Computers and Mathematics with Applications 71 (2016) 2407–2420

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Stabilized reduced order models for the advection–diffusion–reaction equation using operator splitting

Benjamin McLaughlin *, Janet Peterson, Ming Ye

Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, United States

ARTICLE INFO

Article history:
Available online 28 February 2016

Keywords:
Reduced order modeling (ROM)
Advection–diffusion–reaction equation
SUPG
SOLD
Operator splitting
Finite element method (FEM)

ABSTRACT

Reduced order modeling (ROM) coupled with finite element methods has been used effectively in many disciplines to efficiently solve complex problems. However, for advection-dominated flows numerical simulations often contain spurious, nonphysical oscillations which will also be apparent in the ROM simulations. In this work we consider stabilization methods for ROM for the advection–diffusion–reaction (ADR) equation when it is solved both with and without operator splitting. Specifically we consider the streamline-upwind Petrov–Galerkin (SUPG) stabilization method and the spurious oscillations at layers diminishing (SOLD) stabilization method. We build on these methods by constructing a coherent framework which successfully integrates these model reduction, stabilization, and operator splitting approaches, and we provide numerical examples detailing the application of this framework in the ADR setting. The stabilized ROM results are compared numerically with their corresponding full finite element simulations.

1. Introduction

In many complex models understanding the behavior of the system requires obtaining many realizations of the state equations which necessitates performing simulations over a range of model parameter values. Because performing many simulations for complex partial differential equations (PDEs) is typically computationally expensive, methods have been developed to reduce the work. One such class of methods is reduced order modeling (ROM).

The idea of using a reduced basis for computation was first introduced by Noor in structural analysis [1]. Interest in model reduction has expanded to an array of applications such as fluid dynamics [2], aeronautics [3], and climate modeling [4]. The goal of the model reduction method used here is to determine a low-dimensional approximation space over which to pose the state equations. The basis for this space is typically found by generating realizations of the state equations for a range of parameter values and then compressing the data; this procedure is typically implemented as a preprocessing step. In this work we used proper orthogonal decomposition (POD) to compress the data so the method is often referred to as POD-ROM. In Section 2 we present a brief overview of the ROM procedure that we use.

The advection–diffusion–reaction (ADR) equation is of importance because it models reactive solute transport. If the problem being modeled is diffusion dominated, then standard methods can be used to generate the ROM basis and the corresponding ROM solution. However, if the problem is advection dominated it is well known that standard finite element or finite difference approximations contain spurious, nonphysical oscillations unless a fine enough discretization is used. For this reason there has been considerable research into developing methods for adding stabilization to remove these nonphysical oscillations.

* Corresponding author.
E-mail addresses: bm09m@my.fsu.edu (B. McLaughlin), jpeterson@fsu.edu (J. Peterson), mye@fsu.edu (M. Ye).

http://dx.doi.org/10.1016/j.camwa.2016.01.032
0898-1221/

Early attempts to remove spurious oscillations in finite element simulations used upwind discretizations but these tended to add too much diffusion. A method called SUPG which added diffusion only in the streamline direction was introduced in a paper in 1982 [5]. This method was a significant improvement over earlier attempts. Subsequently, there have been many extensions and analytical results published for SUPG; see [6–8] and the references therein. For some problems nonphysical oscillations also occur in the vicinity of steep gradients in the solution which SUPG does not address. To reduce these spurious oscillations a class of methods have been developed which are called SOLD methods; see [9] for a review of these methods. In Section 3 we briefly review these two stabilization methods and discuss how to implement them into the ROM setting and provide numerical results. A third stabilization method, flux correction transport, is also discussed but we argue that it is not easily implemented in the ROM setting. Much of the underlying theoretical analysis supporting the use of streamline diffusion methods in POD-based ROM was introduced in [10], and a recent paper contains further theoretical analysis of streamline diffusion ROM methods applied to convection–diffusion problems and studies the optimal selection of stabilization parameters in this setting [11]. In contrast, the goal of our work is to construct and demonstrate a framework incorporating these methods which can be applied to a variety of ADR problems, including both advection-dominated and diffusion-dominated cases.

Even after stabilization is added to the problem, the computational costs of generating solutions to the full and reduced equations can be significant. To reduce this cost, operator splitting is often used, especially for the case where the reactions are nonlinear. For example, in the ADR equation the transport phase can be solved followed by the reaction phase; in addition, the transport phase can be broken down into advection and diffusion phases creating a three-step process at each time step. For a nonlinear reaction an advantage is that once the reaction phase is separated from the transport phase, the PDE describing transport becomes linear. Moreover, the reaction phase can be solved as an initial value problem at each node thus simplifying computations. Another advantage of operator splitting is that one can tailor the numerical scheme to the phase. The stability of operator splitting for the ADR equation was studied in [12]. In Section 4 we present a novel construction which incorporates model reduction, operator splitting, and stabilization in an integrated framework. This framework accommodates both SUPG and SOLD stabilization strategies, and can be used to apply both two- and three-phase operator splitting for the ADR equation. In this manner, our approach is a strategy for improving both the computational cost of generating realizations as well as the numerical stability of the realizations. We provide numerical examples which allow us to study the accuracy and the computational cost of our approach.

2. Brief overview of POD-based ROM

In the context of solving partial differential equations (PDEs) using reduced order modeling, one generates a set of reduced basis vectors as a preprocessing step. The goal is for the dimension of the reduced space to be much smaller than the size of the system used to solve the PDE by standard methods such as finite elements or finite differences. Once the basis is chosen, the ROM solution is sought as a linear combination of these basis vectors and so the state equations are posed as a Galerkin problem in a standard way.

In this section we give a brief overview of the model reduction procedure and then describe it for the ADR equation. We also indicate how inhomogeneous initial and boundary conditions are handled in the ROM setting.

2.1. Model reduction procedure

In the model reduction process we generate a set of particular solutions to a differential equation which contains one or more parameters; we call these solutions "snapshots" and they are obtained by using standard methods which typically require solving large systems, which may be banded under certain circumstances (such as when a structured computational grid is used). Throughout this work we use standard finite elements methods to generate the snapshots. We sample the solution space along both the parameter domain and time domain, such that each snapshot is a solution of the PDE corresponding to a single parameter set and a particular time, assuming the PDE is time dependent. We assemble the vector representations of these snapshots as columns in a snapshot matrix, S.

Because the columns of the snapshot matrix contain a great deal of redundant information, we compress the information to obtain a set of reduced basis vectors. In this work we do this by using a proper orthogonal decomposition (POD). However, there are alternative methods which may be used to compress the snapshot data, such as CVT [2]. In POD one determines the singular value decomposition of the snapshot matrix S and the first d left singular vectors, ψ_i, of this matrix form the orthonormal reduced basis set. The reduced order solution u^n_{ROM} at time t_n is sought as a linear combination of these basis vectors and the problem is posed as a Galerkin problem over the reduced space.

In general, these basis functions have global support over the domain, which leads to a dense system of equations. Thus the only way ROM is feasible is if $d \ll m$, where m is the dimension of the system of algebraic equations using a high dimensional method.

One can show that the error of the d-dimensional POD space is given by the sum of the squares of the singular values corresponding to the unused left singular vectors of S [13]. We may compute this relative error as

$$e_{\text{POD}} = \sum_{i=d+1}^{l} \sigma_i^2 \Big/ \sum_{i=1}^{l} \sigma_i^2, \tag{1}$$

where $\sigma_i,\ i = 1, \ldots, \ell$ are the ordered singular values of S. However, it is important to realize that e_{POD} is a measure only of the information that is lost in truncating the full snapshot set to form the reduced basis. This measure does not lend any insight regarding the goodness of the original snapshot set, and does not indicate how much information about the solution may have been excluded from the snapshot set.

2.2. Standard ROM for the ADR equation

In this work we consider the time dependent ADR equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{D}\nabla u) - \nabla \cdot (\mathbf{v}u) + R \quad \text{for } (\mathbf{x}, t) \in \Omega \times (0, T], \tag{2}$$

where $\mathbf{D}$ is a tensor describing the diffusivity of u, the vector $\mathbf{v}$ is its linear velocity and R is a source or sink term resulting from a chemical reaction which may depend upon space, time, and/or the unknown u; in particular, R can be a nonlinear function of u making the PDE nonlinear. We restrict Ω to be a bounded domain in $\mathbb{R}^2$. In this work we assume that the diffusion is anisotropic and aligned with the directions of the coordinate axes and that it is constant in each of these directions; i.e., we use constant diffusion $D_{x_i}, i = 1, 2$. In addition to (2), we impose an initial condition $u(\mathbf{x}, 0) = u_0(\mathbf{x})$ and boundary conditions. If the boundary conditions are homogeneous then the standard finite element problem is to seek $u \in V = H_0^1(\Omega)$ satisfying

$$(u_t, v) + G(u, v) = 0 \quad \forall v \in V, \tag{3}$$

where $(\cdot, \cdot)$ denotes the $L^2(\Omega)$ inner product and the bilinear form $G(u, v)$ is defined by

$$G(u, v) = \sum_{i=1}^{2} D_{x_i} \int_\Omega \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i}\, d\Omega + \int_\Omega \nabla \cdot (\mathbf{v}u) v\, d\Omega - \int_\Omega Rv\, d\Omega \tag{4}$$

for all $u, v, \in V$. Then to discretize, we choose a finite dimensional subspace $V^h \subset V$ and pose the weak problem over this space.

For the standard ROM approximation of (3) we pose the problem over a d-dimensional space $V_{\text{rom}} = \text{span}\{\psi_i(x)\}_{i=1}^d$ and so the ROM problem is completely analogous to the discrete finite element problem, the only difference in the discrete weak problem being the approximating spaces. Specifically, for homogeneous Dirichlet boundary conditions the fully discrete ROM problem where we use backward Euler in time is to seek $u_{\text{ROM}}^n = \sum_{j=1}^d c_{j,n}\psi_j$ satisfying

$$(u_{\text{ROM}}^n, \psi_i) + \Delta t G(u_{\text{ROM}}^n, \psi_i) = (u_{\text{ROM}}^{n-1}, \psi_i) \quad \text{for } i = 1, \ldots, d. \tag{5}$$

2.3. Imposing initial and boundary conditions in ROM

It is important to consider how inhomogeneous boundary and initial conditions are enforced in the framework of the reduced model. The homogeneous cases are straightforward because their implementation in the reduced model is analogous to the finite element case. Handling inhomogeneous initial conditions in a standard ROM setting only has to be done once but in the operator splitting setting we will see that it has to be done at each time step.

To enforce inhomogeneous initial conditions in ROM, it is necessary to express the initial condition as a linear combination of the global ROM basis functions. Converting from a ROM solution to a nodal solution is straightforward because each ROM basis vector is a linear combination of the standard finite element basis; however, converting a nodal solution to the ROM space requires extra work. One approach is to project the nodal initial condition into the range of the reduced basis by using a *QR* factorization. However, in the operator splitting case we will see that this has to be done at each time step so it must be done more efficiently. Another approach is to realize that what we need is to calculate integrals such as $(u_{\text{ROM}}^{n-1}, \psi_i)$ when u_{ROM}^{n-1} is known at the nodes. Consequently, we must evaluate u_{ROM}^{n-1} at a quadrature point when we know its nodal values. If we are using continuous piecewise quadratic basis functions and a midpoint quadrature rule, then we automatically have the value at the quadrature point; for other basis functions and quadrature rules we simply interpolate.

Enforcing inhomogeneous Dirichlet boundary conditions is more complex in the ROM setting. The snapshots are particular solutions to the problem, and accordingly satisfy inhomogeneous Dirichlet boundary conditions which can be time and space dependent. Thus a linear combination of basis vectors generated from these snapshots will not, in general, satisfy the desired boundary conditions. There are two approaches for overcoming this challenge, which are thoroughly discussed in [14].

The most commonly used approach is to remove the inhomogeneous Dirichlet boundary data from the snapshots before forming the reduced basis by subtracting from each snapshot a linear combination of particular solutions to make the snapshot satisfy homogeneous boundary data. Oftentimes, an appropriate multiple of the average $\bar{u}$ of the snapshots is used. Then the ROM solution is found as a linear combination of the ROM basis functions satisfying homogeneous Dirichlet boundary conditions plus appropriate linear combinations of particular solutions.

For example, if inhomogeneous Dirichlet boundary conditions are imposed on one portion of the boundary, Γ_1, and homogeneous Dirichlet or Neumann conditions are imposed elsewhere we take

$$u^n_{\text{ROM}} = \sum_{i=1}^{d} \xi_{in}\psi_i(\vec{x}) + \eta_n\bar{u} = \tilde{u}^n_{\text{ROM}} + \eta_n\bar{u}, \tag{6}$$

where the coefficient η_n is chosen to satisfy the inhomogeneous Dirichlet boundary at time t^n on Γ_1 and the basis vectors ψ_i satisfy homogeneous Dirichlet boundary conditions on Γ_1. Then the fully discrete reduced order problem for (3) at time t^n using a backward Euler difference in time is to seek $u^n_{\text{ROM}} \in V_{\text{ROM}}$ satisfying

$$\left(\tilde{u}^n_{\text{ROM}} - \tilde{u}^{n-1}_{\text{ROM}}, \psi_i\right) + \Delta tG(\tilde{u}^n_{\text{ROM}}, \psi_i) = -\Delta tG(\eta_n\bar{u}, \psi_i) - (\eta_n\bar{u} - \eta_{n-1}\bar{u}, \psi_i) \tag{7}$$

for $i = 1, \ldots, d$.

3. Stabilized ROM without operator splitting

The standard ROM approach typically works well in the case where the ADR equation is diffusion dominated, i.e., when the components of **D** are large compared with **v**. However, if the flow is convection dominated then the finite element approximations used for the snapshots typically contain spurious, nonphysical oscillations and thus render the same effect in the ROM discretization. This is to be expected because in simple settings the standard finite element approach reduces to a centered difference approximation. Numerical stability can be achieved by sufficient refinement of the spatial grid, though this may add considerably to the computational load.

Interest in obtaining numerically stable solutions which do not rely on grid refinement has generated research into stabilization methods. A concise historical perspective of the development of stabilization strategies is presented in John and Knobloch [9]. Early methods used upwind finite element discretizations to reduce the oscillations; however, these add too much diffusion and their accuracy is limited to first order. A number of methods have been developed which endeavor to add diffusion into the numerical model in judicious and clever ways, in order to apply artificial diffusion only where it is required for stability and to avoid unnecessary smearing. Notable examples of these methods are flux correction, streamline upwind Petrov–Galerkin, and spurious oscillations at layers diminishing methods. We briefly discuss these methods here and indicate their adaptability to the ROM setting.

3.1. Flux correction transport

Flux correction transport methods (FCT) have seen extensive use for many years as a stabilization method for finite difference models (e.g. [15,16]). More recently, researchers have effectively adapted the flux correction technique for application to finite element models [17].

The flux correction procedure relies on the ability to generate a secondary approximation to the solution which obeys the so-called LED (local extremum diminishing) property. This LED approximation preserves certain stability properties of the solution (e.g., smoothness, positivity, monotonicity). To this end the LED approximation must admit a sufficient amount of diffusion, which may be accomplished through the use of a lower-order accurate numerical scheme. In the standard FEM approach, lumping of the mass matrix is typically used in the lower order scheme. The difference between the LED scheme and the standard finite element scheme can be expressed numerically in terms of fluxes between neighboring grid nodes. These fluxes are used to update the nodal values of the LED approximation, removing the excess diffusion, while performing a suitable limiting of flux updates to prevent the standard approximation's oscillations from returning.

The flux correction is quite effective at suppressing oscillations in the finite element approximation, but it has a couple of drawbacks. First, the flux correction approach does not preserve the order of accuracy of the underlying finite element scheme. Additionally, this procedure requires that we compute two approximations to the solution at each time step, for time-dependent problems. Fortunately, since the LED approximation typically employs a lower-order numerical scheme, it is often much cheaper to compute than the standard approximation, which may partially mitigate this difficulty.

However, FCT is not easily adaptable to the ROM setting. This is because the ROM basis functions are global, rather than the standard nodal FEM basis. Moreover, the construction of an LED approximation in the ROM setting is not straightforward. For these reasons, we consider other stabilization approaches.

3.2. Streamline upwind Petrov–Galerkin method

An alternative to adding artificial diffusion in all directions is to add artificial diffusion only in the direction of the streamlines of the solution because this is the direction where the most severe oscillations tend to form. Brooks and Hughes [5] incorporated this idea into a method they called streamline upwind/Petrov–Galerkin (SUPG).

To see how streamline diffusion is added consider the weak problem (3). Let $\mathbb{T}_h$ denote a triangulation of the domain Ω and K an element of $\mathbb{T}_h$. We then write a consistently stabilized form by adding to (3) a stabilization term over each element.

We have

$$(u_t, v) + G(u, v) + \sum_{K \in \mathbb{T}_h} \big(\mathcal{R}(u), W(v)\big)_K = 0 \quad \forall v \in V \tag{8}$$

where $\mathcal{R}(u)$ is the residual of (2), W is a suitable weighting function, and $(\cdot, \cdot)_K$ denotes an L^2 inner product on the element K. There are several different choices for the weighting function. Here we use

$$W(v) = \tau_K \mathbf{v} \cdot \nabla v, \tag{9}$$

where τ_K is the stabilization parameter on element K. If u and v are smooth enough, then we can view this method as using test functions of the form $v + \tau_K \mathbf{v} \cdot \nabla v$, where v is a basis function of V. The trial functions are still of the form v, so the spaces of the test functions and trial functions are different; hence, the method is of the Petrov–Galerkin type.

The parameter τ_K is a spatially- and temporally-varying parameter which varies per element on a nonuniform grid. Since it is not typically feasible to compute its optimal value directly, heuristic methods are often used to approximate it in practice. These heuristics use the value of advection and diffusion in a particular element and the element diameter h_K to compute τ_K. In our study we use the choice

$$\tau_K = \frac{h_K}{2|\mathbf{v}|}\left(\coth Pe_K - \frac{1}{Pe_K}\right) \quad \text{where } Pe_K = \frac{|\mathbf{v}| h_K}{2\|\mathbf{D}\|} \tag{10}$$

which was suggested in [18]. While the work in [18] considers only the scalar-diffusion case, in our usage we let $|\cdot|$ denote Euclidean length and $\|\cdot\|$ denote the Frobenius norm. We use this choice for both the full finite element calculations and the corresponding ROM simulations. A paper [11] which just appeared delves into methods for choosing the stabilization parameter in the ROM setting but our work does not incorporate their results.

3.3. Crosswind stabilization methods

It is well-known and easily observed in numerical experiments that while the SUPG method excels at enforcing stability in numerical approximations, it still admits nonphysical overshoots and undershoots in the vicinity of steep gradients in the solution. This is due to the fact that SUPG stabilization is not a monotone method nor monotonicity preserving.

To remove oscillations which occur at these layers, a number of methods have been developed which are often categorized as "shock-capturing" or "discontinuity-capturing" methods. In [9] the authors argue that since the solutions do not actually contain shocks or discontinuities a more fitting classification of these methods would be *spurious oscillations at layers diminishing*, or SOLD, and for clarity we will adopt their terminology. These methods are intended to promote monotonicity in the solutions, and include terms which are nonlinear in the solution so that the accuracy is not limited to first order. The idea is to incorporate one of these methods in addition to the SUPG stabilization defined in (8).

A review of common SOLD methods is given in John and Knobloch [9] and based on the computations therein, the authors found the method we employ here to be one of the best; it is a modification of the crosswind-diffusion method developed by Codina [19]. In two dimensions, the diffusion acts along the direction perpendicular to $\mathbf{v} = (v_1, v_2)$, denoted $\mathbf{v}^\perp$, and given by $(-v_2, v_1)/|\mathbf{v}|$. The SOLD term that is added to (8) is

$$\big(\widetilde{D}\mathbf{v}^\perp \cdot \nabla u, \mathbf{v}^\perp \cdot \nabla v\big) \tag{11}$$

where the parameter $\widetilde{D}$ is a function of the unknown u thus introducing a nonlinearity into the weak problem. On each element K it is given by

$$\tilde{D}_K = \max\left\{0, C\frac{h_K|\hat{\mathcal{R}}(u)|}{2|\nabla u|} - \|\mathbf{D}\|\right\} \tag{12}$$

where $\hat{\mathcal{R}}(u) = \mathbf{v}\cdot\nabla u$ and C is a suitable constant (see [18] for choices of C). Because the introduction of the SOLD term results in the linear ADR equation becoming nonlinear, the dependence of $\tilde{D}_K$ on the unknown can be lagged to the previous time step. We incorporate this approach in our computations using SOLD stabilization. In practice, there are additional existing methods for resolving nonlinearities in the reduced model, and these methods alternatively could be applied to resolve the nonlinearity in the sold term in the stabilized weak problem.

3.4. ROM with stabilization

Given the strict dependence of the reduced model on the information contained in the snapshots, it is reasonable to ask whether ensuring the stability of the snapshots is sufficient to stabilize the reduced model. Any of the stabilized finite element methods can be used to generate snapshots which are, themselves, free of oscillations. Because the reduced approximation is a linear combination of the global reduced basis functions, the reduced model does not have the freedom to produce internodal oscillations like those observed in the finite element approximation (unless the basis contains internodal

oscillations). While this characteristic of the reduced model causes the numerical instability to appear in a different manner than in the high-dimensional model, it does not signify an improvement in the quality of the model.

In our numerical experiments, we observe that when using a basis constructed from stabilized snapshots, the reduced model still lacks stability. While this model is incapable of producing oscillations from node-to-node, the approximation exhibits manifestations of the less-localized oscillations that can be produced by linear combinations of the basis functions. While these instabilities have a different appearance than those in the high-dimensional model, the root cause is the same and the effect on the quality of the approximation is similarly destructive.

Consequently, one must also apply a stabilization such as SUPG or SOLD into the ROM model. Because these stabilizations are imposed on the weak problem the implementation can be done in a straightforward manner. For example, adding SUPG stabilization to (5) we have

$$\left(u^n_{\mathrm{ROM}}, \psi_i\right) + \Delta t G(u^n_{\mathrm{ROM}}, \psi_i) + \Delta t \sum_{K\in\mathbb{T}_h} \left(\mathcal{R}(u^n_{\mathrm{ROM}}), \tau_K \mathbf{v}\cdot\nabla\psi_i\right)_K = \left(u^{n-1}_{\mathrm{ROM}}, \psi_i\right) \tag{13}$$

for $i = 1, \ldots, d$ and where $\mathcal{R}(u^n_{\mathrm{ROM}})$ is the residual for (2).

3.5. Numerical examples of ROM with stabilization

Here we demonstrate the effectiveness of using ROM with SUPG and SOLD stabilizations for two problems which are advection-dominated. The first is the advection problem of moving a cylinder where its direction varies between prescribed directions; consequently, the parameters for the problem are the two components of the cylinder's direction as well as time. The second problem is the ADR equation (2) in two spatial dimensions with uniform flow and a nonlinear reaction term. In addition to time, we assume that there are three coefficients to be used as parameters to sample.

The computational grid for these experiments is a partition of the rectangular domain into a set of regular triangles, constructed by forming a simple rectilinear mesh and subdividing each rectangular cell into two triangular cells by adding a partition from the lower left-hand corner of the rectangular cell to the upper right-hand corner of the cell. In the finite element model, piecewise-continuous polynomials are a common choice of basis, and the order of the polynomials will influence the spatial order of convergence. When a finite element model is used as the high-dimensional model to produce snapshots, the resulting reduced model will have basis functions of the same type (i.e., from the same class of continuous piecewise polynomials). In our numerical examples, we use basis functions that are continuous piecewise-quadratic polynomials defined over the partition. The time derivative terms are discretized using a first or second-order backward-difference method and the Newton–Raphson method is used to solve the discretized nonlinear equations in the second example.

3.5.1. Advection of a cylinder

For this pure advection example we define the spatial domain $\Omega = (0, 1) \times (0, 1)$ with boundary Γ and set

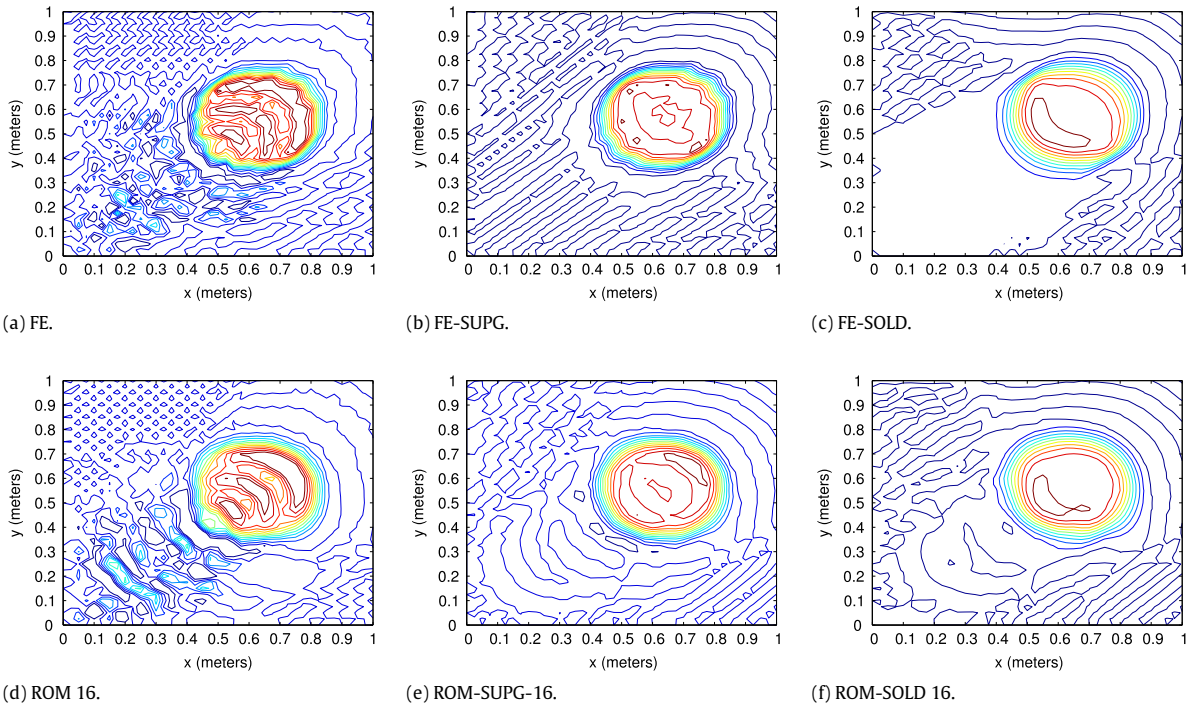
$$\begin{aligned} \frac{\partial u}{\partial t} &= -\mathbf{v}\cdot\nabla u \quad \forall(\mathbf{x}, t) \in \Omega \times (0, T] \\ u(\mathbf{x}, 0) &= \begin{cases} 1 \text{ for } |(x, y)^T - (0.25, 0.25)^T| \le 0.2 \\ 0 \text{ otherwise.} \end{cases} \\ u(\mathbf{x}, t) &= 0 \quad \mathbf{x} \in \Gamma. \end{aligned} \tag{14}$$

We consider a scenario in which the speed of advection has unit magnitude but may manifest in any direction chosen from a 15° range of vectors; specifically we take $\mathbf{v}$ to be a point on the unit circle corresponding to $\pi/6 \le \theta \le \pi/4$. To generate snapshots in time for a particular choice of $\mathbf{v}$ we solve the stabilized problem from $t = 0$ to $t = 0.5$, using $\Delta x = \Delta y = 0.05$, $\Delta t = 0.001$. This computational grid is too coarse to obtain a stable solution using standard finite elements; i.e., without using stabilization.

To obtain a reduced order model that is capable of solving any problem with a velocity in this range, we sample the region $\pi/6 \le \theta \le \pi/4$ according to a uniform distribution to obtain 20 sample values of the velocity vector, $\mathbf{v}$. We use a stabilized finite element simulation to produce snapshots, sampling the time domain at intervals of $\Delta t = 0.01$ for a total of 1000 snapshots. For this problem, there are no inhomogeneous Dirichlet boundary conditions to consider.

To test the ROM solution we choose advective directions different from the ones used to generate the snapshots. In Fig. 1 we compare the full finite element solutions at $t = 0.5$ using (i) no stabilization, (ii) SUPG, (iii) SOLD with the corresponding ROM solution for a representative choice of $\mathbf{v}$ not used in the snapshot generation, specifically $\mathbf{v} = (0.774603071265697, 0.632447691106348)^T$. As expected, in the ROM method an increase in the number of basis functions causes the resulting approximation to more closely resemble the corresponding finite element approximation. For brevity we only present results for sixteen basis vectors.

In the case of the standard finite element/ROM approximations, in which no stabilization is added, the approximations contain too many oscillations to be of practical use. However, the results are included here to demonstrate that even in the case of the unstable approximation, ROM is able to recover the features of the corresponding finite element approximation. The use of stabilization comes with inherent trade-offs: a smooth solution often comes at the expense of the sharpness



(a) FE. (b) FE-SUPG. (c) FE-SOLD.

(d) ROM 16. (e) ROM-SUPG-16. (f) ROM-SOLD 16.

Fig. 1. Comparison of results for cylinder advection using stabilized finite element and ROM approximations with 16 basis vectors (contours: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9).

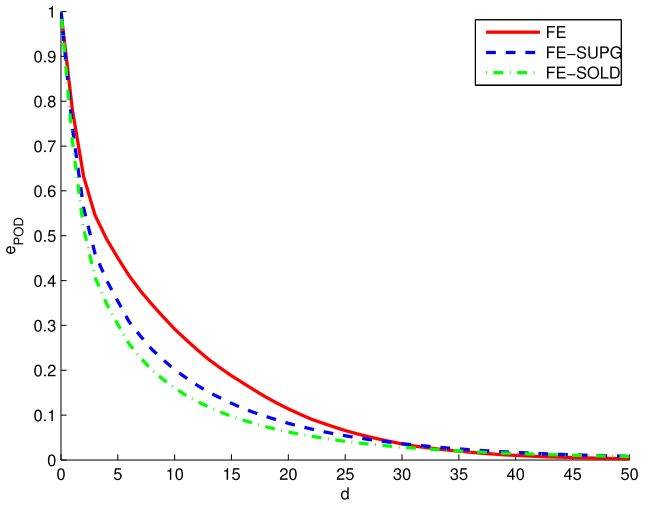


Fig. 2. Relative error, e_{POD}, for snapshots of cylinder advection sampled over a trajectory range of 15°.

of steep gradients in the solution. In this particular example, while the SOLD models reduce spurious oscillations in the approximations compared to the SUPG models, not all oscillations are removed from the cylinder. This may be further diminished by varying the SOLD parameter, C, but at the cost of further smearing of the steep gradients in the solution. Thus, the method and parameter presents a trade-off in choosing between removal of spurious oscillations and preservation of steep gradients, which must be balanced. However, no matter which level of stabilization is desired, ROM can be used to recover the preferred approximation in a reduced-basis representation.

One can also compute the relative error e_{POD} defined in (1) for these calculations. From Fig. 2 we note that the stabilized snapshots seem to be producing more redundant information, presumably because they lack the majority of spurious oscillations contained in the non-stabilized snapshot set.

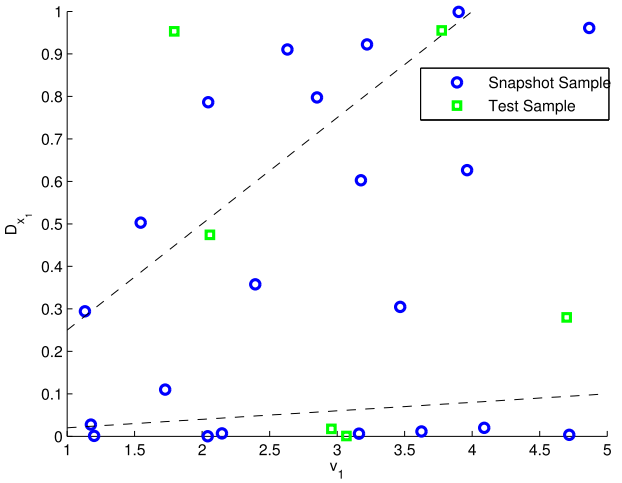


Fig. 3. Parameter space cross-section and sampled points.

3.5.2. A nonlinear ADR example

Let us now consider the application of stabilized ROM to a nonlinear scalar ADR equation in two spatial dimensions. Specifically we solve the prototype problem

$$\frac{\partial u}{\partial t} = \sum_{i=1}^{2} D_{x_i} \frac{\partial^2 u}{\partial {x_i}^2} - v_1 \frac{\partial u}{\partial x} - ku^2 \quad \forall (\mathbf{x}, t) \in \Omega \times (0, T) \tag{15}$$

where $\Omega = (0, 10) \times (0, 10)$ and where we assume uniform flow in the x_1 direction with scalar velocity v_1 such that the two-dimensional advection velocity vector is given by $\mathbf{v} = (v_1, 0)$; k is a reaction parameter. We set the homogeneous initial $u(\mathbf{x}, 0) = 0$ and the boundary conditions as

$$\frac{\partial u(x_1, 0, t)}{\partial x_2} = \frac{\partial u(x_1, 10, t)}{\partial x_2} = \frac{\partial u(10, x_2, t)}{\partial x_1} = 0 \tag{16}$$

$$u(0, x_2, t) = \begin{cases} 1 & \text{for } 0.4 \le x_2 \le 0.6 \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

For simplicity we assume $D_{x_2} = 0.1 D_{x_1}$ to reduce the number of parameters to three in the problem.

Suppose that we wish to be able to solve this equation many times for problems with parameter values that lie in the parameter space

$$v_1 \in [1, 5], \qquad D_{x_1} \in [0, 1], \quad \text{and} \quad k \in [5.0 \times 10^{-2}, 5.0 \times 10^{-1}]. \tag{18}$$

We partition this parameter space into three subsets characterized as non-advection-dominated ($Pe \le 2$), moderately advection-dominated ($2 < Pe < 25$) and strongly advection dominated ($Pe \ge 25$), where Pe denotes the problem Péclet number. This partition of the parameter space is illustrated in Fig. 3 where we use a two-dimensional cross section in which the three-dimensional parameter space has been collapsed onto the (v_1, D_{x_1})-plane (the reaction rate, k, does not affect the Péclet number). We use Latin hypercube sampling (LHS) to generate 7 quasi-random points in each of the three delineated zones, generating a total of 21 parameter samples. The locations of the 21 sample points in the parameter space can be seen in Fig. 3.

For each snapshot sample parameter set we run a finite element simulation from $t = 0$ to $t = 1$ on a uniform triangular grid with $\Delta x = \Delta y = 0.4$ using quadratic basis functions, solving the full nonlinear equations with a Newton–Raphson iteration. The time derivative is discretized using a first-order backward difference approximation with $\Delta t = 0.025$, and snapshots of the solution are written at increments of $\Delta t = 0.05$.

As before, we generate three sets of ROM basis vectors using POD; one set has snapshots generated from the unstabilized finite element approximation, another uses SUPG stabilization and the third uses SOLD stabilization. We use the same set of parameter samples, shown in Fig. 3, to generate snapshots for all three snapshot sets. Because the problem has an inhomogeneous Dirichlet boundary condition we must modify the snapshots to have homogeneous boundary data and use the procedure described in Section 2.3.

A second LHS sampling is used to choose six additional random points in the parameter space (two in each subset of the space); see Fig. 3. We use these points to test the reduced model by comparing the reduced order approximations to the finite element approximations at $t = 1$ for each of the three stabilization methods. Test cases 1 and 2 have parameter choices which make the problem strongly advection-dominated, test cases 3 and 4 are slightly advection-dominated and test cases 5 and 6 are diffusion-dominated.

Table 1
Relative L^2 error of reduced-basis solutions without operator splitting as compared to corresponding full finite element solutions.

Basis size	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
	ROM					
8	0.1036	0.0693	0.0269	0.1539	0.0632	0.0956
16	0.0661	0.0332	0.0140	0.1197	0.0230	0.0219
32	0.0156	0.0151	0.0096	0.0825	0.0107	0.0110
	ROM-SUPG					
8	0.0799	0.0677	0.0199	0.0871	0.0653	0.1196
16	0.0171	0.0129	0.0156	0.1054	0.0602	0.0256
32	0.0090	0.0031	0.0079	0.0521	0.0109	0.0118
	ROM-SOLD					
8	0.1125	0.0753	0.0288	0.1108	0.0626	0.1298
16	0.0176	0.0329	0.0169	0.0622	0.0497	0.0217
32	0.0139	0.0177	0.0089	0.0718	0.0148	0.0119

Table 1 demonstrates the convergence of the reduced model approximations to the finite element approximation as the number of basis functions is increased for each of the three models (ROM, ROM-SUPG, ROM-SOLD). In all test cases except Test 4 the error reduces as the number of ROM basis vectors are increased. In addition, for Test 4, the error is higher than for the other test cases. This is because the value of the advection parameter is higher than all of the sampled points used to generate snapshots. Since the snapshots were taken from $t = 0$ to $t = 1$, this means that in Test 4, the contaminant plume advances beyond the plume data captured in the reduced basis. In this manner, Test 4 is an attempt to use ROM in an extrapolatory sense (in terms of the parameter space), rather than in an interpolatory sense. For all of the advection-dominated test cases, the ROM-SUPG approximations have the best agreement with their finite element counterparts.

Contour plots of the finite element and reduced order approximations (with 8 and 16 basis functions), using the three methods, for the first test case are presented in Fig. 4. This first test case comes from the strongly advection-dominated regime of the parameter space. In the more diffusive cases there is no apparent difference between the stabilized approximations and the standard finite element and ROM approximations. This is exactly what we desire; in cases where the approximation is stable, we do not wish for the stabilization scheme to alter the approximation in any way.

4. Stabilized ROM with operator splitting

In reactive transport models the reaction term can render the equation (or system of equations) nonlinear so that the computational load becomes more onerous. A popular technique for these types of problems is to use operator splitting to separate the computation of the transport and reactive components. In [12] the authors present results for operator splitting for the ADR equation. In particular, consider the ADR equation written as

$$\frac{\partial u}{\partial t} = \mathcal{F}_A(u) + \mathcal{F}_D(u) + \mathcal{F}_R(u), \quad \mathbf{x} \in \Omega,\ t > 0, \tag{19}$$

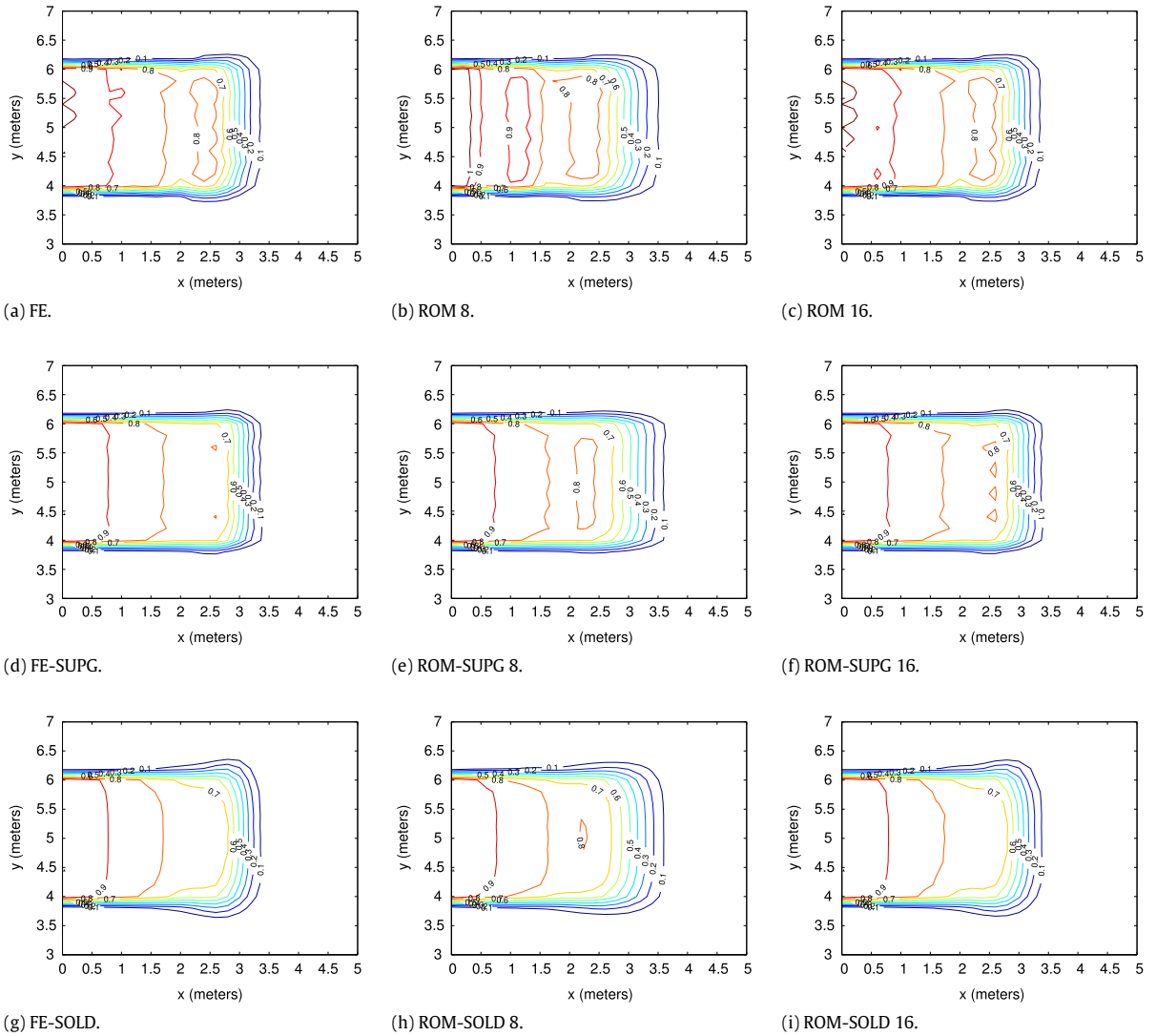
where $\mathcal{F}_A(u)$, $\mathcal{F}_D(u)$, and $\mathcal{F}_R(u)$ are the terms related to advection, diffusion, and reaction, respectively. The following is a first-order splitting method for (19) at time t^{n+1}:

$$\frac{\partial u^*}{\partial t} = \mathcal{F}_A(u), \quad t \in [t^n, t^{n+1}], \quad u^*(t^n) = u^n \tag{20a}$$

$$\frac{\partial u^{**}}{\partial t} = \mathcal{F}_D(u), \quad t \in [t^n, t^{n+1}], \quad u^{**}(t^n) = u^*(t^{n+1}) \tag{20b}$$

$$\frac{\partial u^{***}}{\partial t} = \mathcal{F}_R(u), \quad t \in [t^n, t^{n+1}], \quad u^{***}(t^n) = u^{**}(t^{n+1}), \tag{20c}$$

where $u^{n+1} = u^{***}(t^{n+1})$. To advance the system from time t^n to time t^{n+1}, one can solve the three phases (Eqs. (20a), (20b), and (20c)) sequentially. This splitting scheme is first-order in that we are substituting $u^{**}(t^n) = u^*(t^{n+1})$, and performing an analogous substitution for $u^{***}(t^n) = u^{**}(t^{n+1})$. This introduces an error due to the time-lag from solving the phases separately. It is possible to obtain a higher-order operator splitting scheme, such as a second-order scheme which approximates at half-intervals, $u^{**}(t^n) = u^*(t^{n+1/2})$, $u^{***}(t^n) = u^{**}(t^{n+1/2})$. One feature of this approach is that the order of computation of these different operators may be changed; that is, it is not absolute that at each time level, we must solve first for the effects of advection, second for diffusion, and solve the reactive phase last. Also, we are free to choose different methods to approximate each phase. Of course, we are also free to combine both the advection and diffusion phases into a single transport phase.



(a) FE. (b) ROM 8. (c) ROM 16.

(d) FE-SUPG. (e) ROM-SUPG 8. (f) ROM-SUPG 16.

(g) FE-SOLD. (h) ROM-SOLD 8. (i) ROM-SOLD 16.

Fig. 4. Approximations for strongly advection-dominated test case 1, solving full nonlinear equations (contours at 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9).

Since the reaction phase, Eq. (20c), is not dependent on the spatial variables, we solve this phase as an initial value problem. Because reactions occur locally (not spatially dependent), we may assume that reactions occurring within each grid cell (or at each grid node) are independent of the reactions occurring at other locations in the spatial domain. This allows for the reaction phase to be computed as an initial value problem at each grid node. Furthermore, the nonlinear terms in the unstabilized system are all contained within the reactive phase given in (20c). The transport phases in (20a) and (20b) describing advection and diffusion are linear and uncoupled. We still must solve (20a) and (20b) globally, but this requires solving a set of linear uncoupled PDEs, rather than solving a nonlinear PDE.

In this section we incorporate the SUPG and SOLD stabilization techniques into the ADR equation in an operator splitting setting so that the snapshots for ROM can be computed more efficiently. Then we look at implementing the stabilized reduced order model using operator splitting. Lastly we present some numerical results and timings for the example which was solved in Section 3.5.2 without operator splitting.

4.1. Stabilized FEM with operator splitting

We now want to incorporate the stabilization methods SUPG and SOLD into solving the ADR equation with operator splitting. The choices for splitting that are considered are (i) transport and then reaction which we denote AD–R and (ii) advection, then diffusion, then reactions which we denote A–D–R. Of course we can also split in different orders but our results indicate these are the best for the examples performed here.

Because we are always splitting the reaction phase, we consider the transport equation $(u_t, v) + G_1(u, v) + G_2(u, v) = 0$ where

$$G_1(u, v) = \int_\Omega \nabla \cdot (\mathbf{v}u)v \, d\Omega \quad \text{and} \quad G_2(u, v) = \sum_{i=1}^{2} D_{x_i} \int_\Omega \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_i} \, d\Omega. \tag{21}$$

For the AD–R splitting we have the system

$$(u_t^*, v) + G_1(u^*, v) + G_2(u^*, v) = 0, \quad \text{where } u^*(t^n) = u^n \tag{22a}$$

$$u_t^{**} = R(u^{**}) \quad \text{where } u^{**}(t^n) = u^*(t^{n+1}). \tag{22b}$$

Now to add the SUPG stabilization to (22) we incorporate the term

$$\sum_{K \in \mathbb{T}_h} \tau_K \left(\left(u_t^* - \sum_{i=1}^{2} D_{x_i} \frac{\partial^2 u^*}{\partial x_i^2} - \nabla \cdot (\mathbf{v}u^*) \right), W(v) \right)_K \tag{23}$$

in (22a). To add SOLD stabilization we also add to (22a) the term (11) where u is replaced by u^*. However, this adds a nonlinearity to the linear transport equation so typically one lags the dependence on the unknown in the calculation of $\widetilde{D}_K$ defined in (12).

If we split each phase, i.e., use A–D–R splitting, then there is some ambiguity about which terms to place in the residual in each phase. Suppose first that we split the transport equation into three phases and then add the stabilization. In this case there is no need to add stabilization to the diffusion phase so we have the advection and diffusion phases given by

$$(u_t^*, v) + G_1(u, v) + \sum_{K \in \mathbb{T}_h} \tau_K \left(\left(u_t^* - \nabla \cdot (\mathbf{v}u^*) \right), W(v) \right)_K = 0 \tag{24a}$$

$$(u_t^{**}, v) + G_2(u^{**}, v) = 0 \tag{24b}$$

with the initial conditions at each time step the same as in (22).

Another approach is to add the stabilization to the transport equation before discretization. For SUPG stabilization we have

$$(u_t, v) + G_1(u, v) + G_2(u, v) + \sum_{K \in \mathbb{T}_h} \tau_K \left(\mathcal{R}(u), W(v) \right)_K = 0$$

where $\mathcal{R}(u)$ is the residual of the transport equation, i.e., (2) with $R = 0$. Then we have the two phases

$$(u_t^*, v) + G_1(u^*, v) + \sum_{K \in \mathbb{T}_h} \tau_K \left(\mathcal{R}_1(u^*), W(v) \right)_K = 0 \tag{25a}$$

$$(u_t^{**}, v) + G_2(u^{**}, v) + \sum_{K \in \mathbb{T}_h} \tau_K \left(\mathcal{R}_2(u^{**}), W(v) \right)_K = 0 \tag{25b}$$

with the same initial conditions as in (22). We have two obvious choices for the residuals:

$$\text{I} : \mathcal{R}_1(u^*) = \frac{\partial u^*}{\partial t} - \nabla \cdot (\mathbf{v}u^*), \qquad \mathcal{R}_2(u^{**}) = \frac{\partial u^{**}}{\partial t} - \sum_{i=1}^{2} D_{x_i} \frac{\partial^2 u^{**}}{\partial {x_i}^2} \tag{26a}$$

$$\text{II} : \mathcal{R}_1(u^*) = \frac{\partial u^*}{\partial t} - \nabla \cdot (\mathbf{v}u^*) - \sum_{i=1}^{2} D_{x_i} \frac{\partial^2 u^*}{\partial {x_i}^2}, \qquad \mathcal{R}_2(u^{**}) = \frac{\partial u^{**}}{\partial t}. \tag{26b}$$

In our tests the stabilized ROM results using either of the two choices in (26) provided results that were more consistent with the full stabilized finite element results than the choice given in (24) where the stabilization is only applied to the advection phase. For the results presented herein the stabilization terms were incorporated using the first choice (26a) because these were slightly better than (26b) on the limited number of examples we investigated.

In the A–D–R splitting we found that the SOLD stabilization can be added to either the advection Eq. (25a) or to the diffusion equation (25b); the results were almost identical with either choice. As in the AD–R splitting we add the term (11) where u is replaced by u^*.

4.2. Numerical results for stabilized ROM with operator splitting

Because the Galerkin formulation of the ROM equations mimics the formulation of the full equations, we simply pose the fully discrete, stabilized weak finite element problem over the reduced space. When using operator splitting to calculate

Table 2
Relative L^2 error of ROM AD–R models as compared to corresponding finite element AD–R models.

Basis size	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
	ROM AD–R					
8	0.0919	0.0598	0.0283	0.1338	0.0589	0.0507
16	0.0403	0.0229	0.0081	0.1032	0.0179	0.0177
32	0.0083	0.0081	0.0038	0.0544	0.0087	0.0069
	ROM-SUPG AD–R					
8	0.0376	0.0432	0.0174	0.1329	0.0623	0.0293
16	0.0082	0.0048	0.0091	0.058	0.0126	0.0142
32	0.0035	0.0021	0.0033	0.0264	0.0043	0.0061
	ROM-SOLD AD–R					
8	0.0302	0.0374	0.0179	0.1399	0.0633	0.0309
16	0.0712	0.0579	0.0086	0.0732	0.0131	0.0159
32	0.0512	0.0433	0.0043	0.0266	0.0044	0.0065

Table 3
Relative L^2 error of ROM A–D–R models as compared to corresponding finite element A–D–R models.

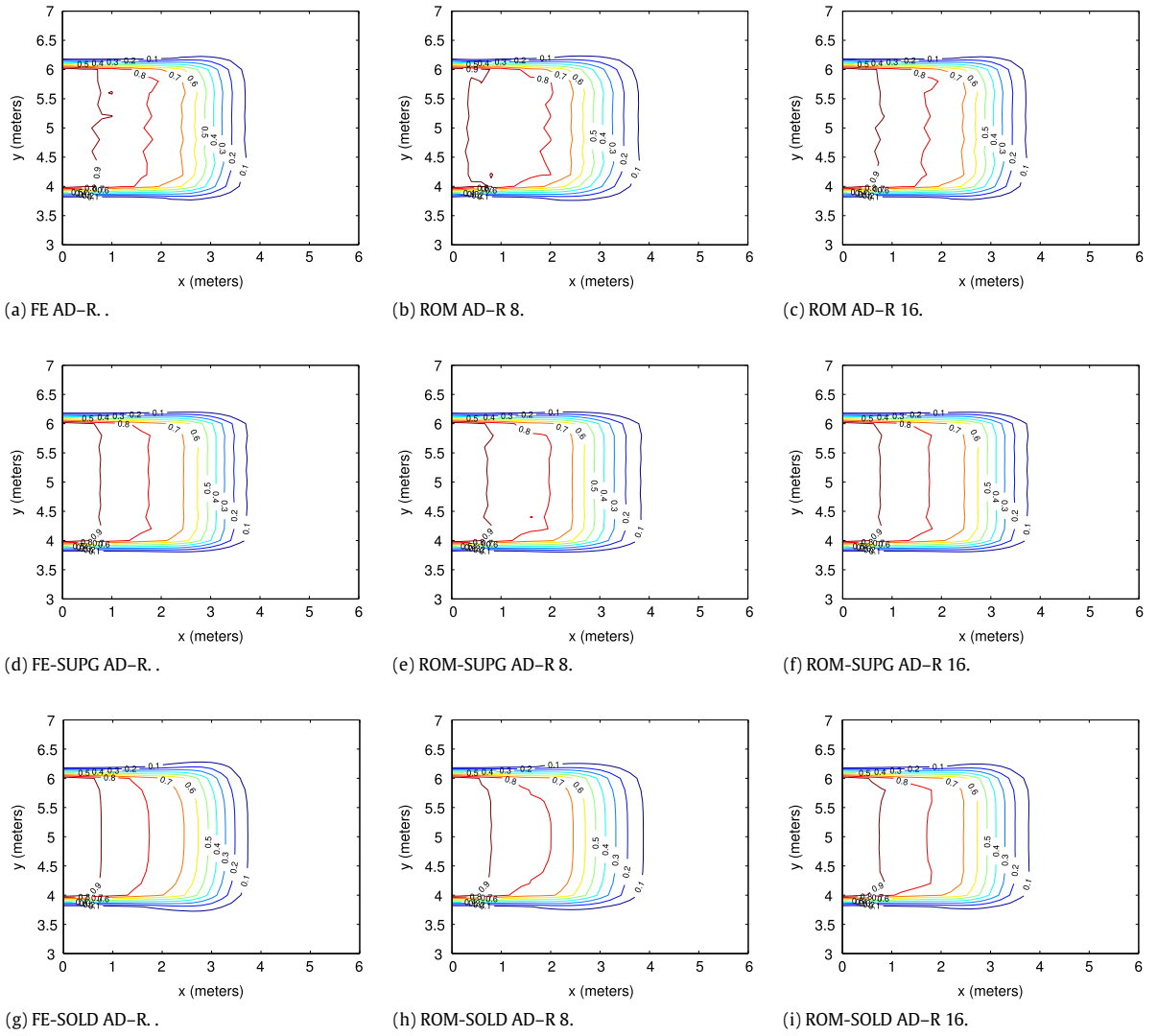
Basis size	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
	ROM A–D–R					
8	0.0886	0.0614	0.0256	0.1126	0.0627	0.0377
16	0.0366	0.0259	0.0082	0.1030	0.0174	0.0219
32	0.0087	0.0081	0.0061	0.0461	0.0100	0.0086
	ROM-SUPG A–D–R					
8	0.0348	0.0387	0.0136	0.1293	0.0635	0.0292
16	0.0106	0.0057	0.0079	0.0542	0.0145	0.0186
32	0.0042	0.0017	0.0074	0.0284	0.0098	0.0085
	ROM-SOLD A–D–R					
8	0.0280	0.0344	0.0167	0.1359	0.0654	0.0341
16	0.0652	0.0525	0.0091	0.0691	0.0158	0.0163
32	0.0485	0.0415	0.0081	0.0409	0.0123	0.0079

a snapshot at a particular time and parameter set, the solution at the end of the final phase (for us, the reaction phase) is used.

When ROM is applied in the operator splitting manner, even without stabilization, the solution at time t^n is computed at the end of the reaction phase and thus we have in hand a solution at each node. To start the transport phase we need an initial condition in the reduced basis space. When we perform ROM without operator splitting we only need to project the initial condition into the ROM space at the first step but using operator splitting requires it to be done at each time step. Consequently, a QR decomposition is too costly to perform; however, as discussed in Section 2.3 all we need is to evaluate the nodal solution at a quadrature point so this can easily be done.

We repeat the calculations for the nonlinear ADR equation given in Section 3.5.2 using the two types of operator splitting AD–R and A–D–R. The reaction phase is solved by a fourth order Runge–Kutta method. Table 2 demonstrates the relative L^2-error in stabilized ROM calculations compared with the full finite element calculations using either SUPG or SOLD stabilizations and the splitting of the equations into two phases, AD–R. The analogous table for the three phase splitting A–D–R is given in Table 3. In all cases, the ROM approximations approach the corresponding finite element approximations as we increase the number of basis functions used in the model and are comparable to the errors found in Table 1 where no operator splitting was used (see Fig. 5).

We now want to compare the computational costs for the ROM and full finite element calculations using operator splitting and stabilization. The computation of the POD basis (including the generation of snapshots and data compression) is performed only once for each experiment, and then the POD basis may be used to generate any number of reduced order realizations of the forward model. For this reason, the construction of the POD basis is considered a pre-processing step (which can be performed offline), and we consider the computational time for the reduced model to be only the time required to compute the approximation using the POD basis which was computed offline. These simulations were run on a machine with an Intel Core2Duo E6750 processor running at 2.66 GHz. In the example considered here, the diffusion and advection coefficients are constants so the ROM coefficient matrices can be calculated and factored outside of the time step loop. Thus only the right hand side of the equations need to be calculated at each time step. In the full finite element calculations the linear systems are solved using GMRES (with no preconditioning) which results in a significant savings over using a banded



(a) FE AD–R. . (b) ROM AD–R 8. (c) ROM AD–R 16.

(d) FE-SUPG AD–R. . (e) ROM-SUPG AD–R 8. (f) ROM-SUPG AD–R 16.

(g) FE-SOLD AD–R. . (h) ROM-SOLD AD–R 8. (i) ROM-SOLD AD–R 16.

Fig. 5. Approximations for Test 1, using transport–reaction operator splitting (AD–R) (contours at 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9).

Table 4
Computational time (in seconds) for solving nonlinear ADR example using AD–R operator splitting.

	FE	ROM 4	ROM 8	ROM 16	ROM 32
No stabilization	11.4	2.9	3.3	4.4	6.9
SUPG	12.2	3.4	4.0	5.4	7.6

solver. Table 4 gives the computational times for the AD–R splitting using the full finite element equations and ROM for three choices of the dimension of the ROM space. As can be seen from the timings using sixteen basis vectors results in a savings of more than two compared with the full finite element method using both the stabilized and unstabilized results. Using only eight basis vectors in the ROM space resulted in a savings of more than three. In general, the corresponding timings for the A–D–R are almost twice those for the AD–R splitting because two linear systems are being solved for the transport phase.

Acknowledgment

The first and second authors were supported by the US Department of Energy Office of Science under grant DE-SC0009324.

References

[1] A.K. Noor, J.M. Peters, Reduced basis technique for nonlinear analysis of structures, AIAA J. 18 (4) (1980) 455–462.
[2] J. Burkardt, M. Gunzburger, H.-C. Lee, Centroidal Voronoi tessellation-based reduced-order modeling of complex systems, SIAM J. Sci. Comput. 28 (2) (2006) 459–484.
[3] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA J. 40 (11) (2002) 2323–2330.
[4] Z. Luo, J. Zhu, R. Wang, I.M. Navon, Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical Pacific Ocean reduced gravity model, Comput. Methods Appl. Mech. Engrg. 196 (41) (2007) 4184–4195.
[5] A.N. Brooks, T.J. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1) (1982) 199–259.
[6] P.B. Bochev, M.D. Gunzburger, J.N. Shadid, Stability of the SUPG finite element method for transient advection–diffusion problems, Comput. Methods Appl. Mech. Engrg. 193 (23) (2004) 2301–2323.
[7] L.P. Franca, S.L. Frey, T.J. Hughes, Stabilized finite element methods: I. Application to the advective-diffusive model, Comput. Methods Appl. Mech. Engrg. 95 (2) (1992) 253–276.
[8] L.P. Franca, G. Hauke, A. Masud, Revisiting stabilized finite element methods for the advective–diffusive equation, Comput. Methods Appl. Mech. Engrg. 195 (13) (2006) 1560–1572.
[9] V. John, P. Knobloch, On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations: Part I–A review, Comput. Methods Appl. Mech. Engrg. 196 (17) (2007) 2197–2215.
[10] B. Kragel, Streamline diffusion POD models in optimization, Ph.D., University of Trier.
[11] S. Giere, T. Iliescu, V. John, D. Wells, SUPG reduced order models for convection-dominated convection–diffusion–reaction equations, Comput. Methods Appl. Mech. Engrg. 289 (2015) 454–474.
[12] D.L. Ropp, J.N. Shadid, Stability of operator splitting methods for systems with indefinite operators: Advection–diffusion–reaction systems, J. Comput. Phys. 228 (9) (2009) 3508–3516.
[13] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Rev. 41 (4) (1999) 637–676.
[14] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Methods Appl. Mech. Engrg. 196 (4) (2007) 1030–1047.
[15] J.P. Boris, D.L. Book, Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works, J. Comput. Phys. 11 (1) (1973) 38–69.
[16] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.
[17] D. Kuzmin, S. Turek, Flux correction tools for finite elements, J. Comput. Phys. 175 (2) (2002) 525–558.
[18] P. Knobloch, On the definition of the SUPG parameter, Electron. Trans. Numer. Anal. 32 (2008) 76–89.
[19] R. Codina, A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection–diffusion equation, Comput. Methods Appl. Mech. Engrg. 110 (3) (1993) 325–342.