

“Store and retrieve” representations of dynamic systems motivated by studies in gas phase chemical kinetics

Ioannis P. Androulakis*

Complex Systems Modeling, Corporate Strategic Research, ExxonMobil Research and Engineering, 1545 Route 22 East, Annandale, NJ 08801, USA

Received 25 August 2003; received in revised form 16 February 2004; accepted 18 February 2004

Available online 18 May 2004

Abstract

This paper explores the advantages and highlights the potential limitations of using *off line* information to generate approximations of dynamic systems. The procedure is presented in the context of existing approaches motivated by studies in chemical kinetics. Store and retrieve approaches utilize previously generated information about the kinetic evolution of the reaction system in order to build implicit approximations. The unstructured data reside in memory and computational algorithms, detecting nearest neighbors, are used to define a region around a novel query point. The properties of the query point are determined by linear regression of the stored information. The examples we considered are designed to address the merits of the approach as well as the fundamental issues that warrant further research. © 2004 Elsevier Ltd. All rights reserved.

Keywords: Store and retrieve; Dynamic systems; Gas phase chemical kinetics

1. Introduction

The potential of computer-aided tools for the generation of large detailed kinetic mechanisms has been receiving, over the years, increasing attention. Advances in our fundamental understanding of how gas phase reactions occur, aided by recent progress in the theory and computations related to *ab initio* calculations, have resulted in a number of automated mechanism generation approaches. The final product is a very detailed representation of kinetic transformations involving, possibly, thousands of reacting species, and tens of thousands of reactions. A recent review by (Green *et al.*, 2001) summarizes a number of alternatives, challenges and opportunities. Even manually derived kinetic mechanisms tend to be extremely detailed when modeling complicated phenomena, such as the low-temperature auto-ignition of fuel-like molecules (Curran, Pitz, Westbrook, Callahan, & Dryer, 1998).

Detailed kinetic mechanisms have made significant contributions in a variety of applications in which molecular specificity is critical, such as pollution prevention and

efficiency increases in internal combustion engines (Faravelli, Gaffuri, Ranzi, & Griffiths, 1998), or reactor simulations coupling computational fluid dynamics and detailed kinetic mechanisms (Shah & Fox, 1999).

However, the computational complexity of reactive flow simulations prohibits the use of kinetic mechanisms of increased dimensionality. Therefore, the fundamental advantage of using accurate kinetics is not fully captured due to our inability to properly incorporate such degree of detail in interesting calculations. As a result there have been, over the years, a number of computational advances and approaches that aim at reducing the computational cost associated with the evaluation of the kinetic terms. These approaches attempt to either extract the most significant contributors to the overall kinetic scheme, thus utilizing only parts of the detailed kinetic mechanism, or build approximations to the kinetic models that are subsequently used as necessary. A thorough review of a number of issues recently appeared in Tomlin, Turanyi, and Pilling (1997).

In this paper alternative approaches that store information about the dynamics of the kinetics that are subsequently used as needed are presented. First, a general review of the issues in modeling chemical kinetics is presented. Subsequently, a number of computational approaches used for

* Tel: +1-908-730-2111; fax: +1-908-730-3344.

E-mail address: ioannis.p.androulakis@exxonmobil.com (I.P. Androulakis).

the analysis and reduction of kinetic mechanisms are discussed. These observations lead to a discussion of the “store and retrieve” approaches proposed in this paper. A number of computational issues are discussed and the main focus of the presentation is computational algorithms for locating nearest neighbors in high dimensions. Finally, a critical review of the approach and potential enhancements is discussed.

2. Representing a complex kinetic mechanism

Once a mechanism has been generated, either manually or via one of the available automated mechanism generation approaches (Come et al., 1997; Susnow, Dean, Green, Peczak, & Broadbent, 1997), it defines a set of chemical transformations that capture the rate of change of concentration in a reaction system. The simplest way to think of such representations is by considering a homogeneous environment that focuses exclusively on the contributions of the chemical kinetics. The changes are described by a set of ordinary differential equations:

$$\frac{dy_s}{dz} = \sum_{r=1}^{N_R} \alpha_{rs} R_r, \quad \frac{dT}{dz} = \sum_{s=1}^{N_S} H_s^g \frac{dy_s}{dz}$$

$$R_r = K_r^F e^{-E_r/RT} \prod_{s=1}^{N_S} [X_s]^{\alpha'_{rs}} - K_r^R e^{-E_r/RT} \prod_{s=1}^{N_S} [X_s]^{\alpha''_{rs}} \quad (1)$$

$$y_s(0) = y_s^0, \quad T(0) = T^0$$

Eq. (1) defines the standard heat and material balance. Species mass fractions are denoted by y , and molar concentrations are denoted by x , the stoichiometric coefficient of species “ s ” in reaction “ r ” is denoted by α_{rs} , H is the specific enthalpy of species “ s ”. K denotes rate constants, and E is the activation energy (in standard CHEMKIN format). The superscript 0 denotes initial conditions. The thermo-physical properties and reaction rate definitions are handled using standard software packages such as CHEMKIN, (Kee, Rupley, Meeks, & Miller, 1996).

2.1. Physical description (detailed mechanism)

Incorporating the complete set of species in performing the integration of (1) guarantees that all the available kinetic information has been and properly accounted for. However, the number of reacting species (molecular as well as fragments) grows rapidly. Although considering all of the species naturally results in the highest level of accuracy, computationally the problem becomes hard to solve. As a result mechanism reduction, equivalent to model reduction, has emerged as a viable alternative to representing the detailed kinetic information. Based on the selected approach, a number of possibilities are available.

2.2. Reducing the kinetic mechanism while maintaining its structural integrity

A number of computational approaches have been proposed that aimed at reducing the complexity of the chemical transformations by identifying critical subsets of species and reactions that are most important in generating an expected response. Various approaches based on sensitivity analysis have been implemented in the past with great success. A number of approaches, and various references, have been compiled and discussed in Tomlin et al. (1997). Recently, the problem of identifying a critical sub-set of species and/or reactions was cast as an optimization problem (Androulakis, 2000; Petzold & Zhu, 1999). The outcome of the optimization problem is a sub-set of the original mechanism that reproduces the dynamic behavior of the detailed one.

Regardless of the approach the ultimate result of the analysis is the identification of a sub-set of species and reactions that reproduces the dynamic response of the detailed mechanism. The reduced representation of the detailed kinetics corresponds to one that maintains the structural integrity of the detailed mechanism.

2.3. Quasi-steady state and partial equilibrium

Combination of the quasi-steady state approximation (QSSA) and the partial equilibrium assumption (PA) attempt to replace the ODEs of the steady-state species by their corresponding algebraic equations. The approach has been very successfully applied in a number of situations, for example in Seshadri and Peters (1990) in order to derive compact representation of chemically reacting systems. Further summaries can be found in Warnatz, Maas, and Dibble (1996).

The approach of identifying the fast and slow components of the dynamic system (i.e., the components that reach equilibrium fast) was successfully automated in a number of different approaches. These include the Augmented Reduced Mechanism (ARM) generation (Chen, 1988), the Computational Singular Perturbation (CSP) (Lam & Goussis, 1988), and the Intrinsic Low Dimensional Manifold (ILDm) (Mass & Pope, 1992).

Conceptually, all approaches share the same underlying characteristics. A slow manifold of lower dimension is identified and the dynamics of the detailed system are “projected”, and hence monitored, on this sub-set. The original system of equations is replaced by a system of coupled differential and algebraic equations (CSP and ARM). Therefore the structure of the system has changed. The structural changes are even more pronounced in the case of ILDM. The system dynamics are still monitored on the reduced manifold. However, the manifold has been parameterized in such a way that eventually the system is described with the help of the so-called “progress variables” which may or may not represent actual species concentrations (Mass &

Pope, 1992). It should be pointed out however, that from the point of view of “reduction” the Mass and Pope ILDM approach is one of the most innovative ones. The driving force behind the development was to truly reduce the dimensionality of problem and recast the simulation in terms of the “progress variables”.

The fundamental difference between these approaches and the ones that maintain the structural integrity of the system is that the notion of the original kinetic mechanism is somewhat relaxed.

2.4. Off-line approximations

Recently an alternative way of looking at the dynamics of kinetic mechanisms is emerging. The approach is motivated by the observation that significant improvements in computational efficiency can be achieved by properly utilizing information about the kinetics that has been generated *off-line*.

2.4.1. *Repro-modeling (RM)*

Information for rates is extracted from detailed chemical calculations and stored in the form of high-order multivariate polynomials. The approach is based on a large number of simulations using typical input data. Based on these simulations the input–output relations of the sub-models are approximated by explicit empirical equations. The polynomials are compositions of orthogonal monomials whose coefficients are estimated recursively. As the evaluation of the explicit equations is much faster than the simulation of the original model, the implementation of the approach can result in significant speed-up of the overall calculation (Turanyi, 1994).

2.4.2. *PRISM*

A different approach to address a similar computational issue was undertaken by Tonse, Moriarty, Brown, and Frenklach (1999). Instead of deriving a single approximation to the phase space, the multidimensional chemical component space is parameterized via piecewise continuous polynomial approximations. Factorial design methods are used to reduce the required number of computed points. The polynomial coefficients for each hyper-cube are stored in a data structure for subsequent reuse.

2.4.3. *High Dimensional Model Representations (HDMR)*

The approach proposed by Rabitz, Alis, Shorter, and Shim (1999) aimed at achieving a dramatic reduction in the scaling associated with building input–output representations in multidimensional spaces by capitalizing the often expected low order correlation amongst the input variables having an impact on the output. Unlike the previous polynomial approximations, HDMR, build approximations recursively since the fundamental assumption is that high order correlated effects of the inputs are expected to have negligible input on the output one can

construct the approximation by monitoring low order terms first.

2.4.4. *Atmospheric parameterization models*

A number of parameterization models have been proposed in environmental and atmospheric science applications. The three-dimensional models required for describing the chemistry and the dynamics are becoming important, yet their computational requirements are very demanding. In Duncan, Portman, Bay, and Spivakovsky (2000) such parameterization is presented. Polynomial input–output representations are developed that predict minor species compositions as function of other critical species. A systematic procedure for dividing the domain of interest into sub-domains is presented. The division and adaptive model construction aim at improving the accuracy of the derived polynomial approximations.

2.4.5. *ILDM and repro-modeling*

Buki, Peger, Turanyi, and Maas (in press) combined ideas from repro-modeling and ILDM to enhance computational efficiency. Typical conditions that are expected to occur during the simulation are presented to the repro-modeling framework. The projection of these points onto the reduced manifold, as defined by the ILDM approach is approximated via the use of the ortho-normal families of polynomials whose coefficients are recursively estimated up to the required accuracy.

2.4.6. *ILDM/polynomials*

Niemann, Schmidt, and Maas (1997) derived efficient domain decomposition approaches in order to develop adaptive, yet continuous, approximations of the ILDM data. Simulation data are used to generate the implicit relations defining the slow manifold onto which the reactions take place. Unlike the original tabulation of the values defining the reduced manifold (Niemann et al., 1997) derive explicit approximations in the form of algebraic equations that are used in subsequent simulations.

Similar in spirit is the implementation of the ILDM proposed by Rhodes, Morari, and Wiggins (1999). The equations defining the slow manifold are solved ahead of time and a global quadratic model relating the fast and slow variables is generated. These explicit expressions are substituted back into the original equations thus resulting in a reduction of the dimensionality of the problem.

2.4.7. *Nonlinear PCA*

Kirby and Miranda (1999) explore and expand the concept of auto-associative non-linear neural networks (Kramer, 1991) in order to derive non-linear projections of the original dynamics onto a space of reduced dimensionality. Simulation data generated *off line* are used to train the network. In subsequent simulations the dynamics of the system are followed in terms of the reduced dimension as defined by the auto-associative network.

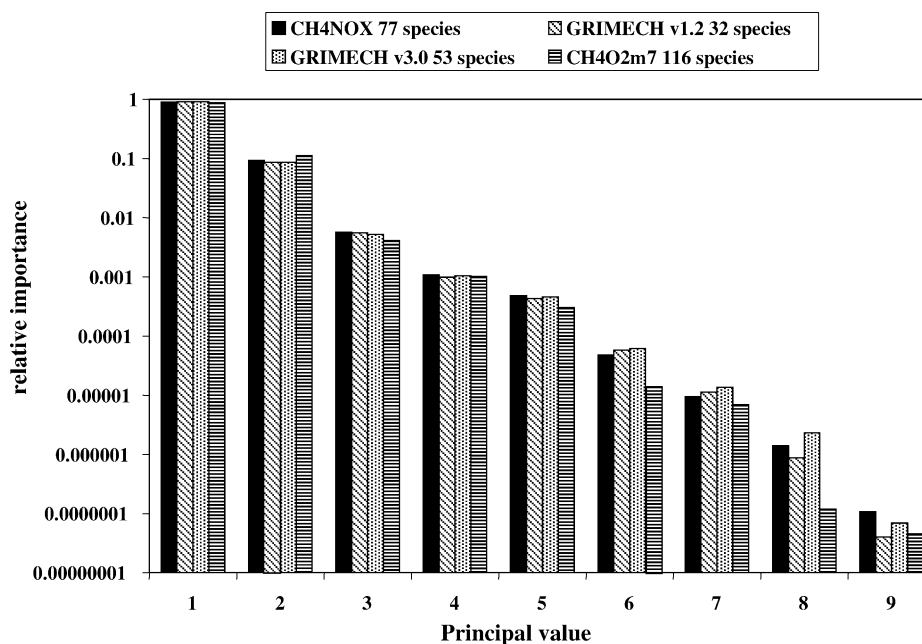


Fig. 1. Distribution of leading singular values for various mechanisms.

2.4.8. In-Situ Adaptive Tabulation (ISAT)

Recently, Yang and Pope (1998) developed a new method to treat chemical reactions in combustion problems. This approach defines a significant departure from all previously described approaches in that no approximation is derived based on *past* calculations. The database is empty at the beginning of the simulation and it is built as required on the fly by the specific simulation. Rate information is generated and is stored in tables for future use. The method was developed primarily for turbulence simulations using probability density functions (PDF) as the target. A novel scheme that exploits the projection of the data set onto principal directions (singular vectors) is used in order to enhance the search and reduce the memory requirements.

Tabulation of values allows the construction of the attainable region during the process of the reaction, whereas projection on the space of leading eigenvectors allows a comprehensive representation of the reaction space. In Fig. 1, we depict some indicative simulation results with various kinetic mechanisms. In short, in all cases we are simulating a plug flow reactor. For a given initial condition (composition, pressure and temperature) the constitutive equations are as defined in Eq. (1).

At each point in time the state space is defined as a multidimensional vector composed of all the compositions and temperature (the simulations are assumed to be isobaric). A large number of simulations from random initial points are conducted and the trajectories are appropriately sampled (this is discussed later in the manuscript). A Singular Value Decomposition of the matrix composed of all the trajectory points is then calculated in order to determine a reduced representation of the original data. In classical data analysis this is also referred to principal as component analysis (PCA).

The trajectory point is defined as the multidimensional vector of compositions in time as generated by the simulation. Fig. 1 depicts the distribution of the relative magnitude of the eigenvalues. Within the context of PCA the relative magnitude of an eigenvalue corresponds to the percent of data variability along the corresponding eigenvector. The fact that most of the variability can be explained by a few leading eigenvectors, out the 32 total, implies that the data can be represented by a projection onto the sub-space defined by a handful leading eigenvectors. This distribution is what Yang and Pope (1998) explored in order to develop their tabulation and search scheme. Karhunen-Loeve decomposition as applied to chemically reacting systems (Graham & Kevrekidis, 1996), also explores this fact by recasting the constitutive equations in this reduced space. What is actually quite interesting is that the number of significant dimensions, i.e., leading eigenvalues, does not really depend on the specific kinetic mechanism used. We derived the decomposition for a number of mechanisms, i.e., varying number of reacting species and reactions. Having the linear projection, based on the associated singular vectors, allows the projection of the entire reaction space onto a lower dimensionality manifold. This is depicted in Fig. 2. A sample run is plotted in which the reacting mixture is a stoichiometric mix of air and hydrogen. The GRI mechanism is used composed of 32 species. A couple of reaction trajectories are depicted in terms of OH, O, and H radical composition (a), as well as in the projection in three eigenvectors. The surface in (b) is composed of a large number of trajectories for various initial conditions in which we have imbedded the specific trajectory depicted in (a). The point of these simulations is to enhance and justify the motivation behind the ISAT in principal directions approach of Yang and Pope, 1998. The ability

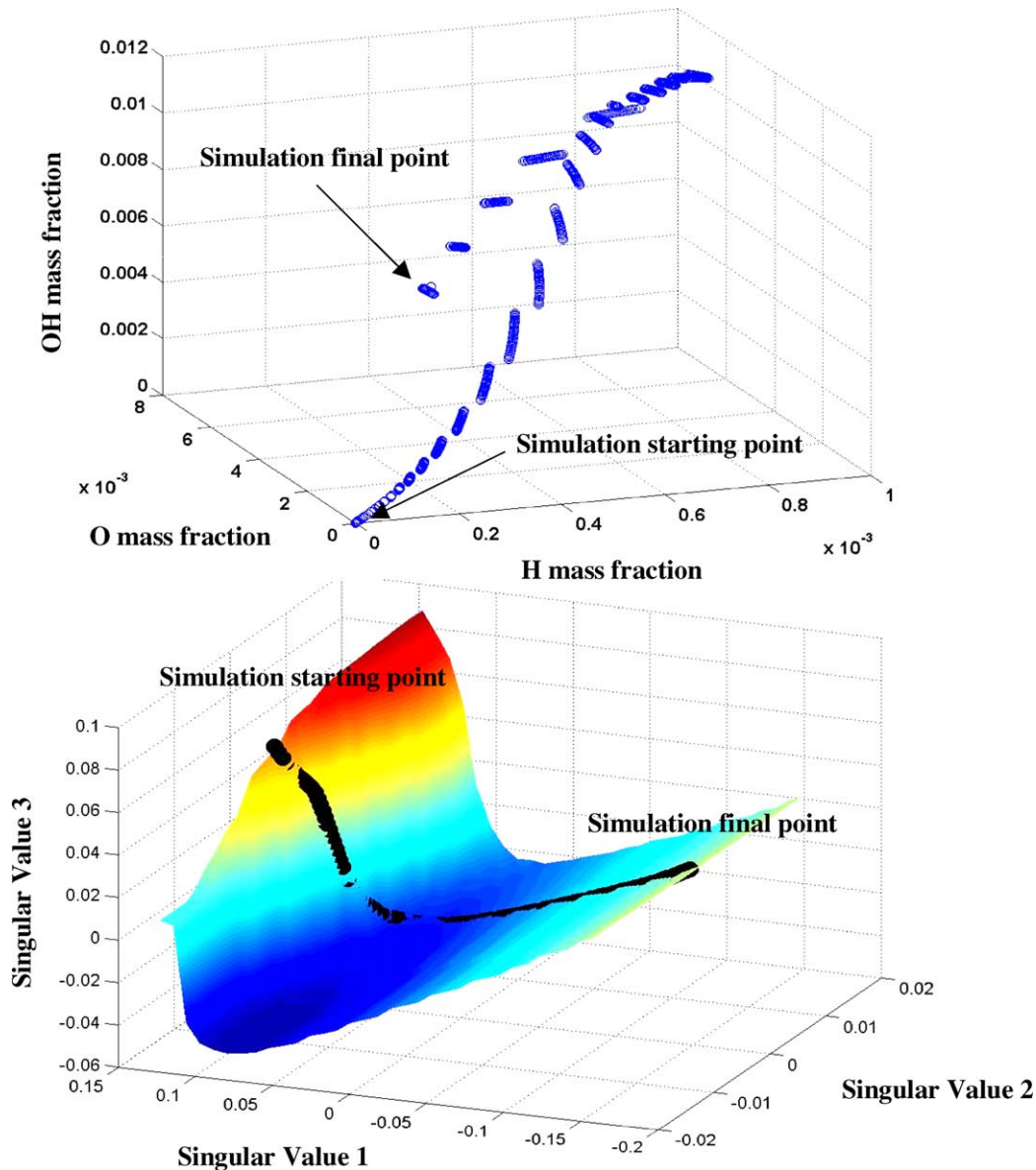


Fig. 2. Reaction trajectories in composition and singular vector spaces.

to have a reduced representation of the reaction phase space makes for a faster and more efficient search of the tabulated values. This is the attribute explored by Yang and Pope (1998).

3. Store and retrieve hybrid representations

3.1. Some comments on representations

As can be seen from the description of various approaches the combustion community, primarily, has struggled for the longest time in an attempt to find ways to represent the dynamic evolution of the gas phase reactions that take place during combustion. Many methods explore the fact that time-scale separation is dominating combustion phenomena.

Hence, researchers would either use intuition and/or experience to identify fast and slow components (Seshadri & Peters, 1990), or explore concepts from non-linear dynamics theory (ARM, CSP). In principle, one could derive an alternative representation by removing the fast components altogether. The steps could be simply described as follows:

1. Assuming that the slow (x) and fast (y) components of the system are known ahead of time, define the dynamics as

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

2. Since y is the fast component replace its equation of state by a corresponding algebraic equation:

$$\frac{dx}{dt} = f(x, y)$$

$$0 = g(x, y)$$

3. If the algebraic equation can be solved in closed form, then define explicitly y :

$$\frac{dx}{dt} = f(x, y)$$

$$y = g^{-1}(x)$$

4. Use the y value to define the dynamics of x :

$$\frac{dx}{dt} = f(x, g^{-1}(x))$$

This is precisely how experienced chemists with a deep and fundamental understanding of reactive phenomena have been analyzing the dynamics of reactions for many years. The advances in our ability to generate very large kinetic mechanisms is what, more recently, gave rise to automated and algorithmic-based approaches that detect fast and slow species and also define explicit or implicit relations between the slow (x) and fast (y) components (ARM, CSP, ILDM).

However, the off-line approximations that were discussed in the previous section take computational-based approaches one step further. The ILDM approach, for instance, is using the concept of time-scale separation. However, the information is tabulated and used as needed. Furthermore, the dynamics is described, hence the tabulation is defined, in terms of a small number of “progress variables” which are used as flags in order to retrieve *on-line* the information that was stored *off-line*. The ISAT approach goes one step further. The information of the exact dynamics is stored, and the tabulation is performed in the space of singular values. That helps reducing the computational complexity associated with searching and retrieving the required information. Although the dimensionality of the system is not reduced the rate at which information about the system is generated is significantly enhanced.

It is clear that the most recent wave of approaches aim at exploring information that can be generated *off-line* and then used as needed during the computation. Although significant issues still exist with the validity of the approach it does provide a very powerful concept with significant extensions. A discussion later will address a number of these issues.

What we propose in this work is a variation of the existing approaches, combined with advanced algorithms for searching multidimensional spaces. The approach aims at implicitly defining either the inverse mapping g^{-1} (Step 3) by tabulating the functional relationship $y = g^{-1}(x)$, thus determining $dx/dt = f(x, g^{-1}(x))$, based on a number of *off-line* simulations, or explicitly defining the relationship $dx/dt = f(x, g^{-1}(x))$ by tabulating the $(x, dx/dt)$ pairs. The

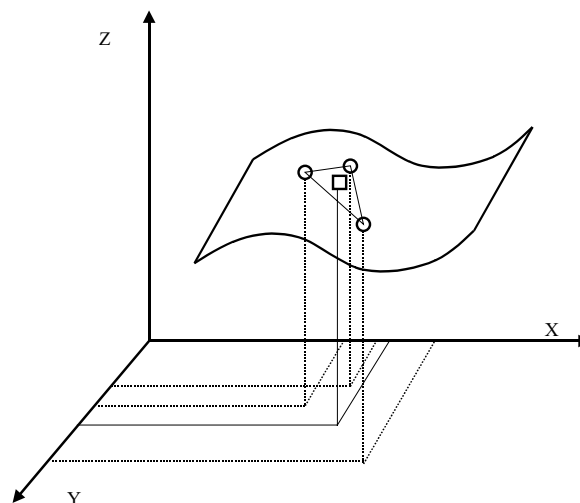


Fig. 3. Neighborhood in an arbitrary surface.

off-line simulations attempt to capture the entire dynamic response of the systems, hopefully alleviating the errors introduced by deriving such relationships based on the fast/slow component concept. The goal is to derive a representation similar to the one defined in Step 4 earlier.

The basic idea of the proposed approach is to generate, *off-line*, a database containing either the $(x, g^{-1}(x))$ or the $(x, dx/dt)$ pairs. Subsequently, in order to use the reduced representation defined by the x -variables only, the database is searched so as to identify a tight neighborhood of points near a query point (Fig. 3). This neighborhood will be then used in order to assign either the $g^{-1}(x)$ or the dx/dt value that corresponds to the query point and proceed with the computation.

The following is a list of attributes that we attempted to address in designing this general framework.

1. The dimensionality of the original system is reduced, by selecting a subset of the original variables. These variables are used as key indicators. Time evolution of these indicators only is defined and recorder.
2. The intrinsic dynamics of the original system are used in building the input–output representations in order to avoid possible errors introduced by exploring the time-scale separation concept.
3. Efficient nearest-neighbor identification algorithms are implemented in order to perform the search in the original space, and not alternative representations of it such as singular value space.
4. The information of a neighborhood of points is used to define the properties of a query point.

3.2. Algorithmic and computational issues

3.2.1. Modeling

We assume that we have the description of a dynamic system. V denotes the variables that define the state of the

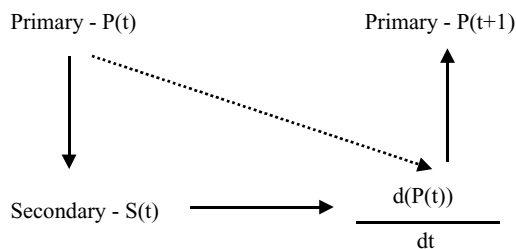


Fig. 4. Computational methodology for integrating primary and secondary species information.

system. We assume that V is partitioned, arbitrarily at this point, into two sets, namely the primary (P) and the secondary (S) components. The P components are quantities of interest that we wish to monitor, or measured quantities whose response is of importance. The goal is to somehow simulate the system in a way that we only deal, explicitly, with the P components. We wish to estimate the dP/dt term in way that $P(t)$, when integrating dV/dt , and $P(t)$ when integrating the “new” dP/dt are equivalent. Of course, the time evolution of P depends on S . What is proposed is to use *off-line* simulations in order to tabulate the values of S that correspond to archived values of P . In that case, $(dP/dt) = f(P, S) = f(P, g^{-1}(P))$. The inverse map is implicitly refined via the tabulated values. In building the right hand side of dP/dt one has two options. Either the values of S that correspond to tabulated values of P are stored, or the actual values of dP/dt that correspond to P are stored. The latter have the contributions of S already absorbed. If such a relationship can be successfully established it would have captured the dynamics of the functional relations between $P(t)$ and $S(t)$. Fig. 4 summarizes the procedure.

3.2.2. Selection of primary and secondary species

The partitioning between P and S is to some extent arbitrary. One can argue that appropriate values for P should be components of V that can either be measured or are of general interest. For instance, major pollutants and reactants in a reactive model simulation could constitute P , whereas intermediate radicals and smaller fragments could define S . A fundamental theoretical issue, which should be explored, is whether the mapping $(P, dP/dt)$ or (P, S) is *unique*. In other words, where for each value of P there is one and only one value for dP/dt or S . However, this is essentially the same problem that other methods, such as the ILDM approach face. The key question is whether the variables that are chosen for the tabulation of the simulation data are appropriate and do not result in multiple answers. This is definitely a question that deserves further exploration. At this point, we would like to lay the computational foundation for performing the database construction and subsequent search and retrieve of information.

3.2.3. Snapshot generation

In order to build the required database so as to establish the relationship between P and S a number of simulations of

the original system need to be evaluated. Since we are dealing at this point with initial value problems, the inputs to the simulation, i.e., initial conditions, will be considered as random variables. The trajectories are recorded and appropriately sampled (see next paragraph). The sampled points are stored for future reference.

3.2.4. Sampling the trajectories

The *off-line* simulations of the original system generate a large number of points when following the temporal evolution of the system. In principle, the integrator appropriately generates time outputs corresponding to the time-step that is selected. This set of points has to be adequately reduced via sampling. We implemented the procedure proposed by Graham and Kevrekidis (1996). Instead of a simple time-average sampling, the procedure emphasizes infrequent events. If we consider a general system of ODEs: $\dot{U} = F(U)$, an arc-length s is defined as (the nomenclature of the original reference is used for consistency):

$$ds = \left| \frac{dU}{dt} \right| dt = |F(U)| dt \Rightarrow \frac{ds}{dt} = |F(U)|. \quad (2)$$

Data are then sampled at equal intervals in arc-length rather than equal intervals in time. A sample case is depicted in Fig. 5. The original trajectory is sampled at 50 points equally spaced in arc-length.

3.2.5. Tabulation

The selected tabulation is rather straightforward. A table is constructed with rows containing the (P, S) or $(P, dP/dt)$ pairs. Yang and Pope (1998) proposed the use of principal directions in order to efficiently tabulate and search the stored values in their algorithm. We will be experimenting with some new search algorithms for nearest neighbor in high dimensions, because we feel that efficient searches in the original space should also be explored as they minimize the potential for error introduced by the linear projection of the database onto the space of principal directions.

3.2.6. Nearest neighbor search in high dimensions

The problem of nearest neighbor search (NNS) can be informally defined as: given a database of n points in some metric space, preprocess the data so as to efficiently find the point(s) in the data base closest to the query point (Knuth, 1988). Since it involves the notion of similarity search it has recently drawn more attention due to its vast applications related to information retrieval. NNS is an integral part in a wide range of applications that include multimedia databases, computational biology, data mining, and information retrieval. The common thread in all applications is similarity search: given a database of objects return the object(s) that are most similar to a given point according to a certain measure. Various approaches have been developed exploring a variety of concepts such as exhaustive search, hashing and indexing, space partitioning, or randomized algorithms. A review of most recent developments is described

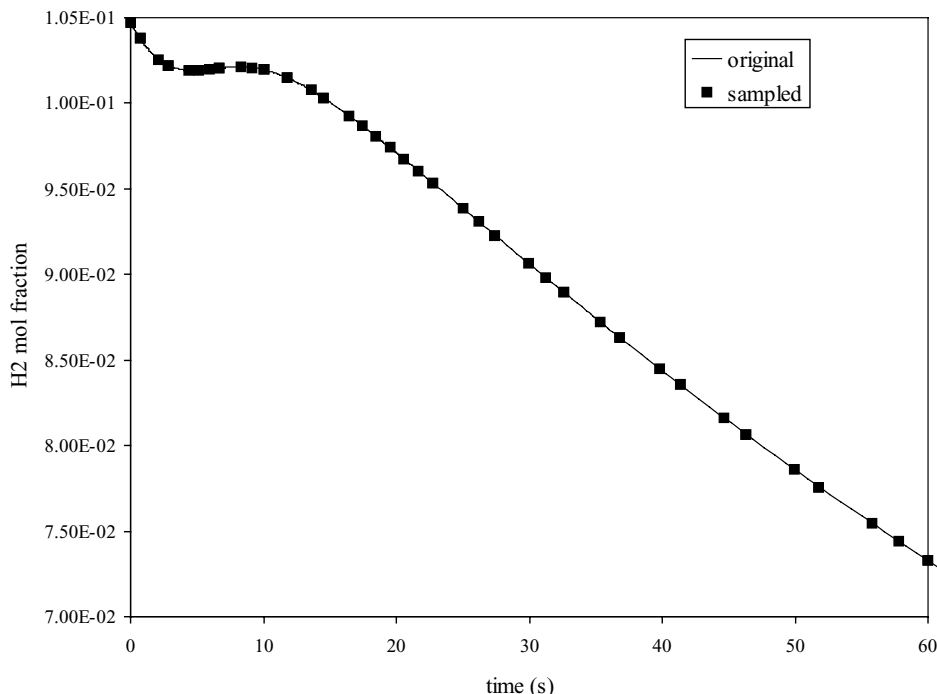


Fig. 5. Sample hydrogen combustion trajectory. The selection of points based on arc-length emphasizes infrequent events generating non uniform samples in time.

in Tsaparas (1999). Most solutions to the problem either create data structures that require storage space exponential in the dimensions, or require query time that is not much better than a linear scan of the data points. Therefore, numerous approaches have been proposed and the subject remains an active area of research.

Given a query point P_q the issue is to efficiently locate, in the database, which point(s) are closest in order to assign values for either S or dP/dt . In fact, the problem at hand has some interesting complexities associated with it in that the database is not a structured one. The raw data points are the ones defining the trajectories of the evolution of the original system. Therefore, it is very difficult, if at all possible, to identify a structured representation of the data that would enable a very efficient search. Therefore, we have to identify search and retrieve algorithms that can efficiently operate on highly unstructured data sets. A number of approaches can be considered.

1. Brute force method in which we estimate the distance (Euclidean) of all points in the database from the query point. Keep the points that are within ε from the query point, i.e., $d(P_p, P) \leq \varepsilon$, or keep the n closest points to P_q . Of course this is a very time consuming approach that scales as $O(N^2)$, where N is the size of the database.
2. The approach taken by Yang and Pope (1998) can also be explored. The original database is pre-processed and the singular values and vectors are calculated. Each entry is assigned values for the corresponding singular vectors. If we keep the number of singular vectors very small the search can be performed very efficiently. For example,

we can bound the singular value of a query point in $O(\log_2(N))$ which is extremely fast. In fact if we wish to bound the query point between the values $Low \leq P_q \leq Upp$ such that $(Upp - Low) = \alpha(Max - Min)$, then for given α the number of operations is independent of the size of the database (N) if the values are sorted and a 1D binary search is implemented (Fig. 6).

3. In order to avoid the projection errors that could potentially be introduced by the forward and backward projection, to and from the space of singular values, we will implement an efficient search in order to identify the set of nearest neighbors to the query point in high dimensions.

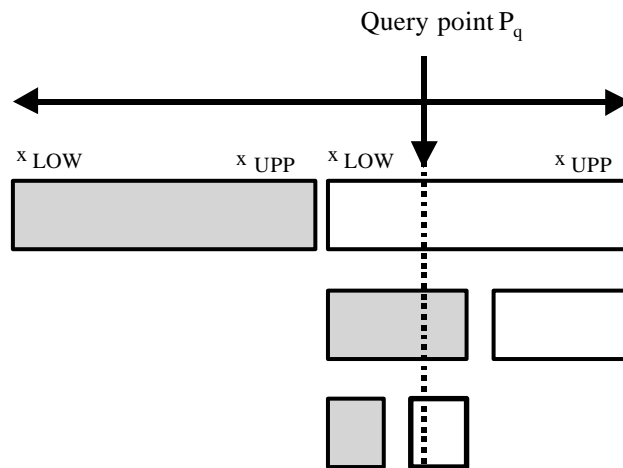


Fig. 6. 1D binary search for bounding a query point.

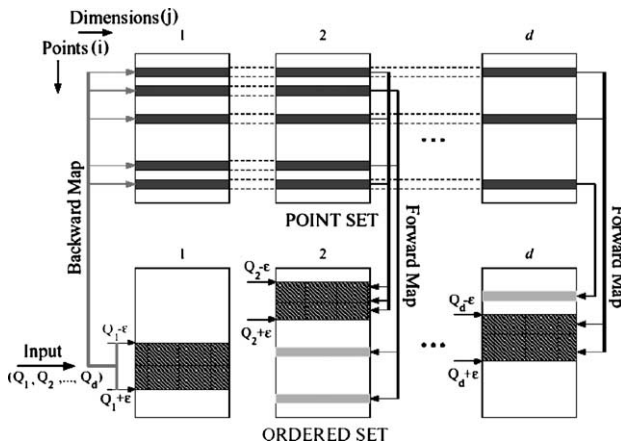


Fig. 7. Data structures used for constructing and trimming the candidate list (Nene & Nayar, 1995).

The algorithm that we implement was proposed by Nene and Nayar (1995, 1997). The central idea is to bound the query point by successively trimming a list of candidate points. The algorithm attempts to identify a cube of size 2ϵ around the query point, as shown in Fig. 8. The algorithm operates in unstructured data. The only provision is that the original database is transformed is stored as collection of 1D arrays such that the j th array contains the j coordinate of the points sorted in j . This is a process that takes place only once. Given that very efficient algorithms for sorting exist (Knuth, 1998), this is an overhead that is bearable.

Once the data are sorted two maps are constructed. The *backward map* maps a coordinate in the ordered set to the corresponding coordinate in the original set and, conversely, the *forward map* maps a coordinate in the point set to a coordinate in the ordered set (Fig. 7). As mentioned earlier, in order $O(\log_2(N))$ the coordinates in the ordered set that lie between the parallel hyper-planes positioned at $Q_1 - \epsilon$ and $Q_1 + \epsilon$ are efficiently identified as shown in Fig. 8. Using the backward map, we find the corresponding points in the ‘point set and add appropriate points to the candidate list. Next we trim the candidate list by iterating through $k = 2, 3, \dots, d$ by checking, at each iteration, every point in the candidate list, by using the forward map, to see if its k^{th} coordinate lies in the limits $Q_1 - \epsilon$ and $Q_1 + \epsilon$ (Fig. 7). The complexity analysis of the search and a number of additional observations are discussed in Nene and Nayar (1997).

The algorithm has been implemented by the author of this manuscript and should not be considered optimized.

3.2.7. Generating non-empty sets of near neighbors

It is apparent that the algorithm strongly depends on the choice of ϵ . Nene and Nayar (1997) present a number of approaches for estimating adequate bounds on ϵ so that a non-empty set of neighbors is identified. We are implementing the value that corresponds to the construction of the smallest hyper-cube (Nene & Nayar, 1997). More

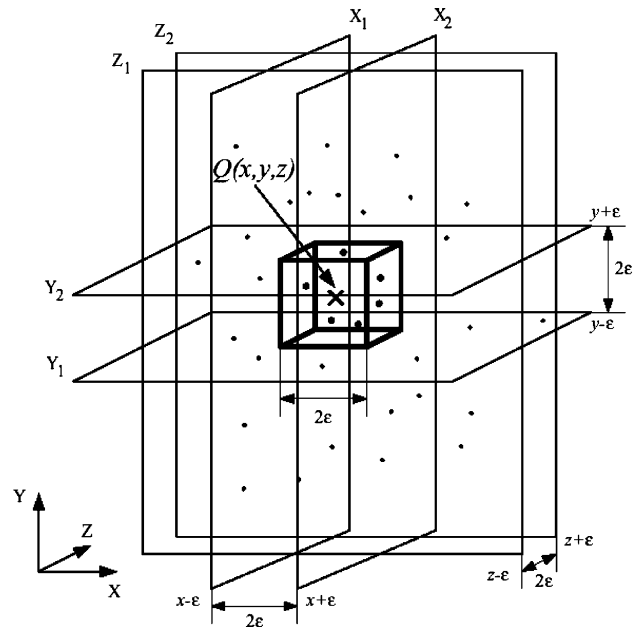


Fig. 8. Location of nearest neighbors (Nene & Nayar, 1995).

specifically the value of ϵ is estimated as

$$\epsilon = \frac{1}{2}L(1 - (1 - p)^{1/n})^{1/d}$$

where L is the range of the variable ($L = |\max - \min|$), n the number of data points, d the dimensionality, and p is probability that the domain to be determined will be non-empty. In the current implementation an iterative check is performed to guarantee that a minimum number of points is inside the neighborhood. If, for a given value of ϵ , the number is smaller, the value of ϵ is increased by a factor α .

Overall, the algorithm scales very nicely as seen by the results depicted in Fig. 9. The average, relative, CPU requirement for determining a non-empty neighborhood as

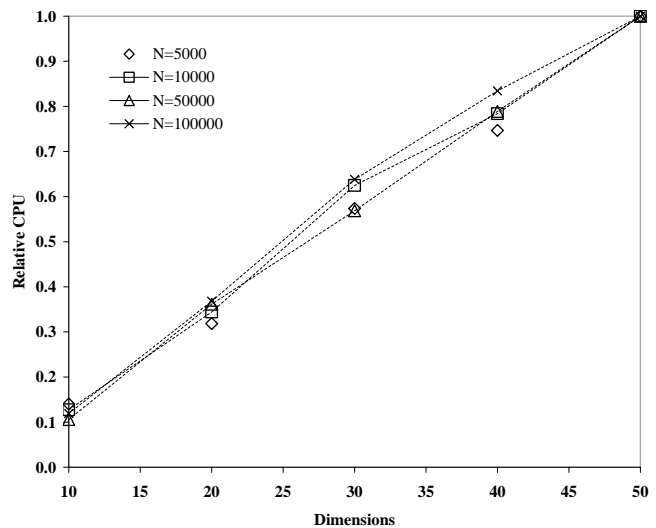


Fig. 9. Relative execution time for searching for the set of nearest neighbors.

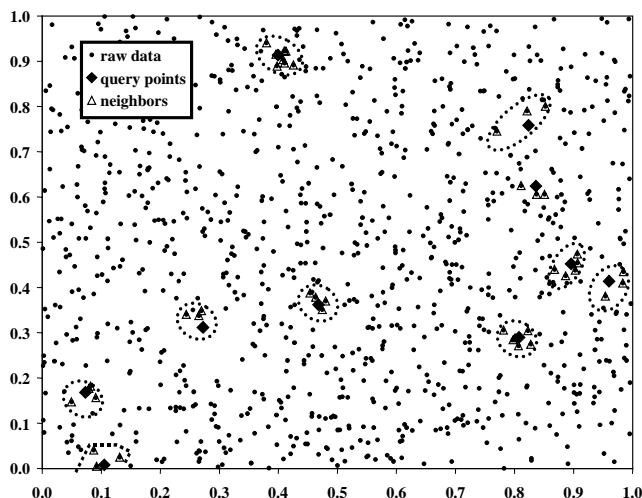


Fig. 10. Neighborhood definition for randomly generated points on the plane.

a function of the number of records (N), and the number of dimensions is presented. The search scales linearly with the size of the database. A pictorial representation of the neighborhoods are shown in Fig. 10 for uniformly generated points

A very important point has to be made at this time. The construction of the algorithm is based on the assumption that the L_∞ norm is used as opposed to the often more meaningful L_2 norm (Euclidean distance). Candidate trimming list is performed based on the L_∞ norm. As a result points which are closer to the query point in the L_2 norm could be rejected (Fig. 11). A number of remedies were discussed in Nene and Nayar (1997), however, for our purposes, this not so important as it will be shown next that the absolute-nearest point does not represent the query point best.

3.2.8. Assigning properties to the query point

Once the query point has been enclosed by points already in the database values for all the required quantities must be assigned to it. The simplest approach would be to augment the nearest neighbor search with an additional step that finds the point closest to the query point and assign its properties

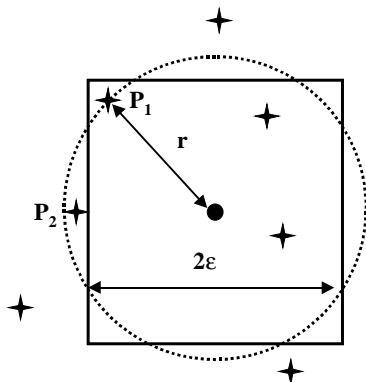


Fig. 11. L_2 vs. L_∞ norms in defining the neighborhood.

to it. However, given that the final list of points is rather short an improved approach is to linearly interpolate the points belonging to the neighborhood of the query point and use the interpolation scheme to predict the properties of it. Linear interpolation is as good as any other one given the fact that, by design, the neighborhood is quite restricted. Standard algorithms can be used for that and it is a very efficient computational step.

It was empirically determined that averaging the property values in a neighborhood of the query point can potentially provide an advantage over simply assigning to the query point the property value of the point closest to it (Fig. 12).

4. Motivating examples

4.1. Definition of motivating examples

4.1.1. Two-phase CSTR (Rhodes et al., 1999)

This example is taken from Rhodes et al. (1999). It analyzes the dynamics of a two-phase reactor having a pure gas feed consisting of chemical component A and a pure liquid feed of chemical component B (Fig. 13). A reaction $A + B \rightarrow C$ takes place in the reactor. Components A and C exist in both the liquid and gas phase, whereas component B remains purely in the liquid phase. A detailed description of the reactor along with all the necessary parameters for the simulations are presented in great detail in Rhodes et al. (1999). The dynamics of the reactor are defined by the following set of equations:

$$\begin{aligned} \frac{dN_{Ag}}{dt} &= F_{A0} - N_A - F_G y_A \\ \frac{dN_{Cg}}{dt} &= N_C - F_G y_C \\ \frac{dN_{Al}}{dt} &= N_A - k C_A C_B V_L - F_L x_A \\ \frac{dN_{Bl}}{dt} &= F_{B0} - k C_A C_B V_L - F_L x_B \\ \frac{dN_{Cl}}{dt} &= N_C + k C_A C_B V_L - F_L x_C \end{aligned} \quad (3)$$

The fairly wide range of initial conditions was sampled and equivalent simulations of the original system are used to generate a database with 40,000 points. The detailed trajectories are sampled according to the procedure we outlined earlier. The primary species are selected to be N_{Ag} , N_{Bl} , N_{Cl} consistent with the selection of Rhodes et al. (1999).

4.1.2. Pollution Kinetics (Marsden, Frenklach, & Reible, 1997; Verwer, 1994)

The second test case is a combination of prior work in the area of modeling atmospheric quality models. A standard "Box Model" (Marsden et al., 1987) is employed to account for the processes of reaction and diffusion in atmospheric

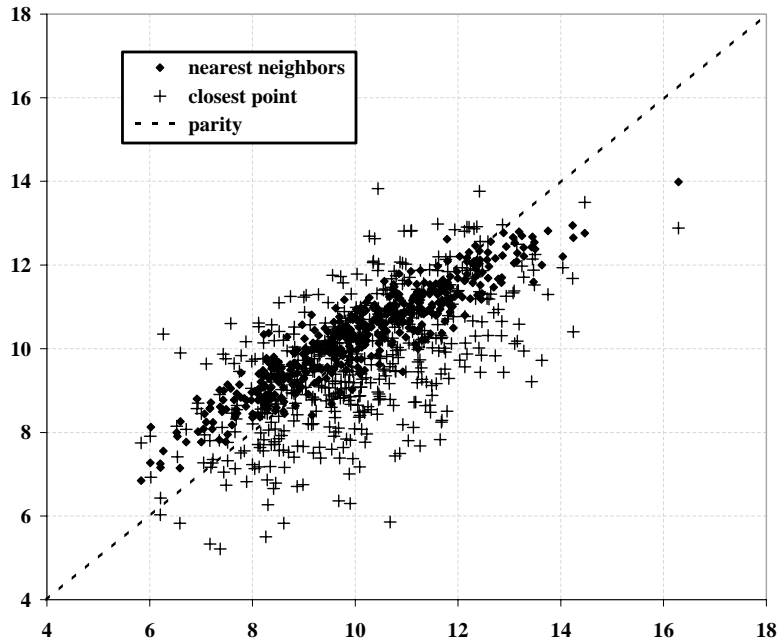


Fig. 12. Property predictions for synthetic test data. The identity relation is retrieved based on previously generated sparse and unstructured data. For a given query point property values are determined by reporting the properties of the closest point, and an average of the set of nearest neighbors. The parity line is plotted for comparison purposes. Average of nearest neighbors results in reduced scatter and better correlation with query point.

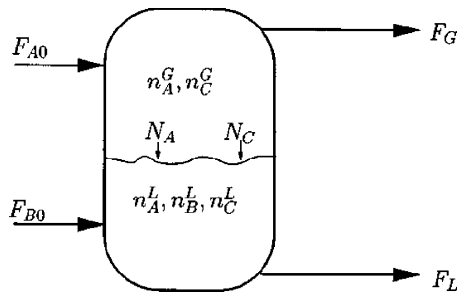


Fig. 13. Two-phase CSTR (Rhodes et al., 1999).

modeling. The Box Model is a simplified model that attempts to account for pollutant emission and dilution to transport in a simplified way. The atmospheric chemistry model discussed in Verwer (1994) was employed to model the reaction mechanism (Fig. 14). The constitutive equations defining the material balances are defined in Eq. (4). The assumption is that the pollutants emitted by various sources are NO, NO₂, CO, and ALD. Typical values for the emission rates are based on the values reported by Marsden et al. (1987).

$r_1 = k_1 \cdot y_1$	$r_{10} = k_{10} \cdot y_{11} \cdot y_1$	$r_{19} = k_{19} \cdot y_{16}$
$r_2 = k_2 \cdot y_2 \cdot y_4$	$r_{11} = k_{11} \cdot y_{13}$	$r_{20} = k_{20} \cdot y_{17} \cdot y_6$
$r_3 = k_3 \cdot y_5 \cdot y_2$	$r_{12} = k_{12} \cdot y_{10} \cdot y_2$	$r_{21} = k_{21} \cdot y_{19}$
$r_4 = k_4 \cdot y_7$	$r_{13} = k_{13} \cdot y_{14}$	$r_{22} = k_{22} \cdot y_{19}$
$r_5 = k_5 \cdot y_7$	$r_{14} = k_{14} \cdot y_1 \cdot y_6$	$r_{23} = k_{23} \cdot y_1 \cdot y_4$
$r_6 = k_6 \cdot y_7 \cdot y_6$	$r_{15} = k_{15} \cdot y_3$	$r_{24} = k_{24} \cdot y_{19} \cdot y_1$
$r_7 = k_7 \cdot y_9$	$r_{16} = k_{16} \cdot y_4$	$r_{25} = k_{25} \cdot y_{20}$
$r_8 = k_8 \cdot y_9 \cdot y_6$	$r_{17} = k_{17} \cdot y_4$	
$r_9 = k_9 \cdot y_{11} \cdot y_2$	$r_{18} = k_{18} \cdot y_{16}$	

1. NO ₂ → NO+O ₃ P	14. NO ₂ +OH → HNO ₃
2. NO+O ₃ → NO ₂	15. O ₃ P → O ₃
3. HO ₂ +NO → NO ₂ +OH	16. O ₃ → O ₁ D
4. HCHO → 2 HO ₂ +CO	17. O ₃ → O ₃ P
5. HCHO → CO	18. O ₁ D → 2 OH
6. HCHO+OH → HO ₂ +CO	19. O ₁ D → O ₃ P
7. ALD → MEO ₂ +HO ₂ +CO	20. SO ₂ +OH → SO ₄ +HO ₂
8. ALD+OH → C ₂ O ₃	21. NO ₃ → NO
9. C ₂ O ₃ +NO → NO ₂ +MEO ₂ +CO ₂	22. NO ₃ → NO ₂ +O ₃ P
10. C ₂ O ₃ +NO ₂ → PAN	23. NO ₂ +O ₃ → NO ₃
11. PAN → C ₂ O ₃ +NO ₂	24. NO ₃ +NO ₂ → N ₂ O ₅
12. MEO ₂ +NO → CH ₃ O+NO ₂	25. N ₂ O ₅ → NO ₃ +NO ₂
13. CH ₃ O → HCHO+HO ₂	

Fig. 14. Reaction rates and reaction scheme definition (Verwer, 1994).

It must be emphasized that the target of the example is neither to discuss the accuracy of the atmospheric chemistry, nor the validity of the Box Model in terms of capturing all the appropriate physical and chemical events. Our goal is to determine whether an *off-line* generation of the implicit model is plausible and to examine the accuracy of the search as defined earlier. The four pollutant species are treated as the primary variables and in a number of *off-line* simulations their net rates of production, as defined in Eq. (4), are stored as in terms of their corresponding concentration values. A total of 50,000 records was generated and will be used for searching based on typical ranges of values provided in Marsden et al. (1987). The parameter W , D and L determine the rate of change of the “box”, U and D simulate the diffusion in and out of the “box”. The details can be found in (Marsden et al., 1987) and are omitted here.

$$\frac{dC_i}{dt} = \frac{KQ_i}{WD(L_0 + \alpha t)} - \frac{aC_i}{L_0 + \alpha t} - \frac{UC_i}{D} + R_i,$$

$$C_i \in \mathfrak{R}^{20}, 0 \leq t \leq 60$$

$$R = \begin{pmatrix} -\sum_{j \in \{1,10,14,23,24\}} r_j + \sum_{j \in \{2,3,9,11,12,22,25\}} r_j \\ -r_2 - r_3 - r_9 - r_{12} + r_1 + r_{21} \\ -r_{15} + r_1 + r_{17} + r_{19} + r_{22} \\ -r_2 - r_{16} - r_{17} - r_{23} + r_{15} \\ -r_3 + 2r_4 + r_6 + r_7 + r_{13} + r_{20} \\ -r_6 - r_8 - r_{14} - r_{20} + r_3 + 2r_{18} \\ -r_4 - r_5 - r_6 + r_{13} \\ r_4 + r_5 + r_6 + r_7 \\ -r_7 - r_8 \\ -r_{12} + r_7 + r_9 \\ -r_9 - r_{10} + r_8 + r_{11} \\ r_9 \\ -r_{11} + r_{10} \\ -r_{13} + r_{12} \\ r_{14} \\ -r_{18} - r_{19} + r_{16} \\ -r_{20} \\ r_{20} \\ -r_{21} - r_{22} - r_{24} + r_{23} + r_{25} \\ -r_{25} + r_{24} \end{pmatrix} \quad (4)$$

4.2. Computational results

As mentioned earlier, two approaches can be defined in order to build the surrogate integration model. In the first ap-

proach, we generate and tabulate pairs of $(P, (dP/dt))$. Essentially, this approach completely eliminates the secondary species and only the time evolution of the primary species is monitored and recorded. Typical results, for randomly generated initial conditions within the range of initial conditions used to build the approximation are shown in Fig. 15.

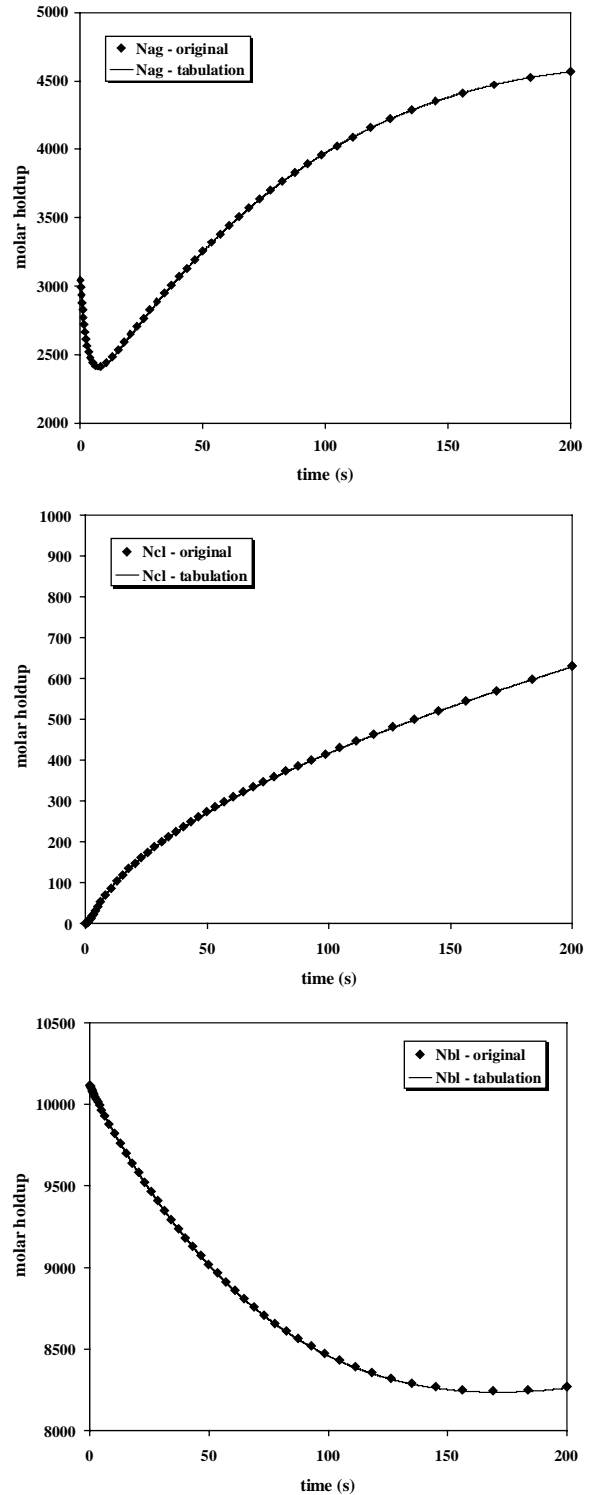


Fig. 15. Two-phase CSTR simulation predicting Nag, Nel and Nbl using $(P, dP/dT)$ tabulation. Original refers to the full model.

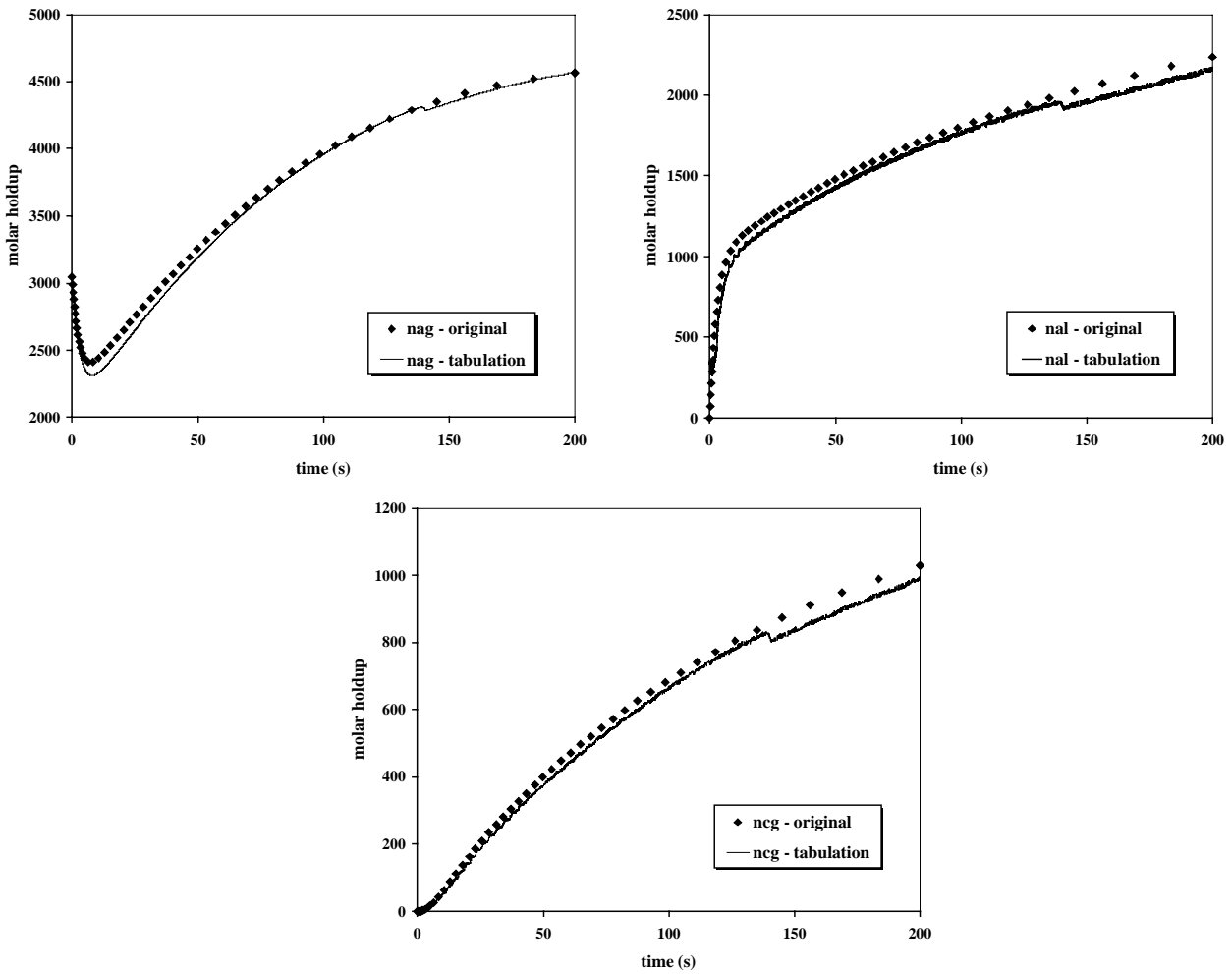


Fig. 16. 2-phase CSTR simulation predicting Nag, Ncl and Nbl using the (P, S) tabulation. Original refers to the full model.

The alternative representation involves the tabulation of the (P, S) values. In that cases the tabulated values of the secondary species are used in conjunction with the original kinetic model to determine the values of (dP/dt) . The advantage of this approach is that the values of secondary

species can be recovered as well (Fig. 16). Typical results for the pollution problem are also depicted in Fig. 17.

In general the agreement between the integration of the original model, and the results obtained using the search-and-retrieve approach are almost identical. Therefore the

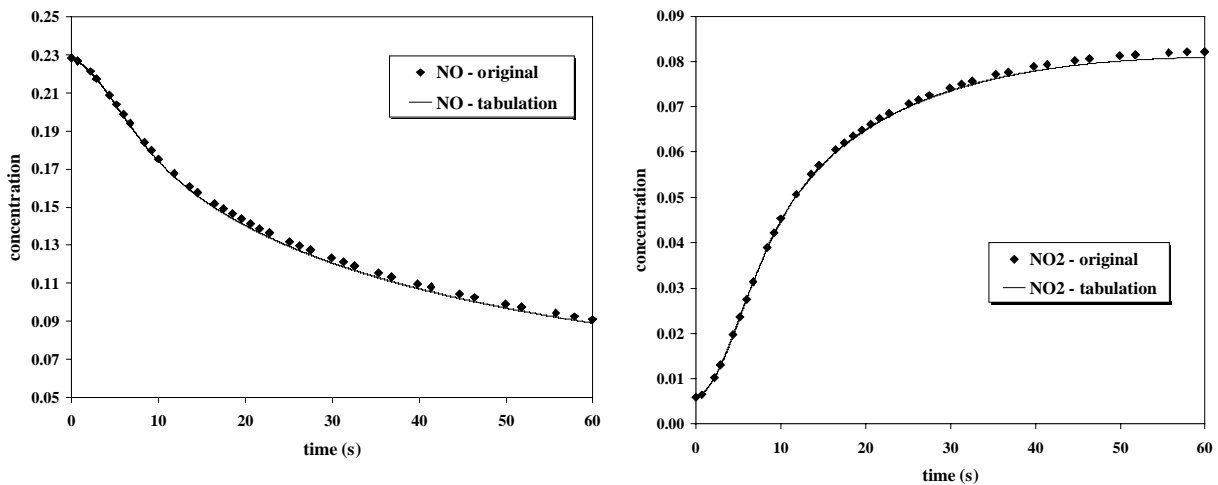


Fig. 17. Pollution problem using the $(P, dP/dt)$ tabulation.

interpolating accuracy of the approach is excellent. However, a number of issues remain to be explored further and these are discussed in greater detail in the following section.

5. Conclusions and future work

In this work, an approach for deriving approximate models for the integration of dynamic systems based on store-and-retrieve methods was presented. The basic premise of the approach is that a number of simulations can be performed off-line. The simulation results are then tabulated and can be efficiently searched in order to retrieve the necessary information for the integration of a reduced dimensionality system. The power of the approach is based on the advantages offered by a novel searching technique that identifies in a computationally efficient way a tight neighborhood around a query point. The test cases showed that indeed the accuracy of the approach rivals that of the integration of the original, detailed, representation.

Despite the very good agreement however, a number of issues still remain to be further explored.

1. *Partition to primary (P) and secondary (S) species:* An *ad hoc* partition is performed in order to define the *P* and *S* sets of species. Although one can argue that a reasonable selection would be that $P = \{\text{set of measurable}^1 \text{ quantities}\}$, $S = \{\text{set of non-measurable quantities}\}$. Species with short lifetime can definitely be part of *S*. In practical applications though one may wish to increase the number of species in *S* so as to reduce the computational burden in terms of the number of *P* variables. Therefore the set *S* may need to be further augmented. However, what is not clear is whether the maps (*P*, dP/dt) and/or (*P*, *S*) are unique. In other words, if similar *P* values could correspond to dramatically differing values of dP/dt or *S*. It should be pointed out, however, that in principle this is the same theoretical problem that the ILDM approach faces.
2. *Approach has interpolating capabilities only:* It should be rather obvious that, like any other approach that attempts to build a model based on data, the search-and-retrieve approach has very strong, if not excellent, interpolating capabilities, but rather weak, if any, extrapolating capabilities.
3. *Discontinuity of the model:* The right-hand side of the integrated equations is determined by searching the stored values in the database. Therefore, the evaluation of it is by definition discontinuous. As a result, the rate of convergence of the integration is adversely affected.
4. *Interpolating model:* Given a query point, a neighborhood around it is identified. Subsequently, the necessary

quantities are identified by linearly interpolating the corresponding values of this neighborhood. It should be further explored whether this linear interpolation is the most computationally efficient and accurate way of assigning the query point properties.

5. *Neighborhood identification:* As was discussed earlier, it is very difficult to identify *a priori* the minimal non-empty hypercube that contains the query point. It was earlier indicated that the difference between the L_2 and L_∞ norms is such that it can create problems in properly determining the list of nearest points.
6. *Convergence:* As indicated in (3) earlier, convergence of the numerical integration scheme could potentially be adversely affected by the discontinuity of the model. Computational results up to this point indicate that although the integrators always converge, the number of steps is required is larger than when the original system is integrated. The advantage of course is that a much smaller system is integrated.

However, we believe that despite the issues that were just raised, there is significant merit in further analyzing “store and retrieve” computational approaches. Recent advances in building and searching databases will provide significant impetus and will further enhance such types of computational approaches. Once the issues just raised are resolved, these methods could significantly boost our capability to perform computations using a combination of reduced representations and stored information.

Acknowledgements

The author would like to thank Prof. S.K. Nayar (Department of Computer Science, Columbia University) for allowing the reproduction of Figs. 5 and 6.

References

- Androulakis, I. P. (2000). Kinetic mechanism reduction based on an integer optimization approach. *AIChE Journal*, *46*, 361–371.
- Buki, A., Perger, T., Turanyi, T., & Maas, U. (in press). Repro-modeling based generation of intrinsic low-dimensional manifolds. *Journal of Mathematics and Chemistry*.
- Chen, J. Y. (1988). A general procedure for constructing reduced reaction mechanisms with given independent reactions. *Combustion Science Techniques*, *57*, 89–94.
- Come, G. M., Warth, V., Clause, P. A., Fournet, R., Battin-Leclerc, F., & Scacchi, G. (1997). Computer-aided design of gas phase oxidation mechanism: Application to the modeling of *n*-heptane and *iso*-octane oxidation. In *Proceedings of the 26th Symposium (International) on Combustion* (pp. 755–762). The Combustion Institute.
- Curran, H. J., Pitz, W. J., Westbrook, C. K., Callahan, C. V., & Dryer, F. L. (1998). Oxidation of automotive primary reference fuels at elevated temperatures. In *Proceedings of the 27th Symposium (International) on Combustion* (pp. 379–387). The Combustion Institute.
- Duncan, B., Portman, D., Bay, I., & Spivakovsky, C. (2000). Parameterization of OH for efficient computation in chemical tracer models. *Journal of Geophysics and Research*, *105*, 12259–12262.

¹ Measurable does not only imply that the quantity can be physically measured. It rather implies that there is some interest in actually measuring it. For example, *P* can be a reactant, a product, or a dominant intermediate.

- Faravelli, T., Gaffuri, P., Ranzi, E., & Griffiths, J. F. (1998). Detailed thermokinetic modeling of alkane auto-ignition as a tool for the optimization of performance of internal combustion engines. *Fuel*, *3*, 147–155.
- Green, W. H., Barton, P. I., Bhattacharjee, B., Matheu, D. M., Schwer, D. A., & Song, J. et al., (2001). Computer construction of detailed chemical kinetic models for gas phase reactions. *Industrial Engineering and Chemical Research*, *40*, 5362–5370.
- Graham, M. D., & Kevrekidis, I. G. (1996). Alternative approaches to the Karhunen–Loeve decomposition for model reduction and data analysis. *Compound Chemical Engineering*, *20*, 495–506.
- Kee, R. J., Rupley, F. M., Meeks, E., & Miller, J. A. (1996). *Chemkin-III: A fortran chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics*. Sandia Report, SAND96-8216.
- Kirby, M., & Miranda, R. (1999). Empirical dynamical system reduction: Global nonlinear transformations. *Centre de Recherches Mathematiques, CRM Proceedings and Lecture Notes*, *20*, 41–63.
- Knuth, D. E. (1998). *The art of computer programming*. Addison-Welsey.
- Kramer, M. A. (1991). Nonlinear principal component analysis using auto-associative neural networks. *AIChE Journal*, *37*, 233–243.
- Lam, S. H., & Goussis, D. A. (1988). Understanding complex chemical kinetics with computational singular perturbation. In *Proceedings of the 22nd Symposium (International) on Combustion* (pp. 931–941). The Combustion Institute.
- Marsden, A. R., Frenklach, M., & Reible, D. D. (1987). Increasing the computational flexibility of urban air quality models that employ complex chemical mechanisms. *JAPCA*, *37*, 370–376.
- Mass, U. A., & Pope, S. (1992). Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Combustion Flame*, *88*, 239–264.
- Nene, S. A., & Nayar, S. K. (1995). *A simple algorithm for nearest neighbor search in high dimensions*. Technical Report No. CUCS-030-95, Department of Computer Science, Columbia University.
- Nene, S. A., & Nayar, S. K. (1997). A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 989–1003.
- Niemann, H., Schmidt, D., & Maas, U. (1997). An efficient storage scheme for reduced chemical kinetics based on orthogonal polynomials. *Journal of Engineering and Mathematics*, *31*, 131–142.
- Petzold, L. R., & Zhu, W. (1999). Model reduction for chemical kinetics: An optimization approach. *AIChE Journal*, *45*, 869–886.
- Rabitz, H., Alis, O. F., Shorter, J., & Shim, K. (1999). Efficient input–output model representations. *Compound Physics and Communication*, *117*, 11–20.
- Rhodes, C., Morari, M., & Wiggins, S. (1999). Identification of low manifolds: Validating the algorithm of Maas and Pope. *CHAOS*, *9*, 108–123.
- Seshadri, K., & Peters, N. (1990). The inner structure of methane-air flames. *Combustion Flame*, *81*, 96–118.
- Shah, J. J., & Fox, R. O. (1999). Computational fluid dynamics simulation of chemical reactors: Application of in situ adaptive tabulation to methane thermochlorination chemistry. *Industrial Engineering and Chemical Research*, *38*, 4200–4212.
- Susnow, R. G., Dean, A. M., Green, W. H., Peczak, P. K., & Broadbent, L. J. (1997). Rate-based construction of kinetic models for complex systems. *Journal of Physics and Chemistry, Part A*, *101*, 3731–3740.
- Tomlin, A. S., Turanyi, T., & Pilling, M. J. (1997). Mathematical tools for the construction, investigation and reduction of combustion mechanisms. In M. J. Pilling (Ed.), *Comprehensive chemical kinetics. Low-temperature combustion and autoignition* (Vol. 35).
- Tonse, S. R., Moriarty, N. W., Brown, N. J., & Frenklach, M. (1999). PRISM: Piecewise reusable implementation of solution mapping: An economical strategy for chemical kinetics. *Israel Journal of Chemistry*, *39*, 97–106.
- Tsaparas, P. (1999). *Nearest neighbor search in multidimensional spaces*. Technical Report 319-02, Department of Computer Science, University of Toronto.
- Turanyi, T. (1994). Application of repro-modeling for the reduction of combustion mechanisms. In *Proceedings of the 25th Symposium (International) on Combustion* (pp. 949–955). The Combustion Institute.
- Verwer, J. G. (1994). Gauss–Seidel iteration for stiff ODEs from chemical kinetics. *SIAM Journal of Science and Computing*, *15*, 1243–1259.
- Warnatz, J., Maas, U., & Dibble, R. W. (1996). *Combustion: Physical and chemical fundamentals, modeling and simulation, experiments, pollutant formation*. Springer.
- Yang, B., & Pope, S. B. (1998). Treating chemistry in combustion with detailed mechanisms—in situ adaptive tabulation in principal directions—premixed combustion. *Combustion Flame*, *112*, 85–112.