Nonlinear Model Reduction Based on the Finite Element Method with Interpolated Coefficients: Semilinear Parabolic Equations

Zhu Wang
Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208

Received 30 August 2013; accepted 17 December 2014
Published online 17 January 2015 in Wiley Online Library (wileyonlinelibrary.com).
DOI 10.1002/num.21961

For nonlinear reduced-order models (ROMs), especially for those with high-order polynomial nonlinearities or nonpolynomial nonlinearities, the computational complexity still depends on the dimension of the original dynamical system. To overcome this issue, we develop an efficient finite element (FE) discretization algorithm for nonlinear ROMs. The proposed approach approximates the nonlinear function by its FE interpolant, which makes the inner product evaluations in generating the nonlinear terms computationally cheaper than that in the standard FE discretization. Due to the separation of spatial and temporal variables in the FE interpolation, the discrete empirical interpolation method (DEIM) can be directly applied on the nonlinear functions in the same manner as that in the finite difference setting. Therefore, the main computational hurdles for applying the DEIM in the FE context are conquered. We also establish a rigorous asymptotic error estimation, which shows that the proposed approach achieves the same accuracy as that of the standard FE method under certain smoothness assumptions of the nonlinear functions. Several numerical tests are presented to validate the proposed method and verify the theoretical results.
Numer Methods Partial Differential Eq 31: 1713–1741, 2015

Keywords: discrete empirical interpolation method; finite element method with interpolated coefficients; nonlinear model reduction; proper orthogonal decomposition

I. INTRODUCTION

Control and optimization problems in realistic engineering applications often require repeated numerical simulations of large-scale dynamical systems. If a fast or real-time control strategy is desired, a brute force direct numerical simulation is impractical. Therefore, the proper orthogonal decomposition (POD) method has been frequently used to generate a reduced-order model (ROM) that can be utilized as an alternative of the original dynamical system (hereinafter referred to as the full-order model) [1]. Such a ROM only contains a handful of degrees of freedom (DOF), yet is computationally feasible and free of storage issues. The POD method has been successfully

Correspondence to: Zhu Wang, Department of Mathematics, University of South Carolina, Columbia, SC 29208 (e-mail: wangzhu@math.sc.edu)

applied in many scientific and engineering problems (see, e.g., a brief summary in [2]). However, for complex dynamical systems, the original promise of the POD as an efficient yet accurate approximation remains to be fulfilled. On the one hand, it may achieve erroneous results even when POD basis functions capture most of the system energy [3]. On the other hand, the efficiency is severely limited by the nonlinearity of the system [4]. For treating the first issue, research has been done in two main directions: (i) to improve the POD basis functions [5–15]; (ii) to improve the ROM by modeling the effect of discarded POD basis [2, 16–40]. To our knowledge, the research for seeking a reliable reduced-order modeling approach for general complex systems is still active, however, beyond the scopes of this article. Instead, we will focus on the second issue and develop an efficient algorithm for the nonlinear ROM.

Indeed, the computational efficiency of POD-ROMs relies on two key components: (i) the dimension of the ROM is extremely small; (ii) the vectors and matrices in the reduced system have small sizes and can be precomputed. For linear systems or nonlinear systems with low order polynomial nonlinearities, both ingredients are satisfied. However, for highly nonlinear dynamical systems, the second one does not hold any more because matrices associated with the nonlinear terms have to be evaluated and assembled at each time step or iteration. As POD basis functions are global, the evaluation would depend on the dimension of the full-order model, and, therefore, greatly increase the computational cost. Several methods have been proposed to resolve this issue.

The two-level algorithms proposed in [2] are motivated by the observation that only a small number of leading POD basis functions are kept in the ROM, and they have larger length scales than the discarded ones. If a computation on a fine mesh is used to obtain all the POD basis, one should be able to use a much coarser mesh to represent the leading POD basis. Therefore, in two-level algorithms, nonlinear closure terms are computed on the coarse mesh. This way can decrease the computational cost by an order of magnitude, while achieving the same level of accuracy as the simulation on the fine mesh. However, the optimal choice of the coarse mesh still needs to be investigated.

Much work devotes to approximate nonlinear terms at a few selected spatial points or within the neighborhoods around the points. The trajectory piecewise-linear method, presented in [41, 42], reduces a nonlinear model to a weighted sum of linearized models at selected points along a state trajectory. The missing points estimation method was developed in [43–45]. In that approach, the full-order system was first reduced by choosing equations only associated with selected spatial points, restricting the POD basis onto these points, and then projecting the extracted system onto the space spanned by the POD basis. The empirical interpolation method (EIM) was first proposed in [46] for approximating nonaffine parameter dependence functions to enable an efficient offline-online computational strategy, and then was further applied to approximate nonlinear functions in [47]. The method selects interpolation points by greedy algorithms that are guided by a posteriori error estimates. However, in certain problems, even the most optimal basis set is of large size [48], which reduces the efficiency of the algorithm. Thus, several improved greedy algorithms have been proposed in [49]. The best-points interpolation method was introduced in [50], which determines interpolation points from a least-square minimization problem. The discrete empirical interpolation method (DEIM) was introduced in [4] and analyzed in [51]. The method combines projection with interpolation and chooses interpolation points from the POD basis of the nonlinear functions. As certain coefficient matrix can be precomputed when approximating nonlinear functions, the complexity of the POD-ROM reduces to be proportional to the number of selected spatial indices.

Another method is the group POD method (GPOD) proposed in [52], which extends the group finite element (FE) method to the POD setting. In that paper, the authors considered dynamical systems with polynomial nonlinearities, such as the Burgers equations. The POD approximation

of the quadratic nonlinear term was rewritten in a group format. If r POD basis functions are used in the ROM, the GPOD requires $r^3 - \frac{1}{2}r^2 - \frac{1}{2}r$ flops less than the standard POD implementation in the computation of the nonlinear term. However, the approach is limited to polynomial nonlinearities.

In this report, we focus on developing an efficient FE discretization algorithm for nonlinear ROMs. In particular, we are interested in applying the DEIM to reduce the intensive computational efforts for evaluating the nonlinear terms. However, in the FE setting, there are two major issues that degrade the effectiveness of the DEIM: (i) generating nonlinear snapshots, which are to be used for seeking the nonlinear POD basis, requires calculations of the inner product in the nonlinear terms, which costs lots of offline computation time; (ii) the online simulation needs evaluations of the inner product over the elements sharing selected DEIM points. Repeated numerical integrations will increase the online simulation time, especially, in cases such as complex flow simulations when many DEIM points are required to achieve a good approximation.

To address the second issue, a modified DEIM was proposed in [53] and further investigated for nonlinear and parametric systems in [54]. It uses an unassembled FE mesh for the nonlinear term discretization so that each selected DEIM point only relates to one element. Thus, the number of element function calls is reduced during the simulation. However, it enlarges the size of nonlinear snapshots, which increases the offline computational cost. In this article, to overcome both hurdles, we develop the finite element method with interpolated coefficients (FEIC) for nonlinear POD-ROMs. This method is also known as the group FE method or the product approximation, which has been successfully applied to find numerical solutions to nonlinear partial differential equations (PDEs) [55–64]. Indeed, we replace the nonlinear function in the POD-ROM with its FE interpolant directly. This simple change leads to great savings in the computation: first, the calculation of the inner product in nonlinear terms becomes a product of a precomputable coefficient matrix and a vector of the nonlinear function values at the FE nodes, which is computationally cheaper than that in the standard FE discretization; second, for the ROMs discretized by the new approach, the DEIM can be directly applied in the same manner as that in the finite difference (FD) method setting. In fact, the nonlinear snapshots become a collection of vectors of nonlinear function values, therefore, the nonlinear data generation does not require any numerical integrations. The accuracy of this approach is, of course, restricted to the smoothness of the nonlinear functions. However, when the nonlinear functions possess certain smoothness ((H2) and (H3) in Section IV), the FEIC can achieve the same accuracy as the standard FE discretization of the ROM. Therefore, the advantages of the new approach over the standard FE discretization of nonlinear POD-ROMs cannot be over-emphasized: (i) the FEIC is easier to implement and computationally more efficient; (ii) the FEIC achieves the same accuracy when nonlinear functions satisfy smoothness assumptions; and (iii) the FEIC is more suitable for combining with the DEIM to further reduce the computational complexity.

Note that the GPOD method is based on a similar idea. Distinguishing from it, the proposed method in this article does not group any variables in terms of the POD basis. Therefore, it fits well for ROMs with polynomial or nonpolynomial nonlinearities. We present the method in the context of time-dependent semilinear parabolic equations in this article and derive a priori error estimates for the approximate solutions. However, the proposed method also works for semilinear parametric equations. Extensions of the method to quasilinear and fully nonlinear systems are under investigation.

The rest of this article is organized as follows: a brief introduction to the POD method is presented in Section II; the FEIC of POD-ROMs for semilinear parabolic equations is proposed in Section III; a rigorous asymptotic error estimate is developed in Section IV; several examples are tested in Section V to numerically demonstrate the accuracy and efficiency of the new approach.

Wherein, the first two examples are used for validation and the third example is for verification. The combination of the proposed approach with the DEIM is discussed in Section VI, whose effectiveness is then illustrated by revisiting the first two examples used in Section V. A few conclusions are drawn in the last section with several ongoing research directions we are pursuing listed.

II. THE POD METHOD

In this section, we briefly introduce the (time-continuous) POD method. For detailed discussions, the reader is referred to [65–70].

Let $\mathcal{H}$ be a real Hilbert space endowed with inner product $(\cdot,\cdot)_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$. Assume the data $\mathcal{V}$ (so-called snapshots), which is a collection of time-varying functions $y(x,t) \in L^2(0,T;\mathcal{H})$, the POD method seeks a low-dimensional basis, $\varphi_1(x),\ldots,\varphi_r(x) \in \mathcal{H}$, that optimally approximates the data. Mathematically speaking, for any positive r, the POD basis is determined by minimizing the error between the data and its projection onto the basis, that is,

$$\min_{\{\varphi_j\}_{j=1}^r} \int_0^T \left\| y(\cdot,t) - \sum_{j=1}^r \big(y(\cdot,t),\varphi_j(\cdot)\big)_{\mathcal{H}}\varphi_j(\cdot)\right\|_{\mathcal{H}}^2 dt, \tag{1}$$

subject to the conditions that $(\varphi_i,\varphi_j)_{\mathcal{H}} = \delta_{ij}, 1 \le i,j \le r$, where δ_{ij} is the Kronecker delta. To solve (1), one can consider the eigenvalue problem

$$Kv_j = \lambda_j v_j, \tag{2}$$

where K is a compact linear operator that satisfies $Kv_j(s) = \int_0^T (y(\cdot,t),y(\cdot,s))_{\mathcal{H}} v_j(t)dt$, v_j is the jth eigenvector, and $\lambda_j \le \cdots \le \lambda_2 \le \lambda_1$ are positive eigenvalues.

It can be shown that the solution of (1) is given by

$$\varphi_j(\cdot) = \frac{1}{\sqrt{\lambda_j}} \int_0^T v_j(t) y(\cdot,t)dt, \quad 1 \le j \le r. \tag{3}$$

Proposition 2.1 ([67]). *Let the POD basis φ_j be given by (3), the POD projection error satisfies*

$$\int_0^T \left\| y(\cdot,t) - \sum_{j=1}^r \big(y(\cdot,t),\varphi_j(\cdot)\big)_{\mathcal{H}}\varphi_j(\cdot)\right\|_{\mathcal{H}}^2 dt = \sum_{j>r} \lambda_j \tag{4}$$

Remark 2.1. In practice, discrete data has to be considered. The POD basis for an ensemble of snapshots, $\mathcal{V} = \text{span}\{y(x,t_1),\ldots,y(x,t_M)\}$, is to minimize the projection error

$$\min_{\{\varphi_j\}_{j=1}^r} \sum_{\ell=1}^M \left\| y(\cdot,t_\ell) - \sum_{j=1}^r \big(y(\cdot,t_\ell),\varphi_j(\cdot)\big)_{\mathcal{H}}\varphi_j(\cdot)\right\|_{\mathcal{H}}^2. \tag{5}$$

The solution can be obtained by solving the eigenvalue problem $Kv_j = \lambda_j v_j$ first, where $K \in \mathbb{R}^{M\times M}$ is the snapshot correlation matrix with $K_{\ell k} = (y(\cdot,t_\ell),y(\cdot,t_k))_{\mathcal{H}}$ and $1 \le \ell,k \le M$, then the POD basis is given by $\varphi_j(\cdot) = \frac{1}{\sqrt{\lambda_j}} \sum_{\ell=1}^M (v_j)_\ell y(\cdot,t_\ell)$, $1 \le j \le r$.

Remark 2.2. FE solutions are commonly utilized as the snapshots, that is, the data $y(x,t) = \sum_{\iota=1}^{n_{\text{dof}}} \mathbf{Y}_\iota(t) h_\iota(x)$, where $\mathbf{Y}(t)$ is the approximation solution vector at time t, $h_\iota(x)$ is the ιth FE nodal basis, and n_{dof} is the number of DOF in the spatial discretization. Then, the POD basis can be written as

$$\varphi_j(x) = \sum_{\iota=1}^{n_{\text{dof}}} \mathbf{Q}_{\iota j} h_\iota(x), \tag{6}$$

where the coefficient matrix $\mathbf{Q}$ is with the entry $\mathbf{Q}_{\iota j} = \frac{1}{\sqrt{\lambda_j}} \int_0^T v_j(t) Y_\iota(t) dt$ for time-continuous data or $\mathbf{Q}_{\iota j} = \frac{1}{\sqrt{\lambda_j}} \sum_{\ell=1}^{M} (v_j)_\ell Y_\iota(t_\ell)$ for time-discrete data. For a detailed discussion on implementing the POD method in the FE setting, readers are referred to [71].

III. THE POD-ROM AND FE DISCRETIZATIONS

Let Ω be a convex domain in $\mathbb{R}^d$ with the smooth boundary $\partial\Omega$, $d = 1, 2, 3$. Also let $\mathcal{T}_h$ be a collection of quasiuniform elements that partitions the domain. The elements are line segments if $d = 1$, triangles if $d = 2$, and polyhedra if $d = 3$. The parameter h is the maximal diameter of the elements. Denoted by X the space $L^2(\Omega)$ equipped with the inner product $(\cdot,\cdot)$ and norm $\|\cdot\|$; by V the Sobolev space $H_0^1(\Omega) = \{v | v \in H^1(\Omega), v|_{\partial\Omega} = 0\}$ with H^1 seminorm $|\cdot|_1$ and H^1 norm $\|\cdot\|_1$; and by V^h the space of piecewise continuous functions on Ω that reduce to polynomials of degree $\leq m$ on each element of $\mathcal{T}_h$, which satisfies $V^h \subset V$. We assume the semilinear problems that we will consider in this section admit a unique solution $u = u(x,t)$ on the time interval $[0,T]$, which ranges in V.

We consider the equivalent variational problems of semilinear parabolic equations with homogeneous boundary conditions: To find $u(x,t) \in V$, such that, either

$$\left(\frac{\partial u}{\partial t}, v\right) + a(u,v) + (N(u), v) = (f, v), \quad \forall v \in V, \tag{7}$$

or

$$\left(\frac{\partial u}{\partial t}, v\right) + a(u,v) + (N(u), \nabla v) = (f, v), \quad \forall v \in V, \tag{8}$$

with the initial condition

$$u(x,0) = u_0(x), \quad \forall x \in \Omega.$$

Where $f = f(x,t)$ is the source term independent with u, the bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ is continuous and coercive, that is, there exist constants α and β such that

$$|a(u,v)| \leq \alpha \|u\|_1 \|v\|_1, \quad \forall u, v \in V, \tag{9}$$

$$|a(u,u)| \geq \beta \|u\|_1^2, \quad \forall u \in V. \tag{10}$$

$N(u)$ is a nonlinear function of u. In particular, we are interested in cases in which $N(u)$ possesses either a nonpolynomial nonlinearity or a high order polynomial nonlinearity. Due to the similarity between (7) and (8), to shorten the presentation, we only discuss (7) in the sequel, but will comment a theoretical result of (8) in Remark 4.2 and test a problem governed by (8) in Section V.

Many well-established methods can be used for seeking a numerical solution to Eq. (7). However, when repeated numerical simulations are required, direct simulations result in a huge computational cost and become infeasible. Therefore, the POD method has been widely used for generating a ROM.

A. The POD-ROM

Given snapshots consisting of either numerical solutions or experimental data, the POD basis functions $\{\varphi_1(x), \ldots, \varphi_r(x)\}$ are determined by (1)–(3), where $\mathcal{H} = L^2$ in our considerations. The associated POD approximation of $u(x, t)$ is given by

$$u_r(x,t) \equiv \sum_{j=1}^{r} \varphi_j(x) a_j(t), \tag{11}$$

where $\{a_j(t)\}_{j=1}^{r}$ are the sought time-varying POD basis coefficient functions. Substituting the POD approximation (11) into (7), applying the Galerkin method, and considering the POD basis functions are orthonormal, we obtain the Galerkin projection-based POD-ROM (POD-G-ROM): to find $u_r(x,t) \in V_r = \text{span}\{\varphi_1, \ldots, \varphi_r\}$, such that,

$$\left(\frac{\partial u_r}{\partial t}, v_r\right) + a(u_r, v_r) + (N(u_r), v_r) = (f, v_r), \quad \forall v_r \in V_r, \tag{12}$$

and

$$u_r(x,0) = u_{0,r}(x) \in V_r \quad \forall x \in \Omega.$$

Let $\mathbf{a}(t) = [a_1(t), \ldots, a_r(t)]^\intercal$, the POD-G-ROM can be rewritten in terms of POD basis functions as:

$$\dot{\mathbf{a}} = \mathbf{A} + \mathbf{B}\mathbf{a} + \mathbf{C}(\mathbf{a}), \tag{13}$$

with, for example,

$$\mathbf{a}(0) = (u_0(x), \boldsymbol{\varphi}(x)).$$

Where $\mathbf{A}_k = (f, \varphi_k)$, $\mathbf{B}_{jk} = -a(\varphi_j, \varphi_k)$, $(\mathbf{C}(\mathbf{a}))_k = -\left(N(\sum_{j=1}^{r} \varphi_j a_j(t)), \varphi_k\right)$, for $k = 1, \ldots, r$, and $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_r]^\intercal$. The resulted dynamic system (13) is of dimension r, which is much smaller than the number of DOF in the full-order model. Once $\mathbf{a}$ is obtained, the POD approximation solution u_r can be recovered by (11).

The most significant advantage of the ROM is its computational efficiency. Indeed, the matrix $\mathbf{B}$ can be precomputed. In certain cases, we can also compute matrices in $\mathbf{C}$ beforehand. For example, in Navier–Stokes equations, one can write $(\mathbf{C}(\mathbf{a}))_k = \mathbf{a}^\intercal \mathbf{C}^k \mathbf{a}$, where $[\mathbf{C}^k]_{ij} = -(\varphi_i \cdot \nabla \varphi_j, \varphi_k)$, and $i, j, k = 1, \ldots, r$. The matrices only need to be computed once and can be used repeatedly in online simulations. However, this attractive property does not hold when the nonlinear function $N(u)$ is with a higher order polynomial nonlinearity or a nonpolynomial one. As a result, the computational efficiency of the ROM decays. Especially, when the FE is used for a spatial discretization. Therefore, in the rest of this article, we restrict ourselves to the FE methods of the nonlinear ROMs (12) and develop a new efficient FE discretization algorithm.

B. FE Discretizations

Let $V_r^h = \text{span}\left\{\varphi_1^h, \ldots, \varphi_r^h\right\}$, where φ_j^h is the FE discretization of φ_j, $j = 1, \ldots, r$. The standard FE discretization of the POD-G-ROM (12) (POD-FEM) is to find $\tilde{u}_r^h(x,t) \in V_r^h$, such that,

$$\left(\frac{\partial \tilde{u}_r^h}{\partial t}, v_r^h\right) + a\left(\tilde{u}_r^h, v_r^h\right) + \left(N(\tilde{u}_r^h), v_r^h\right) = (f, v_r^h), \quad \forall v_r^h \in V_r^h, \tag{14}$$

and

$$\tilde{u}_r^h(x,0) = u_{0,r}^h(x) \in V_r^h, \quad \forall x \in \Omega.$$

Represented by $\mathcal{N}_{\text{FE}}$ the nonlinear term in (14), whose kth entry is as follows.

$$\begin{aligned}(\mathcal{N}_{\text{FE}})_k &= \left(N\left(\tilde{u}_r^h(x,t)\right), \varphi_k^h(x)\right) \\ &= \left(N\left(\sum_{j=1}^{r}\sum_{\iota=1}^{n_{\text{dof}}} \mathbf{Q}_{\iota j} h_\iota(x)\tilde{a}_j(t)\right), \sum_{s=1}^{n_{\text{dof}}} \mathbf{Q}_{sk} h_s(x)\right) \\ &= \sum_{s=1}^{n_{\text{dof}}} \mathbf{Q}_{sk}\tilde{\mathbf{z}}_s,\end{aligned} \tag{15}$$

where $\tilde{\mathbf{z}}_s = (\tilde{\mathbf{z}})_s = (N(\sum_{j=1}^{r}\sum_{\iota=1}^{n_{\text{dof}}} \mathbf{Q}_{\iota j} h_\iota(x)\tilde{a}_j(t)), h_s(x)), \tilde{a}_j = (\tilde{\mathbf{a}})_j$ is the jth POD basis coefficient, $\mathbf{Q}$ is as defined in (6), and $k = 1, \ldots, r$.

Obviously, the evaluation of (15) requires computing $\tilde{\mathbf{z}}$ at each time step. In practice, one computes the inner product by numerical quadratures on each element and then assembles $\tilde{\mathbf{z}}$ as a global vector. However, once the quadrature rule is chosen, one can compute the local quadrature information (the product of weights and the values of the FE basis functions at the associated quadrature points) beforehand. To obtain the nonlinear term, it remains to evaluate the nonlinear function at all the quadrature points followed by matrix multiplications. Then, (15) can be written in the following form.

$$\mathcal{N}_{\text{FE}} = \mathbf{Q}^{\intercal}\mathbf{W}\mathbf{y}, \tag{16}$$

where $\mathbf{W}$ is a $n_{\text{dof}} \times n_e n_q$ matrix containing the quadrature information, $\mathbf{y}$ is a $n_e n_q \times 1$ vector containing the values of the nonlinear function at all quadrature points, n_e is the number of elements, and n_q is the number of quadrature points in each element. For example, if the sth FE node is shared by n elements $e_{s_1}, e_{s_2}, \ldots, e_{s_n}$, the sth row of $\mathbf{W}$ is

$$\mathbf{W}_{s,\cdot} = [0, \cdots, 0, \omega_q^{e_{s_1}} h_s(\mathbf{x}_q^{e_{s_1}}), 0, \cdots, 0, \omega_q^{e_{s_2}} h_s(\mathbf{x}_q^{e_{s_2}}), 0, \cdots, 0, \omega_q^{e_{s_n}} h_s(\mathbf{x}_q^{e_{s_n}}), 0, \cdots, 0],$$

where $\omega_q^{e_{s_i}} h_s(\mathbf{x}_q^{e_{s_i}}) = [\omega_1^{e_{s_i}} h_s(x_1^{e_{s_i}}), \ldots, \omega_{n_q}^{e_{s_i}} h_s(x_{n_q}^{e_{s_i}})]$ is the local quadrature information on element e_{s_i} and $\omega_j^{e_{s_i}}$ is the weight of the jth quadrature point $x_j^{e_{s_i}}$, for $j = 1, \ldots, n_q$. And,

$$\mathbf{y} = [N(\tilde{u}_r^h(\mathbf{x}_q^{e_1})), N(\tilde{u}_r^h(\mathbf{x}_q^{e_2})), \ldots, N(\tilde{u}_r^h(\mathbf{x}_q^{e_{ne}}))]^{\intercal},$$

where $N(\tilde{u}_r^h(\mathbf{x}_q^{e_i})) = [N(\tilde{u}_r^h(x_1^{e_i})), \ldots, N(\tilde{u}_r^h(x_{n_q}^{e_i}))]$ is the nonlinear function values at the quadrature points on the element e_i, for $i = 1, \ldots, n_e$. In (16), $\mathbf{Q}^{\intercal}\mathbf{W}$ is precomputable and its size is $r \times n_e n_q$, and $\mathbf{y}$ needs to be calculated repeatedly during the online simulation.

Suppose that the complexity for evaluating the nonlinear function $N(u)$ with θ components is $\mathcal{O}(\varrho(\theta))$, the total complexity in assembling $\mathcal{N}_{\text{FE}}$ at each time step (or iteration) is roughly $\mathcal{O}\left(4rn_en_q + \varrho(n_en_q)\right)$ flops. Here, we count both the multiplications and additions as flops, where $\mathcal{O}\left(\varrho(n_en_q)\right)$ flops are used in computing the nonlinear function values at the quadrature points over all elements, and $\mathcal{O}\left(4rn_en_q\right)$ flops are utilized for the two matrix-vector products: one is to recover the values of $\tilde{u}_r^h$ at all quadrature points from the POD approximation, the other is for computing $(\mathbf{Q}^\intercal\mathbf{W})\mathbf{y}$.

To improve the efficiency of nonlinear ROMs, we propose a method to use the FEIC for a spatial discretization of the POD-G-ROM (12). The FEIC, also known as the product approximation technique [57] or group FE method [58], has been used as an alternative tool of the FE method for solving nonlinear elliptic problems [61, 62], nonlinear parabolic problems [55, 56, 59, 60], and nonlinear hyperbolic problems [63, 64]. This approach replaces the nonlinear function by its interpolant in the finite dimensional space, which leads to great savings in the computational efforts while keeping the accuracy. To our knowledge, this is the first time that the FEIC is applied in the POD setting with a rigorous error estimate provided.

Define the interpolation operator $\mathcal{I}^h : C(\overline{\Omega}) \to S_h$. The interpolant of $N(\cdot)$ in the FE space satisfies

$$(\mathcal{I}^h N)(u(x_i,t)) = N(u(x_i,t)), \tag{17}$$

where x_i is a FE node, $i = 1,\ldots,n_{\text{dof}}$. The FE method with interpolated coefficients of the POD-G-ROM (12) (POD-FEIC) is the following: to find $u_r^h(x,t) \in V_r^h$, such that,

$$\left(\frac{\partial u_r^h}{\partial t}, v_r^h\right) + a\left(u_r^h, v_r^h\right) + \left(\mathcal{I}^h N(u_r^h), v_r^h\right) = \left(f, v_r^h\right), \quad \forall v_r^h \in V_r^h, \tag{18}$$

and

$$u_r^h(\cdot,0) = u_{0,r}^h(x) \in V_r^h, \quad \forall x \in \Omega. \tag{19}$$

Different from the standard FE discretization, the nonlinear function $N(u_r^h)$ in (14) is replaced by the interpolation $\mathcal{I}^h N(u_r^h)$ in (18). The kth row of the nonlinear term in the new numerical discretization, $\mathcal{N}_{\text{FEIC}}$, reads:

$$\begin{aligned}(\mathcal{N}_{\text{FEIC}})_k &= \left((\mathcal{I}^h N)(u_r^h(x,t)), \varphi_k^h(x)\right)\\ &= \left(\sum_{\iota=1}^{n_{\text{dof}}} N\left(\sum_{j=1}^{r}\mathbf{Q}_{\iota j}a_j(t)\right)h_\iota(x), \sum_{s=1}^{n_{\text{dof}}}\mathbf{Q}_{sk}h_s(x)\right)\\ &= \sum_{s=1}^{n_{\text{dof}}}\sum_{\iota=1}^{n_{\text{dof}}}\mathbf{Q}_{sk}\left(h_\iota(x), h_s(x)\right)N\left(\sum_{j=1}^{r}\mathbf{Q}_{\iota j}a_j(t)\right),\end{aligned}$$

where $a_j = (\mathbf{a})_j$ is the jth POD basis coefficient, $\mathbf{Q}$ is as defined in (6), and $k = 1,\ldots,r$. It can be rewritten as follows.

$$\mathcal{N}_{\text{FEIC}} = \mathbf{Q}^\intercal\mathbf{M}^h\mathbf{z}, \tag{20}$$

where $\mathbf{M}^h$ is the FE mass matrix with $\left[\mathbf{M}^h\right]_{\iota s} = (h_\iota, h_s)$ and $\mathbf{z} = N\left(\mathbf{Q}\mathbf{a}(t)\right)$.

The matrix $\mathbf{Q}^\intercal\mathbf{M}^h$ in (20) only needs to be computed once and its size is $r \times n_{\text{dof}}$. At each time step or iteration, only the nonlinear function values at the FE nodes need to be calculated. It represents the major difference between the POD-FEIC model and the POD-FEM model: the former utilizes the nonlinear function values at the FE nodes, while the latter uses the values at all the quadrature points. When the model contains high-order polynomial or nonpolynomial nonlinearities, high-order quadrature rules should be used to achieve an accurate approximation of the nonlinear integrals. As a result, the total number of FE nodes is much less than that of the quadrature points, which makes the POD-FEIC model computationally much cheaper than the POD-FEM model. Indeed, the total computational complexity of (20) at each time step (or iteration) is reduced to $\mathcal{O}(4rn_{\text{dof}} + \varrho(n_{\text{dof}}))$ flops, where $\mathcal{O}(\varrho(n_{\text{dof}}))$ flops are used in computing the nonlinear function values at all the FE nodes and $\mathcal{O}(4rn_{\text{dof}})$ flops are a result of the two matrix-vector products to evaluate the vector of $u_r^h(x,t)$ from $\mathbf{Qa}$, and to calculate $\left(\mathbf{Q}^\intercal\mathbf{M}^h\right)\mathbf{z}$, respectively.

The efficiency can be further improved by combining with the DEIM, which will be discussed in Section VI.

Remark 3.1. When an implicit time integration method is used in numerical simulations, one may solve the nonlinear equations by Newton iteration. If this is the case, one has to compute the Jacobian of the nonlinear term in each iteration besides the original nonlinear term. In the POD-FEM model, the element of the Jacobian matrix at the lth iteration is as follows.

$$\left[\mathcal{J}_{\text{FE}}^l\right]_{kj} = \left(N_u(\tilde{u}_r^{h,l})\varphi_j^h, \varphi_k^h\right), \tag{21}$$

where N_u is the partial derivative of the nonlinear function $N(u)$. Once a quadrature rule is specified, it can be written in the following form:

$$\mathcal{J}_{\text{FE}}^l = \mathbf{Q}^\intercal\mathbf{H}\mathbf{W}\mathbf{D}_q^l\mathbf{H}^\intercal\mathbf{Q}, \tag{22}$$

where $\mathbf{Q}$ is as defined in (6) and $\mathbf{H}$ contains values of the FE basis at the quadrature points. If the sth FE node is shared by n elements $e_{s_1}, e_{s_2}, \ldots, e_{s_n}$, the sth row of $\mathbf{H}$ is

$$\mathbf{H}_{s,\cdot} = \left[0, \cdots, h_s(\mathbf{x}_q^{e_{s_1}}), 0, \cdots, h_s(\mathbf{x}_q^{e_{s_2}}), 0, \cdots, h_s(\mathbf{x}_q^{e_{s_n}}), \cdots, 0\right].$$

And $\mathbf{W} = \text{diag}[\omega_q^{e_1}, \omega_q^{e_2}, \ldots, \omega_q^{e_{n_e}}]$ is a diagonal matrix, which consists of the weight information. $\mathbf{D}_q^l$ is a diagonal matrix with its diagonal entries be values of $N_u(\tilde{u}_r^{h,l})$ at the quadrature points, that is,

$$\mathbf{D}_q^l = \text{diag}[N_u(\tilde{u}_r^{h,l}(\mathbf{x}_q^{e_1})), N_u(\tilde{u}_r^{h,l}(\mathbf{x}_q^{e_2})), \ldots, N_u(\tilde{u}_r^{h,l}(\mathbf{x}_q^{e_{n_e}}))],$$

where $N_u(\tilde{u}_r^{h,l}(\mathbf{x}_q^{e_i})) = [N_u(\tilde{u}_r^{h,l}(x_1^{e_i})), \ldots, N_u(\tilde{u}_r^{h,l}(x_{n_q}^{e_i}))]$ for $i = 1, \ldots, n_e$. In (22), matrices $\mathbf{Q}^\intercal\mathbf{H}\mathbf{W}$ and $\mathbf{H}^\intercal\mathbf{Q}$ can be precomputed, whose dimensions are $r \times (n_e n_q)$ and $(n_e n_q) \times r$, respectively. In online simulations, the nonlinear function derivative values need to be evaluated at the quadrature nodes in all the elements.

In the POD-FEIC model, the element of the Jacobian term at the lth iteration can be computed as follows.

$$\left[\mathcal{J}_{\text{FEIC}}^l\right]_{kj} = \left(\mathcal{I}^h N_u(\tilde{u}_r^{h,l})\varphi_j^h, \varphi_k^h\right). \tag{23}$$

It can be written in the following matrix form:

$$\mathcal{J}^l_{\text{FEIC}} = \mathbf{Q}^\intercal \mathbf{M}^h \mathbf{D}^l \mathbf{Q}, \tag{24}$$

where $\mathbf{M}^h$ is the FE mass matrix and $\mathbf{D}^l$ is a $n_{\text{dof}} \times n_{\text{dof}}$ diagonal matrix with the diagonal elements be the derivative values at the FE nodes, $[\mathbf{D}^l]_{ii} = N_u(u_r^{h,l}(x_i))$. As in (20), the matrix $\mathbf{Q}^\intercal \mathbf{M}^h$ in (24) only needs to be computed once, whose size is $r \times n_{\text{dof}}$. Furthermore, the nonlinear function derivative values only need to be calculated at the FE nodes during online simulations. Therefore, comparing with $\mathcal{J}_{\text{FE}}$, $\mathcal{J}_{\text{FEIC}}$ can be evaluated at a lower computational cost.

IV. ERROR ESTIMATES

In this section, we analyze the numerical error of the POD-FEIC model (18). Because the obvious distinction between the POD-FEIC (18) and the POD-FEM (14) lies in the special spatial discretization of the nonlinear term, we will consider the semidiscrete model (continuous in time) and only focus on errors caused by the spatial discretization and the POD truncation. Readers interested in the time discretization error analysis of the POD-G-ROM for parabolic PDEs are referred to [67, 68, 72, 73].

To provide the analysis, we proceed in three steps: we begin by gathering a few preliminary results that will be used; we then prove an error estimate for the L^2 projection of u in Lemma 4.6; finally, we establish the approximation error of (18) in Theorem 4.1. For clarity of notation, in the sequel, we will denote by C a generic constant that does not depend on the mesh size h and the number of POD basis functions r in the ROM.

A. Step 1: Preliminary Results

To derive the error estimation, we first make a few necessary hypotheses and present some preliminary results. For the solution u and the nonlinear function $N(u)$ of (7), we assume:

(H1) The solution u belongs to $C^1(0, T; H^{m+1}(\Omega) \cap H_0^1(\Omega))$,
(H2) $N(u)$ belongs to $C(0, T; H^{m+1}(\Omega))$,
(H3) $N(\cdot)$ is locally Lipschitz, that is, let $M = \|u\|_{L^\infty(\Omega)} + 1$, there exists $L = L(M)$ such that for all $x_* \in \Omega$ and $u(x_*), v(x_*) \in (-M, M)$,

$$|N(u(x_*)) - N(v(x_*))| \leq L|u(x_*) - v(x_*)|. \tag{25}$$

Based on the FE method theory, we have the following interpolation error [74]:

Lemma 4.1. *For $0 \leq \gamma \leq m+1$ and $1 \leq p \leq \infty$, if $v \in C(\overline{\Omega}) \cap \prod_{K \in \mathcal{T}_h} W_p^{m+1}(K)$, there exists a constant C independent of h such that*

$$\|v - \mathcal{I}^h v\|_{\gamma,p} \leq C h^{m+1-\gamma} \|v\|_{m+1,p}. \tag{26}$$

The snapshots in our considerations are composed of the FE solutions $u^h(x,t)$, which solves the following approximation problem: to find $u^h(x,t) \in V^h$, such that,

$$\left(\frac{\partial u^h}{\partial t}, v^h\right) + a\left(u^h, v^h\right) + \left(N(u^h), v^h\right) = (f, v^h), \quad \forall v^h \in V^h, \tag{27}$$

with $u^h(x,0) = u_0^h(x) \in V^h, \forall x \in \Omega$. The FE approximation theory for semilinear parabolic equations is well-developed (see, e.g., Chapter 14 in [75] and reference therein). One can easily modify the proof of Theorem 14.1 in [75] and obtain the following error estimate for the FE solution.

Lemma 4.2. *Let u^h and u be the solutions of (27) and (7) under the assumptions of (H1) and (H3), respectively. With appropriately chosen u_0^h in the FE approximation, we have, with $C = C(u,T)$,*

$$\|u^h(t) - u(t)\| + h\|u^h(t) - u(t)\|_1 \le Ch^{m+1}, \quad \text{for } t \in [0,T]. \tag{28}$$

To estimate the interpolation of nonlinear terms in the FEIC, we utilize an auxiliary "Euclidean" norm introduced in [63] on $C(\overline{\Omega})$:

$$\|\chi\|_h = \left[\sum_{i=1}^{n_{\text{dof}}} |\chi(x_i)|^2\right]^{\frac{1}{2}}. \tag{29}$$

For any $\chi \in V_h$, as the space is of finite dimensions, we have the equivalence between $\|\chi\|$ and $\|\chi\|_h$ on the reference element. With a straightforward homogeneity argument (or scaling argument [74]), we have the following lemma:

Lemma 4.3. *There exist two strictly positive constants c_1 and c_2 independent of h such that*

$$c_1 h^{\frac{d}{2}} \|\chi\|_h \le \|\chi\| \le c_2 h^{\frac{d}{2}} \|\chi\|_h, \tag{30}$$

for all $\chi \in V_h$.

Recall that the FE solutions $u^h(x,t)$ are used as snapshots, $\mathcal{H} = L^2$ is considered in the POD method and $\varphi_j(x)$ is the jth POD basis. Besides the POD projection error in $L^2(0,T,L^2(\Omega))$ given in Proposition 2.1,

$$\int_0^T \left\| u^h(\cdot,t) - \sum_{j=1}^{r} \left(u^h(\cdot,t), \varphi_j(\cdot)\right)\varphi_j(\cdot) \right\|^2 dt = \sum_{j>r} \lambda_j, \tag{31}$$

we have the projection error in $L^2(0,T,H^1(\Omega))$ norm as follows.

Lemma 4.4 ([69])**.** *The POD projection error in H^1 norm satisfies*

$$\int_0^T \left\| u^h(\cdot,t) - \sum_{j=1}^{r} \left(u^h(\cdot,t), \varphi_j(\cdot)\right)\varphi_j(\cdot) \right\|_1^2 dt = \sum_{j>r} ||\varphi_j||_1^2 \lambda_j. \tag{32}$$

For the POD approximation, we have the following POD inverse estimate [67]:

Lemma 4.5. *Let $M_r \in \mathbb{R}^{r\times r}$ with $[M_r]_{jk} = (\varphi_j, \varphi_k)$ be the POD mass matrix, $S_r \in \mathbb{R}^{r\times r}$ with $[S_r]_{jk} = [M_r]_{jk} + (\nabla\varphi_j, \nabla\varphi_k)$ be the POD stiffness matrix, and $\|\cdot\|_2$ denote the matrix spectral norm. Then, for all $v \in V_r$, the following estimates hold.*

$$\|v\| \le \sqrt{\|M_r\|_2 \|S_r^{-1}\|_2}\, \|v\|_1, \tag{33}$$

$$\|v\|_1 \leq \sqrt{\|S_r\|_2 \|M_r^{-1}\|_2}\|v\|. \tag{34}$$

As we choose $\mathcal{H} = L^2$ in the POD method, both $\|M_r\|_2$ and $\|M_r^{-1}\|_2$ are one.

B. Step 2: L^2 Projection Error

Next, we define the L^2 projection of u, w_r^h, from L^2 to V_r^h such that

$$\left(u - w_r^h, v_r^h\right) = 0, \quad \forall v_r^h \in V_r^h. \tag{35}$$

We have the following estimation of the L^2 projection error.

Lemma 4.6. *The L^2 projection of u, w_r^h, satisfies the following error estimations:*

$$\int_0^T \|u - w_r^h\|^2 dt \leq C\left(h^{2m+2} + \sum_{j>r}\lambda_j\right), \tag{36}$$

$$\int_0^T \|\nabla\left(u - w_r^h\right)\|^2 dt \leq C\left(h^{2m} + \|S_r\|_2 h^{2m+2} + \sum_{j>r}||\varphi_j||_1^2\lambda_j\right), \tag{37}$$

where $C = C(u, T)$.

Proof.

$$\begin{aligned}\|u - w_r^h\|^2 &= \left(u - w_r^h, u - w_r^h\right)\\ &\overset{(35)}{=} \left(u - w_r^h, u - v_r^h\right), \quad \forall v_r^h \in V_r^h.\end{aligned} \tag{38}$$

It indicates, by Cauchy–Schwartz inequality, for all $v_r^h \in V_r^h$,

$$\|u - w_r^h\| \leq \|u - v_r^h\|. \tag{39}$$

Decomposing $u - v_r^h = u - u^h + (u^h - v_r^h)$ and choosing $v_r^h = P_r u^h = \sum_{j=1}^r \left(u^h, \varphi_j\right)\varphi_j$ in (39), by the triangular inequality and Proposition 2.1, we have

$$\begin{aligned}\int_0^T \|u - w_r^h\|^2 dt &\leq C\left(\int_0^T \|u - u^h\|^2 dt + \int_0^T \|u^h - \sum_{j=1}^r \left(u^h, \varphi_j\right)\varphi_j\|^2 dt\right)\\ &\overset{(31)}{\leq} C\left(\int_0^T \|u - u^h\|^2 dt + \sum_{j>r}\lambda_j\right).\end{aligned} \tag{40}$$

Considering the FE approximation error (28), we have the bound for the L^2 projection error in $L^2(0, T; L^2(\Omega))$ as (36).

By the triangular inequality and adding then subtracting $P_r u^h = \sum_{j=1}^r \left(u^h, \varphi_j\right)\varphi_j$, we have

$$\int_0^T \|\nabla\left(u - w_r^h\right)\|^2 dt \leq C\int_0^T \left(\|\nabla\left(u - u^h\right)\|^2 + \|\nabla\left(u^h - P_r u^h\right)\|^2 + \|\nabla\left(P_r u^h - w_r^h\right)\|^2\right) dt$$

$$\overset{(32),(34)}{\leq} C\left(\int_0^T \|\nabla\left(u-u^h\right)\|^2 dt + \sum_{j>r} ||\varphi_j||_1^2 \lambda_j + \|S_r\|_2 \int_0^T ||P_r u^h - w_r^h||^2\right)$$

$$\leq C\left(\int_0^T \|\nabla\left(u-u^h\right)\|^2 dt + \sum_{j>r} ||\varphi_j||_1^2 \lambda_j + \|S_r\|_2 \int_0^T ||u-u^h||^2\right),$$

where we use $\|w_r^h - P_r u^h\| \leq \|u - u^h\|$ in the last inequality. Considering the FE approximation error estimation (28), we have the bound for the L^2 projection error in $L^2(0,T;H^1(\Omega))$ as (37). This proved the lemma. ∎

C. Step 3: Main Results

Finally, we discuss the main theoretical results of this article, which is about the approximation property of the POD-FEIC (18). We first estimate the difference between the L^2 projection of u, w_r^h, and the approximation solution u_r^h on a certain time interval in Lemma 4.7, which is bounded by the L^2 projection error. The conclusion is then extended to the whole time interval through a continuity argument in Lemma 4.8. Using the triangular inequality, we get the final estimation of approximation error $u - u_r^h$ in Theorem 4.1.

The crucial component in the analysis is the interpolation error of the nonlinear term $\left(N(u) - \mathcal{I}^h N(u_r^h), v_r^h\right)$, which can be decomposed in two different ways:

$$\left(N(u) - \mathcal{I}^h N(u_r^h), v_r^h\right) = \left(N(u) - \mathcal{I}^h N(u), v_r^h\right) + \left(\mathcal{I}^h N(u) - \mathcal{I}^h N(u_r^h), v_r^h\right), \tag{41}$$

and

$$\left(N(u) - \mathcal{I}^h N(u_r^h), v_r^h\right) = \left(N(u) - N(u_r^h), v_r^h\right) + \left(N(u_r^h) - \mathcal{I}^h N(u_r^h), v_r^h\right). \tag{42}$$

The first approach has been used in [63], in which the first term on the RHS of (41) is bounded by the standard FE interpolation error under the smoothness assumption of $N(u)$, and the second term can be estimated with the help of the local Lipschitz continuity assumption of $N(u)$ and an auxiliary "Euclidean" norm. The second approach has been used in [55, 60], in which the first term on the RHS of (42) also appears in the standard FE discretization, thus can be treated as usual. The second term can be estimated by the FE interpolation error of $N(u_r^h)$, whose accuracy relies on the regularity of $N(u_r^h)$. It is determined by a hypothesis on u_r^h and a stronger smoothness assumption of $N(u)$ than that required in the first approach. In this article, we will follow the first approach (41).

Let $v = v_r^h$ in (7) and subtract it from (18), we have the error equation of $e = u_r^h - u$ as follows.

$$\left(e_t, v_r^h\right) + a\left(e, v_r^h\right) + \left(\mathcal{I}^h N(u_r^h) - N(u), v_r^h\right) = 0, \quad \forall v_r^h \in V_r^h. \tag{43}$$

Let $\phi_r^h = u_r^h - w_r^h$ and $\eta = u - w_r^h$, we have the decomposition of error, $e = \phi_r^h - \eta$. Based on the definition of L^2 projection (35), we have $(\eta_t, v_r^h) = 0$. Therefore, the error equation (43) can be rewritten as:

$$\left(\phi_{r,t}^h, v_r^h\right) + a\left(\phi_r^h, v_r^h\right) = a\left(\eta, v_r^h\right) + \left(N(u) - \mathcal{I}^h N(u_r^h), v_r^h\right), \quad \forall v_r^h \in V_r^h. \tag{44}$$

Lemma 4.7. *Under assumptions (H1)–(H3), let u be the solution of (7) and u_r^h be the solution of (18) with the initial condition $u_r^h(\cdot,0)=\sum_{j=1}^{r}(u_0,\varphi_{r,j}^h)\varphi_{r,j}^h$ in (19), respectively. Assume that, for some t_1 with $0<t_1\le T$, we have*

$$\|\phi_r^h(t)\|_{L^\infty(\Omega)}<\frac{1}{2},\quad \textit{for all } 0<t\le t_1. \tag{45}$$

Then, it follows that

$$\|\phi_r^h(t_1)\|^2+\int_0^{t_1}||\nabla\phi_r^h(t)||^2dt\le C(u,T)\left(h^{2m}+\|S_r\|_2h^{2m+2}+\sum_{j>r}||\varphi_j||_1^2\lambda_j\right). \tag{46}$$

Proof. First note that one can certainly find a t_1 to make (45) true by adjusting the mesh size h and the number of POD basis functions r. We choose h to be small enough and r to be large enough to ensure that $\|u-w_r^h\|_{L^\infty(\Omega)}<\frac{1}{2}$, then together with (45), we have both w_r^h and u_r^h locate on the interval $(-M,M)$, where M is defined in (H3). This allows us to take advantage of the local Lipschitz condition of $N(u)$ on the interval $t\in[0,t_1]$. Next, we start estimations.

Let $v_r^h=\phi_r^h$ in (44), we have

$$\begin{aligned}\left(\phi_{r,t}^h,\phi_r^h\right)+a\left(\phi_r^h,\phi_r^h\right)&=a\left(\eta,\phi_r^h\right)+\left(N(u)-\mathcal{I}^hN(u),\phi_r^h\right)+\left(\mathcal{I}^hN(u)-\mathcal{I}^hN(w_r^h),\phi_r^h\right)\\&\quad+\left(\mathcal{I}^hN(w_r^h)-\mathcal{I}^hN(u_r^h),\phi_r^h\right).\end{aligned} \tag{47}$$

By the continuity (9) and coercivity (10) of $a(\cdot,\cdot)$, and Cauchy–Schwartz inequality, we have

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\|\phi_r^h\|^2+\beta\|\nabla\phi_r^h\|^2&\le\alpha\|\nabla\eta\|\|\nabla\phi_r^h\|+\|N(u)-\mathcal{I}^hN(u)\|\|\phi_r^h\|\\&\quad+\|\mathcal{I}^hN(u)-\mathcal{I}^hN(w_r^h)\|\|\phi_r^h\|\\&\quad+\|\mathcal{I}^hN(w_r^h)-\mathcal{I}^hN(u_r^h)\|\|\phi_r^h\|.\end{aligned} \tag{48}$$

For the first term on the RHS of (48), by Young's inequality, we have

$$\alpha\|\nabla\eta\|\|\nabla\phi_r^h\|\le\frac{\alpha^2}{2\beta}\|\nabla\eta\|^2+\frac{\beta}{2}\|\nabla\phi_r^h\|^2. \tag{49}$$

For the second term on the RHS of (48), Young's inequality yields

$$\|N(u)-\mathcal{I}^hN(u)\|\|\phi_r^h\|\le\frac{\|N(u)-\mathcal{I}^hN(u)\|^2}{2}+\frac{\|\phi_r^h\|^2}{2}. \tag{50}$$

Considering the FE interpolation error and the assumption (H2), we have

$$\|N(u)-\mathcal{I}^hN(u)\|\le Ch^{m+1}|N(u)|_{H^{m+1}(\Omega)}. \tag{51}$$

For the third term on the RHS of (48), Young's inequality gives

$$\|\mathcal{I}^hN(u)-\mathcal{I}^hN(w_r^h)\|\|\phi_r^h\|\le\frac{\|\mathcal{I}^hN(u)-\mathcal{I}^hN(w_r^h)\|^2}{2}+\frac{\|\phi_r^h\|^2}{2}. \tag{52}$$

Using Lemma 4.3, the local Lipschitz condition, and the triangular inequality, we have

$$\begin{aligned}\|\mathcal{I}^h N(u) - \mathcal{I}^h N(w_r^h)\| &\overset{(30)}{\leq} c_2 h^{\frac{d}{2}} \|\mathcal{I}^h N(u) - \mathcal{I}^h N(w_r^h)\|_h \overset{(17),(29)}{=} c_2 h^{\frac{d}{2}} \|N(u) - N(w_r^h)\|_h \\ &\overset{(25)}{\leq} L c_2 h^{\frac{d}{2}} \|u - w_r^h\|_h \overset{(17),(29)}{=} L c_2 h^{\frac{d}{2}} \|\mathcal{I}^h u - w_r^h\|_h \\ &\overset{(30)}{\leq} L c_2 c_1^{-1} \|\mathcal{I}^h u - w_r^h\| \\ &\leq L c_2 c_1^{-1} \left(\|u - \mathcal{I}^h u\| + \|u - w_r^h\|\right). \end{aligned} \tag{53}$$

By the regularity assumption (H1) of solution u and the FE approximability (Lemma 4.1), we have

$$\|u - \mathcal{I}^h u\| \leq C h^{m+1} |u|_{H^{m+1}(\Omega)}. \tag{54}$$

Thus, we have, for the third term on the RHS of (48),

$$\|\mathcal{I}^h N(u) - \mathcal{I}^h N(w_r^h)\| \leq C \left(h^{m+1} + \|\eta\|\right). \tag{55}$$

For the fourth term on the RHS of (48), we use similar arguments to those for the third term and get

$$\begin{aligned}\|\mathcal{I}^h N(w_r^h) - \mathcal{I}^h N(u_r^h)\| &\overset{(30)}{\leq} c_2 h^{\frac{d}{2}} \|\mathcal{I}^h N(w_r^h) - \mathcal{I}^h N(u_r^h)\|_h \overset{(17),(29)}{=} c_2 h^{\frac{d}{2}} \|N(w_r^h) - N(u_r^h)\|_h \\ &\overset{(25)}{\leq} L c_2 h^{\frac{d}{2}} \|w_r^h - u_r^h\|_h \overset{(30)}{\leq} L c_2 c_1^{-1} \|w_r^h - u_r^h\| \\ &= L c_2 c_1^{-1} \|\phi_r^h\|. \end{aligned} \tag{56}$$

Substituting (49)–(52), (55), and (56) in (48), we obtain

$$\frac{d}{dt}\|\phi_r^h\|^2 + \beta \|\nabla \phi_r^h\|^2 \leq \frac{\alpha^2}{\beta}\|\nabla \eta\|^2 + C\left(h^{2m+2} + \|\eta\|^2\right) + C_* \|\phi_r^h\|^2, \tag{57}$$

where $C_* = L c_2 c_1^{-1} + 1$. By Gronwall's lemma, on the interval $[0, t_1]$, we have

$$\begin{aligned}&\|\phi_r^h(t_1)\|^2 + \int_0^{t_1} \beta \|\nabla \phi_r^h\|^2 ds \\ &\quad \leq e^{C_* t_1} \|\phi_r^h(0)\|^2 + e^{C_* t_1} \int_0^{t_1} \left(\frac{\alpha^2}{\beta}\|\nabla \eta\|^2 + C h^{2m+2} + C\|\eta\|^2\right) ds. \end{aligned} \tag{58}$$

Considering $t_1 \leq T$ and the choice of the initial condition that indicates $\|\phi_r^h(0)\| = 0$, the above inequality yields

$$\begin{aligned}\|\phi_r^h(t_1)\|^2 + \int_0^{t_1} \beta \|\nabla \phi_r^h\|^2 ds &\leq e^{C_* T} \int_0^T \left(\frac{\alpha^2}{\beta}\|\nabla \eta\|^2 + C h^{2m+2} + C\|\eta\|^2\right) ds, \\ &\overset{(37)}{\leq} C \left(h^{2m} + \|S_r\|_2 h^{2m+2} + \sum_{j>r} ||\varphi_j||_1^2 \lambda_j\right), \end{aligned} \tag{59}$$

where $C = C(u, T)$ independent of t_1. ∎

Lemma 4.8. *Suppose the order of FEs $m \geq 1$ for $d = 1$ and $m \geq 2$ for $d \geq 2$, respectively. With the same conditions as those in Lemma 4.7, we have*

$$\|\phi_r^h(T)\|^2 + \int_0^T ||\nabla\phi_r^h(t)||^2 dt \leq C(u,T)\left(h^{2m} + \|S_r\|_2 h^{2m+2} + \sum_{j>r} ||\varphi_j||_1^2 \lambda_j\right). \tag{60}$$

Proof. Assume t_1^* is the largest value that makes (45) true. If $t_1^* \neq T$, it must be

$$\|\phi_r^h(t_1^*)\|_{L^\infty(\Omega)} = \frac{1}{2}. \tag{61}$$

However, by the inverse inequality,

$$\|\phi_r^h(t_1^*)\|_{L^\infty(\Omega)} \leq Ch^{-\frac{d}{2}}\|\phi_r^h(t_1^*)\| \leq Ch^{-\frac{d}{2}}\left(h^m + \sqrt{\|S_r\|_2}h^{m+1} + \sqrt{\sum_{j>r} ||\varphi_j||_1^2 \lambda_j}\right). \tag{62}$$

Then, for $m \geq 1$ if $d = 1$, and for $m \geq 2$ if $d = 2, 3$, one can always find a h small enough and a r large enough such that $\|\phi_r^h(t_1^*)\|_{L^\infty(\Omega)} < \frac{1}{2}$. This contradicts with the assumption (61). Therefore, $t_1^* = T$, that is, the conclusion (46) is true on the whole time interval $[0, T]$. ∎

Theorem 4.1. *Under assumptions (H1)–(H3), let u be the solution of (7) and u_r^h be the solution of (18) with the initial condition $u_r^h(\cdot, 0) = \sum_{j=1}^r (u_0, \varphi_{r,j}^h)\varphi_{r,j}^h$ in (19), respectively. The order of FEs $m \geq 1$ for $d = 1$, $m \geq 2$ for $d \geq 2$. There exist positive numbers h_0 and r_0 such that, for $h \leq h_0$ and $r \geq r_0$, we have*

$$\|u_r^h(T) - u(T)\|^2 + \int_0^T ||\nabla u_r^h(t) - \nabla u(t)||^2 dt \leq C\left(h^{2m} + \|S_r\|_2 h^{2m+2} + \sum_{j>r} ||\varphi_j||_1^2 \lambda_j\right), \tag{63}$$

where $C = C(u, T)$ is a positive constant independent of h and λ_j.

Proof. The conclusion follows the triangular inequality, Lemma 4.8, and Lemma 4.6. ∎

Remark 4.1. In Theorem 4.1, the estimation for the gradient of errors, $\|\nabla u_r^h - \nabla u\|_{L^2(0,T;L^2(\Omega))}$ is optimal with respect to (w.r.t.) both the spatial discretization ($\sim \mathcal{O}(h^m)$) and the POD truncation ($\sim \mathcal{O}(\sqrt{\sum_{j>r} ||\varphi_j||_1^2 \lambda_j})$). By Poincaré–Friedrichs inequality, it is easy to show the error in $L^2(0, T; H^1(\Omega))$ has the same optimality. Although not proven here, the error in $L^2(0, T; L^2(\Omega))$ is also optimal w.r.t. the spatial discretization ($\sim \mathcal{O}(h^{m+1})$) and the POD truncation ($\sim \mathcal{O}(\sqrt{\sum_{j>r} \lambda_j})$). We shall verify it numerically in the next section.

Remark 4.2. Theorem 4.1 also holds on the FEIC approximation of the POD-G-ROM for the problem (8). The proof follows a similar argument.

V. NUMERICAL TESTS

The goal of this section is twofold: first, the proposed method will be validated by two problems that have appeared in interdisciplinary research. Accuracy and efficiency of the new approach are to be tested; second, the theoretical result in Section IV will be numerically verified by another example.

For a comparison of accuracy, the approximation errors of the POD-FEIC (18) and the POD-FEM (14) are computed, respectively, in two different norms:

$$\mathcal{E}_0(u,v)=\sqrt{\frac{1}{M}\sum_{\ell=1}^{M}||u(\cdot,t_\ell)-v(\cdot,t_\ell)||^2},\quad \mathcal{E}_1(u,v)=\sqrt{\frac{1}{M}\sum_{\ell=1}^{M}||u(\cdot,t_\ell)-v(\cdot,t_\ell)||_1^2},$$

where $\mathcal{E}_0$ is a discrete approximation of the error in $L^2(0,T;L^2(\Omega))$, while $\mathcal{E}_1$ is a discrete approximation of the error in $L^2(0,T;H^1(\Omega))$.

To measure the efficiency, we consider the CPU time, which is the time elapsed for the (online) integration only, excluding the (offline) time for generating basis functions, precomputing matrices, calculating errors, and so forth. We define a speed-up factor of the new POD-ROMs as follows.

$$S_f \equiv \frac{\text{CPU time of the POD-FEM (14) simulation}}{\text{CPU time of the new POD-ROM simulation}}, \tag{64}$$

where the new POD-ROM is either the POD-FEIC (18) or the POD-FEIC-DEIM that will be defined in next section. Furthermore, to evaluate the effectiveness of the FEIC, we compare the profiling time for generating the nonlinear terms $\mathcal{N}_{\text{FE}}$ (16) and $\mathcal{J}_{\text{FE}}$ (22) in the POD-FEM simulations with that for computing $\mathcal{N}_{\text{FEIC}}$ (20) and $\mathcal{J}_{\text{FEIC}}$ (24) in the POD-FEIC simulations. The profiling time is provided by the MATLAB profile function, which is usually longer than the actual simulation time.

For a fair comparison, the same numerical quadrature formulae are used in the implementations of both approaches. All numerical tests reported in this article are implemented on a PC with a 2.6 GHz Intel Core i7 processor. In this article, the snapshot matrix in the POD basis generation is composed of the full-order simulation results at all time steps. Note that this is not necessary in practice. For a discussion on the snapshot locations, the reader is referred to [14, 72, 76].

A. Validation

For the validation purpose, we consider two examples: a one-dimensional (1D) FitzHugh–Nagumo (F-N) system, which possesses a cubic polynomial nonlinearity; and a two-dimensional (2D) Buckley–Leverett equation (BLE), which has a nonpolynomial nonlinearity.

FitzHugh–Nagumo System. We first consider the simplified 1D Hodgkin–Huxley model used in [4], which is a F-N system. The model is a nonlinear PDE system and describes the activation and deactivation dynamics of a spiking neuron. The system reads:

$$\frac{\partial v}{\partial t}-\mu v_{xx}-\frac{1}{\mu}v(v-0.1)(1-v)+\frac{1}{\mu}w=\frac{c}{\mu},\quad x\in[0,L],t\in[0,T],$$

$$\frac{\partial w}{\partial t}-bv+\gamma w=c,\quad x\in[0,L],t\in[0,T],$$

TABLE I. The approximation errors, CPU time, and speed-up factors of the POD-FEM (14) and the POD-FEIC (18).

	POD-FEM			POD-FEIC			
r	$\mathcal{E}_0(\tilde{u}_r^h, u^h)$	$\mathcal{E}_1(\tilde{u}_r^h, u^h)$	CPU time	$\mathcal{E}_0(u_r^h, u^h)$	$\mathcal{E}_1(u_r^h, u^h)$	CPU time	S_f
3	3.64×10^{-3}	9.96×10^{-2}	3.81	3.64×10^{-3}	9.96×10^{-2}	1.20	3.18
5	6.61×10^{-4}	2.27×10^{-2}	3.90	6.60×10^{-4}	2.27×10^{-2}	1.23	3.17
7	1.20×10^{-4}	4.91×10^{-3}	3.93	1.20×10^{-4}	4.91×10^{-3}	1.25	3.14
9	2.03×10^{-5}	1.25×10^{-3}	4.04	2.10×10^{-5}	1.25×10^{-3}	1.27	3.18

Note that the POD-FEIC keeps the same accuracy as the POD-FEM, but saves CPU time by over three times.

$$v_x(0,t) = -i_0(t), \quad v_x(L,t) = 0, \quad t \in [0,T],$$
$$v(x,0) = 0, \quad w(x,0) = 0, \quad x \in [0,L], \tag{65}$$

where $v(x,t)$ and $w(x,t)$ are voltage and recovery voltage, respectively. Let $\mathbf{u} = [v,w]^\intercal$, the weak form of the original system has the form of (7):

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v}\right) + \mathbf{M}_1 a(\mathbf{u},\mathbf{v}) + \mathbf{M}_2(\mathbf{u},\mathbf{v}) + (\mathbf{N},\mathbf{v}) = (\mathbf{f},\mathbf{v}), \tag{66}$$

where $a(\mathbf{u},\mathbf{v}) = \mu(\mathbf{u}_x,\mathbf{v}_x)$, $\mathbf{N} = [\frac{1}{\mu}v(v-0.1)(1-v),0]^\intercal$, $\mathbf{f} = [\frac{c}{\mu},c]^\intercal$, $\mathbf{M}_1 = \left[\begin{smallmatrix}\mu & 0\\ 0 & 0\end{smallmatrix}\right]$, and $\mathbf{M}_2 = \left[\begin{smallmatrix}0 & \frac{1}{\mu}\\ -b & \gamma\end{smallmatrix}\right]$. We choose the same parameters as those utilized in [4], that is, $L=1$, $T=8$, $\mu = 0.015$, $b=0.5$, $\gamma=2$, $c=0.05$ and the stimulus $i_0(t) = 50,000t^3e^{-15t}$.

To generate snapshots, we use linear FEs on a uniform mesh for spatial discretization with mesh size $h = 1/512$, and the Crank–Nicolson scheme for time integration with time step $\Delta t = 1 \times 10^{-2}$. A three-point Gauss quadrature rule is applied for evaluating the inner product. The CPU time for the full-order simulation is 249.72 s. In total, 801 snapshots are collected and used to compute the POD basis in L^2 space. Over 99.9% kinetic energy is captured by only three POD basis functions. As the exact solution is unknown, we regard the FE solution, u^h, as the benchmark.

The approximation errors, simulation time, and speed-up factors of the POD-FEM and the POD-FEIC are listed in Table I when the same number of POD basis functions, r, is used in both ROMs. It is seen that the FEIC discretization achieves the same accuracy as that of the FE discretization, but improves the computational efficiency by over three times. The phase portraits of the POD-FEIC ($r=5$) solutions at different spatial points are plotted in Fig. 1, which are compared with those of the FE solutions and the POD-FEM ($r=5$) solutions at the same spatial points. It shows that the correct limit cycles of the original system have been captured by the POD-FEIC model.

Based on the MATLAB profiler, the command for evaluating the nonlinear terms $\mathcal{N}_{\text{FE}}$ (16) and $\mathcal{J}_{\text{FE}}$ (22) was executed 2418 times in the POD-FEM simulation when $r=5$. The profiling took 5.02 s. In the simulation of the POD-FEIC model based on the same POD basis functions, the command for computing $\mathcal{N}_{\text{FEIC}}$ (20) and $\mathcal{J}_{\text{FEIC}}$ (24) was called 2418 times, too. However, the profiling time decreased to be 1.11 s.

Buckley–Leverett Equation. We then consider the 2D BLE, which is usually used to describe two phase flow in porous media with a gravitation pull in x-direction [77].

$$\frac{\partial u}{\partial t} - \mu\Delta u + \frac{\partial f_1(u)}{\partial x} + \frac{\partial f_2(u)}{\partial y} = 0, \quad x \in \Omega, t \in [0,T],$$

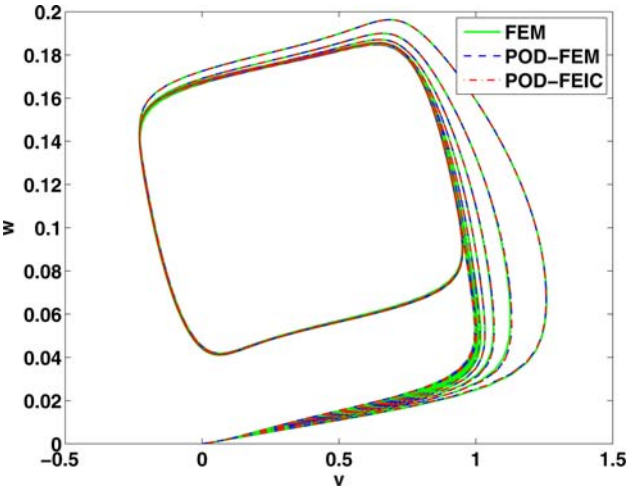

FIG. 1. The limit cycles on the $v-w$ plane: the POD approximation ($r=5$) with either the FEM discretization (POD-FEM) or the FEIC discretization (POD-FEIC) coincides with that of the finite element solution (FEM). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$u(x,t)=0, \quad x\in\partial\Omega, t\in[0,T],$$
$$u(x,0)=e^{-16(x^2+y^2)}, \quad x\in\Omega, \tag{67}$$

where $f_1(u)=\frac{u^2}{u^2+(1-u)^2}$ and $f_2(u)=f_1(u)[1-5(1-u)^2]$. The weak form of the original system has the form of (8) with $a(u,v)=\mu(\nabla u,\nabla v)$ and $N(u)=[-f_1,-f_2]^\intercal$.

In this test, we choose the same parameters as used in [77]: $\mu=0.1$, $T=0.5$, $\Omega=[-1.5,1.5]\times[-1.5,1.5]$. To generate snapshots, we use quadratic FEs on a uniform triangular mesh for spatial discretization with mesh size $h=1/64$, and the Crank–Nicolson scheme for time integration with time step $\Delta t=1\times10^{-2}$. A seven-point Gauss quadrature rule is applied for evaluating the inner product. It takes 1.25×10^4 s to finish the full-order simulation. Fifty one snapshots are collected and used to compute POD basis in L^2 space. Over 99.9% kinetic energy is captured by only five POD basis functions. Due to the lack of exact solution, we also consider the FE solution, u^h, to be the benchmark.

The POD-FEM and the POD-FEIC approximation errors are listed in Table II when r POD basis functions used in both ROMs. It is seen that the POD-FEIC obtains the same accuracy as that of the POD-FEM, but, decreases the CPU time by over three times. A comparison among the FE approximation, the POD-FEM ($r=5$) approximation, and the POD-FEIC ($r=5$) approximation solutions at $t=0.2$ is shown in Fig. 2.

Based on the MATLAB profiler, the command for evaluating the nonlinear term $\mathcal{N}_{\rm FE}$ (16) and $\mathcal{J}_{\rm FE}$ (21) was executed 165 times in the POD-FEM simulation when $r=5$. The profiling took 42.0 s. In the simulation of the POD-FEIC model generated by the same POD basis, the command for computing $\mathcal{N}_{\rm FEIC}$ (20) and $\mathcal{J}_{\rm FEIC}$ (24) was also called 165 times. But, the profiling time reduced to be 9.49 s.

B. Verification

In this subsection, we will verify the theoretical results obtained in Section IV through a problem with a nonpolynomial nonlinearity. It is governed by the following equations:

TABLE II. The approximation errors, CPU time, and speed-up factors of the POD-FEM (14) and the POD-FEIC (18).

	POD-FEM			POD-FEIC			
r	$\mathcal{E}_0(\tilde{u}_r^h, u^h)$	$\mathcal{E}_1(\tilde{u}_r^h, u^h)$	CPU time	$\mathcal{E}_0(u_r^h, u^h)$	$\mathcal{E}_1(u_r^h, u^h)$	CPU time	S_f
5	4.47×10^{-3}	7.68×10^{-2}	34.9	4.47×10^{-3}	7.68×10^{-2}	8.78	3.97
10	3.64×10^{-4}	1.05×10^{-2}	35.6	3.64×10^{-4}	1.05×10^{-2}	9.25	3.85
15	3.46×10^{-5}	1.37×10^{-3}	42.1	3.46×10^{-5}	1.37×10^{-3}	11.3	3.73
20	2.72×10^{-6}	1.35×10^{-4}	47.6	2.74×10^{-6}	1.35×10^{-4}	12.9	3.69

Note that the POD-FEIC keeps the same accuracy as the POD-FEM, but saves CPU time by over three times.

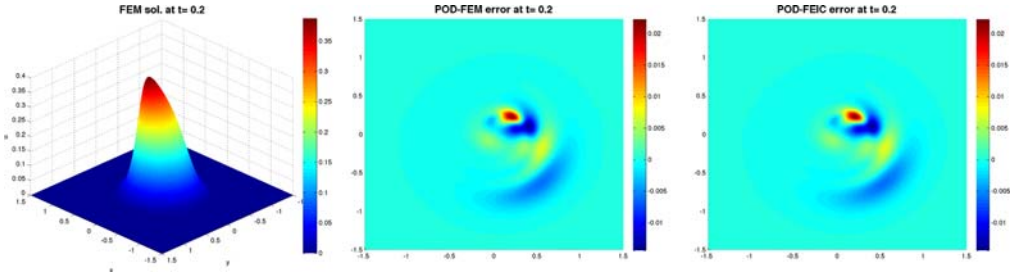

FIG. 2. The simulations at $t = 0.2$: the FEM solution (left), the error of POD-FEM with $r = 5$ (middle), and the error of POD-FEIC with $r = 5$ (right). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$\begin{aligned} \frac{\partial u}{\partial t} - \Delta u + \sin(u) &= f, \quad x \in \Omega, t \in [0, T], \\ u &= 0, \quad x \in \partial\Omega, t \in [0, T], \\ u(x, 0) &= g, \quad x \in \Omega, \end{aligned} \tag{68}$$

where f is determined by substituting a designated exact solution into the LHS of (68) and g is the exact solution at $t = 0$. In the test, we consider the problem in 1D with exact solution $u = 0.5 \sin(\pi x)(10 \tanh(x - t) + 1)$ on the domain $\Omega = [0, 1]$ during the time interval $t \in [0, 1]$. The exact solution is also our benchmark when calculating errors. The weak formulation is of the form (7) with $a(u, v) = (u_x, v_x)$ and $N(u) = \sin(u)$. We investigate the convergence properties of the POD-FEIC solution w.r.t. mesh size h and the number of POD basis r, respectively.

To check the approximation order of the POD-FEIC solution w.r.t. h, we collect the FE solution of the original system with linear elements ($m = 1$) and quadratic elements ($m = 2$), respectively. Backward-Euler method is used for the time integration with a small time step $\Delta t = 1 \times 10^{-6}$. A seven-point Gauss quadrature rule is applied for evaluating the inner product. The number of POD basis functions are chosen such that $\sum_{j>r} \lambda_j < 1 \times 10^{-7}$. In this way, the spatial discretization error dominates the whole approximation property.

The errors in both $\mathcal{E}_0$ and $\mathcal{E}_1$ norms are shown in Table III. Linear regressions indicate that, for linear elements, the order of convergence is 1.97 in $\mathcal{E}_0$ norm and 0.98 in $\mathcal{E}_1$ norm; for quadratic elements, the error convergence order is 2.94 in $\mathcal{E}_0$ norm and 1.95 in $\mathcal{E}_1$ norm. The approximation orders are close to the optimal values in Theorem 4.1. As the error analysis is asymptotic, as h decreases, the order approaches the optimal values (order $m + 1$ in $\mathcal{E}_0$ norm and order m in $\mathcal{E}_1$ norm).

TABLE III. The approximation rate of the POD-FEIC w.r.t. h.

	m = 1		m = 2	
h	$\mathcal{E}_0(u_r^h, u)$	$\mathcal{E}_1(u_r^h, u)$	$\mathcal{E}_0(u_r^h, u)$	$\mathcal{E}_1(u_r^h, u)$
1/8	2.09×10^{-2}	5.41×10^{-1}	2.41×10^{-3}	1.26×10^{-1}
1/16	5.50×10^{-3}	2.81×10^{-1}	3.30×10^{-4}	3.42×10^{-2}
1/32	1.39×10^{-3}	1.42×10^{-1}	4.22×10^{-5}	8.76×10^{-3}
1/64	3.50×10^{-4}	7.12×10^{-2}	5.31×10^{-6}	2.20×10^{-3}
Order	$h^{1.97}$	$h^{0.98}$	$h^{2.94}$	$h^{1.95}$

TABLE IV. The approximation rate of the POD-FEIC w.r.t. r when linear finite elements are used.

	m = 1			
r	$\sqrt{\sum_{j>r} \lambda_j}$	$\mathcal{E}_0(u_r^h, u)$	$\sqrt{\sum_{j>r} \|\|\varphi_j\|\|_{H^1}^2 \lambda_j}$	$\mathcal{E}_1(u_r^h, u)$
3	4.13×10^{-2}	4.60×10^{-2}	6.26×10^{-1}	6.21×10^{-1}
4	2.45×10^{-2}	2.74×10^{-2}	4.48×10^{-1}	4.47×10^{-1}
6	9.07×10^{-3}	1.02×10^{-2}	2.23×10^{-1}	2.31×10^{-1}
7	5.59×10^{-3}	6.29×10^{-3}	1.55×10^{-1}	1.69×10^{-1}
Order	–	1.00	–	0.96

TABLE V. The approximation rate of the POD-FEIC w.r.t. r when quadratic finite elements are used.

	m = 2			
r	$\sqrt{\sum_{j>r} \lambda_j}$	$\mathcal{E}_0(u_r^h, u)$	$\sqrt{\sum_{j>r} \|\|\varphi_j\|\|_{H^1}^2 \lambda_j}$	$\mathcal{E}_1(u_r^h, u)$
3	4.15×10^{-2}	4.60×10^{-2}	6.28×10^{-1}	6.19×10^{-1}
4	2.46×10^{-2}	2.74×10^{-2}	4.50×10^{-1}	4.43×10^{-1}
6	9.18×10^{-3}	1.02×10^{-2}	2.25×10^{-1}	2.21×10^{-1}
7	5.68×10^{-3}	6.29×10^{-3}	1.57×10^{-1}	1.54×10^{-1}
Order	–	1.00	–	1.00

To check the approximation order of the POD-FEIC solution w.r.t. r, we collect the FE solution of the original system when linear elements ($m = 1$) and quadratic elements ($m = 2$) are utilized for spatial discretization, respectively, and backward-Euler method for the time integration. The mesh size $h = 1/64$ and the time step $\Delta t = 1 \times 10^{-6}$ are fixed. The number of POD basis functions is chosen so that $\sqrt{\sum_{j>r} \lambda_j}$ decays by a factor of 2.

The errors in both $\mathcal{E}_0$ and $\mathcal{E}_1$ norms when linear elements are used are shown in Table IV and those for quadratic elements are listed in Table V. Linear regressions indicate that the convergence order of error in $\mathcal{E}_1$ norm w.r.t. $\sqrt{\sum_{j>r} \|\|\varphi_j\|\|_{H^1}^2 \lambda_j}$ is 0.96 for linear elements and 1.00 for quadratic elements. Wherein, the approximation order for quadratic elements is linear, which is optimal as indicated in Theorem 4.1. That of linear elements is close to 1 but slightly smaller than 1, which is because the discretization error tends to dominate the error as r increases. At the same time, note that the approximation order of error in $\mathcal{E}_0$ norm w.r.t. $\sqrt{\sum_{j>r} \lambda_j}$ is 1 for both linear and quadratic elements. Although not proven theoretically, the error in $\mathcal{E}_0$ norm also converges optimally.

VI. THE COMBINATION WITH THE DEIM

The DEIM has been successfully applied in many nonlinear ROMs to reduce the computational complexity of the nonlinear terms [4, 51, 78–81]. For a detailed presentation of the DEIM method, the reader is referred to [4]. In general, it uses the following ansatz on a nonlinear function $N(u(x,t))$:

$$N(u) = \sum_{j=1}^{\hat{r}} \psi_j(x) c_j(t), \tag{69}$$

where $\psi_j(x)$ is the jth nonlinear POD basis obtained by applying the POD method on the nonlinear snapshots and $\hat{r}$ is the rank of the nonlinear POD basis. Based on the nonlinear POD basis vectors $\mathbf{\Psi} = [\psi_1, \ldots, \psi_{\hat{r}}]$, the DEIM optimally selects a set of interpolation points $\wp := [\wp_1, \ldots, \wp_p]$ and approximates the vector of nonlinear function values, $\mathbf{N}(u)$, by

$$\mathbf{N}(u) \approx \mathbf{\Psi}(\mathbf{P}^\intercal \mathbf{\Psi})^{-1} \mathbf{P}^\intercal \mathbf{N}(u), \tag{70}$$

where $\mathbf{P} = [e_{\wp_1}, \ldots, e_{\wp_p}]$ is the matrix for selecting the corresponding p indices $\wp_1, \ldots, \wp_p$, and $e_{\wp_i}$ is the $\wp_i$th column in the identity matrix.

When the FD method is used for the spatial discretization, the nonlinear snapshot data consist of vectors of nonlinear function values, $\mathbf{N}(u)$, on a certain time interval. However, when the FE method is used, the nonlinear snapshot becomes $(N(u(x,t)),h)$ as the nonlinear term in the weak formulation (27). Therefore, generating each nonlinear snapshot requires $\mathcal{O}\left(4n_e n_q + \varrho(n_e n_q)\right)$ flops, which costs lots of offline time. Moreover, in cases such as complex flows are studied, many interpolation points might be required to obtain a good approximation of the nonlinear term. As the inner product has to be calculated over the elements sharing the selected DEIM points, the online simulation time increases. These issues represent the main computational hurdles for applying DEIM in the FE setting, which, however, can be easily overcome by the POD-FEIC approach we proposed in Section III.

Indeed, thanks to the separation of spatial and temporal variables in the nonlinear term of the POD-FEIC model, the value vector of the nonlinear function, $\mathbf{N}(u)$, can be chosen as the nonlinear snapshot. Hence, the computational complexity for generating each nonlinear snapshot is reduced to $\mathcal{O}(\varrho(n_{\text{dof}}))$ flops. Replacing the nonlinear function with the DEIM approximation (70) in the POD-FEIC (18), we get the POD-FEIC-DEIM model, which has a more efficient approximation of the nonlinear term

$$\mathcal{N}_{\text{FEIC-DEIM}} = \mathbf{Q}^\intercal \mathbf{M}^h \mathbf{\Psi}(\mathbf{P}^\intercal \mathbf{\Psi})^{-1} \mathbf{P}^\intercal \mathbf{N}(\mathbf{Q}\mathbf{a}(t)). \tag{71}$$

It is seen that, once $\mathbf{C}_p = \mathbf{Q}^\intercal \mathbf{M}^h \mathbf{\Psi}(\mathbf{P}^\intercal \mathbf{\Psi})^{-1}$ is precomputed, in online simulations, one only needs to calculate the nonlinear functions at p selected DEIM points, $\mathbf{z}_p = \mathbf{P}^\intercal \mathbf{N}(\mathbf{Q}\mathbf{a}(t))$, which does not involve any numerical quadratures. The corresponding computational complexity of the nonlinear term in each time step (or iteration) is only $\mathcal{O}(4rp + \varrho(p))$ flops, where $\mathcal{O}(\varrho(p))$ flops are used in computing the nonlinear function values at the DEIM points and $\mathcal{O}(4rp)$ flops are utilized to implement two matrix-vector products: $\mathbf{P}^\intercal \mathbf{Q}\mathbf{a}(t)$ and $\mathbf{C}_p \mathbf{z}_p$. Therefore, the POD-FEIC-DEIM improves the computational efficiency over the POD-FEIC, which, obviously, outperforms the POD-FEM.

TABLE VI. The approximation errors, CPU time, and speed-up factors of the POD-FEIC-DEIM model with $r = 5$ POD basis and p interpolation points for the nonlinear function approximation.

p	$\mathcal{E}_0(\hat{u}_r^h, u^h)$	CPU time	S_f
3	5.94×10^{-3}	1.01	3.86
5	3.00×10^{-3}	1.01	3.86
7	9.66×10^{-4}	1.02	3.82
9	6.63×10^{-4}	1.02	3.82

A. Validation

We demonstrate the effectiveness of the new approach by considering the first two examples in Section V again. The same spatial and temporal discretization methods and the numerical quadrature rules as those used in the POD-FEIC simulation are considered in the POD-FEIC-DEIM simulations. In the following, we denote the POD-FEIC-DEIM solution by $\hat{u}_r^h$.

FitzHugh–Nagumo System. We revisit the 1D F-N model that has been used in Section V-A. In total, 801 nonlinear snapshots of $N(v) = \frac{1}{\mu} v(v - 0.1)(1 - v)$ are generated and used in the DEIM. The nonlinear function is then approximated by the DEIM basing on p selected interpolation points. In this test, we consider the POD-FEIC-DEIM generated by the same $r = 5$ POD basis functions as those used in FitzHugh–Nagumo System Section and investigate the numerical performance of the new model by varying the number of interpolation points.

The approximation errors in $\mathcal{E}_0$ norm, CPU time, and speed-up factors for simulations are presented in Table VI. As p increases, the POD-FEIC-DEIM result approaches to that of the POD-FEIC ($\mathcal{E}_0(u_r^h, u^h) = 6.60 \times 10^{-4}$ when $r = 5$). It is also seen from Table VI that when $p = 3$, the POD-FEIC-DEIM error is nine times larger than that of the POD-FEIC. However, as the limit cycle on the $v - w$ plane shown in Fig. 3, the difference mainly occurs at the beginning of the simulation (from $t = 0$ to $t = 1$). After the transient interval, the limit cycles of the POD-FEIC and the POD-FEIC-DEIM coincide with each other. Moreover, the CPU time of the POD-FEIC-DEIM is lower than that of the POD-FEIC, although not significantly due to the small size of the tested problem.

To measure the effectiveness of the POD-FEIC-DEIM approximation of the nonlinear terms, we utilize the MATLAB profiler. It shows that, when $r = 5$ and $p = 3$, the command for evaluating the nonlinear terms was executed 2416 times, which took 0.63 s. When $r = 5$ and $p = 9$, the command for evaluating the nonlinear terms was called 2418 times, which took 0.64 s. Hence, in general, it takes less time in generating the nonlinear terms in the POD-FEIC-DEIM model than that in the POD-FEM and POD-FEIC models.

Buckley–Leverett Equation. We also consider the 2D BLE problem that has been utilized in Section V-A. In total, 51 nonlinear snapshots of $f_1(u)$ and $f_2(u)$ are generated, respectively. They are used in the DEIM for selecting p interpolation points. We consider the POD-FEIC-DEIM model generated by the same $r = 5$ POD basis functions as those used in Buckley–Leverett Equation Section and investigate its numerical behavior by varying p.

The approximation errors in $\mathcal{E}_0$ norm, CPU time, and speed-up factors for simulations are listed in Table VII. Note that for the POD-FEIC model when $r = 5$, the error $\mathcal{E}_0(u_r^h, u^h) = 4.47 \times 10^{-3}$ and the CPU time is 8.78 s with the speed-up factor 3.97. The POD-FEIC-DEIM model achieves a close accuracy by only using $p = 5$ interpolation points, while improving the computational efficiency: the error is 4.74×10^{-3} but the speed-up factor reaches 51.4. Figure 4 shows the

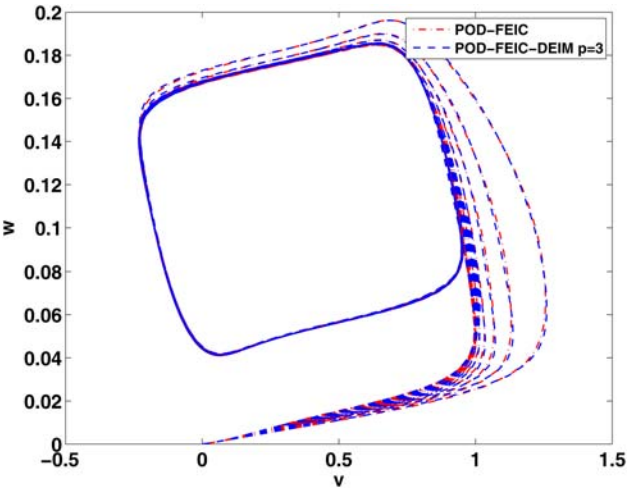

FIG. 3. The limit cycle on the $v - w$ plane: the POD-FEIC results ($r = 5$) and the POD-FEIC-DEIM approximation ($r = 5$ and $p = 3$). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE VII. The approximation errors, CPU time, and speed-up factors of the POD-FEIC-DEIM model with $r = 5$ POD basis and p interpolation points for the nonlinear function approximation.

p	$\mathcal{E}_0(\hat{u}_r^h, u^h)$	CPU time	S_f
5	4.74×10^{-3}	0.679	51.4
10	4.68×10^{-3}	0.692	50.4
15	4.67×10^{-3}	0.705	49.5
20	4.61×10^{-3}	0.711	49.0

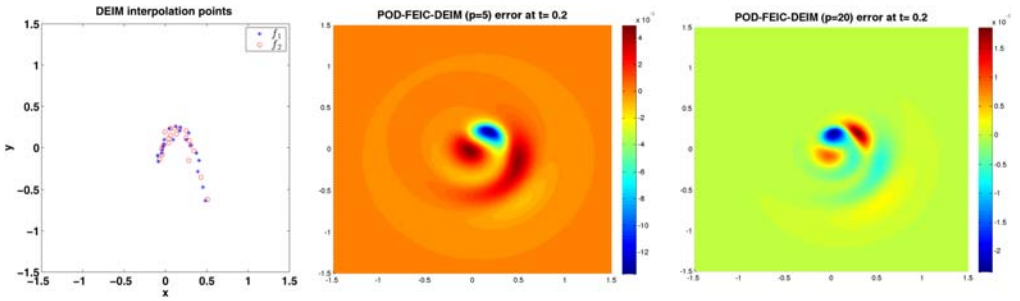

FIG. 4. Distribution of DEIM interpolation points for $f_1(u)$ and $f_2(u)$ (left), the difference between POD-FEIC and POD-FEIC-DEIM at $t = 0.2$ when $p = 5$ (middle) and $p = 20$ (right). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

distribution of 20 first selected DEIM interpolation points (left), and the difference between the POD-FEIC approximation and the POD-FEIC-DEIM solution when $p = 5$ (middle) and $p = 20$ (right).

To measure the effectiveness of the POD-FEIC-DEIM approximation of the nonlinear terms, we use the MATLAB profiler. It shows that, when $r = 5$ and $p = 5$, the command for evaluating the nonlinear terms was executed 166 times, which took 0.05 s. When $r = 5$ and $p = 20$, the command

for evaluating the nonlinear terms was called 166 times, which took 0.07 s. Therefore, in general, it takes less time in generating the nonlinear terms in the POD-FEIC-DEIM model than that in the POD-FEM and POD-FEIC models.

VII. CONCLUSIONS

As a first step of our investigations on efficient FE discretization algorithms for nonlinear model reduction techniques, we develop the FEIC method for nonlinear POD-ROMs. Comparing with the standard FE discretization, which requires the nonlinear function values to be determined at all the quadrature points in order to evaluate the inner product in the nonlinear terms, the proposed approach is computationally more efficient because the nonlinear function values are needed at the FE nodes. The proposed method also achieves the same accuracy as that of the standard FE discretization when nonlinear functions satisfy certain smoothness assumptions. Furthermore, the approach is more suitable for the DEIM. Combining the FEIC method with the DEIM will further reduce the computational complexity for evaluating nonlinear terms.

We plan to continue investigating several research avenues. We will discuss the performance of the proposed method on three-dimensional semilinear parabolic equations. We will further extend the proposed approach to more general nonlinear systems including nonlinear parametric equations and the nonlinear closure ROMs that have been developed for complex flows in [2, 39]. Some realistic engineering application problems will be tested.

The author is grateful to Dr. Saifon Chaturantabut for sharing the MATLAB implementation codes of the discrete empirical interpolation method. The author thank the anonymous reviewers for their constructive comments, which helped improve the manuscript.

References

1. M. Gunzburger, Perspectives in flow control and optimization, Advances in design and control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.
2. Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, Two-level discretizations of nonlinear closure models for proper orthogonal decomposition, J Comput Phys 230 (2011), 126–146.
3. N. Aubry, W. Y. Lian, and E. S. Titi, Preserving symmetries in the proper orthogonal decomposition, SIAM J Sci Comput 14 (1993), 483–505.
4. S. Chaturantabut, D. C. Sorensen, and J. C. Steven, Nonlinear model reduction via discrete empirical interpolation, SIAM J Sci Comput 32 (2010), 2737–2764.
5. D. Amsallem, M. J. Zahr, and C. Farhat, Nonlinear model order reduction based on local reduced-order bases, Int J Numer Methods Eng 92 (2012), 891–916.
6. M. Bergmann, C. H. Bruneau, and A. Iollo, Enablers for robust POD models, J Comput Phys 228 (2009), 516–538.
7. M. Björklund, A hierarchical POD reduction method of finite element models with application to simulated mechanical systems, PhD thesis, Umeå University, Sweden, 2012.
8. J. Borggaard, A. Hay, and D. Pelletier, Interval-based reduced order models for unsteady fluid flow, Int J Numer Anal Model 4 (2007), 353–367.
9. T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders, Goal-oriented, model-constrained optimization for reduction of large-scale systems, J Comput Phys 224 (2007), 880–896.
10. J. Burkardt, M. Gunzburger, and H. C. Lee, POD and CVT-based reduced-order modeling of Navier–Stokes flows, Comput Methods Appl Mech Eng 196 (2006), 337–355.

11. K. Carlberg and C. Farhat, A low-cost, goal-oriented compact proper orthogonal decompositionbasis for model reduction of static systems, Int J Numer Methods Eng 86 (2011), 381–402.

12. D. N. Daescu and I. M. Navon, A dual-weighted approach to order reduction in 4DVAR data assimilation, Mon Weather Rev 136 (2008), 1026–1041.

13. A. Iollo, S. Lanteri, and J. A. Désidéri, Stability properties of POD-Galerkin approximations for the compressible Navier–Stokes equations, Theor Comput Fluid Dyn 13 (2000), 377–396.

14. K. Kunisch and S. Volkwein, Optimal snapshot location for computing POD basis functions, ESAIM Math Model Numer Anal 44 (2010), 509.

15. B. R. Noack, K. Afanasiev, M. Morzyński, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, J Fluid Mech 497 (2003), 335–363.

16. J. Borggaard, A. Duggleby, A. Hay, T. Iliescu, and Z. Wang, Reduced-order modeling of turbulent flows, In: Proceedings of MTNS 2008, Virginia, USA, 2008.

17. J. Borggaard, T. Iliescu, and Z. Wang, Artificial viscosity proper orthogonal decomposition, Math Comput Model 53 (2011), 269–279.

18. L. Cordier, E. Majd, B. Abou, and J. Favier, Calibration of POD reduced-order models using Tikhonov regularization, Int J Numer Methods Fluids 63 (2010), 269–296.

19. M. Couplet, C. Basdevant, and P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, J Comput Phys 207 (2005), 192–220.

20. F. Fang, C. C. Pain, I. M. Navon, A. H. Elsheikh, J. Du, and D. Xiao, Non-linear Petrov-Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods, J Comput Phys 234 (2013), 540–559.

21. B. Galletti, A. Bottaro, C. H. Bruneau, and A. Iollo, Accurate model reduction of transient and forced wakes, Eur J Mech B Fluids 26 (2007), 354–366.

22. B. Galletti, C. H. Bruneau, L. Zannetti, and A. Iollo, Low-order modeling of laminar flow regimes past a confined square cylinder, J Fluid Mech 503 (2004), 161–170.

23. T. Iliescu and Z. Wang, Variational multiscale proper orthogonal decomposition: convection-dominated convection-diffusion-reaction equations, Math Comput 82 (2013), 1357–1378.

24. T. Iliescu and Z. Wang, Variational multiscale proper orthogonal decomposition: Navier-Stokes equations, Numer Methods Partial Differential Equations (30) (2014), 641–663.

25. A. Iollo, A. Dervieux, J. A. Désidéri, and S. Lanteri, Two stable POD-based approximations to the Navier–Stokes equations, Comput Vis Sci 3 (2000), 61–66.

26. V. L. Kalb and A. E. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models, Phys Fluids 19 (2007), 054106.

27. J. L. Lumley and B. Podvin, Dynamical systems theory and extra rates of strain in turbulent flows, Exp Therm Fluid Sci 13 (1996), 180–189.

28. B. R. Noack, M. Morzyński, and G. Tadmor, Reduced-order modelling for flow control, Vol. 528, Springer Verlag, Wien, New York, USA, 2011.

29. B. R. Noack, P. Papas, and P. A. Monkewitz, Low-dimensional Galerkin model of a laminar shear-layer, Technical Report 2002-01, École Polytechnique Fédérale de Lausanne, Switzerland, 2002.

30. B. R. Noack, P. Papas, and P. A. Monkewitz, The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows, J Fluid Mech 523 (2005), 339–365.

31. B. Podvin, On the adequacy of the ten-dimensional model for the wall layer, Phys Fluids 13 (2001), 210–224.

32. B. Podvin, A proper-orthogonal-decomposition–based model for the wall layer of a turbulent channel flow, Phys Fluids 21 (2009), 015111.

33. B. Podvin and J. L. Lumley, A low-dimensional approach for the minimal flow unit, J Fluid Mech 362 (1998), 121–155.

34. D. Rempfer, Investigations of boundary layer transition via Galerkin projections on empirical eigenfunctions, Phys Fluids 8 (1996), 175.

35. D. Rempfer, On low-dimensional Galerkin models for fluid flow, Theor Comput Fluid Dyn 14 (2000), 75–88.

36. D. Rempfer and H. F. Fasel, Dynamics of three-dimensional coherent structures in a flat-plate boundary layer, J Fluid Mech 275 (1994), 257–283.

37. S. Sirisup and G. E. Karniadakis, A spectral viscosity method for correcting the long-term behavior of POD models, J Comput Phys 194 (2004), 92–116.

38. S. Ullmann and J. Lang, A POD-Galerkin reduced model with updated coefficients for Smagorinsky LES, J. C. F. Pereira and A. Sequeira, editors, In: V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010, Lisbon, Portugal, 2010.

39. Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, Comput Methods Appl Mech Eng 237–240 (2012), 10–26.

40. J. Weller, S. Camarri, and A. Iollo, Feedback control by low-order modelling of the laminar flow past a bluff body, J Fluid Mech 634 (2009), 405–418.

41. M. Rewieński and J. White, A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices, IEEE Trans Comput Aided Des Integr Circuits Syst 22 (2003), 155–170.

42. M. Rewieński and J. White, Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations, Linear Algebra Appl 415 (2006), 426–454.

43. P. Astrid, Reduction of process simulation models: a proper orthogonal decomposition approach, PhD thesis, Technische Universiteit Eindhoven, Netherlands, 2004.

44. P. Astrid, S. Weiland, K. Willcox, and T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Trans Automat Control 53 (2008), 2237–2251.

45. K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, Comput Fluids 35 (2006), 208–226.

46. 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 (2004), 667–672.

47. M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM Math Model Numer Anal 41 (2007), 575–605.

48. J. L Eftang and B. Stamm, Parameter multi-domain hp empirical interpolation, Int J Numer Methods Eng 90 (2012), 412–428.

49. J. S. Hesthaven, B. Stamm, and S. Zhang, Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods, ESAIM: Mathematical Modelling and Numerical Analysis, Cambridge University Press, vol. 48, pp. 259–283, 2014.

50. N. C. Nguyen, A. T. Patera, and J. Peraire, A best points interpolation method for efficient approximation of parametrized functions, Int J Numer Methods Eng 73 (2008), 521–543.

51. S. Chaturantabut and D. C. Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction, SIAM J Numer Anal 50 (2012), 46–63.

52. B. T. Dickinson and J. R. Singler, Nonlinear model reduction using group proper orthogonal decomposition, Int J Numer Anal Model 7 (2010), 356–372.

53. P. Tiso and D. J. Rixen, Discrete empirical interpolation method for finite element structural dynamics, Topics in nonlinear dynamics, Vol. 1, Springer, 2013, pp. 203–212.

54. H. Antil, M. Heinkenschloss, and D. C. Sorensen, Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems, A. Quarteroni and G. Rozza, editors, Reduced order methods for modeling and computational reduction, Volume 9 of modeling, simulation & applications, Springer Italia, Milan, 2014, pp. 101–136.

55. C. M. Chen, S. Larsson, and N. Y. Zhang, Error estimates of optimal order for finite element methods with interpolated coefficients for the nonlinear heat equation, IMA J Numer Anal 9 (1989), 507–524.

56. Z. X. Chen and J. Douglas Jr ., Approximation of coefficients in hybrid and mixed methods for nonlinear parabolic problems, Mat Aplicada E Comput 10 (1991), 137–160.

57. I. Christie, D. F. Griffiths, A. R. Mitchell, and J. M. Sanz-Serna, Product approximation for non-linear problems in the finite element method, IMA J Numer Anal 1 (1981), 253–266.

58. C. A. J. Fletcher, The group finite element formulation, Comput Methods Appl Mech Eng 37 (1983), 225–244.

59. J. Douglas Jr . and T. Dupont, The effect of interpolating the coefficients in nonlinear parabolic Galerkin procedures, Math Comput 29 (1975), 360–389.

60. S. Larsson, V. Thomée, and N. Y. Zhang, Interpolation of coefficients and transformation of the dependent variable in finite element methods for the non-linear heat equation, Math Methods Appl Sci 11 (1989), 105–124.

61. J. C. López-Marcos and J. M. Sanz-Serna, Stability and convergence in numerical analysis III: linear investigation of nonlinear stability, IMA J Numer Anal 8 (1988), 71–84.

62. J. M. Sanz-Serna and L. Abia, Interpolation of the coefficients in nonlinear elliptic Galerkin procedures, SIAM J Numer Anal 21 (1984), 77–83.

63. Y. Tourigny, Product approximation for nonlinear Klein-Gordon equations, IMA J Numer Anal 10 (1990), 449–462.

64. Z. Xiong, Y. Chen, and Y. Zhang, Convergence of FEM with interpolated coefficients for semilinear hyperbolic equation, J Comput Appl Math 214 (2008), 313–317.

65. D. Chapelle, A. Gariah, and J. Sainte-Marie, Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples, ESAIM Math Model Numer Anal 46 (2012), 731–757.

66. P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, Turbulence, coherent structures, dynamical systems and symmetry, Cambridge University Press, Cambridge, UK, 2012.

67. K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, Numer Math 90 (2001), 117–148.

68. K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J Numer Anal 40 (2002), 492–515 (electronic).

69. J. R. Singler, New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs, SIAM J Numer Anal 52 (2014), 852–876.

70. L. Sirovich, Turbulence and the dynamics of coherent structures, I. Coherent structures, Q Appl Math 45 (1987), 561–571.

71. K. Kunisch and S. Volkwein, Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition, J Optim Theory Appl 102 (1999), 345–371.

72. Z. Luo, J. Chen, I. M. Navon, and X. Yang, Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier-Stokes equations, SIAM J Numer Anal 47 (2008), 1–19.

73. S. S. Ravindran, Error analysis for Galerkin POD approximation of the nonstationary boussinesq equations, Numer Methods Partial Differential Equations (27) (2011), 1639–1665.

74. S. C. Brenner and L. R. Scott, The mathematical theory of finite element methods, Volume 15 of texts in applied mathematics, 2nd Ed., Springer-Verlag, New York, 2002.

75. V. Thomée, Galerkin finite element methods for parabolic problems, Springer Verlag, Heidelberg, Germany, 2006.

76. R. H. W. Hoppe and Z. Liu, Snapshot location by error equilibration in proper orthogonal decomposition for linear and semilinear parabolic partial differential equations, J Numer Math 22 (2014), 1–32.

77. N. C. Nguyen and J. Peraire, An efficient reduced-order modeling approach for non-linear parametrized partial differential equations, Int J Numer Methods Eng 76 (2008), 27–55.

78. S. Chaturantabut and D. C. Sorensen, Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media, Math Comput Model Dyn Syst 17 (2011), 337–353.

79. M. Hinze and M. Kunkel, Discrete empirical interpolation in POD model order reduction of drift-diffusion equations in electrical networks, Scientific Computing in Electrical Engineering SCEE Proceedings 2010, Springer, Heidelberg, Germany, 2012, pp. 423–431.

80. A. R. Kellems, S. Chaturantabut, D. C. Sorensen, and S. J. Cox, Morphologically accurate reduced order modeling of spiking neurons, J Comput Neurosci 28 (2010), 477–494.

81. R. Stefănescu and I. M. Navon, POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model, J Comput Phys 237 (2013), 95–114.