

L1-based reduced over collocation and hyper reduction for steady state and time-dependent nonlinear equations

Yanlai Chen^{*}, Lijie Ji[†], Akil Narayan[‡], Zhenli Xu[§]

Abstract

The task of repeatedly solving parametrized partial differential equations (pPDEs) in optimization, control, or interactive applications makes it imperative to design highly efficient and equally accurate surrogate models. The reduced basis method (RBM) presents itself as such an option. Accompanied by a mathematically rigorous error estimator, RBM carefully constructs a low-dimensional subspace of the parameter-induced high fidelity solution manifold on which an approximate solution is computed. It can improve efficiency by several orders of magnitudes leveraging an offline-online decomposition procedure. However this decomposition, usually implemented with aid from the empirical interpolation method (EIM) for nonlinear and/or parametric-nonaffine PDEs, can be challenging to implement, or results in severely degraded online efficiency.

In this paper, we augment and extend the EIM approach as a direct solver, as opposed to an assistant, for solving nonlinear pPDEs on the reduced level. The resulting method, called Reduced Over-Collocation method (ROC), is stable and capable of avoiding efficiency degradation exhibited in traditional applications of EIM. Two critical ingredients of the scheme are collocation at about twice as many locations as the dimension of the reduced approximation space, and an efficient L1-norm-based error indicator for the strategic selection of the parameter values whose snapshots span

^{*}Department of Mathematics, University of Massachusetts Dartmouth, 285 Old Westport Road, North Dartmouth, MA 02747, USA. Email: yanlai.chen@umassd.edu.

[†]School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China. Email: sjtujidreamer@sjtu.edu.cn.

[‡]Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute, University of Utah, 72 S Central Campus Drive Room 3750, Salt Lake City, UT 84112 USA. Email: akil@sci.utah.edu.

[§]School of Mathematical Sciences, Institute of Natural Sciences, and MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China. Email: xuzl@sjtu.edu.cn.

L. Ji and Z. Xu acknowledge the support from grants NSFC 11571236 and 21773165 and HPC center of Shanghai Jiao Tong University. Y. Chen was partially supported by National Science Foundation grant DMS-1719698 and by AFOSR grant FA9550-18-1-0383. L. Ji is partly supported by China Scholarship Council (CSC, No.201906230067) during the author's one year visit at University of Massachusetts, Dartmouth. A. Narayan was partially supported by NSF award DMS-1848508. This material is based upon work supported by the National Science Foundation under Grant No. DMS-1439786 and by the Simons Foundation Grant No. 50736 while Y. Chen, L. Ji, and A. Narayan were in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the "Model and dimension reduction in uncertain and dynamic systems" program.

the reduced approximation space. Together, these two ingredients ensure that the proposed L1-ROC scheme is both offline- and online-efficient. A distinctive feature is that the efficiency degradation appearing in alternative RBM approaches that utilize EIM for nonlinear and nonaffine problems is circumvented, both in the offline and online stages. Numerical tests on different families of time-dependent and steady-state nonlinear problems demonstrate the high efficiency and accuracy of L1-ROC and its superior stability performance.

1 Introduction

Numerical simulations of systems, often parametrized, arising from various engineering and applied science disciplines are increasingly becoming of multi-query and/or real-time type. For example, optimization and optimal control require multiple forward solves, and interactive applications demand real-time responses. Design of fast numerical algorithms with certifiable accuracies for these settings has therefore continued to attract researchers' attention. The parameters delineating these systems may include boundary conditions, material properties, geometric settings, source properties etc. The wide variety, the complicated dependence of the system on these parameters, and their potential high dimensionality are the major challenges. In addition, the differential equations governing these equations may be nonlinear.

The reduced basis method (RBM) [43, 29] has proved an effective option for this purpose. RBM was first introduced for nonlinear structure problems [1, 40] in 1970s and has proven to be effective for a variety of differential equations, including linear evolution equations [28], viscous Burgers equation [47], the Navier-Stokes equations [18], and harmonic Maxwell's equation [15, 16]. The key to RBM's success in realizing orders-of-magnitude efficiency gain is an offline-online decomposition process where the basis selection and surrogate space construction are performed offline by a greedy algorithm, see review papers [44, 27] and monographs [43, 29] for details. During the offline process, the necessary preparations for the online reduced solver are performed. The ultimate goal is that the complexity of the reduced solver, called upon in a potentially real-time fashion during the online stage, is independent of the number of degrees of freedom of the high-fidelity approximation of the basis functions.

1.1 A key problem: Online efficiency degradation due to (D)EIM

To achieve the efficiency goals of RBM, the *Empirical Interpolation Method* (EIM) or its discrete version (DEIM) [4, 25, 12, 42] is typically leveraged for nonaffine terms and/or nonlinear equations. However, EIM is often not feasible due to strong nonlinearity and/or nonaffinity of the problem. Even when it is feasible, performing (D)EIM can severely degrade the reduced solver's online efficiency when either the parameter dependence or the nonlinearity is complicated, such as when it encodes geometric variability [16, 5]. The reason for the efficiency difficulties is that the online complexity is dependent on the potentially large

number of terms resulting from the EIM decomposition.

Let us use a simple system with a nonaffine parameter dependence as an example. Assume that we are solving a heat conduction problem with nonaffine parameter dependence $-\nabla \cdot (a(x; \boldsymbol{\mu}) \nabla u) = f$. For RBM to realize its intended efficiency gain, we would first apply EIM to approximate the function $a(x; \boldsymbol{\mu})$ by a linear combination of $\boldsymbol{\mu}$ -independent functions, $a(x; \boldsymbol{\mu}) \approx \sum_{q=1}^{Q_a} \theta_q(\boldsymbol{\mu}) a(x; \boldsymbol{\mu}^q)$, where $\{\boldsymbol{\mu}^q\}_{q=1}^{Q_a}$ is an ensemble, typically chosen through a greedy procedure. With the equation written in its weak form $a(u, v; \boldsymbol{\mu}) := (a(x; \boldsymbol{\mu}) \nabla u, \nabla v) = (f, v)$, the number of terms Q_a affects the online solver as follows: Once the offline learning stage identifies an N -dimensional reduced-order solution space with basis $\{\xi^1, \dots, \xi^N\}$, the reduced solver is assembled for each $\boldsymbol{\mu}$, and the corresponding stiffness matrix has entries

$$(a(\xi_i, \xi_j; \boldsymbol{\mu}))_{i,j=1}^N := (a(x; \boldsymbol{\mu}) \nabla \xi_i, \nabla \xi_j)_{i,j=1}^N = \sum_{q=1}^{Q_a} \theta_q(\boldsymbol{\mu}) (a(x; \boldsymbol{\mu}^q) \nabla \xi_i, \nabla \xi_j)_{i,j=1}^N.$$

The complexity of the online solver is therefore linearly dependent on the number of EIM terms Q_a , potentially suffering substantial reductions in efficiency compared to situations when EIM is not needed, i.e. $Q_a = 1$. The reason is that Q_a can be prohibitively large (i.e. much larger than the reduced space dimension N) when the model involves geometric parametrization, see e.g. [16, 5] even if the more efficient matrix version of EIM [37] is adopted. As far as we are aware, efforts to mitigate this drawback are limited and underdeveloped.

1.2 The proposed approach

In this paper, we propose an L1-norm-based reduced over-collocation method (L1-ROC) algorithm that is empirically stable and achieves full online efficiency without suffering Q_a -based efficiency degradation. Our main tools are an augmentation of EIM technique, a further leveraging of the collocation philosophy originally explored in [13], and an extension of the L1 importance indicator proposed in [17]. We summarize below the two major ingredients of L1-ROC that, together, enable the method to circumvent this degradation.

The first ingredient is a strategy to augment the EIM framework and adopt the collocation approach, in contrast to variational approaches (i.e. Galerkin or Petrov-Galerkin) [6, 9, 8] when seeking the reduced solution. This so-called reduced collocation method is proposed and documented to work well in circumventing the EIM degradation for the reduced solver in previous work [13], but its stability is lacking [14]. Our reduced over-collocation methods mitigate this stability defect by collocating at about twice as many locations as the dimension of the reduced order space. Half of these collocation points are identified from manipulation of a basis for this space, and data on these points interpolates the reduced solution. The other half are chosen according to a computational analysis of the reduced order residuals when these basis functions are identified during the offline procedure. They are present to ensure a good interpolation of the residual corresponding to an arbitrary parameter value when the reduced order space is used to solve the pPDE. This ingredient alone is not enough

to achieve online and offline efficiency as the efficient calculation of the error estimators, critical for the construction of the reduced solution space, still relies on direct application of EIM.

This challenge with computing error estimators is addressed by the second ingredient of the L1-ROC method, an efficient alternative for guiding the strategic selection of parameter values to build the reduced solution space. In particular, we utilize the recently introduced empirical L1 approach [17] and extend it to time-dependent problems. Note that this approach does not employ a traditional rigorous error *estimator*, and instead uses an error *indicator*.

Together, these two ingredients render the L1-ROC scheme online-efficient (i.e. the online cost is independent of the number of degrees of freedom of the high-fidelity truth approximation) and successfully circumvent the efficiency degradation of a direct EIM approach for nonlinear and nonaffine problems. Moreover, the L1-ROC method is highly efficient during the offline stage, in that it requires minimal computation beyond the standard RBM cost of acquiring solution snapshots used to construct the reduced order space. As a consequence, the “break-even” number of simulations for the pPDE (minimum number of simulations that make the offline preparation stage worthwhile) is significantly smaller than traditional RBM. We test the algorithms on the viscous Burgers’ equation [47] and various nonlinear convection diffusion reaction equations, including the Poisson-Boltzmann equation. For all test problems, the L1-ROC is shown to have accuracy on par with the classical RBM while possessing much better efficiency due to complexity that is independent of the number of expansion terms from an EIM decomposition. Our examples include results for the steady-state and time-dependent cases of the diffusion with cubic reaction and the viscous Burgers’ equation.

1.3 Other related techniques

Popular model reduction techniques for linear time-dependent problems include Proper Orthogonal Decomposition (POD) [33], system-theoretic approaches such as balanced truncation, moment matching or Hankel norm approximation [6]. RBM stands out, for parametric problems in particular, with the availability of rigorous *a posteriori* error estimation, an easy-to-implement greedy algorithm, and the fact that it computes a number of full order solutions comparable to the theoretically smallest number for a fixed error level, the latter of which is defined by the Kolmogorov n width of the solution manifold.

The additional challenges posed by nonlinear problems are that a high dimensional reconstruction of the reduced order solution is usually needed for each evaluation of the nonlinearity. Sampling-based approximation techniques were developed to remedy this problem, including the Empirical Interpolation Method and its discrete variants [4, 25, 12, 42] and Hyper-Reduction [45, 46, 10] which are known to be equivalent to DEIM under certain conditions [21, 19]. Other approaches exist which include POD coupled with “the best interpolation points” approach [38, 23], Gappy-POD [20], Missing Point Estimation (MPE) [3] or Gauss–Newton with approximated Tensors (GNAT) [9, 10]. Most of these methods work by first identifying a subset of the important features of the nonlinear function, and then

constructing an approximation of the full solution based solely on an evaluation of these few components.

Our L1-ROC method can be viewed as adopting hyper reduction for *reduced* residual minimization. That is, instead of enforcing that the full residual is small in either a weak or strong formulation, one identifies a small set of locations and ensures that an accurate evaluation of the residual on those locations is small. This is not the first time this type of idea is explored. For example, [3, 2] uses a collocation of the original equations based on missing point interpolation and is followed by a Galerkin projection. The authors in [45] obtain the solution snapshots and collocation points through an adaptive algorithm in the finite element framework. It was also applied to nonlinear dynamical systems with randomly chosen collocation points [7]. However, the proposed L1-ROC differs from these existing works. The first distinctive feature is that the basis functions and collocation points are determined hierarchically via a greedy algorithm operating on reduced residual minimization problems that gradually increase in size. The existing approaches obtain basis functions through POD-type techniques followed by computing collocation points all at once. The second distinctive feature is that the only step during the offline process that depends on the full order model is when we calculate a new high fidelity basis. [We also note that our online solver can be considered as a Generalized EIM interpolation \[36, 35\] of the truth approximation, with a set of carefully designed interpolating functionals. Just like EIM/GEIM, Gappy-POD, and MPE, the proposed L1-ROC tends to minimize the Lebesgue constant leading to its effectiveness.](#)

The paper is organized as follows. In Section 2, we introduce our L1-ROC method. Numerical results for two test problems, in both steady-state and time-dependent modes, are shown in Section 3 to demonstrate the accuracy and efficiency of the scheme. Finally, concluding remarks are drawn in Section 4.

2 The reduced over-collocation method

In this section, we introduce the L1-ROC method for both steady state and time dependent problems. We first describe the problem we are solving. The framework of the online algorithm is then presented in Section 2.1. Specification of part of the algorithm is postponed until the introduction of the ROC offline algorithm in Section 2.2 which repeatedly calls the online solver to construct a surrogate solution space. The design of the main algorithm, the ROC approach, is detailed in Section 2.2.2. To facilitate the reading of this and the following sections, we list our notation in Table 1.

We let $\mathcal{D} \subset \mathbb{R}^p$ be the domain for a p -dimensional parameter $\boldsymbol{\mu}$, and $\Omega \subset \mathbb{R}^d$ (for $d = 1, 2$, or 3) be a bounded physical domain. Given $\boldsymbol{\mu} \in \mathcal{D}$, and a Hilbert space H , the goal is to compute $u(\cdot; \boldsymbol{\mu}) = u(\boldsymbol{\mu}) \in H$ satisfying

$$\mathcal{P}(u(\boldsymbol{x}; \boldsymbol{\mu}); \boldsymbol{\mu}) - f(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega, \quad (1)$$

or to compute time evolution of the transient problem

$$u_t + \mathcal{P}(u(\mathbf{x}; \boldsymbol{\mu}); \boldsymbol{\mu}) - f(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \quad (2)$$

with appropriate boundary (and initial) conditions. For example, for a stationary Laplace problem, the space H is typically the Sobolev space $H^1(\Omega)$. Here, $\mathcal{P}$ encodes a parametric second order partial differential operator that may include linear and nonlinear functions of $u(\mathbf{x}; \boldsymbol{\mu})$, $\nabla u(\mathbf{x}; \boldsymbol{\mu})$, and $\Delta u(\mathbf{x}; \boldsymbol{\mu})$. In the following, we will first focus on steady state problems (1) and then extend the algorithm to time dependent case (2).

To describe our algorithms, we first discretize the equation (1) by a high-fidelity scheme (termed “truth solver” in the RB literature). In this paper, we adopt finite difference methods (FDM) for that purpose. However, extension to other point-wise schemes such as spectral collocation is obvious, and to finite element methods is possible. We let $X^\mathcal{N}$ be a set of $\mathcal{N}$ collocation points on Ω at which the equation is enforced on a discrete level. The discretized equation then becomes to find $u^\mathcal{N}(X^\mathcal{N}; \boldsymbol{\mu})$, a discretization of the solution $u(\boldsymbol{\mu})$ on the grid $X^\mathcal{N}$, such that

$$\mathcal{P}_\mathcal{N}(u^\mathcal{N}(X^\mathcal{N}; \boldsymbol{\mu}); \boldsymbol{\mu}) - f(X^\mathcal{N}) = 0, \quad (3)$$

where $\mathcal{P}_\mathcal{N}(\cdot)$ is the discretized operator resulting from e.g., replacement of continuous differential operators like $\nabla u(X^\mathcal{N}; \boldsymbol{\mu})$, and $\Delta u(X^\mathcal{N}; \boldsymbol{\mu})$ with numerical approximations $\nabla_h u(X^\mathcal{N}; \boldsymbol{\mu})$, and $\Delta_h u(X^\mathcal{N}; \boldsymbol{\mu})$. With a slight abuse of notation, we let $\mathcal{N}$ denote the number of the degrees of freedom in the solver, even though the $\mathcal{N}$ points in $X^\mathcal{N}$ might include, e.g. points on a Dirichlet boundary that are not free.

2.1 Online algorithm

The online component of the L1-ROC is essentially the same as the previously-introduced reduced collocation method [13] with the critical difference being that the number of collocation points is *larger* than the number of reduced basis snapshots. This eponymous *over-collocation* feature provides additional stabilization of the online solver as we will observe in the numerical results.

To describe the online algorithm, given N selected parameters $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$, the corresponding high fidelity truth approximations $\{u_n := u^\mathcal{N}(X^\mathcal{N}; \boldsymbol{\mu}^n), 1 \leq n \leq N\}$, and $M (\geq N)$ collocation points formed from a subset of $X^\mathcal{N}$,

$$X^M = \{\mathbf{x}_*^1, \dots, \mathbf{x}_*^M\}, \quad \text{with } \mathbf{x}_*^j \text{ having index } i_j \text{ in } X^\mathcal{N}, \quad (4)$$

we are able to perform the online algorithm, which we describe next. Note that, whenever there is no confusion, we are adopting the same notation for a function and its discrete representation in the form of a vector of its values at the grid points. These vectors $\{u_n, 1 \leq n \leq N\}$ constitute the columns of a basis matrix $W_n \in \mathbb{R}^{\mathcal{N} \times n}$ for $n \in \{1, \dots, N\}$. Furthermore, we denote the corresponding reduced representation of the basis space on the set X^M , by a matrix of the following form,

$$\begin{aligned} W_{n,M} &= [u_1(X^M), \dots, u_n(X^M)] \in \mathbb{R}^{M \times n}, \quad \text{for } n = 1, \dots, N. \\ &= P_* W_n, \end{aligned}$$

$\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)$	Parameter in p -dimensional parameter domain $\mathcal{D} \subseteq \mathbb{R}^p$
Ξ_{train}	Parameter training set, a finite subset of $\mathcal{D}$
$u(\boldsymbol{\mu})$	Function-valued solution of a parameterized PDE on $\Omega \subset \mathbb{R}^d$
$\mathcal{P}(u(\boldsymbol{\mu}); \boldsymbol{\mu})$	A (nonlinear) PDE operator
K	Number of finite difference intervals per direction of the physical domain
$\mathcal{N} \approx K^d$	Degrees of freedom (DoF) of a high-fidelity PDE discretization, the “truth” solver
$X^{\mathcal{N}}$	A size- $\mathcal{N}$ (full) collocation grid
$u^{\mathcal{N}}(\boldsymbol{\mu})$	Finite-dimensional truth solution
N	Number of reduced basis snapshots, $N \ll \mathcal{N}$
$\boldsymbol{\mu}^j$	“Snapshot” parameter values, $j = 1, \dots, N$
$\hat{u}_n(\boldsymbol{\mu})$	Reduced basis solution in the n -dimensional RB space spanned by $\{u^{\mathcal{N}}(\boldsymbol{\mu}^1), \dots, u^{\mathcal{N}}(\boldsymbol{\mu}^n)\}$
$e_n(\boldsymbol{\mu})$	Reduced basis solution error, equals $u^{\mathcal{N}}(\boldsymbol{\mu}) - \hat{u}_n(\boldsymbol{\mu})$
$\Delta_N(\boldsymbol{\mu})$	A residual-based error estimate (upper bound) for $\ e_N(\boldsymbol{\mu})\ $ or an error/importance indicator
$X_r^{N-1} = \{\boldsymbol{x}_{**}^1, \dots, \boldsymbol{x}_{**}^{N-1}\}$	A size- $(N-1)$ reduced collocation grid, a subset of $X^{\mathcal{N}}$ determined based on residuals
$X_s^N = \{\boldsymbol{x}_*^1, \dots, \boldsymbol{x}_*^N\}$	An additional size- N reduced collocation grid, a subset of $X^{\mathcal{N}}$ determined based on the solutions
X^M	A reduced collocation grid of size M that is $X_r^{N-1} \cup X_s^N$
T	Final time for the time-dependent problems
Δt	Time stepsize for the time dependent problems
$\mathcal{N}_t$	Total number of time levels, i.e. $\mathcal{N}_t = \frac{T}{\Delta t}$
t^j	Time level j , $j = 1, \dots, \mathcal{N}_t$
ϵ_{tol}	Error estimate stopping tolerance in greedy sweep
Offline component	The pre-computation phase, where the reduced solver is trained using a greedy selection of snapshots from the solution space
Online component	The process of solving the offline-trained reduced problem, yielding the reduced order solution.

Table 1: Notation and terminology used throughout this article.

where the operator $P_* \in \mathbb{R}^{M \times \mathcal{N}}$ is defined as,

$$P_* = [e_{i_1}, \dots, e_{i_M}]^T,$$

with $e_i \in \mathbb{R}^{\mathcal{N} \times 1}$ the i th canonical unit vector, and i_j is as defined in (4).

Reduced approximations of the solution for any given parameter $\boldsymbol{\mu}$ are sought in the form,

$$\widehat{u}_n(\boldsymbol{\mu}) = W_n \mathbf{c}_n(\boldsymbol{\mu}).$$

The condition for obtaining the coefficients $\mathbf{c}_n(\boldsymbol{\mu})$ is (a reduced version of) equation (3)

$$\mathcal{P}_{\mathcal{N}}(W_n \mathbf{c}_n(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx f(X^{\mathcal{N}}). \quad (5)$$

Realizing that this is an over-determined system as we have in principle $n \ll \mathcal{N}$, the authors of [13] proposed a Petrov Galerkin approach or collocation on n points which produces a square system. The distinctive feature of what we propose in this paper for locating the unknown coefficients $\mathbf{c}_n(\boldsymbol{\mu})$ is to minimizing the residual of (5) on the set of nodes X^M . Namely, we seek $\mathbf{c}_n(\boldsymbol{\mu})$ by solving the following optimization problem:

$$\mathbf{c}_n(\boldsymbol{\mu}) = \underset{\omega \in \mathbb{R}^n}{\operatorname{argmin}} \| P_* (\mathcal{P}_{\mathcal{N}}(W_n \omega; \boldsymbol{\mu}) - f(X^{\mathcal{N}})) \|_{\mathbb{R}^M}. \quad (6)$$

We note that this is a (potentially) nonlinear system of equations for $\mathbf{c}_n$, with $\nabla_h \widehat{u}_n(\boldsymbol{\mu})$ and $\Delta_h \widehat{u}_n(\boldsymbol{\mu})$ computed on the full grid and then evaluated on the reduced grid X^M according to

$$\begin{aligned} \nabla_h \widehat{u}_n(\boldsymbol{\mu}) &= P_* [(\nabla_h u_1), \dots, (\nabla_h u_n)] \mathbf{c}_n(\boldsymbol{\mu}), \\ \Delta_h \widehat{u}_n(\boldsymbol{\mu}) &= P_* [(\Delta_h u_1), \dots, (\Delta_h u_n)] \mathbf{c}_n(\boldsymbol{\mu}). \end{aligned}$$

We use iterative methods, such as Newton's method, to solve for the coefficients $\mathbf{c}_n(\boldsymbol{\mu})$ during the online phase. The collocation nature of this scheme allows for solving this system with a cost only dependent on M and n even when $\mathcal{P}_{\mathcal{N}}$ is nonlinear and nonaffine. In particular, it is independent of the degrees of freedom $\mathcal{N}$ of the underlying truth solver. Indeed, the next section describes the offline procedure where the N selected parameters $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ are identified sequentially through a greedy algorithm. Once these parameters are selected, we precompute as many quantities as possible so that a minimal update is performed at each iteration of the iterative method. The online procedure of the nonlinear solve for obtaining $\mathbf{c}_n(\boldsymbol{\mu})$ from equation (6) involves:

- 1) realizing/updating $W_{n,M} \mathbf{c}_n$, $\nabla_h(W_{n,M}) \mathbf{c}_n$, and $\Delta_h(W_{n,M}) \mathbf{c}_n$ at each iteration taking $O(Mn)$ operations;
- 2) calculating the forcing term $f(X^M)$ taking $O(M)$ operations; and
- 3) solving the reduced linear systems at each iteration of the nonlinear solve taking $O(n^3)$ operations.

2.2 Offline algorithm

In this section, we describe the offline procedure of the reduced over collocation framework based on the L1-approach proposed in [17]. The remaining ingredients of the offline procedure are identical to a traditional RBM algorithm [44, 27, 43, 29].

2.2.1 A greedy algorithm based on an L1 importance indicator

We first briefly describe the procedure for selecting the representative parameters $\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N$ that identify the solution space W_N . RBM utilizes a greedy scheme to iteratively construct W_N relying on efficiently-computable error estimates that quantify the discrepancy between the dimension- n RBM surrogate $\hat{u}_n(\boldsymbol{\mu})$ and the truth solution $u^{\mathcal{N}}(\boldsymbol{\mu})$. Denoting such an estimate as $\Delta_n(\boldsymbol{\mu})$, it traditionally satisfies $\Delta_n(\boldsymbol{\mu}) \geq \|\hat{u}_n(\boldsymbol{\mu}) - u^{\mathcal{N}}(\boldsymbol{\mu})\|$. Assuming existence and feasible computability of this error estimate, the greedy procedure for constructing W_N then starts by selecting the first parameter $\boldsymbol{\mu}^1$ randomly from Ξ_{train} (a discretization of the parameter domain $\mathcal{D}$) and obtaining its corresponding high-fidelity truth approximation $u^{\mathcal{N}}(\boldsymbol{\mu}^1)$ to form a (one-dimensional) RB space given by the range of $W_1 = [u^{\mathcal{N}}(\boldsymbol{\mu}^1)]$. Next, we obtain an RB approximation $\hat{u}_n(\boldsymbol{\mu})$ for each parameter in Ξ_{train} together with an error bound $\Delta_n(\boldsymbol{\mu})$. The greedy choice for the $(n+1)$ th parameter ($n = 1, \dots, N-1$) is made and the RB space augmented by

$$\boldsymbol{\mu}^{n+1} = \underset{\boldsymbol{\mu} \in \Xi_{\text{train}}}{\operatorname{argmax}} \Delta_n(\boldsymbol{\mu}), \quad W_{n+1} = [W_n \quad u^{\mathcal{N}}(\boldsymbol{\mu}^{n+1})] \quad (7)$$

The design and efficient implementation of the error bound Δ_n is usually accomplished with a residual-based *a posteriori* error estimate from the truth discretization. Mathematical rigor and computational efficiency of this estimate are crucial for the accuracy of the reduced basis solution and its efficiency relative to the truth approximation. When $\mathcal{P}(u; \boldsymbol{\mu})$ is a linear operator, the Riesz representation theorem and a variational inequality imply that Δ_n can be taken as

$$\Delta_n^R(\boldsymbol{\mu}) = \frac{\|f - \mathcal{P}_{\mathcal{N}}(\hat{u}_n; \boldsymbol{\mu})\|_2}{\sqrt{\beta_{LB}(\boldsymbol{\mu})}},$$

where $\beta_{LB}(\boldsymbol{\mu})$ is a lower bound for the smallest eigenvalue of $P_{\mathcal{N}}(\boldsymbol{\mu})^T P_{\mathcal{N}}(\boldsymbol{\mu})$ with $P_{\mathcal{N}}(\boldsymbol{\mu})$ being the matrix corresponding to the discretized linear operator $\mathcal{P}_{\mathcal{N}}(\cdot; \boldsymbol{\mu})$. Thus, Δ_n^R defined above is a rigorous bound (with the R -superscript denoting it is based on the full residual).

Deriving a counterpart for this estimator for a general nonlinear equation is far from trivial. Moreover, even for linear equations, the robust evaluation of the residual norm in the numerator is delicate [11, 17]. One typically resorts to an offline-online decomposition to retain efficiency which usually means application of EIM for nonlinear or nonaffine terms. This complication degrades, sometimes significantly [5, 37], the online efficiency due to the large number of resulting EIM terms. What exacerbates the situation further is that the (parameter-dependent) stability factor $\beta_{LB}(\boldsymbol{\mu})$ must be calculated by a computationally efficient procedure such as the successive constraint method [31, 30]. For these reasons, we

adopt the following empirical alternative, an *importance indicator* proposed in [17], in place of Δ_n^R :

$$\Delta_n^L(\boldsymbol{\mu}) = \|\mathbf{c}_n(\boldsymbol{\mu})\|_1.$$

The L -superscript denotes that it is based on the L1-norm making our scheme L1-based. We note that this is not an error estimator because Δ_n^L does not decrease as we increase n since $\Delta_n^L(\boldsymbol{\mu}^i) = 1$ for $i \in \{1, \dots, n\}$. Nevertheless, we demonstrate that it is a reliable quantity to monitor when deciding which representative parameters $\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N$ will form the surrogate space. We finish this subsection by pointing out that the calculation of Δ_n^L is independent of $\mathcal{N}$ while naive approaches to evaluate the traditional estimator Δ_n^R for nonlinear problems would depend on $\mathcal{N}$. This difference leads to the dramatic efficiency gain of the L1-ROC, as numerically confirmed in Section 3.

Algorithm 1 Offline: construction of W_N and the collocation set $X^{2N-1} = X_s^N \cup X_r^{N-1}$.

- 1: Choose $\boldsymbol{\mu}^1$ randomly from Ξ_{train} , compute $u_1 := u^\mathcal{N}(X^\mathcal{N}; \boldsymbol{\mu}^1)$.
 - 2: Compute $\mathbf{x}_*^1 = \operatorname{argmax}_{\mathbf{x} \in X^\mathcal{N}} |u_1|$, define $\xi_1 = u_1/u_1(\mathbf{x}_*^1)$. Let i_1 be the index of $\mathbf{x}_*^1$ and $P_* = [e_{i_1}]^T$.
 - 3: Initialize $m = n = 1$, $X^m = X_s^n = [\mathbf{x}_*^1]$, $W_1 = \{\xi_1\}$, $W_{1,m} = P_* W_1$, and $X_r^0 = \emptyset$.
 - 4: **For** $n = 2, \dots, N$
 - 5: Solve $\mathbf{c}_{n-1}(\boldsymbol{\mu})$ with W_{n-1}, P_* and calculate $\Delta_{n-1}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \Xi_{\text{train}}$.
 - 6: Find $\boldsymbol{\mu}^n = \operatorname{argmax}_{\boldsymbol{\mu} \in \Xi_{\text{train}} \setminus \{\boldsymbol{\mu}^i, i=1, \dots, n-1\}} \Delta_{n-1}(\boldsymbol{\mu})$ and solve for $\xi_n := u^\mathcal{N}(X^\mathcal{N}; \boldsymbol{\mu}^n)$.
 - 7: Compute an interpolatory residual for ξ_n : find $\{\alpha_j\}$ and let $\xi_n = \xi_n - \sum_{j=1}^{n-1} \alpha_j \xi_j$ so that $\xi_n(X_s^{n-1}) = 0$.
 - 8: Find $\mathbf{x}_*^n = \operatorname{argmax}_{\mathbf{x} \in X^\mathcal{N}/X^m} |\xi_n(\mathbf{x})|$, $\xi_n = \xi_n/\xi_n(\mathbf{x}_*^n)$, and let $X_s^n = X_s^{n-1} \cup \{\mathbf{x}_*^n\}$, and i_1 be the index of $\mathbf{x}_*^n$.
 - 9: Form the full residual vector $r_{n-1} = \mathcal{P}_\mathcal{N}(\widehat{u}_{n-1}(\boldsymbol{\mu}^n); \boldsymbol{\mu}^n) - f(X^\mathcal{N})$ and compute its interpolatory residual: find $\{\alpha_j\}$ and let $r_{n-1} = r_{n-1} - \sum_{j=1}^{n-2} \alpha_j r_j$ so that $r_{n-1}(X_r^{n-2}) = 0$. Find $\mathbf{x}_{**}^{n-1} = \operatorname{argmax}_{\mathbf{x} \in X^\mathcal{N}/\{X^m, \mathbf{x}_*^n\}} |r_{n-1}(\mathbf{x})|$. Let $r_{n-1} = r_{n-1}/r_{n-1}(\mathbf{x}_{**}^{n-1})$, and $X_r^{n-1} = X_r^{n-2} \cup \{\mathbf{x}_{**}^{n-1}\}$ and i_2 is the index of $\mathbf{x}_{**}^{n-1}$.
 - 10: Update $W_n = \{W_{n-1}, \xi_n\}$, $m = 2n - 1$, $X^m = X_s^n \cup X_r^{n-1}$, $P_* = P_* \cup [e_{i_1}, e_{i_2}]^T$.
 - 11: **End For**
-

2.2.2 Construction of the reduced over-collocation set X^M

Let us now describe how we determine the reduced collocation set X^M to complete the offline algorithm. Toward that end, we first describe the construction of two sets. The first one, denoted by X_s^N , consists of the maximizers from the EIM procedure used on the orthonormalized columns of W_N , which are computed as pivots from an LU decomposition. Realizing the importance of controlling the residuals when solving equations, we need to represent the residuals well on the reduced grid. For that purpose, we introduce a second set of points, and examine the residual of the RB solution at the chosen $\boldsymbol{\mu}^n$ when only $n - 1$

basis elements are used,

$$r_{n-1} = \mathcal{P}_N(\widehat{u}_{n-1}(\boldsymbol{\mu}^n); \boldsymbol{\mu}^n) - f(X^N), \quad n \in \{2, \dots, N\}. \quad (8)$$

We next take these $N - 1$ residual vectors and perform an EIM procedure on them. The $N - 1$ maximizers from this procedure form the second set which is denoted X_r^{N-1} . The reduced collocation approach in [13] is a specialization that takes $M = N$, $X_r^{N-1} = \emptyset$, and $X^M = X_s^N$. The resulting $M = N$ reduced scheme can be unstable particularly when high accuracy (i.e. large N) of the reduced solution is desired. It can be resolved in special cases by an analytical preconditioning approach [14]. The second obvious choice of X^M is to append X_r^{N-1} with one more point such as the maximizer of the first basis. Numerical tests (not reported in this paper) also reveal instability of this scheme.

The stabilization mechanism and name of the reduced over-collocation methods, outlined in Algorithm 1, come from the fact that we combine these two choices by taking

$$X^M = X_s^N \cup X_r^{N-1},$$

and solving a least squares problem on the reduced level by collocating on about twice as many points as the number of basis in the RB space. Note that the first basis function has no accompanying residual vector (8), so that from the second onward there are two collocation points selected whenever a new parameter is identified by the greedy algorithm. Here $M \leq 2N - 1$ since, even though $|X_s^N| = N$ and $|X_r^{N-1}| = N - 1$ are guaranteed by the EIM procedures, we don't necessarily have mutual exclusivity between X_s^N and X_r^{N-1} .

Remark 1. *This framework allows for more points being added. For example, when we make the greedy choice $\boldsymbol{\mu}^n$ (the maximizer of the error estimator), we can record the parameter $\tilde{\boldsymbol{\mu}}_n$ attaining the second largest error estimator and generate another set of residuals when only $n - 1$ basis elements are used,*

$$\tilde{r}_{n-1} = \mathcal{P}_N(\widehat{u}_{n-1}(\tilde{\boldsymbol{\mu}}^n); \tilde{\boldsymbol{\mu}}^n) - f(X^N), \quad n \in \{2, \dots, N\}.$$

We thus have (up to) $3N - 2$ ROC points. Numerical results in Section 3 show no significant improvement over $2N - 1$ points for the scalar nonlinear equations of concern in this paper. However, we expect that this strategy will be necessary for some systems of coupled nonlinear equations, a topic of ongoing investigation.

2.3 Extension of L1-ROC for time dependent problems

For the time-dependent problem (2), the semi-discretized L1-ROC solver remains identical to the steady-state case. That is, we seek the reduced approximation of the solution for any given parameter $\boldsymbol{\mu}$ in the form of

$$\widehat{u}_n(\boldsymbol{\mu}, t) = W_n \mathbf{c}_n(\boldsymbol{\mu}, t).$$

The unknown coefficient vector $\mathbf{c}_n(\boldsymbol{\mu}, t) \in \mathbb{R}^{n \times 1}$ is obtained by solving the following optimization problem:

$$\mathbf{c}_n(\boldsymbol{\mu}, t) = \underset{\omega \in \mathbb{R}^n}{\operatorname{argmin}} \| P_* (W_n \omega_t + \mathcal{P}_{\mathcal{N}}(W_n \omega; \boldsymbol{\mu}) - f(X^{\mathcal{N}})) \|_{\mathbb{R}^M}. \quad (9)$$

To discretize in time, our L1-ROC aligns with the parameter-time greedy framework [25, 26], as opposed to POD [33, 38] or POD-greedy [24]. We discretize the time and denote the (full) set of temporal nodes as $\mathcal{T}_f := \{t_i : i = 0, \dots, \mathcal{N}_t\}$ with t_0 being the initial time and $\mathcal{N}_t = T/\Delta t$ where Δt is the temporal step-size. We extend the L1-based importance indicator of [17] to the time-dependent case here. Toward that end, we define a *reduced* set of temporal nodes $\mathcal{T}_r$ that starts from the empty set and is gradually enriched in the greedy algorithm.

To initiate the reduced solver construction we start with a deterministically or randomly chosen $\boldsymbol{\mu}^1$ (similar to the steady-state case) and invoke the truth solver to obtain the snapshots $\{u^{\mathcal{N}}(t_i, x; \boldsymbol{\mu}^1)\}_{i=0}^{\mathcal{N}_t}$. $\mathcal{T}_r$ is initiated by the time instant when the corresponding snapshot has the largest variation. That is,

$$\mathcal{T}_r = \{t_{\boldsymbol{\mu}^1}^1\} \text{ where } t_{\boldsymbol{\mu}^1}^1 = \underset{t \in \mathcal{T}_f}{\operatorname{argmax}} \left(\max_{x \in X^{\mathcal{N}}} u^{\mathcal{N}}(t, x; \boldsymbol{\mu}^1) - \min_{x \in X^{\mathcal{N}}} u^{\mathcal{N}}(t, x; \boldsymbol{\mu}^1) \right).$$

The RB space W_1 is initiated with $u^{\mathcal{N}}(t_{\boldsymbol{\mu}^1}^1, x; \boldsymbol{\mu}^1)$. The (first) collocation point is set to be the EIM point of this first basis, i.e. the spatial maximizer of $|u^{\mathcal{N}}(t_{\boldsymbol{\mu}^1}^1, x; \boldsymbol{\mu}^1)|$,

$$\mathbf{x}_*^1 = \underset{x \in X^{\mathcal{N}}}{\operatorname{argmax}} |u^{\mathcal{N}}(t_{\boldsymbol{\mu}^1}^1, x; \boldsymbol{\mu}^1)|.$$

Once these ingredients are in place after the first pair $(\boldsymbol{\mu}^1, t_{\boldsymbol{\mu}^1}^1)$ is determined, we can solve (a temporally discretized version of) the reduced problem (9) for every $\boldsymbol{\mu} \in \Xi_{\text{train}}$. Similar to the traditional greedy algorithm, the next step is to determine the subsequent $(\boldsymbol{\mu}, t)$ pairs. Our greedy algorithm manifests itself in the following three aspects:

- **Greedy in $\boldsymbol{\mu}$:** We define the following importance indicator for each $\boldsymbol{\mu}$ after its corresponding (reduced) solver of (9) is performed,

$$\Delta_n^{Lt}(\boldsymbol{\mu}) := \max_{t \in \mathcal{T}_r} \{\|\mathbf{c}_n(\boldsymbol{\mu}, t)\|_1\}. \quad (10)$$

We note that: 1) the maximization is done only on the reduced temporal grid $\mathcal{T}_r$ which is much smaller than the full temporal grid $\mathcal{T}_f$; 2) the signature feature of the L1-based approach carries over to the time-dependent case in that the indicator requires nothing more than the reduced solution coefficients. Our greedy choice for the $\boldsymbol{\mu}$ -component of the $(\boldsymbol{\mu}, t)$ pair is through maximizing $\Delta_n^{Lt}(\boldsymbol{\mu})$ over the training set Ξ_{train} :

$$\boldsymbol{\mu}^{n+1} = \underset{\boldsymbol{\mu} \in \Xi_{\text{train}}}{\operatorname{argmax}} \Delta_n^{Lt}(\boldsymbol{\mu}).$$

- **Greedy in t :** Next, the t -component of the $(\boldsymbol{\mu}, t)$ pair is determined and the set $\mathcal{T}_r$ enriched with a new temporal node through a greedy choice as well. Given the greedy choice $\boldsymbol{\mu}^{n+1}$ and the *reduced* solution $\hat{u}_n(\boldsymbol{\mu}^{n+1}, t) = W_n \mathbf{c}_n(\boldsymbol{\mu}^{n+1}, t)$ for all time levels $t \in \mathcal{T}_f$, we compute the *full* residual vectors $r_n \in \mathbb{R}^{N \times 1}$ for this $\boldsymbol{\mu}^{n+1}$. The greedy t -choice is given by

$$t_{\boldsymbol{\mu}^{n+1}}^{k_{\boldsymbol{\mu}^{n+1}}} := \operatorname{argmax}_{t \in \mathcal{T}_f} \{ \varepsilon(t; \boldsymbol{\mu}) := \|r_n(t; \boldsymbol{\mu}^{n+1})\|_\infty \}, \text{ and } \mathcal{T}_r = \mathcal{T}_r \cup \{t_{\boldsymbol{\mu}^{n+1}}^{k_{\boldsymbol{\mu}^{n+1}}}\}. \quad (11)$$

Here, $k_{\boldsymbol{\mu}^{n+1}} \geq 1$ is introduced to account for the possibility that multiple temporal nodes might be selected for the same $\boldsymbol{\mu}$, at different rounds of the greedy algorithm. We note in particular that, consistent with typical greedy scheme, we choose one maximizer (as opposed to multiple maximizers) in (11). However, as we proceed with building up the reduced solution space, the same $\boldsymbol{\mu}$ (and a different temporal node) may be chosen by the greedy algorithm at a later step due to the lack of resolution of its corresponding temporal history.

- **X^M expansion:** Once a new greedy pair $(\boldsymbol{\mu}^{n+1}, t_{\boldsymbol{\mu}^{n+1}}^{k_{\boldsymbol{\mu}^{n+1}}})$ is fixed, we solve for the truth approximations $u(t, X^N; \boldsymbol{\mu}^{n+1})$ for $t \leq t_{\boldsymbol{\mu}^{n+1}}^{k_{\boldsymbol{\mu}^{n+1}}}$. The expansion of X^M by two more collocation points, with one from the EIM procedure of the solution $u(t_{\boldsymbol{\mu}^{n+1}}^{k_{\boldsymbol{\mu}^{n+1}}}, X^N; \boldsymbol{\mu}^{n+1})$ and the other from that of the residual $r_n(t_{\boldsymbol{\mu}^{n+1}}^{k_{\boldsymbol{\mu}^{n+1}}}; \boldsymbol{\mu}^{n+1})$, is identical to the steady state case.

The full offline algorithm is seen in Algorithm.2.

3 Numerical results

In this section, we present the numerical results of the L1-ROC method applied to the nonlinear steady-state and time-dependent problems, in Sections 3.1 and 3.2 respectively. The equations we test in each subsection include the classical viscous Burgers' equation and nonlinear convection diffusion reaction equations. [We test three methods:](#) 1) [the traditional RBM with the residual-based error estimator \$\Delta_n^R\$ and Petrov-Galerkin projection as the reduced solver \(i.e. \(6\) without \$P_*\$ \);](#) 2) [the semi-traditional RBM with the residual-based error estimator \$\Delta_n^R\$ and the ROC solver \(6\);](#) and finally 3) [the proposed L1-ROC method with the L1-based error indicator \$\Delta_n^L\$ and the ROC solver \(6\).](#)

3.1 L1-ROC for steady-state nonlinear problems

3.1.1 Viscous Burgers' equation

First, we show the results of our algorithm applied to the one-dimensional (viscous) Burgers' equation,

$$\begin{aligned} uu_x &= \boldsymbol{\mu} u_{xx}, \\ u(x = -1) &= 1, \quad u(x = 1) = -1. \end{aligned} \quad (12)$$

Algorithm 2 L1-ROC algorithm for time dependent problems

- 1: Choose $\boldsymbol{\mu}^1$, and set $k_{\boldsymbol{\mu}^1} = 1$ the first temporal node to be $t_{\boldsymbol{\mu}^1}^{k_{\boldsymbol{\mu}^1}} = \operatorname{argmax}_{t \in \mathcal{T}_f} (\max_x u^\mathcal{N}(t, x; \boldsymbol{\mu}^1) - \min_x u^\mathcal{N}(t, x; \boldsymbol{\mu}^1))$. Define $\xi_1 := u^\mathcal{N}(t_{\boldsymbol{\mu}^1}^{k_{\boldsymbol{\mu}^1}}, X^\mathcal{N}; \boldsymbol{\mu}^1)$.
 - 2: Find $\mathbf{x}_*^1 = \operatorname{argmax}_{\mathbf{x} \in X^\mathcal{N}} |\xi_1|$, and let $P_* = [e_{i_1}]^T$, where i_1 is the index of $\mathbf{x}_*^1$.
 - 3: Initialize $m = n = 1$, $X^m = X_s^n = \{\mathbf{x}_*^1\}$, $W_1 = \{\xi_1\}$, $W_{1,m} = P_* W_1$, and $X_r^0 = \emptyset$.
 - 4: **For** $n = 2, \dots, N$
 - 5: Solve the reduced problem for $\mathbf{c}_{n-1}(\boldsymbol{\mu}, t_k)$.
 - 6: Find $\boldsymbol{\mu}^n = \operatorname{argmax}_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_{n-1}^{Lt}(\boldsymbol{\mu})$, and a new temporal node $t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}} = \operatorname{argmax}_{t \in \mathcal{T}_f} \varepsilon(t; \boldsymbol{\mu}^n)$.
 - 7: Solve $\xi_n = u^\mathcal{N}(t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}}, X^\mathcal{N}; \boldsymbol{\mu}^n)$.
 - 8: Compute an interpolatory residual for ξ_n : find $\{\alpha_j\}$ and let $\xi_n = \xi_n - \sum_{j=1}^{n-1} \alpha_j \xi_j$ so that $\xi_n(X_s^{n-1}) = 0$. Find $\mathbf{x}_*^n = \operatorname{argmax}_{\mathbf{x} \in X^\mathcal{N}/X^m} |\xi_n|$, $\xi_n = \xi_n / \xi_n(\mathbf{x}_*^n)$. Let $X_s^n = X_s^{n-1} \cup \{\mathbf{x}_*^n\}$, and i_1 be the index of $\mathbf{x}_*^n$.
 - 9: Form the full residual vector $r_{n-1} = (\hat{u}_{n-1})_t(t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}}; \boldsymbol{\mu}^n) + \mathcal{P}_\mathcal{N}(X^\mathcal{N}, \hat{u}_{n-1}(t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}}; \boldsymbol{\mu}^n); \boldsymbol{\mu}^n) - f(X^\mathcal{N}, t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}})$. Compute an interpolatory residual r_{n-1} : find $\{\alpha_j\}$ and let $r_{n-1} = r_{n-1} - \sum_{j=1}^{n-2} \alpha_j r_j$ so that $r_{n-1}(X_r^{n-2}) = 0$. Find $\mathbf{x}_{**}^n = \operatorname{argmax}_{\mathbf{x} \in X^\mathcal{N}/\{X^m, \mathbf{x}_*^n\}} |r_{n-1}|$. Let $r_{n-1} = r_{n-1} / r_{n-1}(\mathbf{x}_{**}^n)$, and $X_r^{n-1} = X_r^{n-2} \cup \{\mathbf{x}_{**}^n\}$. i_2 is the index of $\mathbf{x}_{**}^n$.
 - 10: Update $W_n = \{W_{n-1}, \xi_n\}$, $m = 2n - 1$, $X^m = X_s^n \cup X_r^{n-1}$, $P_* = [P_*; (e_{i_1})^T; (e_{i_2})^T]$.
 - 11: **End For**
-

Here the viscosity parameter $\boldsymbol{\mu}$ varies on the interval $\mathcal{D} = [0.05, 1]$. The computational domain $[-1, 1]$ is divided uniformly into $\mathcal{N} + 1$ intervals with grid points denoted by

$$\{x_0, x_1, \dots, x_{\mathcal{N}+1}\}.$$

With $h = \frac{2}{\mathcal{N}+1}$, the following finite difference discretization based on the conservative form of equation (12), $\left(\frac{u^2}{2}\right)_x - \boldsymbol{\mu} u_{xx} = 0$, is then used

$$\frac{u_{i+1}^2 - u_{i-1}^2}{4h} - \boldsymbol{\mu} \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = 0, \quad i \in \{1, \dots, \mathcal{N}\}. \quad (13)$$

This leads to a nonlinear truth solver of size $\mathcal{N}$ to resolve (12). The parameter domain $\mathcal{D}$ is sampled 50 times logarithmically spaced to form the training set for the Offline procedure.

We test our method on a subset of Ξ_{test} of $\mathcal{D}$ that has empty intersection with the training set Ξ_{train} . We compute the relative errors $E(n)$ over all $\boldsymbol{\mu}$ in Ξ_{test} of the reduced basis solution using n basis functions, $\hat{u}_n(\boldsymbol{\mu})$, in comparison to the high fidelity truth approximation. That is,

$$E(n) = \max_{\boldsymbol{\mu} \in \Xi_{\text{test}}} \left\{ \frac{\|u(\boldsymbol{\mu}) - \hat{u}_n(\boldsymbol{\mu})\|_\infty}{\|u\|_{L^\infty(\Xi_{\text{test}}, L^\infty(\Omega))}} \right\} \quad (14)$$

where

$$\|u\|_{L^\infty(\Xi_{\text{test}}, L^\infty(\Omega))} = \max_{\boldsymbol{\mu} \in \Xi_{\text{test}}} \|u(\boldsymbol{\mu})\|_\infty.$$

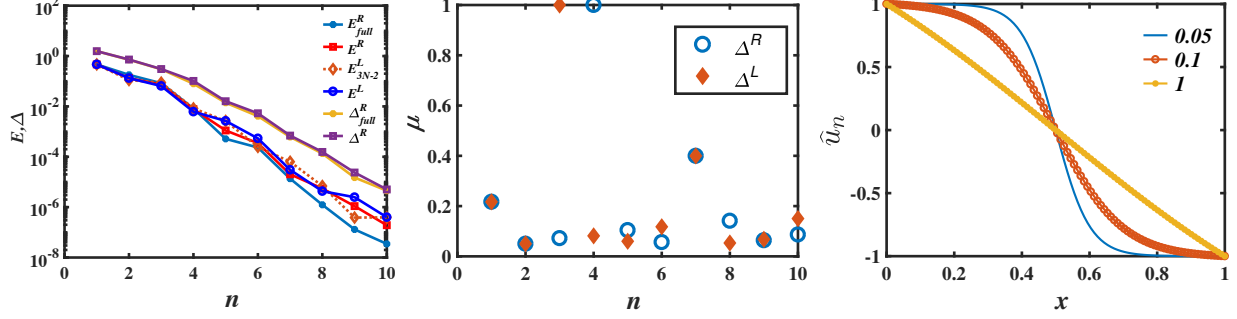


Figure 1: Steady viscous Burgers' result. (Left) Histories of convergence for the error and error estimator for the traditional residual-based RBM and proposed L1-ROC. Here, E^R and E^L refer to the $E(n)$ in (14) with the reduced solution $\hat{u}_n$ constructed by following the residual-based error estimator Δ^R and L1-based importance indicator Δ^L , respectively. (Middle) Distribution of selected parameters μ^n , using estimator Δ^R and Δ^L , as a function of n . (Right) Sample RB solutions at three parameter values themselves. Note that Δ^L does not decay to zero for large n , but such decay is not expected or needed for this function.

Error curves and the distribution of the first $N = 10$ selected parameters with $\mathcal{N} = 100$ are showed in Figure 1. It shows a clear exponential convergence as n increases and a concentration of the selected μ values toward the lower end of the parameter domain. The error convergence of the L1-ROC (E^L) matches that of the more expensive semi-traditional RB (E^R and Δ^R) which is comparable to the most expensive traditional RB (E_{full}^R and Δ_{full}^R). We also test the effect of including more ROC points (E_{3N-2}^L) which shows no noticeable improvement over the $2N - 1$ scheme. We note that the distributions of chosen parameters between the traditional residual-based scheme and the nascent L1-based scheme are very much similar which underscores the reliability of the new L1-ROC approach.

3.1.2 Nonlinear reaction diffusion equations

Here we consider the following cubic reaction diffusion,

$$\begin{aligned} -\mu_2 \Delta u + u(u - \mu_1)^2 &= f(\mathbf{x}) \text{ in } \Omega := [-1, 1] \times [-1, 1], \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \quad (15)$$

We take $f(\mathbf{x}) = 100 \sin(2\pi x_1) \cos(2\pi x_2)$, and the parameter domain $\mathcal{D}$ is set to be $[0.2, 5] \times [0.2, 2]$. $\mathcal{D}$ is discretized by a 128×64 uniform tensorial grid. Denoting the step size along the μ_1 direction by h_1 , and the other by h_2 , we specify the training set and test set as follows,

$$\begin{aligned} \Xi_{\text{train}} &= (0.2 : 4h_1 : 5) \times (0.2 : 4h_2 : 2), \\ \Xi_{\text{test}} &= ((0.2 + 2h_1) : 4h_1 : (5 - 2h_1)) \times ((0.2 + 2h_2) : 4h_2 : (2 - 2h_2)), \end{aligned}$$

where $(a : h : b)$ denotes an equidistant mesh over $[a, b]$ with stepsize h . Note in particular that the two sets defined above are disjoint. The nonlinear solver, based on the 5-point stencil

with $\sqrt{N}$ interior points at each direction of Ω , for the high fidelity truth approximation linearizes, at the $(\ell + 1)^{\text{th}}$ iteration, the equation according to

$$-\mu_2 \Delta u^{(\ell+1)} + g'(u^{(\ell)})u^{(\ell+1)} = g'(u^{(\ell)})u^{(\ell)} - g(u^{(\ell)}, \mu_1) + f(\mathbf{x}) \quad (16)$$

where $g(u; \mu_1) = u(u - \mu_1)^2$.

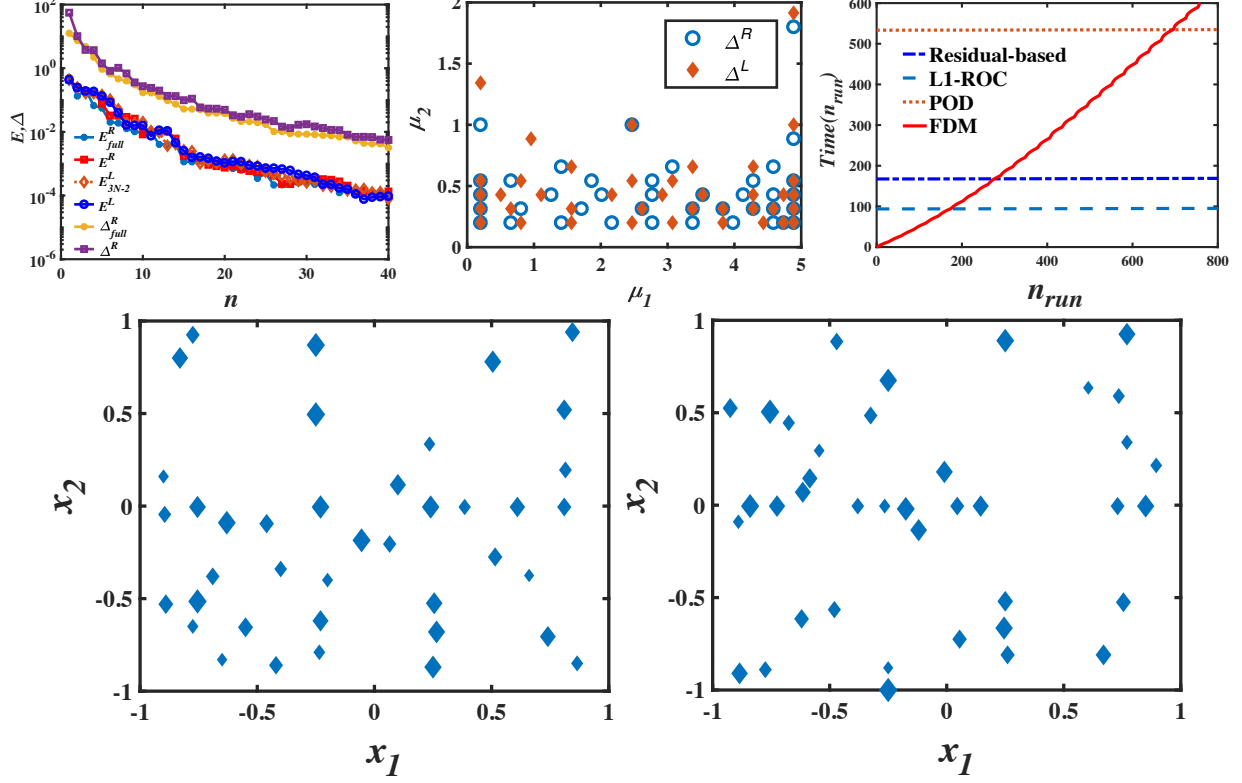


Figure 2: Cubic reaction diffusion result. Top row:(Left) comparison of the histories of convergence with $\sqrt{N} = 400$ for the errors and the error estimator for the ROC method. Here, E^R and E^L refer to the $E(n)$ in (14) with the reduced solution $\hat{u}_n$ constructed by following the residual-based error estimator Δ^R and L1-based importance indicator Δ^L , respectively. (Middle) Selected $N(= 40)$ parameters of the ROC method for residual-based and L1-based approaches. (Right) cumulative runtime of the FDM, the residual-based, and L1-based RBM. Bottom row: selected 40 collocation points X_s^M from solutions (Left) and 39 collocation points X_r^M from residual vector (Right).

Relative errors of the RB solution $E(n)$ with $K = \sqrt{N} = 400$ are displayed in Figure 2 top left. Steady exponential convergence is again observed for the L1-ROC method. It matches those of the more expensive semi-traditional (E^R and Δ^R) and traditional RB (E_{full}^L and Δ_{full}^R) methods. Again, raising the number of ROC points has no noticeable impact as the L1-ROC already matches the traditional RBM. The set of selected parameters are shown in Figure 2 top middle, while the collocation points are shown on the bottom row.

We note again that the distributions of chosen parameters between the traditional residual-based scheme and the more nascent L1-based scheme are quite similar for this example underscoring the reliability of the L1-ROC approach.

Lastly, we showcase the vast saving of the offline time for the L1-ROC approaches. Toward that end, the comparison in cumulative computation time for the residual-based, L1-ROC, and the high fidelity truth approximations is shown in Figure 2 top right. The initial nonzero start of the L1-ROC corresponds to the offline time. We observe that, when $n_{\text{run}} > 172$, L1-ROC starts to save time in comparison to repeated runs of the truth solver. In that regard, the residual-based ROC is effective when $n_{\text{run}} > 276$ with $\sqrt{N} = 200$. The difference in this “break-even” point is because the overhead cost, devoted to calculating Δ_n^L (for L1-ROC), is significantly less than that for Δ_n^R . The latter involves (an offline-online decomposition of) the calculation of the full residual norm while the former only requires, in the L1-ROC case, obtaining an $N \times 1$ vector and evaluating its L1-norm. It is worth noting that the “break-even” number of runs is insensitive to $\sqrt{N}$. Though L1-ROC has a much more efficient offline procedure than the residual-based ROC, their online time for any new parameter is comparable, see Table 2. We observe that the L1-ROC method accelerates the iterative truth solver by $2000 \sim 50000$ times. The results also confirm that time consumption of the online ROC methods is independent of $K = \sqrt{N}$. In order to demonstrate the time savings more intuitively, we present the online calculation time for the different algorithms in two different parameter regimes. The first regime is when μ_1 is large and μ_2 small, in particular we choose $\mu_1 = 4.55, \mu_2 = 0.42$. The second regime has the relative sizes reversed. The reduced solver requires 27 iterations for the nonlinear system in the first regime, while only requiring 8 iterations in the second regime. Therefore, the full-order time consumption seems very different. However, Table 2 does indicate a speedup range of $3000 \sim 17000$ when $\sqrt{N} = 400, 800$.

(μ_1, μ_2)	K	Residual-based ROC	L1-ROC	Direct FDM
(4.55, 0.42)	200	0.003150	0.003159	2.310034
	400	0.003067	0.003136	11.779558
	800	0.003258	0.003162	53.727031
(1, 1.82)	200	0.001125	0.001060	0.662095
	400	0.001141	0.001205	3.338956
	800	0.001207	0.001261	15.173460

Table 2: Online computational times (seconds) with different grid sizes K , when $N = 40$.

Remark 2. *The L1-ROC works well for other nonlinear convection diffusion reaction equations. For example, we tested the dimensionless nonlinear nonaffine Poisson-Boltzmann equation*

$$D\nabla^2 u = \sinh u + g(\mathbf{x}), \text{ with } g(\mathbf{x}) = \exp[-50((x_1 - 0.2)^2 + (x_2 + 0.1)^2)] \quad (17)$$

modeling a source distribution centered at $(0.2, -0.1)$. The parameters are diffusion coefficient D and the voltage differential V at the boundary. The authors have previously designed

a RBM for this equation [32]. However, due to the desire to avoid applying EIM directly, we observed limited speedup (less than one order of magnitude). With L1-ROC, we achieved a speedup factor of up to four orders of magnitude, see Table 3. This significant progress underscores the power of the L1-ROC approach. In addition, we tested an equation with nonlinear convection term

$$-\mu_2 \Delta u + u (\|\nabla u\| + \mu_1)^{1.5} = f(x). \quad (18)$$

The L1-ROC works equally well on this example, see Table 4.

K	Residual-based ROC	L1-ROC	Direct FDM
200	0.000678	0.000688	1.439812
400	0.000770	0.000646	6.492029
800	0.000728	0.000625	33.722112

Table 3: Online computational times (in seconds) for the Poisson-Boltzmann equation (17) at different grid sizes K , when $V = 3.85$, $D = 0.15^2$, $N = 30$.

K	Residual-based ROC	L1-ROC	Direct FDM
200	0.000422	0.000428	0.569732
400	0.000397	0.000410	2.838783
800	0.000424	0.000425	12.582593

Table 4: Online computational times (seconds) for the nonlinear convection diffusion equation (18) at different grid sizes K , when $N = 20$, $\mu_1 = 32$, $\mu_2 = 3$.

3.1.3 Numerical comparison with POD and random generation

To further establish numerically the reliability of the L1-ROC algorithm, we compare it with two alternative methods of building the reduced basis space. On one end, the proper orthogonal decomposition (POD) [22, 33, 48, 34] based on an exhaustive selection of snapshots (i.e. we include all solutions $u^{\mathcal{N}}(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \Xi_{\text{train}}$) produces the best reduced solution space and thus the most accurate, albeit costly, surrogate solution. We note that this version of POD only serves as reference and is in general not feasible as the full solution ensemble must be generated. On the other end, a random selection of N parameters as our RB snapshots is a fast but crude method. Comparison results of three steady-state test problems are shown in Figure 3 with FDM points per dimension $\sqrt{\mathcal{N}}$ set to be 400 for first two cases (results with different $\sqrt{\mathcal{N}}$ are similar) and $\sqrt{\mathcal{N}} = 100$ for the third case. Not surprisingly, the exhaustive POD is the most accurate. Our L1-ROC is one order of magnitude worse than POD, but in fact slightly better or comparable to the the best possible random generation. It is roughly one order of magnitude better than the median performance of random generations. [The cost comparison between L1-ROC and the POD approach is given in Figure 2 top right which shows that the POD offline cost is more than 5 times more costly than that of L1-ROC for the cubic reaction diffusion problem.](#)

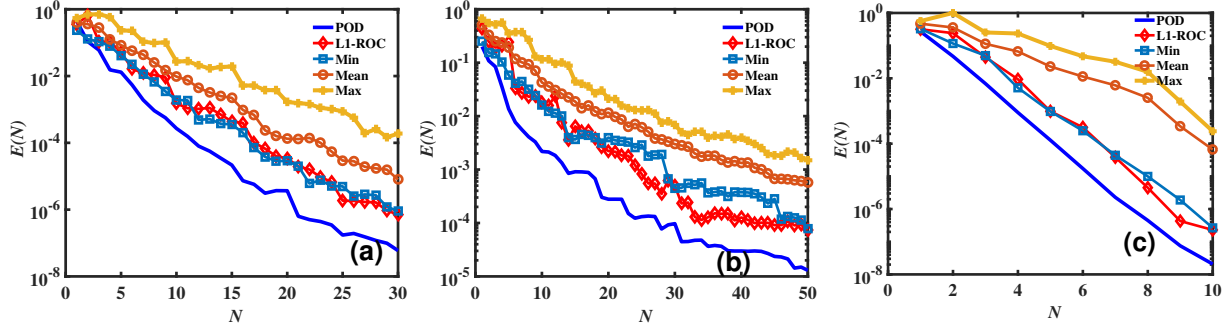


Figure 3: Convergence comparison for the L1-ROC, exhaustive POD and (best, median, and worst cases of) random generation approaches. (a) Poisson-Boltzmann equation (17) with $\sqrt{N} = 400$, (b) cubic reaction diffusion (15) with $\sqrt{N} = 400$, (c) steady viscous Burgers' equation (12) with $\sqrt{N} = 100$.

3.2 Time dependent nonlinear problems

In this section, we test the time-dependent equations corresponding to stationary problems in the last section, namely viscous Burgers' and cubic reaction diffusion equations.

3.2.1 Viscous Burgers' equation

We test the viscous Burgers' equation adopting settings similar to [41, 39]

$$\begin{aligned} u_t + uu_x &= \mu u_{xx} + f(x), \quad (x, t, \mu) \in (0, 1) \times (0, 1] \times \mathcal{D}, \\ u(x, t = 0; \mu) &= 0, \\ u(0, t; \mu) &= \alpha, \quad u(1, t; \mu) = \beta. \end{aligned} \tag{19}$$

The authors of [41] takes $\mathcal{D} = [0.1, 1]$, $f = 0$, $T = 1$, $\Delta t = 10^{-4}$, $(\alpha, \beta) = (-1, 1)$ and monitor the average error in a Frobenius norm-based metric,

$$\text{Error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} \frac{\|u(\cdot, \cdot; \boldsymbol{\mu}) - \hat{u}(\cdot, \cdot; \boldsymbol{\mu})\|_F}{\|u(\cdot, \cdot; \boldsymbol{\mu})\|_F}, \quad \|v(\cdot, \cdot)\|_F^2 := \sum_{\mathbf{x} \in X^N, t_i \in \mathcal{T}_f} v(\mathbf{x}, t_i)^2$$

while the authors in [39] set $\mathcal{D} = [0.005, 1]$, $f = 1$, $T = 2$, $\Delta t = 2 \cdot 10^{-6}$, $(\alpha, \beta) = (0, 0)$ and observe the error in L^2 . We investigate L1-ROC results from both of these setups. The results are showed in Figure 4. These results are similar to those of [41, 39]. However, we note that they come at a much smaller computational expense.

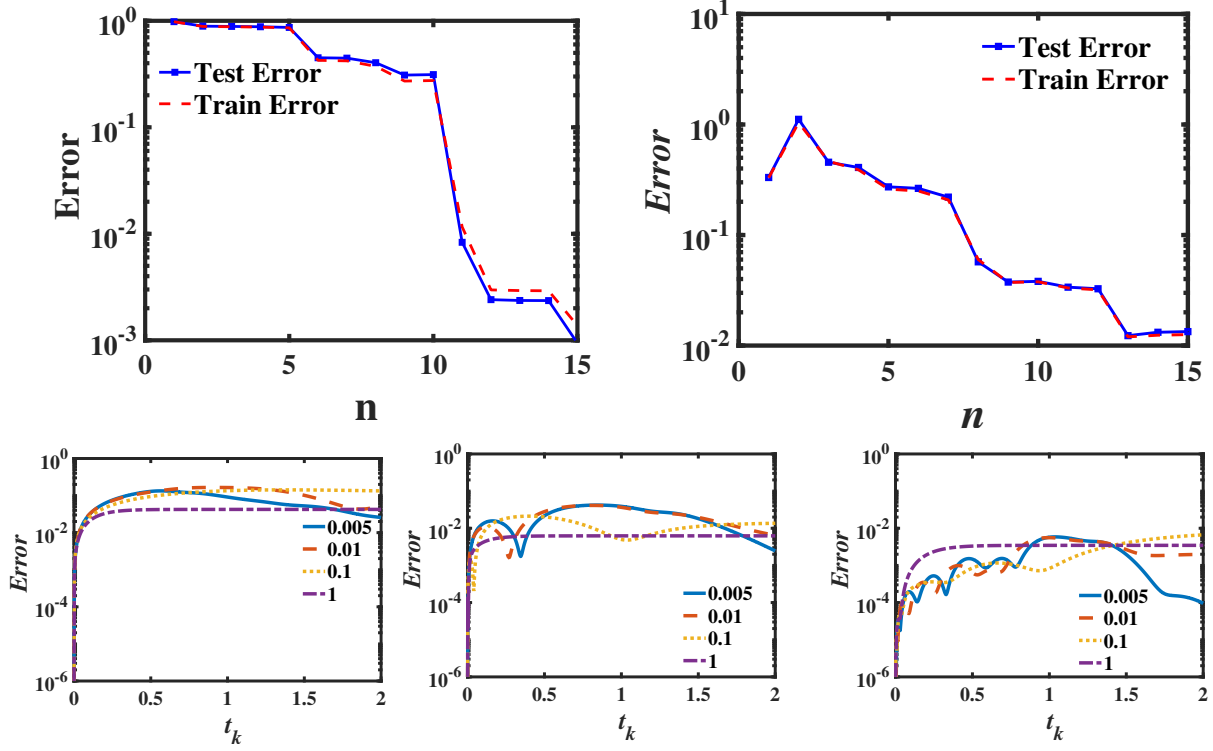


Figure 4: Transient viscous Burgers' result. On the top row are the error curves of L1-ROC with $N = 15$ basis elements for the setup in [41] (left) and [39] (right). Plotted at the bottom are the actual L^2 error, $\|u^N(:, t_k; \boldsymbol{\mu}) - u_N(:, t_k; \boldsymbol{\mu})\|$ as a function of discrete time t_k . The left, center and right plots show $N = 5, 10, 15$, respectively, each for parameter values $\boldsymbol{\mu} = 0.005, 0.01, 0.1, 1$ with the setup as in [39].

3.2.2 Nonlinear reaction diffusion problems

Next, we consider accordingly the following time dependent nonlinear reaction diffusion equation,

$$\begin{aligned} u_t - \mu_2 \Delta u + u(u - \mu_1)^2 &= f(\mathbf{x}), \text{ in } \Omega = [-1, 1] \times [-1, 1], \\ u &= 0 \text{ on } \partial\Omega, \\ u(\mathbf{x}, t = 0) &= u_0(\mathbf{x}). \end{aligned} \tag{20}$$

Here $f(\mathbf{x}) = 100 \sin(2\pi x_1) \cos(2\pi x_2)$, and $[\mu_1, \mu_2] \in \mathcal{D} := [1, 5] \times [0.2, 1]$. The parameter space $\mathcal{D}$ is discretized by a 128×32 uniform tensorial grid. Denoting the step size along the μ_1 direction by h_1 , and the other by h_2 , we specify the training and test sets as follows,

$$\begin{aligned} \Xi_{\text{train}} &= (1 : 8h_1 : 5) \times (0.2 : 2h_2 : 1), \\ \Xi_{\text{test}} &= ((1 + 2h_1) : 4h_1 : (5 - 2h_1)) \times ((0.2 + h_2) : 4h_2 : (1 - h_2)), \end{aligned}$$

For the truth approximation, we use backward Euler for time marching and the same non-linear spatial solver as the steady-state case (16).

We report the $\boldsymbol{\mu}$ -component of the parameter values selected by L1-ROC in Figure 5 (top). Note that the RB space is built from the snapshots

$$\left\{ u(t_{\boldsymbol{\mu}^n}^1, \cdot; \boldsymbol{\mu}^n), \dots, u(t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}}, \cdot; \boldsymbol{\mu}^n) \right\}_{n=1}^N.$$

That is, for each distinct parameter value $\boldsymbol{\mu}^n$ chosen by L1-ROC, there are $k_{\boldsymbol{\mu}^n} \geq 1$ time level snapshots $\{t_{\boldsymbol{\mu}^n}^1, \dots, t_{\boldsymbol{\mu}^n}^{k_{\boldsymbol{\mu}^n}}\} \subset \{t_0, t_1, \dots, t_{N_t}\}$. The red number label plotted next to each $\boldsymbol{\mu}^n$ value in the left pane denotes the corresponding value of $k_{\boldsymbol{\mu}^n}$. It is interesting to note that, consistent with the tendency of RBM selecting parameter values on the boundary of the domain, our L1-ROC tends to select multiple snapshots along time for the selected parameters along the boundary of the parameter domain. The right pane is the corresponding 3D-image of the left. The bottom row of Figure 5 shows the L1-ROC error curve, which shows clear exponential convergence, and collocation points in the physical domain.

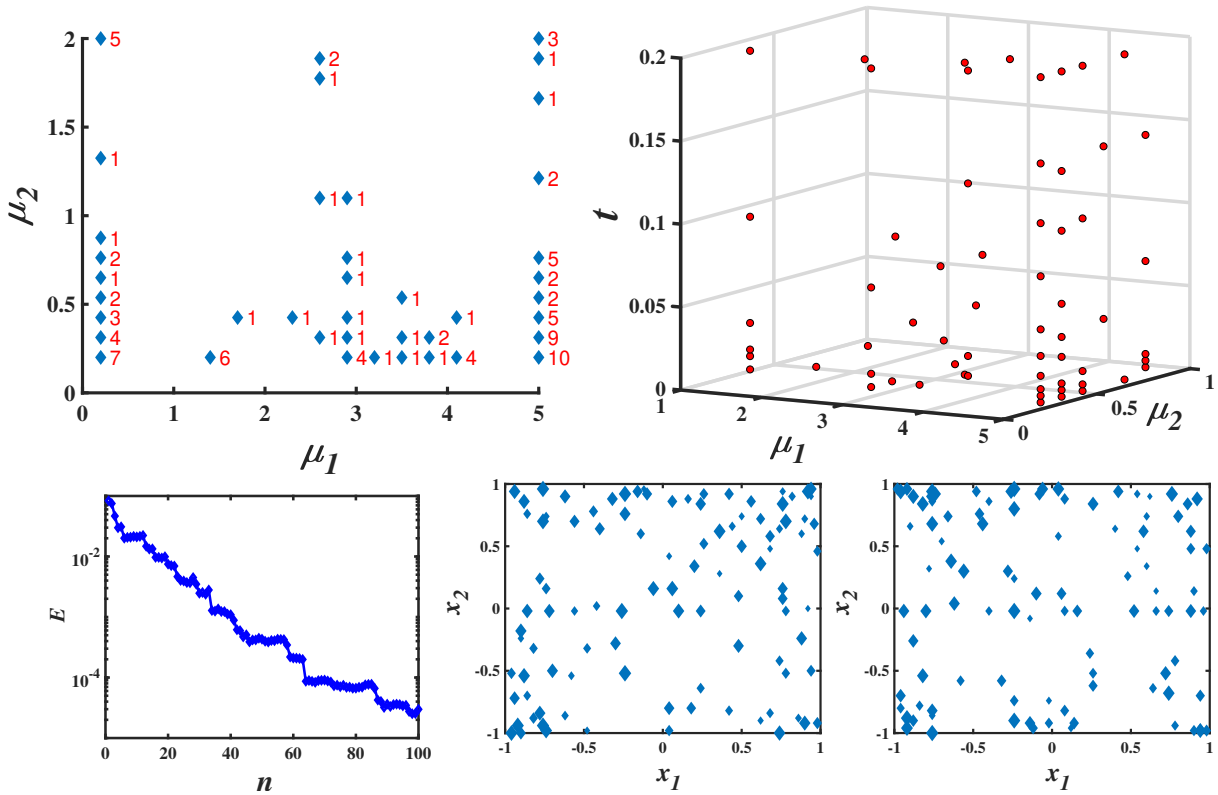


Figure 5: Transient cubic reaction diffusion result. Top Left: Selected parameters when $N_{max} = 100$. The number means corresponding parameter is selected at many different time nodes. Top Right: a three-dimensional view of the selected parameters. Error curves of L1-ROC algorithm, and collocation points from solutions and residuals are shown at the bottom row from left to right respectively.

4 Conclusion

This paper proposes a novel reduced over-collocation method, dubbed L1-ROC, for efficiently solving parameterized, nonlinear, nonaffine, and stationary or transient PDEs. By integrating EIM technique on the solution snapshots and well-chosen residuals, the collocation philosophy, and the simplicity of the L1-based importance indicator that is extended to time-dependent problems, L1-ROC has online computational complexity independent of the degrees of freedom of the underlying FDM, and is furthermore immune from efficiency degradation due to a large number of EIM expansion terms. This expansion typically significantly degrades the efficiency of a traditional RBM algorithm when applied to the nonaffine and nonlinear terms in the equation. The lack of such precomputations of nonlinear and nonaffine terms makes the method dramatically faster offline and online, and significantly simpler to implement than any existing RBM method. For future directions, we plan to apply L1-ROC to systems of equations resulting from CFD systems with more complicated nonlinear and nonaffine terms. A deeper understanding of the theory of this L1-ROC algorithm, [through exploring its connection with GEIM](#), is also under consideration. [The difficulties are two-fold. First, the much more challenging one, is to show the L1 indicator leads to subspaces with distance converging similarly to that of the Kolmogorov N-width of the solution manifold. The second, more amenable via the GEIM connection, is the stability and error analysis of the ROC solver. Another interesting direction to explore is adaptivity of the ROC points as time varies for the transient case, resulting in the ability to choose different points for regimes with different solution behavior.](#)

References

- [1] B. O. Almroth, P. Stern, and F. A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA J.*, 16(5):525–528, 1978.
- [2] P. Astrid. Fast reduced order modeling technique for large scale ltv systems. In *Proceedings of the 2004 American control conference*, volume 1, pages 762–767. IEEE, 2004.
- [3] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, 2008.
- [4] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Math.*, 339(9):667–672, 2004.
- [5] A. Benaceur, V. Ehrlacher, A. Ern, and S. Meunier. A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems. *SIAM J. Sci. Comput.*, 40(5):A2930–A2955, 2018.

- [6] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.
- [7] R. Bos, X. Bombois, and P. Van den Hof. Accelerating large-scale non-linear models for monitoring and control using spatial and temporal correlations. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3705–3710. IEEE, 2004.
- [8] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction. *J. Comput. Phys.*, 330:693–734, 2017.
- [9] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *Int J Numer Methods Eng.*, 86(2):155–181, 2011.
- [10] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.*, 242:623 – 647, 2013.
- [11] F. Casenave, A. Ern, and T. Lelièvre. Accurate and online-efficient evaluation of the *a posteriori* error bound in the reduced basis method. *ESAIM Math. Model. Numer. Anal.*, 48(1):207–229, 2014.
- [12] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.
- [13] Y. Chen and S. Gottlieb. Reduced collocation methods: Reduced basis methods in the collocation framework. *J. Sci. Comput.*, 55(3):718–737, 2013.
- [14] Y. Chen, S. Gottlieb, and Y. Maday. Parametric analytical preconditioning and its applications to the reduced collocation methods. *C. R. Acad. Sci. Paris, Ser. I*, 352:661–666, 2014.
- [15] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodriguez. Certified reduced basis methods and output bounds for the harmonic Maxwell’s equations. *SIAM J. Sci. Comput.*, 32(2):970–996, 2010.
- [16] Y. Chen, J. S. Hesthaven, Y. Maday, J. Rodriguez, and X. Zhu. Certified reduced basis method for electromagnetic scattering and radar cross section estimation. *Comput. Methods Appl. Mech. Engrg.*, 233-236:92–108, 2012.
- [17] Y. Chen, J. Jiang, and A. Narayan. A robust error estimator and a residual-free error indicator for reduced basis methods. *Computers & Mathematics with Applications*, 77:1963–1979, 2019.
- [18] S. Deparis and G. Rozza. Reduced basis method for multi-parameter-dependent steady Navier–Stokes equations: applications to natural convection in a cavity. *J. Comput. Phys.*, 228(12):4359–4378, 2009.

- [19] G. Dimitriu, R. Ștefănescu, and I. M. Navon. Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems. *Journal of Computational and Applied Mathematics*, 310:32–43, 2017.
- [20] R. Everson and L. Sirovich. Karhunen–loève procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.
- [21] F. Fritzen, B. Haasdonk, D. Ryckelynck, and S. Schöps. An algorithmic comparison of the hyper-reduction and the discrete empirical interpolation method for a nonlinear thermal problem. *Mathematical and Computational Applications*, 23(1):8, 2018.
- [22] P. H. G. Berkooz and J. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25(1):539–575, 1993.
- [23] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *Int J Numer Methods Eng.*, 81(12):1581–1608, 2010.
- [24] M. A. Grepl. Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations. *Mathematical Models and Methods in Applied Sciences*, 22(03):1150015, 2012.
- [25] M. A. Grepl, Y. Maday, N. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM-Math. Model. Numer. Anal.*, 41(3):575–605, 2007.
- [26] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.
- [27] B. Haasdonk. *Chapter 2: Reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems*, volume 15, pages 65–136. SIAM Philadelphia, 2017.
- [28] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *M2AN Math. Model. Numer. Anal.*, 42(2):277–302, 2008.
- [29] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*. SpringerBriefs in Mathematics. Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016. BCAM Springer-Briefs.
- [30] D. B. P. Huynh, D. J. Knezevic, Y. Chen, J. S. Hesthaven, and A. T. Patera. A natural-norm Successive Constraint Method for inf-sup lower bounds. *CMAME*, 199:1963–1975, 2010.

- [31] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Acad. Sci. Paris, Série I.*, 345:473–478, 2007.
- [32] L. Ji, Y. Chen, and Z. Xu. A reduced basis method for the nonlinear Poisson-Boltzmann equation. *Adv. Appl. Math. Mech.*, 11(5):1200–1218, 2019.
- [33] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2002.
- [34] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. Proper orthogonal decomposition and its applications-Part I: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.
- [35] Y. Maday, O. Mula, and G. Turinici. Convergence analysis of the generalized empirical interpolation method. *SIAM Journal on Numerical Analysis*, 54(3):1713–1731, 2016.
- [36] Y. Maday, N. C. Nguyen, A. T. Patera, and S. H. Pau. A general multipurpose interpolation procedure: the magic points”, 2009.
- [37] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comput. Phys.*, 303:431–454, 2015.
- [38] N. C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for nonlinear parametrized partial differential equations. *Int J Numer Methods Eng.*, 76(1):27–55, 2008.
- [39] N.-C. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers’ equation. *Calcolo*, 46(3):157–185, 2009.
- [40] A. K. Noor and J. M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA J.*, 18(4):455–462, 1980.
- [41] B. Peherstorfer. Sampling low-dimensional markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference. *arXiv preprint arXiv:1908.11233*, 2019.
- [42] B. Peherstorfer, D. Butnaru, K. Willcox, and H. Bungartz. Localized discrete empirical interpolation method. *SIAM J. Sci. Comput.*, 36(1):A168–A192, 2014.
- [43] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: An introduction*, volume 92. Springer, 2015.
- [44] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008.

- [45] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *J. Comput. Phys.*, 202(1):346–366, 2005.
- [46] D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *Int J Numer Methods Eng.*, 77(1):75–89, 2009.
- [47] K. Veroy, C. Prud’Homme, and A. T. Patera. Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds. *C. R. Math.*, 337(9):619–624, 2003.
- [48] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA J.*, 40(11):2323–2330, 2002.