Journal of Computational and Applied Mathematics 259 (2014) 259–280

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Parametric dominant pole algorithm for parametric model order reduction

Maryam Saadvandi [a,*], Karl Meerbergen [a], Wim Desmet [b]

[a] Celestijnenlaan 200a - bus 2402, 3001 Leuven, Belgium

[b] Celestijnenlaan 300a - bus 2402, 3001 Leuven, Belgium

ARTICLE INFO

Article history:
Received 8 March 2013
Received in revised form 29 August 2013

Keywords:
Parametric system
Parametric transfer function
Modal approximation
Parametric eigenvalue problem
Parametric model reduction
Dominant poles algorithm

ABSTRACT

Standard model order reduction techniques attempt to build reduced order models of large scale systems with similar input–output behavior over a wide range of input frequencies as the full system models. The method known as the dominant pole algorithm has previously been successfully used in combination with model order reduction techniques to approximate standard linear time-invariant dynamical systems and second order dynamical systems as well as nonlinear time-delay systems. In this paper, we show that the dominant pole algorithm can be adapted for parametric systems where these parameters usually have physical meaning. There are two approaches for finding dominant poles. These algorithms are illustrated by the second order numerical examples.

1. Introduction

Consider the single-input–single-output system, formulated in the Laplace or frequency domain, as follows:

$$\begin{cases} A(s)x(s) = bu(s) \\ y(s) = c^* x(s) \end{cases} \tag{1}$$

where $A(s)$ is an $n \times n$ matrix with large n. Function $u(s)$ is the input and $y(s)$ the output. Vector $x(s)$ is called the state vector. Matrix A is a function of s and is linear, e.g., $A(s) = sE - A_0$, for linear systems of ODEs, or quadratic, e.g., $A(s) = K + s^2M$ or $A(s) = K + sC + s^2M$, for the analysis of vibrations or even nonlinear as, e.g., $A(s) = s^2M(s) + sC(s) + K(s)$ or the delay differential equation, where $A(s) = sE - A_0 + e^{-\tau s}A_1$ and τ is the delay.

In applications arising from PDEs and large systems of ODEs, the solution of the original system is expensive although matrices are sparse. The reduced model is of small dimension but dense. There are numerous examples that show significant speed up by model reduction [1]. There are situations where the computation and evaluation of the reduced model do not lead to a speedup; these applications are not considered in this paper.

The model should be built so that the output of the reduced model and the original one are close. There are basically three classes of MOR methods. Moment matching or Padé type methods are very popular in circuit design and vibrations [2–4]. They match high order moments around a central point or a selection of points. Krylov methods and rational Krylov methods belong to this class of methods. Balanced truncation methods are usually more expensive, but build smaller models for the same error level as Krylov–Padé methods. There are various algorithms for large scale problems, based on rational Krylov spaces [2,5–7].

* Corresponding author. Tel.: +32 16 32 77 35.
E-mail addresses: maryam.saadvandi@cs.kuleuven.be (M. Saadvandi), karl.meerbergen@cs.kuleuven.be (K. Meerbergen), wim.desmet@mech.kuleuven.be (W. Desmet).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.09.012

Finally, modal approximation is very popular in the study of vibrations, but also for the reduction of models of power lines [8–10]. The dominant pole algorithm method approximates the solution as a sum of linear rational functions of which the poles are eigenvalues of the system. The dominant pole algorithm with subspace projection [9,11] is related to the Jacobi–Davidson method for computing eigenvalue problems. We discuss it in more detail in Section 2. There are variants for polynomial [12] and nonlinear problems [13].

In our case, y does not only depend on s but also on parameters, which we denote by $\gamma = (\gamma_1, \ldots, \gamma_p) \in \Gamma \subset \mathbb{R}^p$, that arise in A, b or c. These parameters usually have a direct physical meaning. Such parametric studies appear in uncertainty quantification [14,15] or design optimization [1]. These applications solve (1) for hundreds or thousands of different combinations of s and γ. It is this cost that should be reduced dramatically. The classical approach is to build a reduced model for each value of $\gamma \in \Gamma$ for which the system is evaluated. This is probably highly inefficient since these systems are built independently. Parametric model order reduction, on the other hand, allows for cheaply evaluating $y(s, \gamma)$ for a wide range of values of γ. Padé type methods such as PIMTAP [16–18] and TAP [16] build reduced models whose output matches multivariate moments with the exact output function. Interpolatory reduced models [19] are built by interpolating subspaces generated at a selection of interpolation points in the parameter space. The reduced model is obtained by merging these subspaces and using them for evaluating the system output in other than the interpolation points [20].

In this paper, we will iteratively compute the k parametric dominant poles. We consider two approaches. In the first approach, we compute the parameter dependent poles one by one, i.e., all parameters are taken into account together. We will use interpolation in the parameter space to achieve this. In the second approach, the dominant eigenpairs are computed for a selection of interpolation points in the parameter space, independently from each other. As for the iterative rational Krylov algorithm [20] and Krylov–Padé methods [1], the transfer function is Hermite interpolated in the iteration points in the Laplace domain and the parameter domain. Related work can be found in [21]. The dominant pole algorithm was modified for finding the most sensitive pole of a parametric system. In this paper, we look at quite a different angle in that we compute dominant poles for the entire parameter space, which are not necessarily the most sensitive poles. We indeed noticed that for the models in the numerical examples, the sensitivity increases with the frequency, where usually, the most dominant poles are the poles with lowest frequency.

The paper is organized as follows. In Section 2 the dominant pole algorithm for nonparametric problems is reviewed. Section 3 introduces the dominant pole algorithm adapted for the parametric case. Numerical examples are shown in Section 4 and the conclusions are given in Section 5.

2. The dominant pole algorithm

The transfer function of system (1), $H : \mathbb{C} \to \mathbb{C}$, is defined as

$$H(s) = c^* A(s)^{-1} b.$$

The poles of the system (1) are the poles of $H(s)$. These form a subset of the eigenvalues $\lambda \in \mathbb{C}$. Define an eigentriplet (λ_j, x_j, y_j) of $A(s)$ as:

$$\begin{cases} A(\lambda_j)x_j = 0 & x_j \neq 0 \\ A(\lambda_j)^* y_j = 0 & y_j \neq 0 \end{cases} \tag{2}$$

where λ_j is an eigenvalue and $x_j, y_j \in \mathbb{C}$ are corresponding right and left eigenvectors. Note that the number of eigentriplets depends on the type of problem: when $A(s) = K + s^2 M$ with M symmetric positive definite and K symmetric, then there are n linearly independent right eigenvectors with associated λ_j^2 value. When $A(s) = K + sC + s^2 M$, then there are $2n$ eigenvalues λ_j and if $A(s)$ originates from the delay differential equation then the system has an infinite number of eigenvalues.

The Dominant Pole Algorithm (DPA) computes 'dominant poles' of the system. We now explain this concept and how a reduced model is built from those. We assume the transfer function $H(s)$ can be expressed as

$$H(s) = \sum_j \frac{R_j}{s - \lambda_j},$$

where the sum is taken over all eigenvalues and where

$$R_j = \frac{\left(c^* x_j\right)\left(y_j^* b\right)}{y_j^* \frac{\mathrm{d}A(\lambda_j)}{\mathrm{d}\lambda_j} x_j}$$

is called the residue [22,23]. The weighted residue is defined as

$$\rho_j = \frac{|R_j|}{|\mathrm{Re}(\lambda_j)|}.$$

The poles are sorted following decreasing weighted residue, i.e.,

$$\rho_1 \geq \rho_2 \geq \rho_3 \geq \cdots \geq \rho_k.$$