Computers & Fluids 43 (2011) 74–81

Contents lists available at ScienceDirect

Computers & Fluids

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

High-fidelity real-time simulation on deployed platforms

D.B.P. Huynh[a], D.J. Knezevic[a], J.W. Peterson[b], A.T. Patera[a,*]

[a] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] Texas Advanced Computing Center, The University of Texas at Austin, Austin, TX 78758-4497, USA

ARTICLE INFO

Article history:
Received 29 April 2010
Accepted 16 July 2010
Available online 20 September 2010

Keywords:
Reduced basis method
a posteriori error bounds
Parametrized partial differential equations
High-performance computing
Deployed platforms
Real-time computing

ABSTRACT

We present a certified reduced basis method for high-fidelity real-time solution of parametrized partial differential equations on deployed platforms. Applications include *in situ* parameter estimation, adaptive design and control, interactive synthesis and visualization, and individuated product specification. We emphasize a new hierarchical architecture particularly well suited to the reduced basis computational paradigm: the expensive Offline stage is conducted pre-deployment on a parallel supercomputer (in our examples, the TeraGrid machine Ranger); the inexpensive Online stage is conducted "in the field" on ubiquitous thin/inexpensive platforms such as laptops, tablets, smartphones (in our examples, the Nexus One Android-based phone), or embedded chips. We illustrate our approach with three examples: a two-dimensional Helmholtz acoustics "horn" problem; a three-dimensional transient heat conduction "Swiss Cheese" problem; and a three-dimensional unsteady incompressible Navier–Stokes low-Reynolds-number "eddy-promoter" problem.

1. Introduction

Many engineering applications require high-fidelity real-time simulation on deployed platforms "in the field." Examples include *in situ* parameter estimation and identification procedures, embedded adaptive design and control systems, virtual reality/synthesis and visualization environments (from music to medicine), and individuated context-dependent product specification frameworks. In all these cases the mathematical model must be sophisticated, the numerical approximation must be accurate, and the response to a query must be rapid—commensurate with real-time decision or interaction requirements—despite the limited processor power and storage capacity available in the field. We shall furthermore be interested in both input–output evaluation and visualization; the latter places additional demands on memory.

We shall suppose that the system input $\mu \in \mathscr{D} \subset \mathbb{R}^P$ enters as a parameter in a partial differential equation (PDE) which describes the relevant physical phenomena over the time interval of interest $0 \leqslant t \leqslant t_f$ and the appropriate spatial domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3. This PDE, say a linear-time-invariant (LTI) parabolic equation, yields (i) a field variable over Ω, $u(t;\mu) \in X(\Omega)$ (where $X(\Omega)$ is an appropriate function space), and (ii) a scalar output of interest, $s(t;\mu) \in \mathbb{R}$, which can be expressed as a (say) linear functional of the field variable, $s(t;\mu) = \ell(u(t;\mu))$. (In actual practice we may consider many outputs.) Note that the parameter dependence proceeds from the PDE through the field variable and finally to the engineering output.

We shall distinguish between the *pre-deployment* period and the *post-deployment* or equivalently *deployed* period. The pre-deployment period takes place in the laboratory: we prepare the system and associated computational model for subsequent service. The deployed period takes place in the field: we put the system and associated computational model—now implemented on an embedded or more generally "deployed platform"—into service. In the deployed stage the computational task is well-prescribed: given a *query instance* $\mu' \in \mathscr{D}$ we wish to (a) predict the output, $\mu' \in \mathscr{D} \to (u(t^k;\mu') \to) s(t^k;\mu')$, $0 \leqslant k \leqslant K$, and (b) visualize the field, $\mu' \in \mathscr{D} \to u(t^k;\mu')|_{\mathscr{R}}$, $0 \leqslant k \leqslant K$; here $\mathscr{R} \subset \Omega$ is a region or manifold selected for rendering. (We reserve μ' to denote a query instance – a request post-deployment.) Note the field variable plays an important role both in input–output evaluation and of course in visualization.

To perform this computational task the PDE is typically discretized by a finite difference discretization in time and a finite element (FE) discretization in space. In time we consider a (say) Crank–Nicolson scheme associated to time levels $t^k = k\Delta t$, $0 \leqslant k \leqslant K$, where $\Delta t = t_f/K$; in space we consider Galerkin projection over a FE approximation subspace $X^{\mathscr{N}} (\subset X)$ of large dimension $\mathscr{N}$. Our "truth" approximation is then given, for any $\mu \in \mathscr{D}$, by $u^{\mathscr{N}}(t^k;\mu)$, $s^{\mathscr{N}}(t^k;\mu) = \ell(u^{\mathscr{N}}(t^k;\mu))$, $0 \leqslant k \leqslant K$. We note that, given our restriction to $\mu \in \mathscr{D}$, all solutions of interest perforce reside on the parametrically induced manifold $\mathscr{M}^{\mathscr{N}} \equiv \{u^{\mathscr{N}}(t^k;\mu) | 0 \leqslant k \leqslant K, \mu \in \mathscr{D}\}$. We observe that this manifold is relatively low-dimensional; we

* Corresponding author.
E-mail addresses: huynh@mit.edu (D.B.P. Huynh), dknez@mit.edu (D.J. Knezevic), peterson@tacc.utexas.edu (J.W. Peterson), patera@mit.edu (A.T. Patera).
URL: http://augustine.mit.edu/ (A.T. Patera).

0045-7930/$ - see front matter © 2011 Published by Elsevier Ltd.
doi:10.1016/j.compfluid.2010.07.007

can further anticipate, and in many cases demonstrate, that this manifold is smooth.

Our truth approximation shall provide, for sufficiently small Δt and in particular for sufficiently large $\mathcal{N}$, the desired accuracy. However, we cannot expect real-time response in particular on deployed platforms typically characterized by limited processor power and memory capacity. We thus pursue the certified reduced basis (RB) approach [1–5] *as an approximation to the truth approximation*. In time we directly inherit the Crank–Nicolson discretization of the truth; in space we consider Galerkin projection over an RB approximation space $X_N(\subset X^{\mathcal{N}})$ of small dimension N. Our RB approximation is then given, for any $\mu \in \mathcal{D}$, by $u_N(t^k;\mu)$,$s_N(t^k;\mu) = \ell(u_N(t^k;\mu))$, $0 < k \leqslant K$. We also provide rigorous *a posteriori* bounds, $\Delta_N(t^k;\mu)$ and $\Delta^s_N(t^k;\mu)$, for the error in the RB field approximation and the RB output approximation, respectively: for any $\mu \in \mathcal{D},\ \|u^{\mathcal{N}}(t^k;\mu) - u_N(t^k;\mu)\|_X \leqslant \Delta_N(t^k;\mu),\ |s^{\mathcal{N}}(t^k;\mu) - s_N(t^k;\mu)| \leqslant \Delta^s_N(t^k;\mu)$, $0 \leqslant k \leqslant K$. We may thus say that our RB approximation is *certified*.

The RB approximation space X_N is specifically designed to well approximate functions which reside on the parametrically induced manifold of interest, $\mathcal{M}^{\mathcal{N}}$: indeed, X_N is developed as the span of optimally selected (and combined) snapshots on the manifold $\mathcal{M}^{\mathcal{N}}$. (In contrast, even in a mesh-adaptive context, the truth FE approximation space $X^{\mathcal{N}}$ can represent a large class of functions very distant from $\mathcal{M}^{\mathcal{N}}$.) We can thus expect $N \ll \mathcal{N}$. The latter may in certain simple instances be proven, may in general be confirmed *a posteriori* through our error bounds, and may in practice be observed in a wide variety of problems. (Of course the "constants" will certainly depend on the particular problem under study and especially on the number of parameters, P.) This reduction in dimension *in conjunction* with an Offline–Online computational approach provides the RB advantage in the real-time deployed context. We now discuss the Offline–Online decomposition.

In the Offline stage we develop the RB space: we identify optimal (combinations of) snapshots on $\mathcal{M}^{\mathcal{N}}$ and we "precompute" various parameter-independent functionals of these snapshots implicated in subsequent RB approximations and associated RB error bounds; this Offline stage is expensive—$O(\mathcal{N}^{\gamma})$ FLOPs, where γ is a problem-dependent factor related to the computational cost of the truth solves. The Offline stage yields a (problem-dependent) Online Dataset; this dataset is small—$O(Q,N)$ data, where Q measures the parametric complexity of our PDE. In the Online stage we invoke the Online Dataset to perform rapid certified output evaluation: given any $\mu' \in \mathcal{D}$ we calculate the RB output approximation and associated RB output error bound, respectively $s_N(t^k;\mu')$ and $\Delta^s_N(t^k;\mu')$, $0 \leqslant k \leqslant K$. This Online stage is very inexpensive—$O(Q,N,K)$ FLOPs with $N \ll \mathcal{N}$. (In the next section we also discuss Online certified visualization; in this case the Online Dataset and Online operation count will depend on $\mathcal{N}$ and in particular on the number of FE degrees of freedom associated with $\mathcal{R}$. We note, however, that the visualization is a useful but optional step: the key quantities are the output(s) and associated error bound(s).)

We now associate the Offline stage to the pre-deployment period and the Online stage to the post-deployment period. The expensive Offline stage is conducted prior to deployment and hence the considerable Offline cost is not our principal concern. (Of course, control of the Offline cost is, in practice, very important; we discuss this further in the next section.) Only the inexpensive Online stage is invoked in the deployed period and hence only the very low Online cost will determine our primary performance metric—reliable and rapid response in the field. We may thus achieve our objective of high-fidelity real-time simulation on deployed platforms, as we now describe.

In the Online stage, the response to each query instance, $\mu' \in \mathcal{D} \to s_N(t^k;\mu'), \Delta^s_N(t^k;\mu'),\ 0 \leqslant k \leqslant K$, requires sufficiently few operations—$O(Q,N,K)$ FLOPs, independent of $\mathcal{N}$—and sufficiently little data—$O(Q,N)$ storage for the Online Dataset, independent of $\mathcal{N}$—to achieve *real-time* response on *deployed* ("thin") platforms. Furthermore, our rigorous error bound $\Delta^s_N(t^k;\mu')$, $0 \leqslant k \leqslant K$, will guarantee the accuracy of the RB output prediction relative to the *high-fidelity* truth. (We emphasize that the error bound does *not* require appeal to $u^{\mathcal{N}}(t^k;\mu'),\ s^{\mathcal{N}}(t^k;\mu')$.) We thus obtain not just rapid, but also accurate, optimal, and safe decisions in the field.

In Section 2 we describe, for a simple model problem, the reduced basis approach. We emphasize the computational aspects: the RB approximation and associated RB *a posteriori* error estimation "kernels"; the procedure for identification of optimal RB approximation spaces; and the Offline, Online Dataset, and Online decomposition. In Section 3 we elaborate upon the Offline and Online procedures within a hierarchical architecture: we present the Offline procedure from a parallel perspective and describe a particular implementation on the TeraGrid supercomputer Ranger at the Texas Advanced Computing Center (TACC); we present the Online procedure from a deployed/embedded perspective and describe a particular implementation on a Nexus One Android phone "model platform." In Section 4 we present results for three examples: a frequency-domain acoustics problem in a two-dimensional horn configuration Ω—to illustrate the necessity of high-fidelity PDE models and accurate numerical solutions; a transient linear heat conduction problem in a three-dimensional "Swiss Cheese" configuration Ω—to illustrate treatment of many parameters; and a transient incompressible fluid flow problem in a three-dimensional sphere-in-duct configuration Ω—to illustrate extension to (quadratic) nonlinearities.

2. Certified reduced basis formulation

2.1. Model problem

We shall illustrate the approach for a very simple model problem. We consider steady heat conduction in a (say, polygonal) domain $\Omega = \Omega^1 \cup \Omega^2$: the normalized thermal conductivity in Ω^1 (respectively, Ω^2) is unity (respectively, κ). We apply a uniform unit heat source over the entire domain Ω. We require that the temperature field, u, vanish—zero Dirichlet conditions—on the domain boundary $\partial\Omega$. We consider a single ($P = 1$) parameter: $\mu \equiv \kappa$, the conductivity in Ω^2; $\mathcal{D}$, the parameter domain, is given by (say) the interval [1,10]. We take for our output of interest, s, the integral of the temperature over Ω^1. (Note we may, for example, expand the model to include convection by a prescribed incompressible velocity field. Many other extensions are possible.)

In mathematical terms, $u(\mu) \in X$, where $X = H^1_0(\Omega)$; here $H^1_0(\Omega) = \{v \in H^1(\Omega) | v|_{\partial\Omega} = 0\},\ H^1(\Omega) = \{v \in L^2(\Omega) | \nabla v \in (L^2(\Omega))^2\}$, and $L^2(\Omega)$ is the space of square integrable functions over Ω. We associate to the space X the inner product $(w,v)_X \equiv \int_\Omega \nabla w \cdot \nabla v$ and induced norm $\|w\|_X \equiv \sqrt{(w,w)_X}$ and to the space $L^2(\Omega)$ the inner product $(w,v) \equiv \int_\Omega wv$ and induced norm $\|w\| \equiv \sqrt{(w,w)}$. We then define the continuous and coercive bilinear forms $a^1 \equiv \int_{\Omega^1} \nabla w \cdot \nabla v$, $a^2 \equiv \int_{\Omega^2} \nabla w \cdot \nabla v$, and

$$a(w,v;\mu) \equiv a^1(w,v) + \mu a^2(w,v), \quad \forall w,\ v \in X, \tag{1}$$

and the bounded linear forms $f(v) = \int_\Omega v$ and $\ell(v) = \int_{\Omega^1} v$. We can now provide the weak statement of our PDE: given $\mu \in \mathcal{D}$, find $u(\mu) \in X$ such that $a(u(\mu),v;\mu) = f(v)$, $\forall v \in X$; evaluate $s(\mu) = \ell(u(\mu))$. (We may readily accommodate several or even many outputs.)

We note that (1) is a special case of a more general hypothesis. We say that our bilinear form a is "affine in parameter" (or more precisely, "affine in functions of the parameter") if we can write

$$a(w,v;\mu) = \sum_{q=1}^{Q} \Theta^q(\mu) a^q(w,v), \tag{2}$$