Max Planck Institute Magdeburg
Preprints

Peter Benner Sara Grundel Nils Hornung

Parametric Model Order Reduction with a Small $\mathscr{H}_2$-Error Using Radial Basis Functions

MPIMD/14-01 **January 16, 2014**

Abstract

Given the optimal interpolation points $\sigma_1, \ldots, \sigma_r$, it is well-known how to obtain the $\mathscr{H}_2$-optimal reduced order model of order r for a linear time-invariant system of order $n \gg r$. Our approach to linear time-invariant systems depending on parameters p is to approximate their parametric dependence as a so-called metamodel, which in turn allows us to set up the corresponding parametrized reduced order models. The construction of the metamodel we suggest involves the coefficients of the characteristic polynomial together with k-means clustering and radial basis function interpolation, and thus allows for an accurate and efficient approximation of $\sigma_1(p), \ldots, \sigma_r(p)$. As the computation still includes large system solves, this metamodel is not sufficient to construct a fast and truely parametric reduced system. Setting up a medium size model without extra cost, we present a possible answer to this. We illustrate the proposed method with several numerical examples.

Impressum:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for
Dynamics of Complex Technical Systems

Address:
Max Planck Institute for
Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

http://www.mpi-magdeburg.mpg.de/preprints/

1 Introduction

We consider large-scale parameter-dependent single-input, single-output systems that are linear and time-invariant. For each value of the parameter we have a large system of ordinary differential equations with inputs and outputs. Model order reduction can reduce the computational complexity of such linear time-invariant systems at the expense of some error in the approximated response. In order to keep this error small, $\mathscr{H}_2$-optimal model order reduction methods have been proposed in the literature, such as the Iterative Rational Krylov Algorithm (IRKA). IRKA, as an iterative algorithm that does not require the solution of the original large ordinary differential equation, is a practical and fast algorithm. It computes a reduced-order locally $\mathscr{H}_2$-optimal system, iterating the poles, or eigenvalues, of the reduced system.

Assume now, our goal is to conduct an extensive parameter study or solve an optimization problem where the system has to be evaluated for many realizations of our parameter. To compute the reduced system anew by IRKA for each parameter value can produce prohibitive costs, even though the solution itself is then accelerated by the obtained reduction. There are several methods in the literature that deal with this issue, of which many are described by Benner et al. (2013). These methods include e. g. matrix interpolation and interpolation of the transfer function, see (Baur et al., 2011a). We address the problem by pre-computing few "snapshots", i. e. we evaluate a discrete sampling of parameter space, in order to approximate $\mathscr{H}_2$-optimal model order reduction during an "online phase". Our hope is to find *almost* $\mathscr{H}_2$-*optimal* reduced-order systems by an approximation of the reduced order system poles. The poles are an important characteristic of the reduced order model. When they are known, we can create the reduced model via interpolation.

Reflected in a resulting $\mathscr{H}_2$-error of even lower order, an error in the mirror poles can be observed to have moderate consequences. This is the main observation on which we base our approach. Hence, we strive to find a good approximation for these *mirror poles*, which, as to our knowledge, primarily distinguishes our proposition from previous works on the topic. Especially, the details seem to be novel: our approach takes characteristic traits and thus possible discontinuities of the mirror poles into account in the formulation of the interpolant and reuses the information obtained when snapshots are computed to speed up all further calculations. In fact, we approximate the mirror poles based on standard radial basis function interpolation of the corresponding characteristic polynomial, thus allowing for an accurate representation of eigenvalue bifurcations. Furthermore, clustering is employed to treat remaining discontinuities and a medium-size model incorporates results from the offline phase, otherwise wasted, to speed up the online phase.

The following section introduces the basics of $\mathscr{H}_2$-optimal model order reduction of linear time-invariant systems, while Sect. 3 focusses on parametric systems in particular. Sections 4 and 5 present details on our approximation of the mirror poles, including background on standard radial basis function interpolation with polynomial reproduction. Enhancements such as clustering and reuse of offline computations are treated in Sects. 6 and 7, the complete algorithm being recapitulated in Sect. 8. Finally, numerical results are presented in Sect. 9. We conclude by a discussion of the proposed method and its shortcomings that need to be addressed in future work.

2 $\mathscr{H}_2$-Optimal Model Order Reduction

Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^n$, $C \in \mathbb{C}^{1\times n}$. Throughout this paper we call a large-scale single-input single-output dynamical-system

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \tag{1}$$

stable if A has eigenvalues only in the left half plane. Model order reduction methods try to find a reduced stable dynamical system

$$\hat{\Sigma} : \begin{cases} \dot{x}(t) = \hat{A}x(t) + \hat{B}u(t), \\ \hat{y}(t) = \hat{C}x(t), \end{cases}$$

where $\hat{A} \in \mathbb{R}^{r\times r}$, $\hat{B} \in \mathbb{R}^r$, $\hat{C} \in \mathbb{C}^{1\times r}$ and $\hat{A}$ has eigenvalues only in the left half plane. Here $r \ll n$ and the map from the input $u \in \mathscr{L}_2(\mathbb{R}^+)$ to the output y of the original system is desired to be well approximated by the map from input to output $\hat{y}$ of the reduced system. In the frequency domain the input-output behaviour is characterized by the transfer function. The original transfer function H and the transfer function $\hat{H}$ of the reduced system are given by

$$H(s) = C(sI - A)^{-1}B, \qquad H(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B},$$

respectively. H and $\hat{H}$ are complex-valued rational functions defined on the complex plane and, due to the stability assumption, analytic in the closed right half-plane.

2.1 Optimality

Looking at the maximal error given as the absolute difference between the true and the reduced output we get (Gugercin et al., 2008)

$$\sup_{t>t_0} |y(t) - \hat{y}(t)| \leq \|H - \hat{H}\|_{\mathscr{H}_2} \|u\|_{\mathscr{L}_2},$$

where the $\mathscr{H}_2$-norm is defined as

$$\|H - \hat{H}\|_{\mathscr{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(\imath\omega) - \hat{H}(\imath\omega)|^2 d\omega. \tag{2}$$

The reduced order model we want to find is, therefore, the one which minimizes the $\mathscr{H}_2$-norm of the transfer function error for a given r.

First discovered by Meier and Luenberger (1967) and later developed for control theory (Gugercin et al., 2008; Bunse-Gerstner et al., 2010; van Dooren et al., 2008), necessary conditions for the reduced order transfer function to satisfy $\mathscr{H}_2$-optimality exist:

Theorem 2.1. *Given a stable dynamical system of the form (1), a reduced order system of order r minimizes (2) if it Hermite interpolates the original system at its mirror poles.*

The mirror poles of (1) are, therefore, the optimal interpolation points and are denoted by $\sigma_1, \ldots, \sigma_r$. Throughout this paper we assume that the reduced order size r is fixed. This means we always have r many interpolation points. Given these interpolation points one can create a reduced order model by a Petrov-Galerkin projection (Grimme, 1997), i. e. we can find V, W with $V^T W = I$ such that

$$\hat{A} = V^T A W, \quad \hat{B} = V^T B, \quad \hat{C} = CW. \tag{3}$$

Lemma 2.1. *Given $\sigma_1, \ldots, \sigma_r$. Moreover,*

$$(\sigma_i I - A)^{-1} B \in im(V),$$
$$(\sigma_i I - A)^{-T} C^T \in im(W).$$

Then the reduced order model generated by Petrov-Galerkin projection with W and V Hermite interpolates the original system at $\sigma_1, \ldots, \sigma_r$.

2.2 IRKA

Since we do not know these interpolation points a priori we use the well-known algorithm IRKA developed by Gugercin et al. (2008) to compute them. Upon convergence of IRKA a set of interpolation points is found. The system created by Hermite interpolation of them is guaranteed to be a local minimizer to the $\mathscr{H}_2$-optimization problem and often turns out to even yield the global optimum (though there is no theoretical backing for this so far). In the remainder of the paper these optimal interpolation points $\sigma_1, \ldots \sigma_r$ play an important role and will, for parametric systems, be considered as functions of p. As a function of p they are the argmin of a minimization problem which is dependent on p. For each individual problem this argmin does not need to be unique, which means that it is not actually a function of p. Furthermore, we will compute $\sigma_1, \ldots \sigma_r$ through IRKA which can only guarantee local optimal solutions. Furthermore the outcome is not unique. If we start the fixed point iteration with different starting values, the local optimium to which IRKA converges does not need to be the same. We use this algorithm in the creation of our parameteric model order reduction method. The input is a given set of matrices A, B, C, and the output will be the optimal interpolation points together with the projection matrices V, W given as in Lemma 2.1.

3 Parametric Systems

Having understood $\mathscr{H}_2$-optimal model order reduction for non-parametric systems, we now consider parametric systems. These systems have the characteristic trait that all A, B and C may be matrices depending on a parameter

$$\Sigma : \begin{cases} \dot{x}(t) = A(p)x(t) + B(p)u(t), \\ y(t) = C(p)x(t). \end{cases}$$

Here, as in the non-parametric case, $A(p) \in \mathbb{R}^{n \times n}$ is assumed stable, $B(p) \in \mathbb{R}^n$ and $C(p) \in \mathbb{C}^{1 \times n}$. We assume that the parametric dependence is smooth and $p \subset \Omega \subset \mathbb{R}^d$, where Ω is

a compact domain. Each realization of the parameter defines a system as described in (1). Therefore, we know how to create a reduced order model that (locally) minimizes the $\mathscr{H}_2$-error for a given parameter p^*. It is just as possible to compute $\sigma_1(p^*),\ldots,\sigma_r(p^*)$. We are, however, interested in a reduced order model for *all* parameters. The goal is to design an algorithm that can fast compute $\hat{A}(p),\hat{B}(p),\hat{C}(p)$ for all parameters p such that each system is close to the $\mathscr{H}_2$-optimal one. We denote the transfer function for the full system by

$$H(s,p) = C(p)(sI - A(p))^{-1}B(p) \tag{4}$$

and $\hat{H}$ accordingly for the reduced order system.

The first step is to create a metamodel for $\Sigma(p) = [\sigma_1(p),\ldots,\sigma_r(p)]$, the optimal interpolation points ordered ascending by absolute value of the imaginary part and then ascending in the value of the real part. If there are complex conjugate pairs they are ordered with the negative real part first. By metamodel we understand an approximation to a given smooth function, solely from evaluations of this function at discrete sampling locations within its domain. In particular, interpolatory methods based on radial functions are employed here.

3.1 Smoothness

The function $\Sigma(p)$, if it is a function at all, is not necessarily smooth. To understand this in more detail, we define $\Lambda : \mathbb{C}^{r\times r} \to \mathbb{C}^r$ as the function that takes a matrix to its eigenvalues ordered such that $-\Lambda$ is ordered as described above. We can write Σ as

$$\Sigma(p) = -\Lambda(\hat{A}(p)), \tag{5}$$

where $\hat{A}(p)$ is the optimal reduced order system matrix depending on the parameter p. The matrix $\hat{A}$ is not unique for two reasons. The first reason is that a system (1) and its reduction are only unique up to state-space transformations. This is why matrix interpolation for parametric model order reduction is a challenging problem (see Panzer et al. (2010); Amsallem and Farhat (2008)). On the other hand, this does not affect $\Lambda(\hat{A}(p))$, since eigenvalues, or poles, stay invariant under state-space transformations. The second problem is that, for a given parameter, the $\mathscr{H}_2$ optimal local minimizer is in general *not* unique. In other words, neither $\hat{A}(p)$ nor $\Lambda(\hat{A}(p))$ are uniquely defined and, therefore, $\Sigma(p)$ is not a function. In practice one can however assume that the global minimizer is unique almost everywhere. If there are several global optima in an dense set, we can pick one and otherwise we can ignore the point set in which multiple optima occur. (They are the point of jumps) And we can therefore define a function with jumps. We deal with jumps by clustering. The employed clustering algorithm is described in Sect. 6 and the criterion when to cluster is identified by computing the norm of the interpolant. If the norm of the interpolant is too large we increase the number of clusters until it no longer is. All further particulars of this heuristic are explained in Sect. 8 and discussed in some more details for specific numerical experiments, see Sect. 9. The definition of the norm and its meaning for radial basis function interpolation are discussed in Sect. 5.

Even if $\hat{A}(p)$ were a perfectly well-behaved and smooth function, one last important issue is that $\Sigma(p)$ could still be non-differentiable, as eigenvalues might intersect (see Fig. 1). Furthermore, splittings can occur if the eigenvalues go from real to imaginary pairs. In all the

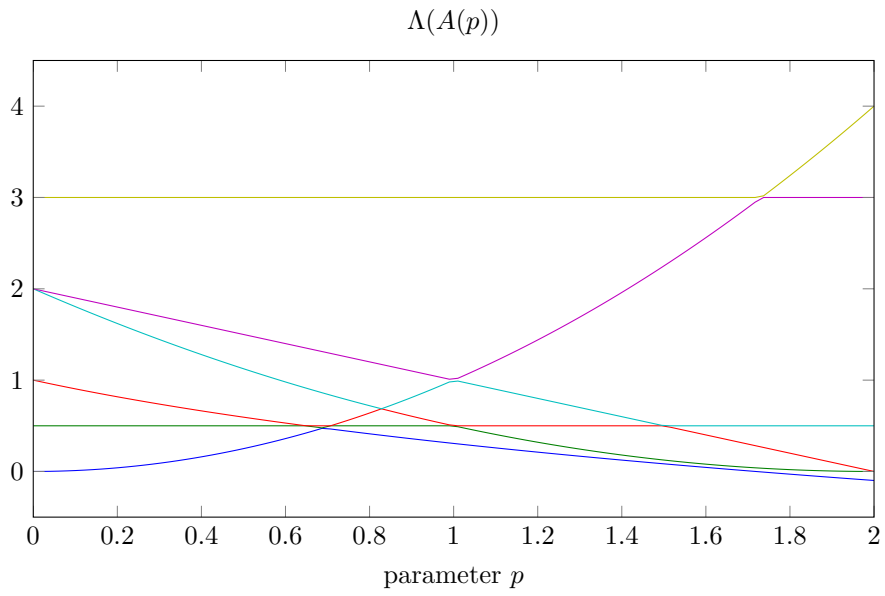

Figure 1: Eigenvalue Crossing: For a constructed matrix $A(p)$ we plot the eigenvalues as a function of p.

benchmark problems we consider, neither can such phenomena be observed, nor do difficulties arise with jumps. We, however, discuss a synthetic model in detail, which is constructed to exhibit the described issues. As a general solution strategy, we do not actually interpolate Σ but the coefficients of degree 4 polynomials instead, with which we can partially avoid the problem of crossing and splitting eigenvalues.

3.2 Offline-Online Decomposition

In the context of model order reduction methods for parametric systems one typically talks about an offline and an online phase in the reduction procedure. In the offline phase, time and computing resources are available and a reduced order system is created from the given large scale system. The online phase is where time and computing resources are limited. This is the phase in which we evaluate the reduced order model for different parameters. In our particular problem, the offline phase comprises three steps: We set up a reduced model at finite sampling locations in the domain, construct a metamodel for the optimal interpolation points from this sampling, together with a medium size model whose details are explained in Sects. 4 and 7, respectively. The online phase can be split into two parts, of which we are only interested in the first: The creation of the reduced order model for a given parameter. It involves the prediction of the optimal interpolation points by our metamodel and the projection of the medium model onto them. The second online phase is then the simulation of the reduced order model for that parameter. For linear systems this step depends mainly on the system size, and since

we assume that the reduced order system size is given and fixed, the complexity of the entire step is considered invariable by assumption. In view of a certain application, however, it is in general necessary to evaluate *all* costs in order to see whether our proposed algorithm is a good choice for the problem at hand.

4 Metamodel of Σ

The focus of this section is on creating a metamodel for the multivariate function $\Sigma(p) = -\Lambda(\hat{A}(p))$. As discussed in Sect. 3 this function can behave in a non-smooth way. Many efficient approximation approaches, in particular interpolation with finitely or infinitely smooth radial basis functions, assume the approximated relation f to be an element of a space of smooth functions, though. Only if a smooth formulation is found, full advantage can be taken of such approximation methods. Section 3 implies that $\hat{A}(p)$ is determined and computable up to similarity for a fixed parameter p^* if we assume $\mathscr{H}_2$-optimization by IRKA to converge to a unique minimum. An equivalence class of similar matrices possesses the same characteristic polynomial and eigenvalues, so that $\Sigma(p^*)$ is uniquely determined for each parameter by these assumptions.

The coefficients of the characteristic polynomial seem to be sufficiently smooth. We could however not find a numerical stable algorithm to compute roots of polynomials. We therefore group 4 eigenvalues together to get several polynomials of degree 4. Using the closed form representation creates a stable algorithm to compute the roots of the polynomial of degree 4. In order to describe our method we introduce functions Q and ρ, where $Q : \Pi_4 \to \mathbb{R}^{4+1}$ takes a polynomial to its coefficients, and $\rho : \mathbb{C}^4 \to \Pi_4$ is defined by $\rho(x)(s) = \prod_{i=1}^{4}(s - x_i)$. We are, via standard methods, going to interpolate

$$f(p) = [f_1(p), \ldots, f_{r/4}(p)] \in \mathbb{R}^{r/4+r}, \text{ where } f_i(p) = Q \circ \rho(-[\Sigma_{4i-3}(p), \ldots, \Sigma_{4i}(p)]) \in \mathbb{R}^5. \tag{6}$$

Here, we assume for simplicity that r is a multiple of 4. Suppose there exists an attractive local minimum to which IRKA converges locally. If this minimum is stable for perturbations of p and if we can find a good enough starting point for IRKA, $f(p)$ inherits the smoothness of $A(p)$. Hence, evaluating samples of f and interpolating between them is rather straightforward.

The metamodel $\tilde{\sigma}$ of Σ is such that

$$\tilde{\Sigma} = -[\rho^{-1} \circ Q^{-1} \circ \tilde{f}_1, \ldots, \rho^{-1} \circ Q^{-1} \circ \tilde{f}_{r/4}].$$

In order to evaluate ρ^{-1}, we use the MATLAB® built-in function `roots`. One can by tracing the eigenspaces possibly check if the created model is reasonable since we do that somewhat undesirable splitting of the eigenvalues into clusters of 4. This is not something that has been implemented yet. Since the computed Σ, considered as a set, is closed under conjugation and the real parts of its elements are positive, the range of f consists of real positive values only. It is crucial to make sure that $\tilde{f}$ returns only positive values, too.

Details on how we set up a metamodel $\tilde{f}$ of

$$f : \Omega \to \mathbb{R}^{r+1}$$

and ways to enforce positivity follow in Sect. 5. Interpolation by radial basis functions with polynomial detrending is a natural choice if smooth, nonlinear, multivariate functions need to be approximated. First, possibly scattered sampling locations $p_1, \ldots, p_N$ are selected. Scattered samplings must be considered since IRKA is not guaranteed to converge, so that we can only include converged results in the sampling and gaps may remain. For each sample point p_j we compute the $\mathscr{H}_2$-optimal reduced order model and its reduced order poles, or eigenvalues, $\lambda_1(p_j), \ldots, \lambda_r(p_j)$. Now we can evaluate $f(p_j) \in \mathbb{R}^{r/4+r}$, as in (6). The resulting interpolation data $p_1, \ldots, p_N, f(p_1), \ldots, f(p_N)$ are then used to create the interpolant $\tilde{f}$.

Yet, $\rho(\hat{A}(p))$ and thus $f(p)$, being an optimization result, can still show discontinuities with regard to the variable p, if IRKA, convergence given, returns from different local minima, see Sects. 2.2 and 3.1. Since particularly *local* discontinuities can be observed for $f(p)$ in synthetic problems (details to be conveyed in Sect. 9.4), we apply a clustering approach in order to partition the domain Ω into subdomains Ω_k where $f|_{\Omega_k}$ is sufficiently smooth, see Sect. 6. Given the interpolant $\tilde{f}$ on a (sub)domain $\Omega_{(k)}$, we use a norm (details in Sect. 5) to decide whether clustering is necessary. The clustering can also help to detect splittings that occur between the groups of eigenvalues formed above.

5 Radial Basis Function Interpolation

In the following we give a brief introduction to the theory of reproducing kernels, positive definite functions and radial basis functions in order to be able to discuss their application to parametric model order reduction as described in the previous section.

Consider a domain $\Omega \subset \mathbb{R}^d$ and a class of functions $f : \Omega \to \mathbb{C}$ that form a Hilbert space $\mathscr{H}$ with inner product $(\cdot,\cdot)$. A reproducing kernel $\kappa : \Omega \times \Omega \to \mathbb{C}$ is defined by the two relations (Aronszajn, 1950)

$$\begin{aligned} \forall y \in \Omega: &\quad \kappa(\cdot, y) \in \mathscr{H} \\ \forall f \in \mathscr{H}, y \in \Omega: &\quad f(y) = (f(\cdot), \kappa(\cdot, y)). \end{aligned} \tag{7}$$

If a reproducing kernel κ exists for a given space $\mathscr{H}$ on a domain Ω it is unique and thus characterizes $\mathscr{H}$. We, therefore, denote the inner product as $(\cdot,\cdot)_\kappa$ and the norm by $\|\cdot\|_\kappa$. The Hilbert space $\mathscr{H}$, on the other hand, is the closure of the set F of all linear combinations of the kernel

$$F = \operatorname{span}\{\kappa(\cdot, x) : x \in \Omega\}$$

and is called the native space of its corresponding kernel (Schaback, 1999). For functions $f = \sum_i \xi_i \kappa(\cdot, x_i)$ and $g = \sum_j \zeta_j \kappa(\cdot, x_j)$, (7) implies that

$$(f,g)_\kappa = \left(\sum_i \xi_i \kappa(x, x_i), \sum_i \zeta_i \kappa(x, x_i) \right)_\kappa = \sum_{i,j} \xi_i \overline{\zeta}_j (\kappa(x, x_i), \kappa(x, x_j))_\kappa = \sum_{i,j} \xi_i \overline{\zeta}_j \kappa(x_j, x_i).$$

Let arbitrary $\xi_i \in \mathbb{C}$, $x_i \in \mathbb{R}^d$ define some function $f \in F$ as above. From the corresponding norm $\|f\|_\kappa^2 = (f,f)_\kappa$ we get

$$\sum_{i,j} \xi_i \overline{\xi}_j \kappa(x_j, x_i) \geq 0, \tag{8}$$

the inequality being strictly satisfied for nontrivial κ if all $\xi_i \neq 0$. A kernel κ for which (8) holds, is called positive definite or, if it defines a norm, strictly positive definite.

Among the many specific kernels and their native Hilbert spaces studied in the literature, we particularly consider those that arise from continuous, strictly positive definite functions ϕ (Stewart, 1976), such as the so-called radial basis functions (Schaback, 1999; Fasshauer, 2007)

$$\kappa(x,y) \equiv \phi(\|x-y\|_2).$$

Radial basis functions are a popular choice for interpolation, since they allow for scattered sampling data and – depending on their smoothness – polynomial or spectral approximation orders. The interpolation process for a function $f \in \mathscr{H}$, sketched in Sect. 4, can now be given in more detail: Once f has been evaluated for each x_i of a sampling $X \subset \Omega$, where $|X|$ is finite, the interpolant $\tilde{f} \in F$ is then chosen such that $N = |X|$, $x_i \in X$, where all $\xi_i \in \mathbb{C}$ are determined by the interpolation condition

$$\forall i = 1,\ldots,N: \quad \tilde{f}(x_i) = \sum_j \xi_j \kappa(x_i,x_j) \equiv f(x_i), \tag{9}$$

which constitutes a system of linear equations for $\xi_1,\ldots,\xi_N$.

Notice that any additional inequality, such as the assertation $\tilde{f}(\cdot) \geq 0$ from the previous section, turns the solution of (9) into a constrained optimization problem. Since this is not required by our numerical examples, it has not been implemented here.

For a given sampling and strictly positive definite kernel, radial basis function interpolation is known to be optimal in the sense that the interpolant $\tilde{f}$ has minimal native norm $\|\cdot\|_\kappa$ among all other interpolants of f from the native space. In addition, the interpolant $\tilde{f}$ is the best approximation to f among all other approximants

$$\tilde{s} = \sum_i \xi_i \kappa(\cdot,x_i), \quad x_i \in X, \xi_i \in \mathbb{C},$$

given the sampling X (Fasshauer, 2007, Chap. 18). Here, "best approximation" is again defined with respect to the norm of the native space

$$\forall \tilde{s}: \ \|f-\tilde{f}\|_\kappa \leq \|f-\tilde{s}\|_\kappa.$$

Often radial basis function interpolation is used with polynomial detrending. Then the ansatz function is given by

$$\tilde{f}(x) = \sum_{i=1}^{N} \xi \, \kappa(x,x_i) + \sum_{j=1}^{k} \alpha_j r_j(x),$$

where $r_1,\ldots,r_k$ span the chosen polynomial space. The function is determined by $\tilde{f}(x_i) \equiv f(x_i)$ and $\sum_i r_j(x_i)\xi_i \equiv 0$ for all j leading to a linear saddle point system. The latter condition ensures that, if a polynomial is interpolated, the first term in the ansatz is the zero function.

Since we apply radial basis function interpolation to characteristic polynomials arising as local minima of (2), detection of discontinuities is crucial. The native norm can be understood as an indicator for smoothness. Such a notion of the norm is used to detect discontinuities and find a partitioning into areas where the interpolated function is smooth. In the following we present some arguments that illustrate how the norm is related to smoothness.

An interesting way to relate an increasing norm $\|f\|_\kappa$ to decreasing smoothness of f is by sampling inequalities (Rieger et al., 2010). Sampling inequalities bound a weaker norm by the sum of a stronger norm added to a norm depending on the given sampling X. Theorems 1 and 3 of Rieger et al. (2010) bound derivatives of functions $f \in \mathscr{H}$ in $L_q, 1 \le q \le \infty$ by the norm of $\mathscr{H}$

$$\|D^\alpha f\|_{L_q(\Omega)} \le C_1 \left(h^\tau \|f\|_\kappa + h^{-|\alpha|} \|f(X)\|_{\ell_\infty(\mathbb{R}^{|X|})} \right), \tag{10}$$

where C is a positive constant, α is a multi-index, h the so-called fill distance, and τ the sampling order depending on α, q and κ. Here, the fill distance

$$h \equiv \sup_{y \in \Omega} \max_{x \in X} \|x - y\|_2, \tag{11}$$

given X and Ω, is a general way to assess the quality of a sampling. From (10) we see that large derivatives lead to a large norm $\|f\|_\kappa$.

There are further arguments that support the idea that nonsmoothness is reflected in an increasing native norm $\|\cdot\|_\kappa$. It is known that the native spaces of Matérn functions are Sobolev spaces, the popular thin plate splines generate Beppo-Levi-spaces (Fasshauer, 2007, Chap. 13). Thus the norm includes derivatives of or up to a certain degree. A characterization of the native space of strictly positive definite functions in terms of Fourier transforms $\mathscr{F}f$ and $\mathscr{F}g$ of functions $f, g \in \mathscr{H}$ gives us additional evidence (Fasshauer, 2007, p. 107)

$$(f,g)_\kappa = \left(\frac{\mathscr{F}f}{\sqrt{\mathscr{F}\phi}}, \frac{\mathscr{F}g}{\sqrt{\mathscr{F}\phi}} \right)_{L_2(\mathbb{R}^n)}.$$

If $\mathscr{F}\phi(\omega)$ is smooth and vanishes at infinity, $\mathscr{F}f$ must vanish at infinity, too, decaying even faster (Fasshauer, 2007, p. 110). This applies, for example, to the Fourier transform of the popular Gaussian radial basis functions (Fasshauer, 2007, Chap. 4).

To summarize the above reasoning, we conclude that the norm of an interpolant in the native space is a valid indicator that allows us to detect a variety of problems related to smoothness, which is important in our application.

Numerical results are presented only for Gaussian radial basis function interpolation with linear detrending. In other words the kernel is given by $\kappa(x,y) = e^{-\|x-y\|/2\sigma^2}$ and $r_1(x) = 1$, $r_2(x) = x$. The function `ModelRBF` used in our algorithm, see Sect. 9, computes all ξ_i and α_i via a linear system. Except for σ which can be selected by leave-one-out cross-validation, these are the only quantities used to determine the radial basis function interpolant.

6 Clustering Approach

The general idea of clustering is: Given a set of points $S = \{s_1, \ldots, s_N\}$ in $\mathbb{R}^n$ we want to partition them into k disjunct subsets S_i (clusters) with $S = \bigcup_{i=1}^k S_i$. Visually this is often easy for the human eye in $\mathbb{R}^2$ but, algorithmically, it poses a number of problems. A well-known and popular algorithm is the so-called k-means algorithm which attempts to minimize distortion.

1. Initial Step: Choose k arbitrary centers $c_1^{(1)}, \ldots, c_k^{(1)} \in S$.

2. Assignment Step: Each point gets assigned to a unique cluster such that

$$S_i^{(t)} = \{s_j : \|s_j - c_i^{(t)}\| \leq \|s_j - c_{i^*}^{(t)}\| \text{ for all } i^* = 1, \ldots, k\}.$$

3. Update Step: Calculate the new centers $c_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{s_j \in S_i^{(t)}} s_j$.

6.1 Spectral Clustering

If the smooth parts of the graph of $f(p)$ are spatially well separated from one another, as in all our test cases, this simple clustering algorithm is an adequate solution to the problem of discontinuities described in the previous sections. There are, however, alternative clustering approaches better fit to partition an evaluated sampling into its linked subsets, e. g. so-called spectral clustering. The results obtained with both methods do not differ significantly for the presented test problems, though.

One of the steps within spectral clustering is the k-means algorithm again. In that step any other clustering algorithm can be employed as well. Here we give an overview of spectral clustering:

1. Create $A \in \mathbb{R}^{N \times N}$ $A_{ij} = \exp(-\|s_i - s_j\|^2 / 2\sigma^2)$ if $i \neq j$ and $A_{ii} = 0$.
2. With D a diagonal matrix whose diagonal entries are given by the sum of A's rows, then construct $L = D^{-1/2} A D^{-1/2}$.
3. Let $x_1, \ldots, x_k$ be the k largest eigenvectors of L, orthogonal. $X = [x_1 \ldots x_k] \in \mathbb{R}^{N \times k}$.
4. Form Y such that $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$ (normalizing each of X's rows).
5. Do k-means clustering on the set of points made up of the rows of Y.
6. Assign s_i to cluster j if row i was assigned to cluster j.

The first step involves Gaussian radial basis functions, where the scaling parameter σ^2 can be chosen automatically to get the "best" clustering results. An analysis of this algorithm and more details are given by Ng et al. (2002).

6.2 Criteria for Clustering

The function `Cluster-f` used in the Algorithm RBF-IRKA in Sect. 8 performs the spectral clustering. Its input is a set of points $S = \{s_1, \ldots, s_N\}$ where $s_i = (p_i, f(p_i)) \in \mathbb{R}^d \times \mathbb{R}^{5/4r}$ and its outputs are k subsets $S_1, \ldots, S_k$ together with the centers $c_1, \ldots, c_k \in \mathbb{R}^d$, one for each cluster. Here we pick just the part of the centers belonging to the parameter domain.

7 Medium Size Model

The last step of the puzzle to derive a fast parametric model order reduction algorithm is the creation of a model of intermediate size. This step is particularly important for very large scale problems. The additional cost in the offline phase is negligible, but the amount of time that can be saved in the online phase is significant. The basic idea is to create a medium size model by projecting the large scale model onto a subspace via Galerkin projection. This method is only efficient if the original system matrices are given in an affine form

$$A(p) = \sum_{j=1}^{M} \alpha_j(p) A_j, \quad B(p) = \sum_{j=1}^{M} \beta_j(p) B_j, \quad C(p) = \sum_{j=1}^{M} \gamma_j(p) C_j, \tag{12}$$

where M is not too large. We then construct a matrix $W \in \mathbb{R}^{n \times m}$ and a medium system

$$A_m(p) = \sum_{j=1}^{M} \alpha_j(p) W^T A_j W, \quad B_m(p) = \sum_{j=1}^{M} \beta_j(p) W^T B_j, \quad C_m(p) = \sum_{j=1}^{M} \gamma_j(p) C_j W.$$

We use a Galerkin projection to guarantee stability in case A is dissipative. The chosen projection matrix W can be generated from the byproducts of our parametric metamodel construction process. During that process we compute the optimal projection matrices (Petrov-Galerkin projection as in (3))

$$V(p_1), W(p_1), \ldots, V(p_N), W(p_N)$$

for each sampled value of the interpolation parameter. W then originates from the matrix

$$V = [V(p_1), \ldots, V(p_N), W(p_1), \ldots, W(p_N)]$$

by taking the relevant left singular vectors of V, which are those whose singular values are not close to zero. Such a choice guarantuees that the medium size model interpolates at $p_i, \sigma_k(p_i)$ for all i,k. We have that $W^T W = I_m$ with $m \leq 2r \times N$. Not actually a good reduced order model, this medium size model is just a good approximation to the original model in the neighborhood of the interpolation points. However, this is all we need when we set up the reduced order models.

Other methods can be used to generate a medium size model: the framework of two variable rational interpolation by Antoulas et al. (2012) or projection-based methods as decribed by Baur et al. (2011a). On the other hand, the easy and effective approach presented here typically leads to stable systems and has error behaviour, in fact almost as good as the original system, see Sect. 9. Moreover, we do not have to perform any extra work in comparison to the other methods, which is why we prefer this method, and we get results that are as good or better according to preliminary tests not presented in this paper. The reason for the good performance might be found in the very specific purpose this medium model serves.

8 Algorithm

This section puts together all the points discussed so far. We can sum up our proposed parametric model order reduction in pseudo-code, see Algorithm 1.

Algorithm 1 Offline phase

Require: $p_1,\ldots,p_N \in \mathbb{R}^d$, reduction order r (multiple of 4), $A_j, B_j, C_j,\ j = 1,\ldots,M$

Ensure: metamodels $\tilde{f}_1,\ldots,\tilde{f}_k$ for each cluster and cluster centers $c_1,\ldots,c_k \in \mathbb{R}^q$, A_m^j, B_m^j, C_m^j

Let $\Sigma_0 \in \mathbb{R}$ be a random r-dimensional vector
for $i \leftarrow 1 : N$ **do**
    $[\Sigma_i, V_i, W_i] \leftarrow \text{IRKA}(A(p_i), B(p_i), C(p_i), \Sigma_{i-1})$
    sort Σ
    $f(p_i) \leftarrow [\texttt{poly}(-[\Sigma_1,\ldots,\Sigma_4]),...,\texttt{poly}(-[\Sigma_{r-3},\ldots,\Sigma_r])]$
$[W, S, \,] \leftarrow \text{svd}([V_1,\ldots V_N, W_1,\ldots,W_N]), 0)$
$R \leftarrow \text{rank}(S)$,
$W \leftarrow W(:, 1:R)$
$A_m^i \leftarrow W^T A_i W,\ B_m^i \leftarrow W^T B_i,\ C_m^i \leftarrow C_i W$
for $i \leftarrow 1 : N$ **do**
    $s_i \leftarrow [p_i, f(p_i)] \in \mathbb{R}^{q+5/4r}$
$S \leftarrow \{s_1,\ldots,s_N\}$
$k \leftarrow 1$
while $k < k_{\max}$ **do**
    $[I_1,\ldots,I_k, c_1,\ldots,c_k] \leftarrow \texttt{Cluster-f}(S,k)$
    for $i \leftarrow 1 : k$ **do**
        $\tilde{f}_i \leftarrow \text{ModelRBF}(p(I_i), f(p(I_i))$
    if $\max_i(\|\tilde{f}_i\|_\kappa) < N_{\max}$ **then**
        done
    else
        $k \leftarrow k + 1$

Comments

1. We typically pick the parameters $p_1, \ldots, p_N$ by latin hypercube sampling in the parameter domain. If $q = 1$ we use a regular grid, however. (There are many other sampling techniques, which we did not investigate. This could be future work.)

2. The decision on the chosen reduction order r can be made by looking at a good reduction order for a particular parameter. In general, for a complex problem r can vary over the parameter domain. Then it makes sense to decompose the parameter domain into regions of different reduction orders and create a metamodel for the domains individually.

3. The matrices A_j, B_j, C_j are as in (12).

4. The centers belong to parameter space, even though we cluster in the orthogonal direct sum of parameter space and coefficient space.

5. We start IRKA first with a random vector Σ_0, but for subsequent runs we use solutions of a close-by parameter. In the setup we assume that parameters p_i are ordered in such a way that neighbors in the ordered sequence are also close in the parameter domain. A parallelization of the algorithm could just start with a random starting vector for all parameters.

6. We need to sort Σ since we heuristically separate the points into quadruples in order to convert them to several distinct polynomials. They are sorted by the value of the real part. But one needs to make sure that the set in one grouping is invariant under complex conjugation.

The sub-functions used in this algorithm are IRKA as described in Sect. 2.2, `Cluster-f` as described in Sect. 6.2 and a radial basis function approximation algorithm. We use Gaussian radial basis functions with linear detrending in all our numerical experiments, but our preliminary tests have shown that other choices give equivalently good results as expected. Algorithm 1 returns the reduced order model, which in this case is given by the cluster centers $c_1, \ldots, c_k$, the metamodels $\tilde{f}_j$ on each of the clusters, as well as the medium size matrices A_m^i, B_m^i, C_m^i. With this information we can fast set up a reduced order state-space system for any given parameter. And fast is important here. Two constants of the algorithm need to be chosen beforehand, $k_{\max}$ and $N_{\max}$. We simply pick $k_{\max}$ to be a reasonable number depending on the total number of data points N such that the average sample size of a cluster can still be expected to cover the corresponding cluster domain well in terms of the fill distance. Since it turns out that $k = 1$ is sufficient in all tested real world examples we cannot report on best choices from experience. The constant is still included to prevent an infinite while-loop, though. The more important number by far is $N_{\max}$ in practice. We choose it to be $50/h$ where h is the fill distance as in equation (11). In the online phase we want to create the system matrices of size r given parameter p^*. The first step is to determine to which of the k clusters the chosen parameter belongs by finding the center c_j which is closest to p^*. Once we know its index j we can feed the radial basis function approximation $\tilde{f}_j$ together with p^* into the algorithm. For details of the online phase see Algorithm 2.

Algorithm 2 Online phase

Require: p^*, $\tilde{f}_j$, A_m^i, B_m^i, C_m^i
Ensure: $A_r(p^*), B_r(p^*), C_r(p^*)$

$f \leftarrow \tilde{f}_j(p^*)$
$\Sigma \leftarrow -\texttt{roots}(f)$
$A_m \leftarrow \sum p_i^* A_m^i,\ B_m \leftarrow \sum p_i^* B_m^i,\ C_m \leftarrow \sum p_i^* C_m^i$
for $i \leftarrow 1 : r$ **do**
    $V_i \leftarrow (\Sigma_i I - A_m)^{-1} B_m$
    $W_i \leftarrow (\Sigma_i I - A_m^T)^{-1} C_m^T$
$[V, \ , \ ] \leftarrow \text{svd}(\text{real}([V_1, \dots V_N]), \text{imag}([V_1, \dots, V_N]))$
$[W, \ , \ ] \leftarrow \text{svd}(\text{real}([W_1, \dots W_N]), \text{imag}([W_1, \dots, W_N]))$
$V \leftarrow V(:, 1 : r)$
$W \leftarrow W(:, 1 : r)$
$A_r \leftarrow W^T A_m V,\ B_r \leftarrow W^T B_m,\ C_r \leftarrow C_m V$

When we refer to the algorithms RBF-IRKA and RBF-IRKAm in Sect. 9 we mean the combination of the offline and the online algorithm described above, where RBF-IRKAm is just as described, equalling RBF-IRKA except for the additional creation of a medium model. When referring to RBF-IRKA, W in Algorithm 1 is simply assigned the identity.

9 Numerical Results

In order to show the accuracy and efficiency of our method we present four different parametric examples. The first example is a parametric benchmark problem describing a cantilever beam. Its challenging feature is a transfer function that cannot easily be interpolated in the frequency domain as it has many peaks. The second and third examples are represented by one and the same large-scale system, modelled once with a single real parameter and once again with three independent real parameters. Actual timings are shown for the example, which confirm the efficiency of our proposed algorithm for a three-dimensional parameter domain. The fourth example has been artificially created and illustrates why clustering is needed and what problems can appear at least theoretically, including a detailed discussion.

9.1 Parametric Beam Model

The parametric beam model is the finite element discretization of a 3D cantilever Timoshenko beam (Panzer et al., 2009). Parametrized by the length $\in [0.8\,\text{m}, 1.2\,\text{m}]$ of the beam, the system has a size given by $n = 240$. In Fig. 2 we can see the transfer function for different values of the parameter.

In order to compare the $\mathscr{H}_2$-error between the full and the reduced order model it is helpful to know the $\mathscr{H}_2$-norm of the system and how it varies with the parameter shown in Fig. 3.

In Fig. 4 (Fig. 5, respectively) we see the $\mathscr{H}_2$-error for reduced order systems of size $r = 4$ with $N = 3$ ($N = 5$, respectively) interpolation points. We compare the error for systems

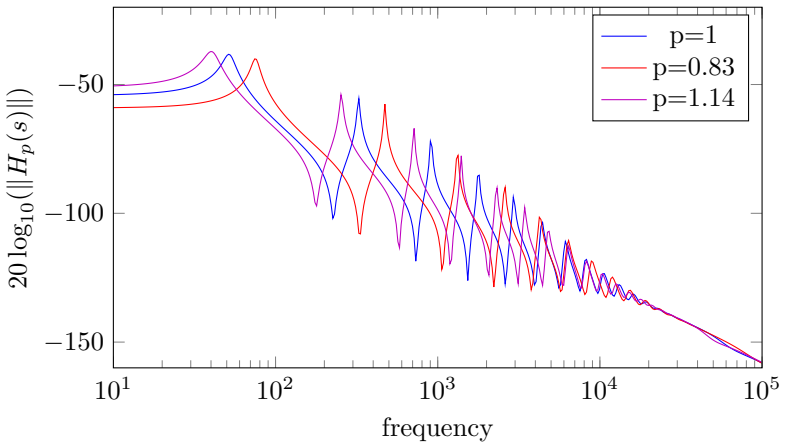

Figure 2: Bode plot of the parametric beam model for three different choices of the parameter. The model is parametrized by the beam length. Transfer response peaks in the frequency range move left along the x-axis, when the length increases

reduced via IRKA directly, another reduced system created by radial basis function interpolation with subsequent projection on the interpolated poles (RBF-IRKA), and a third being the projection of the approximated medium size system onto the computed interpolation points (RBF-IRKAm). It turns out there is no need to cluster at all, partly due to the little number of samples N in this beam example. Since $r = 4$ we directly use *all* coefficients of the characteristic polynomial as the function f.

Notice the interpolation points in Fig. 4 being those where the curves match. The optimal reduced order model performs better than the one created by interpolation. However, the RBF-IRKA and RBF-IRKAm are still comparably good and the difference between inclusion and exclusion of the medium size model is not visible in the plot. A reduction of the system size from 240 to 4 has been achieved with an $\mathscr{H}_2$-error of around 10^{-6} which corresponds to a

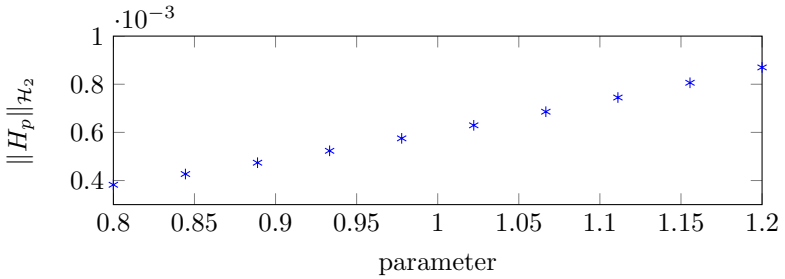

Figure 3: $\mathscr{H}_2$-norm of the Timoshenko beam system as a function of the parameter

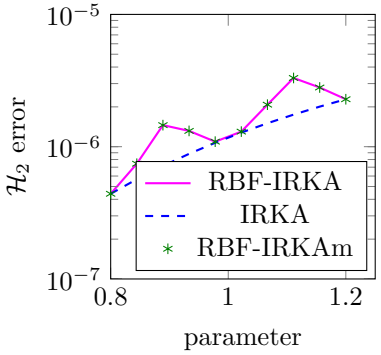

Figure 4: $\mathscr{H}_2$-error comparison between IRKA, RBF-IRKA and RBF-IRKAm for three interpolation points and reduced size $r = 4$. IRKA behaves slightly better, while there exists no visible difference between the other two algorithms

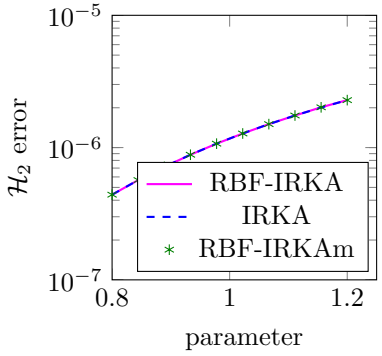

Figure 5: $\mathscr{H}_2$-error comparison between IRKA, RBF-IRKA and RBF-IRKAm for five interpolation points and reduced size $r = 4$. All three algorithms exhibit errors of comparable magnitudes

relative error of about 10^{-3}. This is achieved by a parametric model and the optimal error (IRKA error) is of the same magnitude in the case of five interpolation points (see Fig. 5).

Creating the reduced order model for each individual parameter directly via IRKA is not an option in practice as it depends on the reduced size r and the number of IRKA iterations $\#\texttt{IRKA}_{\texttt{iter}}$. Let T be the time it takes to solve one large system IRKA takes $2r \times \#\texttt{IRKA}_{\texttt{iter}} T$ to construct the model for one parameter. In the case of the RBF-IRKA algorithm it takes $2rT$ and in the case of the RBF-IRKAm it only takes $2rt$ where t is the time for one medium size system solve. The large system size of this Timoshenko beam is 240 compared to a medium system size of 32 (40 respectively) for 3 (5 respectively) interpolation points. The average number of IRKA iterations in this example is around 10. Actual timings are given for the following example, which is large scale.

9.2 Anemometer

A hot-wire anemometer is a device which consists of a heater and temperature sensors on each side. It is used to measure the fluid flow past the device. The underlying model is a convection-diffusion partial differential equation. In this first example, we will study the one-parameter model where the parameter is given by the fluid velocity. This model can be

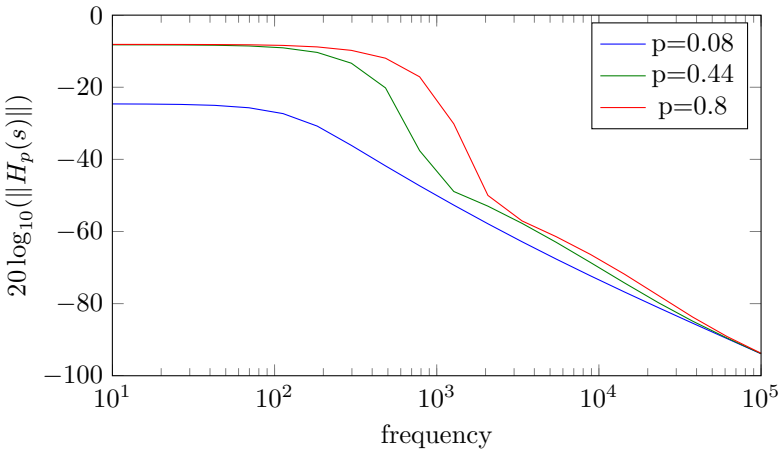

Figure 6: Bode plot for the 1D anemometer, the three curves representing different choices for the parameter

found in the benchmark collection "MOR Wiki"[1] and is explained in detail by Moosmann et al. (2005); Baur et al. (2011b). The system size n of the anemomenter equals $29,008$. One single system solve is expensive and we, therefore, need to avoid them at least in the online phase. Furthermore, the anemometer's parametric dependency is affine, and only the system matrix A depends on the parameter: $A(p) = A_0 + pA_p$, where A_p and A_0 are given system matrices. We show the transfer function, as for the beam example, considering different values of the parameter, see Fig. 6. In order to illustrate the quality of the reduced order model we compare for several parameters the $\mathscr{H}_2$-error between the reduced and full order model, where the reduced order model is either created by IRKA directly or by RBF-IRKAm. Both reduced order models are of size $r = 4$, where we take $N = 5$ interpolation points for RBF-IRKAm. The corresponding $\mathscr{H}_2$-error is plotted in Fig. 7. Use of clustering is not necessary, as in the beam example above partly due to the sample size of N which is small but sufficient. Furthermore $r = 4$ such that we again compute all coefficients of the characteristic polynomial. In order to understand the relative $\mathscr{H}_2$-error we compute the $\mathscr{H}_2$-norm for some values of the parameter to find that it grows from $p = 0$ to $p = 1$ with

$$\|H_0\|_{\mathscr{H}_2} \approx 0.014, \quad \|H_1\|_{\mathscr{H}_2} \approx 24.$$

In addition, Table 1 compares construction timings of reduced models, divided into offline and online phase for $r = 4, N = 5$ as above and $r = 8, N = 10$. In these cases the $\mathscr{H}_2$-error is of the same magnitude for IRKA and RBF-IRKAm. Trivially by definition, IRKA spends its entire time in the online phase. RBF-IRKAm's online phase timings comprise radial basis function evaluation and set-up of the one-parameter reduced order system. It can be observed

[1] http://www.modelreduction.org

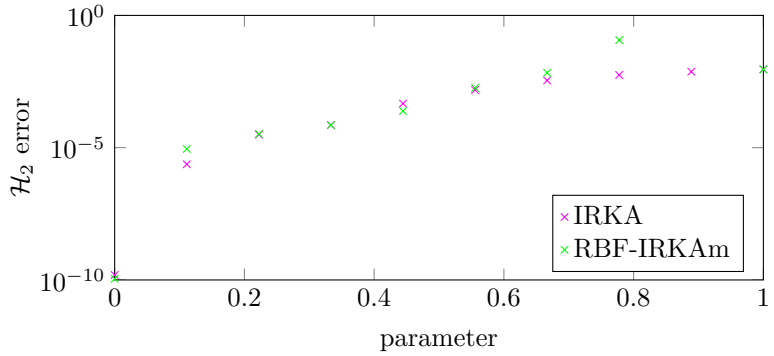

Figure 7: $\mathscr{H}_2$-error, comparison between IRKA and RBF-IRKAm, with $N = 5$, $r = 4$

	$r = 4, N = 5$		$r = 8, N = 10$	
	offline	online	offline	online
IRKA	0 s	61 s	0 s	421 s
RBF-IRKAm	285 s	0.04 s	1,234 s	0.2 s

Table 1: Timings for the online and offline phase of the 1D anemometer example, IRKA compared to RBF-IRKAm

from these results, that for $r = 4$ RBF-IRKAm is faster overall (considering both online and offline phase), when we need to evaluate the system at five different parameter locations at least. For $r = 10$ system evaluations for two parameter locations already compensate what has been spent in the offline phase. In general, we see that the timings for the online phase, the crucial phase, are relatively low and mostly yield a comparable error of the same order of magnitude as IRKA.

9.3 Anemometer 3D

In order to show that the proposed algorithm is feasible for problems in which the parameter does not lie in a one-dimensional interval but in a multidimensional space, we present results for a three-dimensional version of the previously discussed anemomoter model. The three parameters are given by $p_0 = c$ (specific heat), $p_1 = v$ (fluid velocity) and $p_2 = \kappa$ (thermal conductivity). The affine structure is then given by

$$A(p) = A_0 + p_2 A_1 + p_0 p_1 A_2, \quad E = E_0 + p_0 E_1.$$

Error bounds for radial basis function interpolation (Rieger et al., 2010, Sect. 4.3) depend on the dimension of the parameter space in many ways. While the involved constant as well

	$r=4, N=5$	$r=8, N=5$	$r=8, N=10$	$r=12, N=10$
IRKA	3.19×10^{-5}	3×10^{-8}	3×10^{-8}	1.3×10^{-8}
RBF-IRKAm	3.21×10^{-5}	1×10^{-6}	2×10^{-8}	1.6×10^{-8}

Table 2: Average $\mathscr{H}_2$-error of the 3D anemometer example, IRKA compared to RBF-IRKAm

as the norm of the native space increase with dimensionality, the approximation order gets worse. Therefore, larger errors are to be expected in higher dimension. On the other hand, standard radial basis function interpolation scales comparably good and well-behaved or, in other words, *smooth* functions in a high-dimensional parameter domain can still be approximated and accurate results can be achieved. The modest increase to three dimensions does not pose a problem in this test case. The average $\mathscr{H}_2$ error for different reduction orders r and number of interpolation points N of the reduced model and the full model are shown in Table 2. "Average" signifies we take the mean for ten randomly selected parameters $p_1, \ldots, p_{10}$ in the domain, i. e.

$$\frac{1}{10}\sum_{i=1}^{10} \|\hat{H}_{p_i} - H_{p_i}\|_{\mathscr{H}_2},$$

where $\hat{H}_{p_i}$ is computed via IRKA and RBF-IRKAm, respectively.

9.4 Synthetic Model

Finally, we include this synthetic model in our discussion because none of the presented practical problems exhibits the discontinuity issues described in Sect. 3.1. The synthetic model, however, makes clustering necessary. It is also the only model we found in which direct interpolation of $\sigma_1, \ldots, \sigma_r$ fails as it is represented by a system where eigenvalue splittings and crossings occur. First we have a look at this system's transfer function in Fig. 8 for a number of different parameter values. Here we can observe that the transfer functions look rather different for different parameters. In fact, the system is constructed in order for the peaks to vary in location and intensity over the parameter range, such that no parts of the different transfer functions are equal. Not even their behaviour towards infinity is the same.

The next step is to consider how the eigenvalues behave as a function of the parameter. This can be seen in Fig. 9, where the real parts of the four eigenvalues are plotted. For most of the interval one can see three values, which means we have two real eigenvalues and one complex conjugate pair of eigenvalues. Even in such a simple one-parameter example it is not clear how to create a metamodel: Notice that some eigenvalues split. If we, however, compare eigenvalues to normalized coefficients, i. e. to the function f, see the coloured dots in Fig. 10, it becomes clear that the matter can be solved by a good clustering algorithm. For the human eye, the parameter domain clustering in this picture is obvious.

As explained above in Sect. 6.2 we select the $d+5/4r$-dimensional set of the parameters together with the coefficients of the characteristic polynomial as clustering features. The resulting piecewise defined metamodel for the function $f(p)$, with the parameter space partitioned by clustering, is shown in Fig. 10, see the black lines. The clustering algorithm generates three subsets and thus yields a metamodel $\tilde{f}(p)$ which involves jumps. One can see that around the

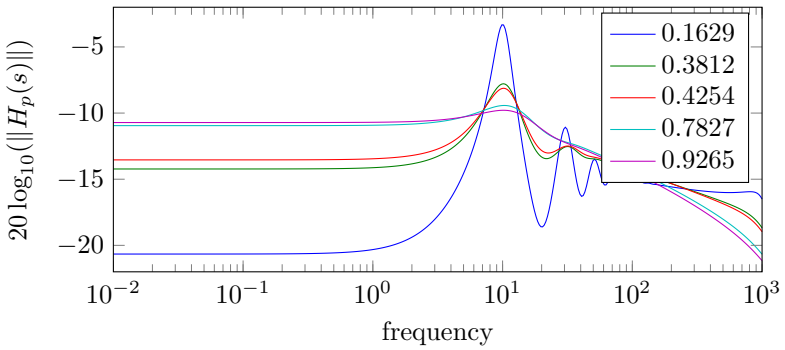

Figure 8: Transfer functions for several parameter values, synthetic example. This parametric system is designed to exhibit peaks varying in location and density

second jump, data points are missing due to the fact that IRKA does not converge for parameters in that region. We started with 30 samples of p to get 28 data points. Here we pick $k_{\max} = 5$ and $N_{\max} = 100N = 3{,}000$. The filling distance in this case is $h = 1/(2N)$, meaning $N_{\max} = 50/h$. Hence, we accept a certain clustering if within each cluster the native norm of the radial basis function approximant is below $50/h$. This bound has been determined empirically to give adequate results. The reason we do include h here is that for smooth functions one can expect the norm to grow anti-proportional to an exponent of h. We do not know this exponent and, therefore, pick it to equal 1 which seems satisfactory in this case.

The algorithm including clustering compares favourably with direct IRKA for the synthetic example. A medium size model has not been computed here, as this system, having a rather small system size of 100, is constructed to show the difficulties that can occur when $f(p)$ is interpolated. The overall results for the synthetic problem are illustrated in Fig. 11. The error for the RBF-IRKA algorithm is displayed in a collection of three curves which reflects the fact three partitions obtained by clustering. The curve formed by the triangles does not converge for some parameters but does create significantly better reduced order models when it does. Along the parameter range a fine sampling is used to compute error curves, while a coarser sampling is used to set up the metamodel. For each sample location the starting value for IRKA is selected to be the result of IRKA at a nearby location. Hence, IRKA may converge during evaluation even if it does not return during metamodel creation. Indeed, the parameter region discussed above does only yield IRKA results in the evaluation phase.

10 Discussion

Regarding parametric linear time-invariant systems, we desire to decrease the computational effort to apply $\mathscr{H}_2$-optimal model order reduction by IRKA for a large number of individual realizations of the parameter. For that purpose, we discuss how radial basis function interpolation can be applied to approximate the system reduced by IRKA, given a number of IRKA runs for different parameter locations, a so-called sampling. The idea to relax $\mathscr{H}_2$-optimality

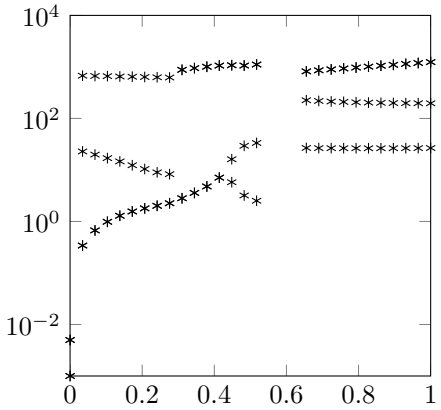

Figure 9: Real parts of the eigenvalues $\Sigma(p)$ for $r = 4$ and several p, including crossings and bifurcations. In most of the parameter domain, three different values are visible

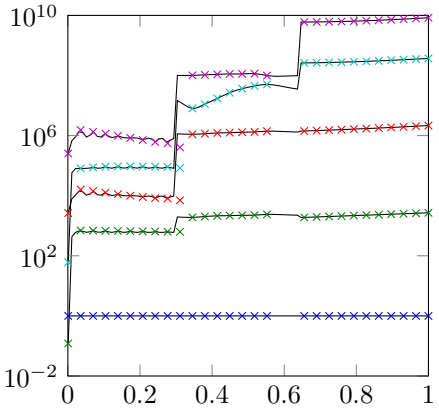

Figure 10: Coefficients $f(p)$ of the corresponding characteristic polynomial for $r = 4$ and several p (colored dots) and the interpolated coefficients $\tilde{f}(p)$ (black line)

to gain a speed-up is based on our hope to achieve an interpolation that violates $\mathscr{H}_2$-optimality only slightly in practical applications.

Previous publications have already shown that an interpolation of reduced matrices is viable (Amsallem and Farhat, 2008; Panzer et al., 2010). In constrast to these methods, our main suggestion is to use the offline phase sampling in order to approximate all eigenvalues, or poles, of the reduced-order system matrix instead. Here the approximation is based on a radial basis expansion of the coefficients of the corresponding polynomial. Our proposed algorithm is, moreover, designed to deal with discontinuities by a clustering approach involving the so-called native Hilbert space norm of the interpolant. In addition, we recommend to use the results from the sampling phase to construct a model of intermediate size and further speed up the set-up of our approximate reduced-order system.

We have extensively discussed the details of the approach, including basic assumptions we make, as well as possible difficulties that might occur. Three simpler practical examples and a more challenging synthetic problem are presented including numerical results and $\mathscr{H}_2$-error comparisons. These first tests encourage our hope that the computation of approximate solutions of a parametric linear time-invariant system can be accelerated without a significant loss of accuracy in $\mathscr{H}_2$.

The three practical examples provided are characterized by a low-dimensional parameter space, simple and smooth parameter dependencies and a lack of eigenvalue bifurcations of the reduced system. Eigenvalue crossings and splittings have indeed been explored with a synthetic problem. Examples including higher-dimensional parameter spaces as well as stronger

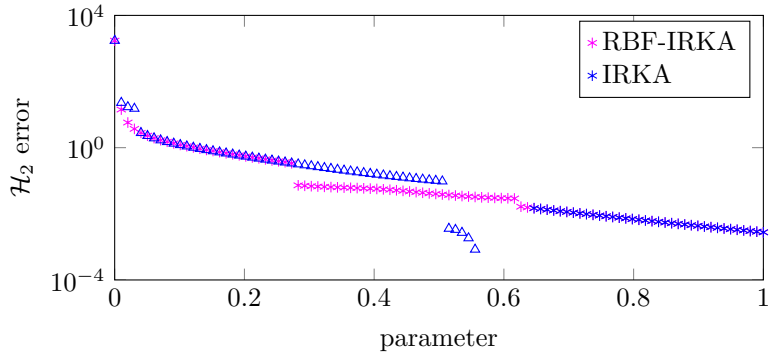

Figure 11: Error Comparison for Synthetic Model

and possibly local nonlinearities at different scales have been left for future work, though. It is also not unlikely to encounter non-synthetic problems that exhibit eigenvalue bifurcations, such as the Navier-Stokes equations with varying Reynold's number. An examination of the numerical performance, including error behaviour, of the presented method for such problems is much desirable. Further numerical investigations might focus on the case where the interpolated reduced system is unstable and has to be stabilized. The approach itself might also be further extended. As mentioned before, kernel approximation can incorporate finite-dimensional Hilbert spaces of functions such as polynomials of a certain degree. Known functional parametric dependencies of entries $A_{ij}(p)$ of the system matrix can thus be considered and reproduced explicitely by radial basis function interpolation. Finally, known error bounds of model order reduction and radial basis function interpolation might be combined in order to be able to compare the theoretical possibilities of the approach versus competitive methods.

Acknowledgements

Without Serkan Gugercin, Virginia Tech, and his idea of a medium model we would not have been able to create an efficient algorithm. Also very special thanks to Christian Rieger, University of Bonn, for helpful discussions and for the suggestion to exploit the native space norm in order to determine the number of cluster centres.

References

Amsallem, D. and Farhat, C. (2008). Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA J.*, 46(7):1803–1813.

Antoulas, A., Ionita, A., and Lefteriu, S. (2012). On two-variable rational interpolation. *Linear Algebra and its Applications*, 436(8):2889–2915.

Aronszajn, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68(3):337–404.

Baur, U., Beattie, C. A., Benner, P., and Gugercin, S. (2011a). Interpolatory projection methods for parameterized model reduction. *SIAM J. Sci. Comput.*, 33(5):2489–2518.

Baur, U., Benner, P., Greiner, A., Korvink, J., Lienemann, J., and Moosmann, C. (2011b). Parameter preserving model reduction for MEMS applications. *Mathematical and Computational Modelling of Dynamical Systems*, 17(4):297–317.

Benner, P., Gugercin, S., and Willcox, K. (2013). A survey of model reduction methods for parametric systems. Max Planck Institute Magdeburg Preprint MPIMD/13-14. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

Bunse-Gerstner, A., Kubalińska, D., Vossen, G., and Wilczek, D. (2010). h_2-norm optimal model reduction for large scale discrete dynamical MIMO systems. *J. Comput. Appl. Math.*, 233(5):1202–1216.

Fasshauer, G. E. (2007). *Meshfree Approximation Methods with MATLAB (with CD-ROM)*. World Scientific.

Grimme, E. J. (1997). *Krylov projection methods for model reduction*. Dissertation, Univ. of Illinois at Urbana-Champaign, USA.

Gugercin, S., Antoulas, A. C., and Beattie, C. (2008). $\mathscr{H}_2$ Model Reduction for large-scale dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638.

Meier, L., I. and Luenberger, D. (1967). Approximation of linear constant systems. *Automatic Control, IEEE Transactions on*, 12(5):585 –588.

Moosmann, C., Rudnyi, E., Greiner, A., Korvink, J., and Hornung, M. (2005). Parameter preserving model order reduction of a flow meter. *Technical Proceedings of Nanotech 2005*.

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856.

Panzer, H., Hubele, J., Eid, R., and Lohmann, B. (2009). Generating a parametric finite element model of a 3D cantilever Timoshenko beam using MATLAB. *Technical Reports on Automatic Control*.

Panzer, H., Mohring, J., Eid, R., and Lohmann, B. (2010). Parametric model order reduction by matrix interpolation. *at-Automatisierungstechnik*, 58(8):475–484.

Rieger, C., Schaback, R., and Zwicknagl, B. (2010). Sampling and stability. In Dæhlen, M., Floater, M., Lyche, T., Merrien, J.-L., Mørken, K., and Schumaker, L., editors, *Mathematical Methods for Curves and Surfaces*, volume 5862 of *Lect. Notes Comput. Sc.*, pages 347–369. Springer Berlin Heidelberg.

Schaback, R. (1999). Native Hilbert spaces for radial basis functions I. In Müller, M. W., Buhmann, M. D., Mache, D. H., and Felten, M., editors, *New Developments in Approximation Theory*, volume 132 of *ISNM Internat. Ser. Numer. Math.*, pages 255–282. Birkhäuser Basel.

Stewart, J. (1976). Positive definite functions and generalizations, an historical survey. *Rocky Mountain J. Math.*, 6(3):409–434.

van Dooren, P., Gallivan, K., and Absil, P.-A. (2008). $\mathscr{H}_2$-optimal model reduction of MIMO systems. *Appl. Math. Lett.*, 21(12):1267–1273.