Modelling the Temporal Evolution of a Reduced Combustion Chemical System With an Artificial Neural Network

J. A. BLASCO, N. FUEYO,* C. DOPAZO, and J. BALLESTER
Fluid Mechanics Group, CPS/LITEC María de Luna 3, 50.015 Zaragoza, Spain

The present work introduces a way of embedding a combustion chemical system in a neural network, in such a way that it can be used, with considerable CPU time and RAM memory savings, in fluid-flow-simulation codes. The system is composed of four neural networks, with three of them simulating the evolution of the reactive species and one providing density and temperature as a function of composition. The performance in terms of accuracy of the networks is assessed by comparison with the results of the direct integration of the thermochemical system for a large number of random samples. Error measurements are reported, and sample evolutions of the chemical system with both methods are compared. It can be summarized that the results of this exercise are satisfactory, and the CPU-time and memory savings encouraging.

INTRODUCTION

Motivation

In the numerical simulation of fluid flow, it is often required to use information concerning the temporal evolution of a chemical system. This is the case, particularly, in the simulation of laminar, chemically-reacting flows, of Direct Numerical Simulation [1], and of certain models of turbulent chemical reaction (such as the transported PDF approach [2] or the Eddy Dissipation Concept [3]).

This temporal-evolution information is normally supplied by a system of ordinary differential equations, generally of the form:

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = \dot{\omega}_i(\boldsymbol{y}, T, P), \tag{1}$$

where $\dot{\omega}_i$ is the chemical reaction rate of species i, $\boldsymbol{y}$ is the mass fraction vector, T is the temperature, and P is the pressure. Two additional equations of state supply the temperature and density for a given thermochemical state.

The equations represented by Eq. (1) constitute normally a stiff system, which exhibits a wide spectrum of characteristic evolution timescales. The integration of Eq. (1) requires specialized solvers [4], which are in general computer-time intensive.

Since most numerical methods for the calculation of fluid flow (e.g., those mentioned above) are by themselves very demanding in computer time and require many integrations of the system of Eq. (1), a direct integration of the flow and the ODE solvers, is in general impractical.

Most numerical methods resort then to a pre-calculation of the thermochemistry, which is stored in the form of a table where the values of the thermochemical variables at the end of a given time step are recorded as a function of the thermochemical state at the beginning of the time step. Although the thermochemical state is (coarsely) discretized for the tabulation, the storage requirements quickly grow as the dimensions of the compositional space increase. Efficient tabulation methods, such as that proposed by Chen [5] or by Pope [6], somehow alleviate the problem without ultimately solving it.

In contrast, Artificial Neural Networks (ANN) emerge as a general, compact, and fast way of modelling complex dynamical systems, of which a chemically-reacting one is a good example. ANNs [7] are a recent data processing tool, the structure of which are inspired by the animal nervous system. ANNs are achieving good results in fields where classical programming procedures have failed, such as complex system modelling, image and voice recognition, nonlinear control, pattern classification, and forecasting [8].

The main feature of ANNs is their ability to model complex, non-linear problems simply by

*Corresponding author.

0010-2180/98/$19.00
PII S0010-2180(97)00211-3

Published by Elsevier Science Inc.

being presented with examples, or sets of input-output patterns. These patterns may include noisy and redundant items.

An ANN consists of interconnected layers of non-linear processing elements, which resemble biological neurons. The network stores the information in the strength of the neuron interconnections through so-called weights. These weights are adjusted during an iterative process called a learning phase in which all the examples are presented to the ANN repeatedly.

After this learning phase, the ANN can be used to obtain the answer to an input pattern (possibly not present in the training set) by processing it through its layers of neurons and is therefore ready to be embedded in the simulation program. This is called the recall phase. Using these techniques, good results can be achieved with a low processing time and small memory resources.

Aim and Scope of the Present Work

The aim of this work is to show how neural networks can be used to represent a thermochemical system given by equations such as Eq. (1). Specifically, the system chosen is a reduced chemistry set for methane combustion. The system will be simulated with three neural networks for three different reaction time steps, plus an additional one to supply temperature and density as a function of composition.

The bulk of this work is geared toward showing the performance, mainly in terms of accuracy, of the neural network representation of the chemical system compared with the "exact" (or, properly expressed, numerical) solution of the system of ODE. The simulation errors are reported and analyzed, and the exact temporal evolutions of the system are compared with the same evolutions as predicted by the ANNs. An indication of the savings in computing resources (time and memory) is also given.

Literature Survey

Although the use of ANNs is widespread in many other areas [9], their application to the modelling of chemical systems has remained, as far as the authors know, a largely unexplored realm. The only exception to this outlook is the recent work by Christo et al. [10, 11], which has been to a great extent concurrent with the present one.

In [10], Christo et al. report the application of a neural network to the modelling of a single-step chemical-system and the integration of the ANN with a PDF (Monte Carlo) solver.

In [11], the system modelled is more complex, namely, a systematically-reduced H_2-CO_2 system, consisting of three steps and four controlling scalars. The ANN is applied to the modelling of a turbulent H_2-CO_2 flame using a transported-PDF (Monte Carlo) approach. The results are compared with the classical tabulated-chemistry approach in terms of memory and CPU performance. However, as far as accuracy is concerned, no detailed account of the error is reported and only a global comparison of peak values of selected variables provided by the network and the classical look-up table method is shown in the article. Although an error function is defined (as simply the sum of quadratic errors) and used to measure fitting errors, it can be argued that some normalization of the error is needed since the approximated quantities (the mass fractions) display a wide disparity of scales.

The present contribution differs from the two previous ones in several respects: first, a methane system is considered; second, an ANN is also used to replace the look-up table for density and temperature; and third, the emphasis is placed in analyzing the network error, comparing its predictions with the direct numerical integration of the ODE system. With respect to the latter, it is thought that the study of this error in isolation is a cautious, mandatory step before addressing the task of integrating the ANN with a flow simulation system; the latter is a more complex environment in which isolating the several sources of errors as well as attributing unequivocally causes to them are not trivial matters.

THE REDUCED CHEMICAL MODEL

The chemical system considered for the present application is the reduced methane-air combustion system of Peters [12]. The system has been used previously by other authors (e.g., [5] and [13]) and will only be described here insofar as

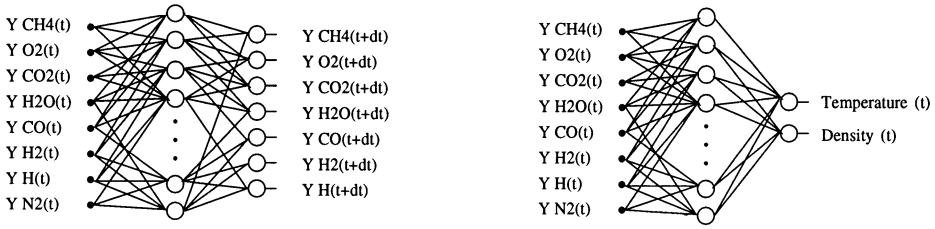

Fig. 1. a: General Artificial Neural Network structure showing inputs and outputs used to approximate the reactive-species mass-fractions. b: General ANN structure for temperature and density evaluation.

the description is relevant to the present application.

The reduced model is derived from a complete kinetic scheme [13] having 18 steps and 13 reactive species. The reduction process proceeds by assuming steady state for six chemical species (namely, CH_3, CH_2O, CHO, HO_2, O, and OH) and partial equilibrium for two reactions. The resulting reduced system has only four steps and seven reactive species (CH_4, O_2, CO_2, H_2O, CO, H_2 and H), and an inert one (N_2):

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

Assuming equal diffusivity of species and enthalpy, no heat losses by radiation, constant pressure, and ideal-gas density, the thermochemical state of the mixture can be fully described by five scalars, which are taken to be the mixture fraction, the specific number of moles (moles per unit mass) of CH_4, CO and H, and the total number of moles per unit mass.

THE ANN MODEL OF THE CHEMICAL SYSTEM

Aim and Strategy

The aim of the model is the generation of two types of ANN, one representing the temporal evolution of the reactive chemical species and another one for the temperature and density of the mixture as a function of the chemical composition of the system. These are shown schematically in Fig. 1.

The aim of the reactive-species ANN is to predict, given a composition at the beginning of a time step, the mass fractions of the species at the end of this time step. The time step is fixed for the network, and thus three different ANNs are used (for $\Delta t = 10^{-5}$, 10^{-4} and 10^{-3} s) to cover the span of timescales which are encountered in a typical combustion simulation. Of course, intermediate time steps can be simulated by dividing them into smaller ones having the above sizes.

The goal of the temperature-density ANN is to model the time-independent relationship between these quantities and the composition (mass fractions) of the system. Hence, a single ANN suffices.

The independent variables chosen as the inputs to both ANNs are the set of eight mass fractions. Alternatively, the five independent controlling scalars could be used as input and output for the reactive-species ANN, which has the advantage of ensuring the conservation of the number of element atoms. However, when this was done in the present work, higher errors were attained. This fact could be attributed to a more complex functional relationship between the input and output for the controlling scalars than for the mass fractions.

The four steps involved in the generation of an ANN are example generation, scaling, training, and error measurement. These steps are dealt with in the next subsections.

Example Generation

In order to compute the ANN weights, a set of examples containing input-output data must be generated. For the reactive-species ANNs, this generation is carried out by choosing at random a set of initial values for the composition and

integrating the system over the corresponding time interval using an ODE-integration package.

The numerical integration of the thermochemical system involves the four chemical steps indicated in the previous section and the eight participating species. It has, however, been established that only five scalars are linearly independent, and that these will be taken as the mixture fraction (f), the specific number of moles (moles per unit mass) of CH_4, CO, and H, and the total number of moles per unit mass.

For these five controlling scalars, the starting values for the integration of an example are chosen at random within an allowable compositional domain. The range of allowable values for each variable is determined using the hierarchical bounds contrived by Chen [5], which is applicable to non-premixed flames. It must be noted that these bounds limit a subdomain of the compositional space, which is still larger than, but nevertheless contains, the one actually accessed in flame calculations. The bounds are described in more detail in the Appendix.

Once the values of the controlling scalars have been determined, then the mass fractions of the rest of the chemical species can be obtained by applying atomic species conservation.

For the neural network to adequately identify (and respond to) points in compositional space in which a time evolution is taking place and points in which the system is in steady state, the training data set has to provide an adequate balance between these two situations. In the course of this investigation, it was soon noticed that the unfiltered choice of random training samples resulted in a bias of the training set towards the steady-state situation (the reason being, of course, the narrowness of the reacting zone in compositional space). The strategy adopted to solve this problem was to qualify the initial data point prior to its admission into the data set. The qualifying criterion is to require that Y_{CH_4} changes in the first time step by a given percentage, as a way of ensuring that the initial point is chemically active. Thereafter, the chemical system is integrated through a number of intermediate steps to (nearly) steady state. Thus, for instance, for the 10^{-5} s ANN, 15 time steps (of 10^{-5} s) are taken, all of which contribute to the input-output training set for this ANN. For the 10^{-4} s ANN, five time steps are taken and two for the 10^{-3} s ANN.

This process has proved (empirically) to yield good results. But it must be recognized that the performance of the ANN is clearly very sensitive to the quality of the selected training set, and that the influence of these parameters must be further studied.

For the density-temperature ANN, a similar (but simpler) approach is used. For the five given values of the controlling scalars, the eight species mass fractions are computed. Then the thermochemical database of CHEMKIN [14] is used to determine temperature and density, which are then recorded as the output part of the input-output pair. The selection of data samples is for this ANN a purely random process within the allowable domain, without any data qualification.

Scaling

Once the training data-set has been selected, some preprocessing must be applied to the input and output values before they can be fed to the ANN for training. Two successive linear transformations have been performed on the data, in which the input and output values are treated separately. These transformations are

(i) Standardization: the data are transformed to achieve zero mean and unity variance:

$$x_i' = \frac{x_i - \overline{x_i}}{\sigma_{x_i}}, \tag{3}$$

where x_i is the value of input (or output) i, $\overline{x_i}$, and σ_{x_i} are the average and standard deviation of all the training-set values for input (or output) i, respectively, and x_i' is the standardized value corresponding to x_i.

(ii) Range adjustment: after standardization, the values can still lie outside the $[-1, +1]$ interval, which the ANN theory shows to be the optimal working range. Hence, a linear transformation is applied to map the data from the range resulting from (i) to $[-1, +1]$:

$$x_i'' = -1 + 2\,\frac{x_i' - \min\{x_i'\}}{\max\{x_i'\} - \min\{x_i'\}}, \tag{4}$$

where $\max\{x_i'\}$ and $\min\{x_i'\}$ denote the maximum and minimum of the standardized values of input (or output) i within the training-set data, and x_i'' is the x_i' value adjusted to the range $[-1, +1]$.

Training

In the training phase, the scaled input and output samples of the generated examples are presented to the ANN and a procedure to adjust the ANN weights is followed. This procedure is a second-order scaled-conjugate-gradient method [15], which has proved to be more robust and faster than the conventional and widespread backpropagation learning-algorithm. It is also free from adjustable convergence parameters.

The topology or architecture of the feedforward ANN can be described in terms of the number of intermediate (or hidden) layers of neurons and the number of neurons in each layer. The ANN architecture is to be chosen in a trial-and-error process to maximize the quality of the fitting of the data set.

If the number of adjustable parameters in an ANN (i.e., the number of weights) is too large, then the ANN provides a very accurate answer for the data which has been used for training, but it yields poor results for unseen data. This is called overfitting. To avoid overfitting, the performance of the ANN is measured during the learning phase using a separated set of examples called the test set. This procedure is called cross-validation.

Accordingly, several ANN architectures have been trained for the present problem, and the performance of each of them has been measured. The results of this exercise are reported in the next section.

Error Measurement

Some measurement of the ANN performance must be chosen in order to evaluate how accurately the ANN is approximating the training and test data. For a given ANN, the measurement of the (percentage) error incurred by the ANN has been defined as

$$\text{Error} = 100\,\frac{1}{PO}\sum_{n}^{P}\sum_{i}^{O}\frac{|y_{i,n}^{\text{tar}} - y_{i,n}^{\text{ANN}}|}{\langle y_i^{\text{tar}}\rangle} \tag{5}$$

where P is the number of patterns (examples) within a data set, O is the number of output variables of the ANN, and $y_{i,n}^{\text{tar}}$ is the target value for output i and pattern n. Target values are time-evolved mass fraction values calculated by the integration package and temperature-density values computed by the thermochemical package. $y_{i,n}^{\text{ANN}}$ is the value provided by the ANN for output i and pattern n and $\langle y_i^{\text{tar}}\rangle$ is the mean target value for output i: $1/P\ \sum_n^P y_{i,n}^{\text{tar}}$. With this definition, the error $|y_{i,n}^{\text{tar}} - y_{i,n}^{\text{ANN}}|$ is compared to the mean of the output target $\langle y_i^{\text{tar}}\rangle$ and expressed as a percentage.

RESULTS

The results of the ANN models for the methane-air combustion system are presented in this section. The presentation is divided into four subsections. First, the results of the search for the optimal ANN architecture are reported, and an optimal architecture is selected. Then, the performance of this architecture is assessed through the examination of the errors, including individual output errors, error distribution, and predicted against target values. In the third subsection, attention is turned to the predicted evolution of the chemical system from a given initial point. And finally, a brief study is included with the computational performance of the ANN compared with direct numerical integration and tabulated chemistry.

ANN Architecture Selection

The type of ANN chosen for this work is the Multilayer Perceptron (MLP) [7]. It is a supervised feedforward neural network, the architecture of which can be summarized as follows:

— Type of processing element. Each unit performs a dot product between the unit's weight vector and the vector formed by the outputs of previous-layer units. Then, an activation function is applied to the result of the dot-product.
— Type of activation function. A sigmoidal

function with slope $\beta = 1.0$ and saturation in -1 and $+1$ is employed.
— Type of unit connectivity. Every unit is connected to all the units in the immediately-following layer and no other connections exist (e.g., there are no connections between units of the same layer, between units of non-adjacent layers, or backward connections).
— Number of inputs and outputs. This number is given by the modelling strategy adopted in our work, which has been explained in a previous section. The number of inputs/outputs is 8/7 for the reactive-species ANN and 8/2 for the temperature-density ANN.
— Number of hidden layers and hidden units: this is the only aspect that has still to be specified, and it will be the matter of discussion of the rest of this section.

The tests for the selection of the optimal ANN architecture are all based on the same four ensembles of examples, three for each of the reactive-species ANNs and one for the temperature-density ANN. Each of these ensembles is split into two sets, one of which is used for training and the other is reserved for test. Each set is composed of 2000 input-output pairs. This number is judged to be large enough to be representative of the range of the thermo-chemical states within the system.

In order to select the optimal architecture, both the test-set error and the ANN size should be taken into account. The first provides a measure of ANN future simulating errors, whereas the second determines the CPU time required for the ANN to process the input data to yield an answer.

For the reactive-species ANNs, an increasing number of hidden units and hidden layers has been tried, ranging from one hidden layer of only 5 units to two layers of 20 units each.

The errors of the three time step ANNs are shown in Table 1. These errors are calculated by applying Eq. (5) separately to the training set (first figure) and test set (second figure) of each time-step data-set. As indicated by Eq. (5), errors of individual outputs are averaged to obtain a single measure for each trial. Individual output errors will be presented in the next section for the optimal ANN.

TABLE 1

Training-Set Error (First Figure) and Test-Set Error (Second Figure) for the Three Time Step (in s) ANNs as a Function of the Number of Hidden Neurons

	Hidden units					
Step	5	10	20	40	10×10	20×20
10^{-5} s	7/7	4/5	3/4	3/4	3/4	2/3
10^{-4} s	15/17	7/9	6/8	5/7	5/7	3/5
10^{-3} s	21/23	15/17	12/14	10/12	9/11	4/7

Table 1 shows that the error in all cases decreases as the number of hidden neurons is increased. It also shows that, as the time step increases, a more complex ANN is needed for the same error to be attained. This is undoubtedly a reflection of the fact that the relationship between input and output mass-fractions is simpler for shorter time steps.

Trials have been performed also for ANNs with 30×30 and 50×50 neurons, but the improvement in error was very small; in some cases the results attained with larger ANNs were worse than those obtained for the 20×20 ANN. The reason for this small or null improvement is thought to be that the error surface (which has to be minimized by the conjugate gradient algorithm) turns much more complicated as the number of free parameters (weights) increases, which implies that the minimization algorithm is more likely to be trapped in local minima of the error rather than reaching the ideal global minimum.

In nearly all cases, the test error is slightly higher than the training error. This is the usual behaviour of any ANN. Should this difference be significant, it would indicate that the training set is not representative of all the possible cases.

The foregoing analysis permits to conclude that the optimal ANN for the three time steps is a two-hidden-layer ANN with 20 units in each layer, the results of which will be further analyzed in the remainder of this section.

Similarly, several ANNs have been trained with temperature and density data, and the converged ANN errors [calculated with Eq. (5)] are shown in Table 2 for the training and test sets. Since the relationship between the eight mass fractions and the corresponding temperature and density is not too complicated, the

TABLE 2

Training-Set and Test-Set Errors for the Temperature-Density ANN for Different Number of Hidden Neurons

Hidden units		
5	10	20
0.76/0.77	0.57/0.55	0.45/0.43

number of hidden neurons that needs to be employed is smaller compared with the reactive-species ANNs. A lower error is consistently attained as the number of hidden neurons is increased; and training and test set errors show nearly identical values.

Since the errors shown in Table 2 are already acceptable, larger ANNs were not considered and the optimal ANN selected is a one-hidden-layer ANN with 20 units. Its associated simulation results are presented in the next section.

Detailed ANN-Performance Results

The optimal network errors are further analyzed in this section. The analysis is largely based on the investigation of the individual errors associated to each network output.

TABLE 3

Errors [Computed With Eq. (5)] for the Optimal ANN in the Prediction of Mass Fractions in the Training Set (First Row) and Test Set (Second Row) for Each Time Step (in s)

Step (s)	Chemical species						
	CH_4	O_2	CO_2	H_2O	CO	H_2	H
10^{-5}	4.4	2.1	0.6	0.5	0.5	0.8	3.0
	5.7	3.7	0.9	0.9	0.9	1.5	5.7
10^{-4}	7.2	4.8	1.4	0.9	1.3	1.4	6.3
	13.1	6.8	2.1	1.3	2.0	2.0	9.7
10^{-3}	9.6	6.7	2.1	1.1	1.8	1.7	8.6
	14.4	9.9	3.3	2.0	2.7	2.4	16.0

These errors are presented in Table 3 for the seven mass fractions and for each of the three time steps. The results are good both for the training and the test sets. Some species (CO_2, H_2O, CO, and H_2) are predicted with considerable accuracy (less than 1%, 2%, and 3% test-set errors for 10^{-5}, 10^{-4}, and 10^{-3} time steps, respectively), whereas others (CH_4, O_2, and H) are predicted with a lower precision (errors under 5%, 10%, and 15% for 10^{-5}, 10^{-4}, and 10^{-3} time steps, respectively). This larger error is due to these species having a smaller characteristic time, and therefore a much higher reaction rate.

The error distributions show good properties, being symmetrically distributed around zero (i.e., unbiased), for all the species and time steps. In Fig. 2, the error distributions of CO_2 for 10^{-5} and 10^{-4} s are shown as an example. As the time step increases, the values of the mass fractions at the end of the time steps will also (in general) increasingly differ from the initial value. This in turn implies a more complex relationship between initial and final species concentrations, which results in a small broadening of the histogram of errors for larger time steps, as shown in Fig. 2.

The graphs in Fig. 3 contain two types of information. The bars indicate the number of output targets for each range of output values and the lines indicate the average error incurred into by the ANN in each of these intervals. The figure reveals that the errors are, in general, reasonably uniform across the range of output values. It also shows that the error in an output

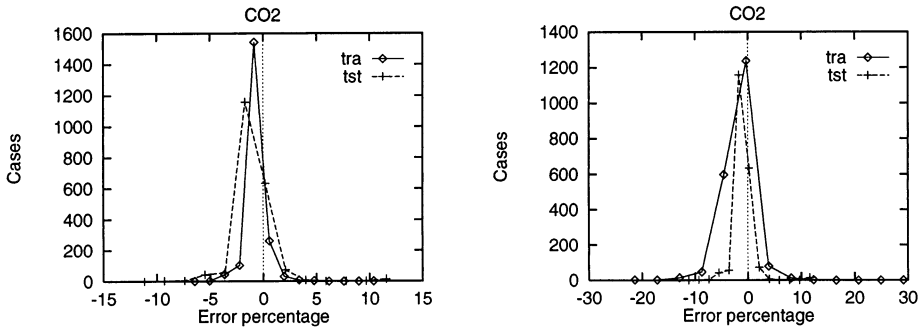

Fig. 2. Distribution of the training and test-set errors in the ANN-prediction of CO_2 for 10^{-5} (left) and 10^{-4} s (right).

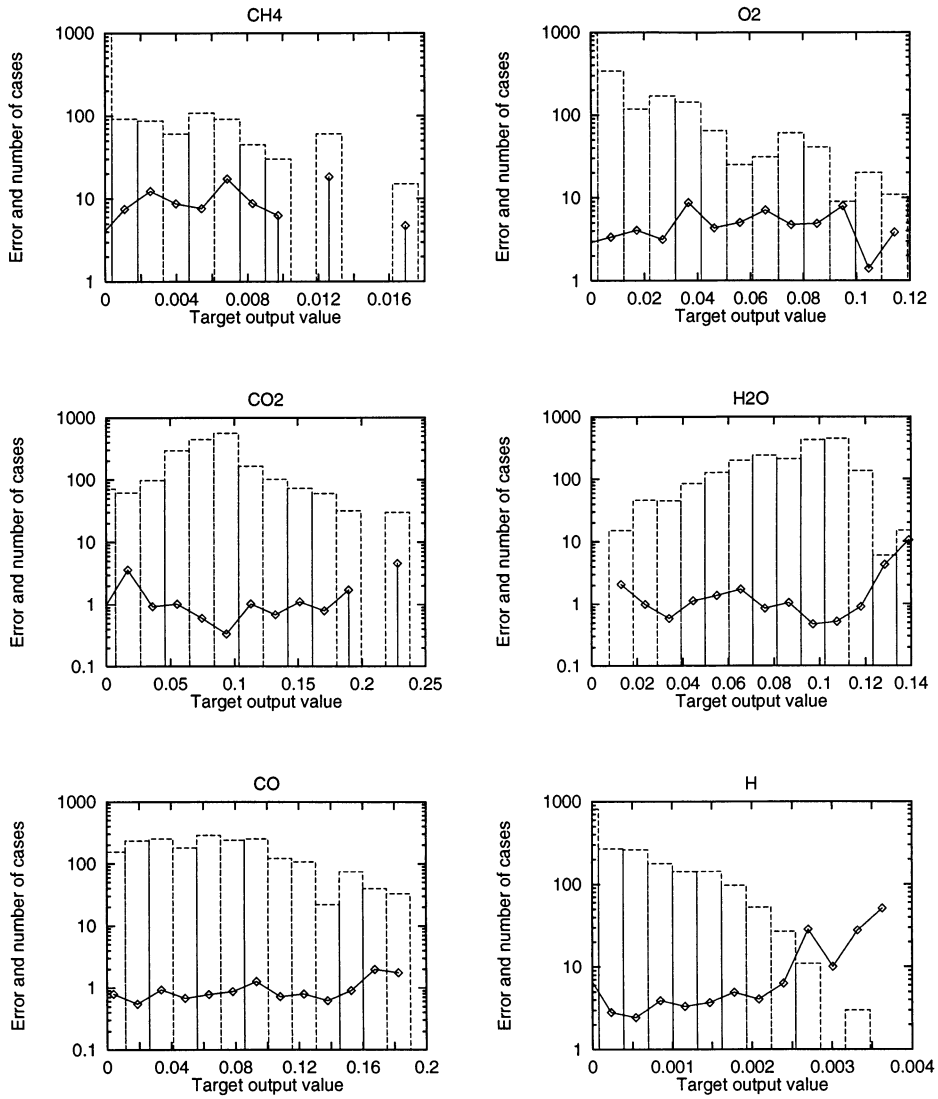

Fig. 3. Errors in the test set predictions averaged (in module) in target bins (lines) and number of cases in each bin (bars) for CH_4, O_2, CO_2, H_2O, CO, and H ($\Delta t = 10^{-5}$ s).

interval is clearly correlated with the number of samples in the interval, with larger numbers leading to smaller errors. Figure 3 is for the 10^{-5} s ANN, but similar results are obtained for the other two time steps.

Figure 4 is a scatter plot of the mass fractions predicted by the ANN vs the target ones for all the data for the 10^{-5} s ANN for six of the intervening species. The vertical distance from a point to the diagonal is therefore a measure of the error made by the network. The plotted points are reasonably close to the identity line, with a larger scatter for CH_4 and H.

Attention is now turned to the density-temperature ANN. The errors of the optimal ANN are shown in Table 4. These results indicate that a good interpolation (i.e., good fitting of the training data) has been achieved by the optimal ANN, the mean error being smaller than 1%. The table shows also good agreement between the training and the test set errors, which indicates that the number of training patterns is large enough to provide good generalization (i.e., good approximation of data outside the training examples).

Good error distribution is observed in Fig. 5 for

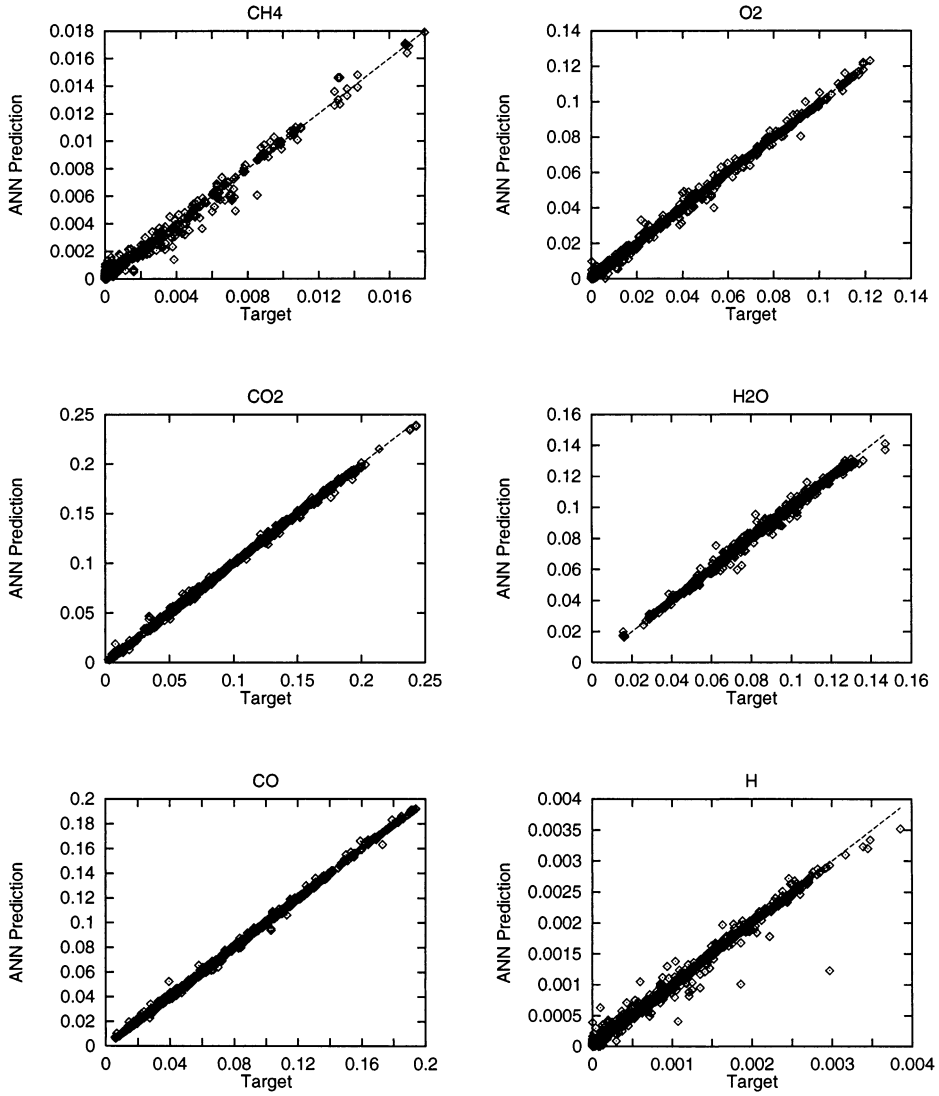

Fig. 4. ANN-predicted mass fractions plotted against target mass fractions in the training and test sets for CH_4, O_2, CO_2, H_2O, CO, and H ($\Delta t = 10^{-5}$ s).

both training and test sets. The errors are reasonably unbiased and are located within [−0.5%, +0.5%] and [−5%, +5%] for temperature and density respectively for nearly all the cases.

TABLE 4

Errors of the Optimal ANN for the Prediction of Temperature and Density in the Training and Test Sets

	Temperature	Density
Training set	0.19	0.70
Test set	0.19	0.67

ANN predictions of temperature and density follow very closely the target values, as can be ascertained from the scatter plot of Fig. 6.

Prediction of System Evolution

The "acid test" for the performance of the ANN would be its ability to reproduce the temporal evolution of the chemical system from a given initial point. In this subsection, some of such evolutions are compared with the result of a direct numerical integration of the chemical system.

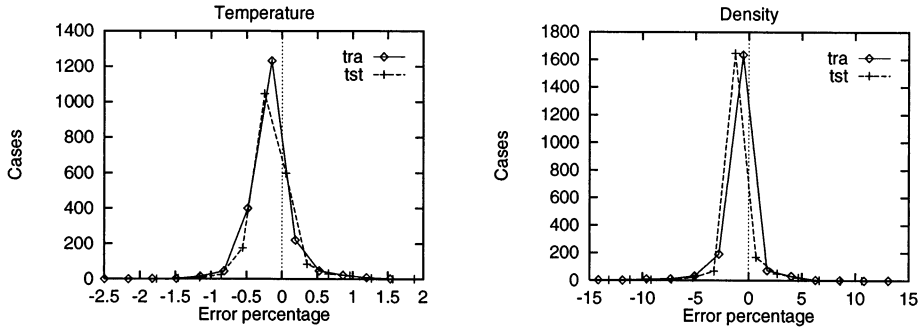

Fig. 5. Distribution of errors for the optimal temperature-density ANN for the training and test sets.

Figure 7 presents an example of such evolutions for one of the training series described in a previous section. The figure shows the evolution of six mass fractions predicted by two of the ANNs (the 10^{-4} and 10^{-5} s ones). These predictions are shown with symbols (squares for the 10^{-4} s ANN and diamonds for the 10^{-5} s ANN), and the numerically-integrated evolution is shown with lines. It can be gathered from the figure that both ANNs successfully capture the diversity of time scales present in the chemical system and accurately predict both trends and absolute values.

However, the data represented in the previous figure are taken from the training set, in which the network prediction at an intermediate time step uses as input the actual values at the end of the previous time step. An even more stringent test of the quality of the ANN model would be to feedback the ANN predictions at the end of a time step as the input to the next time step. Some results from such an exercise are shown in Figs. 8 and 9. These are typical comparisons between the ANN predictions and the "exact" integrated evolutions from two random initial points in compositional space, which are not in the training set. The discrepancies are of course slightly higher than in Fig. 7, but the network clearly proves to be capable of reproducing the range of time scales present in the problem, both qualitatively and quantitatively.

Computational Performance

A brief comparative study of the computational demands by different chemistry simulations is presented in this subsection with the aim of highlighting the advantages and drawbacks of each of these techniques.

The direct integration of the ODE system, the tabulated-chemistry approach, and the use of ANNs are compared. The chemical representation considered here contains information for three time steps (10^{-5}, 10^{-4}, and 10^{-3} s) as well as temperature and density values. Computer demands are measured in terms of two parameters: computer memory requirements (RAM) and CPU time. This comparative study has been

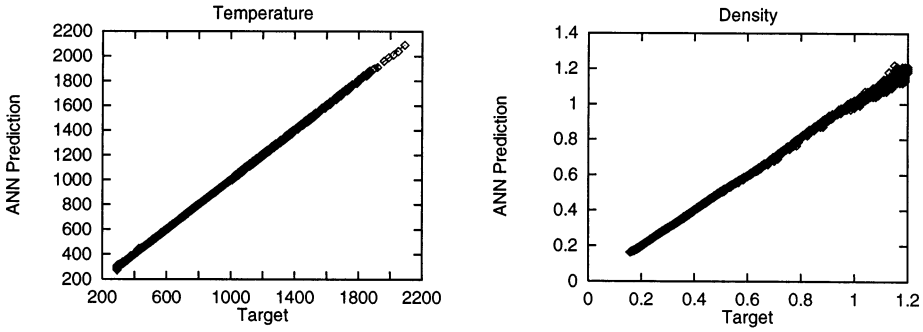

Fig. 6. ANN-predicted temperature and density against the target values.

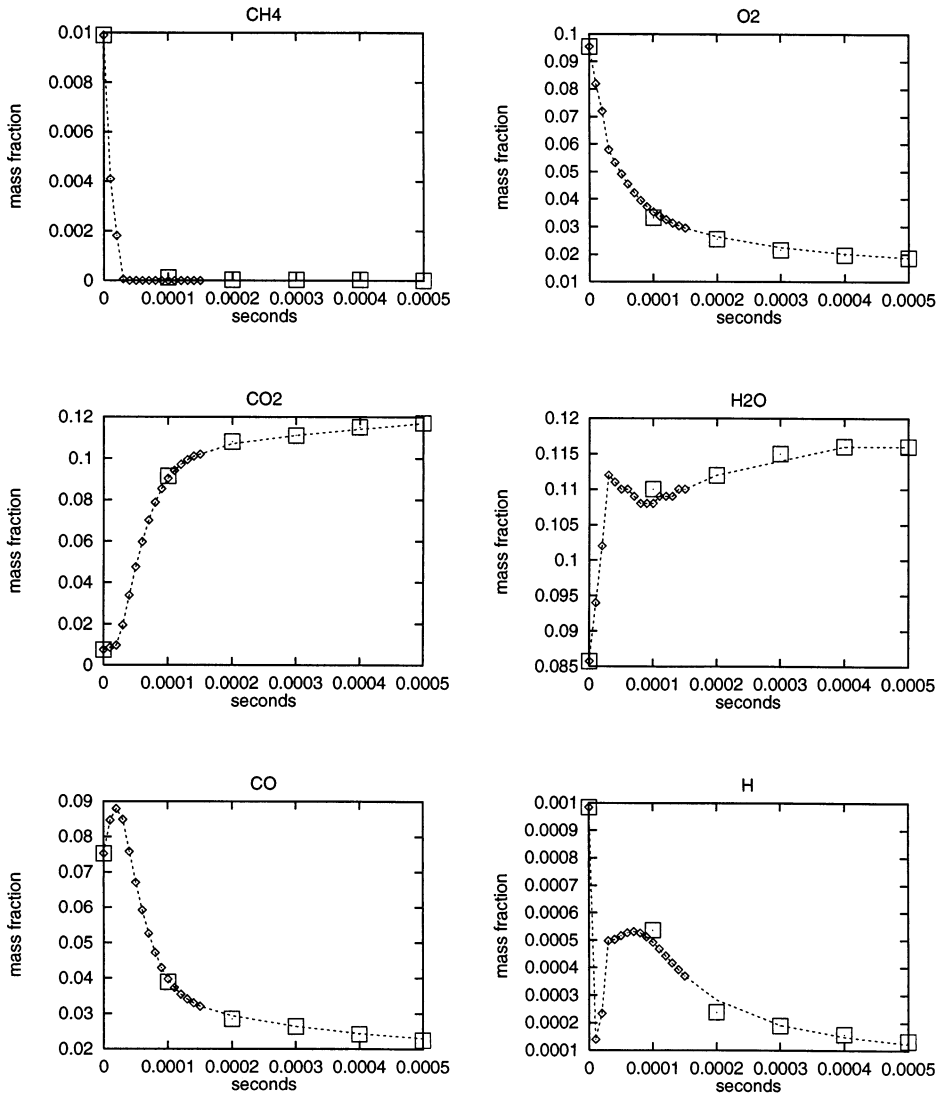

Fig. 7. Evolution of reactive species learned for the 10^{-5} and 10^{-4} ANNs (diamonds and squares, respectively) compared with the actual species evolution (lines).

performed on an IBM RS/6000 model 590 workstation.

The RAM resources utilized by each technique are summarized in Table 5. This table presents the total amount of RAM (in kbytes) used by each method in order to provide chemical evolution for the three time steps, and the temperature and density values. A large difference between the tabulated-chemistry approach and the other two (three orders of magnitude compared to the ANN, and two orders of magnitude compared to the direct integration method) is apparent. This is the result of the tabulated chemistry having to store in RAM mass fraction increments in a mesh in compositional space with large number of points. In the present example, the mesh considered is taken from [5] and is structured in an array, each entry being of the form of (f, $n\mathrm{CH}_4$, $n\mathrm{CO}$, $n\mathrm{TOT}$, $n\mathrm{H}$), of dimensions (25, 10, 9, 9, 9). For each of these points, four number-of-mole increments for each of the three time steps are stored, as well as the temperature and density values. Thus, the total number of values contained in the table are $25 \times 10 \times 9 \times 9 \times 9 \times (4 \times 3 + 2) = 2{,}551{,}500$. This is in turn converted to

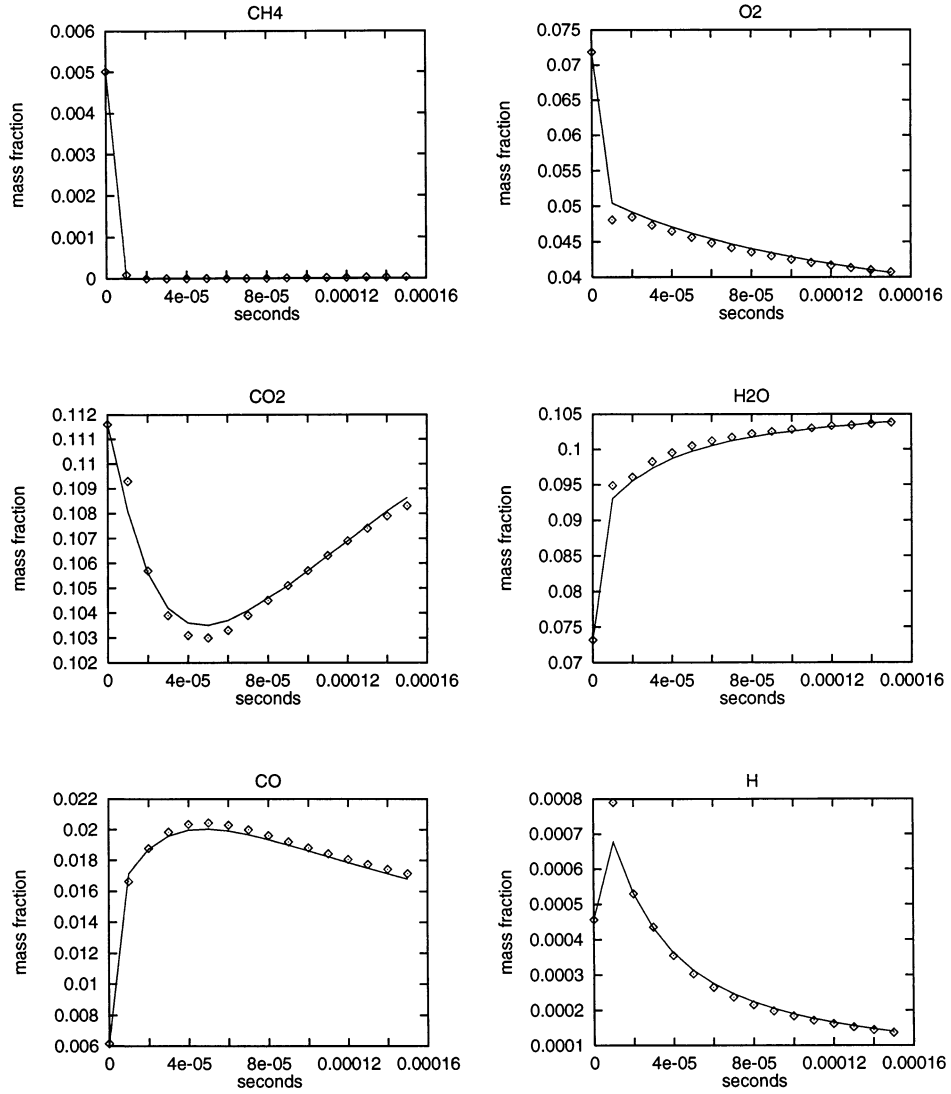

Fig. 8. An example of evolution of reactive species obtained for the 10^{-5} ANN by feedback of outputs (symbols) compared with the actual species evolution (lines).

kbytes by assuming 8 bytes for each stored value (double precision).

The direct integration method makes a fairly small use of computer memory, but even it is 11 times higher than the ANN approach if memory for both data and instructions is considered.

On the other hand, the ANN only makes use of a few data items (weights and scale factors) and short programming coding (matrix multiplications), thereby offering low RAM requirements. The ANN RAM usage has been calculated by counting the number of weights and scale factors, and multiplying this by four (the number of ANNs employed): $[9 \times 20 + 21 \times 20 + 21 \times 7 + (8 + 7) \times 2] \times 4 = 777 \times 4 = 3108$ float numbers = 24 kbytes (assuming 8 bytes for each float number).

Table 6 shows the CPU time required by each technique in order to provide the reactive-species or the temperature and density values.

In Table 6, three figures are shown for the direct-integration method, which correspond to the 10^{-5}, 10^{-4}, and 10^{-3} time steps, respectively. The direct-integration time clearly de-

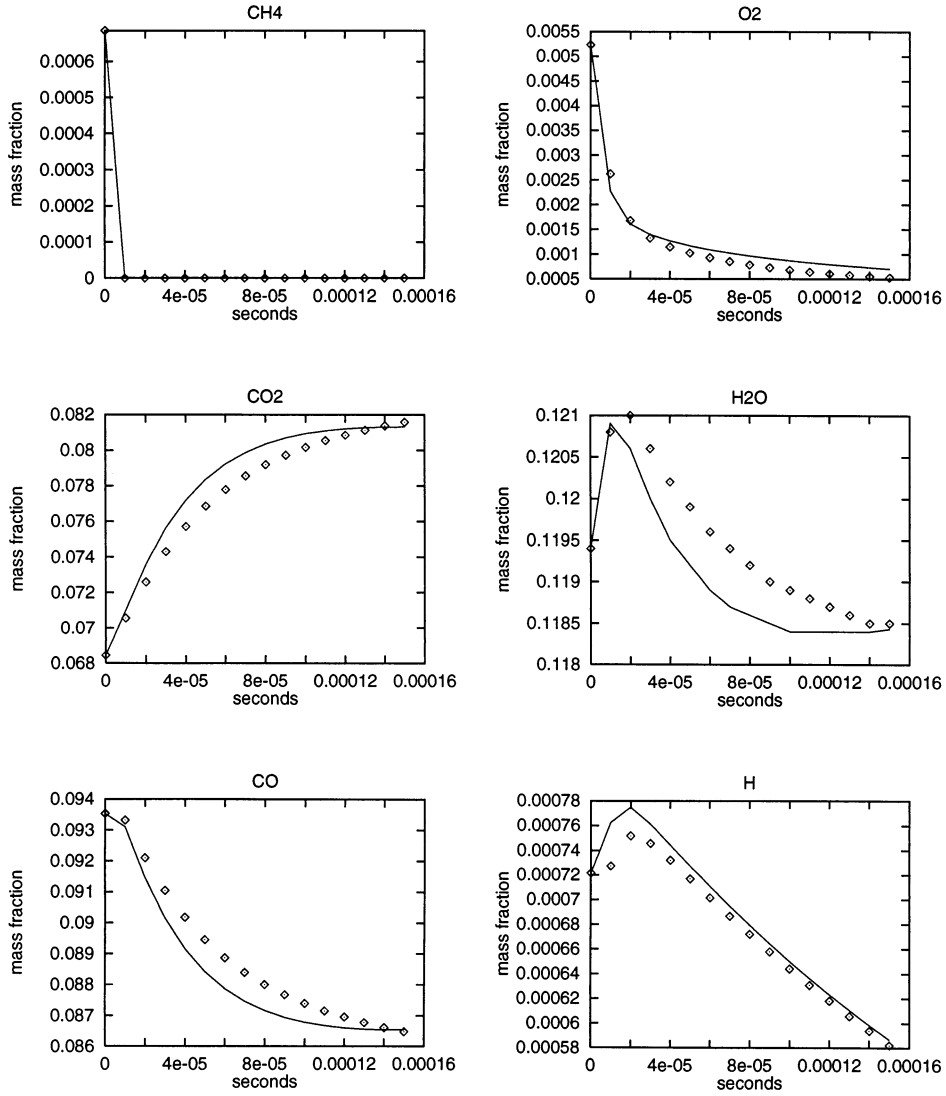

Fig. 9. Another example of evolution of reactive species obtained for the 10^{-5} ANN by feedback of outputs (symbols) compared with the actual species evolution (lines).

TABLE 5

Comparison of RAM Requirements (in kbytes) for the Different Chemistry-Representation Approaches, and Ratios (in Brackets) With Respect to the ANN Requirements (in the Tabulated-Chemistry and ANN Figures, Only Data-Usage RAM Has Been Reported)

	RAM requirements	Ratio
Direct Integration	276	(11)
Tabulated chemistry	19,934	(831)
Artificial Neural Network	24	(1)

TABLE 6

Comparison of Averaged CPU Time Usage (in ms) for the Reactive-Species and Temperature-Density Evaluation for Different Techniques (Ratios Are Given in Brackets for These Quantities With Respect to ANN Time)

	Reactive-species	Temperature-density
Direct integration	29, 61, 90 (165, 347, 511)	0.200 (6.6)
Tabulated Chemistry	0.030 (0.17)	0.030 (0.36)
Artificial Neural Network	0.176 (1)	0.083 (1)

pends on the time step used, being higher for the larger time step. The table reveals that direct-integration time is 3 orders of magnitude greater that ANN calculation time, which is in turn three to five times slower than the tabulated-chemistry approach.

The ANN response time is only a function of the network architecture. The networks employed are a two-hidden-layer for the reactive-species ANNs and a one-hidden-layer for the temperature-density ANN. Therefore, the reactive-species ANNs yield a longer response time (about twice) than the temperature-density ANN. The tabulated-chemistry response-time depends only on the input-space dimension. Thus, the same CPU time is spent for reactive-species as for temperature-density evaluation.

This brief analysis seems to indicate that ANNs are probably the best approach in terms of computer performance of the three methods compared. They are superior in RAM-storage requirements and show reasonably good CPU times, compared to the other techniques.

CONCLUSIONS AND FUTURE WORK

This contribution has illustrated how ANNs can be advantageously used to represent a chemical system of the kind typically employed in combustion simulation, with considerable economy in computer time (compared with direct integration of the ODE system) and memory (compared with tabulated chemistry). The methodology for building the ANN has been described, and the role of an adequate selection of the training samples has been stressed.

The results have been thoroughly examined in terms of accuracy, as it is considered that this is a necessary step before embedding the ANN models in more complex applications. Reasonable accuracy is achieved, with higher errors for the more reactive species. The different ANNs are successful in capturing the several time-scales involved in the combustion process. It is nevertheless felt that further work on sample selection can contribute to improvements in accuracy. Work is in progress in order to find more efficient automatic sample-selection criteria. Closely linked to this is the issue of determining the subregion of interest in composition space since it is not always possible (or simple) to set up a hierarchical calculation of scalar bounds such as the ones used in this work. Christo et al. [10] suggest the use of a method termed "statistical mapping," but this requires the performance of a complete simulation using conventional methods for integrating the thermochemistry. Alternative methods are being explored.

The support of the European Commission under Brite-Euram Project BE95-1927 is gratefully acknowledged, as is the help of Mr. Juan Carlos Larroya with the intricacies of the reduced chemical model used in this work. The authors also thank the paper's referees for their very useful comments. One of the authors (J. A. Blasco) is supported by the Spanish Ministry of Education and Culture through grant AP96-25449366.

REFERENCES

1. Baritaud, T., Poinsot, T., and Baum, M. *Direct Numerical Simulation for Turbulent Reacting Flows*, Éditions Technip, Paris, 1996.
2. Dopazo, C., in *Turbulent Reacting Flows* (P. A. Libby and F. A. Williams, Eds.), Academic Press, San Diego, 1994, p. 375.
3. Magel, H. C., Schnell, U., and Hein, K. R. G., in *3rd Workshop on Modelling of Chemical Reaction Systems*, Heidelberg, 1996.
4. Hindmarsh, A. C., in *Scientific Computing* (R. S. Stepleman et al, Eds.), North-Holland, Amsterdam, 1983, p. 55.
5. Chen, J. Y., Kollmann, W., and Dibble, R. W., *Combustion Science and Technology* 64:315 (1989).
6. Pope, S. B., *Computationally Efficient Implementation of Combustion Chemistry Using In Situ Adaptive Tabulation*, Sibley School of Mechanical and Aerospace Engineering Report No. FDA 96-02, 1996.
7. Haykin, S. *Neural Networks*: *A Comprehensive Foundation*. Macmillan College Pub. Comp., New York, 1994.
8. Lippmann, R. P., *IEEE ASSP Magazine* 4:4 (1987).
9. Widrow, B., Rumelhart, D. E., and Lehr, M. E., *Communications of the ACM* 37:93 (1994).
10. Christo, F. C., Masri, A. R., Nebot, E. M., and Pope, S. B., in *Twenty Sixth (International) Symposium on Combustion*, Naples, Italy, 1996.
11. Christo, F. C., Masri, A. R., and Nebot, E. M., *Combust. Flame* 106:406 (1996).
12. Peters, N., in *Numerical Simulation of Combustion Phenomena* (R. Glowinsky, B. Larrouturou, and R. Temam, Eds.), Lecture Notes in Physics 241, Springer-Verlag, 1985, p. 90.

13. Peters, N. and Kee, R. J., *Combust. Flame* 68:17 (1987).
14. Coltrin, M. E., Kee, R. J., and Rypley, F. M., *Int. J. Chem. Kinet* 23:1111 (1991).
15. Moller, M., *Neural Networks* 6:525 (1993).

APPENDIX

In this appendix, the hierarchical bounds for the five-controlling scalars contrived by Chen [5], which are applicable to the chemical system under study [12], are described in detail.

— The mixture fraction f takes values between 1 (in the fuel stream) and 0 (in the oxidizer stream). However, the stoichiometric value of f for methane-air combustion is 0.055, and far off this point the chemical activity can be neglected. Therefore, the training samples will take values of f chosen at random between $f^{\min} = 0$ and $f^{\max} = 0.3$; chemical reaction rates will be presumed to be zero outside this range.

— For a given mixture fraction, f, the allowable range for the specific number of CH_4 moles, $n\mathrm{CH}_4$, is

$$n\mathrm{CH}_4^{\max} = \frac{f}{W_{\mathrm{CH}_4}};\ n\mathrm{CH}_4^{\min} = \max\left\{\frac{f}{W_{\mathrm{CH}_4}} - \frac{2\psi(1-f)}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}},\ 0\right\}. \tag{6}$$

— Given f and $n\mathrm{CH}_4$, the specific number of CO moles, $n\mathrm{CO}$, must lie between

$$n\mathrm{CO}^{\max} = \chi;\ n\mathrm{CO}^{\min} = \max\left\{2\chi - \frac{2\psi(1-f)}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}},\ 0\right\}. \tag{7}$$

— Given f, $n\mathrm{CH}_4$ and $n\mathrm{CO}$, the limits for the total number of moles per unit mass, $n\mathrm{TOT}$, are

$$n\mathrm{TOT}^{\max} = \frac{(1+\psi)(1-f)}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}} + \frac{2f}{W_{\mathrm{CH}_4}} - n\mathrm{CH}_4 + \frac{1}{2}n\mathrm{CO},$$

$$n\mathrm{TOT}^{\min} = \frac{1-f}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}} + \frac{3f}{W_{\mathrm{CH}_4}} - 2n\mathrm{CH}_4 + \max\left\{\frac{\psi(1-f)}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}} - 2\chi + \frac{1}{2}n\mathrm{CO},\ 0\right\}. \tag{8}$$

— Once f, $n\mathrm{CH}_4$, $n\mathrm{CO}$, and $n\mathrm{TOT}$ are determined, the domain for the specific number of H moles, $n\mathrm{H}$, is

$$n\mathrm{H}^{\max} = 2n\mathrm{TOT} - 6\frac{f}{W_{\mathrm{CH}_4}} + 4n\mathrm{CH}_4 - 2\frac{1-f}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}} - 2\max\left\{\psi\frac{1-f}{W_{\mathrm{N}_2} + \psi W_{\mathrm{O}_2}} - 2\frac{f}{W_{\mathrm{CH}_4}} + 2n\mathrm{CH}_4 + 5n\mathrm{CO},\ 0\right\}, \tag{9}$$

$$n\mathrm{H}^{\min} = 0.$$

In the case of the upper limit of $n\mathrm{H}$, the expression given is not exactly the one proposed by Chen in [5]. Instead, a new expression has been derived which reduces the round-off error involved in its computation.

In these limits, W_X is the molecular weight of species X, $\chi \equiv f/W_{\mathrm{CH}_4} - n\mathrm{CH}_4$, and ψ is the O_2:N_2 atomic ratio in air = 0.79/0.21.

Received 17 February 1997; revised 3 June 1997; accepted 11 June 1997