Comput Geosci (2015) 19:769–780
DOI 10.1007/s10596-015-9488-5

ORIGINAL PAPER

Multilayer perceptrons as function approximators for analytical solutions of the diffusion equation

Laura D. Campisi[1]

Received: 16 September 2014 / Accepted: 14 April 2015 / Published online: 17 May 2015

Abstract A novel method using neural networks to analyse diffusion profiles is presented. Multilayer perceptrons are used to approximate the analytical solution of the diffusion equation and find within it any unknown parameter that best fits a given data set. An example based on published data of diffusion of helium is examined to illustrate the main steps of the method. The exercise shows that it is possible to refine the value of diffusion coefficients up to two orders of magnitude in terms of precision. A particular feature of the method is that calibration curves can be taken into account when choosing the best setup of a network, which allows minimization of instrument specific error. A general version of the method giving the opportunity to define a local diffusion coefficient is also discussed, which could be considered for cases where the analytical solution of the diffusion equation cannot be identified a priori or does not exist.

Keywords Diffusion · Neural network · Function approximator · Inverse modelling · Nonlinear regression

1 Introduction

The use of artificial intelligence for analysing data sets has received much attention in recent decades [1–5] and has been successfully considered for several geosciences applications [6–10]. Yet, to the best of my knowledge, no papers have been devoted to methods using neural networks (NN) to analyse diffusion profiles.

One of the distinctive features of multilayer perceptrons (MLP) and radial basis function networks is their recognized potential as universal approximators [11–17]. The use of this characteristic will be the focus of the present work.

Crank [18] and Carslaw and Jaeger [19] offer a wide range of analytical solutions for different initial and boundary conditions so diffusive processes could be potentially described with an analytical approach. However, in practice most of the available solutions of the diffusion equation remain a curiosity for mathematicians since only a few of them are considered when analysing data sets or designing a diffusion experiment. This is because back calculating even a single parameter (typically the diffusion coefficient D) from measurements of diffusion profiles can be cumbersome. Consequently, the most common method for identifying the magnitude of D still remains the graphical method, which is possible only with a limited number of experimental setups and solutions [18, 20–23].

It is important to point out that being able to use an analytical approach for understanding diffusion gives an invaluable insight when analysing data sets. Firstly, when considering a function, the relationships between all the parameters involved are clear. Moreover, the strategy of considering certain solutions would give the opportunity to exclude the presence of other processes that might occur during diffusion (evaporation, interface resistance or chemical reactions).

It is worth mentioning a very interesting review by Fredericks [24]. The author illustrated the wide range of the diffusion coefficients in halides obtained by different research groups. He emphasized the pros and cons of every setup and type of instrument used for extracting the magnitude of D.

✉ Laura D. Campisi
ldrcampisi@gmail.com

[1] School of Environmental Sciences, University of East Anglia, Norwich, UK

In spite of the simplicity of the medium (cubic lattices and a clear electrostatic potential), Fredericks showed the difficulty of trying to completely isolate and measure the effect due to diffusion.

The present paper has been written to underline the potential of artificial intelligence to facilitate the analysis of data sets from diffusion experiments. It will be focused on showing how NNs with relatively simple architectures could be used to approximate analytical solutions of the diffusion equation. A description of a NN is given in Section 2, which also introduces the main features of a MLP. In Section 3 a practical example of how the method works is made based on a published data set. In Section 4 a general method is examined, which could be useful for cases where the analytical solution of the diffusion equation cannot be identified a priori or does not exist. In Section 5 the merits and the limits of the method are summarized and the reason for its uniqueness is discussed.

2 Neural networks

It is beyond the intention of this section to give a detailed explanation of what a NN is and the reading of a good handbook on the theory of such networks is suggested [25, 26].

The fundamental unit of a NN is a neuron, which is often represented with one or more inputs, the main body (consisting of weights, a bias, a weight function and an activation function) and a single output leaving the neuron (Fig. 1). Any input I_i, which can be a scalar or a vector, is first weighted with its appropriate weight w_i after entering into the main body of a neuron. Thereafter, a weight function f combines all the weighted inputs and the bias b to generate a single parameter y. This parameter is finally used to calculate the output z of the neuron with the activation function g.

Two or more neurons can be combined to form a layer by a weight matrix W. Two layers are connected when the output of neurons of one layer becomes an input of neurons of the other layer (Fig. 1). The layer of a NN generating the final output is called the output layer, whilst the others layers of neurons are referred as hidden layers. Several categories of NNs are defined, mainly depending on the architecture of the network (the framework generated by the connections between neurons and between layers), the main components of the neurons and/or the learning technique.

A MLP is a feedforward NN, which means that the connections between neurons do not form any cycle and the outputs of each neuron always move towards the output layer.

It utilizes a supervised learning technique because during training all weights and biases are iteratively adjusted to minimize the distance between modelled and desired outputs. Particularly, a MLP makes use of the backpropagation algorithm in conjunction with an optimization method. The term backpropagation refers to the fact that, at each iteration, the update starts with the weights and biases of the output layer and moves backward in respect to the direction of the inputs of the network.

3 Diffusion experiments and neural networks: a practical example

In this section a practical example is given to show the relative simplicity of creating a NN that would approximate a solution of the diffusion equation and find the value of any unknown parameter within the solution that best describes a data set.

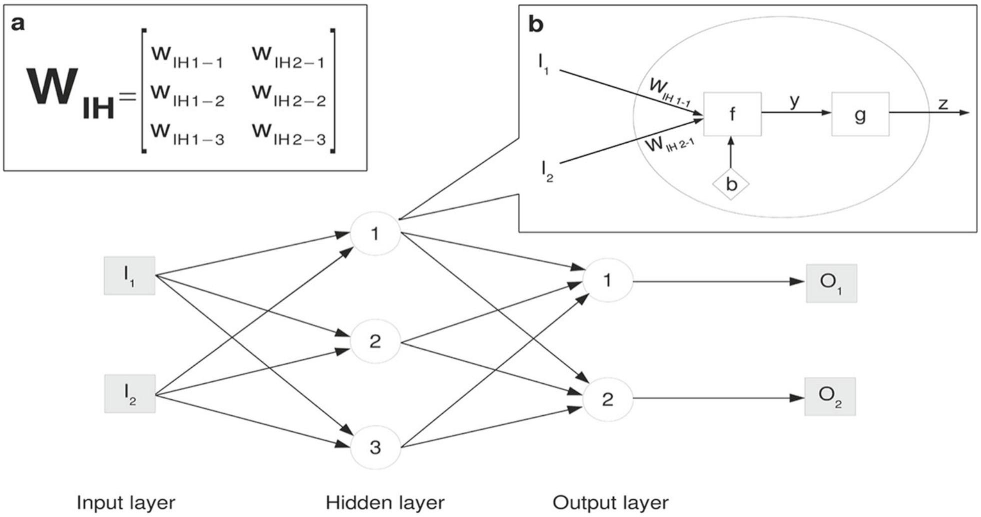

a
$$W_{IH}=\begin{bmatrix} w_{IH1-1} & w_{IH2-1} \\ w_{IH1-2} & w_{IH2-2} \\ w_{IH1-3} & w_{IH2-3} \end{bmatrix}$$
b
I_1
I_2
W_{IH1-1}
W_{IH2-1}
f
y
g
z
b
I_1
I_2
1
2
3
1
2
O_1
O_2
Input layer
Hidden layer
Output layer

Fig. 1 Diagram of a two input, two output NN with three layers. **a** The matrix W_{IH} connecting the input layer with the hidden layer. **b** The main body of a neuron

The NNs in this paper have been created by using Matlab built-in functions. The reader is referred to the user's guide [27] for a detailed description on how to create, train and test a custom network with the Neural Network Toolbox.

This example is based on published data concerning diffusion of helium in rutile and titanite [28]. The authors considered the solution for a semi-infinite medium ($x > 0$) with an impermeable boundary (i.e. no flux at $x = 0$) and a bell-shaped initial distribution (Eq. 1, from [29]).

$$N = \frac{N_m}{2\sqrt{1+A}}\left\{\exp\left(\frac{-(x-R)^2}{2\Delta R^2(1+A)}\right)\left[1+\text{erf}\left(\frac{AR+x}{\Delta R\sqrt{2A(1+A)}}\right)\right] +\exp\left(\frac{-(x+R)^2}{2\Delta R^2(1+A)}\right)\left[1+\text{erf}\left(\frac{AR-x}{\Delta R\sqrt{2A(1+A)}}\right)\right]\right\} \left(A=\frac{2Dt}{\Delta R^2}\right) \quad (1)$$

N is the concentration of the implanted species, N_m is its maximum concentration at the start of the diffusive process, R is the position (depth in the material) of the peak of implanted species and Dr (ΔR in Eq. 1) is the range straggle (full width at half maximum of the bell representing the initial distribution). The study case has been chosen because the authors used a solution that is more complex than the relationships that are usually considered for diffusion experiments. In [28] the parameters R and Dr for rutile and titanite are obtained with the Monte Carlo simulation programme SRIM2006 [30]. According to the authors, two or three points of the diffusion profiles are determined with nuclear reaction analysis (NRA) and the diffusant concentration up to a maximum depth of a few microns is measured. The diffusion coefficient is then found with a graphical method, by comparing the measurements with the calculated profiles for as-implanted unannealed samples.

For this exercise it is assumed that N from three points of a diffusion profile is available and that the parameters D, Dr and R of Eq. 1 that best fit this data set have to be found. It is assumed that $t = 10^3$ (s) and $N_\mathrm{m} = 100$ (arbitrary unit).

One of the advantages when using this method is that finding Dr and R at the same time as D allows verification that ion implantation has been made consistently in all the samples. In other words, the variability of R and Dr due to implanting can be taken into account and investigated. This potential source of error would be otherwise difficult to quantify.

The first step is to determine the ranges within which to search for the unknown parameters. With two or more parameters to find, it is best to restrict their ranges as much as possible, otherwise the training times might be too long or the overall performance of the NN could be poor. As will be shown later in this section, if the network setup is selected carefully, it is possible to verify whether the initial guess was wrong and have the opportunity to start again with new parameter ranges.

Table 1 shows the ranges within which the parameters Dr, R and D are searched. Such ranges are chosen based on the simulated values of Dr and R with SRIM2006 for rutile and titanite and the retrieved values of D in both phases when the temperature is in the range 400–500 °C [28].

Figure 2 shows diffusion profiles in the first 1.5 μm calculated with Eq. 1 for four different diffusion coefficients and using the values of R and Dr obtained with SRIM2006 for the rutile (R = 3790 Å and Dr = 710 Å). Similarly, the effects of R and Dr on N have been investigated.

The choice of which points of the profile to measure is crucial for refining the unknown parameters and determining the overall precision of the method. This is true for two reasons.

Firstly, the function to be approximated could be such that at some points small changes in any unknown parameter might cause large differences in N. The NN could therefore require a high number of neurons or a different optimization technique to achieve a good level of performance. Thus, the setup of a NN that works for a definite set of points might not be a good choice if a different set is considered.

Secondly, measurement error could make two sets of measurements corresponding to different parameters indistinguishable. As an example, let us compare two diffusion profiles generated with the same parameters (N_m = 100 (a.u.), $t = 10^3$ (s), Dr = 710 (Å) and $R = 3790$ (Å)) except for D(7×10^{-17} and 10^{-16} (m^2/s) respectively) (Fig. 3). There are three additional profiles in Fig. 3, which have been generated by calculating N with $D = 7\times10^{-17}$ and then by subtracting at each point 1, 3 and 5 % of N, respectively. These three profiles have been calculated to visualize the effect of the maximum uncertainty due to measurement error.

It is evident that if there are only a few points measured and they are all too close to the surface ($x = 0 - 0.8\mu$m), then it is virtually impossible to distinguish the two profiles with different diffusion coefficients if the error on N is above 3 % (Fig. 3). As a result, regardless of its performance during training, any network would fail to provide a good answer when submitting a data set. If the region

Table 1 Ranges within which the unknown parameters Dr, R and D of Eq. 1 are sought

Dr (Å)	R (Å)	D (m^2/s)
650–850	3500–4500	10^{-17}–10^{-16}

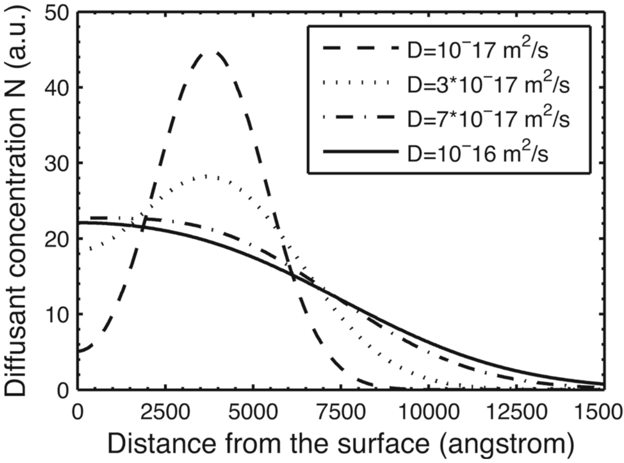


Fig. 2 The diffusant concentration within the first 1.5 μm of the semi-infinite medium calculated with Eq. 1 for four different diffusion coefficients in the range 10^{-17}–10^{-16} (m^2/s)

$x = 0.8$–1.5 μm is considered, an error of 5 % or higher would not make the two profiles indistinguishable. However, the concentration might be too low (up to six orders of magnitude compared to N_{m}) to be measured.

It can be easily verified that two diffusion profiles with distinct diffusion coefficients have common points, which identify the region of maximum uncertainty due to measurement error. However, uncertainty on Dr and R would shift such points so it might not be trivial to determine which area of the profile can minimize the effect of error on N.

As a general rule, if the points of the profile are too close, measurement errors have a larger impact on the precision of the method and more points might be needed to find the three unknown parameters. Conversely, if the points chosen are too distant from each other and a small change in a parameter results in a large differences in N, more complex architectures and/or longer training times might be required.

Another crucial step for the method is to determine a good set of units of measurement so that the pairs [inputs; outputs] to submit to the network lie in a suitable range. The units of measurement for this study case are $u = 100(\mathring{A})$ and $v = 100$ (s).

It is now described how the best results for this exercise have been obtained. The concentration is calculated with Eq. 1 at three points (0.3, 0.7 and 1.1 (μm) from the surface, respectively) by considering all the combinations of 11 values of Dr, 21 values of R and 31 values of D, each linearly distributed within their own ranges.

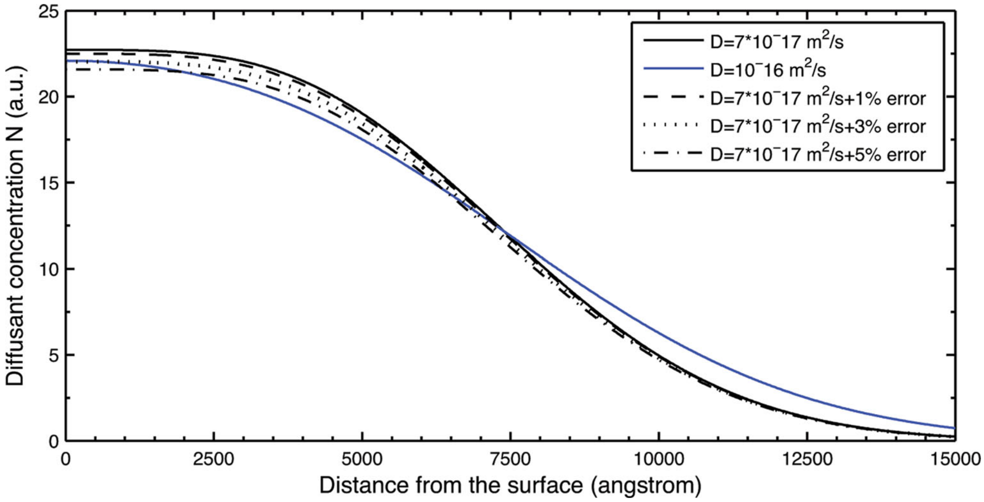
Overall, the training set consists of $11 \times 21 \times 31 = 7161$ pairs [inputs; output]. The three inputs to the network are the calculated N at the three points. The desired output of the network is a vector with the corresponding values of D, Dr and R that generated those concentrations. In this case the best architecture proved to be $30 + 8 + 3$ neurons (Fig. 4).

The weight function f of all neurons is a simple sum of the weighted inputs and the bias. The two types of activation functions g used are the linear function $z = y$ for the output layer and the logistic function $z = 1/[1 + \exp(-y)]$ for the two hidden layers.

The network training function updates weight and bias values according to the Levenberg-Marquardt optimization method. The performance of the network is monitored by considering the sum of the means of squared errors. Each neuron error is calculated as the difference between desired and modelled output.

It is important to point out that the performance of the network cannot be used to define the overall precision of the method. This is because, in case of overfitting, a network might show a good performance but a poor generalization. Generalization is the ability of networks to give reasonable answers when inputs that they have never seen are presented. Since there is no general rule to avoid the problem of poor generalization, several attempts have to be made on

Fig. 3 Two diffusion profiles are calculated with Eq. 1 and D equal to 7×10^{-17} and 1×10^{-16} (m^2/s), respectively. Three additional profiles are generated by calculating N with $D = 7 \times 10^{-17}$ (m^2/s) and then by subtracting at each point 1, 3 and 5 % of N, respectively



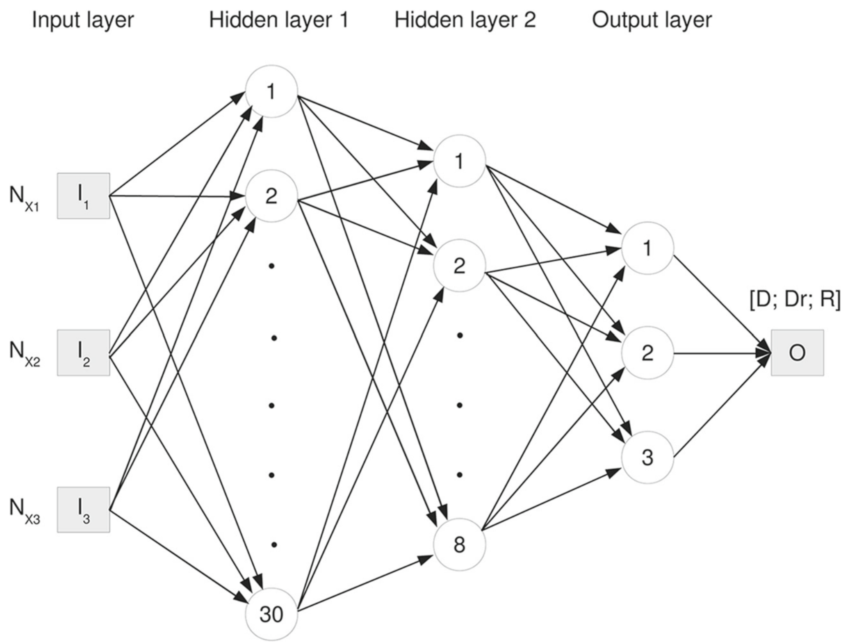


Fig. 4 The architecture of the best function approximator of Eq. 1

the setup of a network until it is proved that a good generalization has been reached. In Table 2 the modifications in the network setup that have been made in order to search for the best function approximator are listed.

It is crucial to ensure that generalization is acceptable when submitting values from measurements, which are always associated to errors. Therefore, once the training is terminated, the following procedure is chosen to test the overall precision of the trained network.

Ten numbers randomly distributed in their respective intervals are generated for R and Dr and 100 numbers randomly distributed are generated in the interval of D. Then, all the combinations of the numbers of this set (called $[R_{\text{random}}, \text{Dr}_{\text{random}}, D_{\text{random}}]$) are used to calculate the concentrations [N1sim; N2sim; N3sim] at the three points of the profile with Eq. 1. This gives a set of 10000 new inputs, which are then arranged in ascending order of D_{random}. The effect of measurement error is simulated by adding noise to [N1sim; N2sim; N3sim]. The resulting 10000 vectors [N1noise; N2noise; N3noise] are submitted to the trained network for simulation. The output $[R_{\text{net}}, \text{Dr}_{\text{net}}, D_{\text{net}}]$ obtained from the network are compared with the true value $[R_{\text{random}}, \text{Dr}_{\text{random}}, D_{\text{random}}]$. The relative error for R is measured as $[(R_{\text{random}} - \text{R}_{\text{net}}/R_{\text{random}}) \times 100]$. The error for the other two parameters is defined in the same way.

Table 2 Summary of the modifications of the network setup that have been tested to find the best approximator of Eq. 1

- Multilayer perceptron (MLP) vs. radial basis function (RDF)
- Training set size 1000–30000
- Number of hidden layers 1–3 (MLP only)
- Number of neurons per layer 1–100
- Number of points of the profile 3–10
- Position of measured points of the profile
- Range of inputs and outputs 0.001–1000
- Activation function: log-sigmoid vs hyperbolic tangent sigmoid (MLP); radial basis function vs. normalized radial basis function (RDF)
- Optimization method: Levenberg-Marquardt, gradient descent, conjugate gradient, Bayesian regulation

Figure 5 shows the best results obtained when the error on N is 1 %. In this case, the three parameters have an error that is usually below 2 %. The uncertainty on D and R increases with the magnitude of D, whilst Dr has an error that is below 1 % regardless of the magnitude of the other two parameters.

The same network gave an uncertainty always below 6 % if error on N is 3 %. The uncertainty on D and R again increases with the magnitude of D, whilst Dr has a constant error that is below 2.5 %. When the uncertainty on N is 5 %, the maximum error on D, R and Dr is 13 %. Again, Dr has a constant error of 5 % whereas the error on D and R increases with the magnitude of D.

Whenever a better precision is needed, it could be worth creating a new NN that deals with narrower ranges of the parameters by taking into account the parameters found by the first network and their relative uncertainty. The training of the first network can take 2–10 h with a MacBook Pro

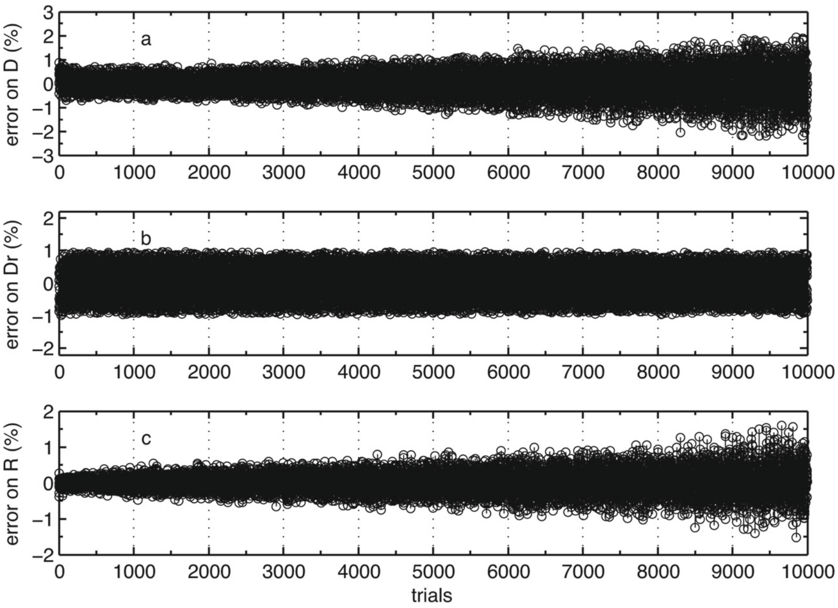


Fig. 5 Relative error (%) on D (**a**), Dr (**b**) and R (**c**) when the targets are randomly chosen from the ranges of the training set

with 2.3 GHz (Intel Core i7) and 4 GB of RAM. By contrast, a similar network that considers parameter ranges that are double the precision of the first network usually reaches a satisfactory level of performance with the same laptop within an hour.

The strategy of applying the method several times by decreasing the parameter ranges and increasing the resolution (i.e. the average distance between two targets of the training set) at each step could also be used when it is not possible to identify in advance limited ranges for the unknown parameters.

A good network should be able to indicate whether the initial guess of the parameter ranges was correct. As an example, by following the procedure to test the precision of the method when the error on N is 1 %, let us examine the case where 100 numbers for D randomly distributed within the range [10, 300] are considered. The network would still be able to indicate that D is in a higher range compared to that which was expected, even though the uncertainty will be higher (up to 23 %, Fig. 6).

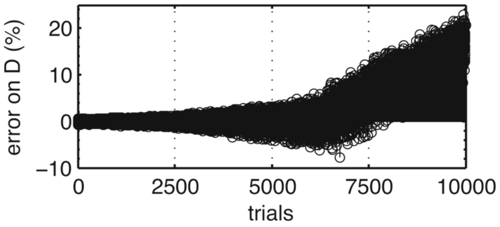


Fig. 6 Relative error (%) on D when the target is randomly chosen from the range [10, 300] and Dr and R are randomly chosen from the ranges of the training set

Similarly, if 100 numbers for D randomly distributed within the range [5, 20] are considered, the network will indicate when D lies in a range that is lower than expected.

Moreover, if the range [4, 10] is tested for Dr instead of the initial [6.5, 7.5], a maximum error of 8 % is found on Dr, and only when D is in the lower range. When the range for R is [25, 55], the network will also indicate the correct value of R within an error of up to 12 % when D is in the lower range. Figure 7 indicates the error of the three parameters if the ranges [5, 9], [30, 50] and [8, 200] are considered for Dr, R and D respectively.

The results in [28] with D in the range $1 \times 10^{-17} - 1 \times 10^{-16}$(m^2/s) and the relative uncertainties are shown in Table 3. To be able to make a comparison with this data set, the following procedure is chosen. The values assigned to R and Dr are 3790 and 710 Å respectively, and ten values linearly distributed in the range of D are chosen. N is then calculated at the three points of the profile. Each set [N1sim; N2sim; N3sim] is repeated 1000 times and random noise of fixed maximum magnitude is added to each vector. The resulting sets [N1noise; N2noise; N3noise] are finally submitted to the network for simulation. The maximum error e is calculated for each output of the network [R_{net}, Dr$_{\text{net}}$, D_{net}] by comparison with [R, Dr, D_i] ($i = 1$, 2, ..10). Since error in [28] is reported in terms of $\log_{10}D$, the greatest value between $|\log_{10}(D + e) - \log_{10}\text{D}|$ and $|\log_{10}(D - e) - \log_{10}D|$ is retained as overall uncertainty (± in Table 4).

This comparison will be purely illustrative, since the authors did not specify in the paper the location of the points measured. Overall, the use of this network gives an uncertainty on D that is on average an order of magnitude better

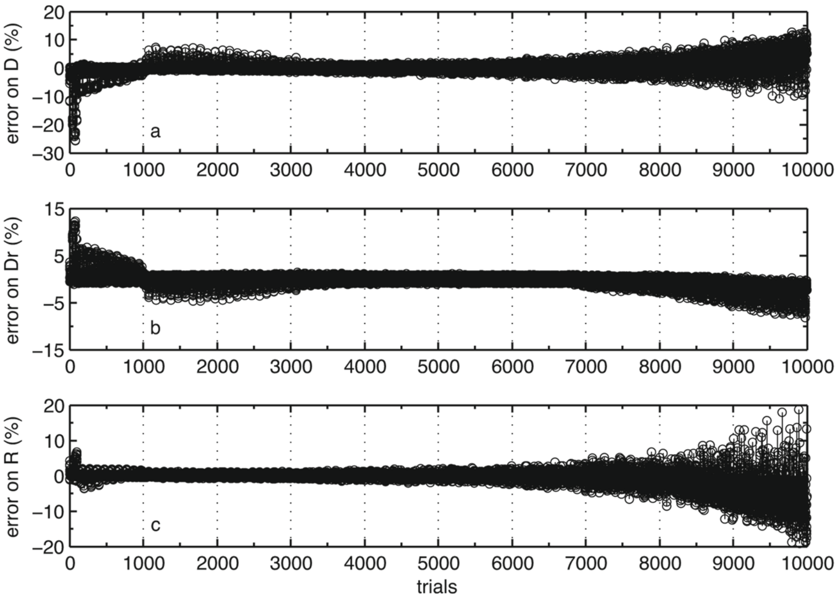


Fig. 7 Relative error (%) on D (**a**), Dr (**b**) and R (**c**) when the targets are randomly chosen from the ranges [8, 200], [5, 9] and [30, 50] respectively

than the D obtained with a graphical method. This is partially due to the initial search for the best set of data points that would minimize the effect of measurement error. The search for the points and the best setup for the network have been made by considering a 1 % error on N. It is likely that a similar search considering 10 % error would give a better result for the cases with higher uncertainties on N.

The choice of a measurement error proportional to N has been made because it represents the worst condition. In fact, the error in the original paper should have a complex behaviour, since measured N depends on the depth in the material, the beam energy and the solid angle subtended by the detector. Because the signal N also depends on the probability that a number of deuterons would impinge on the target, it is expected that higher diffusant concentrations (i.e. higher D) would result in better precision. A network created to take into account the precision and the resolution of a particular instrument could certainly be designed to improve further the overall precision of the method.

The case with only three points available has been examined because it represents the greatest challenge when three unknown parameters have to be found. In order to achieve the goal, the networks described in this paper performed a nonlinear regression.

If several points or entire profiles are measured, a simpler network setup will usually be sufficient and the training time much shorter. For instance, a training set with the same output of the NN of this section and the diffusant concentration from nine points grouped as a single input vector give similar results within only 10 min of training if the same architecture is chosen.

4 A general method

The purpose of this section is to describe a general method that could be used to extract a local D regardless of the geometrical features of the entire system and the causes

Table 3 Selection of measured diffusion coefficients published in [28]

Sample name	T (°C)	Time (s)	D (m^2/s)	Log$_{10}D$	±
He3Ru-5	450	3.6×10^3	2.81×10^{-17}	−16.55	0.25
He3Ru-15	448	1.8×10^3	8.81×10^{-17}	−16.06	0.21
He3Ru-18	559	1.8×10^3	1.40×10^{-17}	−16.85	0.32
He3JpSph-2	452	2.4×10^3	5.23×10^{-17}	−16.28	0.25
He3JpSph-3	398	7.2×10^3	2.06×10^{-17}	−16.69	0.29
He3JpSph-9	398	1.8×10^3	2.52×10^{-17}	−16.60	0.37
He3LOSph-2	451	2.4×10^3	3.13×10^{-17}	−16.51	0.20
He3LOSph-1	401	8.1×10^3	1.41×10^{-17}	−16.85	0.23
Average					*0.26*

Table 4 Method uncertainty when D is in the range 10^{-17}–10^{-16} (m^2/s) and error is 1–10 % of measured N

D(m^2/s)	Log$_{10}D$	± (1 % error)	± (3 % error)	± (5 % error)	± (10 % error)
1.00×10^{-17}	−17	0.0026	0.0073	0.0132	0.0259
2.00×10^{-17}	−16.70	0.0027	0.0079	0.0132	0.0297
3.00×10^{-17}	−16.52	0.0033	0.0098	0.0182	0.0356
4.00×10^{-17}	−16.40	0.0040	0.0125	0.0219	0.0454
5.00×10^{-17}	−16.30	0.0046	0.0141	0.0226	0.0551
6.00×10^{-17}	−16.22	0.0053	0.0165	0.0298	0.0748
7.00×10^{-17}	−16.15	0.0064	0.0211	0.0373	0.0930
8.00×10^{-17}	−16.10	0.0079	0.0236	0.0485	0.1182
9.00×10^{-17}	−16.04	0.0098	0.0305	0.0535	0.1920
1.00×10^{-16}	−16	0.0112	0.0378	0.0730	0.2624
Average		*0.0058*	*0.018*	*0.033*	*0.093*

that generated diffusion. It would allow investigation of the chance that the system is a composite medium or that D depends on time or concentration. This general method could apply when the analytical solution of the diffusion equation cannot be determined or if such a solution does not exist.

As an example, let us compare the different perspectives of two imaginary observers of a system similar to the one examined in the previous section, where a diffusant is implanted near to the surface of the sample. The first observer is aware that Eq. 1 is the general solution of the diffusion equation so only a few points of the diffusion profile are measured to extract the magnitude of D. The second observer does not know the general solution, so detailed measurements from a small portion of the profile (from X_0 to X_1) along the direction of the linear flow are undertaken. Particularly, the diffusant concentration in the region $X_0 - X_1$ at time $t_0 = 0$ and at the boundary X_0 during $\Delta t = t_1 - t_0$ are measured.

In order to identify the magnitude of D, the second observer considers the solution of the semi-infinite medium ($X > 0 = X_0$) with initial condition $f(x)$ and varying concentration $\phi(t)$ at the boundary X_0 (Eq. 2).

$$\begin{aligned} N &= u + v \\ &= \frac{1}{2\sqrt{\pi Dt}} \int_0^\infty f(x') \left[e^{-\frac{\left(x-x'\right)^2}{4Dt}} - e^{-\frac{\left(x+x'\right)^2}{4Dt}} \right] dx' \\ &\quad + \frac{x}{2\sqrt{\pi D}} \int_0^t \varphi(\lambda) \left[\frac{e^{-\frac{x^2}{4D(t-\lambda)}}}{(t-\lambda)^{\frac{3}{2}}} \right] d\lambda \end{aligned} \tag{2}$$

When $f(x)$ is an exponential ($f(x) = a \times \exp(bx)$) and $\phi(t)$ is a linear function ($\phi(t) = kt + q$), Eq. 2 is reduced to Eq. 3.

$$\begin{aligned} N = \frac{a}{2} \Bigg\{ & e^{b(bDt+x)} \left[1 + erf\left(\frac{2bDt+x}{2\sqrt{Dt}} \right) \right] \\ & - e^{b(bDt-x)} \left[1 + \mathrm{erf}\left(\frac{2bDt-x}{2\sqrt{Dt}} \right) \right] \Bigg\} \\ & + qerfc\left(\frac{x}{2\sqrt{Dt}} \right) + 4kti^2\mathrm{erfc}\left(\frac{x}{2\sqrt{Dt}} \right) \end{aligned} \tag{3}$$

If D is constant throughout the medium, the calculated value of D of the two observers should agree. To demonstrate this, let us consider the parameters of Table 5. Figure 8 shows the diffusant distribution at time $t = t_1$ in the region $[X_0, X_1]$ calculated with Eqs. 1 and 3. It also indicates the relative difference (%) between the two profiles, which is inversely proportional to N and never higher than 4.5 %. If uncertainty on measured N is lower than 4.5 %, it is worth finding a more accurate approximation of $f(x)$ by considering a Gaussian or a polynomial in x.

In spite of the complexity that the function $N = u + v$ might exhibit, the second observer does not need to consider explicitly any effect due to diffusant implantation or the possibility of additional surface processes. Thus, if $f(x)$ and $\phi(t)$ appropriately describe the initial and boundary conditions of the sub-system $X_0 - X_1$ over Δt, it is possible to restrict the area of investigation and identify its local D.

It should be noted that in this example only Eq. 1 is the solution of the diffusion equation, as it describes the diffusant concentration at any point of the whole semi-infinite medium at any time. On the other hand, the analytical expressions $f(x)$ and $\phi(t)$ could depend on the subsystem $X_0 - X_1$ and the interval of time Δt chosen. For example,

Table 5 Parameters used for calculating N with Eqs. 1 and 3

X_0 (u)	X_1 (u)	$t_1 - t_0$ (v)	D (u^2/v)	N (a.u.)	R-Dr (u)	f(x) = a × exp(b × x)	ϕ (t) = k×t + q
90	120	10–0	10	100	37.9–7.1	$a = 9.35$; $b = -0.065$	$k = 0.2333$; $q = 9.156$

if $f(x)$ and $\phi(t)$ are polynomial interpolations, the analytical expression of Eq. 2 is unlikely to be suitable to describe other subsystems of the semi-infinite medium (or a larger subsystem including the region $X_0 - X_1$) or a different interval of time.

If only one parameter of the analytical solution is unknown, the use of the method is particularly fast. For instance, when searching for the best D within the range [10, 100] (u^2/v) with a data set that has an error of 1 % on N, an uncertainty below 5 % is obtained for D in 1 min of training of a network approximating Eq. 3.

As an additional example, let us consider the case with initial concentration N_{in} and $\phi(t) = N_{\text{in}} \times \exp(bt)$ at the boundary $x = X_0 = 0$ (Eq. 4).

$$N = N_{\text{in}}\text{erf}\left(\frac{x}{2\sqrt{Dt}}\right) + N_{\text{in}}\frac{e^{\text{bt}}}{2}\left\{e^{-x\sqrt{b/D}} \times \text{erfc}\left[\frac{x}{2\sqrt{Dt}} - \sqrt{bt}\right] + e^{x\sqrt{b/D}} \times \text{erfc}\left[\frac{x}{2\sqrt{Dt}} + \sqrt{bt}\right]\right\} \quad (4)$$

A simple network was created to search for D in a range of three orders of magnitude and was tested by considering a 1 % error on N. As a result, an uncertainty of 5 % on D was obtained after less than a minute of training. Assuming the same error, a precision of 16 % on D was achieved in an hour of training when searching for D within a range of four orders of magnitude.

This method could also be used when only one diffusion profile at time $t = t_1$ is measured and some parameters in $f(x)$ and/or $\phi(t)$ are unknown. For instance, a network was created to approximate Eq. 4 by considering the case of D in the range [10, 100] and b in the range [0.3, 0.8]. The best result when the error on N is 1 % has been obtained after half an hour of training (maximum uncertainty on D is 2 %).

There are conditions that have to be met before using this method. First, the antiderivatives of the integrands of Eq. 2 have to be found; therefore it is necessary to know what type of functions $f(x)$ and $\phi(t)$ are with reasonable accuracy. Moreover, the term $D \times t$ should be sufficiently small so that the solution using the error function converges (if not, the analogous case using the Fourier series has to be used). In addition, it is assumed that $f(x)$ represents reasonably well the initial concentration in the region beyond the point X_1, where it is either a decreasing or a constant function. Finally, it is assumed that there is no impermeable boundary after the point X_1, so that no reflection occurs. When the last two assumptions cannot

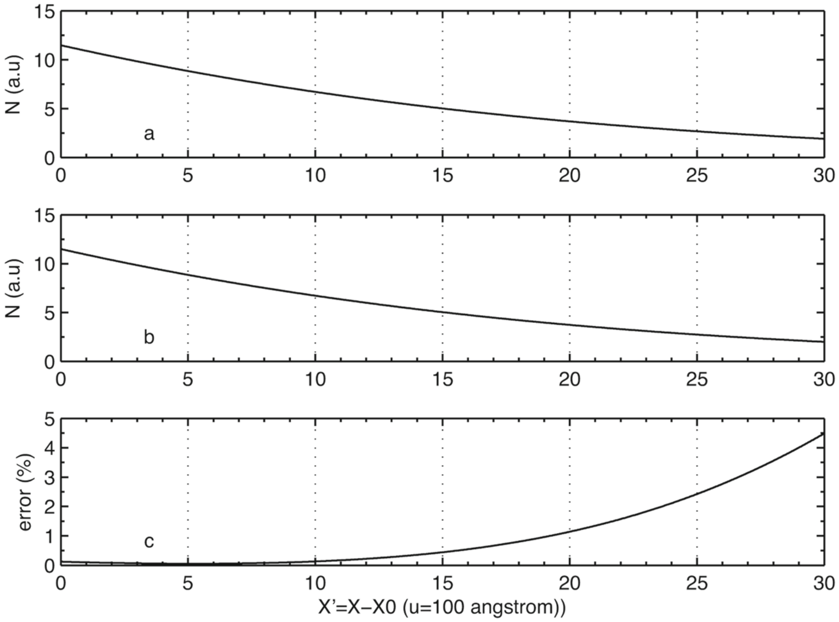


Fig. 8 The diffusant concentration calculated with Eq. 1 (**a**) and Eq. 3 (**b**) with the parameters of Table 5. The relative difference (%) between the calculated profiles is shown in **c**

be made, the analogous solution for a finite medium with initial concentration $f(x)$ and two boundary conditions $\phi_1(t)$ and $\phi_2(t)$ could be considered.

5 Discussion

This section is focused on some of the differences between the method presented in this paper and other computing programmes commonly used to study diffusive processes, highlighting its merits but also its limits.

Nowadays different algorithms, software packages and computer simulations techniques are available to address problems related to diffusion [31–36]. In general, the literature on the subject confirms the clear trend of the preference for numerical methods over the analytical approach [37, 38].

The main quality of this novel method is that it allows one to combine the precision of the analytical approach with the flexibility of a NN. It is important to stress the fact that a MLP will approximate any function, regardless of its complexity, provided the number and location of points of the profile to be measured are chosen correctly and the parameter ranges are reasonably limited. The development of this method has therefore been conceived to facilitate the use of analytical solutions when interpreting diffusion profiles.

The method shares similar goals with studies based on optimization techniques and regression analyses [39–42] but without the need for manipulating formulas, decomposing matrices and performing integration and differentiation in search of the Matano plane. This feature makes the method suitable for scientists or students that do not have a strong mathematical background, giving them the opportunity to compare intuitively complex solutions of the diffusion equation when analysing data sets. Nevertheless, there is also a potential target of authors that might be interested in using the method after adapting existing solutions or defining new ones.

Another characteristic is that a network setup could be selected based on its capability to minimize the instrument specific error. This is because calibration curves could be used to define the noise that is added to virtual data sets before testing for generalization. The goal of undertaking regression analyses by incorporating measurement uncertainties has only recently been accomplished by performing both a linear and a nonlinear regression after eigen decomposition of the matrix associated with multicomponent diffusion [42].

An additional advantage is the chance to identify the uncertainty of the method with any desired detail. Because there is no limit to the size and range of the virtual data set to submit before a real data set, a trained approximator can be tested for generalization as much as required. Once a good generalization is obtained, it is possible to verify that the approximated function describes the system by calculating the diffusant concentrations with the parameters found with the network. If calculated concentrations do not fit the data within the expected uncertainty, the function approximated by the trained NN should not be considered the solution of the diffusion equation.

When entire diffusion profiles are available, it is possible to verify whether the appropriate solution has been identified by comparison of the unknown parameters obtained by applying the method to different subsets. Since the solution of the diffusion equation should be valid for the whole system at any time, results from different subsets should agree. This test is possible because considering a restricted number of points rather than entire diffusion profiles might require longer training times but does not compromise the results in terms of precision.

The main drawback of the method is that it could be time consuming, particularly when there are three or more unknown parameters. Because a good network performance may require a long training time, building the experience to find a good network setup in fewer attempts is certainly desirable. Besides, the success of the method will also depend on the ability to restrict the range of the unknown parameters and/or the possibility of using faster processors.

The heuristic approach for the search of the best function approximator has been preferred in this paper. Alternatively, specific algorithms could be considered to determine a priori the setup for a network [43]. Matlab built-in functions are easy to use but might not be suitable for this purpose and the creation of a completely custom network could be necessary.

6 Conclusions

A novel method using multilayer perceptrons to approximate solutions of the diffusion equation has been presented in this paper. An exercise showing how to refine the value of measured diffusion coefficients from existing data sets illustrated its main steps. The method could be helpful before undertaking diffusion experiments to find the number and locations of points that minimize the error based on the calibration curves of an instrument and the function to be approximated. In some circumstances, it could be applied when the analytical solution of the diffusion equation is not known or does not exist.

Events that are difficult to control during diffusion experiments (undesired chemical reactions acting as sources/sinks in the medium, defect interactions, variable boundary conditions [24]) might be explicitly considered in the analytical solution of the diffusion equation. As a result, corrections of experimental Ds could resolve some disagreements between laboratories [44].

In the future similar function approximators could be used to analyse data sets from geological samples and investigate their thermal histories by approximating the exact analytical solution of Dodson's equation [45].

It is important to point out that the ability of neural networks as universal approximators is not the only useful characteristic that can be considered when analysing diffusion profiles. Networks relatively similar to the ones described in this paper could be thought of as pattern recognizers or adaptive filters performing blind signal separation. The latter might be potentially used to explore coupling effects associated with diffusion, provided time series are available.

Acknowledgments I wish to thank Professor Fernando Fermi (University of Parma) for his encouragement and invaluable comments on the draft of the paper.

References

1. Chen, S., Billings, S.A., Grant, P.M.: Non-linear system identification using neural networks. Int. J. Control **51**, 1191–1214 (1990)
2. Kosmatopoulos, E.B., Polycarpou, M.M., Christodoulou, M.A., Ioannou, P.A.: High-order neural network structures for identification of dynamical systems. IEEE T. Neural Networ. **6**, 422–431 (1995)
3. Demartines, P., Hérault, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. IEEE T. Neural Networ **8**, 148–154 (1997)
4. Hsieh, W.W., Tang, B.: Applying neural network models to prediction and data analysis in meteorology and oceanography. Bull. Am. Meteorol. Soc **79**, 1855–1870 (1998)
5. Nelles, O.: Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer Publishing, New York (2001)
6. Singer, D.A., Kouda, R.: Application of a feedforward neural network in the search for Kuroko deposits in the Hokuroku district. Japan Math. Geol. **28**, 1017–1023 (1996)
7. Van der Baan, M., Jutten, C.: Neural networks in geophysical applications. Geophysics **65**, 1032–1047 (2000)
8. Krasnopolsky, V.M., Schiller, H.: Some neural network applications in environmental sciences. Part I: forward and inverse problems in geophysical remote measurements. Neural Netw. **16**, 321–334 (2003)
9. Ramakrishnan, D., Singh, T.N., Purwar, N., Barde, K.S., Gulati, A., Gupta, S.: Artificial neural network and liquefaction susceptibility assessment: a case study using the 2001 Bhuj earthquake data, Gujarat, India. Computat. Geosc. **12**, 491–501 (2008)
10. Singh, N., Singh, T.N., Tiwary, A., Sarkar, K.M.: Textural identification of basaltic rock mass using image processing and neural network. Computat. Geosc. **14**, 301–310 (2010)
11. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989)
12. Hornik, K., Stinchcombe, M., White, H.: Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Netw. **3**, 551–560 (1990)
13. Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw. **6**, 861–867 (1993)
14. Scarselli, F., Chung Tsoi, A.: Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results. Neural Netw. **11**, 15–37 (1998)
15. Castro, J.L., Mantas, C.J.: Neural networks with a continuous squashing function in the output are universal approximators. Neural Netw. **13**, 561–563 (2000)
16. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Comput. **3**, 246–257 (1991)
17. Liao, Y., Fang, S.C., Nuttle, H.L.: Relaxed conditions for radial-basis function networks to be universal approximators. Neural Netw. **16**, 1019–1028 (2003)
18. Crank, J.: The mathematics of diffusion. Oxford university press, Oxford (1979)
19. Carslaw, H.S., Jaeger, J.C.: Conduction of heat in solids. Clarendon Press, Oxford (1959)
20. Serin, B., Ellickson, R.T.: Determination of diffusion coefficients. J. Chem. Phys. **9**, 742–747 (1941)
21. Le Claire, A.D.: The analysis of grain boundary diffusion measurements. Br. J. Appl. Phys. **14**, 351–356 (1963)
22. Gray, P.G., Do, D.D.: A graphical method for determining pore and surface diffusivities in adsorption systems. Ind. Eng. Chem. Res. **31**, 1176–1182 (1992)
23. Sheikha, H., Pooladi-Darvish, M., Mehrotra, A.K.: Development of graphical gethods for estimating the diffusivity coefficient of cases in bitumen from pressure-decay data. Energy Fuels **19**, 2041–2049 (2005)
24. Fredericks, W.J.: Diffusion in alkali halides. In: Nowick, A.S., Burton, J.J. (eds.) Diffusion in solids, pp. 381–444. Academic, New York (1975)
25. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall PTR, New York (1994)
26. Lau, C.: Neural networks: theoretical foundations and analysis. IEEE press, Piscataway (1991)
27. Demuth, H., Beale, M.: User's guide: neural network toolbox for use with Matlab. The Mathworks Inc, Natick (2009)
28. Cherniak, D.J., Watson, E.B.: Helium diffusion in rutile and titanite, and consideration of the origin and implications of diffusional anisotropy. Chem. Geol. **288**, 149–161 (2011)
29. Ryssel, H., Ruge, I.: Ion implantation. Wiley, New York (1986)
30. Ziegler, J.F., Biersack, J.P., Ziegler, M.D.: SRIM, the stopping and range of ions in matter. SRIM Co, Chester (2008)
31. Engström, A., Höglund, L., Ågren, J.: Computer simulation of diffusion in multiphase systems. Metall. Mater. Trans. A **25**, 1127–1134 (1994)
32. Morral, J.: Computer simulations of NiCrAl multiphase diffusion couples. Acta Mater. **45**, 1189–1199 (1997)
33. Borgenstam, A., Höglund, L., Ågren, J., Engström, A.: DICTRA, a tool for simulation of diffusional transformations in alloys. J. Phase Equilib. **21**, 269–280 (2000)
34. Tully, J.C., Gilmer, G.H., Shugard, M.: Molecular dynamics of surface diffusion. I. The motion of adatoms and clusters. J. Chem. Phys. **71**, 1630–1642 (2008)
35. Limoge, Y., Bocquet, J.L.: Monte Carlo simulation in diffusion studies: time scale problems. Acta Metall **36**, 1717–1722 (1988)
36. Gautheron, C., Tassan-Got, L.: A Monte Carlo approach to diffusion applied to noble gas/helium thermochronology. Chem. Geol. **273**, 212–224 (2010)
37. Hyman, J., Morel, J., Shashkov, M., Steinberg, S.: Mimetic finite difference methods for diffusion equations. Computat. Geosc. **6**, 333–352 (2002)

38. Watson, E.B., Wanser, K.H., Farley, K.A.: Anisotropic diffusion in a finite cylinder, with geochemical applications. Geochim. Cosmochim. Ac **74**, 614–633 (2010)
39. Chen, Y., Choong, E.T., Wetzel, D.M.: Evaluation of diffusion coefficient and surface emission coefficient by an optimization technique. Wood Fiber Sci. **27**, 178–182 (1995)
40. Dayananda, M.A., Sohn, Y.H.: A new analysis for the determination of ternary interdiffusion coefficients from a single diffusion couple. Metall. Mater. Trans. A **30**, 535–543 (1999)
41. Campbell, C.E.: A new technique for evaluating diffusion mobility parameters. J. Phase Equilib. Diff. **26**, 435–440 (2005)
42. Jaques, A.V., LaCombe, J.C.: A stable and efficient regression approach for determination of coefficients in linear multicomponent diffusion. J. Phase Equilib. Diff. **33**, 181–188 (2012)
43. Sartori, M.A., Antsaklis, P.J.: A simple method to derive bounds on the size and to train multilayer neural networks. IEEE T. Neural Networ. **2**, 467–471 (1991)
44. Zhang, Y., Charniak, D.J.: Diffusion in minerals and melts. Rev. Min. Geochem. **72**, 1–1038 (2010)
45. Dodson, M.H.: Closure temperature in cooling geochronological and petrological systems Contrib. Mineral. Petr. **40**, 259–274 (1973)