Solving parametric PDEs with an enhanced model order reduction method based on Linear/Ridge expansions

Constantin Greif *, **Philipp Junk** **, **Karsten Urban** *

* *Ulm University, Institute for Numerical Mathematics, Helmholtzstr. 20, 89081 Ulm (Germany),*
{constantin.greif,karsten.urban}@uni-ulm.de
** *Justus-Liebig-University Giessen, Lehrstuhl Numerische Mathematik, Arndtstr. 2, 35392 Giessen (Germany),*
Philipp.Junk@math.uni-giessen.de

Abstract: Classical projection-based model order reduction methods, like the reduced basis method, are popular tools for getting efficiently solvable reduced order models for parametric PDEs. However, for some problems, the error-decay with respect to the dimension of the linear projection space is predetermined to be slow, e.g., for parameterized wave equations with jump discontinuities.
In order to cope with this issue, we consider approximations formed by a linear combination of given functions enhanced by ridge functions – a Linear/Ridge expansion. For an explicitly or implicitly solution of a parameter-dependent problem, we reformulate finding a best Linear/Ridge expansion in terms of an optimization problem that we solve with a particle grid algorithm.
The linear functions as well as the ridge profiles are built offline with a greedy-type algorithm. By training the directions offline, we can achieve an efficient online evaluation to solve the projected parametric PDE.

Keywords: Model reduction of distributed parameter systems, Subspace methods, Particle filtering/Monte Carlo methods, Parametric optimization, Evolutionary algorithms, Control of partial differential equations

1. MOTIVATION

As a motivating example for the developed method, we recall the parametric linear wave equation.

Parametric wave Consider the linear wave equation $\partial_{tt}^2 u - \mu^2 \partial_{yy}^2 u = 0$ for $t > 0$ and $y \in \mathbb{R}$ with initial conditions $u(0) = u_0$ and $\dot{u}(0) = 0$. The parameter-dependent solution is given by the famous d'Alembert formula as $u(t, y; \mu) = \frac{1}{2}(u_0(y - \mu t) + u_0(y + \mu t))$. Hence, choosing $v_1 = v_2 = u_0$, $c_1 = c_2 = \frac{1}{2}$, $b_1 = b_2 = 0$ as well as $a_1 = (-\mu, 1)^\top, a_2 = (\mu, 1)^\top$ and $x = (t, y)^\top$ yields a representation of the solution $u(t, y; \mu) = c_1 v_1(a_1^\top x + b_1) + c_2 v_2(a_2^\top x + b_2)$ as a sum of two ridge functions. Besides, also for $\dot{u}(0) \neq 0$, the wave equation is a sum of two, but then different, ridge functions.

This problem is particularly interesting since it is known that projection-based (i.e., linear) model order reduction techniques do not work in the sense that the decay of the Kolmogorov N-width is at most $\mathcal{O}(N^{-1/2})$, Greif and Urban (2019). Nevertheless, the use of ridge functions can enhance such error reduction and is in fact convenient to use here since we only need two appropriate ridge profiles.

2. LINEAR/RIDGE EXPANSIONS

We consider a given function $u : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$ is an open bounded domain and $u \in L_2(\Omega)$. In order to formulate the approximation problem under consideration, let $X_N := \operatorname{span}(\Phi_N) \subset L_2(\Omega)$, $\Phi_N := \{\varphi_1, ..., \varphi_N\}$ be a given linear space of dimension $N \in \mathbb{N}$ with φ_i, $i = 1, ..., N$, being given functions.

In addition to Φ_N, we assume that we are given a finite number $M \in \mathbb{N}$ of (ridge) profiles $\mathcal{V}_M := \{v_1, ..., v_M\} \subset L_2(\mathbb{R})$ and consider the approximation problem for $x \in \Omega$

$$u(x) \approx \sum_{i=1}^{N} \alpha_i \varphi_i(x) + \sum_{j=1}^{M} c_j v_j(a_j^\top x + b_j) =: u_\delta(x) \in U_{N,M},$$

where $U_{N,M}$ is the nonlinear space built with Φ_N and $\mathcal{V}_M$. The objective is to minimize the residual of the PDE, where $u = u(\cdot; \mu)$ is the implicit solution.

Given directions and offsets For fixed directions $a_j \in \mathbb{R}^d$ and offsets $b_j \in \mathbb{R}$, the coefficients $\alpha_i \in \mathbb{R}$ and $c_j \in \mathbb{R}$ are just given as the solution of a linear system of equations (Lemma 2.5 in Greif et al. (2022)). Therefore we reformulate the approximation to just search for the optimal $(a_j, b_j) \in \mathbb{R}^{d+1}$.

2.1 A particle grid algorithm

Since the determination of directions and offsets amounts to solving a complex optimization problem, we aim at using a well-known heuristic method, the particle swarm algorithm. In order to reduce computational complexity, we arrange our particles (which are associated to the collection of all directions $a_j \in \mathbb{R}^d$ and offsets $b_j \in \mathbb{R}$, $j = 1, ..., M$) in a dynamic grid. For each profile, we collect the direction and the offset in one vector $d_j := (a_j, b_j) \in \mathbb{R}^{d+1}$. These vectors are then associated to some component $p_j \in (-1, 1)^D =: \mathbb{S}^D$. The vector $(d_j)_{j=1,...,M} \in \mathbb{R}^{DM}$ of all directions and offsets is then associated to one particle $\mathbf{p} \in (-1, 1)^{DM} = \mathbb{S}^P$.

The algorithm produces a sequence of particle grids, where each grid (i.e., a swarm in form of a grid) $\mathbf{P}^{(k)}$ consists of m_{par} particles in $\mathbb{S}^P$. We choose n_{par} nodes in each dimension, i.e., $m_{\text{par}} = n_{\text{par}}^P$ for $n_{\text{par}} \in \mathbb{N}$. Then, we initialize the initial particle grid $\mathbf{P}^{(0)}$ by taking the tensor product, yielding a regular grid. Each particle has uniquely defined next neighbors in each diagonal direction. This next neighbor relation does not change in the course of the iteration. This means that each swarm is a grid whose internal geometry does not change even if the position of each particle may vary. We may associate each particle grid $\mathbf{P}^{(k)}$ with a tensor of dimension P (e.g., a matrix for $P = 2$).

Solve parametric PDEs We used the method with the particle grid algorithm to solve two different PDEs, the already introduced wave equation as well as the thermal block, that is a classical problem for model reduction, Haasdonk (2017). (For the thermal block, we used the domains $\Omega = (0, 1)^2, \overline{\Omega}_i := [0, 1] \times [\frac{i-1}{4}, \frac{i}{4}]$, $i = 1, ..., 4$). We fed the method with $\Phi_N = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ and $\mathcal{V}_M = \{v_1, v_2\}$ and the algorithm was able, for a new parameter μ, to choose the appropriate functions and discard the remaining functions by setting the coefficients to zero. The results can be seen in Table 1. Obviously there are more iterations needed for the case of the wave equation.

PPDE	parameter μ	no. iterat. K	L_2-error
Thermal block	$(0.1, 10, 1, 0.6)$	1	$5.1019e-15$
Thermal block	$(10, 2, 0.1, 0.5)$	1	$1.4446e-14$
Thermal block	$(0.4, 2, 0.3, 5)$	1	$6.0861e-15$
Wave	1/4	20	$7.6682e-05$
Wave	1/4	83	$8.9850e-16$
Wave	1	21	$8.2400e-05$
Wave	1	86	$3.1765e-16$
Wave	4	28	$4.3012e-05$
Wave	4	83	$8.9850e-16$

Table 1. Errors and iterations for both parametric PDEs and different parameter values.

3. COMPLETE MODEL REDUCTION METHOD

Generation of basis functions Until now, we considered Φ_N and $\mathcal{V}_M$ as given. However, we need to extract them from the problem. (Due to page limitation we can just sketch the idea here.) Using a greedy-type algorithm, we build a linear basis from snapshots and successively add ridge functions attained by integration along directions according to Pinkus (2015).

Efficient online computation By assuming certain affine parameter dependence, we can get an efficient online computation that is faster than the classical particle grid algorithm. Therefore we offline train the parameter-dependence on the directions using the particle grid algorithm and online use this as an initial guess to just evaluate a very local optimization.

3.1 Full model reduction method

- Offline: Build basis functions $X_N = \{\varphi_1, \ldots, \varphi_N\}$ and profiles $V_M = \{v_1, \ldots, v_M\}$.
- We first build X_N by a greedy method minimizing an residual error.
- If the error decay is not "fast" any more, we switch and add profiles to V_M to further reduce the error.
- With the obtained space $U_{N,M}$, we train the parameter-dependence of the directions using a training set of snapshots.
- Online: For a parameter μ find directions with a local optimization method by solving the PDE on the projected space $U_{N,M}$.

3.2 Conclusion

We introduced a model order reduction method that is able to solve a broader class of problems than classical projection-based methods. Unfortunately, the method also has some drawbacks, like the harder to reach online efficiency. Computing the optimal ridge directions is more costly than just solving a linear system of equation.
Furthermore, the Linear/Ridge expansions are still a restricted approximation type. One can easily think of function classes that are still hard to solve with approaches like this, like PDEs with nonlinear characteristics. But unlike in the linear case, there won't be a nonlinear method that fits all problems. Related approaches are an active field of research, c.f. Black et al. (2020).

REFERENCES

Black, F., Schulze, P., and Unger, B. (2020). Projection-based model reduction with dynamically transformed modes. *ESAIM: Math. Model. Numer. Anal.*, 54(6), 2011–2043.

Buhmann, M.D. and Pinkus, A. (1999). Identifying linear combinations of ridge functions. *Adv. Appl. Math.*, 22(1), 103–118.

Greif, C., Junk, P., and Urban, K. (2022). Linear/ridge expansions: Enhancing linear approximations by ridge functions. *Advances in Computational Mathematics*, 48, 15.

Greif, C. and Urban, K. (2019). Decay of the kolmogorov n-width for wave problems. *Appl. Math. Letters*, 96, 216 – 222.

Haasdonk, B. (2017). Reduced Basis Methods for Parametrized PDEs — A Tutorial. In P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), *Model Reduction and Approximation*, chapter 2, 65–136. SIAM, Philadelphia.

Ohlberger, M. and Rave, S. (2016). Reduced basis methods: Success, limitations and future challenges. *Proceedings of the Conference Algoritmy*, 1–12.

Pinkus, A. (2015). *Ridge Functions*. Cambridge University Press.