A single-step time-integrator of a methane–air chemical system using artificial neural networks

Article *in* Computers & Chemical Engineering · November 1999

DOI: 10.1016/S0098-1354(99)00278-1

Javier Blasco
University of Zaragoza
7 PUBLICATIONS 405 CITATIONS

Norberto Fueyo
University of Zaragoza
86 PUBLICATIONS 2,364 CITATIONS

Cesar Dopazo
University of Zaragoza
118 PUBLICATIONS 3,716 CITATIONS

ELSEVIER

Computers and Chemical Engineering 23 (1999) 1127–1133

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

A single-step time-integrator of a methane–air chemical system using artificial neural networks

J.A. Blasco [a], N. Fueyo [a,*], J.C. Larroya [a], C. Dopazo [a], Y.-J. Chen [b]

[a] *Fluid Mechanics Group, CPS/LITEC, María de Luna 3, 50.015 Zaragoza, Spain*

[b] *Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720-1740, USA*

Received 16 October 1998; received in revised form 25 March 1999; accepted 25 March 1999

Abstract

The present paper reports a novel method for embedding a reduced chemical system, suitable for the simulation of methane–air combustion, in an artificial neural network (ANN). The use of ANNs as a means of storing in a compact manner the chemical kinetics of a system is an emerging alternative to other methods, the full potential of which remains to be exploited. The current contribution introduces two novelties: firstly, the compositional domain is split into subdomains, for each of which an ANN fitting is attempted; and secondly, the timestep is introduced as an additional input to the network, thus increasing the accuracy and speed of the method. The paper introduces three alternative types of network, and describes in detail the methodology used for their construction and validation, as well as the validation results. The level of accuracy attained is at least one order of magnitude better than with previously-published ANN approaches. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Reduced chemical system; Artificial neural networks; Chemical kinetics; Combustion; Methane–air systems; Data fitting

1. Introduction

The increasing power and wider availability of computers is fostering the development and use of a host of new turbulent chemical-reaction models and simulation-methods for which the level of description of the thermochemistry has to match the level of detail of the flow description and/or the accuracy of the simulation method. Direct numerical simulation, large eddy simulation or the transported PDF approach are all examples of techniques or models in which an oversimplified chemical description can easily become the weakest link in the numerical simulation chain.

However, the chemistry of even relatively simple combustion systems is exceedingly complex. For methane and air, for example, it involves 32 species and 177 elementary reactions (Frenklach, Wang, Goldenberg, Smith, Golden, Bowman et al., 1995). Although such a detailed mechanism may be known, and rigorously described by means of a system of ordinary differential equations (ODEs), its use in practical multi-dimensional calculations of turbulent combusting flows is impractical, because the integration of the system of ODEs requires too much computer time.

Recently, several techniques to handle chemical systems with various degrees of efficiency and accuracy have been proposed in the literature. Examples of such procedures are the look-up table (LUT) approach (Chen, Kollman and Dibble, 1989), the in-situ adaptive tabulation (ISAT) (Pope, 1997), the intrinsic low dimension manifold (ILDM) method (Maas & Pope, 1992), the repro-modelling (RM) technique (Turanyi, 1994) and the use of artificial neural networks (ANN) (Blasco, Fueyo, Dopazo & Ballester, 1998). Each of these approaches aims at building an accurate representation of the chemical kinetics of a reaction mechanism.

Some of these techniques are based on the storage of a large number of species increments for different points in the chemical-composition space, as is the case of the LUT, ISAT and ILDM methods. The main disadvantage of these approaches is the rapid growth of their memory requirements as the number of species involved increases.

On the other hand, the philosophy adopted by other methods such as RM and ANN is to build compact

[*] Corresponding author. Tel.: +34-976-761000; fax: +34-976-761882.

0098-1354/99/$ - see front matter © 1999 Elsevier Science Ltd. All rights reserved.
PII: S0098-1354(99)00278-1

non-linear models for the species changes as a function of the initial location in compositional space and of the time step.

In this paper, the focus will be placed on exploiting the high non-linear approximating capabilities of the ANN method. Although the application of ANNs to combustion chemistry has been reported in several recent works (Christo, Masri & Nebot, 1996; Blasco et al., 1998), the authors believe that the whole potential of this approach remains still largely unexploited, as will be partly shown in this contribution.

In a previous work by the authors (Blasco et al., 1998), the applicability of the ANN to chemical kinetics integration has been exemplified for a reduced methane–air combustion system. The strategy adopted then was to build neural networks capable of representing the evolution of the entire subdomain of compositional space in which reaction can occur. Given an initial point in compositional space, the ANN predicts the location of the point at the end of a timestep which is fixed for the ANN. Reference Blasco et al. (1998) illustrates how this procedure works for networks for three different time steps (10^{-5}, 10^{-4}, and 10^{-3} s).

With respect to this previous work, the present one introduces several interesting new features. Firstly, time is introduced as an input to the network, which becomes then capable of predicting the evolution of the chemical system for any time step. Secondly, recognizing that the reaction regime can differ widely from one region of compositional space to another, several schemes are devised in which a network is used to represent only a subdomain of the entire allowable compositional domain. It will be shown that a crude division of the domain and a moderate number of networks provide a sufficient degree of accuracy.

The third difference with respect to the previous strategy is the use of increments, instead of absolute values, as the output of the ANNs. The prediction of scalar increments (as opposed to predicting scalar values directly) results in an increase of the sensitivity of the network, and hence an improved accuracy.

In the next sections, the strategy just outlined is explained in more detail, and the results obtained are presented and discussed. Finally, some remarks on the applicability and future improvements of the method are given.

2. ANN modelling approach

2.1. The chemical system

The chemical system considered for the development of the ANN integration system is similar to the reduced methane–air combustion system of Peters and Kee (1987). However, the system employed here has been numerically reduced using the CARM technique of Chen (Chang, 1996) (CARM stands for computer assisted reduction mechanism). The chemical system consists of four steps and seven reactive species:

$$\begin{aligned} &CH_4 + 2H + H_2O \rightleftharpoons CO + 4H_2 \\ &CO + H_2O \rightleftharpoons CO_2 + H_2 \\ &2H + M \rightleftharpoons H_2 + M \\ &O_2 + 3H_2 \rightleftharpoons 2H + 2H_2O \end{aligned} \qquad (1)$$

However, the thermochemical state of the mixture can be fully described by specifying the value of only five variables under the following hypotheses: equal diffusivity of all the species and enthalpy, no heat losses by radiation, constant pressure and ideal-gas density. These five controlling scalars are chosen to be the mixture fraction, and the specific number of moles (or moles per unit mass) of CH_4, CO, O_2, and H. These five controlling scalars will be denoted as f, nCH$_4$, nCO, nO$_2$, and nH.

It is important to remark that these five scalars cannot take independent values. Instead, the choice of each of the scalars fixes the ranges of values within which the following scalars can take values. These hierarchical bounds arise from mixture-fraction and atom-conservation requirements and have been described in detail elsewhere (Chen et al., 1989; Blasco et al., 1998).

2.2. Partitioning of compositional space

One of the novelties introduced in the ANN methodology in the present work is the partitioning of the compositional domain into subdomains, and the use of a dedicated network for each subdomain. The aim of this strategy is ultimately to provide specialized networks for each of the various (and widely differing) combustion regimes that can be found in the entire compositional domain, thus hopefully increasing the accuracy of the representation. Three alternative partitioning strategies have been studied, and these are described next.

Given the relevance that mixture fraction has in determining the combustion regime (and, in fact, in the occurrence of combustion at all), the first partitioning scheme is devised so that each neural network represents the compositional subdomain arising from a single value of the mixture fraction f. A total of 100 mixture-fraction values are taken, distributed between the flammability limits (taken as $0.01 \leq f \leq 0.3$). The points are not distributed uniformly, but the spacing decreases as the stoichiometric value is approached. For each value of f, an ANN is generated in the manner which is described in the next subsection. The ANNs resulting from this scheme will be referred as ANN-1D.

A second strategy results from the generalization of the previous scheme to two dimensions in compositional space (namely to f and $n\mathrm{CH_4}$). A total of 250 pairs of (f and $\mathrm{CH_4}$) values are taken (arising from a 25×10 grid), and an ANN is used to represent the subdomain stemming from each of these points. This strategy will be referred to as ANN-2D.

While providing, in principle, highly specialized ANNs, it must be however pointed out that the use of the above strategies (e.g. in a flow-calculation program) will normally require interpolation; this is so because the initial point for which the evolution of the chemical system is desired will seldom have the f or $n\mathrm{CH_4}$ values for the points at which an ANN is available. In such a case, the normal practice would be to use the ANNs in the neighbouring points (two in the one-dimensional strategy and four in the two-dimensional one) and interpolate the ANNs predictions to the desired point.

A third alternative, which circumvents the need for interpolation, is the training of the ANN for intervals of the composition variables, as opposed to training them for discrete points. A one-dimensional discretization (of the mixture-fraction domain) is explored in this work. Thus, the mixture-fraction range is divided (between the flammability limits) in ten intervals, and each of the resulting compositional subdomains is represented with an ANN. The intervals are not uniform in size, but smaller close to the stoichiometric mixture fraction. This third approach has been termed ANN-I.

2.3. ANN generation

All the ANNs reported in this work are based on the well-known multilayer perceptron network (Haykin, 1994). Fig. 1 is a schematic of a typical network. The network is composed of an input layer, an output layer and a number of intermediate (hidden) layers which interconnect the previous two through the so-called weights. Each neuron i in the hidden and output layers performs the following operation:

$$y_i = f\left(\sum_{j=1}^{N} w_{ij}x_j - \theta_i\right) \qquad (2)$$

where y_i is the neuron output value, f is the transfer function, w_{ij} is the weight connecting neuron i with neuron j in the previous layer, x_j is the output value from neuron j in the previous layer, and θ_i is the neuron bias or threshold. The parameters w_{ij} and θ_i are calculated from the data in the so-called ANN training phase, to be described later.

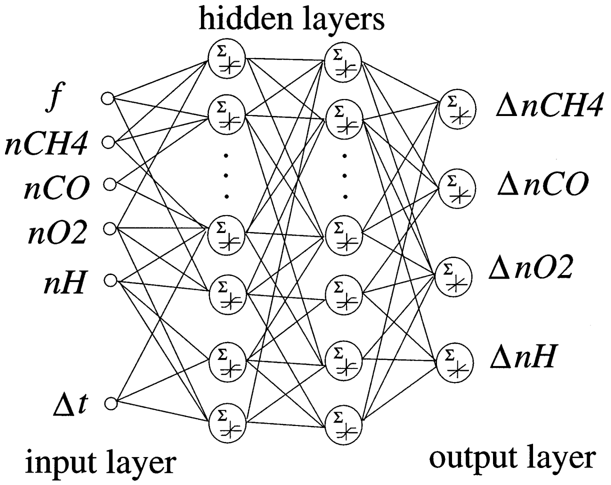

Fig. 1. Schematic of a typical artificial neural network.

The transfer function f used is the hyperbolic tangent for the hidden layers and the identity function for the output layer.

The ANN receives as an input the values of the controlling scalars (f, $n\mathrm{CH_4}$, $n\mathrm{CO}$, $n\mathrm{O_2}$, and $n\mathrm{H}$) and the desired time step (Δt). The input is processed by the network, which provides as output the change in the controlling scalars over the given time step ($\Delta n\mathrm{CH_4}$, $\Delta n\mathrm{CO}$, $\Delta n\mathrm{O_2}$, and $\Delta n\mathrm{H}$). The mixture fraction is an input but not an output, since as a passive scalar it does not change with time.

The internal parameters of the network (the so-called weights) are adjusted in the course of an iterative process termed training. The training process uses an ensemble of pre-calculated input–output pairs which are called the training set. The generation of this set is described next.

The input points for the training set are generated at random within the compositional domain that the network attempts to represent. Thus, in general, a value for the first scalar (e.g. the mixture fraction) is chosen at random within its interval; this value of the mixture fraction determines (see Chen et al., 1989) an interval of allowable values for the next scalar (e.g. $n\mathrm{CH_4}$), in which the value of $n\mathrm{CH_4}$ is chosen at random; the value of $n\mathrm{CH_4}$ generates a similar allowable interval for the next scalar, where a value is again chosen at random; and this cascading process is repeated for the rest of the scalars. The three partitioning strategies differ in the way the first (or the first two) scalars are selected. Thus, ANN-ID represents the allowable space for a single value of f; and therefore f itself is not an input to the network, and the random selection of the training set starts with the second scalar ($n\mathrm{CH_4}$). For the two-dimensional strategy (ANN-2D), each network models the allowable space for a fixed (f, $n\mathrm{CH_4}$) combination, and hence neither of them are input to the ANN. Finally, for the third strategy, in which each ANN is trained for an f interval, f is an input to the network, and it is chosen at random in the f interval that the network must fit.

In all the cases, the time-interval input is chosen at random between 0 and 10^{-4} s.

For each of the input vectors selected in this way, an output vector must be supplied as well. In this work, the output corresponding to an input vector is computed by solving numerically the system of ordinary differential equations which define the evolution of the chemical system (1).

In addition to this training set of data just described, a so-called 'test set' is also generated. The aim of the latter is to measure the performance of the network with data points which are not in the training set.

2.4. ANN numerical details

The actual topology of each ANN (i.e. the number of hidden layers and neurons per layer) is determined at the moment by means of trial-and-error optimization. Thus, for ANN-ID and ANN-2D, a single hidden layer of ten neurons has shown to provide a good fitting. Three hundred examples are used for the training, and 100 for cross-validation (i.e. for the test set).

For ANN-I, the smaller number of ANNs affords the selection of more complex ANN topologies, and the use of larger training and test sets. Two hidden layers with 20 neurons each seem to provide the best results, without resulting in an overly-complex network. Typically, the training set used for the adjustment of the network weights has 2000 examples for each network, and a further 500 additional examples are used for the test set. An advanced training algorithm, namely the scaled conjugate gradient algorithm (Moller, 1993), is used for ANN training in all cases.

3. Results

The results obtained with the ANN methodology will be presented in this section. Three types of results are presented: the overall error incurred into by the ANN approximation of the chemical system; the performance of the ANN in the simulation of plug-flow reactors; and a comparison of the computational performance of the ANN approach with more conventional ones (such as the direct integration of the chemical system or the lookup tables).

3.1. Errors

The selection of a suitable error measurement is an important issue in assessing how well the ANN performs. While the definition of an absolute error is straightforward (as simply the difference between the expected value and the predicted one), the definition of an average relative error poses the problem of choosing appropriate normalizing quantities. In the present work, the following measurement of the relative error is taken:

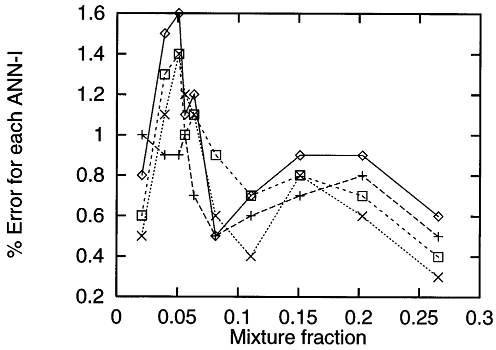

Fig. 2. Error from ANN-I in each of the 10 f-subinterval test sets for the four scalars: CH_4 (diamonds), CO (plus signs), O_2 (squares), and H (crosses).

$$\mathrm{Error}_i = 100\frac{1}{P}\sum_{n=1}^{P}\frac{|\Delta y_{i,n}^{\mathrm{tar}} - \Delta y_{i,n}^{\mathrm{ANN}}|}{\left(\max_n(\Delta y_{i,n}^{\mathrm{tar}}) - \min_n(\Delta y_{i,n}^{\mathrm{tar}})\right)} \qquad (3)$$

where the error is averaged over the P examples (also termed patterns) of a set, $\Delta y_{i,n}^{\mathrm{ANN}}$ is the ANN-predicted increment of scalar i and example number n, and $\Delta y_{i,n}^{\mathrm{tar}}$ is, in this equation and in the remainder of this paper, the target value, obtained via the numerical integration of the chemical system (1) using an ODE solver.

The best-performing strategy turns out to be ANN-I. The errors obtained with ANN-I using Eq. (3) have been represented in Fig. 2, where the errors for each of the ten ANNs modelling the ten subintervals in which the f interval is divided are presented with different symbols for the various species. The error presented is the average one for the 500 data points in the test set of each scalar (i.e. the data which have not been used to train the ANNs); the corresponding error for the training set (not shown) is smaller for all values of f, the peak relative error being 1.1%. The figure also reveals very clearly how the errors are larger around the stoichiometric mixture fraction, undoubtedly as a consequence of the higher chemical activity in that zone. This is the reason why the partitioning of the f interval has been chosen to be finer around the stoichiometric mixture fraction, while larger subintervals are used near the extremes of the f interval. The overall average error (for all the networks, scalars and patterns) is 0.64%.

The overall error for the ANN-I approach proposed here can be compared with the one from a previous ANN approach (Blasco et al., 1998), which uses a single fixed-timestep ANN for the entire computational domain. Using the error measurement reported in (Blasco et al., 1998), the ANN-I would provide an overall error of 0.47%, while the previous approach yielded an overall error of 4.9%. Thus, the fitting errors have been improved with the current ANN strategy by an order of magnitude, even though the timestep was fixed in the previous approach and variable in the present one.

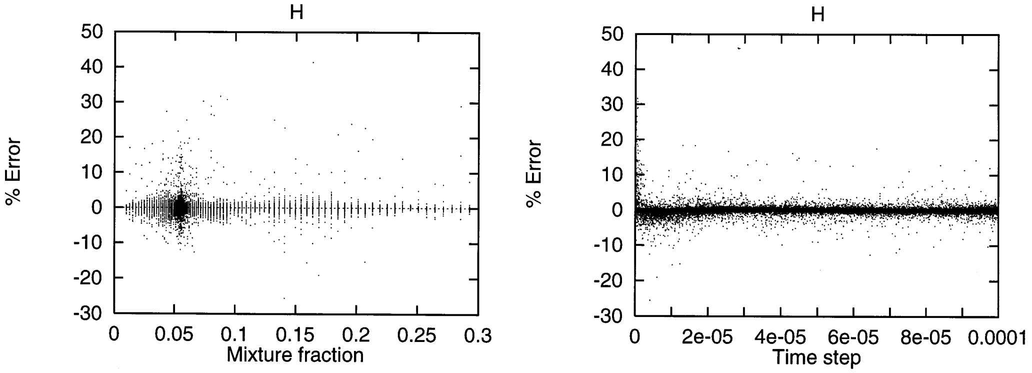

Fig. 3. Scatter plot with relative error for all the training examples of the *n*H scalar against mixture fraction and timestep size (*s*).

ANN-ID and ANN-2D provide slightly larger errors, the overall relative error being respectively 1.68% and 1.4%. On this account, only the results from ANN-I will be reported from now on.

Fig. 3 contains two scatter plots of relative errors (for the training data-points) for one of the species (the *H* radical) as a function of the mixture fraction and the time interval. The data set encompasses all ten ANNs. The mixture-fraction dependence shows the same trend as described above (i.e. larger errors for values of *f* near the stoichiometric one). The time-interval dependence shows that the error is nearly independent of the time-interval to be predicted, except for very small time intervals where the error is larger. This was indeed expected, since the atomic hydrogen has very small creation and destruction time scales.

A final report of the overall error is shown in Fig. 4, which includes the correlation between ANN-I results and actual values for the increments of all four reactive variables in a large set (4000 values) of fresh data randomly chosen in the entire compositional space and in the timestep interval $[0, 10^{-4}]$ s. These data have been chosen at random, and have not been used in training or test.

3.2. Plug-flow reactor (PFR) simulations

An additional test of the predictive performance of ANN-I can be designed around the PFR idea. The test is carried out at follows. First, an initial point in compositional space is chosen at random. Then, the system of ODE is integrated with respect to time from that point. The graphical result of this integration is a curve for each of the scalars representing their evolutions with time. Then, the system of ANNs is called upon to perform the same prediction, and the results are compared. Fig. 5 presents the result of such an exercise for three different initial, randomly-chosen points in compositional space. (The three initial points are chosen out of a large set of similar comparisons as representative of lean, nearly-stoichiometric and rich mixtures. Around 1000 comparisons of this type have been carried out, all with similar results to the three sample ones shown in Fig. 5). In analysing Fig. 5, it is important to note that each ANN prediction (represented with a symbol) is calculated ab initio, i.e. by supplying as input to the ANN the *initial* composition and the time elapsed from $t = 0$, because it is thought that the quality of the ANN fitting is best put to test by performing plug-flow calculations in which the ANN is asked to predict the species evolution by advancing from a fixed initial state and using an increasing timestep size.

The results presented in Fig. 5 show that the ANN technique is able to provide very accurate predictions for the CH_4 and O_2 concentrations irrespectively of the timestep prescribed or the mixture fraction value. This behaviour is also observed for the CO and H concentration values, with some discrepancies in the nearly-stoichiometric case.

3.3. Computational performance

The training of each of the ANN-Is takes about 24 h of CPU time on a Pentium PC at 133 MHz running Linux and with the GNU gcc compiler.

Once trained, the use of the ANN-I's in an actual chemical-reaction calculation is of course much faster, since it involves only the processing of the input data through three vector-times-matrix multiplications. Furthermore, the strategy used in the ANN-I avoids the need both for interpolating (in scalar space) and cycling (in time) in order to reach a solution for a given initial composition and timestep.

A comparison of the CPU performance and storage requirements of the ANN-I method and direct integration of the ODE (on one hand) and tabulation of the pre-calculated chemistry shows that the ANN-I method is much faster than direct integration and requires much less RAM space than the look-up table.

A detailed comparison of CPU times indicates that ANN-I is on average 2750 times faster (for 4000 ran-

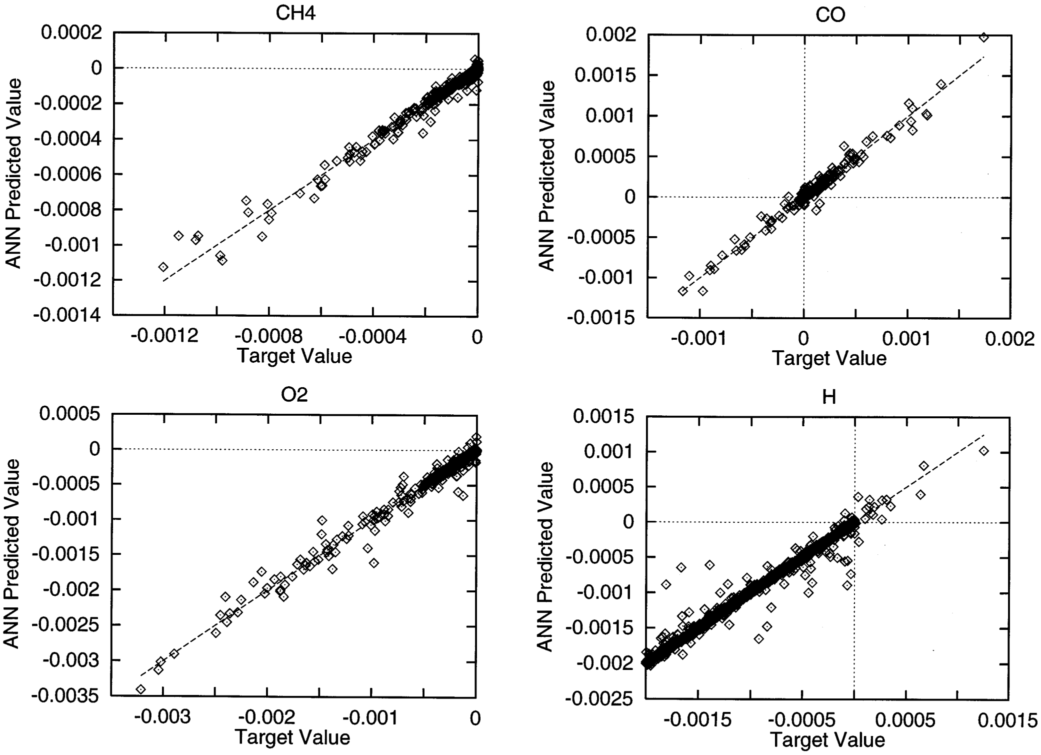

Fig. 4. Correlation plots between scalar increments predicted by ANN-I and calculated by direct integration for 4000 brand new examples not used during training or test.

domly-chosen input data-set) than the direct integration of the chemical system using LSODE (Hindmarsh, 1980). The storage requirements for five scalars are roughly 1000 times smaller than for the LUT (an exact comparison is not possible, since ANN-I provides results for any time step, while the LUT is calculated for a single one. The figure reported is for three LUTs for four different timesteps).

Furthermore, the CPU time required by ANN-I does not depend on the timestep for which a solution is required nor on the local stiffness of the ODE system; and, most importantly, the storage requirements are expected to grow only moderately as the number of input scalars is increased. As a corollary, the method has a great potential for systems with a large number of scalars, for which the storage requirements for the LUT and the CPU-time consumption of direct integration are prohibitive.

3.4. Conclusions and future work

The present paper has introduced a method to economically represent the temporal evolution of a reduced chemical system with an ANN. The method is based on the partitioning of the chemical space into subdomains, and the training of an ANN for each subdomain. This new strategy provides much better accuracy than previously-published ANN approaches, even though the elapsed time is now an input variable to the system. Furthermore, the ANN predictions are performed with low CPU and memory requirements compared to alternative methods, thus allowing the inclusion of the ANN approach in other computer-intensive simulation techniques.

Work is now under progress to further increase the accuracy of the predictions. The use of a (larger) number of (relatively simpler) networks, each of them focused on a chemical subdomain, seems to be a promising approach. The authors are working on an automated way of identifying such subdomains, based on the strength of another type of artificial neural network (the self-organizing map, or SOM) as a data classifier. This would additionally extend the methodology to chemical systems in which it is not possible to find ‘a priori’ a characteristic property (such as the mixture fraction in this case) from which chemical subdomains can be generated.

Acknowledgements

The authors would like to thank the European Union for support under BRITE-EURAM contract BE95-1927, and the Science and Technology IBERDROLA Visiting-Professors Programme for supporting financially the stay of one of the authors (JYC) at the

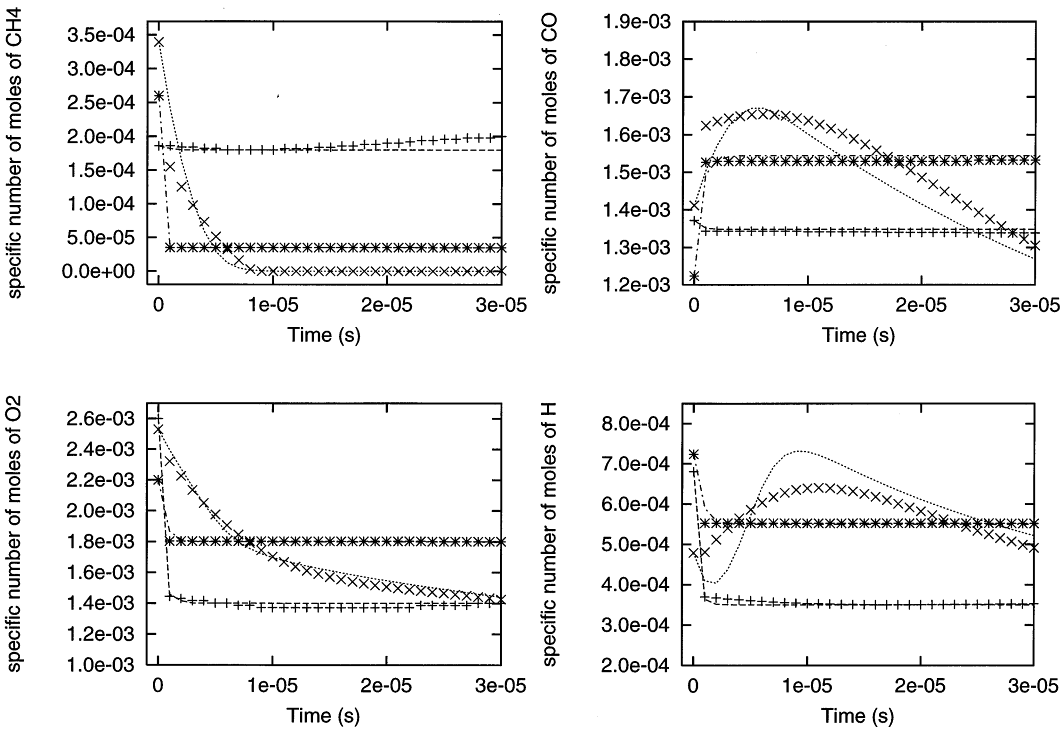

Fig. 5. Temporal evolution of the scalar values from direct integration method (line) compared to ANN-predicted values for a lean mixture ($f = 0.010$) (diamonds), near-to-stoichiometric mixture($f = 0.052$) (plus signs) and a rich mixture ($f = 0.27$) (squares). (The ordinates of the lean and rich evolutions have been re-scaled so that the three evolutions fit in the same graph).

University of Zaragoza. JAB and JCL have Doctoral Fellowships from the Spanish Ministry of Education and Culture and the Regional Government of Aragon, respectively.

References

Blasco J. A., Fueyo, N., Dopazo, C., & Ballester, J. (1998). Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network. Combustion and Flame *113* (1/2), 38–52.

Chang, W.-C. (1996). Modeling of NOx formation in turbulent flames. Development of reduced mechanisms and mixing models. PhD thesis, Department of Mechanical Engineering, UC Berkeley.

Chen, J.-Y., Kollmann, W., & Dibble, R. W. (1989). PDF modelling of turbulent nonpremixed methane jet flames. Combustion Science and Technology *64*, 315–346.

Christo, F. C., Masri, A. R. & Nebot, E. M. (1996). Artificial neural network implementation of chemistry with pdf simulation of H_2/CO_2 flames. Combustion and Flame *106*, 406–427.

Frenklach, M., Wang, H., Goldenberg, M., Smith, G. P., Golden, D. M., Bowman, C. T., Hanson, R. K., Gardiner, W. C. & Lissianski, V. (1995). GRI-mech. an optimized detailed chemical reaction mechanism for methane combustion. Topical Report GRI-95/0058, Gas Research Institute

Haykin, S. (1994). *Neural networks. A comprehensive foundation*. New York: Macmillan.

Hindmarsh, A. C. (1980). LSODE Livermore solver for ordinary differential equations. Mathematics and Statistics Section, L-300 Lawrence Livermore Laboratory.

Maas, U. & Pope, S. (1992). Simplifying chemical kinetics: intrinsic low-dimensional manifolds in composition space. Combustion and Flame *88*, 239–264.

Moller, M. (1993). A scaled conjugated gradient algorithm for fast supervised learning. Neural Networks *6* (4), 525–533.

Peters, N., Kee, R. J. (1987). The computation of stretched laminar methane air–difussion flames using a reduced four-step mechanism. Combustion and Flame *68*, 17–29.

Pope, S.B. (1997). Computationally efficient implementation of combustion using in situ adaptive tabulation. Combustion Theory Modelling *1*, 41–63.

Turanyi, T. (1994). Parameterization of reaction mechanisms using orthonormal polynomials. Computers in Chemistry *18* (1), 45–54.