure 7. Plot of the *Residuals* of α calculated from equation (19) with the values of the parameters obtained by applying the ANN and those obtained from equation (20) after the application of the complete algorithm HA.

After analyzing the values of all the statistical moments of errors (*SD(A)* and *SD(Ea)*) and Deviations, it may be concluded that the optimal architecture of the neural network is (50,3,2), constituted by 3 *hidden layers* with a configuration of 10/20/10 used to carry out the later processes of *ANN prediction* and AGDC optimization.

The process of *prediction* to determine the final optimized parameters consisted of application of the HA to new non-isothermal kinetic curves with *"noise",* generated from the values of A and Ea within the maximum intervals defined in the ED. For the previous step of *prediction* applying ANN we used the neural network that proved to be optimal after the exhaustive process of *training* -architecture (50,3,2) and configuration (20/10/10)-, in all cases computing a base of 4 kinetic curves, which proved to be the ideal number for carrying out the multiple and simultaneous *prediction* of the initial ANN step. Table 3 shows the results obtained for the *prediction* of 4 pairs of A/Ea values *("Real Values")* that served to generate the *input curve base* with *"noise"*. After later application of the AGDC we obtained the optimized values of the parameters *("Optimized Values")* resulting from application of the complete HA. Bearing in mind that the percentages of deviations (*"% Dev"*) have an order of magnitude similar to that of the deviations of the *training* process of the neural network, the optimized values of A and Ea can be accepted as being definitive for the 4 *prediction* processes.

HA Algorithm (ANN+AGDC)					
Real Values (A) /10^6 min^{-1}	**Real Values (Ea) /10^4 $Jmol^{-1}$**	**Optimized Values (A) /10^6 min^{-1}**	**Optimized Values (Ea) /10^4 $Jmol^{-1}$**	**% Dev. (A)**	**% Dev. (Ea)**
0.980001	4.18914	0.9897	4.1865	-0.99	-0.06
0.960018	4.19443	0.9792	4.1891	-2.00	-0.12
0.980001	4.18914	0.980001	4.1891	-0.00015	-2.11
0.859108	4.18082	0.8537	4.1824	0.61	0.04

Table 3. Values of the ATPs, Errors and Deviations obtained after application of the complete algorithm HA for several *Prediction* processes when a neural network with a optimal architecture (50,3,2) and configuration (20/10/10) has been applied

5. Conclusions

Upon the analysis and comparison of the results shown in Figures 5 and 6 we observed that the graphical representation of the values of targets and outputs jointly corresponding to both parameters (A and Ea) according the plot obtained by the ANN application of Matlab provides a correlation that seems acceptable. However, on considering the *outputs/targets* regression lines of each of the parameters separately, where the appropriate scales are used, a great dispersion is shown and the results are unacceptable. That justifies the need of later application of the AGDC optimization algorithm in a second step to improve the results as Table 3 and Figure 7 reveals when the *Residuals* from ANN and HA are jointly plotted. The application of the hybrid algorithm (HA) that we propose in the present work provides good and satisfactory results for the optimized values of the ATPs. HA offers several advantages with respect to the isothermal classic methods since it allows to optimize ATPs directly without the need to determine the kinetic constants previously. In addition for the case of non-isothermal experiments, a single replicated kinetic experiment is enough since it allows the computation of a larger number of kinetic data that implies a considerable saving of reagents and laboratory time.

6. References

[1] J. L. González-Hernández, M. M. Canedo, C. Grande, Optimization of kinetic parameters. Multipurpose KINAGDC(MW) non-linear regression program, *Chemom. Intell. Lab. Syst.* **39** (1997) 77-84.

[2] M. Maeder, K. J. Molloy, M. M. Schumacher, Analysis of non-isothermal kinetics measurements, *Anal. Chim. Acta* **337** (1997) 73-81.

[3] A. Niazi, R. Leardi, Genetic algorithms in chemometrics, *J. Chemometrics* **26** (2012) 345-351

[4] M. Maeder, Y. M Neuhold, G. Puxty. Application of a genetic algorithm: Near optimal estimation of the rate and equilibrium constants of complex reaction mechanisms, *Chemom. Intell. Lab. Syst.* **70** (2004) 193-203.

[5] C. Hervás, J. A. Algar, M. Silva, Correction of temperature variations in kinetic-based. Determinations by use of pruning computational neural networks in conjunction with genetic algorithms, *J. Chem. Inf. Comput. Sci.* **40** (2000) 724–731**.**

[6] S. Curteanu, H. Cartwright; Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks, *J. Chemometrics* **25** (2011) 527-549.

[7] 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.

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

[9] M. M. Canedo, J. L. González-Hernández, A new computacional application of the AGDC algorithm for kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* **66** (2003) 63-78.

[10] M. M. Canedo, J. L. González-Hernández, ANALKIN(AGDC): a multipurpose computational program for the kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* **66** (2003) 93-97.

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

[12] M. M. Canedo, J. L. González-Hernández, S. Encinar, Application the computational method KINMODEL(AGDC) to the simultaneous determination of kinetic and analytical parameters, *Appl. Math. Comput*. **219** (2013) 7089-7101 .

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

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

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

[16] MathWorks MatLab, R2012a, Vs7.14.0.739 (2012).

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

[18] M. M. Canedo, J. L. González-Hernández, ANALKIN(AGDC): a multipurpose computational program for the kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* **66** (2003) 93-97.

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

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

[21]. M. Meloun, J. Havel, E. Hölfeldt, *Computation on Solution Equilibria*, Horwood, Chichester, 1988.