Efficient Reduced-Basis Approximation of Scalar Nonlinear Time-Dependent Convection-Diffusion Problems, and Extension to Compressible Flow Problems

by

Han Men

Submitted to the School of Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Computation for Design and Optimization
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2006

© Massachusetts Institute of Technology 2006. All rights reserved.
Efficient Reduced-Basis Approximation of Scalar Nonlinear Time-Dependent Convection-Diffusion Problems, and Extension to Compressible Flow Problems

by

Han Men

Submitted to the School of Engineering on September, 2006, in partial fulfillment of the requirements for the degree of Master of Science in Computation for Design and Optimization

Abstract

In this thesis, the reduced-basis method is applied to nonlinear time-dependent convection-diffusion parameterized partial differential equations (PDEs). A proper orthogonal decomposition (POD) procedure is used for the construction of reduced-basis approximation for the field variables. In the presence of highly nonlinear terms, conventional reduced-basis would be inefficient and no longer superior to classical numerical approaches using advanced iterative techniques. To recover the computational advantage of the reduced-basis approach, an empirical interpolation approximation method is employed to define the coefficient-function approximation of the nonlinear terms. Next, the coefficient-function approximation is incorporated into the reduced-basis method to obtain a reduced-order model of nonlinear time-dependent parameterized convection-diffusion PDEs. Two formulations for the reduced-order models are proposed, which construct the reduced-basis space for the nonlinear functions and residual vector respectively. Finally, an offline-online procedure for rapid and inexpensive evaluation of the reduced-order model solutions and outputs, as well as associated asymptotic a posteriori error estimators are developed. The operation count for the online stage depends only on the dimension of our reduced-basis approximation space and the dimension of our coefficient-function approximation space. The extension of the reduced-order model to a system of equations is also explored.

Thesis Supervisor: Jaime Peraire
Title: Professor of Aeronautics and Astronautics
Acknowledgments

I would first like to thank my thesis advisor Professor Jaime Peraire, for his support, encouragement and guidance during the course of my thesis work as well as my stay in MIT. I am deeply grateful for his insights, flexibility, humor and trust.

I would also like to express my deepest gratitude to Doctor Ngoc Cuong Nguyen, for his constant help, selfless teaching, invaluable suggestions and genuine encouragement throughout. I truly appreciate his patience, kindness, and care, which inspire and keep me striving for success.

I am also sincerely grateful to my colleagues, Alex Yong Kwang Tan, Revanth Reddy Garlapati, Ho Man Lui. Thanks for the many helpful and enlightening discussions on and off the topic.

Finally, above all I want to express my love and deepest appreciation to my beloved family, my parents, Chang Gui Men and Xiao Hong Huang. Your love, support and encouragement have been keeping me all the way till today and always. To them, I dedicate this thesis.
Contents

1 Introduction ............................................ 13
  1.1 Motivation ............................................. 13
  1.2 Problem Statement ..................................... 14
  1.3 Literature Review ...................................... 15
    1.3.1 Finite Element Methods ............................ 15
    1.3.2 Reduced-basis Methods ............................ 16
    1.3.3 Model Order Reduction ............................ 17
  1.4 Approach .............................................. 18

2 Runge-Kutta Discontinuous Galerkin Method ............... 21
  2.1 The DG Space Discretization ........................... 21
  2.2 The RK Time Discretization ........................... 24

3 Reduced-Basis Methodology ................................ 25
  3.1 POD Procedure ......................................... 25
  3.2 Galerkin Approximation ................................ 27
  3.3 Empirical Interpolation Procedure ....................... 28
  3.4 Reduced-Basis Formulation ............................. 29
    3.4.1 Formulation I: Function Approximation ............. 29
    3.4.2 Formulation II: Residual Approximation ............ 34
  3.5 A Posterior Error Estimator ........................... 36
  3.6 Numerical Examples .................................... 38
    3.6.1 One-Dimensional Viscous Burger’s Equation ........ 38
### 3.6.2 Two-Dimensional Buckley-Leverett Equation

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Problem Formulations</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>Fully Discrete Equations</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>Offline/Online Procedure</td>
<td>51</td>
</tr>
<tr>
<td>4.4</td>
<td>Numerical Examples</td>
<td>53</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Problem Description</td>
<td>53</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Numerical Results-Function Formulation</td>
<td>54</td>
</tr>
<tr>
<td>5.1</td>
<td>Summary</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>Future Work</td>
<td>58</td>
</tr>
</tbody>
</table>
List of Figures

3-1 Convergence of solution of the reduced-basis Galerkin approximation for One-Dimensional Burger’s equation .................................................. 39
3-2 Convergence of output of the reduced-basis Galerkin approximation for One-Dimensional Burger’s equation .................................................. 39
3-3 Convergence of solution of the reduced-basis empirical interpolation approximation for One-Dimensional Burger’s equation .................................................. 40
3-4 Convergence of output of the reduced-basis empirical interpolation approximation output for One-Dimensional Burger’s equation .................................................. 40
3-5 Contour plot of the DG solution for Two-Dimensional Buckley-Leverett equation at $T = 0.5$ .................................................. 42
3-6 Approximation accuracy of nonlinear terms for Two-Dimensional Buckley-Leverett equation .................................................. 43
3-7 Convergence of the reduced-basis approximation output for Two-Dimensional Buckley-Leverett equation .................................................. 44
3-8 Convergence of the reduced-basis approximation solution for Two-Dimensional Buckley-Leverett equation .................................................. 44
4-1 DG solution for One-Dimensional Euler equation at $T = 1.0$ .................................................. 54
List of Tables

3.1 Effectivites of Galerkin approximation for One-Dimensional viscous Burger’s equation 41
3.2 Effectivites of empirical interpolation approximation for One-Dimensional viscous
Burger’s equation ................................................................. 41
3.3 Online computational time for One-Dimensional viscous Burger’s equation .......... 42
3.4 Effectivites for Two-Dimensional Buckley-Leverett equation ............................ 45
3.5 Online computational times (normalized) for Two-Dimensional Buckley-Leverett
 equation ................................................................................ 45
4.1 Maximum relative errors of output and solutions for One-Dimensional Euler equation 54
4.2 Error estimators of output and solutions for One-Dimensional Euler equation ..... 55
4.3 Effectivites of output and solutions for One-Dimensional Euler equation ........... 55
4.4 Online computational times (normalized) for One-Dimensional Euler equation (per
timestep) .............................................................................. 55
Chapter 1

Introduction

1.1 Motivation

As modern engineering design and optimization problems become progressively sophisticated both in depth and breadth, the role of mathematical modeling and numerical simulation of these problems becomes increasingly important. Often, system behavior or component attributes are modeled using governing partial differential equations (PDEs) which are in line with the underlying physics principles. Typically, the quantities of primary concern can be the solutions to certain field variables, e.g., velocity, density, and pressure — or in addition, certain outputs which are defined as functionals of the field variables, e.g., energy, flowrate, and temperature. These outputs serve to identify a certain configuration of the system parameters, or inputs — e.g., geometry, loads, and material properties — and are therefore modeled as functions of corresponding inputs.

While the solutions and outputs of the PDEs are of great importance to us, the analytical solutions to these governing equations are generally not available due to the complex nature of the systems considered, numerical procedures are sought to this end. Virtually all classical numerical approaches — e.g., finite element method (FEM)/ finite difference method (FDM)/ finite volume method (FVM) etc. — are adequate to solve the most common PDEs to the desired engineering accuracy. However, there are still problems with these standard approaches. Since they consider very dense approximation subspaces of the PDEs, the computational time for a particular input-output query is rather long despite the continual advance of computer speeds and expansion of hardware capacities; typical engineering design and optimization problems require considerable
number of input-output evaluations in realtime, which are not feasible to be addressed under such response time. Consequently, it is inefficient to perform any system design and optimization, robust parameter estimation of properties and states, or control of missions and processes.

Hence, the objective of this thesis is to develop a numerical approach that permits efficient real-time evaluation of the solutions and outputs of the PDEs and produces certifiable good results as compared to the classical PDE solution ("truth") approximation.

1.2 Problem Statement

Convection and diffusion arise in various engineering applications such as turbomachinery, gas and thermal dynamics, laminar and turbulent flows, viscoelastic flows, shallow water transport, and transport of contaminant in porous media. Numerical methods for convection-diffusion problems are of considerable interest in computational fluid dynamics (CFD).

In this dissertation, we will address the parameterized non-linear time-dependent convection-diffusion PDE of the following type:

\[ \frac{\partial u}{\partial t} + \nabla \cdot f(u) - \nu \nabla^2 u = 0, \quad \text{in } \Omega \times (0, T], \tag{1.1} \]

with appropriate initial condition \( u(x; \nu, t = 0) = u_0(x) \) and boundary conditions, here \( \Omega \) is the closed physical domain in \( \mathbb{R}^d \) with Lipschitz boundary \( \partial \Omega \); and \( f(u) \equiv (f_1(u), \ldots, f_d(u)) \), where \( f_g(u) \) — a nonlinear function of the field variable \( u \) — is defined as the flux in the direction of \( g \)th coordinate. To generalize the parameter-dependence of the problem considered, the viscosity constant \( \nu \) is denoted by a general parameter \( \mu \), which varies within a prescribed parameter domain \( \mathcal{D} \subset \mathbb{R} \). The field variable of interest \( u(x; \mu, t) \) is thus a function of the spatial coordinate \( x \), the parameter \( \nu \), as well as the time \( t \). Moreover, we are also concerned with certain output of the PDE, defined as a functional of the field variable, hence, also a function of the parameter \( \mu \) and time, \( s(\mu, t) = \ell(u(\mu, t)) \).

In actual practice, due to infinite dimensionality, the exact solution of \( u(\mu, t) \) and \( s(\mu, t) \) are not obtainable. Instead, we calculate what is known as "truth" solution and output, which are the approximation of the "exact" solution and output. We henceforth introduce \( X \subset X^\epsilon \), a "truth" finite element approximation space of finite but very large dimension \( \mathcal{N} \). The finite element
approximation of the “exact” problem can then be restated as: given $\mu \in \mathcal{D}$, our “truth” finite element solution is now $u_h(\mu, t_k), t_k = k\Delta t, 0 \leq k \leq K \equiv T/\Delta t$, which resides in the $p$th-order finite element approximation space $X_h^p$ of very large dimension $N$ — $\Delta t$ is a constant timestep; thus our “truth” output becomes $s_h(\mu, t_k) = l(u_h(\mu, t_k))$. We shall assume that $X_h^p$ is sufficiently rich and $\Delta t$ is reasonably small such that $u_h(\mu, t_k)$ (respectively, $s_h(\mu, t_k)$) is sufficiently indistinguishable from $u(\mu, t)$ (respectively, $s(\mu, t)$) for all $\mu \in \mathcal{D}$ at the accuracy level of interest.

Typically, the required dimension $N$ to satisfy this assumption is very large, — even with the application of appropriate adaptive mesh generation/refinement schemes — and in fact too large to perform real-time design and optimization queries effectively. The reduced-basis (RB) approximation is therefore needed and shall be built upon the “truth” finite element approximation. The evaluation of the RB approximation should also be conducted based on the comparison with this “truth” approximation.

We shall explore both finite element methods, and reduced-basis methods, as well as several model order reduction techniques in the following sections.

1.3 Literature Review

1.3.1 Finite Element Methods

Finite element methods (FEM) are most frequently used for solving PDEs that govern many engineering problems whose exact solutions by analytical techniques are very hard or even impossible to find. FEM, as a numerical method, is hence used to obtain an approximation solution. The point of departure for the FEM is a weighted-integral statement of a differential equation, called the weak formulation.

In the FEM, we seek the approximate solution over a discrete spatial domain known as triangulation $T_h$ of a physical domain $\Omega : \overline{\Omega} = \bigcup_{T_h \in T_h} \overline{T_h}$, where $T_h^k, k = 1, \ldots, K$, are the elements; $x_i, i = 1, \ldots, N$ are the nodes, and the subscript $h$ denoting the diameter of the triangulation $T_h$ is the longest edges of all elements. We define a finite element “truth” approximation space $X_h^p \subset X^e$ as

$$X_h^p = \{ v \in X^e | v|_{T_h} \in P^p(T_h), \forall T_h \in T_h \}$$

where $P^p(T_h)$ is the finite element space of polynomials of order up to $p^h$ over element $T_h$. 

15
To obtain the discrete equation of the weak formulation, we express the field variable \( u(\mu) \in X \) in terms of the nodal basic functions \( \varphi_i \in X_h^p \), \( \varphi_i(x_j) = \delta_{ij} \) such that

\[
X_h^p = \text{span}\{\varphi_1, \ldots, \varphi_N\},
\]

(1.3)

\[
u(x) = \sum_{i=1}^{N} u_i(\mu) \varphi_i, \quad v \in X_h^p,
\]

(1.4)

here \( u_i(\mu), i = 1, \ldots, N, \) is the nodal value of \( u(\mu) \) at node \( x_i \).

The spatial discretization (Triangulation) technique described above is the general approach of FEM. Depending on the connectivity of the neighboring elements and its nodes, the projections of the governing equation from “exact” space to “truth” space can be categorized into continuous Galerkin (CG) and discontinuous Galerkin (DG) methods. A large amount of works [11, 12, 7, 31] has been devoted to the development of DG methods for linear and nonlinear convection-diffusion problems recently. DG methods are highly parallelizable and can easily handle complicated geometries and boundary conditions. They are also locally conservative, high-order accurate, and are well-suited to complicated geometries and boundary conditions [12]. All these attractive properties are the main reasons for our choice of the DG methods for the numerical solution of equation (1.1).

### 1.3.2 Reduced-basis Methods

The reduced-basis method was first introduced in the late 1970s in nonlinear structural analysis for single parameter problems [1, 29], and subsequently abstracted and analyzed [3, 6, 16, 34, 37] and extended [19, 20, 32] to a much larger class of parameterized PDEs, and further developed more [16, 32, 34] to include \textit{a priori} error analysis.

The reduced-basis approach is local in parameter space in both practice and theory. Later work [25, 26, 46, 28, 35, 42, 22, 13] differs from these earlier efforts in several important ways [21]: firstly, \textit{global} approximation spaces are developed; secondly, rigorous \textit{a posteriori} error estimators are introduced; thirdly, \textit{offline/online} computational decompositions are exploited [3]. These three ingredients enable us to \textit{reliably} decouple the generation and projection stages of reduced-basis approximation, thereby improving the computation cost by several orders of magnitude.

Progress has also been made in \textit{a posteriori} error estimation for reduced-basis approximations. In particular, \textit{a posteriori} error bounds have been successfully developed for (i) linear [25, 26, 46, 35,
and (ii) at most quadratically nonlinear [28, 42, 22] elliptic partial differential equations that are affine in the parameter. These enable the development of very efficient offline-online computational strategies relevant in the many-query and real-time contexts. The operation count for the online stage — in which, given a new parameter value, we calculate the reduced-basis solution and output — depends only on the parametric complexity of the problem and on the dimension of the reduced-basis space (typically much smaller), but is independent of the dimension of the underlying “truth” finite element approximation space (typically very large).

1.3.3 Model Order Reduction

Generally, the model order reduction (MOR) approaches can be categorized into three types: (i), techniques using Karhunen-Loeve expansion (or Proper Orthogonal Decomposition); (ii), algorithms based Krylov subspace methods; (iii), methods based on Hankel norm approximants and balanced truncations. The need for efficient simulation tools for dynamical (time-varying) systems arising in circuit simulation, structural dynamics and micro-electro-machanical systems is a driving force behind the development of MOR approaches. The basic idea applied by all of the approaches is a projection from high-dimensional state space to very low dimensional state space, which results in the reduced-order model of the original system. In the POD approach, probably the most popular model-order reduction technique, time is considered as the varying parameter, and “snapshots” of the field variable at different times are obtained from either a numerical or experimental procedure. The optimal approximation space is constructed by applying the singular value decomposition to these vectors, and keeping only the $N$ vectors corresponding to the largest singular values. Since the singular values are related to the “energy” of the system, only the modes preserving the most energy are preserved. The reduced-order model is then obtained by a Galerkin projection onto the space spanned by these vectors. POD has been successfully applied in many fields: turbulent flows [24], fluid structure-interaction [14], non-linear structural mechanics [30], and turbo-machinery flows [44].

A great deal of attention has also been devoted to Krylov subspace-based methods for efficient modeling, effective realization, and fast simulation. The basic idea of these methods is to approximate the transfer function of original systems by orthogonal basis functions and projecting original systems onto that subspace [18]. Owing to their low computational cost and robustness, the Krylov
subspace-based methods have been proven very attractive for producing reduced-order model of many large-scale linear systems and have been broadly used in engineering applications: structural dynamics [2], optimal control of fluid flows [23], circuit design [10, 43], and turbomachinery [45].

In balanced truncation approach, the Hankel Singular Values (HSV) of the controllability and observability gramians of the system are computed. The state-space dimensions with low HSV are truncated, leading to a reduced-order model. For high-dimensional systems, computation of the required gramians is very expensive; combining POD and balanced truncation can overcome this limitation.

A large number of model-order reduction techniques has also been developed in particular to treat nonlinear time-dependent problems [2, 9, 8, 27, 33, 36, 40, 13]. Linearization approaches [13], for example, usually suffer from a lack of efficient representation of the nonlinear terms, whereas polynomial approximation approaches [9, 33] usually exhibit a fast exponential growth of computational complexity with the degree of the nonlinear approximation order. These two methods are thus quite expensive and do not address strong nonlinearities efficiently; other approaches for highly nonlinear systems (such as piecewise-linearization) have also been proposed [36, 39] but also at the expense of high computational cost and little control over model accuracy.

It is noted that most model-order reduction techniques focus mainly on reduced-order modeling of dynamical systems in which time is considered the only "variable;" the development of reduced-order models for parametric applications is much less common [41, 15].

1.4 Approach

The goal of this thesis is the rapid and reliable computations of the solution and output for nonlinear time-dependent convection-diffusion PDEs. To achieve this goal, we pursue the reduced-basis method with appropriate model reduction techniques. In our problem, both time and viscosity in equation (1.1) are treated as parameters. For convenience below, we introduce time-discrete sampling set \( T = \{t_0, \ldots, t_k\} \), parameter-time space \( \hat{D} = D \times T \), and parameter-time variable \( \hat{\mu} \equiv (\mu, t_k) \in \hat{D} \). The foundation of the reduced basis method is the realization that the set of all solutions \( u_h(\hat{\mu}) \), as \( \hat{\mu} \) varies, resides in a finite and low dimensional solution manifold \( \mathcal{W}_h = \{u_h(\hat{\mu}) \mid \hat{\mu} \in \hat{D}\} \). Hence, \( u_h(\hat{\mu}) \) can be approximated very well by its projection \( u_N(\hat{\mu}) \) on a finite vector space of \( N \) vectors \( W_N^\mu = \text{span}\{\zeta_n \equiv u_h(\hat{\mu}_n) \mid 1 \leq n \leq N\} \), where \( u_h(\hat{\mu}_n) \), \( 1 \leq n \leq N \), are
the finite element (discontinuous Galerkin) solutions of equation (1.1) at $N$ selected points in the sample $S^N_k = \{ \tilde{\mu}_1^N = (\mu_1^N, t_{k_1}), \ldots, \tilde{\mu}_N^N = (\mu_N^N, t_{k_N}) \in \tilde{D} \}$. 

The reduced-basis approximation approach proposed here will first build a reduced-basis approximation for the field variable. Instead of the classical Lagrangian basis $\zeta_n \equiv u_h(\tilde{\mu}_n), 1 \leq n \leq N$, we shall use a POD basis, $\Phi^N = \text{span}\{ \phi_n, 1 \leq n \leq N \}$. In the framework of this thesis, we view POD as a “procedure” that generates a set of optimal basis functions rather than a “technique” that establishes reduced-order models.

Next, an empirical interpolation method is used to obtain inexpensive approximations for the nonlinear terms of the PDE, which then allows an efficient offline-online computational paradigm.

Two different reduced-basis formulations are then derived for the nonlinear time-dependent convection-diffusion equations. Asymptotic a posterior error estimators are proposed as a means to quantify the reduced-basis approximations.

The thesis begins in Chapter 2 with a description of Runge-Kutta discontinuous Galerkin (RKDG) numerical method for our nonlinear time-dependent convection diffusion PDE in equation (1.1). In Chapter 3, we present the ingredients of reduced-basis approximations, namely, the POD procedure, the Galerkin approximation, and the empirical interpolation procedure; followed by two reduced-basis formulations — function approximation and residual approximation. Extensions of the approach for the compressible Euler equations are finally discussed in Chapter 5. Numerical examples will also be presented in each chapter to access the accuracy and capability.
Chapter 2

Runge-Kutta Discontinuous Galerkin Method

In this chapter, we describe a Runge-Kutta discontinuous Galerkin method for obtaining solutions of nonlinear time-dependent convection-diffusion PDEs, upon which our reduced-basis approach is built and compared with. Two steps are involved in this method, namely, finite element space discretization by a discontinuous Galerkin approximation scheme, and a Runge-Kutta time integration scheme.

2.1 The DG Space Discretization

We first introduce a triangulation, $\mathcal{T}_h$, of the domain $\Omega$ consisting non-overlapping elements $T_h$, such that $\Omega = \bigcup_{T_h \in \mathcal{T}_h} T_h$; for a typical element $T_h$, the set of its boundary edges is denoted by $\mathcal{E}(T_h)$.

Before proceeding to derive the weak formulation for DG discretization, an auxiliary variable $\mathbf{q} = \nabla u$ is introduced. The problem in equation (1.1) is rewritten as follows

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) - \mu \nabla \cdot \mathbf{q} = 0, \quad \text{in } \Omega \times (0,T],$$

$$\mathbf{q} = \nabla u, \quad \text{in } \Omega \times (0,T].$$

(2.1)

Next, we define two spaces in which the DG solutions $(u_h, \mathbf{q}_h)$ to our finite element approxima-
tion reside,

\[ X_h^p = \{ v \in L^2(\Omega) \mid v|_{T_h} \in P^p(T_h), \ \forall T_h \in \mathcal{T}_h \} \]  

\[ Y_h^p = \{ r \in (L^2(\Omega))^d \mid r|_{T_h} \in P^p(T_h), \ \forall T_h \in \mathcal{T}_h, \ 1 \leq g \leq d \} \]  

where \( P^p(T_h) \) is the finite element space of polynomials of order up to \( p \) over element \( T_h \), and \( L^2(\Omega) \) is the Lebesgue space of square-integrable functions over the domain \( \Omega \subset \mathbb{R}^d \).

By multiplying the two equations in (2.1) with test functions \( v_h \in X_h^p \) and \( r_h \in Y_h^p \) respectively, and integrating by parts, we obtain the weak formulation of the problem: find \((u_h, q_h) \in X_h^p \times Y_h^p\) such that \( \forall T_h \in \mathcal{T}_h \),

\[
\int_{T_h} \frac{\partial u_h}{\partial t} v_h + \int_{\partial T_h} f(u_h) \cdot \nu v_h = \sum_{\gamma_h \in \partial T_h} \int_{\gamma_h} \hat{f}(u_h, n_{\gamma_h}) v_h \\
+ \mu \int_{T_h} q_h \cdot \nabla v_h - \mu \sum_{\gamma_h \in \partial T_h} \int_{\gamma_h} \hat{q}_h \cdot n_{\gamma_h} v_h = 0, \ v_h \in X_h^p, \\
\int_{T_h} q_h \cdot r_h + \int_{T_h} u_h \nabla \cdot r_h - \sum_{\gamma_h \in \partial T_h} \int_{\gamma_h} \hat{u}_h r_h \cdot n_{\gamma_h} = 0, \ r_h \in Y_h^p.
\]

where \( \hat{f}(u_h, n_{\gamma_h}) \), \( \hat{q}_h \) and \( \hat{u}_h \) are called numerical fluxes, and are approximations to the values of the respective functions on the boundaries \( \gamma_h \in \mathcal{E}(T_h) \) of the element \( T_h \). In DG methods, different schemes are used to evaluate these terms. For the numerical flux \( \hat{f}(u_h, n_{\gamma_h}) \), a simple local Lax-Friedrichs flux scheme can be used,

\[
\hat{f}(u_h, n_{\gamma_h}) = \frac{1}{2} \left( f(u_h^+) + f(u_h^-) \right) \cdot n_{\gamma_h} - C(u_h, n_{\gamma_h})(u_h^+ - u_h^-)
\]

and

\[
C(u_h, n_{\gamma_h}) = \frac{1}{2} \max \left\{ \left| \frac{\partial}{\partial u} f(u_{T_h}) \cdot n_{\gamma_h} \right|, \left| \frac{\partial}{\partial u} f(\tilde{u}_{T_h}) \cdot n_{\gamma_h} \right| \right\}
\]

where \( \tilde{u}_{T_h} \) and \( \tilde{u}_{T_h}' \) are the means of the numerical solution of elements \( T_h \) and \( T_h' \) sharing the same edge \( \gamma_h \).

For the numerical fluxes \( \hat{q}_h \) and \( \hat{u}_h \), the local flux formulae proposed in [12] are used, which gives rise to what is known as the local discontinuous Galerkin (LDG) method, and allows the
variable $q_h$ to be eliminated locally within each element. The local fluxes are defined as follows:

\[
\begin{align*}
\hat{u}_h &= u_h^- \\
\hat{q}_h &= q_h^+
\end{align*}
\]  

where $(u_h^+, q_h^+)$ (respectively, $(u_h^-, q_h^-)$) are the traces of $(u_h, q_h)$ on $\gamma_h$ from the interior (respectively, the exterior) of element $T_h$. That is, we are taking the exterior and interior limits for the numerical fluxes of $u_h$ and $q_h$. Of course, the other pair — interior limits of $u_h$ and exterior limits of $q_h$ — could have also been taken. To eliminate the auxiliary variable, we introduce a linear operator $L : X_h^p \to Y_h^p$ such that $\forall w_h \in X_h^p$, $L(w_h) \in Y_h^p$ is the solution to

\[
\int_{T_h} L(w_h) \cdot r_h = -\int_{T_h} w_h \nabla \cdot r_h + \sum_{\gamma_h \in \mathcal{E}(T_h)} \int_{\gamma_h} \hat{w}_h r_h \cdot n_{\gamma_h}, \quad \forall r_h \in Y_h^p.
\]  

It thus follows from (2.5) that $q_h = L(w_h)$, and $\hat{q}_h = \hat{L}(w_h)$. Hence the weak formulation from (2.5) can be rewritten in terms of the solution $u_h$ only:

\[
m \left( \frac{\partial u_h}{\partial t}, v_h \right) = \sum_{T_h \in \mathcal{T}_h} \int_{T_h} f(u_h) \cdot \nabla v_h - \sum_{\gamma_h \in \mathcal{E}(T_h)} \int_{\gamma_h} \hat{f}(u_h, n_{\gamma_h}) v_h - \mu u(u_h, v_h), \quad \forall v_h \in X_h^p.
\]  

$m(w, v)$ and $a(w, v)$ are two bilinear forms and are defined as

\[
m(w, v) = \sum_{T_h \in \mathcal{T}_h} \int_{T_h} w v,
\]  

\[
a(w, v) = \sum_{T_h \in \mathcal{T}_h} \int_{T_h} L(w) \cdot \nabla v - \sum_{\gamma_h \in \mathcal{E}(T_h)} \int_{\gamma_h} \hat{L}(w) \cdot n v.
\]  

Note the weak form in (2.11) is defined over the whole domain $\Omega$. It remains to address the time discretization.
2.2 The RK Time Discretization

To begin, we define the residual $r$ as

$$r(w_h, v_h; \mu) = \sum_{T_h \in \mathcal{T}_h} f(w_h) \cdot \nabla v_h - \sum_{\gamma_h \in \mathcal{E}(T_h)} \int_{\gamma_h} \hat{f}(w_h, n_{\gamma_h}) v_h - \mu a(w_h, v_h), \quad \forall v_h \in X_h^p. \quad (2.14)$$

The weak formulation (2.11) can thus be written as

$$m \left( \frac{\partial u_h}{\partial t}, v_h \right) = r(u_h(\mu, t), v_h; \mu), \quad \forall v_h \in X_h^p, \quad \text{in} \,(0, T]. \quad (2.15)$$

We consider the fourth-order accurate explicit Runge-Kutta scheme for time integration. We use a constant timestep $\Delta t$ and discretize the time $t_k = k\Delta t, \ 0 < t_k \leq T$. The problem can be restated as: find $u_h(\mu, t_k) \in X_h^p, 1 \leq k \leq K$ such that $\forall v_h \in X_h^p$,

$$m(u_h(\mu, t_k), v_h) = m(u_h(\mu, t_{k-1}), v_h) + \frac{\Delta t}{6} \left( r(u_h(\mu, t_{k-1}), v_h; \mu) ight. \left. + 2r(u_h(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{h,1}^{k-1}, v_h; \mu) ight) \right) \right), \quad (2.16)$$

where the intermediate solutions of the field variables — $u_{h,1}^{k-1}, u_{h,2}^{k-1}, u_{h,3}^{k-1}$ — are computed as,

$$m(u_{h,1}^{k-1}, v_h) = r(u_h(\mu, t_{k-1}), v_h; \mu), \quad \forall v_h \in X_h^p$$

$$m(u_{h,2}^{k-1}, v_h) = r(u_h(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{h,1}^{k-1}, v_h; \mu), \quad \forall v_h \in X_h^p \quad (2.17)$$

$$m(u_{h,3}^{k-1}, v_h) = r(u_h(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{h,2}^{k-1}, v_h; \mu), \quad \forall v_h \in X_h^p,$$

Note that, the initial value used for time integration is $u_h(\mu, t_0) = u_{h0}(x)$, which is the $L^2$ projection of $u_0(x)$ on the DG discretized space $X_h^p$.
Chapter 3

Reduced-Basis Methodology

In this chapter, we describe the reduced-basis methodology used to solve the nonlinear convection-diffusion problems defined in Section 1.2. The method first applies the proper orthogonal decomposition (POD) procedure to build a reduced-order basis for the field variables. For the parameter-dependent non-linear terms that arise in the PDE, it relies on the empirical interpolation procedure developed in [5, 17] to provide inexpensive coefficient-function approximation and allow efficient offline-online computational decompositions. We then discuss the reduced-basis formulation, which is based on the coefficient-function approximation for the non-linear functions. We will also briefly introduce another simpler formulation, which is based on the coefficient-function approximation for the residual vector. Asymptotic a posteriori error estimators are then developed to quantify the accuracy of the models.

3.1 POD Procedure

We apply the POD procedure to generate basis functions for the field variables \( \{ \phi_n(x), 1 \leq n \leq N \} \) from a set of linearly independent “snapshots” \( \{ U_j(x) \equiv u_h(x; \tilde{\mu}_j), \tilde{\mu}_j \in \tilde{\mathcal{D}}, 1 \leq j \leq P \} \); here \( u_h(x; \tilde{\mu}_j), 1 \leq j \leq P \), are discontinuous Galerkin solutions at different parameter-time values \( \tilde{\mu}_j \); recall that \( \tilde{\mu}_j \) is defined in section 1.4, as a parameter-time variable \( \tilde{\mu} \equiv (\mu, t_k) \in \tilde{\mathcal{D}} \). Often, a large number of snapshots \( P \) will be chosen to describe the behavior of the system as comprehensive as possible; hence, the associated computational cost can be expensive. Given the set of snapshots, a
two-point spatial correlation function can be defined as

\[ K(x, x') = \frac{1}{P} \sum_{j=1}^{P} U_j(x) U_j(x') , \tag{3.1} \]

which accepts the following spectral decomposition

\[ K(x, x') = \sum_{j=1}^{P} \lambda_j \phi_j(x) \phi_j(x') . \tag{3.2} \]

Here the set of basis functions \( \phi_j, 1 \leq j \leq P, \) are orthonormal (i.e., \( \langle \phi_i, \phi_j \rangle = \delta_{ij} \)) and ordered in such a manner that the associated eigenvalues

\[ \lambda_j = \frac{1}{P} \sum_{l=1}^{P} \phi_j(x), U_l(x) \tag{3.3} \]

satisfy \( \lambda_j \geq \lambda_{j+1} . \)

For a given \( N \leq P, \) the POD procedure determines \( \phi_n, 1 \leq n \leq N, \) so as to maximize the captured energy

\[ E_N = \sum_{n=1}^{N} \left( \frac{1}{P} \sum_{l=1}^{P} \phi_n(x), U_l(x) \right) = \sum_{n=1}^{N} \lambda_n . \tag{3.4} \]

The first few basis functions thus represent the main energy-containing structures in the snapshots, with their relative importance quantified by \( \lambda_k. \) Typically, the number of basis functions needed \( N(\ll P) \) is chosen as the smallest integer satisfying \( \sum_{n=1}^{N} \lambda_n / \sum_{j=1}^{P} \lambda_j \geq 0.99. \) It can be shown that maximizing \( E_N \) amounts to solve the eigenfunction equation

\[ (K(x, x'), \phi(x')) = \lambda \phi(x) \tag{3.5} \]

for the first \( N \) eigenfunctions.

The method of snapshots [40] expresses the empirical eigenfunctions \( \phi(x) \) as a linear combination of the snapshots

\[ \phi(x) = \sum_{l=1}^{P} a_l U_l(x) . \tag{3.6} \]
Inserting this representation and (3.1) into (3.5), we immediately obtain

\[ Ca = \lambda a, \]  

(3.7)

where \( C \) is given by \( C_{ij} = \frac{1}{\beta} (U_i(x), U_j(x)), 1 \leq i, j \leq K \). The eigenproblem (3.7) can then be solved for the eigenvalues and eigenvectors from which the POD basis functions \( \phi_n(x), 1 \leq n \leq N \) are constructed by appealing to (3.6).

We denote by \( \Phi_N^p \) the approximation space spanned by these basis functions, i.e., \( \Phi_N^p = \text{span}\{\phi_1(x), \ldots, \phi_N(x)\} \). A reduced-order model for the field variables might be derived through a Galerkin projection onto this approximation space as described below.

### 3.2 Galerkin Approximation

Galerkin approximation of the non-linear terms refers to applying a standard Galerkin projection method and evaluating the non-linear terms of the PDEs from the reduced-basis of the field variables already established.

Applying a Galerkin approximation to our equation (2.15), using the associated reduced-basis model would give: find \( u_N(\mu, t) \in \Phi_N^p \) such that

\[ m(u_N(\mu, t), v_N) = r(u_N(\mu, t), v_N; \mu), \quad \forall v_N \in \Phi_N^p, \text{ in } (0, T]. \]  

(3.8)

where for any \( w_N \in \Phi_N^p \), \( r(w_N, v_N; \mu) \) is defined similarly as in (2.14),

\[ r(w_N, v_N; \mu) = \sum_{T_h \in \mathcal{T}_h} \int_{T_h} f(w_N) \cdot \nabla v_N - \sum_{\gamma_h \in \mathcal{E}(T_h)} \int_{\gamma_h} \hat{f}(w_N, n_{\gamma_h}) v_N - \mu a(w_N, v_N), \quad \forall v_N \in \Phi_N^p. \]  

(3.9)

and the numerical flux function \( \hat{f} \) is defined in (2.6).

Observe that if the function \( f \) consists of low order polynomial functions, we could substitute \( w_N = \sum_{j=1}^N w_{N,j} \phi_j \) into \( f(w_N) = (f_1(w_N), \ldots, f_d(w_N)) \) and \( \hat{f}(w_N, n) \), and easily expand them as the summations of products of the coefficients \( w_{N,j} \) and basis functions \( \phi_j \) [42, 20]; the evaluation of the residual \( r(w_N, v_N; \mu) \) would then be inexpensive and readily implemented by an online-offline procedure.

Unfortunately, this strategy does not work for high-order polynomials or non-polynomial nonline
ities: \( r(w_N, v_N; \mu) \) can only be evaluated by explicitly constructing \( w_N = \sum_{j=1}^{N} w_N^j \phi_j, f(w_N), \) and \( \hat{f}(w_N, n) \) in the online stage; the operation count for the online stage will therefore scale as some power of \( N \) — the dimension of the underlying finite element approximation space. Due to this \( O(N) \) dependence, it is no longer ingenuous to interpret (3.8) as a reduced-order model since the resulting computational advantage relative to classical approaches using advanced iterative techniques may only be modest. Hence, the nonlinear problems raised here require a special treatment such that the incurred computational cost depends only on the dimension of reduced-basis approximation space, but not on \( N \). Towards this end, we develop a collateral reduced-basis expansion for the nonlinear terms using the empirical interpolation procedure [5, 17] reviewed below.

3.3 Empirical Interpolation Procedure

For a general parameter-dependent nonlinear function \( g(w_h(x); x; \tilde{\mu}) \) with \( w_h \in X^p_h \), the idea here is to construct an inexpensive approximation \( g_M^{wh}(x; \tilde{\mu}) \) to \( g(w_h(x); x; \tilde{\mu}) \) via an approximation space \( \Psi^\mu_M = \text{span}\{\psi_m(x), 1 \leq m \leq M\} \) and associated set of interpolation points \( T^\mu_M = \{z_1, \ldots, z_M\} \). In particular, \( g_M^{wh}(x; \tilde{\mu}) \in \Psi^\mu_M \) — which is known as the coefficient-function approximation [5, 17] to \( g(w_h(x); x; \tilde{\mu}) \) — is given by

\[
g_M^{wh}(x; \tilde{\mu}) = \sum_{m=1}^{M} \beta_m(\tilde{\mu})\psi_m(x),
\]

where the coefficients \( \beta_m(\tilde{\mu}), 1 \leq m \leq M \), satisfy

\[
\sum_{m=1}^{M} \psi_m(z_i) \beta_m(\tilde{\mu}) = g(w_h(z_i); z_i; \tilde{\mu}), \quad 1 \leq i \leq M.
\]

Of course, the quality of our approximation depends crucially on the basis functions and interpolation points. Here \( \Psi^\mu_M \) and \( T^\mu_M \) are determined so as to provide good approximation to \( g(w_h(x); x; \tilde{\mu}) \) for \( w_h \) close to the manifold \( W_h = \{u_h(\tilde{\mu}) \mid \tilde{\mu} \in \tilde{D}\} \). Our attention is thus directed to the manifold \( W_h \), not the entire function space \( X^p_h \). The construction of \( \Psi^\mu_M \) and \( T^\mu_M \) are as follows.

The approximation space, \( \Psi^\mu_M = \text{span}\{\psi_m(x), 1 \leq m \leq M\} \), is constructed upon the set of snapshots \( \{g(u_h(\tilde{\mu}_j); \tilde{\mu}_j) \in \tilde{D}, 1 \leq j \leq P\} \) by using the POD procedure described earlier; recall that \( u_h(\tilde{\mu}_j) \) is the discontinuous Galerkin solution at \( \tilde{\mu}_j \). Hence, although the coefficient-function approximation \( g_M^{wh}(x; \tilde{\mu}) \) is defined for general \( w_h \in X^p_h \), we expect good approximation
only for \( w_h \) close to the manifold \( \mathcal{W}_h \) on which \( \Psi_M \) is constructed.

Once the basis set \( \psi_m(x), 1 \leq m \leq M, \) are available, the set of interpolation points, \( T^q_M = \{ z_1, \ldots, z_M \} \), can then be constructed. We first set \( z_1 = \text{arg} \text{ ess sup}_{x \in \Omega} |\psi_1(x)|, B^1_1 = \psi_1(z_1). \) Then for \( m = 2, \ldots, M, \) we solve the linear system \( \sum_{j=1}^{m-1} \sigma_j^{m-1} \psi_j(z_i) = \psi_m(z_i), 1 \leq i \leq m - 1, \) and set \( r_m(x) = \psi_m(x) - \sum_{j=1}^{m-1} \sigma_j^{m-1} \psi_j(x), z_m = \text{arg} \text{ ess sup}_{x \in \Omega} |r_m(x)|, \) and \( B^m_{ij} = \psi_j(z_i), 1 \leq i, j \leq m. \) It can be shown that the matrix \( B^m_{ij} = \psi_j(z_i), 1 \leq i, j \leq M, \) constructed in such way is invertible [5, 17].

Theoretical and numerical aspects of the empirical interpolation have been analyzed in great detail in [5, 17]. Note however that the presented procedure slightly differs from the procedure outlined in [5, 17] in the choice of basis functions: rather than forming the basis with a greedy selection process as in [5, 17], we choose to use the POD procedure; and the orthonormalization of the basis set becomes unnecessary as our POD basis set is already orthonormal with respect to the \( L^2 \) inner product. We are now ready to incorporate this empirical interpolation into the Galerkin approximation for the efficient evaluation of the reduced-order model.

### 3.4 Reduced-Basis Formulation

#### 3.4.1 Formulation I: Function Approximation

**Fully Discrete Equations**

To begin, we first define the necessary approximation spaces and the associated interpolation points for nonlinear terms:

- field variable \( u, \Phi_N^q = \text{span}\{\phi_1(x), \ldots, \phi_N(x)\}; \)
- nonlinear functions \( f_g, 1 \leq g \leq d, \Phi_{Mg}^q = \text{span}\{\phi_{Mg}^1, \ldots, \phi_{Mg}^q\}, T_{Mg}^q = \{ z_1^g, \ldots, z_M^g \}; \)
- nonlinear numerical flux function \( \hat{f}, \Psi_M^q = \text{span}\{\hat{\psi}_1, \ldots, \hat{\psi}_M\}, \hat{T}_M^q = \{ \hat{z}_1, \ldots, \hat{z}_M \}. \)

For simplicity, we assume that \( M^g = \hat{M} \equiv M, 1 \leq g \leq d, \) throughout this section. Note that the coefficient-function approximations are defined on \( \mathcal{T}_h \) for the nonlinear function \( f_g(u_N), 1 \leq g \leq d, \) but on \( \mathcal{E}(\mathcal{T}_h) \) for the nonlinear numerical flux function \( \hat{f}. \)

Next, by replacing the nonlinear terms \( f_g(u_N(\mu, t)), 1 \leq g \leq d, \) and \( \hat{f}(u_N(\mu, t), u) \) with the coefficient-function approximations \( f^{uN,M}_{g,M}(x; \mu, t), 1 \leq g \leq d, \) and \( \hat{f}^{uN,M}_{M,M}(x; \mu, t), \) we obtain the
reduced-basis model: given \( \mu \in \mathcal{D} \), we are to evaluate
\[
s_{N,M}(\mu, t) = \ell(u_{N,M}(\mu, t)), \quad \text{in } (0, T],
\]
where \( u_{N,M} \in \Phi_N^\mu \) is solved by
\[
m(\dot{u}_{N,M}(\mu, t), v_N) = \tilde{r}(u_{N,M}(\mu, t), v_N; \mu), \quad \forall v_N \in \Phi_N^\mu, \quad \text{in } [0, T].
\]

For the reduced-basis model, the residual vector \( \tilde{r}(w_N, v_N; \mu) \) is defined as
\[
\tilde{r}(w_N, v_N; \mu) = \sum_{T_h \in \mathcal{T}} \sum_{g=1}^{d} \int_{T_h} f_{g,M}^{w_N}(x) \frac{\partial u_N}{\partial x_g} + \sum_{\gamma_h \in \mathcal{E}(T_h)} \int f_{M}^{w_N}(x)v_N - \mu a(w_N, v_N), \quad \forall v_N \in \Phi_N^\mu,
\]
where \( f_{g,M}^{w_N}, 1 \leq g \leq d \), and \( f_{M}^{w_N} \) are computed by
\[
f_{g,M}^{w_N}(x) = \sum_{m=1}^{M} \beta_m \psi_m(x), \quad \sum_{m=1}^{M} \psi_m(z_j^g) \beta_m = \tilde{f}_g(w_N(z_j^g), 1 \leq g \leq d, 1 \leq j \leq M
\]
\[
f_{M}^{w_N}(x) = \sum_{m=1}^{M} \hat{\beta}_m \hat{\psi}_m(x), \quad \sum_{m=1}^{M} \hat{\psi}_m(\hat{z}_j) \hat{\beta}_m = \hat{f}(w_N(\hat{z}_j), \hat{n}(\hat{z}_j)), \quad 1 \leq j \leq M.
\]

Again, using the fourth-order accurate explicit Runge-Kutta scheme, we integrate the reduced-basis system (3.13) in time and obtain \( u_{N,M}(\mu, t_k), k = 1, \ldots, K \), from
\[
m(u_{N,M}(\mu, t_k), v_N) = m(u_{N,M}(\mu, t_{k-1}), v_N) + \frac{\Delta t}{6} \left( \tilde{r}(u_{N,M}(\mu, t_{k-1}), v_N; \mu)
\right.
\]
\[
+ 2\tilde{r}(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{N,M,1}(t_{k-1}, v_N; \mu) + 2\tilde{r}(u_{N,M}(\mu, t_{k-1}) + \Delta t u_{N,M,2}(t_{k-1}, v_N; \mu)
\]
\[
+ \tilde{r}(u_{N,M}(\mu, t_{k-1}) + \Delta t u_{N,M,3}(t_{k-1}, v_N; \mu)) \right), \quad \forall v_N \in \Phi_N^\mu,
\]

30
where intermediate solutions of the field variables — \( u_{N,M,1}^{k-1}, u_{N,M,2}^{k-1}, \) and \( u_{N,M,3}^{k-1} \) — are solved from

\[
m(u_{N,M,1}^{k-1}, v_N) = \tilde{r}(u_{N,M}(\mu, t_{k-1}), v_N; \mu), \quad \forall v_N \in \Phi_N^U
\]
\[
m(u_{N,M,2}^{k-1}, v_N) = \tilde{r}(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{N,M,1}^{k-1}, v_N; \mu), \quad \forall v_N \in \Phi_N^U
\]
\[
m(u_{N,M,3}^{k-1}, v_N) = \tilde{r}(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{N,M,2}^{k-1}, v_N; \mu), \quad \forall v_N \in \Phi_N^U.
\]

The RB output at a discrete time instance will then be calculated as

\[
s_{N,M}(\mu, t_k) = \ell(u_{N,M}(\mu, t_k)), \quad 1 \leq k \leq K.
\]

In what follows, we develop a computational procedure which allows to solve the linear system in (3.16) and evaluate our RB output efficiently.

**Offline-Online Procedure**

We first need to express the field variable as a linear combination of the basis functions

\[
u_{N,M}(\mu, t_k) = \sum_{j=1}^{N} \alpha_j(\mu, t_k) \phi_j.
\]

Substituting this representation into (3.16) and using the same bases for \( v_N \), i.e. \( v_N = \phi_i, 1 \leq i \leq N \), we have the following linear system for the coefficients \( \alpha_j(\mu, t_k), 1 \leq j \leq N \),

\[
\sum_{j=1}^{N} m(\phi_j, \phi_i) \alpha_j(\mu, t_k) = \sum_{j=1}^{N} m(\phi_j, \phi_i) \alpha_j(\mu, t_{k-1}) + \frac{\Delta t}{6} \left( \tilde{r}(u_{N,M}(\mu, t_{k-1}), \phi_i; \mu) + 2 \tilde{r}(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{N,M,1}^{k-1}, \phi_i; \mu) + 2 \tilde{r}(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{N,M,2}^{k-1}, \phi_i; \mu) + \tilde{r}(u_{N,M}(\mu, t_{k-1}) + \Delta t u_{N,M,3}^{k-1}, \phi_i; \mu) \right), \quad i = 1, \ldots, N.
\]

The initial coefficient \( \alpha_j(\mu, t_0) \) needs to satisfy

\[
\alpha_j(\mu, t_0) = (\phi_j, u_h(\mu, t_0))
\]
Here \( u^{k-1}_{N,M,1} = \sum_{j=1}^{N} \alpha^{k-1}_{1,j} \phi_j \), \( u^{k-1}_{N,M,2} = \sum_{j=1}^{N} \alpha^{k-1}_{2,j} \phi_j \), and \( u^{k-1}_{N,M,3} = \sum_{j=1}^{N} \alpha^{k-1}_{3,j} \phi_j \), and the coefficients \( \alpha^{k-1}_{1,j}, \alpha^{k-1}_{2,j}, \alpha^{k-1}_{3,j}, 1 \leq j \leq N \), need to satisfy
\[
\begin{align*}
\sum_{j=1}^{N} m(\phi_j, \phi_i) \alpha^{k-1}_{1,j} &= \tilde{r}(u_{N,M}(\mu, t_{k-1}), \phi_i; \mu), \quad 1 \leq i \leq N \\
\sum_{j=1}^{N} m(\phi_j, \phi_i) \alpha^{k-1}_{2,j} &= \tilde{r}\left(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u^{k-1}_{N,M,1}, \phi_i; \mu\right), \quad 1 \leq i \leq N \\
\sum_{j=1}^{N} m(\phi_j, \phi_i) \alpha^{k-1}_{3,j} &= \tilde{r}\left(u_{N,M}(\mu, t_{k-1}) + \frac{\Delta t}{2} u^{k-1}_{N,M,2}, \phi_i; \mu\right), \quad 1 \leq i \leq N.
\end{align*}
\] (3.22)

For any \( w_N = \sum_{j=1}^{N} w_{N,j} \phi_j \) in \( \Psi^M_N \), the reduced-basis residual can now be evaluated as
\[
\tilde{r}(w_N, \phi_i; \mu) = \frac{d}{\gamma} \sum_{g=1}^{M} \beta^g m \sum_{\gamma_n \in \Omega_h} \int_{T_h} \psi^g_m(x) \frac{\partial \phi_i}{\partial x_g} &
- \sum_{m=1}^{M} \tilde{\beta}_m \sum_{\gamma_n \in \Omega_h} \int_{T_h} \hat{\psi}^g_m(x) \phi_i - \mu \sum_{j=1}^{N} w_{N,j} a(\phi_j, \phi_i),
\] (3.23)
where the coefficients \( \beta^g_m, \tilde{\beta}_m, 1 \leq f \leq d, 1 \leq g \leq M \), are calculated from
\[
\begin{align*}
\sum_{m=1}^{M} \psi^g_m(z_i^j) \beta^g_m &= f_g \left( \sum_{j=1}^{N} w_{N,j} \phi_j(z_i^j) \right), \quad 1 \leq g \leq d, 1 \leq i \leq M \\
\sum_{m=1}^{M} \hat{\psi}^g_m(z_i) \tilde{\beta}_m &= \hat{f} \left( \sum_{j=1}^{N} w_{N,j} \phi_j(z_i), \mathbf{n}(z_i) \right), \quad 1 \leq i \leq M.
\end{align*}
\] (3.24)

We have now successfully decomposed the residual \( \tilde{r}(w_N, \phi_i; \mu) \) into a summation of products of parameter-dependent coefficients and parameter-independent quantities, the approximation procedure thus admits an offline/online computation decomposition [26, 35, 4, 20, 28]. For practical implementation purposes, we rewrite the equations (3.20) – (3.24) in the form of vector-matrix products. Let's first introduce the following parameter-independent matrices and vectors:
\[
\begin{align*}
A_{ij}^1 &= m(\phi_j, \phi_i), \quad A_{ij}^2 = a(\phi_j, \phi_i), \quad L_j = \ell(\phi_j), \quad \alpha_{0j} = (u_{ho}, \phi_j), \quad 1 \leq i, j \leq N \\
B_{ij}^g &= \psi^g_m(z_i^j), \quad \hat{B}_{ij} = \hat{\psi}^g_m(z_i), \quad 1 \leq g \leq d, 1 \leq i, j \leq M \\
C_{ij}^g &= \sum_{\gamma_n \in \Omega_h} \int_{T_h} \psi^g_m(\phi_j) \frac{\partial \phi_i}{\partial x_g}, \quad \hat{C}_{ij} = \sum_{\gamma_n \in \Omega_h} \int_{T_h} \hat{\psi}^g_m(\phi_i), \quad 1 \leq g \leq d, 1 \leq i \leq N, 1 \leq j \leq M \\
D_{ij}^g &= \phi_j(z_i^j), \quad \hat{D}_{ij} = \phi_j(z_i), \quad \hat{h}_i = \mathbf{n}(z_i), \quad 1 \leq g \leq d, 1 \leq j \leq N, 1 \leq i \leq M.
\end{align*}
\]
It thus follows that the reduced-basis output can be calculated as

\[ s_{N,M}(\mu, t_k) = L^T \alpha(\mu, t_k), \quad k = 1, \ldots, K, \]  

(3.25)

where \( \alpha(\mu, t_k) \in \mathbb{R}^N \) is the new unknown of interest in our reduced-basis approximation model, instead of \( u(\mu, t_k) \in \mathbb{R}^N \), and is the solution of

\[
\alpha(\mu, t_k) = \frac{\Delta t}{6} \left( \tilde{R}(\alpha(\mu, t_{k-1})) + 2 \tilde{R}(\alpha(\mu, t_k) + \Delta t \alpha_{k-1}^1) 
+ 2 \tilde{R}(\alpha(\mu, t_k) + \frac{\Delta t}{2} \alpha_{k-1}^2) + \tilde{R}(\alpha(\mu, t_k) + \Delta t \alpha_{k-1}^3) \right), \quad k = 1, \ldots, K. 
\]  

(3.26)

From (3.22), the coefficients \( \alpha_{k-1}^1, \alpha_{k-1}^2, \) and \( \alpha_{k-1}^3 \) can be calculated from

\[
\begin{align*}
\alpha_{k-1}^1 &= \tilde{R}(\alpha(\mu, t_{k-1})) \\
\alpha_{k-1}^2 &= \tilde{R}(\alpha(\mu, t_{k-1}) + \frac{\Delta t}{2} \alpha_{k-1}^1) \\
\alpha_{k-1}^3 &= \tilde{R}(\alpha(\mu, t_{k-1}) + \frac{\Delta t}{2} \alpha_{k-1}^2),
\end{align*}
\]  

(3.27)

and from (3.23), for any \( \sigma \in \mathbb{R}^N \), the reduced-basis residual vector can be calculated as

\[
\tilde{R}(\sigma) = E^1 f_1 (D^1 \sigma) + E^2 f_2 (D^2 \sigma) + \tilde{E} \hat{f}(\hat{D}, \hat{n}) - \mu E \sigma,
\]  

(3.28)

where \( E^1 = (A^1)^{-1} C^1 (B^1)^{-1}, \) \( E^2 = (A^1)^{-1} C^2 (B^2)^{-1}, \) \( \tilde{E} = (A^1)^{-1} \hat{C} (\hat{B})^{-1}, \) and \( E = (A^1)^{-1} A^2. \)

Finally, the offline/online procedure for the efficient evaluation of \( s_{N,M}(\mu, t_k) \) is implemented as follows:

---

In the offline stage, (performed once)

- First, solve for the DG solutions of the field variable, which consist the snapshots set \( \{u_h(x; \bar{\mu}_p), \bar{\mu}_p \in \bar{D}, 1 \leq p \leq P\} \)

- Next, apply the POD procedure to construct the approximation space of the field variable \( \Phi^q_M, \) and the approximation space of the nonlinear terms \( \Psi^q_M, \) \( 1 \leq g \leq d. \)

- Then use the empirical interpolation procedure to obtain the sets of interpolation points \( T^q_M, \hat{T}_M, 1 \leq g \leq d, \) for the nonlinear terms.
Finally form and store the parameter-independent quantities $D^0, \hat{D}, \hat{E}, E^0, \hat{E}, \hat{n}$, and $\alpha_0, 1 \leq g \leq d$.

In the online stage (performed many times), we simply perform the sum (3.26) to obtain $\alpha(\mu, t_k) \in \mathbb{R}^N$, and evaluate $s_{N,M}(\mu, t_k)$ from (3.25).

With this decomposition, the cost of our reduced-basis approach is determined by the operation count for the online stage: computing $R(\sigma)$ takes $O(6MN + N^2)$ function evaluations; performing the sum in (3.26) requires four evaluations of $R(\sigma)$; moreover, the output evaluation in (3.25) takes $O(2N)$ operations. In summary, the total operation count is $O(4K(6MN + N^2))$. Not surprisingly, the complexity of online stage is independent of $N$. To expect significant savings relative to the DG approximation $s_h(\mu, t_k)$, it is required that $M, N \ll N$.

### 3.4.2 Formulation II: Residual Approximation

Another reduced-basis formulation, which is simpler in implementation, but less stable than the previous formulation, will be briefly introduced in this section. This second formulation aims to form a reduced-basis model via the coefficient-function approximation for the residual-vector directly, instead of for the nonlinear functions in the first formulation.

**Fully Discrete Equations**

To begin, we write the Galerkin approximation (3.8) in the following form

$$
\Phi^T M \frac{\partial u_N(\mu, t)}{\partial t} = \Phi^T R(u_N(\mu, t); \mu), \quad \text{in } (0, T].
$$

(3.29)

Here $\Phi = [\phi_1, \ldots, \phi_N]$ is the matrix of $N$ basis vectors; $M$ is the mass matrix and is calculated as $M_{ij} = m(\phi_j, \phi_i), 1 \leq i, j \leq N$; and the residual vector $R(w; \mu), \forall w \in \mathbb{R}^N$ is calculated as

$$
R(w; \mu) = \sum_{g=1}^{d} \sum_{T_h \subset T_g} \int_{T_h} f_g \left( \sum_{j=1}^{N} \varphi_j(x)w_j \right) \sum_{i=1}^{N} \frac{\partial \varphi_i(x)}{\partial x_g} v_i
$$

$$
- \sum_{\gamma_h \subset \Gamma} \int_{\gamma_h} f \left( \sum_{j=1}^{N} \varphi_j(x)w_j, u \right) \sum_{i=1}^{N} \varphi_i(x) v_i
$$

$$
- \mu a \left( \sum_{j=1}^{N} \varphi_j(x)w_j, \sum_{i=1}^{N} \varphi_i(x) v_i \right), \quad \forall v \in \mathbb{R}^N ,
$$

(3.30)
where \( \varphi_j(x), 1 \leq j \leq N \), are the finite element basis functions associated with the space \( X^n \).

Similarly as before, besides the approximation space for the field variable, we also need an approximation space for the nonlinear terms. In this case, it is the residual function \( R(w; \mu) \) in (3.30).

We denote the approximation space by \( \Psi^R_M = \text{span}\{\psi_1^R, \ldots, \psi_M^R\} \subseteq \mathbb{R}^{N \times N} \), and the associated set of interpolation points by \( T^R_M = \{z_1^R, \ldots, z_M^R\} \); moreover, we denote the nonlinear term approximation space evaluated at the interpolation points by a square matrix, \( B_{im} = \psi_m^R(z_i^R), 1 \leq i, m \leq M \).

It then follows that our coefficient-function approximation \( R^e_M(x; \mu, t) \) to \( R(w(\mu, t); \mu) \) for any \( w(\mu, t) \in \mathbb{R}^N \) is given by

\[
R^e_M(x; \mu, t) = \Psi \beta(\mu, t), \quad B \beta(x; \mu, t) = b^w(\mu, t),
\]

where \( b^w(\mu, t) \in \mathbb{R}^M \) with \( b^w_m(\mu, t), 1 \leq m \leq M \), are the values of \( R(w(\mu, t); \mu) \) evaluated at the \( z_m^R \). By writing \( u_{N,M}(\mu, t) = \Phi \alpha(\mu, t) \) and replacing \( R(u_N(\mu, t); \mu) \) of (3.29) with \( R_{N,M}^e(x; \mu, t) \) of (3.31), we can reach the reduced-order model: given \( \mu \in \mathcal{D} \), we are to compute the output

\[
s_{N,M}(\mu, t_k) = L^T \alpha(\mu, t_k), \quad 1 \leq k \leq K, \tag{3.32}
\]

where \( \alpha(\mu, t_k) \in \mathbb{R}^N \) is the solution of

\[
\alpha(\mu, t_k) = \alpha(\mu, t_{k-1}) + \frac{\Delta t}{6} E \left( b(u_{NM}(\mu, t_{k-1})) + 2b(u_{NM}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{NM,1}^{k-1}) 
+ 2b(u_{NM}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{NM,2}^{k-1}) + b(u_{NM}(\mu, t_{k-1}) + \Delta t u_{NM,3}^{k-1}) \right) \tag{3.33}
\]

here \( u_{NM,1}^{k-1} = \Phi \alpha_1^{k-1}, u_{NM,2}^{k-1} = \Phi \alpha_2^{k-1}, \) and \( u_{NM,3}^{k-1} = \Phi \alpha_3^{k-1}, \) and the coefficients \( \alpha_1^{k-1}, \alpha_2^{k-1}, \) and \( \alpha_3^{k-1} \) are calculated from

\[
\begin{align*}
\alpha_1^{k-1} &= Eb(u_{NM}(\mu, t_{k-1})), \\
\alpha_2^{k-1} &= Eb(u_{NM}(\mu, t_{k-1}) + \frac{\Delta t}{2} u_{NM,1}^{k-1}), \\
\alpha_3^{k-1} &= Eb(u_{NM}(\mu, t_{k-1}) + \Delta t u_{NM,2}^{k-1}). \tag{3.34}
\end{align*}
\]

and \( A = \Phi^T M \Phi, \ C = \Phi^T \Psi, \ E = A^{-1} CB^{-1}, \) and \( L \in \mathbb{R}^N \) with \( L_j = \ell(\phi_j), 1 \leq j \leq N. \)
Offline-Online Procedure

The reduced-basis model in (3.32) – (3.34) is now ready for offline-online decomposition. Similarly to the first formulation,

— In the offline stage, (performed once)

- First, solve for the DG solutions of the field variable, which consist the snapshot set \({ u_h(x; \tilde{\mu}_p), \tilde{\mu}_p \in \tilde{\mathcal{D}}, 1 \leq p \leq P }\)

- Next, apply the POD procedure to construct the approximation space of the field variable \( \Phi_N^\alpha \), and the approximation space of the nonlinear residual vector \( \Psi_M^R \).

- Then use the empirical interpolation procedure to obtain the set of interpolation points \( T_M^R \) for the residual vector.

- Finally form and store the parameter-independent quantities \( \Phi, \Psi, \) and \( E \).

— In the online stage (performed many times), we simply perform the sum (3.33) to obtain \( \alpha(\mu, t_k) \in \mathbb{R}^N \), and evaluate \( s_{N,M}(\mu, t_k) \) from (3.32).

The online operation count in this formulation is \( O(4K(MN + \kappa M)) \) to solve for the reduced-basis coefficients \( \alpha(\mu, t_k), 0 \leq k \leq K \), and \( O(KN) \) to compute the output \( s_{N,M}(\mu, t_k) \) at \( t^k, 0 \leq k \leq K \). Note that to compute \( b^{uN,M}(\mu, t_k) \) means to compute the residual vector \( R(\Phi \alpha(\mu, t_k); \mu) \) at the \( M \) interpolation points, which costs \( O(\kappa M) \) operations; the factor \( \kappa \) depends on the dimensionality of the problem \( d \) and polynomial order of approximations \( p \).

3.5 A Posterior Error Estimator

The reliability of very low-dimensional reduced-basis approximations of parametrized partial differential equations can only be assured by a posteriori error estimation procedures. To ensure that our RB approximation satisfies the accuracy level of interest, we hence need to develop associated a posterior error estimators. Following [35], we define asymptotic output upper and lower estimators respectively as

\[
s_{N,M}^+(\mu, t_k) = s_{N,M}(\mu, t_k) \pm \Delta_{N,M}^*(\mu, t_k),
\]

(3.35)
where $\Delta_{N,M}^*(\mu, t_k)$, the output error estimator, is given by

$$\Delta_{N,M}^*(\mu, t_k) = \frac{1}{\tau} |s_{2N,2M}(\mu, t_k) - s_{N,M}(\mu, t_k)|$$

(3.36)

for some $\tau \in (0, 1)$. In addition, we define the asymptotic error estimator for the error norm $\|u_h(\mu, t_k) - u_{N,M}(\mu, t_k)\|

$$\Delta_{N,M}^k(\mu, t_k) = \frac{1}{\tau} \|u_{2N,2M}(\mu, t_k) - u_{N,M}(\mu, t_k)\|$$

(3.37)

Here $s_{2N,2M}(\mu, t_k)$ and $u_{2N,2M}(\mu, t_k)$ are the RB output and solution associated with the “twice-richer” approximation spaces $\{\Phi_{2N}^n, \Psi_{2M}^j \}$ (for function approximation formulation), or $\{\Phi_{2M}^n, \Psi_{2M}^j \}$ (for residual approximation formulation). Hence, in terms of computational cost, $\Delta_{N,M}^*(\mu, t_k)$ will be four times more expensive than $s_{N,M}(\mu, t_k)$.

To measure the sharpness of our error estimators, the effectivities are introduced,

$$\eta_{N,M}^*(\mu, t_k) = \frac{\Delta_{N,M}^*(\mu, t_k)}{|s_h(\mu, t_k) - s_{N,M}(\mu, t_k)|}, \quad \eta_{N,M}^k(\mu, t_k) = \frac{\Delta_{N,M}^k(\mu, t_k)}{\|u_h(\mu, t_k) - u_{N,M}(\mu, t_k)\|}.$$

(3.38)

It thus follows that

$$\frac{1}{\tau} (1 - \epsilon^*) \leq \eta_{N,M}^*(\mu, t_k) \leq \frac{1}{\tau} (1 + \epsilon^*), \quad \frac{1}{\tau} (1 - \epsilon^*) \leq \eta_{N,M}^k(\mu, t_k) \leq \frac{1}{\tau} (1 + \epsilon^*).$$

(3.39)

where

$$\epsilon^* = \frac{|s_h(\mu, t_k) - s_{2N,2M}(\mu, t_k)|}{|s_h(\mu, t_k) - s_{N,M}(\mu, t_k)|}, \quad \epsilon^k = \frac{\|u_h(\mu, t_k) - u_{2N,2M}(\mu, t_k)\|}{\|u_h(\mu, t_k) - u_{N,M}(\mu, t_k)\|}.$$

(3.40)

Hence, under the hypothesis that $\epsilon^* \to 0$, as $N \to \infty, M \to \infty$, there exist finite integers $N^*$ and $M^*$ such that $\eta_{N,M}^*(\mu, t_k) \geq 1, \forall N \geq N^*, \forall M \geq M^*$. In general, our error estimator $\Delta_{N,M}^*(\mu, t_k)$ is not quite a rigorous upper bound. However, if $\epsilon^* \to 0$ very fast, we expect that the effectivity $\eta_{N,M}^*(\mu, t_k)$ shall be close to $1/\tau$. A similar argument applies for $\Delta_{N,M}^k(\mu, t_k)$.

In the following section, we will apply the two RB approximation formulations developed so far to two problems, namely, 1D viscous Burger problem and 2D Buckley-Leverret problem. The a posteriori error estimators described in this section will be applied to certify the results.
3.6 Numerical Examples

3.6.1 One-Dimensional Viscous Burger's Equation

The first example we present here is the one-dimensional viscous Burger's equation

$$\frac{\partial u}{\partial t} + \nabla \cdot \left( \frac{1}{2} u^2 \right) - \nu \nabla^2 u = 0, \quad \text{in } \Omega \times (0, 0.3) \tag{3.41}$$

with initial data $u_0(x) = 1/4 + \sin(\pi(2x - 1))/2$ and periodic boundary condition on boundary $\partial \Omega$. Here $\Omega \equiv [0, 1]$, the viscosity $\nu$ varies in the range $D \equiv [0.01, 0.1]$. The output $s(\mu, t)$ is evaluated as $s(\mu, t) = \ell(u(\mu, t))$ for $\ell(v) = \int_{\Omega} v$.

The purpose of this simple 1D nonlinear problem is to illustrate and compare two different reduce-basis treatment for the nonlinear term $\frac{1}{2} u^2$, namely, Galerkin approximation and empirical interpolation procedure. As described in Section 3.2, Galerkin approximation can be used for the low order polynomial present in viscous Burger's equation. However, note that this does not apply to the numerical flux term $f$ that arises in the RKDG method; hence, continuous Galerkin (CG) space discretization has to be used for this purpose, as there is no numerical flux term involved in this method.

In this time-dependent problem, we define the maximum relative error of solution as

$$\epsilon_{N,M,\text{max,rel}}^u = \max_{\mu \in D} \frac{|u(\tilde{\mu}) - u_{N,M}(\tilde{\mu})|_X}{|u(\tilde{\mu})|_X} \tag{3.42}$$

and maximum relative output error as

$$\epsilon_{N,M,\text{max,rel}}^s = \max_{\mu \in D} \frac{|s(\tilde{\mu}) - s_{N,M}(\tilde{\mu})|}{|s(\tilde{\mu})|} \tag{3.43}$$

In Figures 3-1 to 3-4, we present the convergence of solution and output of the two approximations, namely, Galerkin approximation and empirical interpolation approximation.

From these four figures, it is observed that the errors in both solution and output converge as the dimension of the reduced-basis model increases. Eventually, the empirical interpolation approximation method (with sufficient interpolation points, i.e. larger $M$) is able to achieve a comparable accuracy level as the Galerkin approximation method\(^1\).

\(^1\)Note that the convergence of Galerkin approximation only depends on $N$. 38
Figure 3-1: Convergence of solution of the reduced-basis Galerkin approximation for One-Dimensional Burger's equation.

Figure 3-2: Convergence of output of the reduced-basis Galerkin approximation for One-Dimensional Burger's equation.
Figure 3-3: Convergence of solution of the reduced-basis empirical interpolation approximation for One-Dimensional Burger's equation

Figure 3-4: Convergence of output of the reduced-basis empirical interpolation approximation output for One-Dimensional Burger's equation
Table 3.1: Effectivities of Galerkin approximation for One-Dimensional viscous Burger’s equation

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\Delta_{N,2N}^k$</th>
<th>$\eta_{N,M}^{k}$</th>
<th>$\Delta_{N,2N}^u$</th>
<th>$\eta_{N,M}^{u}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$1.8835 \times 10^{-1}$</td>
<td>2.0353</td>
<td>$2.3381 \times 10^{-1}$</td>
<td>1.9006</td>
</tr>
<tr>
<td>4</td>
<td>$2.2233 \times 10^{-2}$</td>
<td>4.0924</td>
<td>$3.0029 \times 10^{-2}$</td>
<td>1.8342</td>
</tr>
<tr>
<td>7</td>
<td>$5.3567 \times 10^{-3}$</td>
<td>2.3378</td>
<td>$9.4860 \times 10^{-3}$</td>
<td>2.2373</td>
</tr>
<tr>
<td>10</td>
<td>$6.2330 \times 10^{-3}$</td>
<td>2.3378</td>
<td>$9.4688 \times 10^{-3}$</td>
<td>2.2373</td>
</tr>
</tbody>
</table>

Table 3.2: Effectivities of empirical interpolation approximation for One-Dimensional viscous Burger’s equation

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\Delta_{N,2N}^k$</th>
<th>$\eta_{N,M}^{k}$</th>
<th>$\Delta_{N,2N}^u$</th>
<th>$\eta_{N,M}^{u}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$1.8862 \times 10^{-1}$</td>
<td>1.9991</td>
<td>$2.3378 \times 10^{-1}$</td>
<td>1.8990</td>
</tr>
<tr>
<td>4</td>
<td>$2.2249 \times 10^{-2}$</td>
<td>3.8922</td>
<td>$3.0193 \times 10^{-2}$</td>
<td>1.8324</td>
</tr>
<tr>
<td>7</td>
<td>$5.3516 \times 10^{-3}$</td>
<td>1.2886</td>
<td>$9.4548 \times 10^{-3}$</td>
<td>1.2737</td>
</tr>
<tr>
<td>10</td>
<td>$6.4935 \times 10^{-3}$</td>
<td>1.6399</td>
<td>$9.4026 \times 10^{-3}$</td>
<td>1.4011</td>
</tr>
</tbody>
</table>

To verify that both our RB approximations satisfy the accuracy level of interest, we use the a posteriori error estimators introduced in Section 3.5, and $\tau = 0.5$. The effectivities are presented in Table 3.1 and 3.2, and they are of order $O(1)$ as $\epsilon \rightarrow 0$.

The empirical interpolation approximation method has shown as good performance as the direct Galerkin approximation method. Moreover, the online stage computational time cost (normalized with respect to CG approach) tabulated in Table 3.3 reveals that the empirical interpolation approximation is more efficient than Galerkin approximation, as it provides special treatment and construct reduced-basis for the nonlinear terms as well.

3.6.2 Two-Dimensional Buckley-Leverett Equation

Problem Description

Our second example is the two-dimensional Buckley-Leverett equation

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) - \nu \nabla^2 u = 0, \quad \text{in } \Omega \times (0,0.5]$$  (3.44)

with the initial data $u_0(x) = e^{-(x^2+y^2)/0.025}$ and homogeneous Dirichlet boundary condition on the boundary $\partial \Omega$. Here $\Omega \equiv [-1.25,1.25[ \times [-1.25,1.25[, the viscosity $\nu$ varies in the range
Table 3.3: Online computational time for One-Dimensional viscous Burger's equation

<table>
<thead>
<tr>
<th>$N$</th>
<th>Galerkin Approx.</th>
<th>$N$</th>
<th>$M$</th>
<th>Empirical Interp. Approx.</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$8.5498 \times 10^{-3}$</td>
<td>2</td>
<td>4</td>
<td>$4.1173 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>$8.6896 \times 10^{-3}$</td>
<td>4</td>
<td>8</td>
<td>$4.2184 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>$9.7093 \times 10^{-3}$</td>
<td>8</td>
<td>16</td>
<td>$4.4452 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>$1.0625 \times 10^{-2}$</td>
<td>16</td>
<td>32</td>
<td>$5.0195 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>$1.0876 \times 10^{-2}$</td>
<td>20</td>
<td>40</td>
<td>$5.0564 \times 10^{-3}$</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 3-5: Contour plot of the DG solution for Two-Dimensional Buckley-Leverett equation at $T = 0.5$

$D \equiv [0.05, 0.1]$, and the flux vector $f(u) = (f_1(u), f_2(u))$ is given by

$$
\begin{align*}
    f_1(u) &= \frac{u^2}{u^2 + (1 - u)^2}, \\
    f_2(u) &= f_1(u)(1 - 5(1 - u)^2).
\end{align*}
$$

The output $s(\mu, t)$ is evaluated as $s(\mu, t) = \ell(u(\mu, t))$ for $\ell(v) = \int_0^1 v$. The contour plot of the computed DG solution at the final time $T = 0.5$ is shown in Figure 3-5. The solutions is obtained with fourth-order finite element approximation space $X_h^{p=4}$ of dimension $N = 12,000$, and $\nu = 0.05$.
Numerical Results-Function Formulation

The three nonlinear terms, $f_1$, $f_2$, $\hat{f}$, in the 2D Buckley-Leverett equation have to be specially approximated by the empirical interpolation method. Before this method is incorporated to the reduced-basis model of the whole system, its approximation accuracy has to be justified. In Figure 3-6 we compare the accuracy of the functions and their approximates. All the three functions can be approximated more accurately with more and more interpolation points being used.

Following the definitions in (3.42) and (3.43), we plot $\epsilon^N_{N,M,max,rel}$ in Figure 3-7 as a function of $N$ and $M$, and $\epsilon^M_{N,M,max,rel}$ in Figure 3-8. The two figures show the same behavior: the errors converge as $N$ increases, and level off at smaller and smaller values as we increase $M$. This is because when $N$ is small, the dominating error is caused by the field variable $u_N$; when $N$ is large enough such that the field variable is approximated more accurately, the errors in the nonlinear terms start to dominate. The final error level of a particular $M$ is determined by the accuracy of the corresponding empirical interpolation approximation.

To verify that our RB approximation satisfy the accuracy level of interest, we use the a posteriori error estimator introduced in Section 3.6, and $\tau = 0.5$. The effectivities are presented in Table 3.4.
Figure 3-7: Convergence of the reduced-basis approximation output for Two-Dimensional Buckley-Leverett equation

Figure 3-8: Convergence of the reduced-basis approximation solution for Two-Dimensional Buckley-Leverett equation
The mean effectivities are of order $O(1)$ as $\epsilon \to 0$.

The efficiency of the reduced-basis system is measured by the online computational times of each timestep. The values are normalized with respect to the computational time of direct calculation of the truth approximation output, i.e. DG approach. Significant saving has been achieved: A factor of more than 100 saving for the largest reduced-basis system in our experiments. Most significantly, it is also possible to use a much larger timestep in our RB system, without incurring any accuracy loss, which further speeds up the calculation of output $s_{N,M}(\mu, t_k)$.

Table 3.4: Effectivities for Two-Dimensional Buckley-Leverett equation

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$\Delta^n_{N,2N}$</th>
<th>$\eta^u_{N,M}$</th>
<th>$\Delta^u_{N,2N}$</th>
<th>$\eta^v_{N,M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>$2.0255 \times 10^{-2}$</td>
<td>2.2322</td>
<td>$2.3682 \times 10^{-4}$</td>
<td>1.8748</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>$4.5120 \times 10^{-3}$</td>
<td>2.7415</td>
<td>$3.6648 \times 10^{-5}$</td>
<td>1.9501</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>$3.9822 \times 10^{-3}$</td>
<td>2.0399</td>
<td>$2.3300 \times 10^{-5}$</td>
<td>2.0425</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>$1.4793 \times 10^{-3}$</td>
<td>2.1870</td>
<td>$3.6034 \times 10^{-6}$</td>
<td>2.0102</td>
</tr>
<tr>
<td>50</td>
<td>40</td>
<td>$1.2819 \times 10^{-3}$</td>
<td>2.1897</td>
<td>$3.3151 \times 10^{-6}$</td>
<td>2.0217</td>
</tr>
<tr>
<td>60</td>
<td>50</td>
<td>$7.2419 \times 10^{-4}$</td>
<td>3.1641</td>
<td>$2.4356 \times 10^{-6}$</td>
<td>2.0579</td>
</tr>
</tbody>
</table>

Table 3.5: Online computational times (normalized) for Two-Dimensional Buckley-Leverett equation

<table>
<thead>
<tr>
<th>$N$</th>
<th>$M$</th>
<th>$RBA$</th>
<th>$DG$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>$3.8905 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>$4.3953 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>$4.4730 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>$6.1853 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>$6.4143 \times 10^{-3}$</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>$7.3889 \times 10^{-3}$</td>
<td>1</td>
</tr>
</tbody>
</table>
Chapter 4

Extension to The Euler Equations

So far we have considered reduced-basis approximations for time-dependent parameterized PDEs of up to two-dimensional scalar problems. In this chapter, we will continue to extend the approach developed to systems of equations, and consider RB approximations to unsteady compressible Euler equations. We present here the function approximation formulation for the one-dimensional problem. The derivation of the residual approximation formulation and generalization to multi-dimensional problems is straightforward.

4.1 Problem Formulations

The one-dimensional Euler equations of compressible flows in conservative form are

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0, \quad \text{in } \Omega \times (0,T]$$

(4.1)

with appropriate boundary conditions and initial condition \(u(x,t=0) = u_0(x)\). Here, the conservative state \(u\) and the inviscid flux vector \(F(u)\) are given by

$$u = \begin{pmatrix} u^1 \\ u^2 \\ u^3 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho v \\ \rho e \end{pmatrix}, \quad F(u) = \begin{pmatrix} f_1(u) \\ f_2(u) \\ f_3(u) \end{pmatrix} = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(\rho e + p) \end{pmatrix},$$

(4.2)
where \( \rho \) is the fluid density, \( v \) is the velocity, \( p \) is the pressure, and \( e \) is the total internal energy per unit mass. The perfect gas equation of state relates the static pressure \( p \) to components of the state vector \( u \) as

\[
p = (\gamma - 1)\rho (e - v^2/2),
\]

(4.3)

where \( \gamma \) is the ratio of specific heats of the fluid. The parameters we consider is the static pressure variation at outflow, which shall be denoted as \( \mu \) and varies in the parameter space \( \mathcal{D} \).

We now define the DG weak formulation for the governing equations (4.1) – (4.3). We assume that we are given a decomposition \( T_h \) of the domain \( \Omega \) and associated function space \( X_h^p \) defined in Section 2.1. In addition, we introduce

\[
Z_h^p = \left\{ \sigma \in (L^2(\Omega))^3 \mid \sigma_i|_{T_h} \in P^p(T_h), \quad \forall T_h \in \mathcal{T}_h, \ 1 \leq i \leq 3 \right\}.
\]

(4.4)

The weak formulation then takes the following form: find \( u_h = (u_h^1, u_h^2, u_h^3) \in Z_h^p \) such that

\[
\sum_{T_h \in \mathcal{T}_h} \left\{ \int_{T_h} \frac{\partial u_h}{\partial t} v_h - \int_{T_h} F(u_h) \nabla v_h + \int_{\partial T_h} \hat{F}(u_h, n) v_h \right\} = 0, \quad \forall v_h \in Z_h^p
\]

(4.5)

where \( \hat{F}(u_h, n) \) is the numerical flux. The Roe flux function [38] is used here to define the numerical flux as

\[
\hat{F}(u_h, n) = \frac{1}{2} \left[ F(u_h^+) + F(u_h^-) \right] \cdot n - \frac{1}{2} |A| (u_h^+ - u_h^-),
\]

(4.6)

where \( A = [\partial F_i(u_h^R)/\partial u] \cdot n \) is the Jacobian matrix evaluated at the Roe average state \( u_h^R \), and \( |A| = S|\Lambda|S^{-1} \), \( S \) being the matrix of the right eigenvectors of \( A \) and \( |\Lambda| \) being the diagonal matrix of the absolute eigenvalues of \( A \).

Finally, by decomposing \( \hat{F}(u_h, n) = [\hat{f}_1(u_h, n), \hat{f}_2(u_h, n), \hat{f}_3(u_h, n)]^T \), we can write (4.5) more explicitly as: for all \( v_h \in X_h^p \), find \( u_h = (u_h^1, u_h^2, u_h^3) \in X_h^p \times X_h^p \times X_h^p \) such that

\[
\sum_{T_h \in \mathcal{T}_h} \int_{T_h} \frac{\partial u_h^1}{\partial t} v_h = \sum_{T_h \in \mathcal{T}_h} \left\{ \int_{T_h} f_1(u_h) \nabla v_h - \int_{\partial T_h} \hat{f}_1(u_h, n) v_h \right\},
\]

\[
\sum_{T_h \in \mathcal{T}_h} \int_{T_h} \frac{\partial u_h^2}{\partial t} v_h = \sum_{T_h \in \mathcal{T}_h} \left\{ \int_{T_h} f_2(u_h) \nabla v_h - \int_{\partial T_h} \hat{f}_2(u_h, n) v_h \right\},
\]

\[
\sum_{T_h \in \mathcal{T}_h} \int_{T_h} \frac{\partial u_h^3}{\partial t} v_h = \sum_{T_h \in \mathcal{T}_h} \left\{ \int_{T_h} f_3(u_h) \nabla v_h - \int_{\partial T_h} \hat{f}_3(u_h, n) v_h \right\}.
\]

(4.7)
with the initial conditions $u_h^1(x; \mu; t = 0) = u_{h0}^1(x)$, $u_h^2(x; \mu; t = 0) = u_{h0}^2(x)$, and $u_h^3(x; \mu; t = 0) = u_{h0}^3(x)$; here $u_{h0}^1(x)$, $u_{h0}^2(x)$, and $u_{h0}^3(x)$ are the $L^2$-projection of $u_0^1(x)$, $u_0^2(x)$, and $u_0^3(x)$ on $X_h^p$, respectively. The FE approximation output is then given by

$$s_h(\mu, t) = \ell(u_h^1(\mu, t)). \quad (4.8)$$

We use the fourth-order explicit RK scheme described in Section 2.2 to integrate the system (4.7)-(4.8) in time, thereby obtaining the truth approximations $u_h(\mu, t_k)$ and $s_h(\mu, t_k), k = 1, \ldots, K$.

### 4.2 Fully Discrete Equations

We begin by introducing several approximation spaces constructed upon the set of snapshots $\{u_h(\mu_p), \mu_p \in \tilde{D}, 1 \leq p \leq P\}$. We first apply the POD procedure to construct RB approximation spaces $\Phi_{N1}^1 = \text{span}\{\phi_1^1, \ldots, \phi_{N1}^1\}$ for $u_h^1$, $\Phi_{N2}^2 = \text{span}\{\phi_1^2, \ldots, \phi_{N2}^2\}$ for $u_h^2$, and $\Phi_{N3}^3 = \text{span}\{\phi_1^3, \ldots, \phi_{N3}^3\}$ for $u_h^3$. We then use POD and empirical interpolation procedures to construct collateral RB interpolation spaces $\Psi_{M2}^2 = \text{span}\{\psi_1^2, \ldots, \psi_{M2}^2\}$, $\Psi_{M3}^3 = \text{span}\{\psi_1^3, \ldots, \psi_{M3}^3\}$, and associated sets of interpolation points $T_{M2}^2 = \{z_1^2, \ldots, z_{M2}^2\}$, $T_{M3}^3 = \{z_1^3, \ldots, z_{M3}^3\}$ for the nonlinear functions $f_2(u_h), f_3(u_h)$, respectively; note that $f_1(u_h) = u_h^3$ is linear and does not need the empirical interpolation treatment. Finally, in a similar manner, we build collateral RB interpolation spaces $\hat{\Phi}_{M1}^1 = \text{span}\{\hat{\psi}_1^1, \ldots, \hat{\psi}_{M1}^1\}$, $\hat{\Phi}_{M2}^2 = \text{span}\{\hat{\psi}_1^2, \ldots, \hat{\psi}_{M2}^2\}$, $\hat{\Phi}_{M3}^3 = \text{span}\{\hat{\psi}_1^3, \ldots, \hat{\psi}_{M3}^3\}$, and associated sets of interpolation points $T_{M1}^1 = \{z_1^1, \ldots, z_{M1}^1\}$, $T_{M2}^2 = \{z_1^2, \ldots, z_{M2}^2\}$, $T_{M3}^3 = \{z_1^3, \ldots, z_{M3}^3\}$ for the nonlinear functions $\hat{f}_1(u_h, n), \hat{f}_2(u_h, n), \hat{f}_3(u_h, n)$, respectively; these functions are defined on the edges $E(T_h)$. For notational convenience, we assume that $N \equiv N_1 = N_2 = N_3$ and $M \equiv M_1 = M_2 = M_3 = M^1 = M^2 = M^3$.

Applying a Galerkin projection to the system (4.7) and replacing the nonlinear functions by the coefficient function approximations, we obtain the reduced-order model: for any $\mu \in D$, we evaluate

$$s_{N,M}(\mu, t) = \ell(u^1_{N,M}(\mu, t)). \quad (4.9)$$
where \( u_{N,M}(\mu, t) = (u^1_{N,M}(\mu, t), u^2_{N,M}(\mu, t), u^3_{N,M}(\mu, t))^T \in \Phi^1_N \times \Phi^2_N \times \Phi^3_N \) satisfy

\[
\sum_{T_h \in T_h} \int_{T_h} \frac{\partial u^1_{N,M}}{\partial t} u_N = \sum_{T_h \in T_h} \left\{ \int_{T_h} u^3_{N,M} \nabla u_N - \int_{\partial T_h} \hat{J}^1_{N,M} v_N \right\}, \quad \forall \nu_N \in \Phi^1_N
\]

\[
\sum_{T_h \in T_h} \int_{T_h} \frac{\partial u^2_{N,M}}{\partial t} u_N = \sum_{T_h \in T_h} \left\{ \int_{T_h} \hat{J}^2_{N,M} \nabla u_N - \int_{\partial T_h} \hat{J}^2_{N,M} v_N \right\}, \quad \forall \nu_N \in \Phi^2_N
\]

\[
\sum_{T_h \in T_h} \int_{T_h} \frac{\partial u^3_{N,M}}{\partial t} u_N = \sum_{T_h \in T_h} \left\{ \int_{T_h} \hat{J}^3_{N,M} \nabla u_N - \int_{\partial T_h} \hat{J}^3_{N,M} v_N \right\}, \quad \forall \nu_N \in \Phi^3_N.
\]

(4.10)

Here \( \hat{J}^1_{N,M}, \hat{J}^2_{N,M}, \hat{J}^1_{1,M}, \hat{J}^2_{2,M}, \hat{J}^3_{2,M}, \hat{J}^3_{3,M} \) are the coefficient function approximations to the nonlinear functions \( f_2(u_{N,M}), f_3(u_{N,M}), \hat{f}_1(u_{N,M}), \hat{f}_2(u_{N,M}, n), \hat{f}_3(u_{N,M}, n) \), respectively.

To arrive at the matrix-vector form for the system (4.9)-(4.10), we first expand the RB approximations \( u^1_{N,M}(\mu, t) \in \Phi^1_N, u^2_{N,M}(\mu, t) \in \Phi^2_N, \) and \( u^3_{N,M}(\mu, t) \in \Phi^3_N \) as

\[
u^1_{N,M}(\mu, t) = \sum_{n=1}^{N} \alpha^1_n(\mu, t) \phi^1_n, \quad \nu^2_{N,M}(\mu, t) = \sum_{n=1}^{N} \alpha^2_n(\mu, t) \phi^2_n, \quad \nu^3_{N,M}(\mu, t) = \sum_{n=1}^{N} \alpha^3_n(\mu, t) \phi^3_n.
\]

(4.11)

It then follows that our coefficient function approximations \( \hat{J}^1_{2,M}, \hat{J}^2_{3,M}, \hat{J}^1_{1,M}, \hat{J}^2_{2,M}, \hat{J}^3_{2,M}, \hat{J}^3_{3,M} \) are given by

\[
\begin{align*}
\hat{J}^1_{2,M}(x; \mu, t) &= \sum_{m=1}^{M} \beta^1_m(\mu, t) \psi^1_m(x), \quad \hat{J}^2_{3,M}(x; \mu, t) = \beta^3(\mu, t) = f_3(D_1^{12} \alpha^1(\mu, t), D_2^{22} \alpha^2(\mu, t), D_3^{32} \alpha^3(\mu, t)) \\
\hat{J}^2_{1,M}(x; \mu, t) &= \sum_{m=1}^{M} \beta^2_m(\mu, t) \psi^2_m(x), \quad \hat{J}^3_{2,M}(x; \mu, t) = \beta^3(\mu, t) = f_3(D_1^{12} \alpha^1(\mu, t), D_2^{22} \alpha^2(\mu, t), D_3^{32} \alpha^3(\mu, t)) \\
\hat{J}^3_{1,M}(x; \mu, t) &= \sum_{m=1}^{M} \beta^3_m(\mu, t) \psi^3_m(x), \quad \hat{J}^3_{3,M}(x; \mu, t) = \beta^3(\mu, t) = f_3(D_1^{12} \alpha^1(\mu, t), D_2^{22} \alpha^2(\mu, t), D_3^{32} \alpha^3(\mu, t))
\end{align*}
\]

where for \( 1 \leq i, m \leq M, 1 \leq n \leq N, \)

\[
\begin{align*}
B^2_{im} &= \psi^2_m(z^2_i), \quad D^{12} = \phi^1_n(z^2_i), \quad D^{22} = \phi^2_n(z^2_i), \quad D^{32} = \phi^3_n(z^2_i), \\
B^3_{im} &= \psi^3_m(z^3_i), \quad D^{13} = \phi^1_n(z^3_i), \quad D^{23} = \phi^2_n(z^3_i), \quad D^{33} = \phi^3_n(z^3_i)
\end{align*}
\]

(4.12)

(4.13)
\[ \hat{B}_{1m} = \hat{\psi}_m^1(\hat{z}_1^1), \quad \hat{D}_{1m}^{11} = \phi_n^1(\hat{z}_1^1), \quad \hat{D}_{1m}^{21} = \phi_n^2(\hat{z}_1^1), \quad \hat{D}_{1m}^{31} = \phi_n^3(\hat{z}_1^1), \quad \hat{n}_1^1 = n(\hat{z}_1^1) \quad (4.14) \]

\[ \hat{B}_{2m} = \hat{\psi}_m^2(\hat{z}_1^2), \quad \hat{D}_{2m}^{12} = \phi_n^1(\hat{z}_1^2), \quad \hat{D}_{2m}^{22} = \phi_n^2(\hat{z}_1^2), \quad \hat{D}_{2m}^{32} = \phi_n^3(\hat{z}_1^2), \quad \hat{n}_1^2 = n(\hat{z}_1^2) \quad (4.15) \]

\[ \hat{B}_{3m} = \hat{\psi}_m^3(\hat{z}_1^3), \quad \hat{D}_{3m}^{13} = \phi_n^1(\hat{z}_1^3), \quad \hat{D}_{3m}^{23} = \phi_n^2(\hat{z}_1^3), \quad \hat{D}_{3m}^{33} = \phi_n^3(\hat{z}_1^3), \quad \hat{n}_1^3 = n(\hat{z}_1^3). \quad (4.16) \]

Inserting the representations (4.11) and our coefficient-function approximations into (4.10) yields

\[
\sum_{j=1}^{N} \sum_{T_h \in T_h} \int_{T_h} \phi_j^1 \phi_j^1 \alpha_j^1(\mu, t) = \sum_{k=1}^{N} \sum_{T_h \in T_h} \int_{T_h} \phi_k^1 \nabla \phi_k^1 \alpha_k^2(\mu, t) - \sum_{m=1}^{M} \sum_{T_h \in T_h} \int_{\partial T_h} \psi_m^1 \phi_m^1 \beta_m^1(\mu, t), \quad 1 \leq i \leq N
\]

\[
\sum_{j=1}^{N} \sum_{T_h \in T_h} \int_{T_h} \phi_j^2 \phi_j^2 \beta_j^2(\mu, t) = \sum_{k=1}^{N} \sum_{T_h \in T_h} \int_{T_h} \phi_k^2 \nabla \phi_k^2 \beta_k^2(\mu, t) - \sum_{m=1}^{M} \sum_{T_h \in T_h} \int_{\partial T_h} \psi_m^2 \phi_m^2 \beta_m^2(\mu, t), \quad 1 \leq i \leq N
\]

\[
\sum_{j=1}^{N} \sum_{T_h \in T_h} \int_{T_h} \phi_j^3 \phi_j^3 \beta_j^3(\mu, t) = \sum_{k=1}^{N} \sum_{T_h \in T_h} \int_{T_h} \phi_k^3 \nabla \phi_k^3 \beta_k^3(\mu, t) - \sum_{m=1}^{M} \sum_{T_h \in T_h} \int_{\partial T_h} \psi_m^3 \phi_m^3 \beta_m^3(\mu, t), \quad 1 \leq i \leq N
\]

in terms of which the RB output can be subsequently calculated as

\[ s_{N,M}(\mu, t) = \sum_{n=1}^{N} \alpha_n^1(\mu, t) \ell(\phi_1^1). \quad (4.17) \]

It remains to develop the offline-online procedure for the rapid evaluation of \( s_{N,M}(\mu, t) \).

### 4.3 Offline/Online Procedure

Let us first introduce the following matrices

\[ A_{1m}^1 = \sum_{T_h \in T_h} \int_{T_h} \phi_n^1 \phi_1^1, \quad A_{1m}^2 = \sum_{T_h \in T_h} \int_{T_h} \phi_n^2 \phi_1^2, \quad A_{1m}^3 = \sum_{T_h \in T_h} \int_{T_h} \phi_n^3 \phi_1^3, \quad (4.18) \]

\[ C_{1m}^1 = \sum_{T_h \in T_h} \int_{T_h} \phi_n^2 \nabla \phi_1^1, \quad C_{1m}^2 = \sum_{T_h \in T_h} \int_{T_h} \phi_n^2 \nabla \phi_1^2, \quad C_{1m}^3 = \sum_{T_h \in T_h} \int_{T_h} \phi_n^2 \nabla \phi_1^3, \quad (4.19) \]

\[ C_{2m}^1 = \sum_{T_h \in T_h} \int_{T_h} \psi_m^1 \phi_1^2, \quad C_{2m}^2 = \sum_{T_h \in T_h} \int_{T_h} \psi_m^2 \phi_1^2, \quad C_{2m}^3 = \sum_{T_h \in T_h} \int_{T_h} \psi_m^3 \phi_1^3, \quad (4.19) \]

\[ C_{3m}^1 = \sum_{T_h \in T_h} \int_{T_h} \psi_m^1 \phi_1^3, \quad C_{3m}^2 = \sum_{T_h \in T_h} \int_{T_h} \psi_m^2 \phi_1^3, \quad C_{3m}^3 = \sum_{T_h \in T_h} \int_{T_h} \psi_m^3 \phi_1^3, \]
for $1 \leq i, n \leq N, 1 \leq m \leq M$; and

$$
E^1 = (A^1)^{-1}(C^1), \quad E^2 = (A^2)^{-1}(C^2)(B^2)^{-1}, \quad E^3 = (A^3)^{-1}(C^3)(B^3)^{-1} \quad (4.20)
$$

$$
\hat{E}^1 = (A^1)^{-1}(\hat{C}^1)(\hat{B}^1)^{-1}, \quad \hat{E}^2 = (A^2)^{-1}(\hat{C}^2)(\hat{B}^2)^{-1}, \quad \hat{E}^3 = (A^3)^{-1}(\hat{C}^3)(\hat{B}^3)^{-1}. \quad (4.21)
$$

It thus follows from (4.17) and (4.12)-(4.16) that we obtain the ODE system

$$
\dot{\alpha}^1(\mu, t) = E^1 \alpha^2 - \hat{E}^1 \hat{f}_1(\hat{D}^1 \alpha^1, \hat{D}^2 \alpha^2, \hat{D}^3 \alpha^3, \hat{n}^1)
$$

$$
\dot{\alpha}^2(\mu, t) = E^2 \alpha^2 - \hat{E}^2 \hat{f}_2(\hat{D}^1 \alpha^1, \hat{D}^2 \alpha^2, \hat{D}^3 \alpha^3, \hat{n}^2) \quad (4.22)
$$

$$
\dot{\alpha}^3(\mu, t) = E^3 \alpha^3 - \hat{E}^3 \hat{f}_3(\hat{D}^1 \alpha^1, \hat{D}^2 \alpha^2, \hat{D}^3 \alpha^3, \hat{n}^3),
$$

with the initial conditions $\alpha^1(\mu, t = 0) = \alpha_0^1$, $\alpha^2(\mu, t = 0) = \alpha_0^2$, and $\alpha^3(\mu, t = 0) = \alpha_0^3$, where

$$
\alpha_{0i}^1 = (\phi_i^1, u_{00}^1), \quad \alpha_{0i}^2 = (\phi_i^2, u_{00}^2), \quad \alpha_{0i}^3 = (\phi_i^3, u_{00}^3), \quad 1 \leq i \leq N. \quad (4.23)
$$

The RB output is thus calculated by

$$
s_{N,M}(\mu, t) = L^T \alpha^1(\mu, t), \quad (4.24)
$$

where $L_i = \ell(\phi_i^1), 1 \leq i \leq N$. By using the 4th order RK scheme to solve the system (4.22), we obtain the RB approximations $s_{N,M}(\mu, t_k)$ to $s_h(\mu, t_k)$ for $k = 0, \ldots, K$. The procedure for the rapid evaluation of $s_{N,M}(\mu, t_k), 0 \leq k \leq K$, is described below.

In the offline stage — performed once — we form and store the parameter-independent quantities $\alpha_{0i}^1, \alpha_{0i}^2, \alpha_{0i}^3, \hat{n}^1, \hat{n}^2, \hat{n}^3, D_{12}, D_{22}, D_{32}, D_{13}, D_{23}, D_{33}, \hat{D}^1, \hat{D}^2, \hat{D}^3, \hat{D}_{12}, \hat{D}_{22}, \hat{D}_{32}, \hat{D}_{13}, \hat{D}_{23}, \hat{D}_{33}, E^1, E^2, E^3, \hat{E}^1, \hat{E}^2, \hat{E}^3$. In the online stage — performed many times, for each new value of $\mu \in \mathcal{D}$ — we simply solve (4.22) for $\alpha^1(\mu, t_k), \alpha^2(\mu, t_k), \alpha^3(\mu, t_k)$ at cost $O \left(4 (20MN + N^2)\right)$, and evaluate $s_{N,M}(\mu, t_k)$ from (4.24) at cost $O(N)$; (note that the 4th order RK scheme requires four evaluations of the right-hand side of (4.22) and each of evaluation takes $O \left(20MN + N^2\right)$). The operation count for the online stage is thus $O \left(4K (20MN + N^2)\right)$. The online complexity is again independent of $N$.

Finally, the a posteriori error estimator $\Delta_{N,M}^k(\mu, t_k)$ can be obtained from (3.25), which in turn
necessitates calculation of \( s_{2N,2M}(\mu, t_k) \). The overall computational cost is thus increased by a factor of 4.

### 4.4 Numerical Examples

#### 4.4.1 Problem Description

We solve a quasi-1D Euler equations for unsteady flow in a duct of cross-section \( a(x) \), on the domain \( \Omega = [0, 1] \), defined as

\[
a(x) \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} (a(x)F(u)) - \frac{da(x)}{dx} p = 0, \quad \text{in} \quad \Omega \times (0, 1]
\]

with the conservative state \( u \) and the inviscid flux vector \( F(u) \) as shown by (4.2), and \( p = (0, p, 0)^T \).

The geometry of the duct is defined by

\[
a(x) = \begin{cases} 
1, & 0 \leq x \leq 0.1 \\
1 - 0.3 \cos^2(\pi(x - 0.5)/0.8), & 0.1 < x < 0.9 \\
1, & 0.9 \leq x \leq 1 
\end{cases}
\]

(4.26)

The initial conditions are given by

\[
\rho_0(x) = a(x), \quad v_0(x) = 1, \quad e_0(x) = 20.3413.
\]

(4.27)

The boundary condition is the static pressure at outlet \( p_{ex} = 7.9365 + \delta \sin(2\pi t) \), where \( \delta \) is the magnitude of a small sinusoidal perturbation of the static pressure at outlet. Here \( \mu \equiv \delta \) is the only parameter of interest and varies in the range \( D \equiv [0, 0.5] \). Note that the case \( \delta = 0 \) results in steady flow. The output \( s(\mu, t) \) is evaluated as \( s(\mu, t) = \ell(\rho(\mu, t)) \) for \( \ell(v) = \int_0^1 v \).

The Mach number of the DG solutions at different time instances \( t = (0.2, 0.4, 0.6, 0.8, 1) \) is shown in Figure (4-1). These solutions are obtained with fifth-order finite element approximation space \( X_h^{p=5} \) of dimension \( \mathcal{N} = 1000 \), and for several \( \delta = (0, 0.2, 0.4, 0.5) \).
4.4.2 Numerical Results-Function Formulation

We now present the numerical results of reduced-basis method for unsteady quasi-1D Euler equations. The convergence of the reduced-basis approximation of output and field variable solutions are tabulated in Table 4.1, note that the \((N,M)\) pairs roughly correspond to the "knees" of what would have been observed in the \(N-M\)-convergence curves. Not surprisingly, again the output converges faster than that of the field variable solutions. Moreover, the three variable solutions converge at almost the same rate.

Next, the \textit{a posteriori} error estimators and the effectivities of the output approximation as well as solution approximations are tabulated in Table 4.2. As \(\epsilon \to 0\), the mean effectivities is of order
Table 4.2: Error estimators of output and solutions for One-Dimensional Euler equation

<table>
<thead>
<tr>
<th>( N )</th>
<th>( M )</th>
<th>( \Delta_{N,M}^1 )</th>
<th>( \Delta_{N,M}^2 )</th>
<th>( \Delta_{N,M}^3 )</th>
<th>( \Delta_{N,M}^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>4.1570 x 10^{-2}</td>
<td>5.5542 x 10^{-3}</td>
<td>1.8432 x 10^{-2}</td>
<td>1.4738 x 10^{-1}</td>
</tr>
<tr>
<td>15</td>
<td>24</td>
<td>1.1829 x 10^{-2}</td>
<td>2.2608 x 10^{-3}</td>
<td>6.2198 x 10^{-2}</td>
<td>6.1518 x 10^{-2}</td>
</tr>
<tr>
<td>20</td>
<td>28</td>
<td>5.6966 x 10^{-3}</td>
<td>1.3240 x 10^{-3}</td>
<td>3.4746 x 10^{-3}</td>
<td>3.4020 x 10^{-2}</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>5.5078 x 10^{-3}</td>
<td>1.3197 x 10^{-4}</td>
<td>3.4706 x 10^{-3}</td>
<td>3.3864 x 10^{-2}</td>
</tr>
<tr>
<td>25</td>
<td>40</td>
<td>2.9277 x 10^{-3}</td>
<td>7.6268 x 10^{-4}</td>
<td>1.8912 x 10^{-3}</td>
<td>1.9222 x 10^{-2}</td>
</tr>
<tr>
<td>30</td>
<td>44</td>
<td>1.8947 x 10^{-3}</td>
<td>6.4576 x 10^{-4}</td>
<td>1.6698 x 10^{-3}</td>
<td>1.6713 x 10^{-2}</td>
</tr>
</tbody>
</table>

Table 4.3: Effectivities of output and solutions for One-Dimensional Euler equation

<table>
<thead>
<tr>
<th>( N )</th>
<th>( M )</th>
<th>( \eta_{N,M}^1 )</th>
<th>( \eta_{N,M}^2 )</th>
<th>( \eta_{N,M}^3 )</th>
<th>( \eta_{N,M}^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>2.3444</td>
<td>2.4053</td>
<td>2.4988</td>
<td>2.3711</td>
</tr>
<tr>
<td>15</td>
<td>24</td>
<td>2.1692</td>
<td>2.2511</td>
<td>2.2440</td>
<td>2.2583</td>
</tr>
<tr>
<td>20</td>
<td>28</td>
<td>2.5113</td>
<td>2.4178</td>
<td>2.5411</td>
<td>2.9331</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>2.9992</td>
<td>2.5743</td>
<td>2.7865</td>
<td>3.0169</td>
</tr>
<tr>
<td>25</td>
<td>40</td>
<td>2.8828</td>
<td>3.0672</td>
<td>3.3748</td>
<td>4.1792</td>
</tr>
<tr>
<td>30</td>
<td>44</td>
<td>5.1459</td>
<td>3.1014</td>
<td>2.9913</td>
<td>2.9692</td>
</tr>
</tbody>
</table>

\( O(1) \).

The computational savings obtained by the use of the reduced-basis approximation in one dimensional problem are not significant. In fact, the offline cost of the reduced basis approach makes it uninteresting in this case. This situation will be clearly different when we consider the two and three dimensional problem, or one dimensional problem with larger finite element approximation space (i.e. larger \( N \)). Moreover, since a larger timestep can be used in the RB approximation, the total online computational saving can be more significant.

Table 4.4: Online computational times (normalized) for One-Dimensional Euler equation (per timestep)

<table>
<thead>
<tr>
<th>( N )</th>
<th>( M )</th>
<th>( RBA )</th>
<th>( DG )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>0.5099</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>24</td>
<td>0.5280</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>28</td>
<td>0.5584</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>0.5586</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>40</td>
<td>0.6124</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>44</td>
<td>0.6626</td>
<td>1</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusion

5.1 Summary

The goal of this thesis has been the development of reduced-basis approximations of the solution and output of nonlinear time-dependent parameterized convection-diffusion equations and a \textit{posteriori} error estimators to certify the approximated results. The developed method has also been extended to the quasi one-dimensional Euler equations for unsteady flow.

We began by introducing the primary numerical method used to solve our targeted PDEs, i.e., Runge-Kutta Discontinuous Galerkin (RKDG) method. While pursuing high-order accuracy, local conservativity, and high parallelizability, the RKDG method introduces additional highly nonlinear numerical flux terms to the weak formulation, on top of the existing nonlinear functions of the PDEs. Conventional Galerkin approximation methods can no longer handle these nonlinear terms efficiently. To this end, we proposed an empirical interpolation approximation method to tackle this problem. For a general parameter-dependent nonlinear function, the empirical interpolation method aims to construct an inexpensive approximation via reduced-basis space and associated set of interpolation points, known as coefficient-function approximation, which further enables efficient offline-online decomposition.

Instead of using Lagrangian bases selected by a greedy algorithm procedure for the field variable approximation, we used the POD approach which is known to concentrate the most information within the fewest number of basis vectors, at the same time, produces a basis which is already orthonormal. The empirical interpolation approximation for the nonlinear terms of the field variable
is then incorporated into our reduced-basis model.

The two numerical examples presented in Chapter 3 served two purposes: (i) The one-dimensional viscous Burger's equation, which contains low order nonlinear function, justifies the accuracy and efficiency of empirical interpolation method while handling nonlinear terms, as compared with conventional Galerkin approximation method; (ii) The two-dimensional Buckley-Leverett equation, which contains high order nonlinear functions and complicated nonlinear numerical flux terms while solved with RKDG method, can be efficiently handled by reduced-basis model incorporated with empirical interpolation approximation. A posteriori error estimators proposed for the nonlinear time-dependent parameterized convection-diffusion problems certified that our reduced-basis model could achieve the desired accuracy level.

We further extended our reduced-basis model to handle systems of equations. We targeted the quasi one-dimensional unsteady compressible Euler equations, and the numerical results verified the accuracy of the reduced-basis approximation.

5.2 Future Work

The successful development and implementation of the reduced-basis model for nonlinear time-dependent parameterized convection-diffusion equations in this thesis suggests another topic of research, i.e., to further extend the model to handle problems with higher dimensions and more degrees of freedom, for example, unsteady Navier-Stokes equation.

The two reduced-basis formulations proposed in this thesis -- function formulation and residual formulation -- are both proven to be numerically accurate, while the residual formulation is believed to be simpler in terms of implementation but less stable for certain configuration of dimensions of the reduced-basis model. This suggests another direction of improvement for future work.

So far, the choice for the size of the reduced-basis model for a particular problem is purely empirical, which might depend on many factors, for example, initial condition, geometry, parameter space, and time span. The optimal dimensions for the reduced-basis models are hardly obtainable beforehand. An interesting question is thus to find the "best" choices for $N$ and $M$ which minimize the computational cost for a desired approximation accuracy.

For complicated parameterized PDEs, it is always of great interest to solve the inverse problem. Efficient and accurate reduced-basis models make it possible and lay the groundwork for this
problem. However, a lot of practical issues still needed to be addressed.
Bibliography


62


65