Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 27 (2003) 951–964
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Enabling stability analysis of tubular reactor models using PDE/PDAE integrators

E.D. Koronaki [a], A.G. Boudouvis [a,*], I.G. Kevrekidis [b]

[a] School of Chemical Engineering, National Technical University of Athens, Athens 15780, Greece
[b] Department of Chemical Engineering, Princeton University, Princeton, NJ 08544, USA

Received 14 March 2002; accepted 11 December 2002

Abstract

We discuss the construction of computational superstructures enabling time-dependent process simulation codes to perform stability, continuation and bifurcation calculations—tasks in principle not accessible to them—directly. The basis of the approach is the so-called Recursive Projection Method of Shroff and Keller (SIAM Journal of Numerical Analysis 31). We discuss its implementation and performance for the detection of different types of bifurcations (with emphasis on Hopf bifurcations) as well as slight modifications appropriate for index 1 partial differential/algebraic equations (PDAE) simulators. Tests that help discriminate between physical and numerical (spurious) bifurcations detected in the process are discussed and illustrated through the standard example of a tubular reactor with a single irreversible exothermic reaction.

Keywords: Recursive projection method; Stability analysis; Timesteppers; Tubular reactor

1. Introduction

The development and validation of reliable dynamical process simulation codes requires a significant investment in time and effort. It is, therefore, desirable to extend, if possible, their capabilities in order to produce results that relate to the stability and bifurcation analysis of the system under study. The modification of 'timesteppers', algorithms that evolve the system state in time, to perform various bifurcation-theoretic tasks has been recently reviewed by Tuckerman and Barkley (2000).

Such modifications are best performed if one has direct access to the source simulation code, and is of course much more straightforward if one is the author of the simulation code. It is not infrequent, however, in chemical process simulation, that the 'timestepper' is a legacy code, and thus not so easily modifiable. It is also often the case that the simulator is a commercial package that integrates systems of partial differential equations (PDEs) and/or partial differential algebraic equations (PDAEs) such as Aspen Plus (http://www.aspentech.com), or gPROMS (http://www.psenterprise.com). Fluid mechanical solver codes like PHOENICS (http://www.cham.co.uk), or CFX (http://www.software.aeat.com/cfx) and FIDAP (http://www.fluent.com) also come to mind. These codes, while often embodying state-of-the-art numerical analysis and scientific computation techniques, are practically a black box for the user. Modifying them to perform bifurcation related tasks may therefore be impractical, or impossible (see, however, the Section 5). Keller and his student Shroff proposed, relatively recently (Shroff & Keller, 1993), a method that circumvents this problem. Through the construction of a computational superstructure around the integrator, which is treated as a 'black box', they were able to compute steady state solutions, perform stability analysis, and continuation/bifurcation tasks without altering the basic timestepper. This computational feature of black box utilization of existing codes is favored by recent developments in open standard interfaces; a case in point is the Global CAPE-OPEN interface (http://www.global-cape-open.org), which per-

* Corresponding author. Tel.: +32-10-7723241; fax: +32-10-7723155.
E-mail address: boudouvi@chemeng.ntua.gr (A.G. Boudouvis).

0098-1354/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0098-1354(03)00004-8

mits the integration of simulation components from different sources, i.e. commercial, academic or 'in house'. The Shroff and Keller technique (which they named Recursive Projection Method (RPM)) was proven to work for problems with well-defined spectral properties: a few eigenvalues of the linearization around the steady states of interest should lie in a strip around the imaginary axis in the complex plane; then a spectral gap should exist, corresponding to a 'large' separation of time scales for the physical problem in question; and finally the rest of the spectrum should lie to the left of the gap, and would correspond to 'strongly' stable modes.

The RPM is used to accelerate and stabilize the convergence of the integration code to the steady state, which may be extremely slow if the steady state is close to marginal stability/bifurcation, or it may simply not happen if the steady state is slightly unstable. The RPM also makes computer-assisted stability analysis possible through small (in our work typically O(10)) eigenvalue problems. The focus of this work lies in investigating the effect of the spectrum of the underlying physical model, as well as its modification by the particular numerical integration method, on the RPM induced convergence. We will see that the RPM can indeed stabilize even mildly unstable timesteppers, no matter if the instability is physical or numerical, and converge to the correct steady states. In addition, we will demonstrate that it may be possible to discriminate between critical modes of the physical problem and spurious critical modes of the particular timestepper used, by the form of the reconstructed eigenvectors of the linearization at the steady state. The presence of spurious unstable modes is the focus of many publications such as Boffi, Duran and Gastaldi (1999) who studied the performance of several finite element schemes with regard to the generation of spurious eigenvalues in a fluid-structure interaction case. The existence of spurious bifurcations could lead to erroneous predictions of critical parameters (Sureshkumar, 2001) and therefore, it is important to either distinguish them from the physical modes or to filter them out altogether. Recently Yeih, Chen and Chang (1999) proposed the natural integral equation of dual multiple reciprocity method (MRM) that was applied on an Euler–Bernoulli beam as an approach that would avoid spurious bifurcations.

Many of the physical problems encountered in engineering practice are described by DAE systems, and the interest in 'wrapping' the RPM around DAE solvers thus arises naturally. DAEs are characterized by their index, which is the number of times that all or part of the constraints of the DAE system must be differentiated with respect to time (t), to yield an ODE system in the original variables (Brenan, Campbell & Petzold, 1995). The index is a measure of the singularity of a system and the higher it is, the more difficult the system is to solve numerically. A typical example of index 2 DAEs are the Navier–Stokes equations. Keller and von Sosen (1995) demonstrated the transformation of the Navier–Stokes equations into a timestepper on the discretized differential variables only.

We will discuss below that the RPM can be transparently applied to semi-explicit DAE systems of index 1 as a procedure operating on the set of differential variables; the important element here is, as one might expect, the issue of consistent initialization.

Our illustrative example is a time dependent reaction–convection–diffusion system (Subramanian & Balakotaiah, 1996). It is compactly written:

$$C\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u},\ \mathbf{p}) \tag{1a}$$

$$B(\mathbf{u},\ \mathbf{p}) = \mathbf{0} \tag{1b}$$

$$\mathbf{u}(t=0) = \mathbf{u}_0 \tag{1c}$$

where C is the capacity matrix; $\mathbf{p}$, the vector of parameters and $\mathbf{u}(\mathbf{x},\ t)$ is the vector of the state variables, where $\mathbf{x}$ is the vector of the spatial coordinates. Eq. (1b) represents the boundary conditions satisfied by $\mathbf{u}$, and $\mathbf{u}_0$ is an initial condition. The operator $\mathbf{f}$ contains linear and nonlinear components that describe the diffusion, convection and reaction terms. Nonlinear models such as the above can display in certain regions of the parameter space multiple steady states and periodic states in space/time. The interest lies in the way the behavior of the system changes as the parameter vector $\mathbf{p}$ varies. In many physical applications, the interest lies in the steady state dependence on a particular parameter λ, the so-called distinguished parameter.

In this work, the RPM is applied to conduct stability analysis of a tubular reactor model that can be integrated in time, upon spatial discretization, either as an ODE or as a DAE system. In the case of the DAE system, the integrator used here is DASSL (Petzold, 1982), an extensively used and validated powerful software.

To illustrate the interplay between the spectrum of the true physical problem, and the spectrum induced by the method of time integration, the PDE problem is spatially discretized to a system of ODEs with the Galerkin/finite element method. The resulting ODE system is then discretized in time with a forward Euler scheme. By altering the time step, the stability of the timestepper is affected, allowing the study of the RPM in conjunction with the time integration scheme.

A number of researchers have studied and compared methodologies for the investigation of the steady state and dynamical behavior of chemically reacting systems; among them, Jensen and Ray (1982) who dealt with the cooled tubular reactor model in one spatial dimension and Subramanian and Balakotaiah (1996) who focused

on distributed reactor models. Most of the techniques investigated in these studies require the steady state formulation of the models and, in addition, generalized eigenvalue computations. As we will demonstrate later on, the RPM enables stability analysis of the steady state problem by using an existing timestepper. This is achieved by approximating the action of the Jacobian of the timestepper in the low dimensional subspace of its most dangerous eigenmodes. The eigenvectors and eigenvalues of the Jacobian of the steady state problem can be approximately reconstructed from those of the reduced timestepper Jacobian matrix.

A thorough presentation and analysis of the RPM is offered in the Ph.D. thesis of Lust (1997). In the thesis of von Sosen (1994) the RPM is applied to DAEs in incompressible flow computations. Love (1999) applied the RPM to Kolmogorov and Taylor-vortex flows. The RPM has also been used successfully for detection of stable and unstable periodic orbits of large scale dynamical systems (Lust, Roose, Spence & Champneys, 1998) as well as of periodically forced systems like pressure swing adsorption (Theodoropoulos, Bozinis, Siettos, Pantelides & Kevrekidis, 2001). Recently the RPM was applied to the solution of linear systems (Burrage, Erhel, Pohl & Williams, 1998). Furthermore, an RPM-based method was proposed, in order to perform 'coarse' stability and bifurcation analysis using microscopic evolution rules directly (Theodoropoulos, Qian & Kevrekidis, 2000; Gear, Kevrekidis & Theodoropoulos, 2002).

2. The mathematical model

In this work the focus is on the case of a first order irreversible reaction for which the equations of the pseudohomogeneous axial dispersion model are two coupled nonlinear parabolic PDEs which, in dimensionless form are (Jensen & Ray, 1982):

$$\left.\begin{aligned}\frac{\partial x_1}{\partial t} &= Pe_1^{-1}\frac{\partial^2 x_1}{\partial z^2}-\frac{\partial x_1}{\partial z}+Da(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\\ \frac{\partial x_2}{\partial t} &= Pe_2^{-1}\frac{\partial^2 x_2}{\partial z^2}-\frac{\partial x_2}{\partial z}-\beta x_2+BDa(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\end{aligned}\right\}, \quad 0<z<1 \tag{2}$$

with boundary conditions:

$$\left.\begin{aligned}\frac{\partial x_1}{\partial z} &= Pe_1 x_1\\ \frac{\partial x_2}{\partial z} &= Pe_2 x_2\end{aligned}\right\}, \quad z=0 \qquad \left.\begin{aligned}\frac{\partial x_1}{\partial z} &= 0\\ \frac{\partial x_2}{\partial z} &= 0\end{aligned}\right\}, \quad z=1 \tag{3}$$

Here x_1 and x_2 are the conversion and the dimensionless temperature, respectively. Da is the Damköhler number, which represents the ratio of reactor space time to the characteristic reaction time. Pe_1 and Pe_2 are the Péclet numbers for mass and heat transfer, β is a dimensionless heat transfer coefficient, B is a dimensionless adiabatic temperature rise and γ is a dimensionless activation energy. In the current implementation, Da is the distinguished parameter and the remaining ones have fixed values: $Pe = Pe_1 = Pe_2 = 5$, $\beta = 1.5$, $B = 12$ and $\gamma = 20$.

The forward Euler scheme is used for discretization in time and the equations are transformed as follows:

$$\begin{aligned}x_1^{n+1} &= x_1^n+\Delta t\left[Pe_1^{-1}\frac{\partial^2 x_1}{\partial z^2}-\frac{\partial x_1}{\partial z}+Da(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\right]\\ x_2^{n+1} &= x_2^n+\Delta t\left[Pe_2^{-1}\frac{\partial^2 x_2}{\partial z^2}-\frac{\partial x_2}{\partial z}-\beta x_2+BDa(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\right]\end{aligned} \tag{4}$$

The (explicit) forward Euler scheme is not the scheme of choice for integrating in time, because it is stable only under very strict mesh/time-step conditions and it has serious disadvantages for solving realistic large scale problems. We use it here, however, to demonstrate the interaction of the RPM with an *unstable* time-integrator, since the forward Euler scheme can become unstable just by slightly increasing the time step, Δt. It will be shown that the RPM-assisted integration can converge to the steady state even though the integrator is unstable, whether the instability is due to a 'true' or 'physical' instability, or to a spurious, numerical instability. This will be correlated to the spectrum of the timestepper and that of the underlying space-discretized physical problem.

The Galerkin method of weighted residuals combined with finite element quadratic basis functions is used for spatial discretization, on a grid consisting of N points. The Galerkin weighted residuals of Eq. (4) read:

$$\begin{aligned}&\int_D\left\{x_1^{n+1}\phi^i-x_1^n\phi^i-\Delta t\left[Pe_1^{-1}\frac{\partial^2 x_1}{\partial z^2}\ \phi^i+\frac{\partial x_1}{\partial z}\ \phi^i-Da(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\phi^i\right]\right\}dz\\ &\int_D\left\{x_2^{n+1}\phi^i-x_2^n\phi^i-\Delta t\left[Pe_2^{-1}\frac{\partial^2 x_2}{\partial z^2}\ \phi^i+\frac{\partial x_2}{\partial z}\ \phi^i+\beta x_2-BDa(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\phi^i\right]\right\}dz\end{aligned} \tag{5a}$$

where $D \equiv [0,1]$, $i = 1, \ldots, N$. The residuals are set to zero:

$$\int_D x_1^{n+1}\phi^i \mathrm{d}z = \int_D x_1^n \phi^i \mathrm{d}z + \Delta t \int_D \left[Pe_1^{-1}\frac{\partial^2 x_1}{\partial z^2} - \frac{\partial x_1}{\partial z} + Da(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\right]\phi^i \mathrm{d}z$$
$$\int_D x_2^{n+1}\phi^i \mathrm{d}z = \int_D x_2^n \phi^i \mathrm{d}z + \Delta t \int_D \left[Pe_2^{-1}\frac{\partial^2 x_2}{\partial z^2} - \frac{\partial x_2}{\partial z} - \beta x_2 + BDa(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\right]\phi^i \mathrm{d}z \quad (5b)$$

The unknowns are approximated by:

$$x_1 = \sum_{j=1,N} x_{1j}(t)\phi^j(z)\; x_2 = \sum_{j=1,N} x_{2j}(t)\phi^j(z) \quad (6)$$

where $\phi^j(z)$ are quadratic finite element basis functions and $x_{1j}(t)$ and $x_{2j}(t)$ are the values of the unknowns x_1 and x_2, respectively, at the nodes of the discretization at time t.

Upon substitution of the unknowns from Eq. (6) into Eq. (5b) the following system is obtained:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \mathbf{M}^{-1}\mathbf{R} = \mathbf{F}(\mathbf{U}^n) \quad (7)$$

where $\mathbf{U}^{n+1}$ and $\mathbf{U}^n$ are the vectors of the nodal values of the unknowns at time $n+1$ and n, respectively, i.e. $\mathbf{U}^{(n)} = [x_{11}^n, x_{12}^n, \ldots, x_{1N}^n, x_{21}^n, x_{22}^n, \ldots, x_{2N}^n]$. $\mathbf{M}$ is the so-called mass matrix, $M_{ij} = \int_D \phi^i\phi^j \mathrm{d}z$; it is symmetric positive definite so $\mathbf{M}^{-1}$ exists. $\mathbf{R}$ is the spatially discretized right-hand-side of Eq. (2), that is, the quantity multiplied by Δt in the residual Eq. (5b); $\mathbf{R}$ is evaluated at $\mathbf{U} = \mathbf{U}^n$.

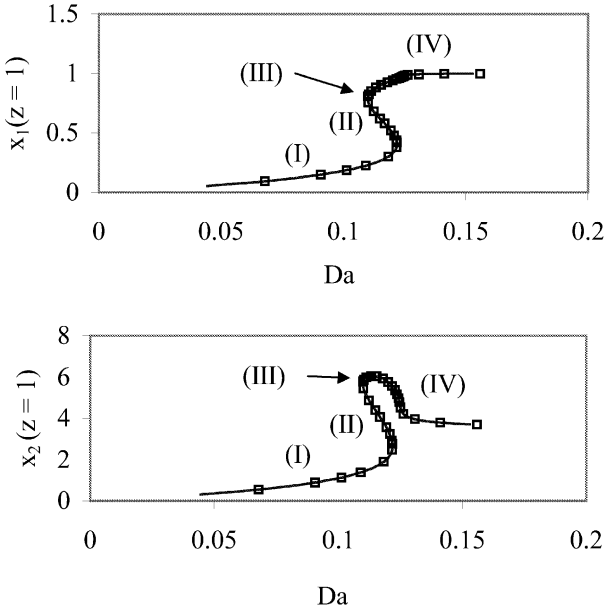

Fig. 1. Exit conversion, $x_1(z=1)$, and dimensionless temperature, $x_2(z=1)$, for varying Da. — RPM, □ full Newton. (I) Stable steady states, (II) unstable steady states, (III) Hopf bifurcation (immediately followed by a turning point), (IV) stable steady states.

At particular parameter values, the system of Eq. (2) has steady solution branches with two turning points as well as a time periodic solution branch which emerges through a Hopf bifurcation. The turning points appear at $Da \approx 0.122$ and $Da \approx 0.110$ and the Hopf at $Da \approx 0.11003$, extremely close to the second turning point.

The exit conversion and dimensionless temperature over a range of parameter values are computed, for purposes of comparison with the RPM results, by solving the nonlinear steady state equations, $\mathbf{R} = \mathbf{0}$ (cf. Eq. (7)), for the nodal values of he (time independent) unknowns; Newton's method along with arc-length continuation (Bolstad & Keller, 1986) are employed. Branches in the solution space are shown in Fig. 1. Typical conversion and temperature profiles for each section of the solution branch are shown in Fig. 2.

3. The recursive projection method

In this section, a brief discussion of the RPM is offered. The method is described in detail by Shroff and Keller (1993) and a family of related 'Newton–Picard' methods in several Theses (e.g. Lust, 1997). Along the same line is earlier work by Jarausch and Mackens (1984, 1987a,b) who propose a so-called 'adaptive condensation' method for symmetric systems.

Steady state solutions of the parameter-dependent dynamical system:

$$\mathbf{u} = \mathbf{f}(\mathbf{u},\ Da) \quad (8)$$

are also solutions of the spatially discretized fixed point iterative procedure (cf. Eq. (7)):

$$\mathbf{U}^{n+1} = \mathbf{F}(\mathbf{U}^n,\ Da;\ T), \quad \mathbf{F}{:}R^n \times R \to R^N \quad (9)$$

It is, in our case, $\mathbf{u} = [x_1, x_2]$ and $\mathbf{f}$ the operator of the right-hand-side of Eq. (2). $\mathbf{U}$ is the vector of nodal values of the unknowns. $\mathbf{F}$ denotes a system timestepper, such as the forward Euler scheme. Specifically, it is the result of numerical integration over a time interval T—the *reporting horizon* of the timestepper—of the spatial discretization, through finite elements on a mesh of N grid points, of the system in Eq. (8). Consider the

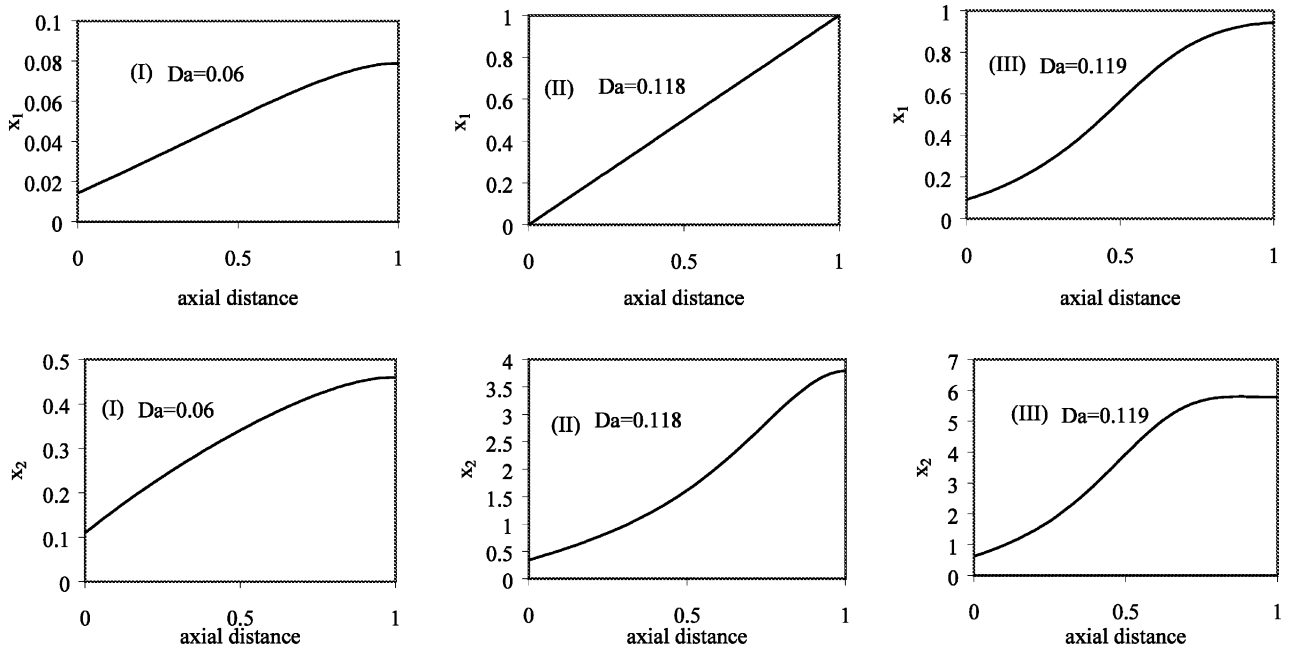

Fig. 2. Typical profiles at different regions of the solution branch: stable branch, (II) unstable branch, (III) at the Hopf point. Regions (I), (II) and (III) are shown in Fig. 1.

invariant subspace P, that corresponds to the eigendirections in which the linearized map is slowly contracting or even slowly expanding and its orthogonal complement Q. By $\mathbf{P}$ and $\mathbf{Q}$ we also denote the orthogonal projectors of R^N on P and Q, respectively. The solution $\mathbf{U}$ is decomposed into $\mathbf{p}$ and $\mathbf{q}$ such that $\mathbf{U}=\mathbf{p}+\mathbf{q}$, where $\mathbf{p}$ and $\mathbf{q}$ are the projections of $\mathbf{F}(\mathbf{U}, Da; T)$ onto P and Q, respectively. The RPM stabilizes, under certain assumptions, time integrators in the form of fixed point iterative procedures, such as Eq. (9) by first computing an approximation of P and consequently of Q. The projection $\mathbf{p}$ is then computed by performing a Newton method step on P; $\mathbf{q}$ is determined by the projection of $\mathbf{F}(\mathbf{U}, Da; T)$ on Q. Subsequently the Picard iteration acts on the sum $\mathbf{U}=\mathbf{p}+\mathbf{q}$.

The fixed point iteration Eq. (9) is stable when all the eigenvalues of the matrix $\mathbf{F}_{\mathbf{U}}(\mathbf{U}, Da)\equiv\partial\mathbf{F}(\mathbf{U}, Da)/\partial\mathbf{U}$ lie in the unit disk (we have now suppressed the dependence on the time step, or integrator reporting horizon, T). In this work, the RPM will be implemented both when the integration scheme is numerically stable and when it is not. The stabilized iterative procedure takes the form:

1. $\mathbf{p}^{(0)}=\mathbf{P}\mathbf{U}^{(0)}(Da), \quad \mathbf{q}^{(0)}=\mathbf{Q}\mathbf{U}^{(0)}(Da)$

2. Do until convergence:

a) Newton step:

$$\mathbf{p}^{(v+1)}=\mathbf{p}^{(v)}+(\mathbf{I}-\mathbf{P}\mathbf{F}_{\mathbf{U}}(\mathbf{U}^{(v)}, Da)\mathbf{P})^{-1}\times(\mathbf{P}\mathbf{F}(\mathbf{U}^{(v)}, Da)-\mathbf{p}^{(v)}) \quad (10)$$

b) Fixed point iteration:

$$\mathbf{q}^{(v+1)}=\mathbf{Q}\mathbf{F}(\mathbf{U}^{(v)}, Da).$$

3. $\mathbf{U}^*(\lambda)=\mathbf{p}^{(v_{\text{final}})}+\mathbf{q}^{(v_{\text{final}})}\equiv\mathbf{p}^*+\mathbf{q}^*$

The main assumption is the existence of a gap between the 'strongly stable' and the 'slow' eigenvalues of the linearization around the fixed point in question. For the method to work in practice, the dimension, m, of this latter 'slow' subspace should be comparatively low (typically O(10) eigenmodes). While this may appear restrictive, there is a large class of dissipative system models for which this is the case of interest close to primary instabilities.

The RPM in conjunction with the pseudo-arc-length procedure proposed by Shroff and Keller, is applied to the fixed point iterative procedure that results from the discretization in time and space of the system of Eq. (2), i.e. Eq. (5b) (cf. Eq. (7)). The maximum dimension of the unstable subspace detected by our implementation of the RPM, in this case (for the bifurcation diagram in Fig. 1 and with a forward Euler time step of 10^{-4}) is $m=6$. The eigenvalues of the $m\times m$ matrix $\mathbf{H}=\mathbf{Z}^{\mathrm{T}}\mathbf{F}_{\mathbf{U}}\mathbf{Z}$, where $\mathbf{Z}\in R^{N\times m}$ is an approximation of the orthonormal basis of the subspace P computed by the RPM, correspond to the eigenvalues of $\mathbf{F}_{\mathbf{U}}$ with the largest absolute values.

Since (cf. Eq. (7)):

$$\mathbf{F}_{\mathbf{U}} = \mathbf{I} + \Delta t \mathbf{M}^{-1}\mathbf{R}_{\mathbf{U}} \tag{11}$$

the eigenvectors of $\mathbf{F}_{\mathbf{U}}$ are eigenvectors of $\mathbf{M}^{-1}\mathbf{R}_{\mathbf{U}}$ as well; namely, if (v, $\mathbf{y}$) is an eigenpair of the simple eigenproblem $\mathbf{F}_{\mathbf{U}}\mathbf{y} = v\mathbf{y}$, then ($\omega$, $\mathbf{y}$) is an eigenpair of the generalized eigenproblem:

$$\mathbf{R}_{\mathbf{U}}\mathbf{y} = \omega\mathbf{M}\mathbf{y} \text{ with } \omega \equiv \frac{v-1}{\Delta t} \tag{12}$$

For fixed $Da = 0.09$, using the forward Euler time integration scheme with time step $T = \Delta t = 10^{-4}$, the RPM yielded convergence to steady state after identifying a slow subspace of dimension $m = 3$. The eigenvectors of the original PDEs, $\mathbf{y}$, of the eigenvalues that are closest to 1 are reconstructed by the eigenvectors, $\mathbf{x}$, of $\mathbf{H}$, using $\mathbf{y} = \mathbf{Z}\mathbf{x}$. In order to independently validate the reconstructed eigenvectors, an implicitly restarted Arnoldi procedure (Saad, 1996) was performed and the dominant eigenvectors of $\mathbf{F}_{\mathbf{U}}$ are approximated. In order to determine a Krylov basis $K_n = \langle \mathbf{r}_0, \mathbf{F}_{\mathbf{U}}\mathbf{r}_0, \ldots, \mathbf{F}_{\mathbf{U}}^{n-1}\mathbf{r}_0 \rangle$, the Arnoldi algorithm need only approximately compute the matrix–vector product $\mathbf{F}_{\mathbf{U}}\mathbf{r}$, where $\mathbf{r}$ is an initial unit vector. In the present implementation, the initial vector was a normalized linear combination of the vectors that are computed by the RPM. The procedure is periodically restarted after a prescribed dimension of the Krylov subspace is reached. Finally, in this implementation the convergence criterion is satisfied after three or four restarts. The RPM-derived and the corresponding Arnoldi-approximated eigenvectors computed at the steady state, at parameter value $Da = 0.09$, are shown in Fig. 3. The difference between them is reasonable, considering that the RPM computes an approximation of the dominant eigenspace. While this approximation is sufficient for stabilizing the time integrator and converging to the fixed point, it is not expected to be quantitative even upon convergence of the overall process.

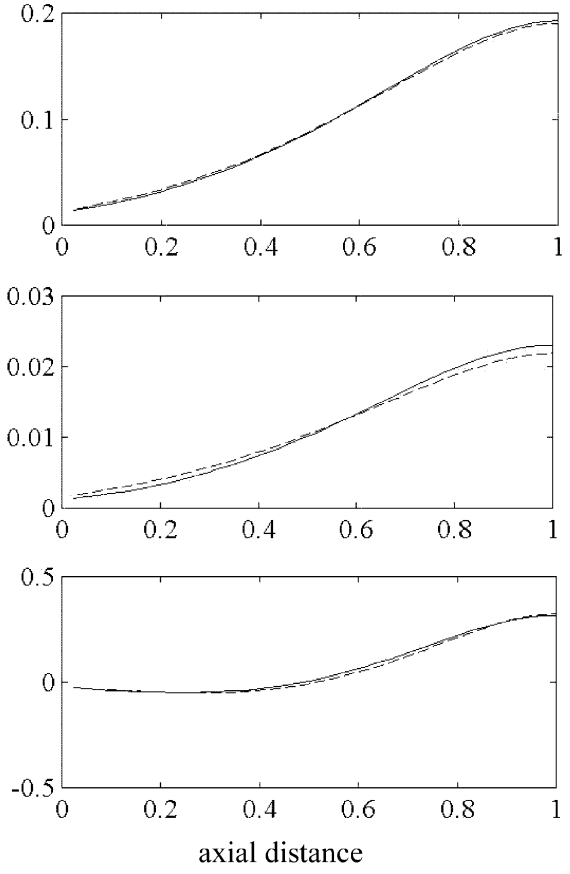

Fig. 3. The three dominant eigenvectors of the matrix $\mathbf{R}_{\mathbf{U}}$ of the steady state problem. Plotted are the axial profiles of the eigenvectors corresponding to x_1; $Da = 0.09$. — RPM-approximated eigenvectors, -- eigenvectors computed with the Arnoldi algorithm.

4. Implementation of the RPM on a system of differential/algebraic equations

Implicit systems of differential/algebraic equations are written in the form:

$$\mathbf{F}(t,\ \mathbf{U},\ \mathbf{U}') = 0$$

$$\mathbf{U}(t_0) = \mathbf{U}_0$$

$$\mathbf{U}'(t_0) = \mathbf{U}'_0 \tag{13}$$

where $\mathbf{U}_0$, $\mathbf{U}'_0$, $\mathbf{U}$ and $\mathbf{U}'$ are N-dimensional vectors.

Problems of this type are the models of choice for several applications and frequently occur in the numerical method-of-lines treatment of PDEs. In these applications, it might not be possible to solve explicitly for $\mathbf{U}'$ because $\mathbf{F}_{\mathbf{U}'}$ may be singular; this happens when the PDE system includes both evolutionary and non-evolutionary equations.

Our purpose here is to implement a form of RPM for a semi-explicit system of index 1 DAEs (from now on when we refer to DAEs it will be implicitly implied that they are semi-explicit index 1 DAEs) without performing an explicit reduction of the DAE system to a system of ODEs on the differential variables.

Sometimes it is conceptually useful to consider that a system of differential/algebraic equations:

$$x = f(x,\ y) \tag{14a}$$
$$g(x,\ y) = 0 \tag{14b}$$

may be thought of as resulting from a pseudo-steady state approximation of a system of the type:

$$x = f(x,\ y) \tag{15a}$$
$$y = \frac{1}{\tau}\ g(x,\ y) \tag{15b}$$

in which Eq. (14b) is replaced by a stiff, but stable, ODE, namely Eq. (15b), in which $\tau \to 0$. Seen in this

light, the RPM could be implemented on the full vector of differential and algebraic variables without any other consideration in exactly the same way that it would be done for the ODE system Eq. (10).

An important practical issue regarding the RPM treatment of DAE systems of equations lies in the fact that at certain steps of the RPM procedure, where the current iterate, **U**, is not a result of integration, the algebraic equations of the system cease to be satisfied. This happens mainly at two points of the RPM algorithm:

1) The computation of the matrix–vector product $\mathbf{F}_\mathbf{U}\mathbf{Z}_i$ through finite differences, involves perturbing the vector **U**, by a small fraction, ε, of the basis vector $\mathbf{Z}_i$ and then integrating in order to compute $\mathbf{F}(\mathbf{U}+\varepsilon\mathbf{Z}_i)$:

$$\mathbf{F}_\mathbf{U}\mathbf{Z}_i \approx \frac{1}{\varepsilon}[\mathbf{F}(\mathbf{U}+\varepsilon\mathbf{Z}_i)-\mathbf{F}(\mathbf{U})] \tag{16}$$

2) At each step of the RPM, **U** is reconstructed by the result of the Picard and the Newton step:

$$\mathbf{U}^{(v)} = \mathbf{p}^{(v)} + \mathbf{q}^{(v)} \tag{17}$$

If the initial condition in the algebraic variables is not consistent, DAE solvers, such as the code DASSL used here, or SPRINT (Berzins & Furzeland, 1985; Petzold, 1982Brenan et al., 1995) typically used to perform an implicit small step in order to achieve consistent initialization. The algebraic equations are, in effect, treated as 'very fast evolving' time dependent equations. Notice that in the context of finding steady states, as we do here, the effect of such a small step on the accuracy of a transient trajectory is not an issue. Alternatively, consistent initialization may be performed by solving the algebraic equations, e.g. through Newton iteration, given the initial values of the differential variables. Extensive research in recent years has focused on algorithms for performing accurate such solutions (Barton & Pantelides, 1994; Campbell, 1986; Kröner, Marquardt & Gilles, 1992; Leimkuhler, Petzold & Gear, 1991; Pantelides, 1988; Brown, Hindmarsh & Petzold, 1998); symbolic preprocessing and graph theoretic algorithms to take advantage of the structure of the equations involved have been invoked (Kröner et al., 1992; Pantelides, 1988). The approach involved in consistently initializing the most recent version of DASPK is described by Brown et al. (1998). For such initializations, we can think that the DAE solver treats the algebraic equations, from the RPM point of view, as 'infinitely' fast differential equations. Whether 'very' or 'infinitely' fast, these equations do not interfere with the detection of the slow stable/unstable subspace by the RPM timestepper (whose reporting horizon is assumed long compared with this 'initialization time'). We, therefore, observe that as long as the DAE solver does a good job (through direct Newton or through a small implicit step) for consistent initialization, RPM can in principle be wrapped around a DAE integrator 'without worrying' about its DAE nature. The correct fixed points will be converged on, and, more importantly, the correct stability (corresponding to the slow stable/unstable) modes will be determined through the time-stepper-based approach.

In order to demonstrate this approach, the tubular reactor model given by Eq. (2) and the boundary conditions in Eq. (3) is solved as a set of differential/algebraic equations, where the two boundary conditions are treated as extra algebraic equations. The PDEs are discretized in space using the Galerkin/finite element method and yield $2N$ differential equations, which, along with the two algebraic equations, are integrated in time with the code DASSL, as follows.

The Galerkin weighted residuals of Eq. (2) read:

$$R_i = \int_D \left[\frac{\partial x_1}{\partial t}\ \phi^i - Pe_1^{-1}\frac{\partial^2 x_1}{\partial z^2}\ \phi^i + \frac{\partial x_1}{\partial z}\ \phi^i - Da(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\phi^i\right]\mathrm{d}z$$

$$R_i = \int_D \left[\frac{\partial x_2}{\partial t}\ \phi^i - Pe_2^{-1}\frac{\partial^2 x_2}{\partial z^2}\ \phi^i + \frac{\partial x_2}{\partial z}\ \phi^i + \beta x_2\phi^i - BDa(1-x_1)\exp\left[\frac{x_2}{1+x_2/\gamma}\right]\phi^i\right]\mathrm{d}z \tag{18}$$

where $D \equiv [0,1]$, $i = 1, \ldots, N$.

The unknowns are approximated by Eq. (6) given in Section 2 and substituted into Eq. (18). The residuals are then set to zero, which yields the following system in compact form:

$$\mathbf{B}\mathbf{x}_1 = \mathbf{F}_1(\mathbf{x}_1,\ \mathbf{x}_2)$$

$$\mathbf{B}\mathbf{x}_2 = \mathbf{F}_2(\mathbf{x}_1,\ \mathbf{x}_2) \tag{19}$$

where, $\mathbf{x}_1 = [x_{11}(t),\ x_{12}(t),\ \ldots,\ x_{1N}(t)]$ and $\mathbf{x}_2 = [x_{21}(t),\ x_{22}(t),\ \ldots,\ x_{2N}(t)]$ are the vectors of the nodal values of the unknowns $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.

The derivatives $(\partial x_1/\partial z)|_{z=0}$ and $(\partial x_2/\partial z)|_{z=0}$ are treated as additional unknowns. They are included, accordingly, in Eq. (19).

Let **X** be a $2N+2$ vector containing the nodal values of the unknowns and the two extra unknowns that correspond to the aforementioned derivatives, i.e. $\mathbf{X} = [\mathbf{x}_1,\ \mathbf{x}_2,\ (\partial x_1/\partial z)|_{z=0},\ (\partial x_2/\partial z)|_{z=0}]$. The equations that close the system are the boundary conditions at the inlet:

$$\mathrm{X}(2N+1) - Pe_1\mathrm{X}(1) = 0$$

$$\mathrm{X}(2N+2) - Pe_2\mathrm{X}(N+1) = 0 \tag{20}$$

The difference between the 'DAE implementation'

and the 'ODE implementation' lies in the different treatment of the boundary conditions; in the ODE formulation, the first derivatives $(\partial x_1/\partial z)|_{z=0}$ and $(\partial x_2/\partial z)|_{z=0}$ are replaced in the residual equations by their values specified by the boundary conditions following integration by parts of the second-order spatial derivative term. Thus, the system consists of $2N$ differential equations. In contrast, in the DAE formulation, the values of the derivatives are treated as two additional unknowns and are not replaced; instead the two boundary conditions augment the system of equations to be solved. The results of the 'DAE implementation' of RPM will be demonstrated below, in the Section 5.2.

5. Results and discussion

5.1. The interplay of the RPM with the subspace of the timestepper: the role of Δt

The performance of the RPM is inextricably linked to the spectrum of the chosen time integration scheme for the particular spatial discretization, because it detects the dominant part of the spectrum of $\mathbf{F}$, the result of time integration of the spatially discretized PDEs, given in Eq. (9), and not of the *original* steady state problem itself. The effect of the numerical stability of the timestepper is reflected in the dimension of the RPM-detected slow subspace. The more unstable the numerical integration scheme, the larger the number of modes that need to be approximated by the procedure in order to induce convergence.

A brief analysis of the dependence of the eigenvalues of the timestepper on the time step follows. Let us consider the parameter dependent nonlinear dynamical system:

$$\mathbf{u} = \mathbf{f}(\mathbf{u},\ p) \tag{21}$$

Here $\mathbf{u}$ is the vector that contains the state variables; the equivalent linearized system around an equilibrium point, $\mathbf{u}_0$ is:

$$\dot{\xi} = \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,\ p)\xi, \quad \text{where} \quad \xi \equiv \mathbf{u} - \mathbf{u}_0 \tag{22}$$

In our case $\mathbf{u} = [x_1,\ x_2]$ and the operator $\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,\ p)$ is a two-by-two matrix operator. The solution of this system is:

$$\xi = \mathrm{e}^{\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,p)t}\xi_0 \tag{23}$$

The eigenvalues of $\mathrm{e}^{\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,p)t}$ are $\mu_i = \mathrm{e}^{\lambda_i t}$, where λ_i are the eigenvalues of $\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,\ p)$. Therefore, when the real part of λ, λ_{Re}, is negative then $|\mu| \leq 1$ and when $\lambda_{\mathrm{Re}} \geq 0$ then $|\mu| \geq 1$ for every value of $t > 0$.

Nevertheless, in practice time integration is performed with a numerical scheme such as the forward Euler, which yields:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\mathbf{f}(\mathbf{u}^n,\ p) \tag{24}$$

or, in discretized form:

$$\mathbf{U}^{n+1} = \mathbf{F}(\mathbf{U}^n,\ p) \tag{25}$$

The resulting map can be chosen to be the result of integration over time T using several smaller forward Euler steps, or it can be the result of a single time step Δt. The latter case is dealt with here, without precluding the possibility of working with the former one. If λ_i are the eigenvalues of the Jacobian matrix which results from $\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,\ p)$ after spatial discretization, the eigenvalues, κ_i, of the matrix $\mathbf{F}_{\mathbf{U}}$ for an explicit forward Euler integrator with step Δt are:

$$\kappa_i = 1 + \Delta t\lambda_i \tag{26}$$

These eigenvalues are hereafter called multipliers. The scheme is stable when all the multipliers, κ_i, are inside the unit circle, whereas a turning point instability occurs when one real multiplier crosses the unit circle through 1. A Hopf point appears when a pair of conjugate complex multipliers have modulus equal to 1. It must be noted that, for certain values of the time step, Δt, the scheme may be numerically unstable even though $\lambda_{\mathrm{Re}} < 0$ as discussed later. These numerical instabilities depend strongly on the size of the time step, in contrast to a true turning point of the discretized physical problem which is independent of Δt, because when $\lambda = 0$ then $\kappa = 1$ for every Δt.

As far as the Hopf bifurcation is concerned, it is easy to demonstrate that a dependence on the time step in fact exists. Once more, κ denotes the multipliers of the forward Euler scheme and λ the eigenvalues of the Jacobian matrix which results from $\nabla_{\mathbf{u}}\mathbf{f}(\mathbf{u}_0,\ p)$ after spatial discretization, and the subscripts Re and Im denote the real and imaginary component, respectively. It holds that:

$$\kappa_{\mathrm{Re}} \pm \kappa_{\mathrm{Im}}i = 1 + \Delta t(\lambda_{\mathrm{Re}} \pm \lambda_{\mathrm{Im}}i) = (1 + \Delta t\lambda_{\mathrm{Re}}) \pm \Delta t\lambda_{\mathrm{Im}}i$$

At a Hopf bifurcation of the timestepper:

$$\sqrt{\kappa_{\mathrm{Re}}^2 + \kappa_{\mathrm{Im}}^2} = \sqrt{(1 + \Delta t\lambda_{\mathrm{Re}})^2 + (\Delta t\lambda_{\mathrm{Im}})^2} = 1$$
$$\Rightarrow \Delta t\lambda_{\mathrm{Re}}^2 + 2\lambda_{\mathrm{Re}} + \Delta t\lambda_{\mathrm{Im}}^2 = 0 \Rightarrow \lambda_{\mathrm{Re}} = \frac{-1 \pm \sqrt{1 - \Delta t^2\lambda_{\mathrm{Im}}^2}}{\Delta t}$$

Therefore, $\lambda_{\mathrm{Re}} \neq 0$, when $|\kappa| = 1$ which means that a Hopf bifurcation resulting from the timestepper does not, in general, coincide with a Hopf bifurcation for the discretized problem. However, if the time step is very small it follows that there are two values for λ_{Re}: $\lambda_{\mathrm{Re}} \approx 0$ and $\lambda_{\mathrm{Re}} \approx -2/\Delta t$. The former approximates a 'true' Hopf bifurcation, whereas the latter is clearly a spurious, numerical one.

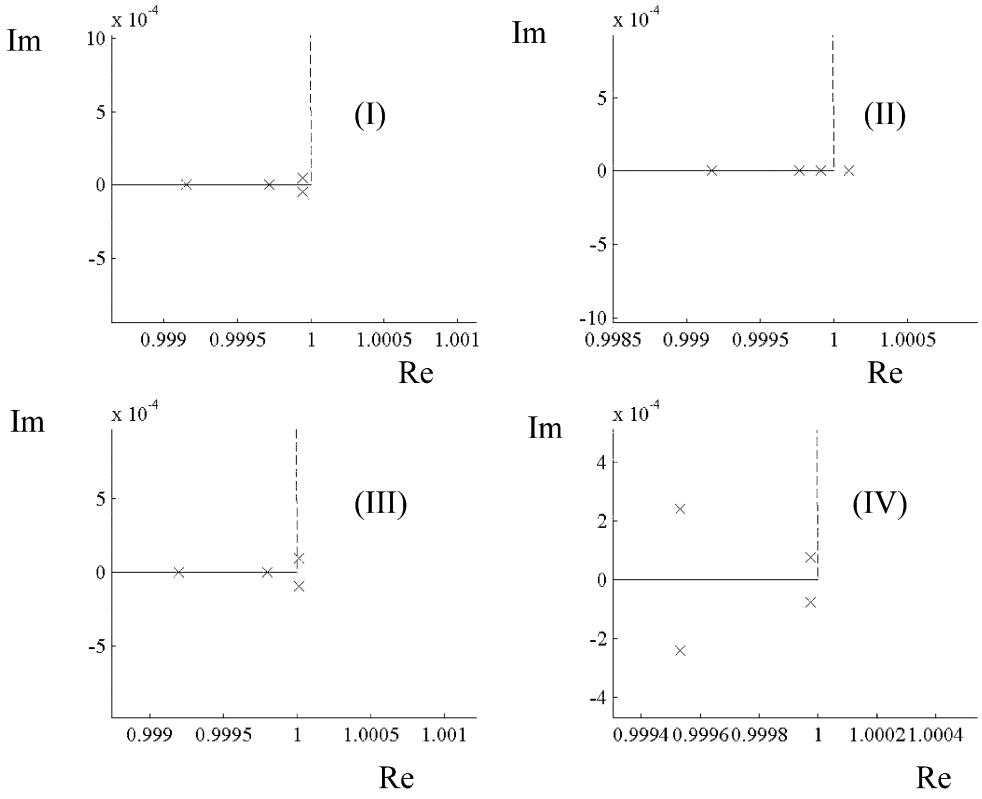

Fig. 4. RPM approximations of the dominant eigenvalues of the timestepper, at different regions of the solution branch. Regions (I), (II), (III) and (IV) are shown in Fig. 1.

Returning to the discretized tubular reactor model given compactly in Eq. (9), the implementation of the RPM produces the matrix $\mathbf{H} = \mathbf{Z}^{\mathrm{T}}\mathbf{F}_{\mathbf{U}}\mathbf{Z}$. The dominant multipliers of $\mathbf{H}$ in all four sections of the solution branch are shown in Fig. 4. Along the stable branch (I), all multipliers lie within the unit disk, whereas past the first turning point, along the unstable branch (II), one real multiplier becomes larger than 1. A second turning point appears in (III) meaning that a second real multiplier becomes larger that 1. Subsequently the pair of real unstable multipliers becomes a pair of unstable conjugate complex multipliers. A Hopf bifurcation occurs when the modulus of the complex multipliers becomes equal to 1 and the system is stabilized going into region (IV). It is worth noting that the second turning point and the Hopf bifurcation are very close and so region (III) is very small.

By increasing the time step, the spectrum of the timestepper changes, resulting in certain multipliers approaching and eventually crossing the unit circle from -1. When this happens, numerical instabilities appear that are not related to the stability of the spatially discretized physical problem. Typical spectra for three different time steps, for fixed $Da = 0.09$, are shown in Fig. 5. The first case corresponds to $\Delta t = 0.0003$, for which the RPM computed a slow subspace of dimension $m = 8$ and the integrator converged to steady state but none of these multipliers is outside the unit disk. For $\Delta t = 0.0005$ the timestepper becomes unstable since a number of multipliers become smaller than -1. The RPM approximates the subspace that corresponds to these multipliers and stabilizes the procedure. The inclusion of extra directions results in a significantly larger subspace, $m = 19$, making the procedure less computationally efficient. Among the 19 multipliers, six lie outside the unit disk. The last case corresponds to $\Delta t = 0.001$ for which the RPM-assisted algorithm did not converge.

The distinction between the critical multipliers of the timestepper and the critical multipliers of the underlying spatially discretized physical problem can sometimes be rationalized through plotting the reconstructed eigenvectors in space.

Shown in Fig. 6, are the eigenvectors $\mathbf{y}$, of the generalized eigenvalue problem $\mathbf{R}_{\mathbf{U}}\mathbf{y} = \omega\mathbf{M}\mathbf{y}$ discussed in Section 3, as they were reconstructed from the RPM-derived eigenvectors. Those that are due to numerical instabilities appear to have irregular, mesh-level noise-like appearance because they correspond to the left-most part of the spectrum of the matrix $\mathbf{M}^{-1}\mathbf{R}_{\mathbf{U}}$ which is far from the imaginary axis (high wavenumbers in space). In contrast, the ones that correspond to the dominant

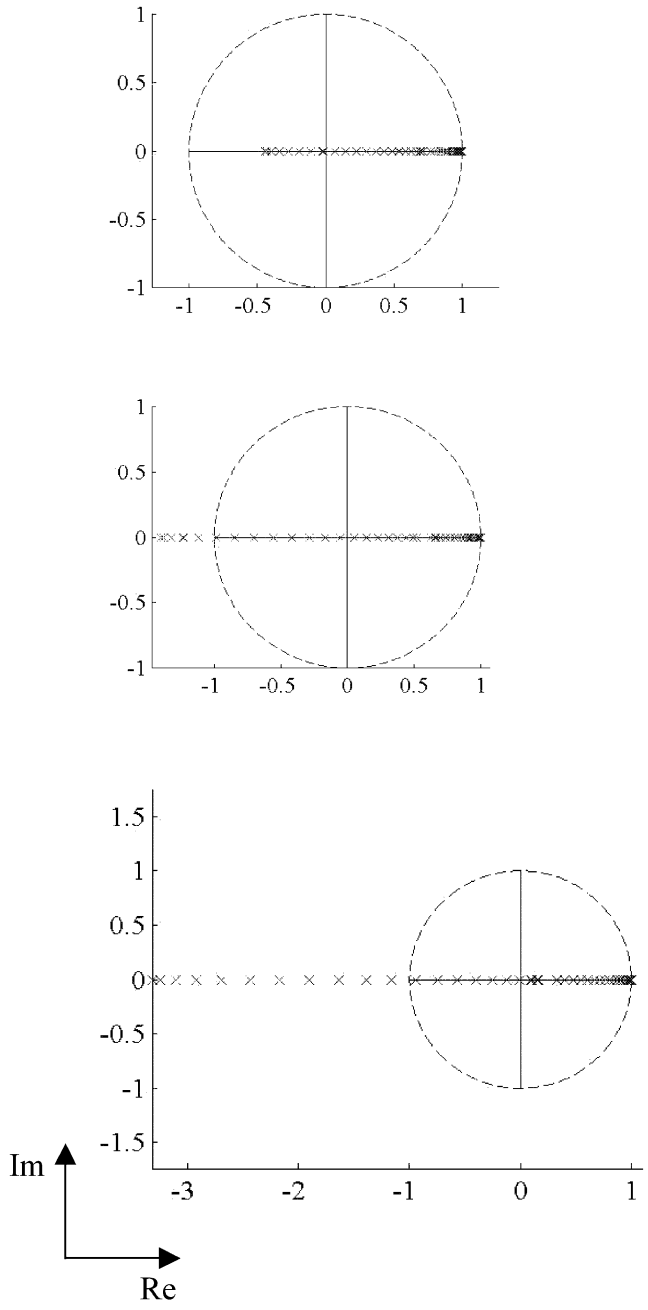

Fig. 5. Typical spectra of the timestepper at $Da = 0.09$ for (i) $\Delta t = 0.0003$, (ii) $\Delta t = 0.0005$, (iii) $\Delta t = 0.001$.

eigenspectrum of $\mathbf{M}^{-1}\mathbf{R}_\mathbf{U}$ have smooth profiles: they correspond to physical instabilities and are converged with respect to the discretization.

The first column of Table 1 shows typical values of the eigenvalues of the aforementioned generalized eigenvalue problem for a certain spatial discretization. The subsequent columns show the corresponding eigenvalues of $\mathbf{F}_\mathbf{U}$ for two values of Δt, one where there are no numerical instabilities and one for which numerical instabilities are present. We note that for the particular, coarse but illustrative discretization, the negative eigenvalues shown in the first column are the ones with the largest absolute value. For $\Delta t = 0.0005$, the eigenvalues of $\mathbf{F}_\mathbf{U}$ exit the unit disk from -1 signaling, therefore, the presence of numerical instabilities. For smaller Δt, the corresponding eigenvalues are well within the unit disk causing no numerical instabilities.

As a consequence of the above, stability analysis of the original PDEs, is performed by studying the portion of the spectrum of the RPM-derived matrix $\mathbf{H} = \mathbf{Z}^\mathrm{T}\mathbf{F}_\mathbf{U}\mathbf{Z}$ that corresponds (can be attributed to) the dominant portion of the spectrum of $\mathbf{M}^{-1}\mathbf{R}_\mathbf{U}$. Fig. 7 shows the largest multipliers along each segment of the solution branch. The behavior is the same as in the case of the timestepper without numerical instabilities.

5.2. Implementation of the RPM on a system of semi-explicit index 1 DAEs

Implicit systems of differential/algebraic equations compactly given in Eq. (8) are solved by the code DASSL, replacing the time derivative by a kth order backward differentiation formula (BDF), where k ranges from 1 to 5. For the first order formula, the resulting equation is:

$$\mathbf{F}\left(t_n,\ \mathbf{U}_n, \frac{\mathbf{U}_n - \mathbf{U}_{n-1}}{\Delta t_n}\right) = \mathbf{0} \tag{27}$$

It is solved using Newton's method:

$$\mathbf{U}_n^{m+1} = \mathbf{U}_n^m - \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}'} + \frac{1}{\Delta t}\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^{-1} \times \mathbf{F}\left(t_n,\ \mathbf{U}_n^m, \frac{\mathbf{U}_n^m - \mathbf{U}_{n-1}}{\Delta t_n}\right) \tag{28}$$

where m is the iteration counter. The method converges rapidly given a good enough initial guess, and therefore, at the current time t_n, DASSL uses a polynomial which interpolates the computed solution at the last $k+1$ times. The iteration is considered converged when:

$$\frac{\rho}{1-\rho}\|\mathbf{U}^{m+1} - \mathbf{U}^m\| < 0.3, \qquad \rho = \left(\frac{\|\mathbf{U}^{m+1} - \mathbf{U}^m\|}{\|\mathbf{U}^1 - \mathbf{U}^0\|}\right)^{1/m} \tag{29}$$

Here the norms are scaled norms that depend on the error tolerances specified by the user. If $\rho > 0.9$, or $m > 4$, and the iteration has not yet converged, then the stepsize is reduced and the step is attempted again. The linear systems are solved using routines from LAPACK.

The steady states of the differential/algebraic equations described in Section 4 were computed by RPM-assisted timestepping. The RPM procedure is implemented on the differential variables only. The timestepping is performed by DASSL, with a total reporting

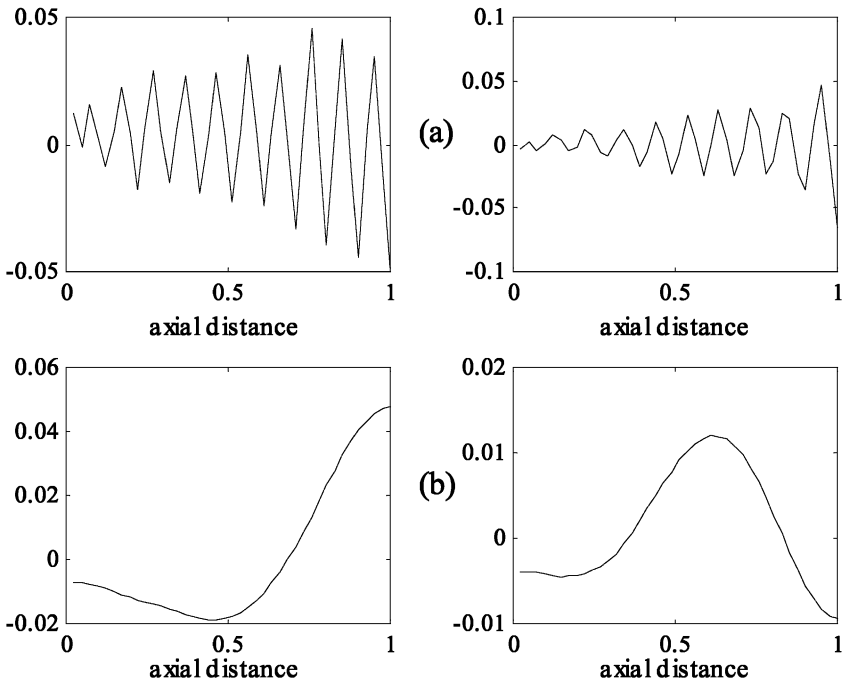

Fig. 6. RPM eigenvector approximations of the timestepper that correspond (a) to numerical instabilities, (b) to physical instabilitites. Plotted as explained in the caption of Fig. 3.

Table 1
Eigenvalues of the generalized eigenproblem $\mathbf{R}_{\mathbf{U}}\mathbf{y} = \omega\mathbf{M}\mathbf{y}$ and the corresponding eigenvalues of the timestepper for two time steps: $\Delta t = 0.0001$, $\Delta t = 0.0005$

Generalized eigenvalue	Time step	
	0.0001	0.0005
−4804.142	0.5196	−1.4021
−4804.709	0.5195	−1.4024
−4769.994	0.5230	−1.3850
−4246.662	0.5753	−1.1233
−3981.132	0.6019	−0.9906
−3981.698	0.6018	−0.9908
−3696.557	0.6303	−0.8483

horizon for the integration (involving several steps) $\Delta t = 0.1$. To integrate over this time interval the code performs approximately ten BDF steps. Whenever $\mathbf{U}$ is not the result of integration, the constraint $\mathbf{F}(t, \mathbf{U}, \mathbf{U}') = \mathbf{0}$ is satisfied by integrating the system by a very small time step, $\Delta t_s \ll \Delta t$. The DASSL software internally performs this small time step which is chosen to be $\Delta t_s = \Delta t/1000$.

Again, the arc-length continuation procedure is implemented and the solution branch is successfully computed. The maximum dimension of the dominant subspace along the computed solution branch is $m = 2$. The corresponding eigenvectors at two parameter values, one close to the first turning point ($Da = 0.121816$), where both multipliers are within the unit disk, and one at the Hopf bifurcation ($Da = 0.110036$), where the modulus of the two conjugate complex multipliers is equal to 1, are reconstructed by the RPM subspace. They are compared with the corresponding eigenvectors computed by the RPM for the ODE formulation that results from the spatial discretization of Eq. (4). The reported eigenvectors from the DAE system result from the RPM implementation on the $2N$ differential variables. Initial values for the two extra variables are prescribed using Eq. (16). The reporting horizon in the case of the ODEs is also $\Delta t = 0.1$ and the maximum dimension of the subspace detected by the RPM is $m = 4$. In both the ODE and the DAE case, the multipliers of the small matrix $\mathbf{H}$ are the same for the same parameter value, namely, $\mu_1 = 0.976142$ and $\mu_2 = 0.757455$ in the first case and $\mu = 1.0041319 \pm 0.162419i$ in the second case. The corresponding eigenvectors are also in agreement between them as shown in Fig. 8. It is important here to note that, in solving the complete differential algebraic problem, one would need to solve a singular generalized eigenvalue problem in order to find the stability of the solution converged to. RPM provides, as a byproduct, a direct small eigenproblem (neither generalized nor singular) whose solution gives a good approximation of the stability properties of the desired steady state.

6. Conclusions

The performance of the RPM is studied in relation to the stability of the time integration scheme and of the

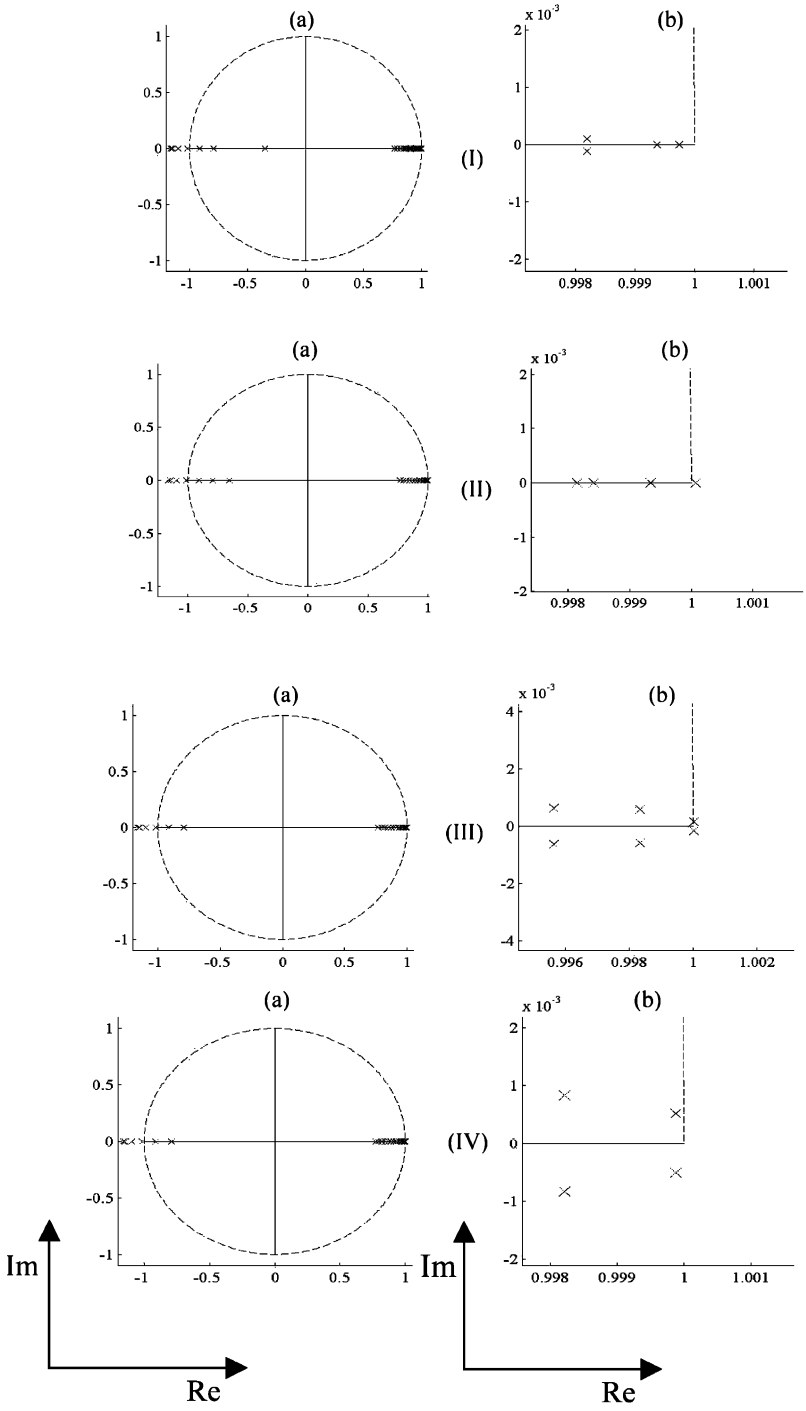

Fig. 7. (a) Dominant eigenspectrum of the timestepper approximated by the RPM (b) the part of the spectrum that corresponds to the dominant eigenspectrum of the steady state Jacobian, $\mathbf{R}_U$.

discretized physical problem on which it is implemented. The effect of the time step of the integrator and the interplay with the RPM-assisted convergence is examined with emphasis on the distinction between the critical eigenvalues of the timestepper and those of the discretized PDEs.

We demonstrate that it may still be possible to perform stability analysis of the discretized physical problem from the RPM subspace, even when it contains eigendirections corresponding to spurious instabilities of the scheme as well as of the true, physical instabilities of the original PDEs.

In addition the RPM is implemented on a system of semi-explicit, index 1 differential/algebraic equations, solved by the code DASSL, without explicitly performing any reduction of the original equations to ODEs. This is

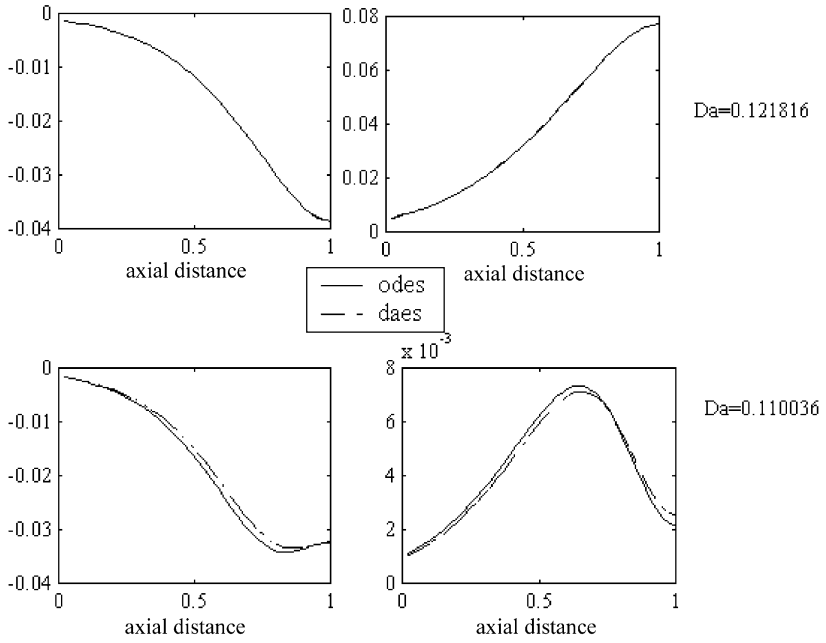

Fig. 8. RPM generated eigenvectors for $Da = 0.121816$ and $Da = 0.110036$ — ODEs, -- DAEs. For the DAE case, the RPM subspace is defined by using the differential variables only.

relevant since DAEs are often used to describe physical systems of chemical engineering interest and there are several commercial codes that are used to that purpose. A case in point is the process design code gPROMS (Process Systems Enterprise Ltd., 2002a,b); it has an open software architecture that allows the embedding of other software within gPROMS, or the embedding of gPROMS within other software. It is exactly the aspect of openness of a commercial code that offers the possibility of using the RPM as a computational 'shell' around the code in order to extend its capabilities in the field of nonlinear stability analysis, in the context presented in the current work with regard to the implementation of the RPM on a system of index 1 DAEs.

Acknowledgements

This work was partially supported by the General Secretariat for Research and Technology of Greece through the ΠΕΝΕΔ program. It was also supported by the National Science Foundation, the Air Force Office of Scientific Research/USA and UTRC (IGK). The authors would like to thank Dr K. Lust, Dr C. Theodoropoulos and Dr C. Siettos for valuable discussions, and also Professor C. Pantelides and Professor L. Petzold for many useful comments on the application of the RPM to systems of DAEs.

References

Barton, P. I., & Pantelides, C. C. (1994). Modeling of combined discrete/continuous processes. *American Institute of Chemical Engineers Journal 40*, 966.

Berzins M. & Furzeland R. M. (1985). *A user's manual for SPRINT—a versatile software package for solving systems of algebraic ordinary and partial differential equations: part 1—algebraic and ordinary differential equations*. Report TNER.85.058, Shell Research Ltd, Thornton Research Centre, Chester.

Boffi, D., Duran, R. G., & Gastaldi, L. (1999). A remark on spurious eigenvalues in a square. *Applied Mathematics Letters 12*, 107.

Bolstad, J. H., & Keller, H. B. (1986). A multigrid continuation method for elliptic problems with folds. *SIAM Journal on Science and Statistical Computing 7*, 1081.

Brenan, K. E., Campbell, S. L. & Petzold, L. R. (1995). *Numerical solution of initial-value problems in differential–algebraic equations*. 2nd edition, SIAM, Philadelphia.

Brown, P. N., Hindmarsh, A. C., & Petzold, L. R. (1998). Consistent initial condition calculation for differential–algebraic systems. *SIAM Journal on Science and Computing 19*, 1495.

Burrage, K., Erhel, J., Pohl, B., & Williams, A. (1998). A deflation technique for linear systems of equations. *SIAM Journal on Science and Computing 19*, 1245.

Campbell, S. L. (1986). Consistent initial conditions for linear time varying singular systems. In *Frequency domain and state space methods for linear systems*. Amsterdam: Elsevier.

Gear, W., Kevrekidis, I. G., & Theodoropoulos, C. (2002). 'Coarse' integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods. *Computers and Chemical Engineering 26*, 941.

Jarausch, H., & Mackens, W. (1984). Numerical treatment of bifurcation problems by adaptive condensation. In T. Küpper, H. D. Mittelmann & H. Weber (Eds.), *Numerical methods for bifurcation problems*. Basel: Birkhäuser.

Jarausch, H., & Mackens, W. (1987a). Computing bifurcation diagrams for large nonlinear variational problems. In P. Deuflhard & B. Engquist (Eds.), *Large scale scientific computing. Progress in scientific computing*, vol. 7. Boston: Birkhäuser.

Jarausch, H., & Mackens, W. (1987b). Solving large nonlinear systems of equations by an adaptive condensation process. *Numerische Mathematik 50*, 633.

Jensen, K. F., & Ray, W. H. (1982). The bifurcation behavior of tubular reactors. *Chemical Engineering Science 37*, 199.

Keller, H. B., & von Sosen, H. (1995). New methods in CFD: DAE and RPM. *Proceedings of the first Asian conference on computational fluid dynamics (CFD)*. Hong Kong.

Kröner, A., Marquardt, W., & Gilles, E. D. (1992). Computing consistent initial conditions for differential–algebraic equations. *Computers and Chemical Engineering 16*, 131.

Leimkuhler, B., Petzold, L. R., & Gear, C. W. (1991). Approximation methods for the consistent initialization of differential–algebraic equations. *SIAM Journal on Numerical Analysis 28*, 205.

Love, P. (1999). *Bifurcations in Kolmogorov and Taylor-vortex flows*. Ph.D. thesis, Caltech, Pasadena, CA.

Lust, K. (1997). *Numerical bifurcation analysis of periodic solutions of partial differential equations*. Ph.D. thesis, KU Leuven.

Lust, K., Roose, D., Spence, A., & Champneys, A. R. (1998). An adaptive Newton–Picard algorithm with subspace iteration for computing periodic solutions. *SIAM Journal on Science and Computing 19*, 1188.

Pantelides, C. C. (1988). The consistent initialization of differential–algebraic systems. *SIAM Journal on Science and Statistical Computing 9*, 213.

Petzold, L. R. (1982). *A description of DASSL: a differential/algebraic system solver*. SAND82-8637, Livermore, CA: Sandia National Laboratories.

Process Systems Enterprise Ltd. (2002a). *gPROMS v2.1 introductory user guide and gPROMS v2.1 advanced user guide*, London, UK.

Process Systems Enterprise Ltd. (2002b). *gPROMS v2.1 system programmer guide and the gPROMS v2.1 server*. London, UK.

Saad, Y. (1996). *Iterative methods for sparse linear systems*. PWS Publishing Company.

Shroff, G. M., & Keller, H. B. (1993). Stabilization of unstable procedures: the recursive projection method. *SIAM Journal on Numerical Analysis 31*, 1099.

Subramanian, S., & Balakotaiah, V. (1996). Classification of steady-state and dynamic behavior of distributed reactor models. *Chemical Engineering Science 51*, 401.

Sureshkumar, R. (2001). Local linear stability characteristics of viscoelastic periodic channel flow. *Journal of Non-Newtonian Fluid Mechanics 97*, 125.

Theodoropoulos C., Bozinis N., Siettos C. I., Pantelides C. C. & Kevrekidis I. G. (2001). *A stability/bifurcation framework for process design*, presented at the Annual Meeting of the American Institute of Chemical Engineers (AIChE), Reno, NV.

Theodoropoulos, C., Qian, Y.-H., & Kevrekidis, I. G. (2000). 'Coarse' stability and bifurcation analysis using time-steppers: a reaction–diffusion example. *Proceedings of the National Academy of Sciences 97*, 9840.

Tuckerman, L. S., & Barkley, D. (2000). Bifurcation analysis for timesteppers. In E. Doedel & L. S. Tuckerman (Eds.), *Numerical methods for bifurcation problems and large-scale dynamical systems*. New York: Springer.

von Sosen, H. (1994). *I. Folds and bifurcations in the solutions of semi-explicit differential-algebraic equations. II. The recursive projection method applied to differential–algebraic equations and incompressible fluid mechanics*. Ph.D. thesis, Caltech, Pasadena, CA.

Yeih, W., Chen, J. T., & Chang, C. M. (1999). Applications of dual MRM for determining the natural frequencies and natural modes of an Euler–Bernoulli beam using the singular value decomposition method. *Engineering Analysis with Boundary Elements 23*, 339.