Reliable Real–Time Solution of Parametrized Elliptic Partial Differential Equations: Application to Elasticity

K. Veroy, T. Leurent, C. Prud’homme, D. Rovas, and A.T. Patera

Abstract—The optimization, control, and characterization of engineering components or systems require fast, repeated, and accurate evaluation of a partial-differential-equation-induced input–output relationship. We present a technique for the rapid and reliable prediction of linear–functional outputs of elliptic partial differential equations with affine parameter dependence. The method has three components: (i) rapidly convergent reduced–basis approximations; (ii) a posteriori error estimation; and (iii) off-line/on–line computational procedures. These components — integrated within a special network architecture — render partial differential equation solutions truly “useful”; essentially real–time as regards operation count; “blackbox” as regards reliability; and directly relevant as regards the (limited) input–output data required.

Keywords—reduced–basis, a posteriori error estimation, output bounds, elliptic partial differential equations, distributed simulations, real–time computing

I. Introduction

The optimization, control, and characterization of an engineering component or system requires the prediction of certain “quantities of interest,” or performance metrics, which we shall denote outputs — for example, deflections, maximum stresses, maximum temperatures, heat transfer rates, flowrates, or lift and drags. These outputs are typically expressed as functionals of field variables associated with a parametrized partial differential equation which describes the physical behavior of the component or system. The parameters, which we shall denote inputs, serve to identify a particular “configuration” of the component: these inputs may represent design or decision variables, such as geometry — for example, in optimization studies; control variables, such as actuator power — for example, in real–time applications; or characterization variables, such as physical properties — for example, in inverse problems. We thus arrive at an implicit input–output relationship, evaluation of which demands solution of the underlying partial differential equation.

Our goal is the development of computational methods that permit rapid and reliable evaluation of this partial-differential-equation-induced input–output relationship in the limit of many queries — that is, in the design, optimization, control, and characterization contexts. Our particular approach is based upon the reduced–basis method, first introduced in the late 1970s for nonlinear structural analysis [1], [10], and subsequently developed more broadly in the 1980s and 1990s [3], [5], [12], [13], [17]. Our work differs from these earlier efforts in several important ways: first, we develop (in some cases, provably) global approximation spaces; second, we introduce rigorous a posteriori error estimators; and third, we exploit off–line/on–line computational decompositions. These three ingredients allow us — for a restricted but important class of problems — to reliably decouple the generation and projection stages of reduced–basis approximation, thereby effecting computational economies of several orders of magnitude.

In this paper, we discuss these components in the context of symmetric coercive problems. We begin in Section II by introducing an abstract problem formulation and an illustrative instantiation: a microtruss problem. In Section III we describe the reduced-basis approximation; and in Section IV we develop associated a posteriori error estimators. Finally, in Section V we discuss the system architecture in which these numerical objects reside.

II. Problem Statement

A. Abstract Formulation

We consider a suitably regular domain Ω ⊂ IR^d, d = 1, 2, or 3, and associated function space X ⊂ H^1(Ω), where H^1(Ω) = \{v | v ∈ L^2(Ω), ∇v ∈ (L^2(Ω))^d\}, and L^2(Ω) is the space of square integrable functions over Ω. The inner product and norm associated with X are given by ( · , · )_X and ∥ · ∥_X = (· , ·)_X^{1/2}, respectively. We also define a parameter set D ⊂ IR^p, a particular point in which will be denoted µ. Note that Ω does not depend on the parameter.

Our abstract problem may be stated as: for any parameter µ ∈ D, find s(µ) ∈ IR given by

s(µ) = ℓ(u(µ)),

where ℓ(v) is a linear functional, and u(µ) ∈ X satisfies the partial differential equation (in weak form)

a(u(µ), v; µ) = f(v), \quad \forall v ∈ X.  \hspace{1cm} (2)

We assume that the bilinear form a(· , · ; µ) is symmetric,

a(w, v; µ) = a(v, w; µ), \quad \forall w, v ∈ X; \hspace{1cm} (3)
we first present the compliance case before

\begin{align}
    a(w, v; \mu) &\leq \gamma(\mu) \|w\|_X \|v\|_X \quad (4) \\
    &\leq \gamma_0 \|w\|_X \|v\|_X, \quad \forall \mu \in D; \quad (5)
\end{align}

and coercive,

\begin{align}
    0 < \alpha_0 \leq \alpha(\mu) = \inf_{w \in X} \frac{a(w, w; \mu)}{\|w\|_X^2}, \quad \forall \mu \in D; \quad (6)
\end{align}

we also require that the linear functionals \( f(v) \) and \( \ell(v) \) are bounded. The choice of output \( \ell(v) = f(v) \) — termed compliance — considerably simplifies the formulation; in Sections III and IV we first present the compliance case before proceeding to the noncompliance case (i.e., \( \ell(v) \neq f(v) \)).

We shall also make certain assumptions on the parametric dependence of \( a, f, \) and \( \ell \). In particular, we shall suppose that, for some finite (preferably small) integer \( Q \), \( a \) may be expressed as

\begin{equation}
    a(w, v; \mu) = \sum_{q=1}^{Q} \sigma^q(\mu) a^q(w, v), \quad (7)
\end{equation}

\( \forall w, v \in X, \forall \mu \in D, \) for appropriately chosen functions \( \sigma^q \) and associated \( \mu \)-independent bilinear forms \( a^q, q = 1, \ldots, Q \). For simplicity of exposition, we assume that \( f \) and \( \ell \) do not depend on \( \mu \); in actual practice, affine dependence is readily admitted.

We note that we pose our problem on a fixed reference domain \( \Gamma \) (i.e., \( \Gamma \) does not depend on \( \mu \)) to ensure that the parametric dependence on geometry enters through \( a(\cdot, \cdot; \mu) \) and ultimately through the \( \sigma^q(\mu) \).

B. Motivation: A Particular Instantiation

To further motivate and illustrate our methods we consider a particular example: a truss structure. Truss structures are used in a variety of applications such as actuation (see, for example, [9]) and multifunctional components (see, for example, [3], [19]). Truss structures are typically designed for minimum weight subject to certain strength constraints; that is, the structure must be able to support the prescribed loads without buckling, yielding, or undergoing excessive deformation.

We consider here the prismatic truss structure shown in Figure 1 which consists of a frame (upper and lower faces, in dark gray) and a core (trusses and middle sheet, in light gray). The structure transmits a force per unit depth \( \bar{F} \) uniformly distributed over the tip of the middle sheet, \( \bar{\Gamma}_2 \), through the truss system to the fixed left wall, \( \bar{\Gamma}_0 \). The physical model is simple plane-strain (two-dimensional) linear elasticity: the displacement field \( \bar{u}_i(\mu), i = 1, 2 \), satisfies

\begin{equation}
    \int_{\bar{\Omega}} \frac{\partial \bar{u}_i}{\partial \bar{x}_j} \bar{E}_{ijkl} \frac{\partial \bar{u}_k}{\partial \bar{x}_l} = - \left( \frac{\bar{F}}{\bar{t}_c} \right) \int_{\bar{\Gamma}_2} \bar{v}_2, \quad \forall v \in \bar{X}, \quad (8)
\end{equation}

where \( \bar{\Omega}(\mu) \) is the truss domain, \( \bar{E}_{ijkl} \) is the elasticity tensor, \( \bar{t}_c \) is the (dimensional) thickness of the core sheet, and \( \bar{X} \) refers to the set of functions in \( H^1(\bar{\Omega}(\mu)) \) which vanish on \( \bar{\Gamma}_0(\mu) \). We assume summation over repeated indices.

We now \( (i) \) nondimensionalize the weak equations (8), and \( (ii) \) apply a continuous piecewise-affine transformation to map \( \bar{\Omega}(\mu) \) to a fixed (\( \mu \)-independent) reference domain \( \Omega \). The abstract problem statement (2) is then recovered [18]. It is readily verified that \( a \) is continuous, coercive, and symmetric; and that the “affine” assumption (7) obtains for \( Q = 44 \).

The truss structure is characterized by four design parameters, or “inputs,” \( \mu \in D \subset \mathbb{R}^{P=4} \), where \( \mu^1 = t_f \), \( \mu^2 = t_t \), \( \mu^3 = H \), and \( \mu^4 = \theta \). Here \( t_f \) and \( t_t \) are the thicknesses of the frame and trusses, respectively; \( H \) is the total height of the structure; and \( \theta \) is the angle between the trusses and the faces; \( t_f, t_t, \) and \( H \) are normalized relative to the core thickness. The design or parameter set is given by \( D = [0.08, 1.0] \times [0.2, 2.0] \times [4.0, 10.0] \times [30.0^\circ, 60.0^\circ] \). The material properties — Poisson’s ratio, \( \nu = 0.3 \); the frame Young’s modulus, \( E_f = 75 \) GPa; and the core Young’s modulus, \( E_c = 200 \) GPa — are held fixed.

We choose our performance metrics, or “outputs,” to be \( (i) \) the average downward deflection (compliance) at the core tip, \( \Gamma_3 \), nondimensionalized by \( \bar{F}/\bar{E}_{ijkl} \); and \( (ii) \) the average normal stress across the critical (yield) section \( \Gamma_3^i \) in Figure 1, nondimensionalized by \( \bar{F}/\bar{t}_c \). We denote our outputs as \( s^1(\mu) = \ell^1(u(\mu)) \) and \( s^2(\mu) = \ell^2(u(\mu)) \), respectively, where

\begin{align}
    \ell^1(v) &= - \int_{\Gamma_3} v_2, \\
    \ell^2(v) &= \frac{1}{t_f} \int_{\Gamma_3} \frac{\partial \chi_i}{\partial x_j} E_{ijkl} \frac{\partial u_k}{\partial x_l} \quad (10)
\end{align}

are bounded linear functionals; here \( \chi_i \) is any suitably smooth function in \( H^1(\Gamma^p) \) such that \( \chi_i \bar{n}_i = 1 \) on \( \Gamma_2^i \) and \( \chi_i \bar{n}_i = 0 \) on \( \Gamma_2^j \), where \( \bar{n} \) is the unit normal. Note that \( s^1(\mu) \) is a compliant output, whereas \( s^2(\mu) \) is “noncompliant”.

III. Reduced Basis Approach

We discuss in this section the formulation, properties, and computational realization of the reduced basis method for both compliant and noncompliant outputs.
A. Compliant Outputs

A.1 Reduced-Basis Approximation

To define our reduced-basis procedure, we first introduce a sample set in parameter space, \( S_N = \{ \mu_1, \ldots, \mu_N \} \), where \( \mu_i \in \mathcal{D}, \ i = 1, \ldots, N \). We then define our reduced-basis approximation space as

\[
W_N = \text{span} \{ \zeta_n \equiv u(\mu_n), \ n = 1, \ldots, N \},
\]

where \( u(\mu_n) \in X \) is the solution to (2) for \( \mu = \mu_n \). In actual practice, \( u(\mu_n) \) is replaced by a finite element approximation on a suitably fine truth mesh. For any \( \mu \in \mathcal{D} \), our reduced-basis approximation \( u_N(\mu) \in W_N \) satisfies

\[
a(u_N(\mu), v; \mu) = f(v), \ \forall \ v \in W_N; \quad (12)
\]

we then evaluate our reduced-basis output approximation as

\[
s_N(\mu) = f(u_N(\mu)).
\]

Recall that compliance implies \( \ell(v) = f(v) \).

A.2 A Priori Convergence Theory

It can readily be shown that our approximation \( u_N(\mu) \) is optimal in the sense that

\[
|||u(\mu) - u_N(\mu)||| \leq \min_{w_N \in W_N} ||u(\mu) - w_N||
\]

where \( ||| \cdot (\mu)||| = a(\cdot, \cdot; \mu) \) is the energy norm. Furthermore, for our compliant output,

\[
s(\mu) = s_N(\mu) + f(u - u_N) \\
= s_N(\mu) + a(u, u - u_N; \mu) \\
= s_N(\mu) + a(u - u_N, u - u_N; \mu)
\]

from symmetry and Galerkin orthogonality. It follows that \( s(\mu) - s_N(\mu) \) converges as the square of the error in the best approximation and, from coercivity, that \( s_N(\mu) \) is a lower bound for \( s(\mu) \).

It now remains to bound the dependence of the error in the best approximation as a function of \( N \). It can be proven [9] — although only for special problems — that if the \( \mu_n, \ n = 1, \ldots, N \), are logarithmically distributed over \( \mathcal{D} \), then:

\[
|||u(\mu) - u_N(\mu)||| \leq c_1 e^{-c_2 N}, \ \forall \ \mu \in \mathcal{D},
\]

where \( c_2 \) depends only weakly on the range of the parameter. That is, the best approximation \( u_N(\mu) \) converges to the the exact solution \( u(\mu) \) exponentially.

We present in Table I the error \( |s(\mu) - s_N(\mu)|/s(\mu) \) as a function of \( N \), at a particular representative point \( \mu \in \mathcal{D} \), for the truss problem of Section II-B. Since tensor-product grids are prohibitively exorbitant as \( P \) increases, the \( \mu_n \) are chosen “log-randomly” over \( \mathcal{D} \); we sample from a multivariate uniform probability density on \( \log(\mu) \). We observe, as the theory suggests, that the error is remarkably small even for very small \( N \), and that very rapid convergence obtains as \( N \to \infty \). In numerous numerical tests [18], the logarithmic distribution suggested by theory performs considerably better than other obvious candidates, in particular for large ranges of the parameter. For instance, the results presented in Table II — for the same point \( \mu \) in \( \mathcal{D} \), but based on a (non–log) uniform random point distribution — exhibit slower convergence than that for the logarithmic distribution of Table I.

A.3 Computational Procedure

The theoretical and empirical results of Sections III-A.1 and III-A.2 suggest that \( N \) may, indeed, be chosen very small. We now develop off-line/on-line computational procedures that exploit this dimension reduction.

We first express the reduced-basis approximation \( u_N(\mu) \) as

\[
u_N(\mu) = \sum_{j=1}^{N} u_N j(\mu) \zeta_j = (u_N(\mu))^T \zeta,
\]

where \( u_N(\mu) \in \mathbb{R}^N \); we then choose for test functions \( v = \zeta_i, \ i = 1, \ldots, N \). Inserting these representations into (12) yields the desired algebraic equations for \( u_N(\mu) \in \mathbb{R}^N \),

\[
\Delta_N(\mu) u_N(\mu) = F_N,
\]

in terms of which the output can then be evaluated as

\[
s_N(\mu) = F_N^T u_N(\mu).
\]

- **Table I**

| \( N \) | \( |s(\mu) - s_N(\mu)|/s(\mu) \) | \( \Delta_N(\mu)/s(\mu) \) | \( \eta_N(\mu) \) |
|---|---|---|---|
| 10 | 3.26 × 10^{-2} | 6.47 × 10^{-2} | 1.98 |
| 20 | 2.56 × 10^{-4} | 4.74 × 10^{-4} | 1.85 |
| 30 | 7.31 × 10^{-5} | 1.38 × 10^{-4} | 1.89 |
| 40 | 1.91 × 10^{-5} | 3.59 × 10^{-5} | 1.88 |
| 50 | 1.09 × 10^{-5} | 2.08 × 10^{-5} | 1.90 |
| 60 | 4.10 × 10^{-6} | 8.19 × 10^{-6} | 2.00 |

- **Table II**

| \( N \) | \( |s(\mu) - s_N(\mu)|/s(\mu) \) | \( \Delta_N(\mu)/s(\mu) \) | \( \eta_N(\mu) \) |
|---|---|---|---|
| 10 | 1.91 × 10^{-2} | 2.56 × 10^{-2} | 1.54 |
| 20 | 6.29 × 10^{-3} | 1.21 × 10^{-2} | 1.93 |
| 30 | 1.56 × 10^{-3} | 2.94 × 10^{-3} | 1.88 |
| 40 | 2.19 × 10^{-4} | 4.00 × 10^{-4} | 1.83 |
| 50 | 1.45 × 10^{-5} | 2.64 × 10^{-4} | 1.82 |
| 60 | 9.17 × 10^{-5} | 1.72 × 10^{-4} | 1.87 |
Here $A_N(\mu) \in \mathbb{R}^{N \times N}$ is the SPD matrix with entries $A_{N,i,j}(\mu) \equiv a(\zeta_j, \zeta_i; \mu)$, $i, j = 1, \ldots, N$, and $F_N \in \mathbb{R}^N$ is the “load” (and “output”) vector with entries $F_N_i \equiv f(\zeta_i)$, $i = 1, \ldots, N$.

We now invoke (7) to write

$$A_{N,i,j}(\mu) = a(\zeta_j, \zeta_i; \mu) = \sum_{q=1}^{Q} \sigma^q(\mu) a^q(\zeta_j, \zeta_i),$$

or

$$A_N(\mu) = \sum_{q=1}^{Q} \sigma^q(\mu) A^q_N,$$

where $A^q_{N,i,j} = a^q(\zeta_j, \zeta_i)$, $i, j = 1, \ldots, N$, $q = 1, \ldots, Q$. The off-line/on-line decomposition is now clear. In the off-line stage, we compute the $u(\mu)$ and form the $A^q_N$ and $F_N$: this requires $N$ (expensive) “a” finite element solutions and $O(QN^2)$ finite-element-vector inner products. In the on-line stage, for any given new $\mu$, we first form $A_N$ from (21), then solve (18) for $u_N(\mu)$, and finally evaluate $s_N(\mu) = F_N^T u_N(\mu)$: this requires $O(QN^2) + O(\frac{3}{2}N^3)$ operations and $O(QN^2)$ storage.

Thus, as required, the incremental, or marginal, cost to evaluate $s_N(\mu)$ for any given new $\mu$ is very small: first, because $N$ is very small, typically $O(10)$ — thanks to the good convergence properties of $W_N$; and second, because (18) can be very rapidly assembled and inverted — thanks to the affine dependence of $a$ on $\mu$ and the associated off-line/on-line decomposition (see [2] for an earlier application of this strategy within the reduced-basis context). For the problems discussed here and (for instance) in [14], the resulting computational savings relative to standard (well-designed) finite-element approaches are significant — at least $O(10)$, typically $O(100)$, and often $O(1000)$ or more.

B. Noncompliant Outputs

In the previous section we formulate the reduced-basis method for the case of compliant outputs, $\ell(v) = f(v)$, $\forall v \in X$. We briefly summarize here the formulation and theory for more general linear bounded output functionals.

As a preliminary, we first generalize the abstract formulation of Section II-A. As before, we define the “primal” problem as in (2). However, we now also introduce an associated adjoint or “dual” problem: for any $\mu \in X$, find $\psi(\mu) \in X$ such that

$$a(v, \psi(\mu); \mu) = \ell(v), \quad \forall v \in X;$$

recall that $\ell(v)$ is our output functional.

B.1 Reduced-Basis Approximation

To develop the reduced-basis space, we first choose — randomly or log-randomly as described in Section III-A.2 — a sample set in parameter space, $S_{N/2} = \{\mu_1, \ldots, \mu_{N/2}\}$, where $\mu_i \in \mathcal{D}$, $i = 1, \ldots, N/2$ ($N$ even); we then define an “integrated” reduced-basis approximation space as $W_N = \text{span} \{ (u(\mu_n), \psi(\mu_n)) \}$, $n = 1, \ldots, N/2$ (see [14] for discussion of a somewhat more efficient “non-integrated” approach).

For any $\mu \in \mathcal{D}$, our reduced-basis approximation is then obtained by standard Galerkin projection onto $W_N$. To wit, for the primal problem, we find $u_N(\mu) \in W_N$ such that $a(u_N(\mu), v; \mu) = \ell(v)$, $\forall v \in W_N$; and for the adjoint problem, we define (though, for the particular formulations described here, do not compute) $\psi_N(\mu) \in W_N$ such that $a(v, \psi_N(\mu); \mu) = \ell(v)$, $\forall v \in W_N$. The reduced-basis output approximation is then calculated from $s_N(\mu) = \ell(u_N(\mu))$.

B.2 A Priori Convergence Theory

Turning now to the $a$ priori theory, it follows from standard $u_N(\mu)$ and $\psi_N(\mu)$ are optimal in the sense that

$$||u(\mu) - u_N(\mu)|| \leq \min_{w_N \in W_N} ||u(\mu) - w_N||,$$

$$||\psi(\mu) - \psi_N(\mu)|| \leq \min_{w_N \in W_N} ||\psi(\mu) - w_N||.$$
we know that, in theory, we can obtain error estimators for cases, $N$ can be chosen: this will depend on the desired accuracy, the latter in unacceptable uncertainty. We thus need uncertainty, either too many or too few basis functions will be obtained $s_N(\mu)$ very inexpensively: the on-line stage scales as $O(N^3) + O(N^2)$; and $N$ can, in theory, be chosen quite small. However, in practice, we do not know how small $N$ can be chosen: this will depend on the desired accuracy, the selected output(s) of interest, and the particular problem in question; in some cases $N = 5$ may suffice, while in other cases, $N = 100$ may still be insufficient. In the face of this uncertainty, either too many or too few basis functions will be retained: the former results in computational inefficiency; the latter in unacceptable uncertainty. We thus need a posteriori error estimators for $s_N$.

We present in [7], [8], [11], [14] and [15] an approach to output error estimation which guarantees rigorous error bounds. We present here an alternative method which provides greater computational efficiency, albeit at the loss of complete certainty. As in Section III, we first discuss the method for the compliant case, then extend the formulation to the noncompliant case.

| $N$ | $|s(\mu) - s_N(\mu)|/s(\mu)$ | $\Delta_N(\mu)/s(\mu)$ | $\eta_N(\mu)$ |
|-----|-----------------------------|-------------------------|--------------|
| 20  | $2.35 \times 10^{-2}$       | $4.67 \times 10^{-4}$   | 1.99         |
| 40  | $1.74 \times 10^{-4}$       | $3.19 \times 10^{-4}$   | 1.83         |
| 60  | $5.59 \times 10^{-5}$       | $1.06 \times 10^{-4}$   | 1.90         |
| 80  | $1.44 \times 10^{-5}$       | $2.73 \times 10^{-5}$   | 1.89         |
| 100 | $7.45 \times 10^{-6}$       | $1.40 \times 10^{-5}$   | 1.88         |
| 120 | $2.92 \times 10^{-6}$       | $5.85 \times 10^{-6}$   | 2.00         |

TABLE III
Error, error bound, and effectivity as a function of $N$, at a particular representative point $\mu \in D$, for the truss problem (noncompliant output) and a logarithmic distribution.

| $N$ | $|s(\mu) - s_N(\mu)|/s(\mu)$ | $\Delta_N(\mu)/s(\mu)$ | $\eta_N(\mu)$ |
|-----|-----------------------------|-------------------------|--------------|
| 20  | $1.52 \times 10^{-2}$       | $2.06 \times 10^{-4}$   | 1.36         |
| 40  | $4.90 \times 10^{-3}$       | $9.42 \times 10^{-3}$   | 1.92         |
| 60  | $1.31 \times 10^{-3}$       | $2.46 \times 10^{-3}$   | 1.88         |
| 80  | $1.90 \times 10^{-4}$       | $3.50 \times 10^{-4}$   | 1.84         |
| 100 | $1.12 \times 10^{-4}$       | $2.07 \times 10^{-4}$   | 1.84         |
| 120 | $7.82 \times 10^{-5}$       | $1.48 \times 10^{-4}$   | 1.89         |

TABLE IV
Error, error bound, and effectivity as a function of $N$, at a particular representative point $\mu \in D$, for the truss problem (noncompliant output) and a uniform distribution.

the noncompliant output associated with the truss problem of Section II-B. We again observe the rapid convergence as $N \to \infty$, and the superior performance of the logarithmic point distribution.

IV. A Posteriori Error Estimation: Output Bounds

From Section III we know that, in theory, we can obtain $s_N(\mu)$ very inexpensively: the on-line stage scales as $O(N^3) + O(N^2)$; and $N$ can, in theory, be chosen quite small. However, in practice, we do not know how small $N$ can be chosen: this will depend on the desired accuracy, the selected output(s) of interest, and the particular problem in question; in some cases $N = 5$ may suffice, while in other cases, $N = 100$ may still be insufficient. In the face of this uncertainty, either too many or too few basis functions will be retained: the former results in computational inefficiency; the latter in unacceptable uncertainty. We thus need a posteriori error estimators for $s_N$.

We present in [7], [8], [11], [14] and [15] an approach to output error estimation which guarantees rigorous error bounds. We present here an alternative method which provides greater computational efficiency, albeit at the loss of complete certainty. As in Section III, we first discuss the method for the compliant case, then extend the formulation to the noncompliant case.

A. Compliant Outputs

A.1 Formulation

To begin, we set $M > N$, and introduce a parameter sample $S_M = \{\mu_1, \ldots, \mu_M\}$ and associated reduced–basis approximation space $W_M = \text{span} \{\zeta_m \equiv u(\mu_m), m = 1, \ldots, M\}$; both for theoretical and practical reasons we require $S_N \subset S_M$ and therefore $W_N \subset W_M$. The procedure is simple: we first find $u_M(\mu) \in W_M$ such that $a(u_M(\mu), v; \mu) = f(v), \forall v \in W_M$; we then evaluate $s_M(\mu) = f(u_M(\mu))$; and, finally, we compute our upper and lower output estimators as

$$s_{N,M}^+ = \bar{s}(\mu) \pm \frac{1}{2} \Delta_{N,M}(\mu), (23)$$

where $\bar{s}$ is the improved predictor and $\frac{1}{2} \Delta_{N,M}(\mu)$ is the half-bound gap; $\bar{s}$ and $\Delta_{N,M}(\mu)$ are given by

$$\bar{s}(\mu) = s_N(\mu) + \frac{1}{2} \Delta_{N,M}(\mu), (24)$$

$$\Delta_{N,M}(\mu) = \frac{1}{\tau}(s_M(\mu) - s_N(\mu)) (25)$$

for some $\tau \in (0, 1)$. For our purposes here, we shall consider $M = 2N$.

A.2 Properties

We consider in this section the validity of our lower and upper estimators, and the sharpness of our output bound gap $\Delta_{N,M}(\mu)$.

We would like to prove the lower and upper effectivity inequalities

$$1 \leq \eta_{N,2N}(\mu) \leq \text{Const}, (26)$$

where the effectivity of the approximation, $\eta$, is defined as

$$\eta_{N,2N}(\mu) = \frac{\Delta_{N,2N}(\mu)}{s(\mu) - s_N(\mu)}; (27)$$

the lower effectivity inequality ensures bounds; the upper effectivity inequality ensures sharp bounds. In fact, we will only be able to demonstrate an asymptotic form of this inequality; and furthermore, we shall require the hypothesis that

$$\varepsilon_{N,2N}(\mu) \equiv \frac{s(\mu) - s_{2N}(\mu)}{s(\mu) - s_N(\mu)} \to 0 \quad \text{as } N \to \infty. (28)$$

This assumption is certainly plausible: if our a priori bound of (16) in fact reflects asymptotic behavior, then $s(\mu) - s_N(\mu) \sim c_1 e^{\gamma_2 N}$, $s(\mu) - s_{2N}(\mu) \sim c_1 e^{2\gamma_2 N}$, and hence $\varepsilon_{N,2N}(\mu) \sim e^{\gamma_2 N}$, as desired.

We can then prove the lower effectivity inequality (bounding property): $s_{N,2N}(\mu) \leq s(\mu) \leq s_{N,2N}(\mu)$, as $N \to \infty$. To prove the lower bound we again appeal to (15) and the coercivity of $a$; indeed, this result (still) obtains for all $N$. To
demonstrate the right inequality, we write
\[
\begin{align*}
\eta_{N,2N}(\mu) &= \frac{1}{\tau} \frac{s_{2N}(\mu) - s_N(\mu)}{s(\mu) - s_N(\mu)} \\
&= \frac{1}{\tau} \frac{(s_{2N}(\mu) - s_N(\mu))}{s(\mu) - s_N(\mu))} \\
&= \frac{1}{\tau} (1 - \varepsilon_{N,2N}(\mu)).
\end{align*}
\] (33)

From the definitions of \(\eta_{N,2N}(\mu), s_{N,2N}(\mu)\) and \(\varepsilon_{N,2N}(\mu)\), we directly obtain
\[
\eta_{N,2N}(\mu) = \frac{1}{\tau} \frac{s_{2N}(\mu) - s_N(\mu)}{s(\mu) - s_N(\mu)}.
\] (31)

We now prove the upper effectivity inequality (sharpness property). From the definitions of \(\eta_{N,2N}(\mu), s_{N,2N}(\mu)\) and \(\varepsilon_{N,2N}(\mu)\), we directly obtain
\[
\begin{align*}
\eta_{N,2N}(\mu) &= \frac{1}{\tau} \frac{s_{2N}(\mu) - s_N(\mu)}{s(\mu) - s_N(\mu)} \\
&= \frac{1}{\tau} \frac{(s_{2N}(\mu) - s_N(\mu))}{s(\mu) - s_N(\mu))} \\
&= \frac{1}{\tau} (1 - \varepsilon_{N,2N}(\mu)).
\end{align*}
\] (32)

\(\eta_{N,2N}(\mu)\) is bounded from above by \(1/\tau\) for all \(N\). We can also readily be shown that \(\eta_{N,2N}(\mu)\) is non-negative: since \(W_N \subset W_{2N}\), it follows from (14), (15), and standard variational arguments that \(s(\mu) \geq s_{2N}(\mu) \geq s_N(\mu)\). We thus conclude that \(0 \leq \eta_{N,2N}(\mu) \leq 1/\tau\) for all \(N\). Furthermore, from our hypothesis on \(\varepsilon_{N,2N}(\mu)\), (28), we know that \(\eta_{N,2N}(\mu)\) will tend to \(1/\tau\) as \(N\) increases.

The essential approximation enabler is exponential convergence: we obtain bounds even for rather small \(N\) and relatively large \(\tau\). We thus achieve both “near” certainty and good effectivities. We demonstrate this claim in Tables I and II in which we present the bound gap and effectivity for our truss example of Section II-B; the results tabulated correspond to the choice \(\tau = 1/2\). We clearly obtain bounds for all \(N\); and we observe that \(\eta_{N,2N}(\mu)\) does, indeed, rather quickly approach \(1/\tau\), particularly for the logarithmic (random) distribution.

A.3 Computational Procedure

Since the error bounds are based entirely on evaluation of the output, we can directly adapt the off-line/on-line procedure of Section III-A.3. Note that the calculation of the output approximation \(s_N(\mu)\) and the output bounds are now integrated: \(A_N(\mu)\) and \(L_N(\mu)\) (yielding \(s_N(\mu)\)) are a sub-matrix and sub-vector of \(A_{2N}(\mu)\) and \(L_{2N}(\mu)\) (yielding \(s_{2N}(\mu)\)), \(\Delta_{N,2N}(\mu)\) and \(s_{N,2N}(\mu)\) respectively. The on-line effort for this predictor/error estimator procedure (based on \(s_N(\mu)\) and \(s_{2N}(\mu)\)) will require eightfold more operations than the predictor procedure of Section III.

The essential computation enabler is again exponential convergence, which permits us to choose \(M = 2N\) — hence controlling the additional computational effort attributable to error estimation — while simultaneously ensuring that \(\varepsilon_{N,2N}(\mu)\) tends rapidly to zero. Exponential convergence also ensures that the cost to compute both \(s_N(\mu)\) and \(s_{2N}(\mu)\) is “negligible”. In actual practice, since \(s_{2N}(\mu)\) is available, we can of course take \(s_{2N}(\mu)\), rather than \(s_N(\mu)\), as our output prediction; this greatly improves not only accuracy, but also certainty — \(\Delta_{N,2N}(\mu)\) is almost surely a bound for \(s(\mu) - s_{2N}(\mu)\), albeit an exponentially conservative bound as \(N\) tends to infinity.

We note that the computational complexity and storage for this method increases like \(O(Q)\); this represents a factor of \(Q \) reduction in expense compared to our earlier approaches [7], [8], [11], [14] and [15]. This approach is therefore particularly advantageous in cases in which \(Q\) is large.

B. Noncompliant Outputs

We briefly discuss here the extension of the method of Section IV-A to noncompliant outputs. As in Section III-B, we begin by setting \(M > N\), \(M\) even, and introduce a parameter sample \(\hat{S}_{M/2} = \{\mu_1, \ldots, \mu_{M/2}\}\) and associated integrated reduced-basis approximation space \(W_M = \text{span} \{u(\mu_n), \psi(\mu_n), n = 1, \ldots, M\}\). We first find \(u_M(\mu) \in W_M\) such that \(\hat{a}(u_M(\mu), v, \mu) = f(v), \forall v \in W_M\); we then evaluate \(s_M(\mu) = \ell(u_M(\mu))\); and finally, we compute our upper and lower output estimators as
\[
\begin{align*}
s_{N,M}^+(\mu) = \bar{s}_N(\mu) &\pm \frac{1}{2} \Delta_{N,M}(\mu),
\end{align*}
\] (34)
where
\[
\bar{s}_N(\mu) = s_N(\mu) + \frac{1}{2\tau} (s_M - s_N)
\] (35)
\[
\Delta_{N,M}(\mu) = \frac{1}{\tau} |s_M(\mu) - s_N(\mu)|
\] (36)
and \(\tau \in (0, 1)\). The effectivity of the approximation is defined as
\[
\eta_{N,M}(\mu) = \frac{\Delta_{N,M}(\mu)}{|s(\mu) - s_N(\mu)|}.
\] (37)

We shall again only consider \(M = 2N\).

As before, can prove that
\[
1 \leq \eta_{N,2N}(\mu) \leq \text{Const} \quad \text{as} \quad N \rightarrow \infty.
\] (38)

The proof parallels that in Section IV-A.2. In particular, it can again be shown [14] that
\[
\eta_{N,2N}(\mu) \rightarrow \frac{1}{\tau} \quad \text{as} \quad N \rightarrow \infty;
\] (39)
however, \(\eta_{N,2N}\) is no longer strictly bounded from above by \(1/\tau\).

We present in Tables III (logarithmic distribution) and IV (uniform distribution) the error, bound gap, and effectivity
for the noncompliant output of the truss example of Section II-B; the results tabulated correspond to the choice \( \tau = 1/2 \). We clearly obtain bounds for all \( N \); and the effectivity ratio quickly approaches \( 1/\tau \) (in particular, for the logarithmic (random) distribution, \( \eta_{N,2N} \) remains fixed at \( 1/\tau = 2.0 \) for \( N \geq 120 \)).

V. System Architecture

A. Introduction

The numerical methods proposed are rather unique relative to more standard approaches to partial differential equations. Reduced–basis output bound methods — in particular the global approximation spaces, \( a \ posteriori \) error estimators, and off-line/on–line computational decomposition — are intended to render partial–differential-equation solutions truly “useful”: essentially real–time as regards operation count; “blackbox” as regards reliability; and directly relevant as regards the (limited) input–output data required.

But to be truly useful, the methodology — in particular the inventory of on–line codes — must reside within a special framework. This framework must permit a User to specify — within a native applications context — the problem, output, and input value of interest; and to receive — quasi–instantaneously — the desired prediction and certificate of fidelity (error bound). We describe such a (fully implemented, fully functional) framework here: we focus primarily on the User point of view; see [16] for a more detailed description of the technical foundations and ingredients.

B. Overview of Framework

We show in Figure 4 a virtual schematic of the framework. The framework comprises three key components: the Network, Client software, Server software, and Directory Service. Each User interacts with the system through a selected Client (interface) which resides, say, on the User’s Computer; we shall describe briefly below two Clients. Based on directives from the User, the Client broadcasts over the Network a Problem Label–Output Label Pair (e.g., Truss–Deflection). This Pair is received by the Directory Service — a White Pages — which informs the Client of the Simulation Resource Locator “SRL” — the physical location on a particular Computer — of a Server which can respond to the request. The Client then sends the Input (\( \mu P \)-tuple) Value to the designated SRL. The Server — essentially a suite of on–line codes and associated input–output utilities — is awaiting queries at all times; upon receipt of the Input it executes the on–line code for the designated Output Label and Input Value, and responds to the Client with the Output Value (\( s_N \) and the Half Bound Gap (\( \frac{1}{2} \Delta_N \)). The Client then displays or acts upon this information, and the cycle is complete.

Typically many identical (as well as different) Servers will be available, typically on many different Computers: there are multiple instances of the on–line codes. The Directory Service indicates to the Client the least busy Server so as to provide the fastest response possible. In some cases Clients may issue several Input Values — that is, \( L \) sets of \( P \)-tuples. In this case the Directory Service will distribute the calculations over multiple (e.g., as many as \( L \)) Servers — in particular, Servers on multiple Computers — so as to respond more quickly to this multiple–input query.

Our framework is clearly an example of “grid” computing, similar to GLOBUS, NetSolve, and Seti@HOME, to name but a few. Indeed, we exploit several generic tools upon which grid and network computing applications may be built; for example, we appeal to CORBA\(^1\) (standardized by OMG\(^2\)) to seamlessly manipulate the Server software as if it resided on the Client Computer. We remark that our reduced–basis output bound application is particularly well-suited to grid computing: the computational load on participating Computers (on which the Servers reside) is very light; and the Client–Server input/output load on the Network is very light. The network computing paradigm also serves very well the archival, collaboration, and integration aspects of standardized input–output objects.

C. Clients

We briefly describe here two Clients: SimTeX, which is a PDF–based “dynamic text” interface for interrogation, exploration, and display; and SimLAB, which is a MATLAB–based “mathematical” interface for manipulation and integration.

C.1 SimTeX

SimTeX combines several standardized tools so as to provide a very simple interface by which to access the Servers. It consists of an authoring component, a display and interface component, and an “intermediary” component. A particularly nice feature of SimTeX is the natural context which it provides — in essence, defining the input–output relationship and problem definition in the language of the application. The SimTeX Client should prove useful in a number of different contexts: textbooks and technical manuscripts; handbooks; and product specification and design sheets.

The actionable PDF version of an extended version of this paper [14] (in which is embedded an actionable equation)

\(^1\)Common Object Request Broker Architecture — http://www.corba.org

may be found on our website\(^3\); readers are encouraged to access the electronic version [14] and exercise the \textsc{SimTeX} interface; a brief users manual for which may be found again on our website\(^4\). A more involved description of the \textsc{SimTeX} client may be found in [16].

$$\begin{pmatrix} \sigma \\ \mathcal{D} \\ W \end{pmatrix} = \mathcal{F} \begin{pmatrix} H = \\ \theta = \\ t_f = \\ t_t = \\ \gamma = \end{pmatrix}$$

\[\pm\]

**FIGURE**

C.2 \textsc{SimLab}

The main drawback of \textsc{SimTeX} is the inability to manipulate the on-line codes. \textsc{SimLab} is a suite of tools that permit Users to incorporate Server on-line codes as MATLAB functions within the standard MATLAB interface; and to generate new Servers and on-line codes from standard MATLAB functions (which themselves may be built upon other on-line codes). In short, \textsc{SimLab} permits the User to treat the inputs and outputs of our on-line codes as mathematical objects that are the result of, or an argument to, other functions — graphics, system design, or optimization — and to archive these higher level operations in new Server objects available to all Clients once registered in the Directory Service.

For example, once the Truss input–output relationship has been incorporated into MATLAB (this is done by calling a MATLAB script \texttt{st2m}), we may set the values for the four components of the parameter vector by entering

- \texttt{p.values(1).value = 10;}
- \texttt{p.values(1).name = 'H';}
- \texttt{p.values(2).value = 2.0;}
- \texttt{p.values(2).name = 't_t';}
- \texttt{p.values(3).value = 1.0;}
- \texttt{p.values(3).name = 't_f';}
- \texttt{p.values(4).value = 45;}
- \texttt{p.values(4).name = 'theta';}

within the MATLAB command window. To determine the output value and bound gap for this value of the 4-uple parameter, we then enter

\[
\text{[Stress, Bound\_Stress] = Truss\_Stress(p)}
\]

which returns

\[
\text{Stress} = 12.6354 \\
\text{Bound\_Stress} = 0.0037
\]

where \texttt{Stress} corresponds to the improved predictor \(s_N^2(\mu)\) (given by (24), our definition of \(s_N^2(\mu)\), and (10)) and \texttt{Bound\_Stress} represents the half bound gap, \(\frac{1}{2}\Delta_{N,M}(\mu)\). It is also now possible of course to find all values of \texttt{Stress} greater than 12.0 for \(\theta\) in the range \([30^\circ, 60^\circ]\) and all other parameters fixed as in the list above. To wit, we enter

\[
\text{for i=1:100} \\
\text{p.values(4).value = 30+i*30/100;}
\text{t(i)=p.values(4).value;}
\text{[o(i),e(i)] = Truss\_Stress(p);}
\text{end}
\text{plot(t(o<12.0),o(o<12.0),'b'); hold on; grid on;}
\text{plot(t(o>=12.0),o(o>=12.0),'r--');}
\]

which generates Figure 3. Note that, through the expression \texttt{t(o<12.0)}, we now have the values of \(\theta\) for which the stresses are less than the specified critical value (here chosen to be 12.0).

\[
\text{Fig. 3} \\
\text{(Nondimensionalized) Stress as a function of the truss-angle \(\theta\).}
\]

Obviously, once the on-line code is incorporated within the MATLAB environment, we have the full functionality of MATLAB at our disposal; and the rapid response of the reduced–basis output bounds maintains the immediate response expected of an interactive environment, even though we are in fact solving — and solving reliably — partial differential equations.

**Acknowledgements**

We would like to thank Shidrati Ali of the Singapore-MIT Alliance and Yuri Solodukhov for very helpful discussions. We would also like to acknowledge our longstanding collaborations with Professor Yvon Maday of University Paris VI, Professor Jaime Peraire of MIT, and Professor Einar Rønquist of the Norwegian University of Science.
and Technology. This work was supported by the Singapore-
MIT Alliance, by DARPA and ONR under Grant F49620-01-
1-0458, by DARPA and AFOSR under Grant N00014-01-1-
0523 (Subcontract Grant 340-6218-3), and by NASA under
Grant NAG-1-1978.

REFERENCES

of global shape functions in structural analysis. AIAA Journal,

Theory and applications. Mechanical Systems and Signal Process-
ing, 10(4):381–394, 1996.


[4] A. G. Evans, J. W. Hutchinson, N.A. Fleck, M. F. Ashby, and
H. N. G. Wadley. The topological design of multifunctional cellular

[5] J. P. Fink and W. C. Rheinboldt. On the error behavior of the re-
duced basis technique for nonlinear finite element approximations.

flexural actuator. Journal of the Mechanics and Physics of Solids,

Output bounds for reduced-basis approximations of symmetric posi-
tive definite eigenvalue problems. C. R. Acad. Sci. Paris, Série I,

box reduced-basis output bound methods for shape optimization.
In Proceedings 12th International Domain Decomposition Confer-

gence theory for reduced-basis approximation of single-parameter
symmetric coercive elliptic partial differential equations. C. R.


methods for elliptic partial differential equations. SIAG/OPT Views-and-News,
11(1), April 2000.

1989.

method solution of nonlinear equations. Mathematics of Compu-

G. Turinici. Reliable real-time solution of parametrized partial dif-
ferential equations: Reduced-basis output bound methods. Journal of Fluids Engineering,

A.T. Patera, and G. Turinici. Reduced–basis output bound meth-
ods for parametrized partial differential equations. In Proceedings

[16] Christophe Prud’homme, Dimitrios V. Rovas, Karen Veroy, and
Anthony T. Patera. A mathematical and computational frame-
work for reliable real-time solution of parametrized partial differen-

[17] W.C. Rheinboldt. On the theory and error estimation of the re-
duced basis method for multi-parameter problems. Nonlinear Anal-

[18] K. Veroy. Reduced-Basis Methods Applied to Problems in Elastic-
ity: Analysis and Applications. PhD thesis, Massachusetts Insti-