Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10, D-10711 Berlin - Wilmersdorf

Peter Deuflhard Jörg Heroth

Dynamic Dimension Reduction in ODE Models

Preprint SC 95–29 (November 1995)

Dynamic Dimension Reduction in ODE Models

Peter Deuflhard Jörg Heroth

Abstract

The paper analyzes a splitting technique into fast and slow dynamical components of ODE systems as suggested by MAAS AND POPE recently. Their technique is based on a real block – Schur decomposition of the Jacobian of the right hand side of the ODE. As a result of the analysis, a computationally cheap monitor for the possible necessary recovering of the splitting is derived by singular perturbation theory. Numerical experiments on moderate size, but challenging reaction kinetics problems document the efficiency of the new device within a linearly-implicit stiff integrator.

Contents

1 Introduction

The economization or reduction of chemical reaction models is an important task in the simulation of complex reaction kinetics. One standard mathematical tool is the well-known technique of *quasi-steady-state approximation* (QSSA). It is typically based on chemical insight into the reaction mechanism: by identifying certain chemical species as "radicals", the associated time derivatives are set to zero thus indicating the fact that these radicals are "fast" components, which approach a local equilibrium state quite quickly in comparison with the other so-called "slow" components of the system. In the spatially homogeneous case, this traditional QSSA technique can be characterized by a deliberate transition from a system of ordinary differential equations (ODEs) to a system of differential-algebraic equations (DAEs), wherein the fast components enter through algebraic equations. The mathematical justification of such a procedure is the *singular perturbation technique* – see, for example, the well-known textbook of O'Malley [19]. Unfortunately, as it turns out, chemical insight is often not enough to construct a DAE of so-called *index* 1, which assures that it has a *unique* solution – compare e.g. the illustrating example in [6]. Moreover, even when the index is 1, there can be a quite large difference between the solutions of the ODE and the associated DAE obtained from the QSSA procedure – which led the authors of [8] to talk about "QSSA – fact or fiction".

Recently, Maas and Pope [18] suggested another method for the splitting of fast and slow components, which is based on a real Schur decomposition of the Jacobian of the right hand side of the ODE. This technique is reported to be quite powerful especially in the treatment of combustion problems – see Maas[17]. Within the context of partial differential equations (PDEs) of reaction–diffusion type, it permits an overall reduction of the dimension of the whole PDE system, which makes problems at the computational complexity border still tractable. Its implementation is performed as a table look-up for the manifold defined by the algebraic conditions. However, even though this new kind of QSSA assures index 1, it does not check for the approximation error introduced by the transition from the ODE to the DAE. As a consequence, it does not allow to adapt the reduction dynamically in the time dependent PDE case.

The present paper steps into this gap. In order to emphasize ideas rather than technicalities, the more complicated PDE situation is replaced here, for the time being, by the much simpler ODE situation. In Section 2 below, the splitting technique of Maas and Pope is revisited in mathematical terms of ODEs and associated DAEs. As implementation the linearly-implicit Euler discretization [4] is exemplified. In Section 3, a cheap estimation technique for the introduced QSSA error is analytically derived and its implementation discussed. This estimation technique permits the desired adaptive control of the QSSA error also dynamically. Finally, in Section 4, the thus developed dynamic dimension reduction method for ODE models is illustrated by three moderate size, but nevertheless quite challenging examples from chemical reaction kinetics. The positive effect of the new dimension monitor on the robustness and efficiency of the numerical simulation is well documented. The transfer of the herein presented techniques to the PDE situation will be published in a forthcoming paper.

2 Basic Splitting Scheme

Consider the following initial value problem (IVP) for ordinary differential equations (ODEs)

$$\dot{x} = F(x), \quad x(0) = x_0 \tag{1}$$

wherein $x \in \mathbf{R}^n$ may represent concentrations of chemical species, temperature, pressure etc. When the system is stiff, it will have so-called "fast" components expected to reach their steady state after a short transient period. As a consequence, a smaller number of degrees of freedom may be sufficient to describe the system dynamics in terms of the remaining "slow" components only.

Notational frame. Assume that we split x according to $(y, z) = (Px, Qx)$ where P and Q are projections on the dynamically slow and fast parts, respectively. Let $d < n$ denote the number of slow components, which means that $d = \mathrm{rank}(P)$. The projections may depend on the solution itself. Upon formally applying the projections to the above ODE, we arrive at

$$P\,\dot{x} = PF(x), \quad Q\,\dot{x} = QF(x), \quad x(0) = x_0. \tag{2}$$

Upon imposing the so-called *quasi-steady-state assumption* (QSSA)

$$Q(x)\dot{x} = 0,$$

we finally come to the differential algebraic equation (DAE)

$$P\dot{x} = PF(x), \quad QF(x) = 0, \quad x(0) = x_0 \tag{3}$$

A necessary condition for this DAE to have a solution at all is that the initial value x_0 lies within the *slow manifold* $\mathcal{M} = \{x : QF(x) = 0\}$. Otherwise, a projection of x_0 onto this manifold will be necessary to make (3) consistent. But, as already RHEINBOLDT has shown in [20], not every system of the form (3) has a unique solution, even if the initial values are consistent with the algebraic conditions. This has led to a classification of DAE systems with respect to the so-called *index*. In particular, for such systems to have a unique solution, its index should be 1.

In order to simplify the subsequent presentation, we will turn to the popular notation of *singular perturbation* theory: assume we have successfully performed the splitting $(Px, Qx) = (y, z)$ such that, in terms of some perturbation parameter ε, we may write equation (2) in the well separated form

$$\begin{aligned} \dot{y} &= f(y,z), \qquad y(0) = Px_0 \\ \varepsilon\,\dot{z} &= g(y,z), \qquad z(0) = Qx_0, \end{aligned} \tag{4}$$

In this notation, the above quasi-steady state assumption now reads $g(y, z) = 0$, which implies an associated DAE of the form

$$\begin{aligned} \dot{y}_0 &= f(y_0,z_0), \qquad y_0(0) = \bar{y}_0 \\ 0 &= g(y_0,z_0), \qquad z_0(0) = \bar{z}_0, \end{aligned} \tag{5}$$

where the initial values $(\bar{y}_0, \bar{z}_0)$ are assumed to be consistent, i.e.

$$(\bar{y}_0, \bar{z}_0) \in \mathcal{M}_d := \{(y,z) : g(y,z) = 0\}.$$

By fixing the slow components according to

$$\bar{y}_0 = y(0), \tag{6}$$

we will have to solve the nonlinear system $g(\bar{y}_0, z) = 0$ to obtain the perturbed value for the fast components

$$\bar{z}_0 = z(0) - \Delta z_0. \tag{7}$$

If this nonlinear system has a locally unique solution (say due to g_z regular in some neighborhood), then the DAE (5) has locally index 1, which means that it also has a locally unique solution.

As stated above, the projections may theoretically depend on the solution itself, which makes the treatment quite complicated alredy in the Index-1 case – for a rather compact presentation of this case, the reader may look up the recent textbook [3]. In actual computation, instead of a global projection $P(x)$ with global rank d, we will at best be able to adjust P and d as a sequence of local projections and their associated dimensions. Within a (stiff) one-step discretization scheme, the usual sequence of integration steps

$$\mathrm{IVP}_1 \quad\longrightarrow\quad \mathrm{IVP}_2 \quad\longrightarrow\quad \ldots \quad\longrightarrow\quad \mathrm{IVP}_N \quad\longrightarrow\quad \text{solution}$$

will then be replaced by a sequence of the kind

$$\begin{array}{ccccccccc} \mathrm{IVP}_1 & & \mathrm{IVP}_2 & & & \mathrm{IVP}_N & & \\ \downarrow & \nearrow & \downarrow & \nearrow & \cdots & \downarrow & \nearrow & \text{solution} \\ \mathrm{DAE}_1 & & \mathrm{DAE}_2 & & & \mathrm{DAE}_N & & \end{array}$$

wherein a projection of the new initial values onto the new algebraic manifolds will be necessary at each integration step. An important task in this context will therefore be to maintain firm control of the approximation error introduced by the repeated switch from the ODE to the DAE – this topic is left to the subsequent Section 3.

Local splitting algorithm. The aim is to separate the slow and fast components of the dynamical system. The slow components will include the unstable subspace and part of the stable subspace. Of course, such an aim can only be achieved locally. In order to find appropriate coordinate changes, one may consider transforming the Jacobian $F_x(x_0)$ into Jordan canonical form thus obtaining the set of eigenvectors. However, the Jordan canonical form as a whole may be ill–conditioned due to the occurrence of "nearly linearly dependent" eigenvectors – for reference see the beautiful survey paper by Golub and Wilkinson [11]. Fortunately, the computation of the associated invariant subspaces may nevertheless be well–conditioned. On the basis of this insight, U.Maas [17] suggests to apply some *block-Schur decomposition*, which is recalled here for convenience: starting from the matrix $A = F_x(x_0)$, this decomposition supplies certain nonsingular matrices T_d, which generate the following block splitting as a similarity transform:

$$T_d^{-1}\, A\, T_d \;=\; S \;=\; \begin{pmatrix} S_{11} & 0 \\ 0 & S_{22} \end{pmatrix}$$

Herein S_{11} and S_{22} appear in so-called *real Schur form*, which means that they are upper triangular with possible nonvanishing elements on the first subdiagonal to take care of complex conjugate eigenvalue pairs. The splitting dimension d is just the dimension of the submatrix S_{11}. The eigenvalues λ can be grouped according to their real parts such that some splitting parameter μ can be defined as

$$\mu := \max_{\lambda \in S_{22}} \Re\lambda < 0 \qquad \text{and} \qquad \min_{\lambda \in S_{11}} \Re\lambda > \mu .$$

The columns of T_d will now serve as the new basis vectors, where the first d columns may be interpreted as "slow" directions and the last $n-d$ ones as "fast" directions. The above numerical decomposition is achieved by first performing a real Schur decomposition, which yields

$$Q_d^T\, A\, Q_d = \bar{S} = \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix}$$

with an orthogonal matrix Q_d. Now the desired block structure is obtained by solving the Sylvester equation

$$S_{11}Z - ZS_{22} = -S_{12}$$

and setting

$$T_d^{-1} \;=\; \left(I - \begin{pmatrix} 0 & Z \\ 0 & 0 \end{pmatrix} \right) Q_d^T \quad \text{and} \quad T_d \;=\; Q \left(I + \begin{pmatrix} 0 & Z \\ 0 & 0 \end{pmatrix} \right).$$

The whole algorithm with more details is given in [10]. Fortran-subroutines are included in the package [16]. During the second step of the algorithm, the condition number of T_d, which depends on Z only, can be monitored by means of an estimation of the separation of the invariant subspaces. In this way, new coordinates

$$\begin{pmatrix} y \\ z \end{pmatrix} := T_d^{-1}\, x, \tag{8}$$

arise naturally together with the perturbation parameter $\varepsilon := 1/\mu$ so that the above singularly perturbed ODE system (4) is substantiated. Note that the choice of μ assures index 1 for the associated DAE (5).

Discretization. The above DAE can, in principle, be discretized by any suitable method, given for instance in textbooks like [12] or [3]. For ease of presentation, we here concentrate on the rather basic *linearly implicit Euler discretization.* When applied to the separated form (5), which requires the block-Schur transformation first, this discretization reads:

$$\begin{pmatrix} I - \tau f_y & -\tau f_z \\ -\tau g_y & -\tau g_z \end{pmatrix} \begin{pmatrix} \Delta y \\ \Delta z \end{pmatrix} = \tau \begin{pmatrix} f \\ g \end{pmatrix}. \tag{9}$$

In this notation, the numerical solution $(y_\tau, z_\tau)(\tau)$ at the next point is defined as

$$\begin{aligned} y_\tau(\tau) &= y(0) + \Delta y \\ z_\tau(\tau) &= z(0) + \Delta z \end{aligned}$$

Note that we did not make use of any property like $f_z = 0$, $g_y = 0$, since this is only true in x_0, and we want to keep the transform T_d over several steps, because the Schur decompostitions are very costly.

As shown in [4], the discretization error of this scheme has a perturbed asymptotic expansion, where the perturbations depend on the choice of the applied Jacobian approximation within the linearly implicit scheme. Up to the order $p = 3$, there is no perturbation of the asymptotic expansion, if only the submatrices f_z and g_z are approximated to sufficient accuracy. The choice of the submatrices in the places of f_y and g_y is not relevant in this respect. Consequently, if we confine the method to just this order, we are able to apply τ-extrapolation including stepsize control without specifying the approximations for f_y, g_y. In addition, this order restriction also goes with the eventual aim of solving PDEs, which means concentrating on less stringent accuracies and therefore low orders.

The above considerations have also been used to construct a simple method for the iterative solution of the linear block system (9). Let A denote the Jacobian approximation in (9) and $\hat{A}$ the associated matrix with g_y replaced by 0. Then the system $\hat{A}x = b$ can be solved easily because of its nearly upper triangular block structure. On this basis, we constructed a *fixed point iteration*, which is known to converge with contraction rate not greater than the spectral radius $\rho(I - \hat{A}^{-1}A)$.

Obviously, since $g_y = 0$ in the starting point x_0, we will have $\rho < 1$ in some neighborhood, which can be monitored.

Remark. Note that the discretization error analysis from [4] also allows the exclusive use of only the *first* iterate. We have tested this version of discretization, too, and have not obtained better performance. That is why we finally decided to stick to the full discretization scheme, which requires the approximation of g_y. In addition, in almost all of our experiments the remaining slow system could be integrated *explicitly*, i.e. with the choice $f_y = 0$ in both A and $\hat{A}$. In this case, we only need to decompose g_z once per integration step – which has already been done in order to compute the projection of the initial data onto the manifold by Newton's method.

3 Estimation of the QSSA Error

As derived in the preceding section, we want to solve a sequence of DAE systems of index 1 with varying dimension $n - d$ of the algebraic constraints. Standard numerical integrators for both ODEs and DAEs are known to control certain *local discretization error estimates* ε_k by testing $\varepsilon_k \leq$ TOL with TOL being a user prescibed error tolerance. In our context of dynamic dimension reduction the local errors will *additionally* contain errors introduced by the switching from an ODE system of type (4) to a DAE system of type (5). These errors need to be estimated in a cheap and reliable way in order to enable a theoretically backed splitting into slow and fast components of the dynamical system. Moreover, we want to keep a splitting scheme over several steps, if possible, since the actual computation of the splitting (see preceding section) is quite expensive compared with the evaluation of the right hand side. In [13], a first step towards the estimation of this type of QSSA error has been made. However, the techniques there used *operator norm estimates*, which have a tendency to overestimate the special effect. For this reason, we present here *componentwise estimates* instead, which can also be computed quite conveniently in the course of the integration of the DAE.

Let, as in Section 2, (y, z) denote the solution of the ODE system (4) and (y_0, z_0) the solution of the DAE system (5). Then the QSSA error of interest after one integration step is $\alpha = \|(y, z)(\tau) - (y_0, z_0)(\tau)\|$, where τ is the timestep chosen by the applied numerical integrator. In the special situation, we can apply standard results from singular perturbation theory, in particular a quite well-known result of VASIL'EVA [21] – see, for instance, the textbook [19]. If we assume the right-hand side F to be at least twice differentiable, the following asymptotic expansion is known to hold:

$$y(t) \quad = \quad y_0(t) + \varepsilon(y_1(t) + \eta_1(t/\varepsilon)) + O(\varepsilon^2)$$

$$z(t) \;=\; z_0(t) + \zeta_0(t/\varepsilon) + \varepsilon(z_1(t) + \zeta_1(t/\varepsilon)) + O(\varepsilon^2)$$

Here ζ_0, ζ_1 and η_1 are exponentially decaying boundary layer terms in the sense that

$$\|\zeta_0(s)\| \;\leq\; \|\zeta_0(0)\|\, e^{-\kappa s}, \quad \|\eta_1(s)\| \;\leq\; \|\eta_1(0)\|\, e^{-\kappa s} \tag{10}$$

with $\kappa \approx 1$ due to the scaling of ε. We assume the fast components to be controlled by the system dynamics and focus on the approximation error in the slow components. Furthermore, we restrict our analysis to the first order error term $y_1(\tau)$ and neglect the exponentially decaying $\eta_1(\tau)$. Thus we arrive at the estimate

$$[\alpha] \;:=\; \varepsilon\, \|y_1(\tau)\|.$$

VASIL'EVA's theorem shows that y_1 is the solution of the following IVP:

$$\dot{y}_1 = (f_y - f_z\, g_z^{-1} g_y)\, y_1 - f_z\, g_z^{-2} g_y\, f, \quad y_1(0) = -\eta_1(0)$$

where for ease of writing we have dropped the argument $(y_0, z_0)(t)$.
The initial value is given by

$$y_1(0) = -\eta_1(0) = \int_0^\infty \left(f(\, y_0(0), z_0(0) + \zeta_0(s)\,) - f(\, y_0(0), z_0(0)\,) \right)\; ds. \tag{11}$$

Variation of constants then leads to the estimate

$$\|y_1(\tau)\| \leq e^{L\tau}\, \|y_1(0) + b\|$$

with properly chosen constants L and b. Since we have decomposed our system at x_0, we have $f_z(y(0), z(0)) = 0$ and $g_y(y(0), z(0)) = 0$, so that in some neighborhood $y_1(0)$ is the dominating part and we can neglect b. (This assumption will be checked during computation – see below.) Any useful stepsize selection device will keep $e^{L\tau} \approx 1$, so that we finally arrive at the estimate

$$[\alpha] \;\approx\; \varepsilon\, \|y_1(0)\|.$$

Hence, in order to compute this estimate, we are just left with the evaluation of the integral (11). As a first step towards this task, we apply the mean value theorem in the form

$$y_1(0) = \int_0^\infty \int_0^1 f_z(z_0(0) + \theta\zeta_0(s)\,)\, \zeta_0(s)\, d\theta\, ds. \tag{12}$$

In order to derive a cheap approximation of this integral, we impose the assumption (compare (10))

(I) $$\zeta_0(s) = \zeta_0(0) e^{-\kappa s},$$

which, inserted above, yields

$$y_1(0) = \frac{1}{\kappa}\int_0^1 \int_0^\infty f_z(z_0(0) + \theta\zeta_0(s/\kappa))\,\zeta_0(0)e^{-s}\,ds\,d\theta.$$

Applying *Gauss-Laguerre quadrature* (for reference see e.g. [5]), we obtain

$$y_1(0) = \frac{1}{\kappa}\int_0^1 f_z(z_0(0) + \theta\zeta_0(1/\kappa))\,\zeta_0(0)\,d\theta + R_1$$

with

$$R_1 = \frac{1}{2\kappa}\int_0^1 \left[\frac{d^2}{ds^2} f_z(z_0(0) + \theta\zeta_0(s/\kappa))\zeta_0(0)\right]_{s=\sigma} d\theta$$

for some intermediate value $\sigma \in (0, \infty)$. At this point, we assume that

(II) $\qquad f_{zz} = \text{constant in the direction of } \zeta_0(s)$

and consequently neglegt R_1. So, finally, we have to compute the estimate

$$y_1(0) \;\approx\; \frac{1}{\kappa}\int_0^1 f_z(z_0(0) + \theta\zeta_0(1/\kappa))\,\zeta_0(0)\,d\theta. \tag{13}$$

To evaluate this integral, we set according to assumption (I) $\zeta_0(0) = \zeta_0(1/\kappa)e^1$ and apply the mean value theorem once again, but now in reverse order. Thereby we get

$$y_1(0) \approx \frac{e}{\kappa}\Big(\, f(\, z_0(0) + \zeta_0(0)e^{-1})\,) - f(\, z_0(0)\,)\,\Big)\,.$$

Note here, that we do not have the value of $f(z_0(0) + \zeta_0(0)e^{-1})$. In linear approximation, however, we would have

$$e\Big(\, f(\, z_0(0) + \zeta_0(0)e^{-1}\,) - f(z_0(0))\,\Big) \doteq f(\, z_0(0) + \zeta_0(0)\,) - f(\, z_0(0)\,).$$

The values of f for these two arguments are given. Then, by observing that $z(0) = z_0(0) + \zeta_0(0)$ and $y(0) = y_0(0)$, we arrive at

$$y_1(0) \approx \frac{1}{\kappa}\Delta f \;:=\; \frac{1}{\kappa}\,(\, f(y(0), z(0)) - f(y(0), z_0(0))\,)\,,$$

which implies the cheaply available estimate

$$[\alpha] \;=\; \frac{\varepsilon}{\kappa}\,\|\Delta f\|. \tag{14}$$

In this derivation we have not made any use of the property $f_z(y(0), z(0)) = 0$. In fact, we can gain a factor 1/2 in (14), if x_0 is a decomposition point. Numerical experiments, with Schur decompositions kept over *several* points, however, showed that this has no relevance for the efficiency of the computations.

The estimate just derived can be used to monitor the splitting via the criterion

$$[\alpha] \leq \mathrm{TOL}. \tag{15}$$

If this condition is violated, the splitting should be performed again and the dimension d should be *possibly increased.* Unfortunately, the criterion may not point out a possible *decrease* of the splitting dimension d. Moreover, the criterion may not indicate clearly enough that the splitting itself should be recovered, when the perturbation parameter no longer serves the purpose to characterize a perturbation. That is why, in addition to the above criterion, we also require *fast* convergence in the iterative process sketched at the end of Section 2 above. Whenever the splitting deteriorates, so will the spectral radius $\rho(I - \hat{A}^{-1}A)$, which is an upper bound of the contraction rate of this iteration. Consequently, whenever the rate of convergence appears to slow down, we regenerate the splitting matrices T_d with possible actualization of d.

Scaling. A subtle question of practical importance is the dependence of all the algorithmic devices upon scaling or gauging of variables including the time variable. Careful examination shows that all suggested devices are invariant under stretching of the time variable. In order to assure also invariance under regauging of the state variables x, we apply an a-priori scaling at each integration step so that the derived variables y, z enter into condition (15) as scaling invariant variables. This is done by scaling the Jacobian *before* the actual computation of the block-Schur decomposition.

4 Numerical Experiments

The dynamic dimension reduction algorithm as derived above has been implemented within the linearly implicit Euler discretization with two extrapolation steps only, thus restricting the discretization order to only $p = 3$. Of course, any other DAE integrator could have been used to demonstrate the performance of the new adaptive technique. For the sake of clarity, we briefly arrange the main steps of the algorithm:

1. Scaling
2. Block-Schur decomposition
3. Transformation from the scaled variable x to the separated variables y, z
4. Projection on $\mathcal{M}_d$
5. Check for QSSA error by criterion (15)
6. DAE-solver : linearly implicit Euler with $p = 3$
7. Iterative solution of the discretized DAE plus check of convergence rate

Due to the quite costly Step 2 ($\approx 15n^3$ flops), this algorithm is essentially designed for moderate size chemistry up to $n \approx 100$ species, say. In the above form, the algorithm has been used for all three subsequent examples. For the user prescribed accuracy we always set $\mathrm{TOL} = 10^{-2}$.

In order not to raise wrong expectations: in all of our experiments, the direct numerical integration of the unprepared stiff ODE system was much faster than the integration of the split DAE system, since the splitting and associated transformation cost quite a bit. However, as stated already in the beginning, the purpose of the paper is to derive a reliable dimension reduction tool for application in the context of PDEs. The presentation, however, is much simpler in the ODE context. The eventual effect of the herein advocated methods on the actual numerical solution of challenging reaction-diffusion problems will be shown in a forthcoming paper.

Example 1: Hydrogen-Oxygen Combustion [14]. This problem consists of $n = 8$ chemical species (and ODEs) and 37 elementary chemical reactions. The reaction mechanism is part of a system investigated by U.Maas in [18] and [17]. The problem has been given in [14] as an example, where the traditional QSSA (as mentioned in the Introduction) fails to be applicable. We were therefore interested to see the performance of our dynamic dimension reduction.

In a first numerical experiment, the system was split at *each* integration step, which sums up to a computational amount of 41 block-Schur decompositions. In Fig. 1, the numerical solution of the thus constructed sequence of DAE systems is represented graphically; it agrees perfectly (to the required accuracy) with the results obtained from solving the original ODE system. The small ignition zone in time is nicely resolved by the stepsize control, which demonstrates that our dimension reduction mechanism does not affect the efficiency of the integrator.

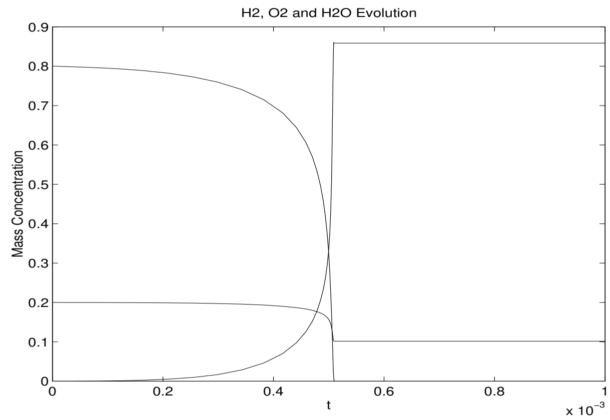


Figure 1: Example 1 - species concentrations H_2, O_2 and H_2O

As for the dimension of the reduced slow part of the system, only *two* parameters appear to be necessary to describe this challenging combustion problem. (It should be noted that we treated the invariants of the system independently.) The adaptive method works satisfactorily: once the hydrogen is exhausted, the system reaches its fixed point with $d = 0$.

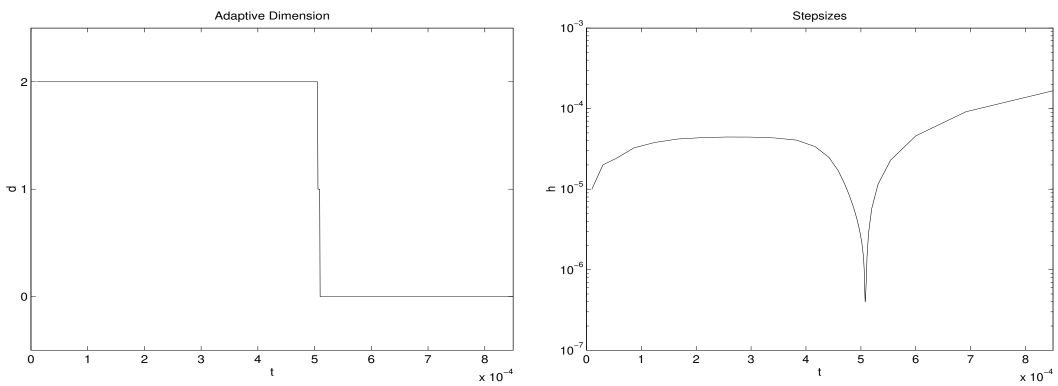


Figure 2: Example 1 - Splitting dimension and timestep behavior for block-Schur decomposition at each discretization step (41 steps)

In a second numerical experiment, we controlled the number of costly block-Schur decompositions by means of a convergence criterion, which monitors the iterative process for the solution of the discretized DAE system (9) (compare the description at the end of Section 2). The obtained numerical results once more agreed with those of Fig. 1. The behavior of the algorithm was quite similar, but much cheaper: only 8 decompositions during 51 steps were necessary to perform a useful dynamical splitting. As can be seen in Fig. 3, the dimension reduction mechanism produced essentially the same results, whereas the stepsizes came out to be slightly smaller than in the first experiment.

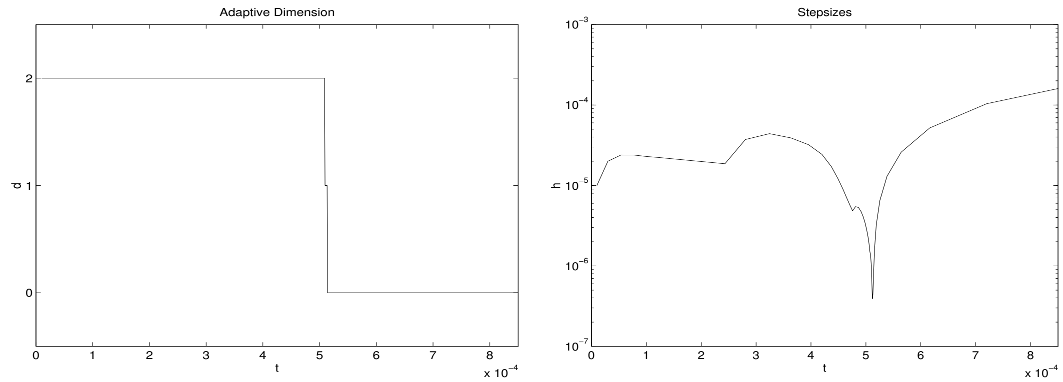


Figure 3: Example 1 - Splitting dimension and timestep behavior for 8 automatically selected block-Schur decompositions

In order to study the question of whether we could have done with even lower dimension, we turned off our QSSA error control and prescribed $d = 1$ throughout the whole integration. In this case, however, we were unable to get any result at all, because the stepsizes shrunk below 10^{-20} already at the starting point. Therefore, we conclude that $d = 2$ is a minimum dimension necessary to model this combustion process correctly. The success of our automatic dimension reduction algorithm can also nicely be seen by comparison with the quite sophisticated analytical singular perturbation treatment of Hoppensteadt et al. in [14].

Example 2: Oregonator (due to Field and Noyes [9]). This problem comprises only 3 ODEs:

$$
\begin{aligned}
\dot{x}_1 &= 77.27(x_2 + x_1(1 - 8.375 \cdot 10^{-6} x_1 - x_2)) \\
\dot{x}_2 &= \frac{1}{77.27}(x_3 - (1 + x_1)x_2) \\
\dot{x}_3 &= 0.161(x_1 - x_2), \quad x(0) = (1, 2, 3)
\end{aligned}
$$

It is, however, well-known to be quite challenging to any numerical stiff integrator due to its oscillatory behavior: any errors introduced at some integration step will not die out furtheron, since the system is not dissipative. It is nice to see that once again our algorithm performs well. The obtained numerical solution, which is known to vary over several orders of magnitude (note the log-scale of presentation!), is given in Fig. 4.

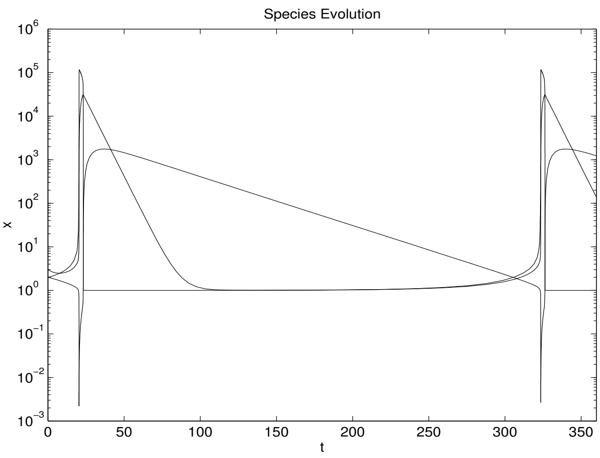
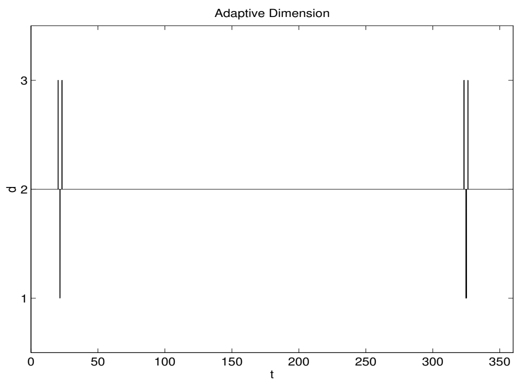


Figure 4: Example 2 - Oscillatory behavior of chemical species.

As shown in Fig. 5, the full dimension $d = 3$ is only needed within the transient phase. In between a dimension $d = 2$ is sufficient.



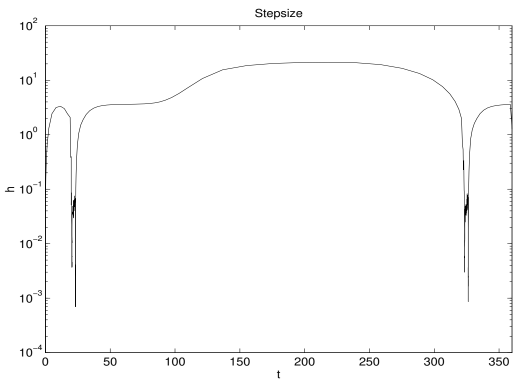


Figure 5: Example 2 - Splitting dimension and timestep behavior.

A total number of 94 block-Schur decompositions turned out to be necessary during 328 integration steps. In this example, using the *explicit* Euler discretization for the slow components of the system caused a slightly non-smooth behavior in the stepsize control. Therefore we chose the linearly implicit Euler discretization for the slow components as well.

Example 3: Thermal decomposition of n-hexane (due to Isbarn, Ederer and Ebert [15]). This problem consists of $n = 59$ species and 240 elementary chemical reactions. The temperature was set to 728^o K . In order to establish the right hand side of the ODE system correctly from the input in terms of chemical reactions only, the algorithm was implemented within the software package LARKIN [2, 1]. The numerical results obtained by the dynamic dimension reduction method were indistinguishable from those obtained by solving the original ODE system. An important feature observed in this larger example turned out to be that a quite good Jacobian approximation was necessary to obtain an efficient dynamical reduction sequence. Any poor Jacobian approximation did affect Newton's method for the projection of initial values onto the slow manifold. As a consequence, the algorithm then chose dimensions higher than necessary. We therefore took the analytic Jacobian option in LARKIN. The associated dynamic dimension reduction sequence is plotted in Fig. 6 for two different time intervals. In this example, the slow components of the system were integrated numerically by an explicit Euler scheme.

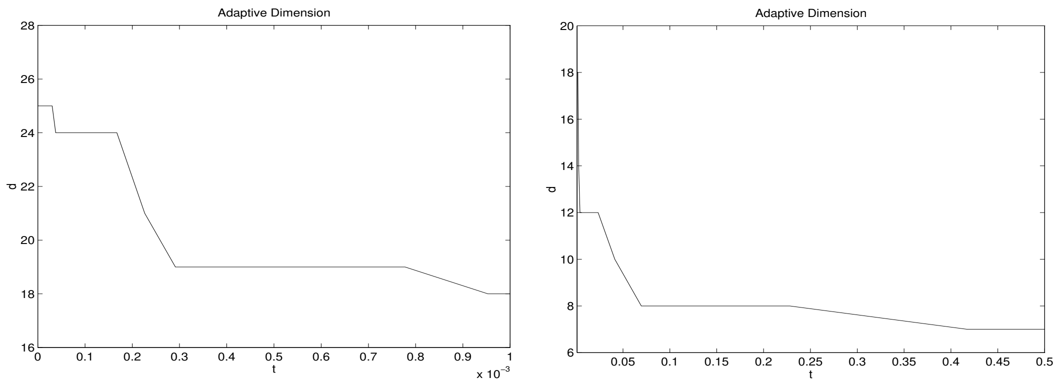


Figure 6: Example 3 - Dynamic dimension reduction over two time intervals.

Acknowledgements

The authors wish to thank Ulrich Maas for many helpful discussions about his chemistry reduction method ILDM. This work was supported by the DFG within the DFG-Schwerpunkt "Ergodentheorie, Analysis und effiziente Simulation dynamischer Systeme".

References

[1] G.Bader, U.Nowak, P.Deuflhard: *An advanced simulation package for large chemical reaction systems.* In: R.Aiken (ed.): *Stiff Computation.* Oxford Univ. Press, pp. 255-264 (1985)

[2] P.Deuflhard, G.Bader, U.Nowak: *LARKIN - a Software Package for the Numerical Simulation of LARge Systems Arising in Chemical Reaction KINetics.* in [7], pp. 38-55 (1981)

[3] P.Deuflhard, F.Bornemann: *Numerische Mathematik II. Integration gewöhnlicher Differentialgleichungen.* de Gruyter: Berlin, New York (1994)

[4] P.Deuflhard, E.Hairer, J.Zugck: *One-step and Extrapolation Methods for Differential-Algebraic Systems.* Numer. Math., vol 51, pp. 501-516 (1987)

[5] P.Deuflhard, A.Hohmann: *Numerical Analysis. A First Course in Scientific Computation.* de Gruyter: Berlin, New York (1993)

[6] P.Deuflhard, U.Nowak: *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems.* Ber. Bunsenges. Phys. Chem., vol 90, pp. 940-946 (1986)

[7] K.H.Ebert, P.Deuflhard, W.Jäger (eds.) *Modelling of Chemical Reaction Systems.* Springer Series in Chem Phys, vol 18, Berlin, Heidelberg, New York (1981)

[8] L.A.Farrow, D.Edelson: *The steady-state approximation, fact or fiction?* Int.J.Chem.Kin, vol 6, pp. 787ff. (1974)

[9] J.Field, R.M.Noyes: *Oscillations in Chemical Systems. IV: Limit cycle behaviour in a model of a real chemical reaction.* J.Chem.Phys, vol 60, pp. 1877-1884 (1974)

[10] G.H.Golub, Ch.F.van Loan: *Matrix Computations.* Johns Hopkins University Press (1985)

[11] G.H.Golub, J.H.Wilkinson: *Ill-conditioned Eigensystems and Computation of the Jordan Canonical Form.* SIAM Review 18, pp. 578-619 (1976)

[12] E.Hairer, G.Wanner: *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer Series in Comp. Math., vol 14, Berlin, Heidelberg, New York (1991)

[13] J.Heroth: *Adaptive Dimensionsreduktion chemischer Reaktionssysteme.* Freie Universität Berlin, Diploma thesis (1995)

[14] F.C.Hoppensteadt, P.Alfeld, R.Aiken: *Numerical Treatment of Rapid Chemical Kinetics by Perturbation and Projection Methods.* in: [7], pp. 31-37 (1981)

[15] G.Isbarn, H.J.Ederer, E.H.Ebert *The Thermal Decomposition of n-Hexane: Kinetics, Mechanism and Simulation.* in: [7], pp. 235-248 (1981)

[16] LAPACK. User's Guide, Philadelphia (1992)

[17] U.Maas: *Automatische Reduktion von Reaktionsmechanismen zur Simulation reaktiver Strömungen.* Institut für Technische Verbrennung der Universität Stuttgart, Habilitation thesis, (1993)

[18] U.Maas, S.B.Pope: *Simplifying Chemical Kinetics: Intrinsic Low-Dimensional Manifolds in Composition Space.* Combustion and Flame, vol 88, pp. 239-264, (1992)

[19] R.E.O'Malley: *Introduction to Singular Perturbations.* Academic Press, New York (1974)

[20] W.C.Rheinboldt: *Differential-Algebraic Systems as Differential Equations on Manifolds.* Math. Comp. vol 43, pp. 473-482 (1984)

[21] A.B.Vasil'eva: *Asymptotic Behaviour of Solutions to Certain Problems involving Nonlinear Differential Equations...* Usp. Mat. Nauk (Russian) vol 18, pp. 15-86 (1963). Russian Math. Surveys, vol 18, Nr.3, pp.13-84 (Translation)