International Journal of QUANTUM CHEMISTRY

An Ill-Posed Inverse Problem in Enzymatic Kinetics: Jack-Bean Urease Denaturation by an Anionic Surfactant

E. Borges,*[a] D. C. Menezes,[a] and J. P. Braga[b]

A neural network procedure is applied to solve an ill-posed inverse problem in chemical kinetics relevant to chemical and biological areas; calculation of rate constants for the jack-bean urease denaturation by an anionic surfactant agent from experimental enzyme concentrations. The efficiency of the algorithm proposed is compared with the Simplex and the Levenberg–Marquardt techniques, often used in nonlinear regression methods to solve this kind of kinetics problem. The neural network approach is numerically stable and robust with respect to different initial guesses for the rate constants and noises in experimental data. © 2012 Wiley Periodicals, Inc.

DOI: 10.1002/qua.24163

Introduction

Actually, an important research field with applications to several sciences is the ill-posed inverse problems theory. The main purpose of this area is the development of methods to retrieve meaningful physical information from experimental data using linear or nonlinear numerical models. This problem is called ill-posed if one or more of three properties of its solution, existence, uniqueness, and continuity with respect to experimental noises, are not addressed. In these cases, special numerical approaches are required to solve inverse problems.[1] Ill-posed inverse problems arise in fields such as optical computerized tomography and radiotherapy planning exploited in medical diagnostics and treatments,[2,3] biochemical processes,[4] prospection techniques in geophysics,[5] technological researches,[6,7] and a wide variety of chemical situations.[8–10]

An important example of chemical application is the inverse problem in chemical kinetics which consists in determining kinetic parameters from concentrations data that follow the course of the reaction in real time. These concentrations can be monitored by spectroscopic techniques such as absorption spectroscopy and laser photolysis. In these approaches, one has to solve an ill-posed inverse problem because noises in experimental data imply large errors in parameters to be estimated.[11] Information concerning kinetic parameters are essential to optimization of several chemical processes and approaches such as rank annihilation factor analysis,[12] curve resolution procedures,[13] stochastic algorithms,[14] control type techniques,[15] and solution mapping programs[16] have been used for this purpose. However, these techniques can fail in removing the ill-posed feature of chemical kinetic inverse problems.[17] It is in this context that recurrent neural networks have been proposed to solve linear and nonlinear scientific and technological inverse problems.[18–22]

In this article, a neural network technique is performed to handle an enzymatic kinetics ill-posed inverse problem, the jack-bean urease denaturation by a surfactant agent below the critical micelle concentration. Urease is a metalloenzyme with two nickel ions per unit forming an active center which catalyzes urea hydrolyzes.[23] Urease can be found in bacteria, yeasts, and plants presenting functions such as insecticide, nitrogen release, and virulence. For example, *Helicobacter pylori* bacteria, a microbiological carcinogenic agent produces measurable amount of urease generating ammonia through stomach, implying an elevation of pH.[24]

Information about subunit dissociation and structural conformations concerning jack-bean urease induced by pH variations and surfactant species have been described in the literature[25,26] and kinetic studies about inhibition behaviors can be useful to help understand the role of active site metal ions in biological catalytic processes. Urease inhibition studies have also medical significance owing to urease inhibitors can be exploited in medical treatments.[27]

In this work, kinetic parameters for the jack-bean urease denaturation reaction are determined from experimental concentrations of the enzyme. Stability of the neural network method is tested in relation to experimental noises in concentrations and with respect to initial guesses for rate constants. Performance of the proposed algorithm is compared with that of the Simplex and the Levenberg–Marquardt programs, routinely applied in nonlinear regression techniques to kinetic problems.[28]

Methodology

A Hopfield artificial neural network (HANN) consists of neurons characterized by a given state u.[29] The neural states are based on input information called initial guesses. An activation

[a] *E. Borges, D. C. Menezes*
Departamento de Química, Universidade Federal de Viçosa, Centro de Ciências Exatas e Tecnológicas, Av. P.H.Rolfs, Centro, Viçosa-MG, CEP 36570-000, Brasil

[b] *J. P. Braga*
Departamento de Química, Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Av. Antônio Carlos, 6627, BH-MG, CEP 31270-901, Brasil
E-mail: emilio.borges@ufv.br

Contract grant sponsor: Fapemig and CNPq.

function $f(u)$ propagates the information through the network until a learning time τ defined by an optimization criterion is reached. In kinetic inverse problems, the function

$$f_i(\tau) = \frac{1}{2}[1 + \tanh(u_i)]$$

is appropriate for determining kinetic parameters so that it guarantees positive values for rate constants to be inverted.[22]

In HANN method, if simulated and experimental data are arranged in vector form as P_{exp} and P_{cal}, then an error function can be proposed as the following sum over m data,

$$E = \frac{1}{2}\sum_{j=1}^{m} e_j^2 \quad (1)$$

where $e_j = P_{cal_j} - P_{exp_j}$. Equation (1) is also called neural network energy (NNE) in HANN terminology. From this point, the algorithm can be described by the following steps:

i. Experimental rate constants are used to numerical integration of the differential model describing the kinetic reaction. Hence, the reagent, intermediate, and product concentrations are accomplished. The reagent or product concentrations can be taken as P_{exp}. In this study, the reagent concentrations are used because these data have information about all the rate constants which control the reaction.

ii. Initial guesses for rate constants are used to numerical integration of the differential kinetic model. Therefore, theoretical concentrations for reagent, intermediate, and product are predicted. Theoretical reagent concentrations are taken as P_{cal_j}.

iii. Initial guesses proposed in step ii are used as the neuron states u. These states are applied to build the equation,

$$\frac{dE}{d\tau} = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(e_j \frac{\partial P_{cal_j}}{\partial f_i}\frac{\partial f_i}{\partial u_i}\frac{\partial u_i}{\partial \tau}\right) \quad (2)$$

where n is the quantity of neuron states and it is equal to the number of rate constants to be estimated.

iv. By introducing the following condition,

$$\frac{du_i}{d\tau} = -\sum_{j=1}^{m}\frac{\partial P_{cal_j}}{\partial f_i} \quad (3)$$

one obtains from Eq. (2),

$$\frac{dE}{d\tau} = -\sum_{i=1}^{n} e_j \frac{\partial f_i}{\partial u_i}\left(\frac{du_i}{d\tau}\right)^2 \quad (4)$$

The basis of Eq. (3) is the HANN architecture, which predicts equality between the derivative of a neural state with respect to learning time and the derivative of NNE with respect to activation function. This means that the NNE is propagated while the neuron states change with learning time. The decreasing condition in NNE is achieved by the negative signal in Eq. (3). Due to

$$\frac{\partial f_i}{\partial u_i} \geq 0$$

$$e_j \geq 0$$

one obtains the inequality,

$$\frac{dE}{d\tau} < 0 \quad (5)$$

with the assurance that the error function always decreases with increasing learning time.

v. Integration of Eq. (4) is performed by a fourth order Runge–Kutta integrator[30] and it finishes when the error function, Eq. (1), reaches its lowest value. At this point, the final values for the neural states, that is, final rate constants, are obtained. These rate constants are then used together with the differential kinetic model to calculate theoretical concentrations P_{cal}. The condition (5) guarantees a good convergence between the properties P_{exp} and P_{cal}.

The flowchart presented in Figure 1 resumes the HANN algorithm.

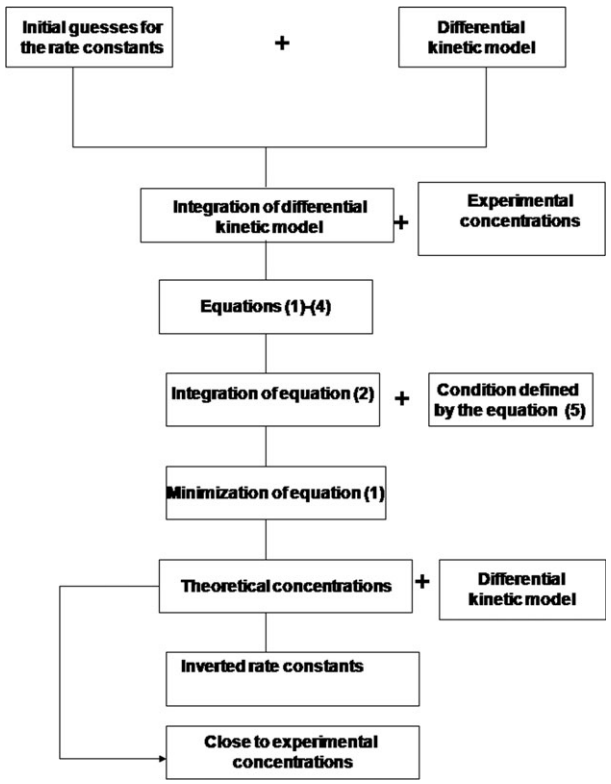

Figure 1. HANN algorithm.

Results and Discussion

For chemical kinetic inverse problems, the vector P_{cal} corresponds to concentrations fulfilled by numerical integration of differential mechanisms using initial guesses for the neuron states representing the chemical rate constants, as mentioned earlier. In inversion approaches, P_{cal} is compared with P_{exp} at each learning time point until the Euclidean distance between these

vectors, Eq. (1), is minimized. Thus, the kinetic parameters that generate the optimal P_{cal} are determined. In addition, kinetic differential models can be found if P_{exp} and preliminary kinetic parameters are known and this is called an 'identification inverse problem." The neural activation provided by $f(u)$ removes the ill-posed aspect of the numerical development; this point is not feasible using ordinary optimization techniques.

The method described here was applied to determine rate constants from experimental enzyme concentrations for the jack-bean urease denaturation by a surfactant agent. This denaturation mechanism has three reversible steps. If k_1, k_{-1}, k_2, k_{-2} are rate constants of the process, the mechanism will be[31]

$$NU \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \text{IU} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \text{DU} \tag{6}$$

where NU, IU, and DUare the native, intermediate, and denatured urease, respectively. A set of differential equations describing the process is:

$$\begin{aligned} \frac{d\text{NU}}{dt} &= -k_1[\text{NU}] + k_{-1}[\text{IU}] \\ \frac{d\text{IU}}{dt} &= k_1[\text{NU}] - (k_{-1} + k_2)[\text{IU}] + k_{-2}[\text{DU}] \\ \frac{d\text{DU}}{dt} &= k_1[\text{IU}] - k_{-2}[\text{DU}] \end{aligned} \tag{7}$$

Kinetics of urease denaturation by sodium *n*-dodecyl sulfate was investigated at neutral pH in Ref. [31]. The experimental rate constants for the fast urease denaturation by dodecyl sulfate below 1.1 and 5.5 mM of the critical micelle concentrations at 27°C and pH 7.0[31] together with the differential model given by Eq. (7) are used here to generate the enzyme concentration vector P_{exp} which is assumed as experimental data. The HANN technique does not require analytical solutions to Eq. (7) because its numerical solutions can be taken to build the P_{cal_j} vector. Initial guesses for the neuron states are inputs in HANN method. In inverse chemical kinetic problems, these neuron states represent the rate constants to be refined, as described in Methodology section.

The chemical kinetics inverse problem involves a differential operator [kinetic model given by Eq. (7)] numerically sensitive to experimental noises in the experimental concentrations. This inverse problem becomes ill-posed owing to these noises because the condition 'continuity with respect to experimental noises' as discussed in the Introduction section is violated. In this way, the noises difficult the performance of the Hopfield neural network as well as any algorithm proposed to solve ill-posed inverse problems. Finally, the rate constants achieved here are $k_1 = 0.1927$ s^{-1}, $k_2 = 5.747 \times 10^{-3}$ s^{-1}, $k_{-1} = 8.487 \times 10^{-2}$ s^{-1}, and $k_{-2} = 1.871 \times 10^{-3}$ s^{-1} and provide theoretical concentrations in excellent agreement with experimental results as the error function, Eq. (1), computed with these constants is $E = 5.116 \times 10^{-20}$.

Figure 2 shows experimental concentrations with several experimental noise levels against theoretical concentrations calculated with the inverted rate constants in each case. Experimental data are represented by full squares with error bars. It

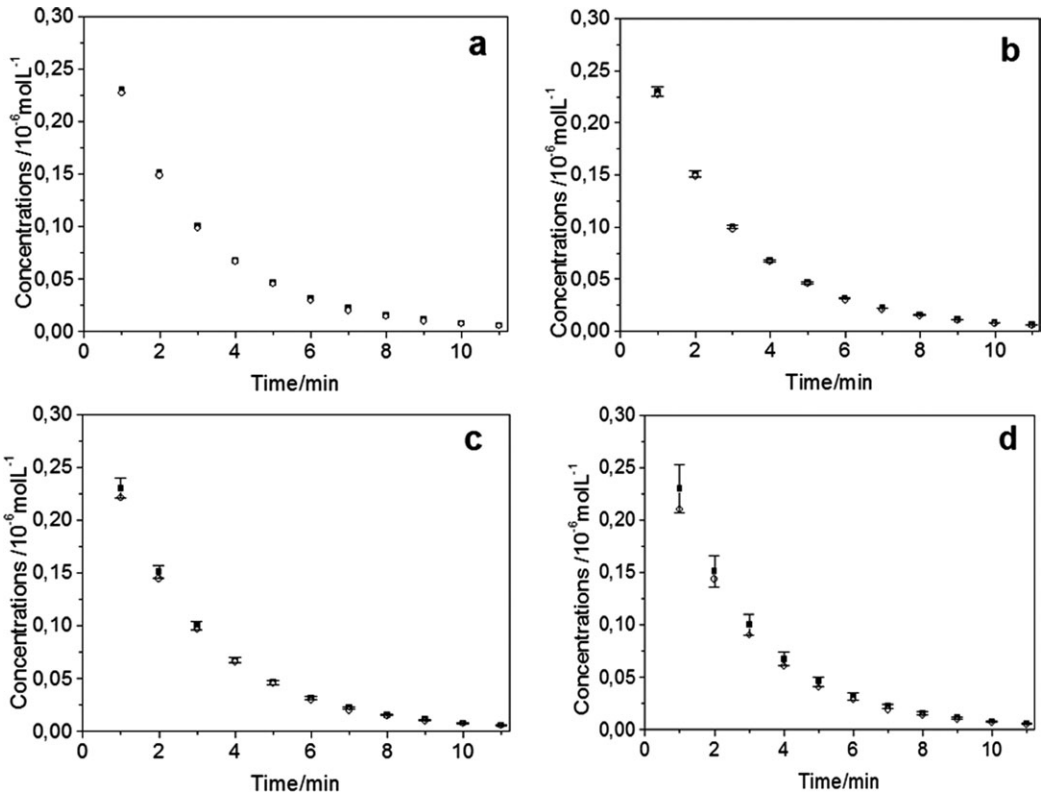

Figure 2. Experimental concentrations at different noise levels (squares) and theoretical concentrations (circles): a) concentrations without noise; b) noise level 2%; c) noise level 6%; and d) noise level 10%.

Table 1. Experimental error analysis-neural network.

Error in concentration (%)	Recovered k_1	Recovered k_2	Recovered k_{-1}	Recovered k_{-2}	Average error (%)
0	0.1927	5.747×10^{-3}	8.487×10^{-2}	1.871×10^{-3}	–
±2	0.1924	5.757×10^{-3}	8.641×10^{-2}	1.547×10^{-3}	4.9
±4	0.1932	5.747×10^{-3}	8.612×10^{-2}	1.587×10^{-3}	4.2
±6	0.2127	5.216×10^{-3}	8.552×10^{-2}	1.802×10^{-3}	6.0
±8	0.1922	5.763×10^{-3}	8.438×10^{-2}	2.613×10^{-3}	10
±10	0.1918	5.797×10^{-3}	8.413×10^{-2}	2.069×10^{-3}	3.2

Rate constants in s^{-1}.

is noted from Figure 2 that for all noise levels tested, theoretical concentrations are close to the experimental values, that is, the theoretical concentrations lie within the range of uncertainty in experimental concentrations. In other words, for all the cases, the rate constants inverted reproduce theoretical concentrations within experimental error. These results show the chemical quality of the rate constants inverted.

Random noises in the range 2–10% following a normal distribution were added to experimental concentrations to check the neural network robustness. As shown in Table 1, average errors (AEs) in inverted rate constants changes from 4.9% (noise level ±2%) to 3.2% (noise level ±10%).

The AEs were computed by the expression

$$\mathrm{AE} = \frac{e_1 + e_2 + e_3 + e_4}{4}$$

whose parameters e_1, e_2, e_3 and e_4 are, respectively, the relative deviations between constants inverted from concentrations without noise and the constants inverted from concentrations with noises. For instance, in Table 1, the values $k_1 = 0.1924\ s^{-1}$, $k_2 = 5.757 \times 10^{-3}\ s^{-1}$, $k_{-1} = 8.641 \times 10^{-2}\ s^{-1}$, and $k_{-2} = 1.547 \times 10^{-3}\ s^{-1}$ are inverted from concentrations with a noise level of ±2. Relative errors with respect to constants computed from concentrations without noises are $e_1 = 0.1557\%$, $e_2 = 0.1740\%$, $e_3 = 1.8145\%$, and $e_4 = 17.3169\%$. Hence, the AE for this case will be about 4.9% as displayed in Table 1. This procedure was applied to determine all the AEs to these parameters in this work.

AEs for the rate constants are closed to random experimental noises in each case, and for all noise levels, the neural network is numerically stable and efficient to give convergent results. Values for the error function are shown in Table 2. These errors are accomplished by Eq. (1). Low values to this equation indicate chemical quality for rate constants estimated because a good agreement between P_{cal} and P_{exp} is achieved.

If random noises are not present, a value of 5.116×10^{-20} is found. This value indicates an acceptable quality of the rate constants because an excellent agreement between P_{cal} and P_{exp} is fulfilled. If random noise level in concentrations is increased of ±2 to ±10%, the error function varies from 2.753×10^{-10} to 7.164×10^{-9}. These values are satisfactory to a stable numerical optimization method.

Also, whenever noises are not taken into account in experimental concentrations, a good convergence between different initial guesses and rate constants inverted from concentrations (taken as a reference) is observed. In this context (concentrations without noises), Table 3 shows a satisfactory convergence even with deviations up to 70% up and down with respect to the reference. In addition, to deviations 30% (up and down) in initial guesses, an efficient convergence was found at all noise levels tested as presented in Table 3. Even though whenever initial guesses with deviations greater than 30% in relation to reference are tested, an efficient convergence is not feasible. This kind of test is a difficult task to any approach used in solving ill-posed inverse problems. Nevertheless, the main objective of these methods is the quality of parameters calculated from experimental data and this can be reached by the HANN as discussed before.

Two techniques often performed in nonlinear regression analyses, Simplex and the Levenberg–Marquardt algorithms[32,33]

Table 2. Error function at different noise levels—neural network.

Random noise in concentrations data (%)	Error function, Eq. (1)
0	5.116×10^{-20}
±2	2.753×10^{-10}
±4	1.117×10^{-9}
±6	2.227×10^{-9}
±8	4.965×10^{-9}
±10	7.164×10^{-9}

Table 3. Convergence of initial guesses-neural network.

Error in concentration (%)	Error in initial guesses (%)	Average error (%)	Error in initial guesses (%)	Average error (%)
0	±30	0	±70	0
±2	±30	5.0	±70	±68
±4	±30	6.0	±70	±71
±6	±30	10	±70	±74
±8	±30	13	±70	±73
±10	±30	18	±70	±79

Table 4. Experimental error analysis—Simplex algorithm.

Error in concentration (%)	Recovered k_1	Recovered k_2	Recovered k_{-1}	Recovered k_{-2}	Average error (%)
0	0.2160	5.127×10^{-3}	8.167×10^{-2}	1.604×10^{-3}	–
±2	0.2125	5.216×10^{-3}	7.211×10^{-2}	1.798×10^{-3}	10
±4	0.2130	5.205×10^{-3}	8.192×10^{-2}	1.879×10^{-3}	6.0
±6	0.2129	5.213×10^{-3}	7.254×10^{-2}	1.886×10^{-3}	8.8
±8	0.2126	5.223×10^{-3}	8.105×10^{-2}	1.926×10^{-3}	6.7
±10	0.2169	5.121×10^{-3}	8.075×10^{-2}	1.574×10^{-3}	11

Table 5. Error function at different noise levels—Simplex algorithm.

Random noise in concentrations data (%)	Error function, Eq. (1)
0	5.560×10^{-12}
±2	8.347×10^{-10}
±4	3.004×10^{-9}
±6	3.081×10^{-9}
±8	7.319×10^{-9}
±10	7.828×10^{-9}

Table 6. Experimental error analysis—Levenberg-Marquardt algorithm.

Error in concentration (%)	Recovered k_1	Recovered k_2	Recovered k_{-1}	Recovered k_{-2}	Average error (%)
0	0.1927	4.656×10^{-3}	8.487×10^{-2}	1.871×10^{-3}	–
±2	0.1948	5.589×10^{-3}	8.667×10^{-2}	2.224×10^{-3}	6.2
±4	0.1880	5.814×10^{-3}	8.801×10^{-2}	1.043×10^{-3}	12
±6	0.1927	4.656×10^{-3}	8.487×10^{-2}	1.871×10^{-3}	4.7
±8	0.1927	4.673×10^{-3}	8.488×10^{-2}	1.885×10^{-3}	4.9
±10	0.1927	4.682×10^{-3}	8.487×10^{-2}	1.868×10^{-3}	4.7

were exploited to solve the present inverse problem to proceed a comparison with the HANN method. The three methods have similar efficiency whether random noises are not considered in experimental concentrations. When experimental noises are taken into account, the AEs in rate constants computed by Simplex are in a general way greater than those obtained by the neural network as can be seen from Table 4.

The error function obtained by Simplex algorithm changes from 8.347×10^{-10} to 7.828×10^{-9} as shown in Table 5 (random noise level in the range ±2 to ±10%) similarly to neural network performance.

The AEs in rate constants estimated by Levenberg–Marquardt and neural network are close as shown in Tables 1 and 6, but the values of error function calculated by the Levenberg–Marquardt technique are greater than those obtained by the neural network as displayed in Table 7.

For example, the following values to Eq. (1) are accomplished by these techniques: noise level at 2%; neural network, 2.753×10^{-10}; Levenberg–Maquardt method, 6.058×10^{-7}. Noise level at 10%; neural network, 7.164×10^{-9}; Levenberg–Maquardt method, 7.040×10^{-5}.

Table 7. Error function at different noise levels—Levenberg–Marquardt algorithm.

Random noise in concentrations data (%)	Error function, Eq. (1)
0	7.165×10^{-5}
±2	6.058×10^{-7}
±4	3.982×10^{-7}
±6	7.318×10^{-5}
±8	7.138×10^{-5}
±10	7.040×10^{-5}

In a general way, the HANN method was more robust than the two algorithms in relation to random noises in the experimental data because the smaller values to Eq. (1) were obtained by the neural network. This point illustrates a reasonable criterion to consider that the HANN approach estimates the best parameters when compared with Simplex and Levenberg–Maquardt approaches.

Conclusions

An algorithm based on artificial neural networks was used to solve a chemical kinetic inverse problem, the jack-bean urease denaturation by sodium *n*-dodecyl sulfate. Rate constants for a two-step kinetic mechanism were computed from the denatured enzyme experimental concentration. Analytical solutions for the differential kinetics equations are not required by the numerical method, which can be exploited for any differential equations model describing a kinetic reaction mechanism.

The rate constants calculated reproduce experimental concentrations with excellent agreement, and the neural network was more robust than Simplex and Levenberg–Maquardt methods in relation to random noises in experimental concentrations used to compute the inversion. Also, a satisfactory stability with respect to different initial guesses for the rate constants was observed. The numerical stability of the technique proposed allows inversion of parameters from data with large experimental errors. This method is applicable to solve ill-posed inverse problems in other kinetics problems.

Keywords: ill-posed inverse problem · chemical kinetics · rate constants · jack-bean urease denaturation

How to cite this article: E. Borges, DC. Menezes, JP. Braga, *Int. J. Quantum Chem.* **2012,** *112*, 3240–3245. DOI: 10.1002/qua.24163

[1] A. Kirsch, An Introduction to the Mathematical Theory of Inverse Problems, 1st ed.; Springer: Berlin, **1996**.
[2] S. R. Arridge, J. C. Schotland, *Inverse Probl.* **2009**, *25,* 123010.
[3] A. V. Chvetsov, D. Calvetti, J. W. Sohn, T. J. Kinsella, *Med. Phys.* **2005**, *32,* 501.
[4] P. Mendes, D. B. Kell, *Bioinformatics***1998**, *14,* 869.
[5] A. Abubakar, T. M. Habashy, M. Li, J. Liu, *Inverse Probl.* **2009**, *25,* 123012.
[6] M. Braun, S. A. Sofianos, H. Leeb, *Phy. Rev. A* **2003**, *68,* 012719.
[7] A. G. Ramm, Inverse Problem: Mathematical and Analytical Techniques with Applications to Engineering, 1st ed.; Springer: Boston, **2004**.
[8] A. Franceschetti, A. Zunger, *Nature* **1999**, *402,* 60.
[9] E. Borges, N. H. T. Lemes, J. P. Braga, *Chem. Phys. Lett.* **2006**, *423,* 357.
[10] L. S. Virtuoso, R. C. O. Sebastião, J. P. Braga, L. H. M. Silva, *Int. J. Quantum Chem.* **2006**, *106,* 2731.
[11] A. M. Greco, Ed. Direct and Inverse Methods in Nonlinear Evolution Equations, 1st ed.; Springer: New York, **2003**.
[12] Z. L. Zhu, W. Li, J. Xia, *Anal. Chim. Acta* **2004**, *527,* 203.
[13] E. Furusjo, O. Svenssonb, L. G. Danielsson, *Chemom. Intell. Lab. Syst.* **2003**, *66,* 1.
[14] A. Eftaxias, J. Font, A. Fortuny, A. Fabregat, F. Stuber, *Comput. Chem. Eng.* **2002**, *26,* 1725.

[15] M. Tadi, R. A. Yetter, *Int. J. Chem. Kinet.* **1998**, *30,* 151.
[16] B. Eiteneer, M. Frenklach, *Int. J. Chem. Kinet.***2003**, *35,* 391.
[17] W. Tang, L. Zhang, A. A. Linninger, R. S. Tranter, K. Brezinsky, *Ind. Eng. Chem. Res.* **2005**, *44,* 3626.
[18] R. C. O. Sebastião, J. P. Braga, *J. Magn. Reson.* **2005**, *177,* 146.
[19] E. Borges, N. H. T. Lemes, J. P. Braga, *Chem. Phys. Lett.* **2006**, *423,* 357.
[20] N. H. T. Lemes, E. Borges, J. P. Braga, *J. Braz. Chem. Soc.* **2007**, *18,* 1342.
[21] N. H. T. Lemes, E. Borges, R. V. Sousa, P. J. Braga, *Int. J. Quantum Chem.* **2008**, *108,* 2623.
[22] N. H. T. Lemes, E. Borges, J. P. Braga, *Chemom. Intell. Lab. Syst.* **2009**, *96,* 84.
[23] N. E. Dixon, G. Gazzola, R. L. Blakeley, B. Zerner, *J. Am. Chem. Soc.***1975**, *97,* 4131.
[24] L. Zhang, S. B. Mulrooney, A. F. K. Leung, Y. Zeng, B. B. C. Ko, R. P. Hausinger, H. Sun, *Biometals* **2006**, *19,* 503.
[25] D. P. Blatter, C. C. Contaxis, F. J. Reithel, *Nature* **1967**, *216,* 274.
[26] M. Hirai, R. Kawai-Hirai, T. Hirai, T. Ueki, *Eur. J. Biochem.* **1993**, *215,* 55.
[27] R. A. Burne, Y. Y. M. Chen, *Microbes Infect.* **2000**, *2,* 533.
[28] J. Nocedal, S. J. Wright, Numerical Optimization, 1st ed.; Springer: New York, **1999**.
[29] J. J. Hopfield, D. W. Tank, *Biol. Cybern.* **1985**, *52,* 141.
[30] G. E. Forsythe, M. A. Malcolm, C. B. Moler, Computer Methods for Mathematical Computations, 1st ed.; Prentice-Hall: London, **1977**.
[31] K. Nazari, A. Mahmoudi, N. Esmaeili, L. Sadeghian, A. A. Moosavi-Movahedi, R. Khodafarin, *Colloids Surf. B* **2006**, *53,* 139.
[32] K. Levenberg, *Quart. Appl. Math.* **1944**, *2,* 164.
[33] D. W. Marquardt, *J. Soc. Ind. Appl. Math.* **1963**, *11,* 431.

Received: 23 January 2012
Revised: 13 April 2012
Accepted: 17 April 2012
Published online on 15 May 2012