Intrinsic Low-Dimensional Manifold Method for Rational Simplification of Chemical Kinetics

University of Notre Dame
Department of Aerospace and Mechanical Engineering

prepared by: Nicholas J. Glassmaker

prepared for: ME 499 - Undergraduate Research
Joseph M. Powers

May 9, 1999

Abstract

The problem of avoiding excessive computational time when finding the complete numerical solution to the full system of ordinary differential equations describing the kinetics of a combustion system is addressed. The intrinsic time scales and stiffness of these equations are both discussed in detail. Three methods that simplify the kinetics description and thus reduce computational time are presented. The quasi-steady state and partial equilibrium techniques are described briefly. The intrinsic low-dimensional manifold method is described in detail and applied to an example problem for a simple mechanism describing the formation of NO in a well-stirred, spatially uniform, isothermal, isochoric combustion system. The complete numerical solution is also obtained for this example in order to demonstrate the accuracy and usefulness of the intrinsic low-dimensional system.

1 Introduction

Numerical computations to solve the chemical kinetics equations of combustion systems for species concentration evolution often require significant computational time. This can be true for even the simplest models. When the kinetics equations are solved in addition to conservation equations for more complex flow models, the computational time may become excessive and impractical [1]. Numerical solutions to such flows are desired in order to better understand these phenomena. Finding such numerical solutions in reasonable computational time often requires mathematical simplifications that reduce the number of computations required to describe chemical kinetics. While these simplifications facilitate the numerical solution, it is important that they give as accurate and complete a description of the reaction kinetics as possible.

This paper explores three techniques that are used to simplify chemical kinetics. The partial equilibrium and quasi-steady state techniques are given brief, qualitative treatments. The focus is on the intrinsic low-dimensional manifold (ILDM) technique. This technique is more rigorous than the other two in that it requires no initial assumptions about the progress of any of the reactions or the state of any of the species of the system.

The paper contains three main sections. Section 2 details why simplification techniques are necessary for combustion kinetics systems and presents a review of each technique. Section 3 describes the ILDM technique in detail. The reasoning behind the method and its importance for simplifying chemical kinetics calculations is explained. Necessary definitions are given, and the mathematics involved is presented. A scheme for implementing the technique is also given. Finally, Section 4 formulates a combustion kinetics problem and presents the results of applying the ILDM technique to it. Specifically, the one dimensional manifold is found for a simple mechanism for the formation of NO. The mechanism contains four reactions and five species and is part of the Zeldovich mechanism [2].

2 Review

Simple well-stirred, spatially uniform, isothermal, isochoric combustion systems are considered herein. Although such systems are difficult to realize physically, they have pedagogical value and are useful as representations of rational limiting cases of problems that are more common in the physical world. The idea is to gain understanding of the simplest systems in order to facilitate the solution of more physically relevant models in other pursuits. For these simple combustion problems, the evolution of n_s species concentrations is described by a system of n_s nonlinear ordinary differential equations (ODEs).

It should be noted that the best description of reaction kinetics results from detailed reaction mechanisms that include as many of the species and elementary reactions known to occur physically in the real system as possible. As more precision is obtained, mechanisms become increasingly complex. In many cases, complete numerical solutions of the full system of ODEs describing chemical kinetics in the combustion system requires more computational time as more detail is included in the mechanism. Sometimes even systems with simple reaction mechanisms require large amounts of computational time to solve the kinetics equations, though. It is the stiffness, not size, of the ODE system which mostly determines the computational time required to completely solve it. This paper explores methods that simplify the kinetics description from the complete solution of the ODEs and in turn reduce the amount of computational time.

2.1 Numerical Challenges

When solving a system of ODEs, it is often the case that the presence of nonlinearities necessitates a numerical solution. This is the case for the combustion systems considered here. Numerical solutions are less desirable than analytical ones for several reasons. A numerical solution does not provide an exact closed functional form. Rather, only a subset of points from the solution space is found. Moreover, this subset is approximate, and there are numerical errors associated with calculation of these solution points [11]. Although correlations may occasionally be applied that fit data points from numerical solutions, the exact closed functional form provided by an analytical solution is always preferred. In addition to the limited nature of the solution set and error associated with the points found, there may be problems of numerical convergence to the correct solution points depending on the method used.

All of the problems in the previous paragraph are important to consider when solving a system of differential equations numerically but are not considered in this paper. A point related and relevant to several of the problems is the focus here. For a reaction mechanism with a certain degree of detail, the character of the solution and numerical convergence requirements for a desired range of solution points contribute to fixing the required computational time to find correct solutions within specified error values at each point. Specifically, there are several characteristic modes associated with the solution of a nonlinear ODE at each value of the independent variable as the solution evolves. These modes depend on the independent variable, and each mode affects the dependent variable's evolution at a different rate. Each of these rates corresponds to a time scale for the system.

In order to attain convergence to the actual solution when solving the entire system numerically, the distance between the discrete points of the numerical solution must be small enough to capture the intrinsic variation in the solution. This means that the resolution of the numerical grid (in terms of the independent variable) must be small enough to show changes due to all time scales. For a detailed reaction mechanism, it is often the case that the time scales of the solution differ by several orders of magnitude. Because calculations must be performed at grid points that are close enough to capture the smallest time scale effects on the solution at each point, it may take many calculations before the effects of the large time scales are noticed.

For combustion systems (which are assumed to be stable), it is desirable to know how the solution evolves from an initial state to an equilibrium state. The equilibrium state occurs when a further increase in the independent variable causes no more changes in the dependent variable. Because the solution decays exponentially to approach this state, equilibrium is only reached as time approaches infinity. The time scales describe different rates of decay that characterize the solution. Small time scale resolution is required for numerical calculation of the solution points near the initial conditions because the fast time scales are the dominant cause for variation in the solution in the early time regime. Many computations are required before the effects of the large time scale terms are noticed to decay if this resolution is maintained. When enough time has passed such that changes in the solution due to all rates of decay approach zero, the value of the dependent variable approaches the equilibrium state. Thus, to describe the solution of a system from the initial state to the equilibrium state requires many computations. It is desirable to find ways to reduce the number of calculations while still preserving an accurate and complete description of the solution to the system.

There is a trade-off between computational time and completeness of the system's kinetics description. Detailed mechanisms with corresponding kinetics ODEs solved to resolve the fastest time scales provide the most accurate and complete descriptions of species evolution. However, for stiff systems, this requires much computational time. The goal of this paper is to find methods for solving the species concentration equations in minimum computational time while retaining the accurate description provided by a full solution to a system with a detailed reaction mechanism.

2.2 Simplification Techniques

There are several strategies for reducing calculations necessary to obtain the kinetics description of a combustion system. The quasi-steady state and partial equilibrium methods briefly described in Sections 2.2.1 and 2.2.2 below are similar because they simplify the kinetics description and consequently reduce calculations by making assumptions about the system behavior based on observations of physical characteristics of the system studied. In effect, these methods assume less detailed mechanisms based on observations of the behavior of known numerical solutions and physical systems. Numerical solutions of the full equations confirm the validity of these methods for specified conditions.

In Section 2.2.3, a brief introduction is given to the technique emphasized in this paper, the ILDM method. This technique makes no assumptions about the system behavior. Rather, it uses the intrinsic time scale characteristics of the system to predict simplified but accurate kinetics behavior at times when only the slow rates of decay associated with the solution are causing significant variations in the solution. A detailed presentation of the ILDM technique continues in Section 3.

2.2.1 Quasi-Steady State

The quasi-steady state (QSS) technique is a classical method for simplifying the solution to a system of kinetics equations. According to Turányi, et. al., [3], it was first applied to chemical kinetic schemes in 1913 by Bodenstein and by Chapman and Underhill. Like most simplification techniques, it is based on the assumption that many of the elementary reactions of a complex mechanism do not contribute significantly to the determination of the rate governing overall the combustion process [2]. Specifically, some of the species concentration ODEs are associated with radicals and other species that react very quickly. It may be assumed that the rates of consumption and production of these fast reactive species are approximately the same [2], [4], [5], [6]. This assumption is the essence of the QSS technique; the net rate of change of fast reactive species concentrations with time is assumed to be zero.

Because the differential equations describing the species concentration evolution of reactive species are assumed to be zero, the dimension n_s of the ODE system is reduced by the number of QSS assumptions n_q made. Each QSS assumption results in an algebraic relationship [4], [5], [7]. Chen gives a general method for finding these algebraic relationships [7]. Once obtained, the algebraic relationships may be used to formulate the remaining $n_s - n_q$ ODEs in terms of the remaining $n_s - n_q$ slow reactive species. By making the correct QSS assumptions, the stiffness of the system is reduced. When this lower-dimensional, reduced-stiffness system is solved, the algebraic equations are used to calculate the concentrations of the species assumed in QSS.

Although the QSS technique is logical, there are problems with applying it. Notably, for some systems the QSS assumption fails when applied directly to the system because it implies contradictory equalities [8]. A change of coordinates or "lumping" technique may be used to overcome the failure of the direct application of the QSS assumption to a system [8]. Another important problem is that the QSS is not usually accurate in low temperature regimes [5], [9]. Finally, choosing the correct species for the QSS assumption is not always straightforward. It is desirable that the correct species be assumed in QSS such that the error between the simplified and full system solutions is reduced to within an acceptable level and the computational time is minimized. This optimization is difficult to achieve because of the ambiguity associated with calculations of error for QSS approximations [3]. In addition, the correctness of choosing the QSS assumption for a certain species varies for different combustion systems [3]. One method for determining the correct QSS species is the computational singular perturbation (CSP) method [6]. Before this method, QSS species were generally chosen based on rational justifications from experience and intuition [6]. One can see the arbitrariness and uncertainty associated with such choices.

In spite of the problems mentioned, the QSS technique does provide a good approximation for the system behavior in many instances [2], [5], when proper species are chosen for the QSS assumption. In addition, if the system reaction mechanisms are simple enough, the QSS technique occasionally allows analytical solutions to systems which otherwise require numerical solutions [2].

2.2.2 Partial Equilibrium

The partial equilibrium (PEQ) technique is a classical kinetics simplification technique similar to the QSS technique. The goal of the two techniques is essentially the same. Both QSS and PEQ are used to find approximate algebraic relationships for the fast reactive species in terms of the slow reactive species. To find the algebraic relationships with the PEQ technique, certain elementary reactions are assumed to be in partial equilibrium. For this condition, their forward and backward reaction rate expressions may be equilibrated [2]. The PEQ assumption is valid for some elementary reactions at high temperatures but tends to fail at lower temperatures [2] as does QSS.

Like the QSS, the PEQ technique relies on experience and intuition. Here, one must choose which elementary reactions to assume in partial equilibria. Lam [6] suggests the CSP method as one way of determining the correct PEQ assumptions to make.

2.2.3 ILDM

While the quasi-steady state and partial equilibrium techniques provide good results for many systems and conditions when they are suitably formulated, a simplification is sought that will always provide correct results. Moreover, it is advantageous if a technique for reducing the calculations to find a solution incorporates as much of the information from the original system as possible. Further, it is also desirable if physical observations of the system characteristics and behavior are not necessary to specify the simplification. The ILDM method for reducing chemical kinetics calculations described in this section and developed in the next features these advantages over the other two techniques. The ILDM technique automatically extracts the required information from the full system description to describe the most essential and interesting details of the system's chemical kinetics. It does not rely on experience or intuition from the person applying it.

The ILDM method for combustion systems like the ones considered in this paper was first definitively documented by Maas and Pope [1] in 1992, but ideas about manifolds have been in the literature for much longer. Fenichel [10] presents important theory dealing with manifolds and perturbation theory. Most notably, when constructing simplified descriptions of the solution to ODEs in the phase space, it is desirable to see how small perturbation away from the simplified description will behave. Fenichel's global perturbation theory describes this situation [10], and it is demonstrated for combustion systems by Maas and Pope [1] and Duchêne [8]. The result obtained for systems assumed to have a stable manifold is that small perturbations in a certain region of the phase space surrounding the manifold are attracted toward the manifold [10]. Since the combustion systems considered by Maas and Pope, Duchêne, and this paper are stable in that they are attracted toward an equilibrium point, it may be shown that a low-dimensional stable attracting manifold exists and provides a good simplified solution description.

Note that the ILDM technique is more abstract than either of the QSS or PEQ techniques. In this regard, it is more difficult to understand and apply. Note also that the ILDM technique requires only the system and a desired dimension for the simplified solution as input parameters. No assumptions about the speed of reactions or rates at which species are consumed or produced are necessary. In this regard, the ILDM technique is more easily and accurately applied. Finally, note that there are no conditions for which the ILDM method fails as long as a stable attracting manifold exists. This makes the ILDM method more reliable than the QSS or PEQ techniques.

3 Performing the ILDM Technique

The detailed presentation of the ILDM technique follows. The definitions and mathematics needed are given first. Next, the method for applying the technique, numerical methods, and accuracy considerations are described. Finally, the ILDM technique is applied to a nonlinear system that models chemical kinetics of a simple mechanism for the formation of NO.

3.1 Definitions

Systems of ODEs with time as an independent variable are characterized by local sets of time scales. The idea of time scales in a system of ODEs describing reaction kinetics stems from the fact that the elementary reactions of combustion processes occur at different rates. The way to capture the different time scales due to the elementary reactions is to find the eigenvalues of the Jacobian of the ODE system [2]. The Jacobian contains the necessary numerical information about the linearization of the system at a point, and the solution of the linearized system describes the local behavior of the system near the point. The reciprocals of the real parts of the eigenvalues of the Jacobian are the time scales for the system behavior near the point; therefore, there are n_s time scales associated with the system, where n_s is the number of species ODEs. Note that if any eigenvalues have nonzero imaginary parts, the imaginary parts of the eigenvalues determine the frequency with which the solution oscillates at the point. Only the real parts of the eigenvalues influence the rate of exponential decay (stable systems) or growth (unstable systems) of the solution which defines the time scales. Note that because the Jacobian of the system changes with time, the time scales of the solution also evolve.

Having stated how to find the time scales associated with a system of nonlinear ODEs describing reaction kinetics, it is important to have an understanding of what these time scales represent. First, consider that at a given time, each species concentration is changing. The local solution describing how species concentrations will change with time has several terms. One time scale is associated with each term, and in general, all time scales affect each species evolution, although the amount that each term affects a given species concentration is not the same for each species. Some species concentrations will evolve more quickly than others for an initial period of time. The essence of the behavior of these species is governed by fast time scale terms for the initial period of time. When enough time has passed, the slower time scale terms begin to affect the behavior of the system. As time passes, the slower time scale terms dominate the evolution of each species concentration.

When studying detailed chemical kinetics, it is observed that in almost all cases, there is a range of time scales in the solution for the evolution of the different species involved. The ratio of the fastest to the slowest time scale of the species equations is called the stiffness of the system. Specifically, if the eigenvalues of the Jacobian of a system are given by $\lambda_1, \lambda_2, \ldots, \lambda_{n_s}$ the stiffness S_λ is given by

$$S_\lambda = \frac{|Re(\lambda)|_{max}}{|Re(\lambda)|_{min}}. \tag{1}$$

In Equation (1), Re represents the real part. Thus, to find the stiffness, first find the real part of each eigenvalue. Then find the magnitude of each of these real parts. Finally, determine the maximum and minimum of these magnitudes. Then use Equation (1) to find the stiffness, S_λ.

The ratio S_λ is large for stiff systems. For stiff systems that have many equations with many terms, the computational time to fully solve all the kinetics ODEs numerically can become excessive. The reason for this is that all time scales must usually be resolved when solving the full system to ensure numerical convergence. This resolution requires small time increments in the solution of the kinetics ODEs. In order to make the solution more computationally tractable, one may change the system by reducing the mechanism as is done in the quasi-steady state and partial equilibrium techniques (Sections 2.2.1 and 2.2.2). Another scheme is to identify the most important part of the solution and solve for this part from the full system description. This is the scheme employed by the ILDM technique.

When the slowest and fastest time scales of a solution differ by several orders of magnitude, the most important information about the system behavior is often contained in the terms of the solution with the slower time scales. At least this behavior is most interesting, because it is often possible to experimentally observe these details of the system response while some of the faster time scale solutions are not presently capable of being observed in the laboratory. The intrinsic variations in the system response due to the fast time scale parts of the solution occur quickly relative to variations caused by the slower time scale portions of the solution. Because of this, the fast time scale variations may be assumed to be at steady state for the times of interest without significant loss to the system kinetics description. Specifically, while the fast time scale part of the solution changes for different specified initial conditions, the slow time scale part of the solution trajectory is followed toward equilibrium for all initial conditions. By this, the slow time scale part of the solution is important and uniquely defines the kinetics of a system for times of interest here. Note this assumed "steady state" of some of the solution terms is different from the QSS technique discussed in section 2.2.1. Here no assumptions are made about the species behavior. Rather, the actual solution characteristics for the unmodified full system are used to approximate the simplified solution. It does not simplify by reducing the mechanism but constantly uses all traits of the system implicitly contained in the ODEs to identify the important (slow time scale) parts of the species evolution solutions.

The ILDM technique allows one to extract information from the system that describes the slow time scale portion of the solution in its phase space. The manifold may be used as a simplified chemical kinetics description in the system phase space. To understand the importance of the manifold, consider the solution trajectory in the phase space for the system that describes species concentration. Assuming that this system is stable, for sets of initial conditions within a certain region of the phase space, all solution trajectories of the system approach the same equilibrium point as time approaches infinity. This equilibrium point is the zero-dimensional attractive manifold for the system.

As the solution trajectories approach the equilibrium point, they tend toward other regions in the phase space for times less than infinity. When the slowest time scale term dominates all other terms at each point, the solution approaches a one-dimensional path in the phase space, the one-dimensional manifold. When the next slowest time scale term dominates all faster terms but is not yet dominated by the slowest time scale term, the solution trajectories approach a two-dimensional surface in the phase space, the two-dimensional manifold. Similar formulations describe higher order manifolds. All of these manifolds are characteristic of the system in that they depend on the time scales at each point. Thus, if the system dynamics description is not required at extremely small times, a manifold provides the location of trajectories in the system phase space after some of the transient effects have decayed. The dimension of the manifold may be associated with the amount of that time must pass before the manifold is a good description of the solution. Choosing the number of solution terms considered completely equilibrated determines the dimension of the manifold.

As described in Section 3.2, manifolds are found using numerical techniques. If numerical methods can accurately find the manifold of a system, this manifold will describe the system behavior at large times very well. There is no chance that a correctly determined manifold for a stable system will predict the wrong behavior for the system after the proper length of time. At least this is true if the initial conditions are within the correct region of the phase space such that the manifolds are attractive for these sets of initial conditions. Sometimes different sets of initial conditions tend toward different equilibrium points. As a final consideration, note that in general, the solution trajectories are never actually on the manifold at any time. Rather, the manifolds are only a very close approximation for the solution during certain time ranges. As time progresses, the solution trajectories approach the manifolds more closely [1]. The exception to this is when the specified initial conditions lie on a manifold. In this case, the solution trajectory never leaves the manifold [1].

3.2 Mathematics

In this section, the mathematics necessary to find manifolds with the ILDM technique is explained in detail. For the notation used in this paper, capital boldface letters indicate matrices while lowercase boldface letters represent vectors. To begin with, note that a n_s-dimensional system of n_s first order ODEs may be represented as

$$\frac{d\mathbf{x}}{dt} = \boldsymbol{f}(x_1, x_2, ..., x_{n_s}), \tag{2}$$

where t is the independent variable, $\mathbf{x}$ is the vector of dependent variables of length n_s, and x_i are the individual dependent variables. The vector $\boldsymbol{f}$ contains functions describing the rates of change of the dependent variables in terms of the dependent variables and other constant parameters. For all systems considered in this paper, the dependent variable t is time and has domain $0 \leq t < \infty$. If the system of Equation (2) is linear, it may be expressed as

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}, \tag{3}$$

where $\mathbf{A}$ is a matrix whose elements are constant real numbers. For the interests of this report, consider a stiff system of ODEs given by Equations (2) or (3). To find a low-dimensional manifold of this system, it is necessary to know the time scales associated with the solution at each point in the solution space. As stated in section 3.1, this is accomplished by finding the eigenvalues of the Jacobian $\mathbf{J}$ at each point in the solution space, where the Jacobian of the system given in Equation (2) is given by

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{\mathbf{o}} . \tag{4}$$

In Equation (4), the f_i are the individual functions describing the rates of change of each dependent variable x_i in terms of the all dependent variables x_i and other constant parameters. The subscript of the matrix, o, represents the point in the solution space at which the Jacobian is evaluated. Note that for a linear system (3), $\mathbf{J} = \mathbf{A}$.

Diagonalization is one technique that gives the eigenvalues of $\mathbf{J}$, but shortly it will be shown that this scheme for finding the time scales does not provide the most suitable information for constructing the manifold. Specifically, when the solution is represented geometrically in a state space, there is a characteristic direction associated with each eigenvalue. When finding a manifold, the essential task is to eliminate the possibility that the solution trajectories travel along the fast time scale directions [1]. Thus a low-dimensional manifold representation of the system solution contains only the slow time scale term(s) of the solution.

It will subsequently become evident that the elimination of characteristic directions associated with fast time scales at each point is most easily accomplished when the directions associated with each time scale are orthogonal. In general, the characteristic directions provided by diagonalization, called eigenvectors, are not orthogonal. Thus, a scheme is sought which provides the eigenvalues and an associated set of orthogonal characteristic directions for the eigenvalues in the geometric phase space. One way to accomplish this is by performing the Schur decomposition on on the Jacobian $\mathbf{J}$ at each point in the phase space. The Schur decomposition of a matrix $\mathbf{M}$ whose elements are real numbers is given by

$$\mathbf{M} = \mathbf{Q}\mathbf{T}\mathbf{Q}^T. \tag{5}$$

For the systems of interest here, the Jacobians always contain only real numbers. It is proven by Golub and Van Loan [12] that the Schur decomposition always exists for a matrix $\mathbf{M}$ whose elements are real or complex numbers. The Schur decomposition is such that the matrix $\mathbf{Q}$ is orthonormal [13], and $\mathbf{T}$ is upper triangular with the eigenvalues of $\mathbf{M}$ along its main diagonal. In addition, the columns of $\mathbf{Q}$ are linear combinations of directions associated with the fast subspace. Since $\mathbf{Q}$ is orthonormal, these column vectors (called the Schur vectors) may form the desired orthogonal set of directions in the geometric phase space associated with the fast time scale parts of the solution. Note that the eigenvalues of $\mathbf{M}$ may be arbitrarily ordered along the main diagonal of $\mathbf{T}$ [12], allowing for the calculation of the proper Schur decomposition with the desired directions in the columns of $\mathbf{Q}$.

Knowing that the Schur decomposition exists at every point in the solution space where Equation (2) is continuous and assuming that it can be accurately calculated at all of these points such that $\mathbf{T}$ contains the eigenvalues of $\mathbf{J}$ in order of descending magnitude down its main diagonal, an n_l-dimensional manifold can be calculated where $n_l < n_s$. The points on the manifold are found by solving Equation (6).

$$\tilde{\mathbf{Q}}^T \cdot \boldsymbol{f}(x_1, x_2, ..., x_{n_s}) = 0 \tag{6}$$

In this equation, $\tilde{\mathbf{Q}}$ is a part of $\mathbf{Q}$ from the Schur decomposition at point $\mathbf{x}$. For an n_l-dimensional manifold, $\tilde{\mathbf{Q}}$ is formed by removing the $n_s - n_l$ rightmost columns of $\mathbf{Q}$. For the Schur decomposition with eigenvalues ordered along the diagonal of $\mathbf{T}$ as discussed above, the removed columns are the characteristic directions for the $n_s - n_l$ slowest time scales. The remaining n_l column vectors are are all orthogonal to the n_l-dimensional manifold. Because the derivative vector $\frac{d\mathbf{x}}{dt}$ is tangent to all solution points in the state space, Equation (6) follows from Equation (2). Solving Equation (6) at every point using the Schur decomposition at each point with ordered eigenvalues to find $\tilde{\mathbf{Q}}$ gives the points $\mathbf{x}$ of the n_l dimensional manifold.

3.3 Applying the ILDM technique

3.3.1 Scheme

The basic scheme used to carry out the ILDM technique is a numerical procedure. When functions $\boldsymbol{f}(x_1, x_2, ..., x_{n_s})$ in Equation (2) are nonlinear, as is usually the case for systems of interest here, an analytical solution for the manifold is not possible. Thus, the goal is to find a large number of points that define the manifold at discrete locations in the phase space.

The procedure begins by finding the Schur decomposition of the system at a point known to be on the manifold, such as the equilibrium point, using a numerical technique. To find the Schur decomposition at a point, one must represent the system using the linear approximation for the system near the point. Namely, the Schur decomposition is performed on the Jacobian matrix $\mathbf{J}$ for the system at the point. Note that the Schur decomposition must have correctly ordered eigenvalues down the main diagonal of the upper triangular matrix $\mathbf{T}$.

Using the fact that all of the Schur vectors in $\tilde{\mathbf{Q}}^T$ are orthogonal to the manifold at the point, the next task is to determine the tangent to the manifold at the point. An approximate guess for the next point on the manifold is obtained by moving a small distance along the along the tangent to the manifold. Specifically, by specifying a small change in the ith coordinate of the initial manifold point,

$$x_i^g = x_i^p + \Delta x_i,$$

the guess for the next manifold point $\mathbf{x}^g$ is obtained by solving Equation (7) for the other coordinates $x_1^g, x_2^g, \ldots, x_{i-1}^g, x_{i+1}^g, \ldots, x_{n_s-1}^g, x_{n_s}^g$. In this notation, the superscript g indicates the guess for the next manifold point, the superscript p indicates the previously known manifold point, and the subscripts $(1, \ldots, n_s)$ indicate different coordinate directions in the phase space.

$$\tilde{\mathbf{Q}}^T \cdot (\mathbf{x}^g - \mathbf{x}^p) = 0 \tag{7}$$

Once $\mathbf{x}^g$ is found from Equation (7), an iterative root-solving procedure is used to solve Equation

(6) for an accurate estimate of the next manifold point. The iteration begins by using $\mathbf{x}^g$ as an initial guess for the next manifold point and continues until the value for the next manifold point converges within a specified tolerance.

Now the procedure is repeated in the same way with the new manifold point substituted for $\mathbf{x}^p$. The repetitions continue until manifold points for a sufficient region of the phase space have been determined. The size of the region depends on the range of initial conditions specified for Equation (2).

3.3.2 Numerical Methods and Accuracy

Because the mechanics of the ILDM technique involve the use of several numerical procedures, accuracy is a concern when calculating each manifold point. There are finite errors associated with the numerical calculations. It is important that these errors are small enough such that the calculated manifold coincides with the exact solution for the manifold. Although it is not always possible to determine the exact solution for the manifold analytically, it is assumed that such a solution does exist for the systems considered here. Note that for the scheme presented in Section 3.3.1, the calculation of each manifold point depends on the calculation of all previous points. Thus, as subsequent manifold points are calculated, the error associated with each point is magnified. (The error due to calculations for all previous points is present at the beginning of calculations for the new point, and the numerical calculations for the new point results in additional error.)

There are several instances when numerical methods are necessary when applying the ILDM technique. First, calculation of the Schur decomposition at a given point must always be carried out by a numerical procedure. Algorithms for efficient calculation of the Schur decomposition using the QR factorization and other advanced matrix computations are given by Golub and Van Loan [12]. In addition, obtaining correct ordering of the eigenvalues down the main diagonal of $\mathbf{T}$ in the Schur is not trivial. Golub and Van Loan [12] give an algorithm to accomplish this as well.

For the problems considered here, the code used to calculate the Schur decomposition with properly ordered eigenvalues is included in Appendix A. It was developed by Mr. Sandeep Singh at the University of Notre Dame and uses several subroutines from the IMSL libraries. The error associated with calculation of the Schur decomposition becomes negligible if enough iterations are performed when calculating it such that it converges to an unchanging value. The subroutines used in the Appendix A code calculate Schur decompositions that meet this convergence requirement.

Unless the system contains only a few short ODEs, calculation of the expressions for the partial derivative elements of the Jacobian $\mathbf{J}$ requires considerable time. Numerical methods for calculating the Jacobian become practical and necessary as the number and length of equations in the system increases. The IMSL libraries among many other software packages provide subroutines to calculate the Jacobian when given a system of equations. A similar convergence requirement is necessary for these calculations as for the Schur decomposition and is met by routines from the IMSL libraries.

It is not trivial to simultaneously solve Equation (6) for the manifold points. Rather, because the functions $\boldsymbol{f}(x_1, x_2, ..., x_n)$ are nonlinear, the solution of Equation (6) requires an iterative root solver. A similar situation exists when solving for the guess at the next manifold point from Equation (7) if the difference between the dimension of the system and the dimension of the manifold $(n_s - n_l)$ is large. Although the equations are not nonlinear here, the number of Gaussian operations required to solve the system becomes excessive for hand calculations. In each case, an iterative root-solver from the IMSL libraries is used to find the solution $\mathbf{x}$ or $\mathbf{x}^g$ when necessary. The subroutine used is programmed to give solutions within a specified tolerance. It is important that the tolerances are specified small enough to guarantee that the manifold points are found accurately to several orders of magnitude smaller than the values for point coordinates.

When developing numerical methods and codes for finding the points on the manifold, it is important to give the accuracy concerns discussed in this section serious consideration. If there is too much error in the calculation of the manifold points, the manifold may be a poor low-dimensional approximation of the actual system solution.

4 Example: Zeldovich Mechanism for NO formation

4.1 Mathematical Representation

Consider part of the Zeldovich mechanism for the formation of NO given in Table 1. The law of mass action with Arrhenius kinetics applied to a well-stirred, spatially uniform, isothermal, isochoric combustion system results in a system of ODEs that describes species concentration evolution. For the given Zeldovich mechanism under these conditions, the system of ODEs may be expressed in matrix form as

$$\frac{d}{dt}\begin{pmatrix}[N]\\ [NO]\\ [N_2]\\ [O]\\ [O_2]\end{pmatrix} = \begin{pmatrix}1 & -1 & -1 & 1\\ 1 & -1 & 1 & -1\\ -1 & 1 & 0 & 0\\ -1 & 1 & 1 & -1\\ 0 & 0 & -1 & 1\end{pmatrix}\begin{pmatrix}k_1[O][N_2]\\ k_2[NO][N]\\ k_3[N][O_2]\\ k_4[NO][O]\end{pmatrix}. \tag{8}$$

Elementary Reaction	k_i
1. $O + N_2 \rightarrow NO + N$	$k_1 = 1.8 \times 10^{14} \ \frac{cm^3}{mol\ s} \exp\left(\frac{-38370\ K}{T}\right)$
2. $NO + N \rightarrow O + N_2$	$k_2 = 3.8 \times 10^{13} \ \frac{cm^3}{mol\ s} \exp\left(\frac{-425\ K}{T}\right)$
3. $N + O_2 \rightarrow NO + O$	$k_3 = 1.8 \times 10^{10} \ \frac{cm^3}{mol\ s\ K} T \exp\left(\frac{-4680\ K}{T}\right)$
4. $NO + O \rightarrow N + O_2$	$k_4 = 3.8 \times 10^{9} \ \frac{cm^3}{mol\ s\ K} T \exp\left(\frac{-20820\ K}{T}\right)$

Table 1: This table contains part of the Zeldovich reaction mechanism for the formation of NO. The kinetic rate constants associated with each elementary reaction are included.

Note that Equation (8) is the same form as Equation (2) with species molar concentration as the dependent variable. It is therefore possible to develop manifolds for the system in Equation (8) using the ILDM technique described in Sections 3.2 and 3.3. The constant parameters k_i for the system are kinetic constants. They are determined by the system temperature T and elementary reaction activation energies. See Table 1 for the values of the constants k_i for each reaction.

Before proceeding with the ILDM method, it is desirable to make any other simplifications to the system that are possible. In this case, the system in Equation (8) may be simplified from a system of five ODEs to a system of two ODEs and three algebraic equations by performing Gaussian elimination on the system. The resulting system in reduced row echelon form is

$$\frac{d}{dt}\begin{pmatrix} [N] \\ [NO]-[N] \\ [N]+2[N_2]+[NO] \\ [N]+[O] \\ 2[O_2]+[NO]-[N] \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} k_1[O][N_2] \\ k_2[NO][N] \\ k_3[N][O_2] \\ k_4[NO][O] \end{pmatrix}. \tag{9}$$

Thus, the system is reduced to two ODEs and three linear algebraic equations. The algebraic equations are

$$\begin{pmatrix} 1 & 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} [N] \\ [NO] \\ [N_2] \\ [O] \\ [O_2] \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}, \tag{10}$$

where the constants C_1, C_2, and C_3 are determined from initial conditions. By performing Gaussian elimination on the linear system of Equation (10), one may obtain explicit expressions for three of the variables in terms of the other two. In Equation (11), this has been done for $[N_2]$, $[O_2]$, $[O]$ in terms of $[N]$ and $[NO]$.

$$\begin{pmatrix} [N_2] \\ [O] \\ [O_2] \end{pmatrix} = \frac{1}{2}\begin{pmatrix} C_1 - [N] - [NO] \\ 2(C_2 - [N]) \\ C_3 + [N] - [NO] \end{pmatrix} \tag{11}$$

The algebraic equations have physical significance. Specifically, it can be shown that they are conservation statements for the total number of atoms of each element and the total number of molecules contained in the system. Now only two ODEs must be solved to solve the entire system. Once the solution of the two-dimensional simplified system is known, the other variables may be determined using Equation (11).

Using the remaining two ODEs from Equation (9) and the algebraic constraints of Equation (11), the resulting two-dimensional system of ODEs in $[N]$ and $[NO]$ is

$$\begin{aligned}
\frac{d[N]}{dt} &= \frac{k_1}{2}(C_1 - [N] - [NO])(C_2 - [N]) - k_2[N][NO] \\
&\quad - \frac{k_3}{2}(C_3 + [N] - [NO]) + k_4[NO](C_2 - [N]), \\
\frac{d[NO]}{dt} &= \frac{k_1}{2}(C_1 - [N] - [NO])(C_2 - [N]) - k_2[N][NO] \\
&\quad + \frac{k_3}{2}(C_3 + [N] - [NO]) - k_4[NO](C_2 - [N]).
\end{aligned} \tag{12}$$

For two-dimensional system given in Equation (12), equilibrium point and the one-dimensional manifold will be found. In addition, because this system sufficiently small and not overly stiff, it may be solved numerically. This is helpful because it will allow examination of how actual solution trajectories behave relative to the manifold.

4.2 Results

A computer program that carries out the ILDM technique using the scheme of Section 3.3 for the Zeldovich example of Section 4.1 is included in Appendix B. The one-dimensional manifold for the system of Equation (12) is found using this code. Before explaining the actual graphical results which display the calculated manifold points, there are several points to consider about how the scheme of Section 3.3 is implemented here.

First, because the system in Equation (12) is small, hand calculations are used to get exact expressions for the terms of the Jacobian $\mathbf{J}$. For more lengthy systems, hand calculations require excessive time, and a numerical method to calculate the Jacobian would be used. Also, because $n_s = 2$ and $n_l = 1$, Equation (7) is easily solved by hand to give expressions for the coordinates of the guess for the next manifold point $\mathbf{x}^g$. For larger values of n_s and n_l, a numerical solution is more practical. Comment statements indicate when hand calculations are used in the program of Appendix B. Besides these two instances, the rest of the code relies on numerical procedures as indicated in Sections 3.3 and 3.3.2.

It is important to verify that the manifold points are correctly calculated for the system. Moreover, it is also important to verify that the manifold points represent an attractive low-dimensional intrinsic characteristic of the system that well describes the system behavior after fast time scale parts of the solution have decayed. Because the system of Equation (12) has small dimension and is not overly stiff, its complete numerical solution may be obtained. Thus, the correctness of the calculated manifold points is easily verified by comparing them with the actual solution points. If the solution points approach the manifold points after the fast time scale part of the solution has decayed, then the manifold is correct.

The complete numerical solution is first performed for the set of initial conditions

$$\begin{aligned} [N] &= 5 \times 10^{-4} \ \frac{mol}{cm^3} \\ [NO] &= 1 \times 10^{-3} \ \frac{mol}{cm^3}. \end{aligned} \tag{13}$$

The time evolution of the species concentrations for these initial conditions is presented in Figure 1, and the corresponding phase plot is given in Figure 2 (a). In Figure 2 (a) and all phase plots, arrows indicate the direction of increasing time along the trajectories.

Results for the calculated manifold are shown graphically in Figure 2 (b). Note that the manifold is calculated in both directions starting from the equilibrium point. The equilibrium point was determined numerically by solving Equation (14) using an iterative root-solver.

$$\boldsymbol{f}(x_1, x_2, ..., x_n) = \mathbf{0} \tag{14}$$

The plot in Figure 3 includes both the calculated complete solution and manifold from Figure 2. Note that as time passes and the solution approaches the equilibrium point, the fast time scale part of the solution decays, and the solution approaches the manifold. Thus, it is verified for the set of initial conditions in Equation (13) that the manifold is a good approximation for the system behavior after an initial transient period during which the fast time scale still influences the solution.

More complete solutions of the reduced Zeldovich system are shown in the phase plot of Figure 4 for other sets of initial conditions. Notice that all solutions tend toward the equilibrium point. Further, the solutions all seem to tend toward a path in the phase space before reaching the equilibrium path. In Figure 5, it is verified that this path to which all solutions are attracted after an initial transient time period is the calculated manifold of Figure 2 (b).

4.3 Concerns

There are several points of concern when applying the ILDM technique presented in Section 3.3. The first problem is finding an initial point on the manifold from which calculations for the other points may begin. It is suggested in Section 3.3.1 that the equilibrium point be used as the initial known point on the manifold. For the example system of Equation (12), it is relatively easy to find the equilibrium point. For systems with larger dimension n_s, solving Equation (14) for the equilibrium point requires increasing and eventually excessive computational time.

Another problem with the technique presented in Section 3.3 is the presence of turning points along the manifold path. The tangent line guess for the next manifold point is best when the radius of curvature of the manifold path is large. If the path curves too much, the tangent line guess for the next point will be too far from the actual next manifold point to allow the iterative solution of Equation (6) to converge. In some cases, specifying a smaller change $\Delta\mathbf{x}$ along the tangent line at points where convergence is a problem will allow further calculations. In some cases, though, the manifold path curves too sharply to allow calculation of all desired manifold points by the tangent line guess method presented here.

Therefore, the simple scheme presented here for calculation of manifolds is not sufficient for more

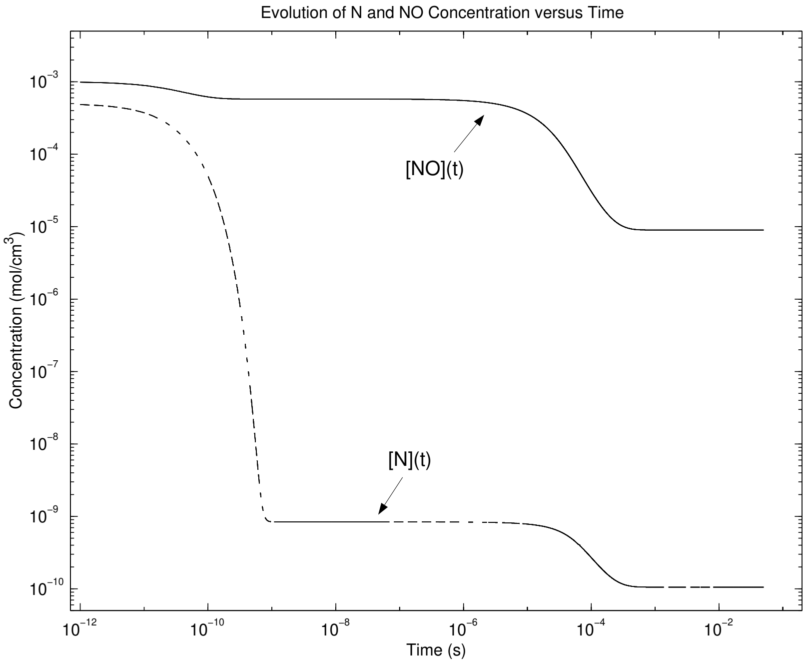
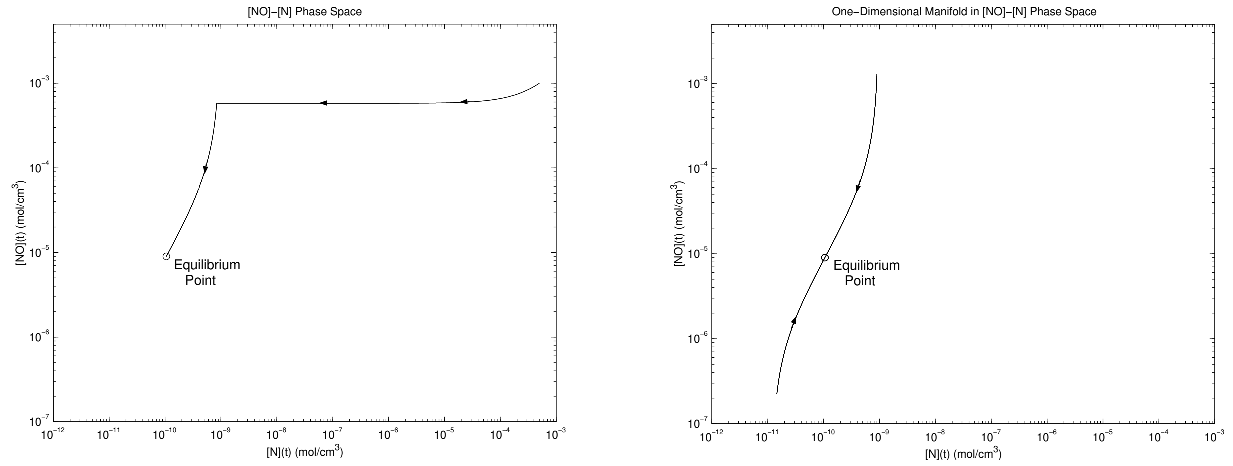


Figure 1: Evolution of species concentrations for the initial conditions $[N] = 5 \times 10^{-4}\ \frac{mol}{cm^3}$ and $[NO] = 1 \times 10^{-3}\ \frac{mol}{cm^3}$. Units are indicated on the axes of the plot.



(a) (b)

Figure 2: (a) Phase plot of the complete solution of Equation (12) for the initial conditions $[N] = 5 \times 10^{-4}\ \frac{mol}{cm^3}$ and $[NO] = 1 \times 10^{-3}\ \frac{mol}{cm^3}$. (b) Plot of the calculated manifold in the phase plane. Units are indicated on the axes of each plot.

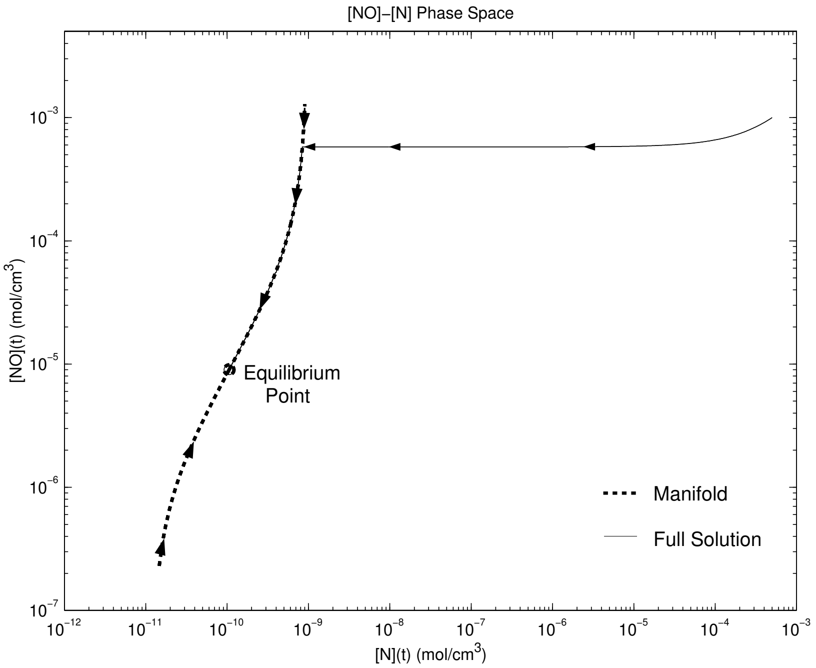


Figure 3: Plot in the phase plane showing both the calculated manifold and the complete solution of Equation (12) for the initial conditions $[N] = 5 \times 10^{-4}\ \frac{mol}{cm^3}$ and $[NO] = 1 \times 10^{-3}\ \frac{mol}{cm^3}$. Units are indicated on the axes of the plot.

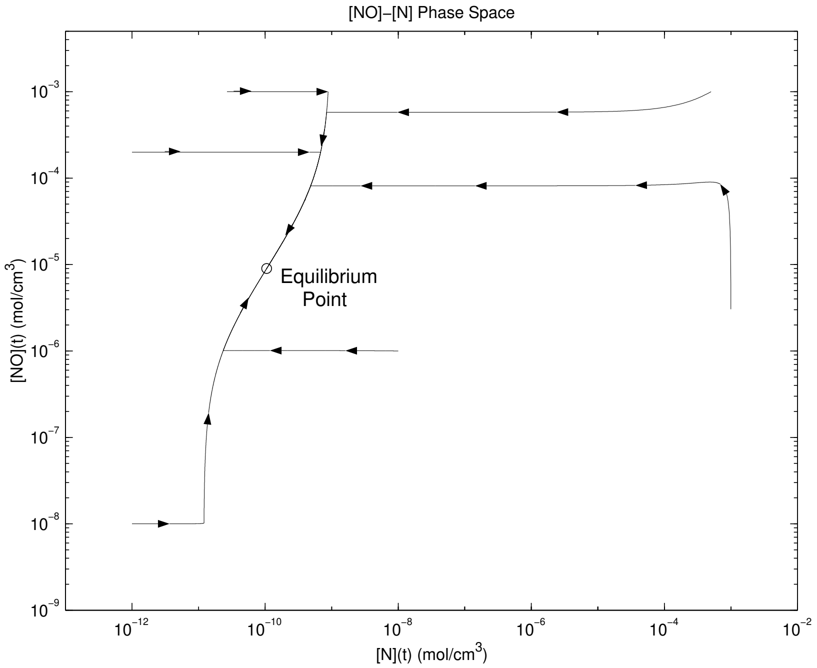


Figure 4: Phase plot of the complete solution of Equation (12) for several sets of initial conditions. Units are indicated on the axes of the plot.

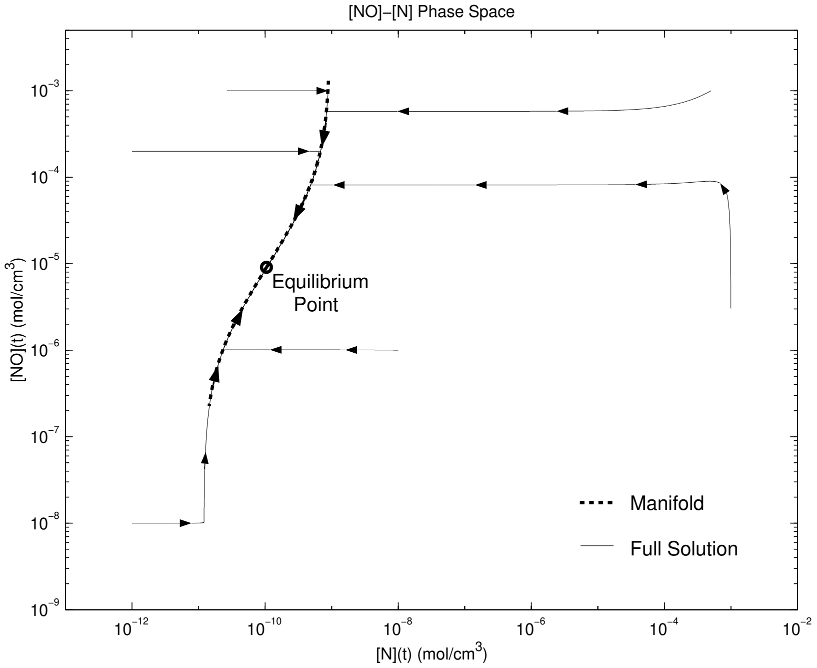


Figure 5: Plot in the phase plane showing the calculated manifold and the complete solution of Equation (12) for several sets of initial conditions. Units are indicated on the axes of the plot.

complicated problems. In fact, even for the simple system in Section 4, numerical convergence errors caused the program to terminate manifold calculations when traveling in each direction from the equilibrium point. It is suspected that the manifold reached turning points at each of the end points in Figure 2 (b). Better schemes must be used to allow calculation of the manifold at all points. The focus here is to understand manifolds and their importance while using a weak but simple method to verify their existence and basic characteristics.

5 Discussion

Consider the results of the example presented in Section 4.2. Observe that the results confirm the existence of a one-dimensional manifold for the ODE system given in Equation (12). Moreover, the manifold is attractive and is a good representation of the system kinetics behavior after the high time scale transient portion of the solution has decayed. Further, the manifold was accurately found using the simple scheme presented in Section 3.3.1.

Similar results have been found for more complex mechanisms and combustion systems [1], [4], [5], [8], [14]. To avoid numerical convergence problems due to poor initial guesses when calculating manifold points, some of these results were obtained using more sophisticated schemes than the one presented in Section 3.3.1. For example, Maas and Pope [1] mention an arc-length continuation method to avoid numerical convergence problems that might occur near turning points in the manifold.

As the detail of the mechanism increases, the manifold method for representing the limiting case of slow time scale kinetics behavior becomes more valuable because of the significant reduction in computational time that it allows. Although the QSS technique also reduces computational time, it may provide results that do not model the system as well as the ILDM technique [5], [8]. This occurs when poor species choices for QSS assumptions are made or when low temperatures characterize the system conditions. In theory, the ILDM technique should always accurately model kinetics behavior when the slow time scale dominates changes in the solution trajectory if an attractive (stable) low-dimensional manifold exists for the system.

Note that the systems considered here only model chemical kinetics of simple combustion systems. Thus, the solutions to such systems only describe kinetics behavior. For a more complete analysis of physical combustion systems, additional information is sought. For example, one may wish to solve reactive Euler and kinetics equations simultaneously to describe velocity, pressure, temperature, species concentration, or other variables' evolution in more physical systems. Some examples of interesting physical combustion systems are reactive flow problems, including laminar premixed and non-premixed flames, turbulent premixed and non-premixed flames, and ignition processes.

Computational time is almost always excessive when reactive Euler and kinetics equations are solved simultaneously for complex reaction mechanisms and reactive flow situations. One important reason for this large computational time requirement is stiffness of the kinetics ODEs. In order to facilitate a solution, one method is to use the ILDM method to produce a table of values of species concentrations at given states of the system in terms of the other variables [1], [5]. The calculation of values for the table of kinetics variables takes place separately and prior to the solution of the reactive Euler equations. Then, when kinetics information is required at a given state while solving the reactive Euler equations, the value is looked up in the table. These tables may be used for solutions to different physical systems that have the same reaction mechanism and other state variables.

It is not known beforehand what values of the state variables the system will take on. Therefore, if the table is calculated prior to the calculations determining the state variables, it must include the entire accessible region of the state space. The resulting table is large and requires significant computational time to create [9]. In addition, retrieving information from such a large table is not a trivial process [9]. Yang and Pope [9] therefore suggest another method that calculates the table of values *in situ* during the other combustion calculations.

6 Conclusions

The relevance of schemes for simplification of chemical kinetics was discussed herein. Notably, the stiffness of species concentration ODEs was examined, and the problem of excessive computational time for stiff ODEs was discussed. Several historical simplification schemes were explained and discussed. Brief explanations of the QSS and PEQ techniques were given, and the advantages and disadvantages of each were discussed. The ILDM technique was presented in more detail. After necessary definitions and mathematics that are important for understanding the ILDM technique were discussed, a simple implementation method was presented.

The ILDM technique was then applied to a simple combustion system with a four reaction, five species mechanism for the formation of NO. Results showed that the ILDM technique provides a good representation of the system kinetics behavior after the high time scale transient portion of the solution decays. The problems with the implementation method were discussed. In addition, schemes for applying the ILDM method to more complex and physically-based reactive flow situations were discussed.

References

[1] Maas, U. and Pope, S., "Simplifying Chemical Kinetics: Intrinsic Low-Dimensional Manifolds in Composition Space," *Combustion and Flame*, 88:239-264, 1992.

[2] Warnatz, J., Maas, U., and Dibble, R., *Combustion.* Springer-Verlag: Berlin, 1996.

[3] Turányi, A., Tomlin, A., and Pilling, M., "On the Error of the Quasi-Steady-State Approximation," *Journal of Physical Chemistry*, 97:163-172, 1993.

[4] Okino, M. and Mavrovouniotis, M., "Simplification of Mathematical Models of Chemical Reaction Systems," *Chemical Reviews*, 98:391-408, 1998.

[5] Eggels, R., Louis, J., Kok, J., and DeGoey, L., "Comparison of Conventional and Low-Dimensional Manifold Methods to Reduce Reaction Mechanisms," *Combustion, Science, and Technology*, 123:347-362, 1996.

[6] Lam, S. and Goussis, D., "The CSP Method for Simplifying Kinetics," *International Journal of Chemical Kinetics*, 26:461-486, 1994.

[7] Chen, J., "A General Procedure for Constructing Reduced Reaction Mechanisms with Given Independent Relations," *Combustion, Science, and Technology*, 57:89-94, 1988.

[8] Duchêne, P. and Rouchon, P., "Kinetic Scheme Reduction via Geometric Singular Perturbation Techniques," *Chemical Engineering Science*, 51:4661-4672, 1996.

[9] Yang, B. and Pope, S., "Treating Chemistry in Combustion with Detailed Mechanisms-*In Situ* Adaptive Tabulation in Principle Directions-Premixed Combustion," *Combustion and Flame*, 112:85-112, 1998.

[10] Fenichel, N., "Geometric Singular Perturbation Theory for Ordinary Differential Equations," *Journal of Differential Equations*, 31:53-98, 1979.

[11] Boyce, W. and DiPrima, R., *Elementary Differential Equations and Boundary Value Problems.* J. Wiley: New York, 1997.

[12] Golub, G. and Van Loan, C., *Matrix Computations.* The Johns Hopkins University Press: Baltimore, 1996.

[13] Hager, W., *Applied Numerical Linear Algebra.* Prentice Hall: Englewood Cliffs, 1988.

[14] Schmidt, D., Segatz, J., Riedel, U., Warnatz, J., and Maas, U., "Simulation of Laminar Methane-Air Flames using Automatically Simplified Chemical Kinetics," *Combustion, Science, and Technology*, 113:3-16, 1996.

Appendix A

The FORTRAN 77 computer program below finds the Schur decomposition of a matrix "`aa`." It outputs an orthogonal matrix of Schur vectors "`z`" associated with the upper triangular matrix "`a`" that has eigenvalues in order of decreasing magnitude down its main diagonal. By reconstructing the matrix "`aa`" from the Schur decomposition, the variable "`fnorm`" is calculated as a measure of the error associated with the Schur decomposition. The error is found to be very small relative to the magnitude of the values in the Schur vector matrix "`z`." This code was developed by Mr. Sandeep Singh at the University of Notre Dame.

```
      program schur_test
      implicit none

      integer n,lda,ldz,sdim,lwork,info,i,j,ifst,ilst
      parameter (n=6,lwork=50)
      logical select,bwork(n)
      character jobvs,sort,compq
      double precision a(n,n),z(n,n),wr(n),wi(n)
      double precision work(lwork),work1(n)
      double precision temp
      double precision zt(n,n),zazt(n,n),za(n,n),aa(n,n)
      double precision fnorm
      integer count

      external dgees,dtrexc

c
c  open the file, sch.out, for output
c
      open(unit=12,file='sch.out')

 100  format(20(f10.5,1x))

c     Enter the sample matrix

      a(1,1) = -12.d0
      a(1,2) = 3.d0
      a(1,3) = 1.d0
      a(1,4) = 1.d0
      a(1,5) = 5.d0
      a(1,6) = 4.d0
      a(2,1) = 2.d0
      a(2,2) = -6.d0
      a(2,3) = 2.d0
      a(2,4) = 8.d0
      a(2,5) = 3.d0
      a(2,6) = 2.d0
      a(3,1) = 1.d0
      a(3,2) = 3.d0
      a(3,3) = -2.d0
```

```
      a(3,4) = 5.d0
      a(3,5) = 1.d0
      a(3,6) = 1.d0
      a(4,1) = 4.d0
      a(4,2) = 3.d0
      a(4,3) = 7.d0
      a(4,4) = 7.d0
      a(4,5) = 3.d0
      a(4,6) = 9.d0
      a(5,1) = 4.d0
      a(5,2) = 6.d0
      a(5,3) = 4.d0
      a(5,4) = 7.d0
      a(5,5) = 2.d0
      a(5,6) = 2.d0
      a(6,1) = 8.d0
      a(6,2) = 3.d0
      a(6,3) = 0.d0
      a(6,4) = 4.d0
      a(6,5) = 1.d0
      a(6,6) = 6.d0

      go to 121

      do i=1,n
 do j=1,n
    a(i,j)=dsin(100.d0*i*j)
            enddo
         enddo

121    continue

      do i=1,n
 do j=1,n
    aa(i,j)=a(i,j)
    enddo
         enddo

      do i = 1,n
        write(*,100) (a(i,j),j=1,n)
      enddo
      write(*,*)

      jobvs = 'V'
      sort = 'N'

      lda = n
      ldz = n

c     Use the subroutine dgees of lapack to find the schur
c     decomposition of the matrix

      call dgees(jobvs,sort,select,n,a,lda,sdim,wr,wi,z,ldz,work,
     &           lwork,bwork,info)
```

```
c     Print the schur decomposition before reordering.

      write(*,*) "Schur Matrix before reordering"

      do i = 1,n
        write(*,100) (a(i,j),j=1,n)
      enddo
      write(*,*)

      write(*,*) "Schur Vectors before reordering"

      do i = 1,n
        write(*,100) (z(i,j),j=1,n)
      enddo
      write(*,*)

      write(*,*) "Start reordering"
      write(*,*)

      pause

c     Reordering...........
c----------------------------------------------------------------
      compq = jobvs
      count = 0
      ilst = 1

 300  ifst = ilst
      do i=ilst+1,n
        if (a(i,i).gt.a(ifst,ifst)) then
          ifst = i
        endif
      enddo

      if (a(ifst+1,ifst).ne.0.d0) then
        temp = a(ifst+1,ifst)
      endif

      if(ifst.ne.ilst)then
c       Use dtrexc to put 'ifst' eigenvalue to position 'ilst'

        call dtrexc(compq,n,a,lda,z,ldz,ifst,ilst,work1,info)

        count = count+1

        write(*,*)
        write(*,*) "Row", ifst,"  is shifted to Row", ilst
        write(*,*)
        write(*,*) "Schur Matrix after reordering no.", count
        write(*,*)
        do i = 1,n
          write(*,100) (a(i,j),j=1,n)
          write(12,*) (a(i,j),j=1,n)
```

```
        enddo
        write(*,*)

        pause
      endif


      if (temp.eq.0.d0) then
        ilst = ilst+1
      else
        ilst = ilst+1
      endif

      if (ilst.lt.n) then
        temp = 0.d0
        goto 300
      endif
c-----------------------------------------------------------------

c     Print the schur decomposition after reordering.

      write(*,*) "Schur Vectors after reordering"

      do i = 1,n
        write(*,100) (z(i,j),j=1,n)
        write(12,*) (z(i,j),j=1,n)
      enddo
      write(*,*)

c  Reconstruction

      do i=1,n
 do j=1,n
    zt(i,j)=z(j,i)
    enddo
         enddo

      call dmrrrr(n,n,z, n,n,n,a, n,n,n,za,  n)
      call dmrrrr(n,n,za,n,n,n,zt,n,n,n,zazt,n)

      write(*,*) "Reconstructed Matrix"

      do i=1,n
 write(*,100)(zazt(i,j),j=1,n)
 enddo
      write(*,*)

      fnorm=0.d0
      do i=1,n
 do j=1,n
    fnorm = fnorm+(aa(i,j)-zazt(i,j))**2
    if(dabs(aa(i,j)-zazt(i,j)).gt.1.d-3)print*,i,j
            enddo
         enddo
```

```
fnorm = dsqrt(fnorm)
print*,' '
print*,'fnorm =',fnorm

stop
end
```

Appendix B

The FORTRAN 77 computer program given below is used to calculate the one-dimensional manifold in [N]-[NO] space for the reduced system of Equation (12). In it, subroutine "`sch`" calculates the Schur decomposition of the Jacobian "`a(neq,neq)`." This subroutine was developed directly from the code in Appendix A written by Mr. Sandeep Singh. Places in the code where hand calculations were used to obtain expressions for calculating certain variable values are indicated with comment statements. The subroutine "(`dneqnf`)" is an iterative root-solver from the IMSL libraries is used to solve Equation (6).

```
      program manifold

      implicit none
      integer stp,neq,lwork,itmax
      parameter(neq=2,lwork=50)
      double precision a(neq,neq),x(neq),nitstp,nitmax
      double precision k1,k2,k3,k4
      double precision wr(neq),wi(neq),work(lwork)
      double precision work1(neq)
      double precision q(neq,neq),t(neq,neq)
      double precision xguess(neq),errrel,fnorm,f(neq)
      double precision nitnew,nonew
      logical bwork(neq)

      external dgees,dtrexc,dneqnf,fnc
      common /a/ k1,k2,k3,k4,q,xguess
cc open output files

      open(unit=13,file='mandata2.out')
      open(unit=14,file='schvect2.out')

cc physical constants
cc
cc  k1, k2, k3, and k4 are constants associated with the 4
cc  reactions of the Zeldovich mechanism considered.

      k1=1.8d14*dexp(-3.837d4/1.6d3)
      k2=3.8d13*dexp(-4.25d2/1.6d3)
      k3=1.8d10*1.6d3*dexp(-4.68d3/1.6d3)
      k4=3.8d9*1.6d3*dexp(-2.082d4/1.6d3)



cc beginning point
cc
cc The variables x(1) (nitrogen concentration) and x(2) (NO concentration)
cc are set to the equilibrium values as a starting point for finding the
cc manifold.  This is the logical place to start because it is where the
cc solutions tend toward.
```

```
      x(1)=1.05537559072647968d-10
      x(2)=8.98669354540823661d-6
      nitstp=1.d-12
      nitmax=9.3d-10


      print110,'n','no'
110   format(a7,2a15)
      print*,' '
      print100,x(1),x(2)
100   format(9(e10.4,1x))
      write(13,100)x(1),x(2)

      do 10 stp=1,100000

cc  Matrix "a" is the Jacobian of the differential equation system
cc  about the current point ([N],[NO]) along the manifold.
cc  The expressions for the elements of "a" were obtained from hand
cc  calculations for the system.
ccccccccc10cccccccc20cccccccc30cccccccc40cccccccc50cccccccc60ccccccccc72
      a(1,1) = -k1*(2.d-3 - x(1))/2.d0 - (k3*x(1))/2.d0
     &         - k1*(4.d-3 - x(1) - x(2))/2.d0
     &         - k3*(3.d-3 + x(1) - x(2))/2.d0 - k2*x(2) - k4*x(2)
ccccccccc10cccccccc20cccccccc30cccccccc40cccccccc50cccccccc60ccccccccc72
      a(1,2) = -k1*(2.d-3 - x(1))/2.d0 + k4*(2.d-3 - x(1))
     &         - k2*x(1) + (k3*x(1))/2.d0
ccccccccc10cccccccc20cccccccc30cccccccc40cccccccc50cccccccc60ccccccccc72
      a(2,1) = -k1*(2.d-3 - x(1))/2.d0 + (k3*x(1))/2.d0
     &         - k1*(4.d-3 - x(1) - x(2))/2.d0
     &         + k3*(3.d-3 + x(1) - x(2))/2.d0 - k2*x(2) + k4*x(2)
ccccccccc10cccccccc20cccccccc30cccccccc40cccccccc50cccccccc60ccccccccc72
      a(2,2) = -k1*(2.d-3 - x(1))/2.d0 - k4*(2.d-3 - x(1))
     &         - k2*x(1) - (k3*x(1))/2.d0


cc ---------------------------------------------------------------------------
cc Schur decomposition:
cc
cc Output of the subroutine "sch" called below is two matrices: "q" has
cc the schur vectors down its columns and is orthonormal.  "t" is upper
cc triangular and has the eigenvalues of the linearized system down its
cc main diagonal.
cc
      call sch(a,neq,t,q,wr,wi,work,lwork,work1,bwork)

cc ---------------------------------------------------------------------------


cc  Guess the next point of the manifold using the line tangent to the
cc  manifold at the current point.  Method: Increase [N] by a small amount
cc  from the previous [N] so the tangent line is still a close
cc  approximation for the new point on the manifold.
cc  The equation below for the new  value of [NO] at the new [N]
cc  is derived in the notes.  Because the Schur vector associated
```

```
cc  with the fastest time scale is perpendicular to the tangent line of
cc  the manifold at the old point, it is used to find the next point on the
cc  tangent line.  The expressions below are derived from Equation (6)
cc  using hand calculations.

      nitnew = x(1) + nitstp
      nonew = -q(1,2)/q(2,2)*(nitnew - x(1)) + x(2)

cc  Once the guess is obtained, an iterative method is used to more accurately
cc  solve for the manifold point using the actual differential equations and
cc  the Schur vector associated with the fastest time scale at the point:


      errrel = 1.d-13
      itmax = 5000
      xguess(1) = nitnew
      xguess(2) = nonew
      call fnc(xguess,f,neq)

      call dneqnf(fnc,errrel,neq,itmax,xguess,x,fnorm)
      print*,x,fnorm
      write(13,100)x(1),x(2)
      write(14,100)q(1,2),q(2,2)

      if(x(1).ge.nitmax)go to 11

10    continue

11    continue
      stop
      end


ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc                                                                         cc
cc                              Subroutines                                cc
cc                                                                         cc
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


cc -------------------------------------------------------------------------------

cc
cc  Subroutine fnc contains the functions f(x) for the system of differential
cc  equations d(x)/dt = f(x), where x is a vector containing the different
cc  species concentrations.
cc

      subroutine fnc(x,f,neq)
      implicit none
      integer neq,nn
      parameter(nn=2)
      double precision q(nn,nn),x(neq),f(neq),xguess(nn)
      double precision k1,k2,k3,k4
```

```
      common /a/ k1,k2,k3,k4,q,xguess

      f(1) = q(1,2)*(k1*(2.d-3 - x(1))*(4.d-3 - x(1) - x(2))/2.d0
     &      -k3*x(1)*(3.d-3 + x(1) - x(2))/2.d0
     &      +k4*(2.d-3 - x(1))*x(2)
     &      -k2*x(1)*x(2))
     &      + q(2,2)*(k1*(2.d-3 - x(1))*(4.d-3 - x(1) - x(2))/2.d0
     &      +k3*x(1)*(3.d-3 + x(1) - x(2))/2.d0
     &      -k4*(2.d-3 - x(1))*x(2)
     &      -k2*x(1)*x(2))

      f(2) = x(1) - xguess(1)
      return
      end

cc ------------------------------------------------------------------------------

cc
cc  Subroutine sch solves for the Schur Decomposition of a matrix "a."  Its
cc  output is an upper triangular matrix "t" and an orthonormal matrix "q."
cc  Calculations are performed such that the output matrix t has
cc  eigenvalues in order of increasing magnitude down its main diagonal.
cc  (This subroutine derives from the code in Appendix A developed by
cc  Mr. Sandeep Singh.)
cc

      subroutine sch(a,neq,t,q,wr,wi,work,lwork,work1,bwork)

      implicit none
      integer neq,i,j
      integer lwork,lda,ldz,count,sdim,info,ifst,ilst
      double precision a(neq,neq),q(neq,neq),t(neq,neq)
      double precision wr(neq),wi(neq),work(lwork)
      double precision work1(neq),temp
      logical select,bwork(neq)
      character jobvs,sort,compq

      external dgees,dtrexc


      do i=1,neq
 do j=1,neq
    t(i,j)=a(i,j)
    enddo
         enddo

      jobvs = 'V'
      sort = 'N'

      lda = neq
      ldz = neq
      call dgees(jobvs,sort,select,neq,t,lda,sdim,wr,wi,q,ldz,work,
     &           lwork,bwork,info)
```

```
      compq = jobvs
      count = 0
      ilst = 1

 300  ifst = ilst
      do i=ilst+1,neq
        if (t(i,i).gt.t(ifst,ifst)) then
          ifst = i
        endif
      enddo

      if (t(ifst+1,ifst).ne.0.d0) then
        temp = t(ifst+1,ifst)
      endif

      if(ifst.ne.ilst)then
c       Use dtrexc to put 'ifst' eigenvalue to position 'ilst'

        call dtrexc(compq,neq,t,lda,q,ldz,ifst,ilst,work1,info)

        count = count+1

      endif

      if (temp.eq.0.d0) then
        ilst = ilst+1
      else
        ilst = ilst+1
      endif

      if (ilst.lt.neq) then
        temp = 0.d0
        goto 300
      endif

      return
      end
```