CONVERGENCE ACCELERATION FOR TIME-DEPENDENT PARAMETRIC MULTIFIDELITY MODELS[*]

VAHID KESHAVARZZADEH[†], ROBERT M. KIRBY[‡], AND AKIL NARAYAN[§]

Abstract. We present a numerical method for convergence acceleration for multifidelity models of parameterized ordinary differential equations. The hierarchy of models is defined as trajectories computed using different timesteps in a time integration scheme. Our first contribution is in novel analysis of the multifidelity procedure, providing a convergence estimate. Our second contribution is development of a three-step algorithm that uses multifidelity surrogates to accelerate convergence: step one uses a multifidelity procedure at three levels to obtain accurate predictions using inexpensive (large timestep) models. Step two uses high-order splines to construct continuous trajectories over time. Finally, step three combines spline predictions at three levels to infer an order of convergence and compute a sequence transformation prediction (in particular we use Richardson extrapolation) that achieves superior error. We demonstrate our procedure on linear and nonlinear systems of parameterized ordinary differential equations.

Key words. multifidelity algorithms, time-stepping schemes, convergence acceleration

AMS subject classifications. 65L99, 65B05

DOI. 10.1137/18M1170339

1. Introduction. We investigate time-dependent models arising from parameterized ordinary differential equations (ODE). Such models arise in, for example, applied uncertainty quantification contexts. The following parameterized ODE defines the unknown u:

$$(1.1) \qquad \frac{\mathrm{d}u}{\mathrm{d}t}(t,k) = f(t,u,k), \qquad\qquad u(0,k) = u_0(k),$$

where $u \in \mathbb{R}^M$ is a vector-valued state variable, $u_0 \in \mathbb{R}^M$ is a given initial condition, $k \in \mathbb{R}^d$ is a Euclidean parameter, and we take the time variable to range over $[0,T]$. The right-hand side function $f : [0,T] \times \mathbb{R}^M \times \mathbb{R}^d$ is also given. We assume the above system is well-posed for all k; in particular, we will codify some assumptions in section 2 so that the solution trajectory $u(\cdot,k)$ is smooth and so that standard discrete-time integration methods (e.g., multistep and multistage methods) provide convergent approximations for fixed T.

[†]Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112 (vkeshava@sci.utah.edu, https://sites.google.com/view/vahid-keshavarzzadeh).

[‡]School of Computing, University of Utah, Salt Lake City, UT 84112 (kirby@sci.utah.edu, https://www.cs.utah.edu/~kirby/).

[§]Department of Mathematics, University of Utah, Salt Lake City, UT 84112 (akil@sci.utah.edu, http://www.sci.utah.edu/~akil).

The technique we adopt was proposed in [21, 28] and begins with the following approximation:

$$(1.2) \qquad u(t,k) \approx \sum_{q=1}^{n} u(t,k_q) v_q(k),$$

where n is small (in practice we use $n = \mathcal{O}(10)$), the $u(t,k_q)$ are discrete-time solution "snapshots" at fixed parameter values computed using a refined timestep, and v_q are computed from a coarse timestep approximation. The approximation above requires n stored solutions computed using a refined timestep, and a single coarse timestep solution for each value of k. The parameter values k_q and the parametric functions v_q are computed via an analysis of coarse time discretizations. Thus, the entire procedure uses time discretizations at different discrete-time refinements ("fidelities"). Once the n solutions $u(t,k_q)$ are stored, then evaluation of (1.2) at a particular k requires only one solution of the coarse timestep model.

Assuming solution trajectories are smooth, we supplement the multifidelity procedure above with two additional steps: Once the approximation above is constructed, we extend the discrete time solutions to continuous time via spline interpolation, and with spline representations on hand for each fidelity level we perform sequence transformations (e.g., Richardson extrapolation) to accelerate convergence.

Our novel contributions are the derivation of mathematical error estimates that prove convergence of the approximation (1.2), and in development of computational algorithms that utilize spline representations and sequence transformations to accelerate convergence. An overview of the algorithm and our theoretical statements is provided below.

1.1. Multifidelity algorithm overview. We compute the coefficient functions v_q in (1.2) via a multifidelity procedure. Our models of different fidelities are outputs from discrete-time integration methods using different timesteps. Let $r > 1$ be an integer, and let $h > 0$ be a stepsize at the coarsest level. We construct three discrete models, defined as

- $u_1(\cdot,k)$: Discrete-time solution computed using a timestep h, a "low-fidelity" model. u_1 is inexpensive to compute for each k.
- $u_2(\cdot,k)$: Discrete-time solution computed using a timestep h/r, a "medium-fidelity" model. u_2 is moderately expensive to compute for each k.
- $u_3(\cdot,k)$: Discrete-time solution computed using a timestep h/r^2, a "high-fidelity" model. u_3 is expensive to compute for each k.

Our procedure performs an analysis of several trajectories of the inexpensive model u_1 to (i) identify the parameter values $\{k_q\}_{q=1}^{n}$ and (ii) compute the coefficient functions v_q for use in (1.2). Precisely, the v_q are defined as

$$\{v_1(k), \ldots, v_q(k)\} = \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^q} \left\| u_1(\cdot,k) - \sum_{j=1}^{q} w_j u_1(\cdot,k_j) \right\|,$$

where $\| \cdot \|$ is an appropriate ℓ^2-type norm so that the v_j can be easily computed as the solution to a linear least-squares problem given the data $u_1(t,k)$.[1] The values

---

[1] The values $\{v_1(k), \ldots, v_q(k)\}$ depend on the value of q, and so we are committing a small notational crime by not explicitly indexing the $v_j(k)$ by q.

$k_1, \ldots, k_n$ are sequentially chosen via the optimization

$$(1.3) \qquad k_{q+1} = \operatorname*{argmax}_{k} \left\| u_1(\cdot, k) - \sum_{j=1}^{q} v_j(k) u_1(\cdot, k_j) \right\|.$$

Computationally, the argmax is evaluated over a finite training set instead of a continuum. Since the above is an ℓ^2-residual, in practice the solution to this greedy optimization problem on the finite training set is given by ordered pivots of a Cholesky or QR matrix factorization. (For the QR factorization, each column of the input matrix contains a snapshot.) Once the k_q values have been computed, n relatively expensive solution trajectories $u_2(t, k_q)$ and $u_3(t, k_q)$ are constructed, and the approximations

$$\widehat{u}_2(t, k) := \sum_{q=1}^{n} u_2(t, k_q) v_q(k), \qquad\qquad \widehat{u}_3(t, k) := \sum_{q=1}^{n} u_3(t, k_q) v_q(k)$$

are built. Evaluation of v_q at a fixed k requires computation of the inexpensive model $u^L(t, k)$. The approximation above allows construction of $\widehat{u}^H(k)$ on the high-fidelity grid using computations on the low-fidelity grid for every value of k. We require only a one-time investment of n solutions of u^H. When n is small and $\widehat{u}^H$ is accurate, this can result in significant computational savings when analyzing the behavior of the family of solutions $u(\cdot, k)$ over the relevant range of k.

The multifidelity strategy described above is essentially a model reduction approach, using a small number of accurate but expensive snapshots to predict global parametric behavior. A well-known approach that uses the same basic idea of snapshot-based approximation is the reduced basis method [18, 12, 23], and numerous related model reduction approaches exist [13, 7, 2, 3]. The method we propose in this paper is an additional approach to model reduction and is a snapshot-based approximation technique, but differs from the reduced basis method in practice. Our approach does not circumvent some of the more important challenges in this field such as stability or conservation for model reduction in nonlinear hyperbolic systems.

1.2. Main contributions. Our first contribution is the derivation of the error estimate

$$(1.4) \qquad \| \mathcal{P}_n u(\cdot, k) - \widehat{u}_j(\cdot, k) \|_{\infty} \lesssim C_1 h^p + C_2 (h/r^{j-1})^p,$$

where p is the global truncation order of the discrete time integration method used to compute u_j, and $\mathcal{P}_n$ is a projection operator onto $\operatorname{span}\{u(\cdot, k_q)\}_{q=1}^{n}$. The precise statement is given by Theorem 3.3.

Our second contribution computationally effects convergence acceleration. The j-dependence in the error estimates above suggest that sequence transformation may be effective in accelerating convergence by eliminating the j-dependent error term. We would like to perform such an extrapolative transform at each instance of time, but the difficulty is that u_2 and u_3 "live" on different grids. To rectify this situation, we perform spline approximations on each level, with the order of the spline matching p, the time integration order. The spline approximations then allow pointwise (in time) sequence transformation. We show that this strategy for convergence acceleration can be effective. The spline procedures and sequence transformation/extrapolation procedure are visually summarized in Figure 1. We observe in our examples that we can obtain h^{p+1}-order convergence in the accelerated solution despite the theoretical presence of j-independent terms in the estimate (1.4). This suggests that $C_1 \ll C_2$ can happen in practice.
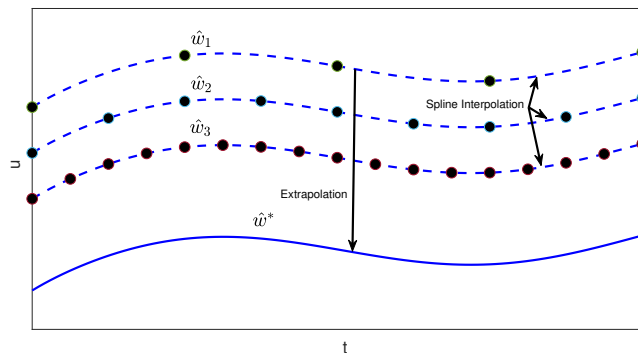
FIG. 1. *Schematic representation for convergence acceleration of time-dependent multifidelity models.*

TABLE 1
Notation used throughout this article.

t, T	Time variable t taking values in $[0, T]$
k, $\mathcal{K}$	Parameter value k taking values in $\mathcal{K} \subset \mathbb{R}^d$
M	Dimension of vector-valued solutions to an ODE
$u(t, k)$	$\mathbb{R}^M$-valued solution to a k-parameterized ODE at time t; the trajectory satisfies $u(\cdot, k) \in H$
H	Hilbert space H containing solution trajectories, $u(\cdot, k) \in H$
h, N	Coarse timestep h with $T = Nh$
$[N]$	The set $\{0, 1, \ldots, N\}$ serving as indices for discrete times
r, h_j, N_j	Integer $r \geq 2$ defining timestep $h_j = h/r^{j-1}$ for "level" j approximation, using $N_j = Nr^{j-1}$ equidistant timesteps to reach T
H_j	Hilbert space containing h_j-discretized solution trajectories
$u_j(i, k)$	$\mathbb{R}^M$-valued discrete solution at time $t = ih_j$ computed using an integration method with timestep h_j with $u_j(k) = u_j(\cdot, k) \in H_j$
p, P	Time integration global truncation error order p and $(P+1)$-point Newton–Cotes quadrature rule
$\mathcal{K}_n$	Collection of n points in $\mathcal{K}$
$\boldsymbol{G}$, $\boldsymbol{G}_j$	$n \times n$ Gramian matrices formed from solution trajectories for $k \in \mathcal{K}_n$
$\mathcal{V}$, $\mathcal{V}_j$	Manifold of solutions for all $k \in \mathcal{K}$; subsets of H and H_j, respectively.

The paper is organized as follows. Section 2 introduces notation and the models of differing fidelities. Section 3 describes the mathematical multifidelity procedure and contains our main error estimate, Theorem 3.3. Section 4 discusses convergence acceleration using spline interpolants and sequence transformations. Section 4.3 summarizes the entire algorithm. Finally, section 5 presents numerical examples for linear and nonlinear ODEs including convergence acceleration for the transient response of a parameterized linear elastic structure.

2. Notation and setup.

2.1. Parameterized ODE solutions. We refer to Table 1 for a summary of much of the notation in this article. The parameterized ODE (1.1), where $u(t, k) \in \mathbb{R}^M$, depends on the parameters $k \in \mathcal{K} \subset \mathbb{R}^d$. We use $u^{(m)}$, $m = 1, \ldots, M$, to denote

the components of u. We assume $\mathcal{K}$ is a compact set in $\mathbb{R}^d$ and that, given some terminal time $T > 0$, then the trajectory $u(\cdot, k)$ is smooth uniformly in k.

Assumption 2.1. The solution trajectories $u(\cdot, k)$ exist and are unique on $[0, T]$ for each $k \in \mathcal{K}$. Furthermore, the function $f(t, u)$ is smooth enough so that for some integer $p \geq 2$,

$$(2.1) \qquad \max_{k \in \mathcal{K}} \max_{1 \leq m \leq M} \sup_{t \in [0,T]} \left| \frac{\partial^s}{\partial t^s} u^{(m)}(t, k) \right| =: U^{(s)} < \infty, \qquad 0 \leq s \leq p.$$

The above is relatively restrictive, requiring smoothness (up to order p) of the solution trajectories, with derivative bounds independent of k. The value of p required is the convergence order of a time integration scheme. The condition (2.1) allows us to conclude that the solution $u(\cdot, k) : [0, T] \to \mathbb{R}^M$ to (1.1) is at least continuously differentiable on the compact interval $[0, T]$. Therefore, we have

$$u(\cdot, k) \in H, \qquad H := L^2\left([0, T]; \mathbb{R}^M\right) = \left\{ v : [0, T] \to \mathbb{R}^M \;\middle|\; \|u\|_H < \infty \right\},$$

with the inner product and norm

$$\langle u, v \rangle = \frac{1}{MT} \sum_{m=1}^{M} \int_0^T u^{(m)}(t) v^{(m)}(t) \mathrm{d}t, \qquad \|u\|^2 = \langle u, u \rangle,$$

where $u^{(m)}$ and $v^{(m)}$ are the components of the M-vectors u and v, respectively. Note that we normalize the inner product by $1/(MT)$.

We are interested in computing approximations to the family of solutions

$$\mathcal{V} = \left\{ u(\cdot, k) \;\middle|\; k \in \mathcal{K} \right\} \subset H.$$

More precisely, given $k \in \mathcal{K}$, we wish to construct an efficient and accurate approximation to the solution map $k \mapsto u(\cdot, k)$.

We require one additional assumption on the function f, namely, that it is continuous in k.

Assumption 2.2. For every $t \in [0, T]$ and $u \in \mathbb{R}^M$, the function $f(t, u, k)$ is k-continuous for $k \in \mathcal{K}$. Also, the initial data $u_0(k)$ is continuous for each $k \in \mathcal{K}$.

2.2. Time integration. We assume we have a stable and convergent numerical method to compute solutions to (1.1) for all $k \in \mathcal{K}$ over $t \in [0, T]$ that uses $N \in \mathbb{N}$ timesteps to reach $t = T$ (e.g., we assume N is large enough for stability of explicit time integration methods uniformly in k.) Thus, define $h := T/N$ as the timestep size, and let h_j, $j \in \mathbb{N}$, be a geometric sequence of timestep sizes

$$(2.2) \qquad h_j = \frac{h}{r^{j-1}}, \qquad\qquad N_j := N r^{j-1},$$

where $r \geq 2$ is an integer. We will use the solutions computed with timesteps h_j (i.e., N_j total timesteps) as our models of different fidelity. Suppose the rate of convergence of our time integration method is $p \geq 1$, and let $u_j(i, k) \in \mathbb{R}^M$ be the discrete solution at discrete time ih_j ($i = 0, \ldots, N_j$) computed using the time integration method with a timestep of h_j.

Let $[N] := \{0, 1, \ldots, N\}$. An accurate time integration method produces vectors $u_j(i, k) \in \mathbb{R}^M$ satisfying

$$u_j(i, k) \approx u(ih_j, k), \qquad\qquad i \in [N_j], \quad k \in \mathcal{K}.$$

We will primarily be interested in the values $j = 1, 2,$ and 3, representing a three-level hierarchy of solutions. We emphasize that, fixing k, the exact ODE solution $u(\cdot, k)$ is a function whose domain is the continuum $[0, T]$, but the discrete solution $u_j(\cdot, k)$ is a function whose domain is the finite set of indices $[N_j]$. The following is a standard estimate for the global truncation error committed by a time integration method of global order p when applied to ODEs with smooth coefficients.

LEMMA 2.3. *Let $u_j^{(m)}(i, k)$, $k \in \mathcal{K}$, $i \in [N_j]$, be the mth component at time index i of the discrete solution computed using an order-p time integration method with timestep h_j. Under Assumption 2.1 with smoothness order p, then for h_j small enough,*

$$\max_{m=1,\ldots,M} \max_{i=0,\ldots,Nr^{j-1}} \left| u^{(m)}(ih_j, k) - u_j^{(m)}(i, k) \right| \le C(T, L) h_j^p = \frac{C(T, L)}{r^{j-1}} h^p,$$

where $C(T, L)$ usually depends exponentially both on T and bounds on derivatives of f.

Note that $C(T, L)$ does not depend on k due to Assumption 2.1. We can define discretized Hilbert spaces H_j that contain the discrete-time solutions for each $k \in \mathcal{K}$:

$$(2.3) \qquad H_j = \ell^2\left([N_j]; \mathbb{R}^M\right), \qquad \langle u, v \rangle_j = \sum_{i=[N_j]} w_{j,i} v(i)^T u(i), \qquad u(i) \in \mathbb{R}^M,$$

where w_j is a vector of $N_j + 1$ weights. We assume the w_j weight vectors are positive for each j and that the entries sum to 1 to reflect the $1/(MT)$ normalization in (2.3). We will make precise choices for these weights in the next section. The discrete solutions induce discretized versions of the compact manifold $\mathcal{V}$:

$$\mathcal{V}_j = \left\{ u_j(\cdot, k) \mid k \in \mathcal{K} \right\} \subset H_j.$$

The convergence in Lemma 2.3 also implies that the discrete solution manifolds have bounded elements. In particular, since $\mathcal{K}$ is compact and $k \mapsto u(\cdot, k)$ is continuous by Assumption 2.2, we have

$$(2.4) \quad U_j = \max_{k \in \mathcal{K}} \|u_j(\cdot, k)\|_j < \infty \quad \implies \quad U := \max\left\{ \max_{k \in \mathcal{K}} \|u(\cdot, k)\|, \ \max_{j \ge 1} U_j \right\} < \infty,$$

where $\{U_j\}_{j \ge 1}$ is a positive and convergent (hence bounded) sequence by Lemma 2.3.

2.3. Norms and inner products on H_j. The discussion at the beginning of section 2.2 constructs the functions $u_j(k)$ so that they represent approximations to the exact solution trajectory $u(\cdot, k)$ evaluated on an equispaced grid. For our procedure, we require the ability to approximate inner products on H using this discrete grid up to the order of accuracy p of the time integration scheme. For this purpose, we turn to a composite Newton–Cotes quadrature rule. A $(P+1)$-point closed Newton–Cotes rule on the interval $[a, b]$ has the form

$$(2.5) \qquad \int_a^b f(x)\mathrm{d}x \approx \sum_{s=0}^{P} \widetilde{w}_s f(x_s), \qquad\qquad x_s = a + (b-a)\frac{s}{P},$$

with known, explicitly computable weights $\widetilde{w}_j$. For $P + 1 \le 7$, the weights are all positive. For a function g whose $(q-1)$th derivative is bounded on $[a, b]$, the rule has

accuracy given by

$$(2.6) \qquad \left| \int_a^b g(x)\mathrm{d}x - \sum_{s=0}^{P} \widetilde{w}_s x_s \right| = \mathcal{O}((b-a)^q), \qquad q = q(P) = 2 \left\lfloor \frac{P}{2} \right\rfloor + 3.$$

Now set $q = p + 1$ and assume the order-p smoothness as stated in Assumption 2.1. Choose $P(p)$ as

$$(2.7) \qquad P(p) = \max \left\{ 2 \left\lfloor \frac{p-1}{2} \right\rfloor, 1 \right\}.$$

Our choice of $P(p)$ above is the smallest P satisfying $q(P) = p + 1$, and thus with this choice integrating a solution trajectory under Assumption 2.1 achieves order of accuracy $p + 1$ on individual subintervals of $[0, T]$ so that the composite rule has order-p accuracy.

We can now define the weights $w_{j,i}$ defining the inner product on H_j. Assume that the number of coarse-level timesteps, N, is divisible by $P(p)$. Then for any $j \geq 1$ the interval $[0, T]$ can be divided up into N_j/P subintervals, and a composite Newton–Cotes rule over $[0, T]$ acting on a function $v \in H$

$$(2.8) \quad Q_j[v] := \sum_{i=1}^{N_j/P} \sum_{s=0}^{P} \widetilde{w}_{j,i,s} v(((i-1)P + s + 1)h_j) := \sum_{i \in [N_j]} w_{j,i} v_i, \quad \sum_{i \in [N_j]} w_{j,i} = 1,$$

where $\{\widetilde{w}_{j,i,s}\}_{s=0}^{P}$ are the weights $\{\widetilde{w}_s\}_{s=0}^{P}$ in (2.5) rescaled for the ith subinterval of $[0, T]$. The condition that the weights $w_{j,i}$ sum to 1 is required for consistency of the H_j discrete inner products with respect to the H continuous inner product. Assuming $P + 1 \leq 7$, we use the vector of positive weights $w_j \in \mathbb{R}^{N_j+1}$ in this composite rule to define the norm and inner product on H_j via the expression (2.3).

The case $P + 1 > 7$ only becomes relevant when we are using a time integration method with order p equal to 7 or greater; this situation rarely happens in practice, so we hereafter assume $p \leq 6$ and $P + 1 \leq 7$. This quadrature rule has order of accuracy p over the whole interval, which we codify below for the special case that we require.

LEMMA 2.4 (composite Newton–Cotes accuracy). *Let $u_{(j)}(\cdot, k) \in H_j$ be a sampling at the timesteps for level j of an element in the solution $u(\cdot, k)$, i.e.,*

$$(2.9) \qquad u_{(j)}(i, k) = u(ih_j, k), \qquad\qquad i \in [N_j], \quad k \in \mathcal{K}.$$

Given $p \geq 1$, if P is chosen as in (2.7), then under the conditions of Assumption 2.1,

$$(2.10) \qquad \left| \langle u(\cdot, k), u(\cdot, k') \rangle - \langle u_{(j)}(\cdot, k) u_{(j)}(\cdot, k') \rangle_j \right| \leq C p^p h_j^p, \qquad k, k' \in \mathcal{K},$$

where C is independent of k and k'.

Proof. Each individual Newton–Cotes rule spans a normalized interval of length Ph_j and is accurate to order $p + 1$. Relation (2.7) implies that $P \leq p - 1$. Under Assumption 2.1, the integrand $u(\cdot, k)u(\cdot, k')$ has bounded derivatives of order $p = q - 1$ on $[0, T]$. Thus for one component of the integrand, the Newton–Cotes rule commits an error scaling like

$$\sum_{i=1}^{N_j/P} (Ph_j)^{p+1}/(MT) = \frac{1}{T} N_j h_j (Ph_j)^p \leq \frac{1}{M} p^p h_j^p,$$

where we have used (2.7) to conclude that $P(p) \leq \max\{p-1, 1\} \leq p$, and the $1/(MT)$ factor arises because of the $1/(MT)$ normalization for the weights in (2.8). Summing over the M components results in the estimate (2.10). The constant C appearing in the conclusion is a k-independent bound on the order-p derivatives of the integrands, for which a loose bound is

$$\left| \frac{\partial^p}{\partial t^p} \left(u^{(m)}(t,k) u^{(m)}(t,k) \right) \right| \overset{(2.1)}{\leq} \sum_{r=0}^{p} \binom{p}{r} U^{(r)} U^{(p-r)}. \qquad \Box$$

3. Time-dependent multifidelity approximations. The analytical result in this section is our first novel contribution: a proof that the multifidelity approximations $\widehat{u}_j$ (formally defined in this section) converge to an appropriate quantity as $h \to 0$. Our major result stating this is Theorem 3.3.

3.1. Projection approximations. Let $\mathcal{K}_n \subset \mathcal{K}$ be a set of $n \in \mathbb{N}$ points in parameter space:

$$\mathcal{K}_n = \{k_1, \ldots, k_n\} \subset \mathcal{K}.$$

For a fixed n, we define subspaces spanned by $u(\cdot, \mathcal{K}_n)$ and $u_j(\cdot, \mathcal{K}_n)$,

$$\mathcal{W}_n = \mathrm{span}\,\{u(k_1), \ldots, u(k_n)\} \subset H, \quad \mathcal{W}_{n,j} = \mathrm{span}\,\{u_j(k_1), \ldots, u_j(k_n)\} \subset H_j.$$

We now define $\mathcal{P}_n$ and $\mathcal{P}_{n,j}$ as the orthogonal projectors onto $\mathcal{W}_n$ and $\mathcal{W}_{n,j}$, respectively:

$$\mathcal{P}_n : H \to \mathcal{W}_n, \qquad\qquad \mathcal{P}_{n,j} : H_j \to \mathcal{W}_{n,j}.$$

We will show that the multifidelity approximation that we form converges to $P_n u$. The approximation error committed by projecting the solution set $\mathcal{V}$ onto the subspace $\mathcal{W}_n$ is

$$(3.1) \qquad e_n(\mathcal{V}) := e(\mathcal{V}, \mathcal{W}_n) = \sup_{v \in \mathcal{V}} \|v - \mathcal{P}_n v\| = \sup_{k \in \mathcal{K}} \|u(k) - \mathcal{P}_n u(k)\|.$$

The best possible error in approximating the true solution space $\mathcal{V}$ by an n-dimensional projection is

$$d_n(\mathcal{V}) = \inf_{\mathcal{W} \subset H,\, \dim W = n} e(\mathcal{V}, \mathcal{W}).$$

The above is formulated on the continuous space H, which is not directly computable since the exact solutions $u(\cdot, k)$ are usually not available. Similar quantities can be defined to measure the error committed on the discrete level, e.g., the H_j-error committed by approximating $\mathcal{V}_j$ with $\mathcal{W}_{n,j}$.

On the discrete spaces H_j, one way to construct a sequence $k_1, k_2, \ldots$ for which $\mathcal{W}_{n,j}$ well-approximates $\mathcal{V}_j$ is by greedy procedure, in particular given $k_1, \ldots, k_n$, by picking k_{n+1} as the k value that maximizes a discrete version of the supremum argument in (3.1). Optimization on this discrete level can provide errors similar to optimizing over the continuous level. Below we cite a sufficient condition on the $j = 1$ level.

LEMMA 3.1 (see [8]). *Suppose the parameter values $k_1, k_2, \ldots$ are chosen via the greedy procedure*

$$(3.2) \qquad k_{q+1} = \operatorname*{argmax}_{k \in \mathcal{K}} \|u_1(k) - \mathcal{P}_{q,1} u_1(k)\|_1, \qquad\qquad q \geq 0,$$

with $\mathcal{P}_{0,1}$ the zero operator. If there is a positive constant $\gamma > 0$ such that

$$(3.3) \qquad \frac{\max_{k \in \mathcal{K}} \|u_1(k) - \mathcal{P}_{q,1} u_1(k)\|_1}{\max_{k \in \mathcal{K}} \|u(k) - \mathcal{P}_q u(k)\|_1} \geq \gamma > 0, \qquad\qquad q \geq 0,$$

then

$$e_{2q}(\mathcal{V}) \leq \frac{\sqrt{2}}{\gamma} \sqrt{d_q(\mathcal{V})}, \qquad\qquad q \geq 1.$$

See also [4] for related estimates. The condition (3.3) ensures that the sequence k_q is generated via a weak greedy algorithm. It is difficult in general to verify the assumption (3.3). However this is required for many computational model reduction methods that utilize snapshots, e.g., the reduced basis method [20], to prove convergence via the strategy in [8]. Note that (3.2) is exactly the choice we make in (1.3) for the multifidelity approximation. The optimization (3.2) above is stated as optimization over the continuum $\mathcal{K}$. In practice optimization is performed on a discretization of $\mathcal{K}$.

Our computations use the discrete projection operators $\mathcal{P}_{n,j}$, so our focus turns now to them. Given $k \in \mathcal{K}$, the projection $\mathcal{P}_{n,j} u_j(k)$ can be written as

$$(3.4) \qquad \mathcal{P}_{n,j} u_j(k) = \sum_{q=1}^{n} v_{j,q}(k) u_j(k_q), \qquad \boldsymbol{v}_j = \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^n} \left\| u_j(k) - \sum_{q=1}^{n} w_q u_j(k_q) \right\|_j.$$

The unknown vector of coefficients $\boldsymbol{v}_j \in \mathbb{R}^n$ can be computed via the normal equations. To state this, we introduce the kernel functions for the continuous solutions $u(\cdot, k)$, and for the discrete solutions $u_j(\cdot, k)$, $j = 1, 2 \ldots$,

$$K : \mathcal{K} \times \mathcal{K} \to \mathbb{R}, \qquad\qquad K_j : \mathcal{K} \times \mathcal{K} \to \mathbb{R},$$
$$K(k, k') = \langle u(k), u(k') \rangle, \qquad\qquad K_j(k, k') = \langle u_j(k), u_j(k') \rangle_j.$$

Then the normal equations formulation of (3.4) is

$$(3.5) \qquad \boldsymbol{G}_j \boldsymbol{v}_j = \boldsymbol{f}_j(k), \qquad (G_j)_{p,q} = K_j(k_q, k_p), \qquad (f_j)_p(k) = K_j(k, k_p)$$

for $p, q = 1, \ldots, n$. For future use, we similarly define the $n \times n$ matrix $\boldsymbol{G}$ and $n \times 1$ vector $\boldsymbol{f}(k)$ as containing inner products between the two exact solutions:

$$(3.6) \qquad (G)_{p,q} = K(k_q, k_p), \qquad\qquad (f)_p(k) = K(k, k_p).$$

The error relation (2.4) allows us to conclude that evaluations of K_j and K are proximal.

LEMMA 3.2. *Under Assumption 2.1, then for any $k, k' \in \mathcal{K}$,*

$$\sup_{k, k' \in \mathcal{K}} |K_j(k, k') - K(k, k')| \leq C(T, L, U, p) h_j^p.$$

Proof. Let $u_{(j)}(\cdot, k) \in H_j$ be the h_j-sampling of the solution $u(\cdot, k)$ as defined in (2.9). Then define the new kernel function

$$K_{(j)}(k, k') = \langle u_{(j)}(k), u_{(j)}(k') \rangle_j.$$

We have

$$(3.7) \quad |K(k,k') - K_j(k,k')| \leq \left|K(k,k') - K_{(j)}(k,k')\right| + \left|K_{(j)}(k,k') - K_j(k,k')\right|.$$

We show that each term on the right-hand side above scales like h_j^p. We can immediately bound the first term,

$$(3.8) \qquad\qquad \left|K(k,k') - K_{(j)}(k,k')\right| \overset{\text{Lemma 2.4}}{\leq} C(p)h_j^p.$$

For the second term, define

$$e_j(k) := u_{(j)}(k) - u_j(k) \in H_j.$$

By Lemma 2.3 and the weight summation condition (2.8),

$$(3.9) \qquad\qquad\qquad\qquad \|e_j(k)\|_j \leq Ch_j^p.$$

Then the triangle and Cauchy–Schwarz inequalities yield

$$\left|K_{(j)}(k,k') - K_j(k,k')\right| \leq \left|\langle u_{(j)}(k), e_j(k')\rangle_j\right| + \left|\langle e_j(k), u_j(k')\rangle_j\right|$$

$$(3.10) \qquad\qquad \overset{(3.9),(2.4)}{\leq} C_1 U h_j^p + C_2 U h_j^p \leq C_3(U,T,L)h_j^p.$$

Using (3.10) and (3.8) in (3.7) proves the result. $\qquad\qquad\qquad\qquad\qquad\square$

3.2. Multifidelity approximations. Our multifidelity approximations $\widehat{u}_j$ are defined by the linear least-squares solution $\boldsymbol{v}_1$ to the $j = 1$ version of (3.4). These approximations are, respectively,

$$(3.11) \qquad\qquad \widehat{u}_j(\cdot, k) = \sum_{q=1}^n v_{1,q}(k)u_j(\cdot, k_q), \qquad\qquad j \geq 1.$$

The "ideal" function that $\widehat{u}_j(k)$ represents is the H-projection approximation $P_n u(\cdot, k)$. The pointwise proximity of these two functions on the h_j grid is of the order h^p uniformly in k.

THEOREM 3.3. *Fix $n \in \mathbb{N}$, and assume that $\sigma_{\min}(\boldsymbol{G}) > 0$, where $\sigma_{\min}$ is the smallest singular value of $\boldsymbol{G}$. Let $w(\cdot, k) := P_n u(\cdot, k) \in H$. Then there exists $\overline{h} > 0$ such that for all $h < \overline{h}$,*

$$(3.12) \qquad \max_{k \in \mathcal{K}} \max_{m=1,\ldots,M} \max_{i \in [N_j]} \left|w^{(m)}(ih_j, k) - \widehat{u}_j^{(m)}(i, k)\right| \leq C_1 h^p + C_2 h_j^p,$$

where

$$C_1(T, L, p, U, n, \boldsymbol{G}) = C(T, L, p)\frac{U^3 n^2}{\sigma_{\min}^2(\boldsymbol{G})}, \quad C_2(T, L, p, U, n, \boldsymbol{G}) = C(T, L, p)\frac{U^2 n}{\sigma_{\min}(\boldsymbol{G})}.$$

Proof. By Lemma 3.2, then $\|\boldsymbol{G}_1 - \boldsymbol{G}\| \to 0$ as $h \downarrow 0$. Choose $\overline{h}$ such that

$$(3.13) \qquad\qquad\qquad \|\boldsymbol{G}_1 - \boldsymbol{G}\| < \frac{1}{2}\sigma_{\min}(\boldsymbol{G})$$

for all $h < \overline{h}$. Define the vectors $\boldsymbol{v}$ and $\boldsymbol{v}_1$ as solutions to the systems

$$\boldsymbol{G}\boldsymbol{v} = \boldsymbol{f}(k), \qquad\qquad\qquad \boldsymbol{G}_1\boldsymbol{v}_1 = \boldsymbol{f}_1(k),$$

where the vectors $\boldsymbol{f}_1$, $\boldsymbol{f}$ and matrices $\boldsymbol{G}$, $\boldsymbol{G}_1$ are defined in (3.5) and (3.6). Both $\boldsymbol{G}$ and $\boldsymbol{G}_1$ are invertible since $\sigma_{\min}(\boldsymbol{G}) > 0$ and due to (3.13). Now note that

$$\left| w^{(m)}(ih_j, k) - \widehat{u}_j^{(m)}(i, k) \right| = \left| \sum_{q=1}^{n} v_q(k) u^{(m)}(ih_j, k_q) - \sum_{q=1}^{n} v_{1,q}(k) u_j^{(m)}(i, k_q) \right|$$

$$\leq \left| \sum_{q=1}^{n} \left(v_q(k) - v_{1,q}(k) \right) u^{(m)}(ih_j, k_q) \right|$$

$$+ \left| \sum_{q=1}^{n} v_{1,q}(k) \left(u^{(m)}(ih_j, k_q) - u_j^{(m)}(i, k_q) \right) \right|$$

$$(3.14) \qquad \leq U\sqrt{n}\,\|\boldsymbol{v} - \boldsymbol{v}_1\| + Ch_j^p \sqrt{n}\,\|\boldsymbol{v}_1\|,$$

where the last inequality uses Cauchy–Schwarz, and in this proof we use $\|\cdot\|$ to denote the standard Euclidean 2-norm on boldface vectors. Therefore, we need only show that $\|\boldsymbol{v} - \boldsymbol{v}_1\|$ is on the order h^p and that $\|\boldsymbol{v}_1\|$ is bounded. We have that

$$(3.15) \qquad \|\boldsymbol{v}_1\| \leq \left\|\boldsymbol{G}_1^{-1}\right\| \|\boldsymbol{f}_1\| \leq \frac{\sqrt{n}}{\sigma_{\min}(\boldsymbol{G}_1)} U^2 \leq \frac{2\sqrt{n}}{\sigma_{\min}(\boldsymbol{G})} U^2,$$

where the last inequality holds since by (3.13),

$$\|\boldsymbol{G}_1 - \boldsymbol{G}\| < \frac{1}{2}\sigma_{\min}(\boldsymbol{G}) \implies \frac{1}{\sigma_{\min}(\boldsymbol{G}_1)} < \frac{2}{\sigma_{\min}(\boldsymbol{G})}.$$

For the second term in (3.14), Lemma 3.2 implies that

$$(3.16) \qquad |(G)_{p,q} - (G_j)_{p,q}| \leq Ch_j^p, \qquad |(f)_p - (f_j)_p| \leq Ch_j^p, \qquad p, q = 1, \ldots, n,$$

i.e., the vector $\boldsymbol{v}_1$ is the solution to a perturbed version of the linear system $\boldsymbol{G}\boldsymbol{v} = \boldsymbol{f}(k)$. We now use a standard result in linear algebra: if $\boldsymbol{G}$ is square and invertible, then

$$\|\boldsymbol{v} - \boldsymbol{v}_1\| \leq \frac{1}{\eta} \left(\|\boldsymbol{f}(k) - \boldsymbol{f}_1(k)\| + \|\boldsymbol{G} - \boldsymbol{G}_1\| \|\boldsymbol{v}\| \right),$$

where η satisfies

$$\eta := \sigma_{\min}(\boldsymbol{G}) - \|\boldsymbol{G}_1 - \boldsymbol{G}\| \overset{(3.13)}{>} \frac{1}{2}\sigma_{\min}(\boldsymbol{G}) > 0.$$

The entrywise proximity relations (3.16) imply that

$$\|\boldsymbol{G} - \boldsymbol{G}_1\| \leq nCh^p, \qquad\qquad \|\boldsymbol{f} - \boldsymbol{f}_1\| \leq \sqrt{n}Ch^p.$$

We therefore have

$$\|\boldsymbol{v} - \boldsymbol{v}_1\| \leq \frac{1}{\eta} \left(C\sqrt{n}h^p + Cnh^p\|\boldsymbol{v}\| \right)$$

$$(3.17) \qquad \leq \frac{Ch^p}{\eta} \left(\sqrt{n} + n\|\boldsymbol{G}^{-1}\|\|\boldsymbol{f}\| \right) \leq \frac{Ch^p}{\sigma_{\min}(\boldsymbol{G})} \left(\sqrt{n} + \frac{n^{3/2}}{\sigma_{\min}(\boldsymbol{G})} U^2 \right).$$

Using (3.15) and (3.17) in (3.14) yields the result. $\qquad\qquad\square$

The assumption $\sigma_{\min}(\boldsymbol{G}) > 0$ is equivalent to assuming that the set of n solutions $u(\cdot, \mathcal{K}_n)$ is linearly independent in H. The appearance of $1/\sigma_{\min}(\boldsymbol{G})$ in (3.12) is expected due to worst-case linear system perturbation theory, but since the bound for this term is loose we expect the estimate (3.12) to be pessimistic in magnitude. The dependence of C_1 on $\sigma_{\min}^2(\boldsymbol{G})$ is another worst-case estimate and is sharp only when $\boldsymbol{f}_1(k)$ has large component pointing in the minimal singular direction of $\boldsymbol{G}_1$.

Remark 3.4. The error in (3.12) is the sum of two terms: One term is independent of j, and another scales like h_j^p. Such an error behavior suggests that we may be able to accelerate convergence to reduce the h_j^p error term by usage of Richardson extrapolation. However, the estimate (3.12) suggests that the right-hand side is dominated by the j-independent h^p term. A Richardson extrapolation technique operating on different j levels cannot eliminate this term, and an extrapolated approximation will have error behaving still like h^p. For the numerical results we have tested, the h_j^p term appears to dominate the error behavior and so extrapolation techniques are successful. The observed h_j^p dependence may result from a lack of sharpness of our estimate or is possibly the result of the particular examples we show and does not hold in general.

The theorem above relates the error of $\widehat{u}_j$ to $P_n u$. If the assumption of Lemma 3.1 holds, then we in addition have that the error between $P_n u$ and u is comparable to $\sqrt{d_{n/2}(\mathcal{V})}$.

4. Convergence acceleration. We have discussed computation of the multi-fidelity approximation $\widehat{u}_j(k)$, which is a member of H_j. The goal of this section is to illustrate that convergence of this approximation can be accelerated if we have knowledge of $\widehat{u}_1(k)$, $\widehat{u}_2(k)$, and $\widehat{u}_3(k)$. The cost of obtaining these three solutions (essentially just the cost of $u_1(k)$) is much less than the cost of computing the three solutions $u_1(k)$, $u_2(k)$, and $u_3(k)$ so that the multifidelity procedure can significantly speed up sequence transformation procedures.

The multifidelity approximation $\widehat{u}_j$ that we have constructed lives in H_j. To accomplish acceleration, we require a method to "lift" this to H. A standard technique for evaluating dense output of a solution trajectory is a continuous extension approach. Since multistep methods are built by constructing approximating polynomials, this approximating polynomial is a natural candidate for continuous extension in practical solvers [24]. Extension approaches for multistage methods, i.e., Runge–Kutta-type methods, likewise exist [22, 27, 14, 9] but are considerably more complicated than their multistep counterparts.

The previously mentioned extension techniques are preferable in general, but we opt for a simpler B-spline approach on trajectory data to accomplish our continuous extension; in particular our examples do not exhibit, e.g., stiff behavior, so that this simple approach is sufficient to accomplish our desired lift to H to demonstrate the utility of the multifidelity method. One advantage of this choice over alternative continuous extension or dense output methods is that spline approaches can be used in a black-box manner on output solution trajectories. However, such an unsophisticated approach would not adhere to desired properties in more complicated scenarios such as contractivity or monotonicity. In such cases, strong-stability-type methods for dense output would be more appropriate [17]. The central focus of this paper is the multifidelity scheme and its extrapolation, so we opt for the simplest approach using B-splines constructed directly from trajectory data.

4.1. Connection operators: Splines. The multifidelity reconstruction $\widehat{u}_j^{(m)}(\cdot, k)$ is a vector in $\mathbb{R}^{N_j+1}$ with data associated to time instances

$$t_{j,i} = ih_j, \qquad\qquad i \in [N_j].$$

For a fixed multifidelity level j and fixed time-stepping order of accuracy $p > 0$, we define a knot vector ξ_i for use in spline construction. The first p knots coincide, followed by equispaced knots, followed again by coincident knot values:

$$(4.1) \qquad \xi_i = \begin{cases} 0, & 0 \le \ i \ \le p, \\ \frac{i-p}{N_j-p}, & p+1 \le \ i \ \le N_j+p+1, \\ 1, & N_j+p+2 \le \ i \ \le N_j+2p+2. \end{cases}$$

In one dimension, basis splines (B-splines) are defined recursively using a knot vector, starting with piecewise constants

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if } \ \xi_i \le \xi < \xi_{i+1}, \\ 0 & \text{otherwise} \end{cases}$$

for $p = 0$ and

$$(4.2) \qquad B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi)$$

for $p > 0$. We choose the uniform knots (4.1) because our data lies on a uniform time grid. Then given a fixed $k \in \mathcal{K}$, we can use the discrete-time approximation $\widehat{u}_j(k)$ as data in a global B-splines approximation problem:

$$(4.3) \quad \widehat{u}_j^{(m)}(i, k) = \sum_{s \in [N_j]} \alpha_i B_{s,p}(t_{j,i}/T), \quad \alpha_0 = \widehat{u}^{(m)}(t_0, k), \quad \alpha_{N_j} = \widehat{u}^{(m)}(t_{N_j}, k)$$

for $1 \le i \le N_j - 1$. The above system represents $N_j + 1$ equations in $N_j + 1$ unknowns α_i, which can be solved. Once the α_i coefficients are known, we can form the spline approximation

$$(4.4) \qquad \widehat{w}_j^{(m)}(t, k) = \sum_{i \in [N_j]}^{n} \alpha_i B_{i,p}(t/T), \qquad\qquad 0 \le t \le T.$$

Repeating this for $m = 1, \ldots, M$, we can create a continuous approximation $\widehat{w}_j(\cdot, k) \in H$ to the discrete multifidelity data $\widehat{u}_j(\cdot, k) \in H_j$ at any time value $t \in [0, T]$, and this approximation is accurate to order p.

4.2. Richardson extrapolation. With $w = P_n u$, Theorem 3.3 states that the multifidelity approximations $\widehat{u}_j$ satisfy

$$(4.5) \qquad\qquad \widehat{u}_j(i, k) \simeq w(ih_j, k) + C_1 h^p + \frac{C_2}{r^{p(j-1)}} h^p.$$

This suggests that the use of sequence transformations operating on the index j to accelerate convergence may be effective [25, 26]. We adopt Richardson extrapolation in particular, though there are many other extrapolation procedures [15, 5, 6]. Throughout this section, we use x_j to denote a generic scalar in the spline-postprocessed multifidelity sequence, i.e., for some fixed $t \in [0, T]$, $k \in \mathcal{K}$, and $m = 1, \ldots, M$, we let

$$x_j^{(m)} := \widehat{w}_j^{(m)}(t, k) \approx w^{(m)}(t, k) + C_1 h^p + C_1 r^{(1-j)p} h^p.$$

Although we know the convergence order p from an a priori understanding of the time integration method, we use three levels to estimate this order. The Richardson extrapolation formulas for estimating the order and the resulting extrapolation are, respectively,

$$(4.6) \qquad p^* = \log_r\left(\frac{x_1 - x_2}{x_2 - x_3}\right), \qquad x^* = \frac{r^{p^*}x_3 - x_2}{r^{p^*} - 1} = c^*x_3 + (1 - c^*)x_2,$$

where we have defined $c^* \coloneqq r^{p^*}/(r^{p^*} - 1)$. With this order of accuracy in hand, the accelerated estimate x^* is our final computed solution. For arbitrary $t \in [0, T]$, $k \in \mathcal{K}$, and $m = 1, \ldots, M$, the above procedure can be repeated on the sequence x_j defined below, producing the estimate x^*:

$$(4.7) \qquad x_j = \widehat{w}_j^{(m)}(t, k), \qquad \widehat{w}^{*(m)}(t, k) \coloneqq x^* \approx w^{(m)}(t, k).$$

The final output of our algorithm is $\widehat{w}^*(\cdot, k) \in H$. Note that, on account of the behavior (4.5), we only expect x^* to approximate w with an error of h^p. However, we will see in our numerical results section that accuracy of order $p + 1$ is observed.

4.3. Computational complexity of the trifidelity construction. This section summarizes the entire algorithm. The simulation cost for the convergence-accelerated trifidelity algorithm is divided in two main parts:

- Offline stage—Identification of $\mathcal{K}_n$ and computation of $u_2(\cdot, \mathcal{K}_n)$ and $u_3(\cdot, \mathcal{K}_n)$.
  - Low-fidelity computations: Compute $Q \gg 1$ simulations of u_1.
  - Important sample selection: Approximate the $j = 1$ optimization in (3.2) using the Q simulations of u_1 via a pivoted Cholesky decomposition. This requires $\mathcal{O}(Qn^2)$ operations. The pivots identify $\mathcal{K}_n = \{k_1, \ldots, k_n\} \subset \mathcal{K}$.
  - Medium- and high-fidelity computations: n simulations of u_2 and u_3 at parameter locations $\mathcal{K}_n$.
- Online stage—Given k, compute $\widehat{w}^*(t, k)$, an approximation to $u(t, k)$.
  - Low-fidelity computation: Evaluate u_1 at parameter value k.
  - Multifidelity interpolation operator: Compute medium- and high-fidelity approximations $\widehat{u}_2$ and $\widehat{u}_3$ in (3.11), involving the solution to the $j = 1$ version of (3.5). The cost of this construction is dominated by the cost of a least-squares solve on the low-fidelity mesh, and hence requires $\mathcal{O}(N_1 n^2)$ operations.
  - Spline interpolation: For $j = 1, 2, 3$, solve (4.3) to obtain the spline approximation (4.4) to obtain $\widehat{w}_j$.
  - Sequence transformation: Use (4.6) and (4.7) to compute the estimator $\widehat{w}^*$ at parameter value k, which can be evaluated at any $t \in [0, T]$.

The major computational burden is only in the computation of n medium- and high-fidelity solutions, which is a one-time ("offline") cost. Once this cost has been invested, one may compute the accelerated multifidelity estimator $\widehat{w}$ at the cost of only the low-fidelity model u_1.

5. Numerical results. We use this section to demonstrate the effectiveness of the accelerated multifidelity procedure. We wish to illustrate that one can use quite general time integrators. To this end, we will use the standard second-, third-, and fourth-order Runge–Kutta (RK, multistage) and Adams–Bashforth (AB, multistep) schemes. The Runge–Kutta schemes RK2, RK3, and RK4 are second-, third-, and fourth-order globally accurate, respectively, and similarly for the Adams–Bashforth schemes, which we denote AB2, AB3, and AB4.
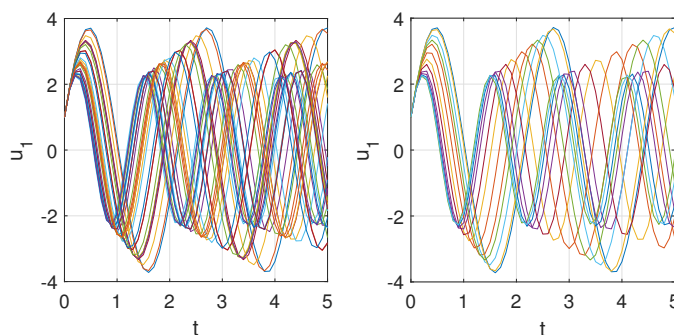
FIG. 2. *Ensemble of solution realizations of the ODE* (5.1) *using a low-fidelity RK4 solver. Left: an ensemble of low-fidelity trajectories. Right: $n = 13$ trajectories identified via a pivoted Cholesky decomposition.*

TABLE 2
Convergence order estimation.

Time-stepping method	RK2		AB2		RK3		AB3		RK4		AB4	
k	11	16	11	16	11	16	11	16	11	16	11	16
p^*	2.20	1.97	2.19	2.04	3.01	3.29	2.97	3.36	4.38	3.95	4.31	3.85

5.1. Damped harmonic oscillator. In this section we consider a second-order linear parameterized ODE and demonstrate different steps of the convergence acceleration algorithm on this illustrative example. The linear ODE model of a particular unforced mass-spring-damper is

$$(5.1) \qquad \ddot{u} + (0.1 + k/100)\dot{u} + ku = 0, \quad u(0) = 1, \; \dot{u}(0) = 10,$$

where $k \in [5, 25]$ is the parameter, and hence the stiffness coefficient and the damping coefficient for the system are dependent parameters.

5.1.1. Multifidelity approximations. We construct three different approximations, $u_1, u_2,$ and u_3, associated with timestep values $h_1 = 0.1, h_2 = 0.05$, and $h_3 = 0.025$, respectively, and run the multifidelity procedure to construct a numerical approximation to the solution u for arbitrary $k \in [5, 25]$. Different solution realizations obtained with the solver RK4 on the low-fidelity model are shown in Figure 2. The first step of the multifidelity procedure is to collect solution trajectories u_1 for many values of k. We choose 100 values of $k \in [5, 25]$ via Monte Carlo sampling. We use this size-100 grid as a proxy for the continuum in the optimization problem (3.2) (i.e., Q from section 4.3 is set to 100); this results in $n = 13$ parameter values $k_1, \ldots, k_n$ along with medium- and high-fidelity solutions computed on these parameter values.

5.1.2. Sequence transformation and acceleration. We investigate the convergence order of $\widehat{u}_3(k)$ at $t = 2.5$, for the two values $k = 11$ and $k = 16$. The convergence order p^* is estimated via (4.6), with x_j being reconstructed multifidelity solutions $\widehat{u}_j(k)$ at the fixed time instance $t = 2.5$, and this p^* is used for all time t. The computed values of p^* for particular parameter values k are given in Table 2. The convergence order p^* mirrors the order of the convergence p of the time-stepping algorithm, regardless of whether a multistage (RK) or multistep (AB) scheme is used.

Once the multifidelity solutions $\widehat{u}_2$ and $\widehat{u}u_3$ solutions are built, these solutions are interpolated with an order-p spline, where p is again the order of the time integration
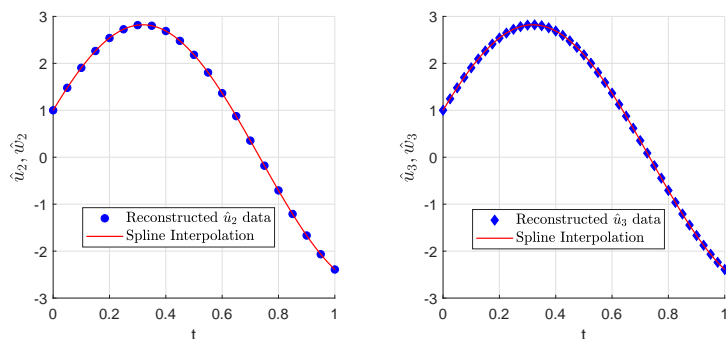
FIG. 3. *Interpolation with high-order spline for reconstructed $\hat{u}_2$ and $\hat{u}_3$ solutions. Note that we do not use $\hat{u}_1$ for the convergence acceleration.*

method. Figure 3 shows the spline curve $\widehat{w}_2$ and $\widehat{w}_3$ computed from the multifidelity data $\widehat{u}_2$ and $\widehat{u}_3$, respectively. For better resolution only results for $t \in [0,1]$ are visualized.

For a given parameter k and time instance t, the spline-reconstructed medium- and high-fidelity solutions $\hat{w}_2$ and $\hat{w}_3$ are used to obtain the extrapolated solution $\widehat{w}^*$ via

$$(5.2) \qquad\qquad \widehat{w}^* = c\widehat{w}_3 - (c-1)\widehat{w}_2.$$

This equation is equivalent to (4.6) with $c = r^p/(r^p - 1)$. In this example, $r = 2$ since the timestep is halved between fidelities ($2h_{j+1} = h_j$). We can explicitly compute the Richardson extrapolation weights for $p = 2, 3, 4$:

$$p^* = 2 \Longrightarrow c^* = \frac{4}{3} \approx 1.33, \quad p^* = 3 \Longrightarrow c^* = \frac{8}{7} \approx 1.14, \quad p^* = 4 \Longrightarrow c^* = \frac{16}{15} \approx 1.06.$$

We test the accuracy of this approach for different values of c; based on our convergence theory, if our spline approximation is of the appropriate order, then we expect that $c = c^*$ will produce the best results (lowest error). We can confirm this behavior in Figures 4 and 5. The relative error is shown for solvers of different convergence orders (RK2, RK3, and RK4 for Figure 4, and AB2, AB3, and AB4 for Figure 5), and different orders of spline interpolation are used. Relative error is measured in the normalized ℓ^2 norm of the vector of values on a fine grid with stepsize $h = 10^{-3}$.

When the spline order of convergence is greater than or equal to the order of the convergence of the time-stepping algorithm, we expect $c = c^*$ to produce the best error. This expectation is realized in these figures: increasing the spline order to the time-stepping order p causes the minimum error to happen at $c = c^*$; increasing the spline order beyond this produces no change since the error is then limited by the time-stepping scheme.

We emphasize that these experiments are testing more than simply standard Richardson extrapolation: they are also verifying the convergence rate of the multifidelity approximation given in Theorem 3.3. The difference between a standard Richardson extrapolation technique and this multifidelity technique is that the standard approach requires solutions u_2 and u_3, which are relatively expensive. In contrast, the multifidelity procedure requires only the surrogates $\widehat{u}_2$ and $\widehat{u}_3$, which can
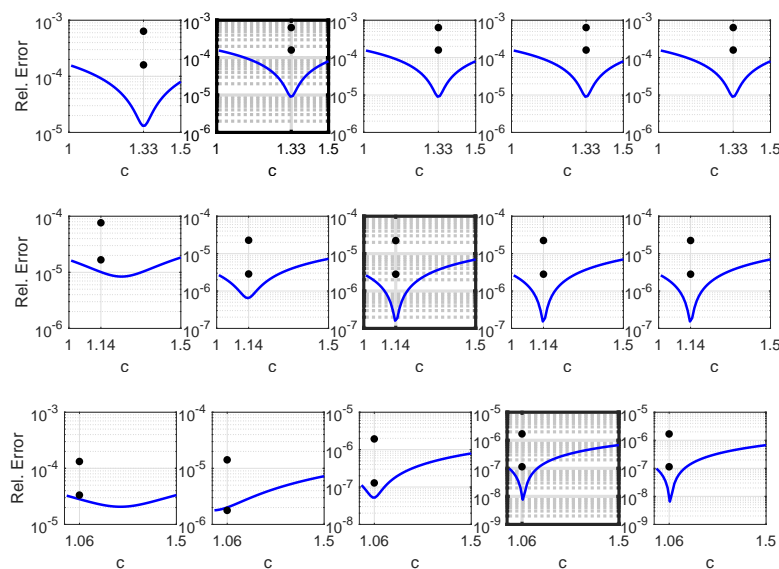
FIG. 4. *Performance of extrapolation weights c in* (5.2) *for different RK schemes and spline interpolation orders: second-, third-, and fourth-order RK (top to bottom rows), spline degrees $p = 1, \ldots, 5$ (left to right). The value c^* is marked on each horizontal axis in the plots. As we proceed left-to-right in each row, we expect that $c = c^*$ produces the optimal (smallest) error when the spline order matches the time-stepping order; this happens in the second, third, and fourth columns of rows 1, 2, and 3, respectively. If we increase the spline order beyond the time-stepping order, no change should be observed since the error rate of the multifidelity spline approximations is then limited by the time-stepping error. The black dots represent the errors $\|\hat{w}_2 - u\|_2$, $\|\hat{w}_3 - u\|_2$ and the blue line is obtained by varying c in $\|\hat{w}^*(c) - u\|_2$ (cf. (5.2)). We plot the black dots at the abscissa corresponding to $c = c^*$ to better visualize the magnitude of accuracy enhancement. The figures with bold axis lines correspond to those where the time integration order matches the spline order.*

be generated at the cost of the much more inexpensive model $\hat{u}_1$. (After some offline work has been invested, see section 4.3.)

Finally we numerically investigate the order of convergence for the convergence-accelerated multifidelity estimator $\hat{w}^*$ and compare it to the expected order of accuracy, p. We observe the convergence rates of order $p + 1$ in Figure 6. Owing to Theorem 3.3, the multifidelity approximations $\hat{u}_j$ have error scaling like $h^p(1 + r^{1-j})$. While a Richardson extrapolation technique can eliminate the $h^p r^{1-j}$ term, we do not expect that it eliminates the h^p term. We believe this discrepancy between theory and results is due to the explanation in Remark 3.4, i.e., that our estimate in Theorem 3.3 resulting in a non-j-dependent h^p term is a loose bound. As a consequence, we numerically observe order-$(p + 1)$ convergence for the accelerated solution instead of the theoretically expected order-p convergence.

5.1.3. Statistical moments. Under the same model (5.1) we interpret k as a random variable, uniformly distributed over $[5, 25]$. We can then use the multifidelity procedure to estimate statistical moments of $u(k, t)$ as a function of t. In all cases, our statistics are computed numerically using a size-1000 ensemble of Monte Carlo values of k. The trajectories of the exact mean and standard deviation of u are shown in the left-hand pane of Figure 7. In the right-hand pane, we compute the statistical moment error using the multifidelity surrogate $\hat{w}^*$ and compare them against the statistical moment errors computed using the original models u_1, u_2, and u_3.
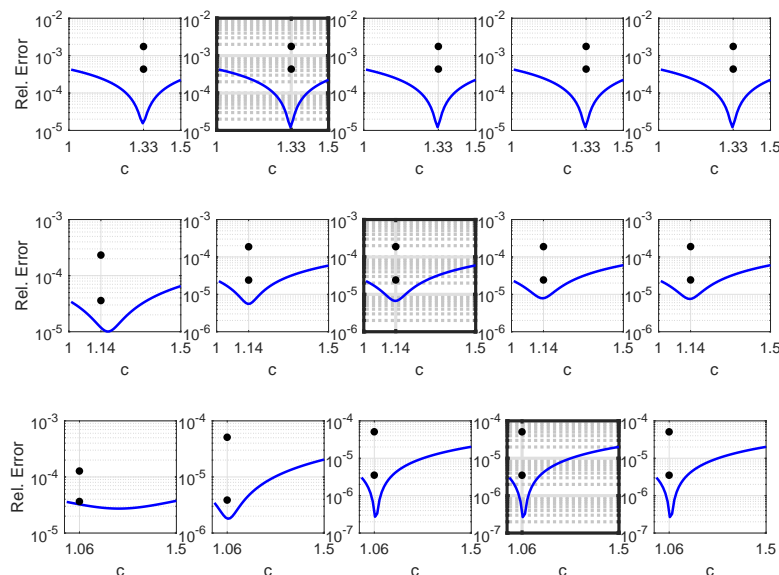
FIG. 5. *Performance of extrapolation weights c in* (5.2) *for different AB schemes and spline interpolation degrees: second-, third-, and fourth-order AB (top to bottom rows), spline degrees $p = 1, \ldots, 5$ (left to right). See Figure 4's caption for a more detailed description.*
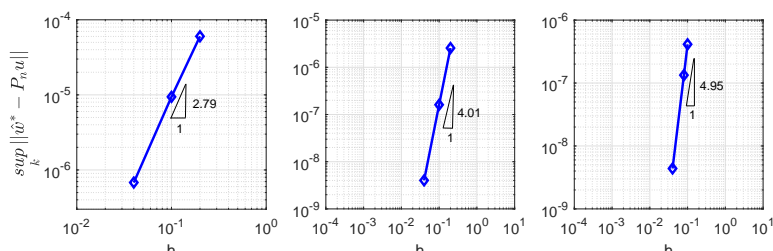

FIG. 6. *Convergence rate corresponding to the slope $p + 1$ in the theoretical estimate: $p + 1 = 2.79, 4.01, 4.95$ for RK2 (left), RK3 (middle), and AB4 (right). The sup is taken over 100 equally spaced points of $k \in [5, 25]$.*

There are two ways to compute the moments for the $j = 3$ solution: via repeated query of $u_3(\cdot, k)$ or via repeated query of the convergence-accelerated multifidelity approximation $\widehat{w}^*(\cdot, k)$. Suppose X is the cost of computing a single solution for $u_3(\cdot, k)$. Then the cost of moments via u_3 is $1000X$. However, the cost of a single online stage query of $\widehat{w}^*$ is $X/4$. The offline stage requires 100 low-fidelity simulations ($100X/4$), 13 medium-fidelity simulations ($13X/2$), and 13 high-fidelity simulations ($13X$). The cumulative cost of $\widehat{w}^*$ is thus $295X$, which is much smaller than the $1000X$ cost of u_3, and $\widehat{w}^*$ is also about an order of magnitude more accurate.

5.2. Nonlinear ODE: Predator-prey equations. We now consider the Lotka–Volterra predator-prey equations. The set of equations is composed of nonlinear ODEs that are primarily used to describe simplified dynamics of biological systems. The evolution of population for prey and predator species $x(t)$ and $y(t)$, respectively, is
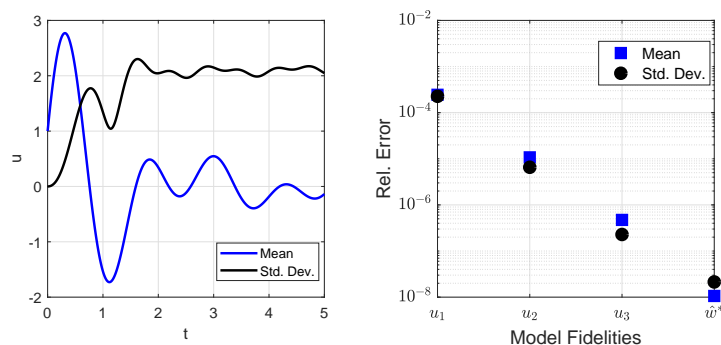
FIG. 7. *Mean and standard deviation of exact ODE solution (left). Error of convergence-accelerated multifidelity approximation $\widehat{w}^*$ vs. error from using low-, medium-, or high-fidelity approximations u_1, u_2, and u_3, respectively. Note that the cost of computing the moments of $\widehat{w}^*$ is much cheaper than computing the moments of u_2 and u_3, requiring a total of only 13 medium- and high-fidelity solutions.*
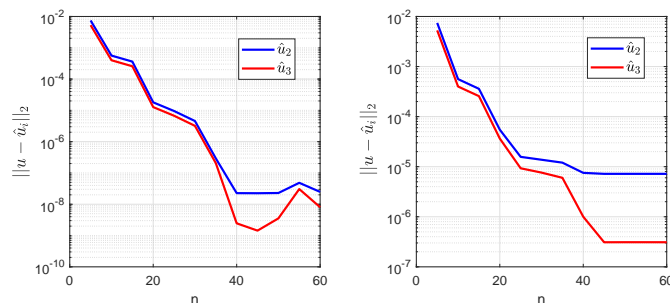

FIG. 8. *Decay of error between the reconstructed multifidelity and exact solutions with respect to the number of higher-fidelity solutions: RK4 (left), AB4 (right).*

modeled as

$$\frac{\mathrm{d}x}{\mathrm{d}t} = ax - bxy, \quad x(0) = x_0,$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = cxy - dy, \quad y(0) = y_0,$$

for an initial population $(x_0, y_0) = (1, 1)$. We parameterize the positive constants $a, b, c,$ and d by

$$a = k + 0.5, \quad b = 3k + 1,$$
$$c = k + 1, \quad d = k + 0.5,$$

where k is a parameter taking values in the interval $[0.5, 1.5]$. In this example since we do not have the analytical solution we solve the nonlinear ODE on a fine grid $h = 10^{-3}$ and use that as an "exact" solution to investigate convergence.

In Figure 8 we show convergence of the unaccelerated multifidelity solutions, $\widehat{u}_j$, as a function of the number of high-fidelity solutions, n. We observe that the error reaches an asymptotic limit as the number of higher-fidelity solutions is increased;
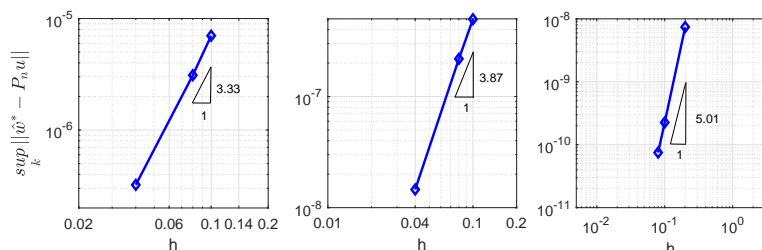
FIG. 9. *Convergence rate corresponding to the slope $p + 1$ in the theoretical estimate: $p + 1 = 3.33, 3.87, 5.01$ for AB2 (left), AB3 (middle), and RK4 (right). The sup is taken over 100 equally spaced points of $k \in [0.5, 1.5]$.*
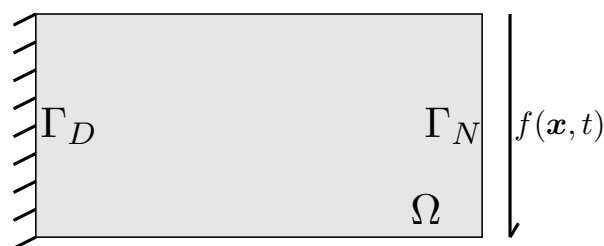

FIG. 10. *Linear elastic structure subjected to a transient shear load.*

this is expected since for small values of n, the time integration error Ch^p is greater than the n-term projective error $\|u - \mathcal{P}_n u\|$.

Figure 9 computes numerically observed rates of convergence for the convergence-accelerated surrogate. We again observe order-$(p+1)$ convergence, despite our theory-based order-p expectation. We again attribute this to the loose bound in our theoretical estimate, as described at the end of section 5.1.3.

5.3. Linear elastodynamics. Our final example demonstrates the effectiveness of our approach in the enhancement of predictive capability for the transient response of structural systems. Such problems are studied for wave propagation in solid materials and structures [1, 19]. We consider the problem of elastodynamics for the linear structure shown in Figure 10, which is subjected to a transient shear load $f(x, t)$. The elastodynamics governing equations are given by

$$
\begin{cases}
-\rho \ddot{u}(\boldsymbol{x}, t) + \nabla \cdot (\mathbb{C} \nabla u(\boldsymbol{x}, t)) + b(\boldsymbol{x}, t) = 0 & \forall \boldsymbol{x} \in \Omega, \quad \forall t \in [0, T], \\
\sigma(\boldsymbol{x}, t) \cdot \boldsymbol{n} = f(\boldsymbol{x}, t) & \forall \boldsymbol{x} \in \Gamma_N, \quad \forall t \in [0, T], \\
u(\boldsymbol{x}, t) = 0 & \forall \boldsymbol{x} \in \Gamma_D, \quad \forall t \in [0, T], \\
u(\boldsymbol{x}, 0) = \boldsymbol{u}_0, \ \dot{u}(\boldsymbol{x}, 0) = \dot{\boldsymbol{u}}_0 & \forall \boldsymbol{x} \in \Omega, \quad \forall t \in [0, T],
\end{cases}
$$

where Γ_D and Γ_N denote the Dirichlet and Neumann boundary surfaces, $\boldsymbol{n}$ denotes the unit vector normal to the structure surface, and ρ, $\mathbb{C}$, b, and ∇ are the mass density, elasticity tensor, body force field, and divergence operator, respectively. We choose a continuous finite element discretization in space, resulting in the semidiscrete form

(5.3)
$$
\begin{aligned}
\boldsymbol{M} \ddot{\boldsymbol{U}} + \boldsymbol{K} \boldsymbol{U} &= \boldsymbol{F}, \\
\boldsymbol{U}(0) = \boldsymbol{U}_0, \ \dot{\boldsymbol{U}}(0) &= \dot{\boldsymbol{U}}_0,
\end{aligned}
$$

where

$$\boldsymbol{M} = \int_{\Omega} \mathbb{N}^T \mathbb{N} \rho \; d\Omega, \qquad \boldsymbol{K} = \int_{\Omega} \mathbb{B}^T \mathbb{C} \mathbb{B} \; d\Omega, \qquad \boldsymbol{F} = \int_{\Omega} \mathbb{N}^T f \; d\Omega$$

are the consistent mass matrix, global stiffness matrix, and load vector with $\mathbb{N}$ and $\mathbb{B}$ the finite element shape function and derivative matrices, respectively. The domain Ω is tessellated into a 40×20 grid of quadrilaterals, with a bilinear isotropic finite element space[2] and assume a plane stress condition; hence

$$\mathbb{C} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1 - \nu}{2} \end{bmatrix},$$

where E is the elastic modulus and ν is the Poisson ratio.

We consider a similar parametric model for both the elasticity tensor (i.e., elastic modulus) and mass density, and an independent parametric model for the load. We prescribe a Karhunen–Loève expansion (KLE) to parameterize the distributed load and the structural properties, i.e., the mass density and elastic modulus. We assume a covariance function

$$\mathbf{R}_{xx'} = \exp\left(- \frac{\|x - x'\|_2^2}{2 l_c^2} \right),$$

where $\|x - x'\|_2$ is the Euclidean distance between locations x and x' and l_c is the correlation length. We discretize the covariance function by assuming that each parametric field is uniform over each element for structural properties over Ω; the value of the field is taken as the value at the centroid of the element. We use n_m first eigenvalue-eigenvector pairs $(\lambda_i, \boldsymbol{\gamma}_i)$ of the covariance matrix to generate the KLE decomposition of a zero mean process as

$$Z(\boldsymbol{x}, \boldsymbol{k}) = \gamma_0 + \sum_{i=1}^{n_m} \sqrt{\lambda_i} \gamma_i(\boldsymbol{x}) k_i,$$

where k_i are modeled as uniform random variables $U[0, 1]$, and γ_i are eigenvectors of the covariance kernel $\boldsymbol{R}_{xx'}$. The factor γ_0 is a positive that ensures positivity of the random field Z. We use $n_m = 2$ and $n_m = 3$ modes to construct random fields for structural properties and load, respectively; therefore we parameterize this model with five independent uniform parameters, i.e., $\boldsymbol{M} \rightarrow \boldsymbol{M}(\boldsymbol{k})$, $\boldsymbol{K} \rightarrow \boldsymbol{K}(\boldsymbol{k})$, $\boldsymbol{F} \rightarrow \boldsymbol{F}(\boldsymbol{k})$ (cf. (5.3)), where $\boldsymbol{k} = (k_1, \ldots, k_5)$.

This example involves a multidimensional parametric space, so we use a quadrature rule [11, 16] to select the training set for the parametric samples, as opposed to previous examples which divided the univariate parametric domain to equispaced increments. In particular we use a sparse grids quadrature [10] and select a 151-point

---

[2]We use the spatial mesh 40×20 to satisfy the Courant–Fredricks–Lewy stability condition for various timesteps h that we consider in this example.
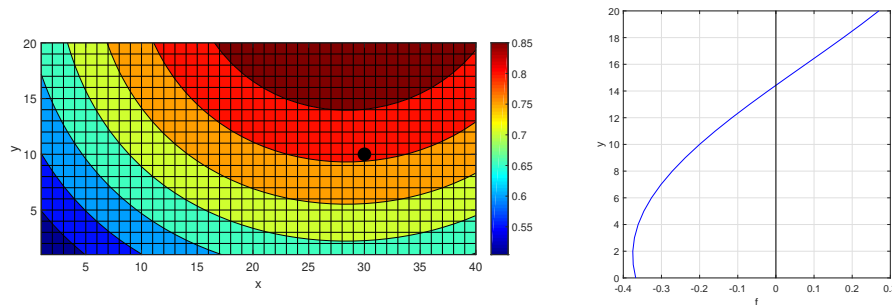
FIG. 11. *A snapshot of the elastic modulus (left) and shear load (right) associated with an arbitrary quadrature point. The vertical displacement of the node indicated with the black circle is used for the numerical study.*
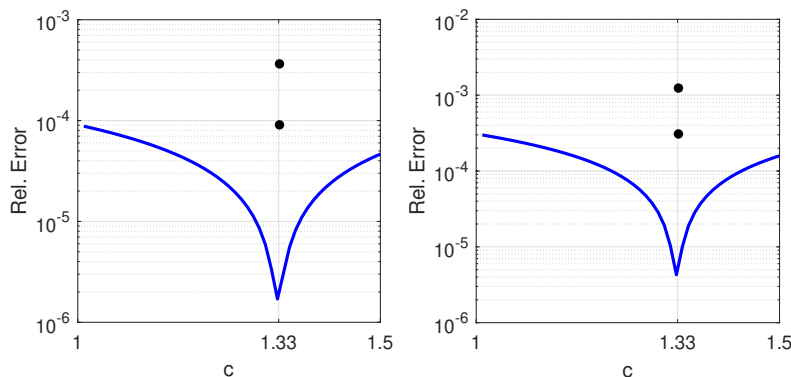

FIG. 12. *Performance of extrapolation weights c in* (5.2) *for RK2 (left) and AB2 (right) with spline interpolation degree $p = 5$.*

quadrature rule that can exactly integrate polynomials up to degree 7 in five dimensions. Figure 11 shows a snapshot of the parametric elastic modulus and shear load associated with a particular value of $\boldsymbol{k}$. To study convergence, we focus on the vertical displacement of the node indicated with the black circle. We use the timestep $h = 10^{-3}$ solution of the ODE (5.3) as the exact solution to investigate convergence. We assume time-dependent load $f(\boldsymbol{x}, t) = \boldsymbol{Z}\cos(2\pi t/5)$, where $\boldsymbol{Z} \equiv Z(\boldsymbol{x})$, and the initial conditions $\boldsymbol{U} = \boldsymbol{K}^{-1}\boldsymbol{Z}$, $\dot{\boldsymbol{U}} = \boldsymbol{0}$. We use the inexpensive time integration schemes RK2 and AB2, whose convergence rates are accelerated via our numerical procedure.

Figure 12 shows the performance of extrapolation weights c (cf. (5.2)) for two time integration schemes using a $p = 2$ spline interpolation. Again the optimal weight coincides with the expected Richardson extrapolation coefficient value $c^* = 4/3$.

We also compute the mean and standard deviation of the transient response and their extrapolated response similarly to section 5.1.3. The errors in the low-, medium-, and high-fidelity approximations, as well as the extrapolated responses for two integration schemes, are shown in Figure 13, where the convergence acceleration is apparent.

Finally, like the previous examples we numerically investigate the order of convergence for the extrapolated estimator by varying the time increment size h. Figure 14
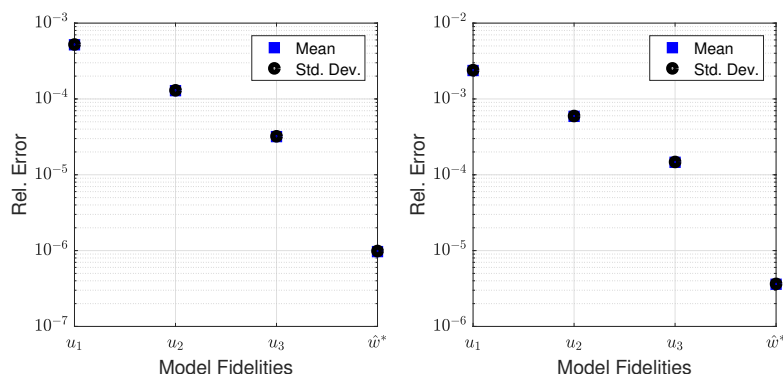
FIG. 13. *Mean and standard deviation error of extrapolated approximation $\widehat{w}^*$ vs. error from using low-, medium-, and high-fidelity approximations u_1, u_2, and u_3, respectively, using RK2 (left) and AB2 (right). In this example, the mean and standard deviation error markers overlap.*
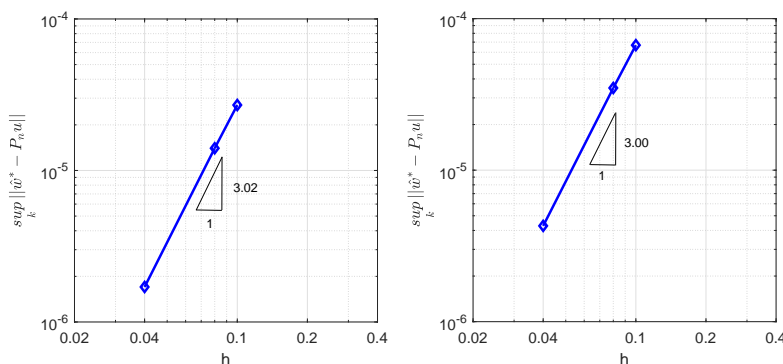

FIG. 14. *Convergence rate corresponding to the slope $p+1$ in the theoretical estimate: $p+1 = 3.02, 3.00$ for RK2 (left) and AB2 (right). The sup is taken over all sparse grids quadrature points $\boldsymbol{k} \in \mathcal{K}$, where $\mathcal{K} = \{\boldsymbol{k}^{(i)}\}_{i=1}^{151}$.*

demonstrates third-order convergence using second-order time integration schemes, which is in accordance with our observation in previous examples.

6. Concluding remarks. A numerical method for leveraging time-dependent multifidelity models under parametric uncertainty is presented. We built interpolation operators on the inexpensive low-fidelity solution in parameter space, and estimated higher-fidelity solutions corresponding at arbitrary parameter locations using the same interpolation rule associated with the low-fidelity solution. We chain this multifidelity procedure together with classical sequence transformation, in particular Richardson extrapolation: Having knowledge of solutions at different fidelity levels allows us to estimate the convergence order and build a sequence transformation operator that attains superior accuracy compared to the standard multifidelity surrogate.

REFERENCES

[1] A. V. Amirkhizi, A. Tehranian, and S. Nemat-Nasser, *Stress-wave energy management through material anisotropy*, Wave Motion, 47 (2010), pp. 519–536, https://doi.org/10.1016/j.wavemoti.2010.03.005.

[2] P. Benner and H. Fassbender, *Model order reduction: Techniques and tools*, in Encyclopedia of Systems and Control, J. Baillieul and T. Samad, eds., Springer, Berlin, 2013, pp. 1–10, https://doi.org/10.1007/978-1-4471-5102-9_142-1.

[3] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715.

[4] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM J. Math. Anal., 43 (2011), pp. 1457–1472, https://doi.org/10.1137/100795772.

[5] C. Brezinski, *Extrapolation algorithms and Padè Approximations: A historical survey*, Appl. Numer. Math., 20 (1996), pp. 299–318.

[6] C. Brezinski, *Convergence acceleration during the 20th century*, J. Comput. Appl. Math., 122 (2000), pp. 1–21.

[7] K. Carlberg, C. Bou-Mosleh, and C. Farhat, *Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations*, Internat. J. Numer. Methods Engrg., 86 (2011), pp. 155–181, https://doi.org/10.1002/nme.3050.

[8] R. DeVore, G. Petrova, and P. Wojtaszczyk, *Greedy algorithms for reduced bases in Banach spaces*, Constr. Approx., 37 (2013), pp. 455–466, https://doi.org/10.1007/s00365-013-9186-2.

[9] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I: *Nonstiff Problems*, Springer Ser. Comput. Math., Springer, Berlin, 1993.

[10] F. Heiss and V. Winschel, *Quadrature on Sparse Grids*, http://www.sparse-grids.de/.

[11] F. Heiss and V. Winschel, *Likelihood approximation by numerical integration on sparse grids*, J. Econometrics, 144 (2008), pp. 62–80.

[12] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs Math., Springer, Berlin, 2016.

[13] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monogr. Mech., Cambridge University Press, Cambridge, 1996, https://doi.org/10.1017/CBO9780511622700.

[14] M. Horn, *Fourth- and fifth-order, scaled Runge–Kutta algorithms for treating dense output*, SIAM J. Numer. Anal., 20 (1983), pp. 558–568, https://doi.org/10.1137/0720036.

[15] D. C. Joyce, *Survey of extrapolation processes in numerical analysis*, SIAM Rev., 13 (1971), pp. 435–490.

[16] V. Keshavarzzadeh, R. M. Kirby, and A. Narayan, *Numerical integration in multiple dimensions with designed quadrature*, SIAM J. Sci. Comput., 40 (2018), pp. A2033–A2061.

[17] D. I. Ketcheson, L. Lóczi, A. Jangabylova, and A. Kusmanov, *Dense output for strong stability preserving Runge–Kutta methods*, J. Sci. Comput., 71 (2017), pp. 944–958, https://doi.org/10.1007/s10915-016-0331-5.

[18] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza, *Generalized reduced basis methods and n-width estimates for the approximation of the solution manifold of parametric PDEs*, in Analysis and Numerics of Partial Differential Equations, F. Brezzi, P. Colli Franzone, U. Gianazza, and G. Gilardi, eds., Springer INdAM Ser., Springer, Berlin, 2013, pp. 307–329, https://doi.org/10.1007/978-88-470-2592-9_16.

[19] C. Le, T. E. Bruns, and D. A. Tortorelli, *Material microstructure optimization for linear elastodynamic energy wave management*, J. Mech. Phys. Solids, 60 (2012), pp. 351–378, https://doi.org/10.1016/j.jmps.2011.09.002.

[20] Y. Maday, A. T. Patera, and G. Turinici, *A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations*, J. Sci. Comput., 17 (2002), pp. 437–446.

[21] A. Narayan, C. Gittelson, and D. Xiu, *A stochastic collocation algorithm with multifidelity models*, SIAM J. Sci. Comput., 36 (2014), pp. 495–521.

[22] S. P. Nørsett and G. Wanner, *Perturbed collocation and Runge-Kutta methods*, Numer. Math., 38 (1981), pp. https://doi.org/10.1007/BF01397089.

[23] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, Unitext 192, Springer, Berlin, 2016, http://link.springer.com/10.1007/978-3-319-15431-2.

[24] L. Shampine and M. Reichelt, *The MATLAB ODE Suite*, SIAM J. Sci. Comput., 18 (1997), pp. 1–22, https://doi.org/10.1137/S1064827594276424.

[25] J. Stoer and R. Bulirsch, *Numerical treatment of ordinary differential equations by extrapolation methods.*, Numer. Math., 8 (1966), pp. 1–13.

[26] E. J. Weniger, *Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series*, Comput. Phys. Rep., 10 (1989), pp. 189–371.

[27] M. Zennaro, *Natural continuous extensions of Runge-Kutta methods*, Math. Comp., 46 (1986), pp. 119–133, https://doi.org/10.1090/S0025-5718-1986-0815835-1.

[28] X. Zhu, A. Narayan, and D. Xiu, *Computational aspects of stochastic collocation with multifidelity models*, SIAM/ASA J. Uncertain. Quantif., 2 (2014), pp. 444–463, https://doi.org/10.1137/130949154.