

Space-time goal-oriented reduced basis approximation for linear wave equation

K. C. Hoang, P. Kerfriden and S. P. A. Bordas*

Cardiff University, School of Engineering,
The Parade, CF24 3AA Cardiff, United Kingdom

Abstract

In this paper, we study numerically the linear damped second-order hyperbolic partial differential equation (PDE) with affine parameter dependence using a goal-oriented approach by finite element (FE) and reduced basis (RB) methods. The main contribution of this paper is the “goal-oriented” proper orthogonal decomposition (POD)–Greedy sampling procedure within the RB approximation context. First, we introduce the RB recipe: Galerkin projection onto a space Y_N spanned by solutions of the governing PDE at N selected points in parameter space. This set of N parameter points is constructed by the standard POD–Greedy sampling procedure already developed. Second, based on the affine parameter dependence, we make use of the offline-online computational procedures: in the offline stage, we generate the RB space; in the online stage, given a new parameter value, we calculate rapidly and accurately the space-time RB output of interest and its associated asymptotic error. The proposed goal-oriented POD–Greedy sampling procedure can now be implemented and will look for the parameter points such that it minimizes this (asymptotic) output error rather than the solution error (or, error indicator which is the dual norm of residual) as in the standard POD–Greedy procedure. Numerical results show that the new goal-oriented POD–Greedy sampling procedure improves significantly the accuracy of the space-time output computation in comparison with the standard POD–Greedy one. The method is thus ideally suited for repeated, rapid and reliable evaluation of input-output relationships within the space-time setting.

Keywords: wave equation; goal-oriented asymptotic error; reduced basis method; goal-oriented POD–Greedy algorithm; space-time domain

1 Introduction

The design, optimization and control procedures of engineering problems often require several forms of performance measures or outputs – such as displacements, heat fluxes or flowrates [1]. Generally, these outputs are functions of field variables such as displacements, temperature or velocities which are usually governed by a PDE. The parameter or input will frequently define a particular configuration of the model problem. Therefore, the relevant system behavior will be described by an implicit input-output relationship; where its computation requires the solution of the underlying parameter-PDE (or μ PDE). We pursue the RB method [2, 3] which permits the *efficient* and *reliable* evaluation of this PDE-induced input-output relationship in *many query* and *real-time* contexts.

The RB method was first introduced in the late 1970s for nonlinear analysis of structures and has been further investigated and developed more broadly. Recently, the RB method was well developed for various kinds and classes of parametrized PDEs such as: the eigenvalue problems, the coercive/non-coercive affine/non-affine linear/nonlinear elliptic PDEs, the coercive/non-coercive affine/non-affine

*Email: {hoangkhacchi, pierre.kerfriden, stephane.bordas}@gmail.com

linear/nonlinear parabolic PDEs, the coercive affine linear hyperbolic PDEs, and several highly nonlinear problems such as Burger’s equation and Boussinesq equation. All of these works, which were proposed and performed by Patera and co-workers, can be found under an abstract list form in the website [4]. For the linear wave equation, the RB method and associated *a posteriori* error estimation was developed successfully with some levels [5, 6, 7]; however, none of these works have focused on goal-oriented RB approximation and its associated error estimation.

Adaptive finite element (FE) methods and goal-oriented error estimates for the wave equation have been investigated widely in many applications [8]. Among those methods, the most well-known one is the dual-weighted residual (DWR) method which was proposed by Rannacher and co-workers [9, 10, 11, 12]. In those works, the authors have used the DWR method to quantify the *a posteriori* error of the interest output in order to refine locally the finite element mesh in an adaptive manner. The final goal is to minimize computational efforts and maximize the accuracy of the interest output in an adaptive and controllable manner. In particular, the DWR method makes use of an auxiliary dual (or sensitivity) equation to derive an *a posteriori* error expression for the interest output from the primal residual and the dual solution of that dual equation in space-time setting [8]. The name “dual-weighted residual” is thus derived from this fact.

It is thus very natural to combine goal-oriented estimation with RB approximations [1, 13] to improve the accuracy of the interest output within RB context. Bringing the “DWR idea” to the RB context, we can also obtain an error expression which relates the output error with the primal residual of the dual solution in space-time setting. Namely, the error between the (space-time) FE and RB outputs will be exactly equal to the primal residual of the FE dual solution [12]. However, to make this error expression truly *a posteriori*, ones prefer to approximate the FE dual solution by some RB ones rather than using it directly within the expression [1, 13]. Hence, the error expression now becomes an *a posteriori* error approximation since the equal sign (in the expression) now becomes the approximation one. This approach is generally good; but it suffers from two main drawbacks: first, it is only an error approximation which is not a true upper error bound and second, underlying concepts and computational effort are high/complicated since we need to solve additionally a dual problem and all related issues. We also implemented this approach recently in the work [14].

Here, we also give a brief review on goal-oriented estimations within reduced order modeling context using the snapshots-POD method [15, 16, 17, 18, 19]. Goal-oriented estimations using the snapshots-POD method were implemented in [20, 21]. In this approach, the authors solved an PDE-constrained optimization problem to find the optimal set of basis functions. In particular, the optimal basis functions are found such that they minimize the true output errors (with appropriate regularization techniques) and subject to equilibrium PDE-constraints [22, 23]. This approach is optimal, however, it is very expensive since ones have to compute all the FEM solutions/outputs in every iteration within optimization solvers; and hence, it would limit the number of input parameters in comparison with the RB approach.

In this paper, we devise a simpler approach that addresses the two aforementioned drawbacks. First, we use asymptotic output errors which are also not truly upper error bounds but asymptotically converged to the true error; their performances are well comparable with the *a posteriori* DWR error expression. Second, the approach completely do not use any information of the dual problem and hence simpler and easier to implement. Based on the standard POD–Greedy sampling procedure and the asymptotic output errors, we propose the new “goal-oriented” POD–Greedy sampling procedure, which will pick up the parameter points such that the error (or error approximation) of the output functional is minimized. This idea is novel and further developed the idea of the standard POD–Greedy sampling procedure currently used [24, 25, 3], where the algorithm will pick up optimally all parameter points such that the error (or error indicator) of the field variable is minimized. By this way, we expect to improve significantly the accuracy of the RB output functional computations; but consequently, we might lose the rapid convergent rate of the field variable as in the standard POD–Greedy algorithm. In fact, as we can see later in the numerical results section, the convergent rate of the field variable by the two algorithms are quite similar; while the convergent rate of the output by

the goal-oriented POD–Greedy algorithm is faster than that of the standard POD–Greedy one¹.

The paper is organized as follows. In Section 2, we introduce necessary definitions, concepts and notations and then state the problem using a semidiscrete approach: fully discretizing in space using Galerkin FEM and marching in time using Newmark’s trapezoidal rule. In Section 3, we describe various topics related to the RB methodology: approximation, the standard versus goal-oriented algorithms, error estimations and offline-online computational procedure. We then present some numerical results of the three-dimensional dental implant problem [3] to verify the performance of the proposed algorithm in Section 4. Finally, we provide some concluding remarks in Section 5.

2 Problem statement

2.1 Abstract formulation

We consider a spatial domain $\Omega \in \mathbb{R}^d$ with Lipschitz continuous boundary $\partial\Omega$. We denote the Dirichlet portion of the boundary by $\Gamma_i^D, 1 \leq i \leq d$. We then introduce the Hilbert spaces

$$Y^e = \{v \equiv (v_1, \dots, v_d) \in (H^1(\Omega))^d \mid v_i = 0 \text{ on } \Gamma_i^D, i = 1, \dots, d\}, \quad (1a)$$

$$X^e = (L^2(\Omega))^d. \quad (1b)$$

Here, $H^1(\Omega) = \{v \in L^2(\Omega) \mid \nabla v \in (L^2(\Omega))^d\}$ where $L^2(\Omega)$ is the space of square-integrable functions over Ω . We equip our spaces with inner products and associated norms $(\cdot, \cdot)_{Y^e}$ ($(\cdot, \cdot)_{X^e}$) and $\|\cdot\|_{Y^e} = \sqrt{(\cdot, \cdot)_{Y^e}}$ ($\|\cdot\|_{X^e} = \sqrt{(\cdot, \cdot)_{X^e}}$), respectively; a typical choice is

$$(w, v)_{Y^e} = \int_{\Omega} \frac{\partial w_i}{\partial x_j} \frac{\partial v_i}{\partial x_j} + w_i v_i, \quad (2a)$$

$$(w, v)_{X^e} = \int_{\Omega} w_i v_i, \quad (2b)$$

where the summation over repeated indices is assumed.

We next define our parameters set $\mathcal{D} \in \mathbb{R}^P$, a typical point in which shall be denoted $\mu \equiv (\mu_1, \dots, \mu_P)$. We then define the parametrized bilinear forms a in Y^e , $a : Y^e \times Y^e \times \mathcal{D} \rightarrow \mathbb{R}$; m, c, f, ℓ are continuous bilinear and linear forms in X^e , $m : X^e \times X^e \times \mathcal{D} \rightarrow \mathbb{R}$, $c : X^e \times X^e \times \mathcal{D} \rightarrow \mathbb{R}$, $f : X^e \times \mathcal{D} \rightarrow \mathbb{R}$ and $\ell : X^e \rightarrow \mathbb{R}$.

For simplicity, we directly consider a time-discrete framework associated to the time interval $I = [0, T]$. We divide I into K subintervals of equal length $\Delta t = \frac{T}{K}$ and define $t^k = k\Delta t$, $0 \leq k \leq K$. We shall consider the Newmark’s trapezoidal scheme with coefficients $(\gamma_s = \frac{1}{2}, \beta_s = \frac{1}{4})$ for the time integration. Our results must be stable as $\Delta t \rightarrow 0, K \rightarrow \infty$.

The “exact” semi-discrete problem is stated follows: given a parameter $\mu \in \mathcal{D} \subset \mathbb{R}^P$, we evaluate the (space-time) output of interest

$$s(\mu) = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \int_{\Gamma_o} u^e(x, t; \mu) \Sigma(x, t) dx dt = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u^e(x, t; \mu)) dt, \quad (3)$$

where the field variable, $u^e(\mu, t^k) \in Y^e$, $1 \leq k \leq K$, satisfies the weak form of the μ -parametrized hyperbolic PDE [26]

¹In subsequent sections, for simplicity we shall call the “standard algorithm” to mention the standard POD–Greedy algorithm, and the “goal-oriented algorithm” to mention the goal-oriented POD–Greedy algorithm, respectively.

$$\begin{aligned}
& \frac{1}{\Delta t^2} m(u^e(\mu, t^{k+1}), v; \mu) + \frac{1}{2\Delta t} c(u^e(\mu, t^{k+1}), v; \mu) + \frac{1}{4} a(u^e(\mu, t^{k+1}), v; \mu) \\
& \quad = -\frac{1}{\Delta t^2} m(u^e(\mu, t^{k-1}), v; \mu) + \frac{1}{2\Delta t} c(u^e(\mu, t^{k-1}), v; \mu) - \frac{1}{4} a(u^e(\mu, t^{k-1}), v; \mu) \\
& \quad + \frac{2}{\Delta t^2} m(u^e(\mu, t^k), v; \mu) - \frac{1}{2} a(u^e(\mu, t^k), v; \mu) + g^{eq}(t^k) f(v; \mu), \quad \forall v \in Y^e, 1 \leq k \leq K-1, \quad (4)
\end{aligned}$$

with

$$g^{eq}(t^k) = \frac{1}{4} g(t^{k-1}) + \frac{1}{2} g(t^k) + \frac{1}{4} g(t^{k+1}), \quad 1 \leq k \leq K-1, \quad (5)$$

and initial conditions: $u^e(\mu, t^0) = 0$, $\frac{\partial u^e(\mu, t^0)}{\partial t} = 0$.

Here, Γ_o is some (output) spatial regions of interest and $\Sigma(x, t)$ is an extractor which depends on the view position of an “observer” in the space-time domain; and $\ell(u^e(x, t; \mu)) = \int_{\Gamma_o} u^e(x, t; \mu) \Sigma(x, t) dx$.

We next introduce a reference finite element approximation space $Y \subset Y^e (\subset X^e)$ of dimension \mathcal{N} ; we further define $X \equiv X^e$. Note that Y and X shall inherit the inner product and norm from Y^e and X^e , respectively. Our “true” finite element approximation $u(\mu, t^k) \in Y$ to the “exact” problem is stated as

$$\begin{aligned}
& \frac{1}{\Delta t^2} m(u(\mu, t^{k+1}), v; \mu) + \frac{1}{2\Delta t} c(u(\mu, t^{k+1}), v; \mu) + \frac{1}{4} a(u(\mu, t^{k+1}), v; \mu) \\
& \quad = -\frac{1}{\Delta t^2} m(u(\mu, t^{k-1}), v; \mu) + \frac{1}{2\Delta t} c(u(\mu, t^{k-1}), v; \mu) - \frac{1}{4} a(u(\mu, t^{k-1}), v; \mu) \\
& \quad + \frac{2}{\Delta t^2} m(u(\mu, t^k), v; \mu) - \frac{1}{2} a(u(\mu, t^k), v; \mu) + g^{eq}(t^k) f(v; \mu), \quad \forall v \in Y, 1 \leq k \leq K-1, \quad (6)
\end{aligned}$$

with initial conditions²: $u(\mu, t^0) = 0$, $\frac{\partial u(\mu, t^0)}{\partial t} = 0$; we then evaluate the interest output from

$$s(\mu) = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(x, t; \mu)) dt. \quad (7)$$

The RB approximation shall be built upon our reference finite element approximation, and the RB error will thus be evaluated with respect to $u(\mu, t^k) \in Y$. Clearly, our methods must remain computationally efficient and stable as $\mathcal{N} \rightarrow \infty$.

We shall make the following assumptions. First, we assume that the bilinear forms $a(\cdot, \cdot; \mu)$ and $m(\cdot, \cdot; \mu)$ are continuous,

$$a(w, v; \mu) \leq \gamma \|w\|_Y \|v\|_Y \leq \gamma_0 \|w\|_Y \|v\|_Y, \quad \forall w, v \in Y, \forall \mu \in \mathcal{D}, \quad (8a)$$

$$m(w, v; \mu) \leq \rho \|w\|_X \|v\|_X \leq \rho_0 \|w\|_X \|v\|_X, \quad \forall w, v \in Y, \forall \mu \in \mathcal{D}, \quad (8b)$$

coercive,

$$0 \leq \alpha_0 \leq \alpha(\mu) \equiv \inf_{v \in Y} \frac{a(v, v; \mu)}{\|v\|_Y^2}, \quad \forall \mu \in \mathcal{D}, \quad (9a)$$

$$0 \leq \sigma_0 \leq \sigma(\mu) \equiv \inf_{v \in Y} \frac{m(v, v; \mu)}{\|v\|_X^2}, \quad \forall \mu \in \mathcal{D}; \quad (9b)$$

and symmetric $a(v, w; \mu) = a(w, v; \mu)$, $\forall w, v \in Y$, $\forall \mu \in \mathcal{D}$, and $m(v, w; \mu) = m(w, v; \mu)$, $\forall w, v \in X$, $\forall \mu \in \mathcal{D}$. (We (plausibly) suppose that $\gamma_0, \rho_0, \alpha_0$ and σ_0 may be chosen independent of \mathcal{N} [1]). We also

²In order to start the procedure (6), $u(\mu, t^1)$ is computed as on page 491 of [27].

require that the linear forms $f(\cdot; \mu) : Y \rightarrow \mathbb{R}$ and $\ell(\cdot) : Y \rightarrow \mathbb{R}$ be bounded with respect to $\|\cdot\|_Y$ and $\|\cdot\|_X$, respectively.

Second, we shall assume that a , m , c and f depend affinely on the parameter μ and thus can be expressed as

$$m(w, v; \mu) = \sum_{q=1}^{Q_m} \Theta_m^q(\mu) m^q(w, v), \quad \forall w, v \in Y, \mu \in \mathcal{D}, \quad (10a)$$

$$c(w, v; \mu) = \sum_{q=1}^{Q_c} \Theta_c^q(\mu) c^q(w, v), \quad \forall w, v \in Y, \mu \in \mathcal{D}, \quad (10b)$$

$$a(w, v; \mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(w, v), \quad \forall w, v \in Y, \mu \in \mathcal{D}, \quad (10c)$$

$$f(v; \mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f^q(v), \quad \forall v \in Y, \mu \in \mathcal{D}, \quad (10d)$$

for some (preferably) small integers $Q_{m,c,a,f}$. Here, the smooth functions $\Theta_{m,c,a,f}^q(\mu) : \mathcal{D} \rightarrow \mathbb{R}$ depend on μ , but the bilinear and linear forms m^q , c^q , a^q and f^q do *not* depend on μ .

Finally, we also require that all linear and bilinear forms be independent of time – the system is thus linear time-invariant (LTI) [1]. We shall point out that one application which satisfies this assumption is the dental implant problem [3, 28].

2.2 Impulse response

In many dynamical systems, generally, the applied force to excite the system (e.g., $g(t^k)$ in (5)) is not known in advance and thus we cannot solve (6) for $u(\mu, t^k)$. In such situations, fortunately, we may appeal to the LTI hypothesis to justify an impulse approach as described now [1]. We note from the Duhamel's principle that the solution of any LTI system can be written as the convolution of the impulse response with the control input: for any control input $g^{\text{any}}(t^k)$, we can obtain its corresponding solution $u^{\text{any}}(\mu, t^k)$, $1 \leq k \leq K$ from

$$u^{\text{any}}(\mu, t^k) = \sum_{j=1}^k u^{\text{unit}}(\mu, t^{k-j+1}) g^{\text{any}}(t^j), \quad 1 \leq k \leq K, \quad (11)$$

where $u^{\text{unit}}(\mu, t^k)$ is the solution of (6) for a unit impulse control input $g^{\text{unit}}(t^k) = \delta_{1k}$, $1 \leq k \leq K$ (δ is the Kronecker delta symbol). Therefore, it is sufficient to build the RB basis functions for the problem based on this impulse response [1].

3 Reduced basis approximation

3.1 Approximation

We introduce the nested samples $S_* = \{\mu_1 \in \mathcal{D}, \mu_2 \in \mathcal{D}, \dots, \mu_N \in \mathcal{D}\}$, $1 \leq N \leq N_{\text{max}}$, and associated nested Lagrangian RB spaces $Y_N = \text{span}\{\zeta_n, 1 \leq n \leq N\}$, $1 \leq N \leq N_{\text{max}}$, where $\zeta_n \in Y_N$, $1 \leq n \leq N_{\text{max}}$ are mutually $(\cdot, \cdot)_Y$ -orthogonal RB basis functions. The sets S_* and Y_N shall be constructed appropriately by the standard and goal-oriented POD-Greedy algorithms described in Section 3.2 afterward.

Our reduced basis approximation $u_N(\mu, t^k)$ to $u(\mu, t^k)$ is then obtained by a standard Galerkin projection: given $\mu \in \mathcal{D}$, we now look for $u_N(\mu, t^k) \in Y_N$ satisfies

$$\begin{aligned}
& \frac{1}{\Delta t^2} m(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{2\Delta t} c(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{4} a(u_N(\mu, t^{k+1}), v; \mu) \\
& = -\frac{1}{\Delta t^2} m(u_N(\mu, t^{k-1}), v; \mu) + \frac{1}{2\Delta t} c(u_N(\mu, t^{k-1}), v; \mu) - \frac{1}{4} a(u_N(\mu, t^{k-1}), v; \mu) \\
& + \frac{2}{\Delta t^2} m(u_N(\mu, t^k), v; \mu) - \frac{1}{2} a(u_N(\mu, t^k), v; \mu) + g^{eq}(t^k) f(v; \mu), \quad \forall v \in Y_N, 1 \leq k \leq K-1, \quad (12)
\end{aligned}$$

where the zero initial conditions are defined and treated as mentioned in Section 2.1; we then evaluate the output estimate, $s_N(\mu)$, from

$$s_N(\mu) = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u_N(x, t; \mu)) dt. \quad (13)$$

3.2 Goal-oriented POD–Greedy sampling procedure

3.2.1 The proper orthogonal decomposition

We aim to generate an optimal (in the mean square error sense) basis set $\{\zeta_m\}_{m=1}^M$ from any given set of $M_{\max} (\geq M)$ snapshots $\{\xi_k\}_{k=1}^{M_{\max}}$. To do this, let $V_M = \text{span}\{v_1, \dots, v_M\} \subset \text{span}\{\xi_1, \dots, \xi_{M_{\max}}\}$ be an arbitrary space of dimension M . We assume that the space V_M is orthogonal such that $(v_n, v_m) = \delta_{nm}$, $1 \leq n, m \leq M$ ((\cdot, \cdot) denotes an appropriate inner product and δ_{nm} is the Kronecker delta symbol). The POD space, $W_M = \text{span}\{\zeta_1, \dots, \zeta_M\}$ is defined as

$$W_M = \arg \min_{V_M \subset \text{span}\{\xi_1, \dots, \xi_{M_{\max}}\}} \left(\frac{1}{M_{\max}} \sum_{k=1}^{M_{\max}} \inf_{\alpha^k \in \mathbb{R}^M} \left\| \xi_k - \sum_{m=1}^M \alpha_m^k v_m \right\|^2 \right). \quad (14)$$

In essence, the POD space W_M which is extracted from the given set of snapshots $\{\xi_k\}_{k=1}^{M_{\max}}$ is the space that best approximate this given set of snapshots and can be written as $W_M = \text{POD}(\{\xi_1, \dots, \xi_{M_{\max}}\}, M)$. We can construct this POD space by using the method of snapshots which is presented concisely in the Appendix of [29].

3.2.2 Goal-oriented POD–Greedy algorithm

We now discuss the POD–Greedy algorithms [24, 3] to construct the nested sets S_* and Y_N of interest. Let Ξ_{train} be a finite set of the parameters in \mathcal{D} ($\Xi_{\text{train}} \in \mathcal{D}$); and S_* denote the set of greedily selected parameters in Ξ_{train} . Initialize $S_* = \{\mu_0\}$, where μ_0 is an arbitrarily chosen parameter. Let $e_{\text{proj}}(\mu, t^k) = u(\mu, t^k) - \text{proj}_{Y_N} u(\mu, t^k)$, where $\text{proj}_{Y_N} u(\mu, t^k)$ is the Y_N -orthogonal projection of $u(\mu, t^k)$ into the Y_N space. The standard and our proposed goal-oriented POD–Greedy algorithms are presented simultaneously in Table 1.

In Table 1, the superscript “st” denotes the standard POD–Greedy and “go” denotes the goal-oriented POD–Greedy algorithms, respectively. The term $\|\mathcal{R}^{\text{st}}(v; \mu, t^k)\|_{Y'}^2, \forall v \in Y, 1 \leq k \leq K-1$, is the dual norm of the associated residual of equation (12), namely,

$$\begin{aligned}
\mathcal{R}(v; \mu, t^k) &= g^{eq}(t^k) f(v; \mu) \\
& - \frac{1}{\Delta t^2} (m(u_N(\mu, t^{k+1}), v; \mu) - 2m(u_N(\mu, t^k), v; \mu) + m(u_N(\mu, t^{k-1}), v; \mu)) \\
& - \frac{1}{\Delta t} \left(\frac{1}{2} c(u_N(\mu, t^{k+1}), v; \mu) - \frac{1}{2} c(u_N(\mu, t^{k-1}), v; \mu) \right) \\
& - \left(\frac{1}{4} a(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{2} a(u_N(\mu, t^k), v; \mu) + \frac{1}{4} a(u_N(\mu, t^{k-1}), v; \mu) \right); \quad (15)
\end{aligned}$$

(T1a)	Set $Y_N^{\text{st}} = 0$	Set $Y_N^{\text{go}} = 0$
(T1b)	Set $\mu_*^{\text{st}} = \mu_0$	Set $\mu_*^{\text{go}} = \mu_0$
(T1c)	While $N \leq N_{\text{max}}$	While $N \leq N_{\text{max}}$
(T1d)	$\mathcal{W}^{\text{st}} = \{e_{\text{proj}}^{\text{st}}(\mu_*^{\text{st}}, t^k), 0 \leq k \leq K\};$	$\mathcal{W}^{\text{go}} = \{e_{\text{proj}}^{\text{go}}(\mu_*^{\text{go}}, t^k), 0 \leq k \leq K\};$
(T1e)	$Y_{N+M}^{\text{st}} \leftarrow Y_N^{\text{st}} \oplus \text{POD}(\mathcal{W}^{\text{st}}, M);$	$Y_{N+M}^{\text{go}} \leftarrow Y_N^{\text{go}} \oplus \text{POD}(\mathcal{W}^{\text{go}}, M);$
(T1f)	$N \leftarrow N + M;$	$N \leftarrow N + M;$
(T1g)	$\mu_*^{\text{st}} = \arg \max_{\mu \in \Xi_{\text{train}}} \{\Delta_u(\mu)\};$	$\mu_*^{\text{go}} = \arg \max_{\mu \in \Xi_{\text{train}}} \{\Delta_s(\mu)\};$
(T1h)	$S_*^{\text{st}} \leftarrow S_*^{\text{st}} \cup \{\mu_*^{\text{st}}\};$	$S_*^{\text{go}} \leftarrow S_*^{\text{go}} \cup \{\mu_*^{\text{go}}\};$
(T1i)	end.	end.
(T1j)	$\Delta_u(\mu) = \frac{\sqrt{\sum_{k=1}^K \ \mathcal{R}^{\text{st}}(v; \mu, t^k)\ _Y^2}}{\sqrt{\sum_{k=1}^K \ u_N^{\text{st}}(\mu, t^k)\ _Y^2}}$	$\Delta_s(\mu) = \left \frac{s_{2N}^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)}{s_{2N}^{\text{st}}(\mu)} \right $

Table 1: (Left) Standard POD–Greedy sampling algorithm and (Right) our proposed goal-oriented POD–Greedy sampling algorithm.

and the subscript “ $2N$ ” means usually as “two times of N ”.

Remarks

- In essence, the term $\Delta_u(\mu)$ is the ratio of the dual norm of the residual to the RB solution. Thus, this term is roughly considered as an error indicator for the error in the solution (or field variable); and has been used mainly in the current standard POD–Greedy algorithm [24, 25, 3].
- On the contrary, the term $\Delta_s(\mu)$ is an asymptotic error for the interest output of the true output error (i.e., $s(\mu) - s_N(\mu)$) as $s(\mu) \approx s_{2N}^{\text{st}}(\mu)$. Therefore, as observed from Table 1, the main difference between these two algorithms is that we somehow try to minimize the error indicator of the output functional ($\Delta_s(\mu)$) rather than an error indicator of the solution ($\Delta_u(\mu)$) by means of Greedy iterations. By this way, we expect to improve further the accuracy of the RB output computation; but consequently, we might lose the rapid convergent rate of the solution as in the standard POD–Greedy algorithm.
- In order to implement the goal-oriented POD–Greedy algorithm, we need to compute $s_{2N}^{\text{st}}(\mu)$ for the term $\Delta_s(\mu)$. Therefore, we proposed implementing the standard POD–Greedy algorithm *first* to build the basis Y_N^{st} ; and then the goal-oriented POD–Greedy algorithm is performed *subsequently*. That means the offline stage of the goal-oriented algorithm is now more expensive than that of the standard one (as it also includes the offline stage of the standard one). However, the online computational cost of the goal-oriented algorithm is completely similar to that of the standard algorithm.

3.3 Error estimations

3.3.1 True errors

We use the true errors to compare the performances of the standard and goal-oriented algorithms in the online computation stage. The relative true errors by the two algorithms for the solutions are

defined as

$$e_u^{\text{st}}(\mu) = \frac{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \|u(\mu, t) - u_N^{\text{st}}(\mu, t)\|_Y dt}{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \|u(\mu, t)\|_Y dt}, \quad e_u^{\text{go}}(\mu) = \frac{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \|u(\mu, t) - u_N^{\text{go}}(\mu, t)\|_Y dt}{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \|u(\mu, t)\|_Y dt}; \quad (16)$$

and for the outputs

$$e_s^{\text{st}}(\mu) = \left| \frac{s(\mu) - s_N^{\text{st}}(\mu)}{s(\mu)} \right| = \left| \frac{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(\mu, t) - u_N^{\text{st}}(\mu, t)) dt}{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(\mu, t)) dt} \right|, \quad (17a)$$

$$e_s^{\text{go}}(\mu) = \left| \frac{s(\mu) - s_N^{\text{go}}(\mu)}{s(\mu)} \right| = \left| \frac{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(\mu, t) - u_N^{\text{go}}(\mu, t)) dt}{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(\mu, t)) dt} \right|. \quad (17b)$$

In the above expressions, $u(\mu, t)$, $u_N^{\text{st}}(\mu, t)$ and $u_N^{\text{go}}(\mu, t)$ are the FE, standard RB and goal-oriented RB solutions; $s(\mu)$, $s_N^{\text{st}}(\mu)$ and $s_N^{\text{go}}(\mu)$ are the FE, standard RB and goal-oriented RB outputs, respectively.

3.3.2 Asymptotic errors

The true errors are good for comparison purposes but are not of practical uses in the online stage, where ones require fast and countless online calculations. We introduce the asymptotic errors for output computations as an alternative mean to evaluate relatively the quality of the RB approximations. Of course asymptotic error is not a rigorous upper error bound such as the *a posteriori* error bounds [2, 30, 25]; however, there are several good reasons for using it in practice. First, the time-marching error bounds for the wave equation so far were shown to be ineffective and pessimistic due to the nature of the wave equation: exponential growing with respect to time [7, 31, 32]. (We also note that the space-time error bounds for the wave equation are, although very promising, still not yet derived in the literature.) Second, the asymptotic error converges asymptotically to the true error and thus can approximate relatively the accuracy of the RB solutions/outputs for various choices of μ . Thirdly – most important, its computational cost is very cheap: only N -dependent (note $N \ll \mathcal{N}$). The asymptotic output errors are defined as follows

$$\hat{e}_s^{\text{off}}(\mu) = \left| \frac{s_{2N}^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)}{s(\mu)} \right| = \left| \frac{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u_{2N}^{\text{st}}(\mu, t) - u_N^{\text{go}}(\mu, t)) dt}{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(\mu, t)) dt} \right|, \quad (18a)$$

$$\hat{e}_s^{\text{onl}}(\mu) = \left| \frac{s_{2N}^{\text{go}}(\mu) - s_N^{\text{go}}(\mu)}{s(\mu)} \right| = \left| \frac{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u_{2N}^{\text{go}}(\mu, t) - u_N^{\text{go}}(\mu, t)) dt}{\sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(\mu, t)) dt} \right|. \quad (18b)$$

In addition, the effectivities corresponding to these asymptotic output errors can be defined as

$$\eta_s^{\text{off}}(\mu) = \left| \frac{s_{2N}^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right| = \frac{\hat{e}_s^{\text{off}}(\mu)}{e_s^{\text{go}}(\mu)}, \quad (19a)$$

$$\eta_s^{\text{onl}}(\mu) = \left| \frac{s_{2N}^{\text{go}}(\mu) - s_N^{\text{go}}(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right| = \frac{\hat{e}_s^{\text{onl}}(\mu)}{e_s^{\text{go}}(\mu)}. \quad (19b)$$

We note that $\hat{e}_s^{\text{off}}(\mu)$ is similar to $\Delta_s(\mu)$ – the error indicator for the output in the offline stage; and $\hat{e}_s^{\text{onl}}(\mu)$ is the useful asymptotic error practically used in the online stage of the goal-oriented POD–Greedy algorithm.

3.4 Offline-online computational procedure

In this section, we develop offline-online computational procedures in order to fully exploit the dimension reduction of the problem [1, 3, 30]. We note that both algorithms (standard and goal-oriented) have the same offline-online computational procedures, they are only different in the ways to build the sets S_* and Y_N within Greedy iterations. We first express $u_N(\mu, t^k)$ as:

$$u_N(\mu, t^k) = \sum_{n=1}^N u_{Nn}(\mu, t^k) \zeta_n, \quad \forall \zeta_n \in Y_N. \quad (20)$$

We then choose a test function $v = \zeta_n, 1 \leq n \leq N$ for the RB equation (12). It then follows that $\underline{u}_N(\mu, t^k) = [u_{N1}(\mu, t^k) \ u_{N2}(\mu, t^k) \ \cdots \ u_{NN}(\mu, t^k)]^T \in \mathbb{R}^N$ satisfies

$$\begin{aligned} & \left(\frac{1}{\Delta t^2} \mathbf{M}_N(\mu) + \frac{1}{2\Delta t} \mathbf{C}_N(\mu) + \frac{1}{4} \mathbf{A}_N(\mu) \right) \underline{u}_N(\mu, t^{k+1}) \\ &= \left(-\frac{1}{\Delta t^2} \mathbf{M}_N(\mu) + \frac{1}{2\Delta t} \mathbf{C}_N(\mu) - \frac{1}{4} \mathbf{A}_N(\mu) \right) \underline{u}_N(\mu, t^{k-1}) \\ &+ \left(\frac{2}{\Delta t^2} \mathbf{M}_N(\mu) - \frac{1}{2} \mathbf{A}_N(\mu) \right) + g^{eq}(t^k) \mathbf{F}_N(\mu), \quad 1 \leq k \leq K-1. \end{aligned} \quad (21)$$

The initial condition is treated similar to the treatment in (6) and (12). Here, $\mathbf{C}_N(\mu)$, $\mathbf{A}_N(\mu)$, $\mathbf{M}_N(\mu) \in \mathbb{R}^{N \times N}$ are symmetric positive definite matrices with entries $\mathbf{C}_{Nij}(\mu) = c(\zeta_i, \zeta_j; \mu)$, $\mathbf{A}_{Nij}(\mu) = a(\zeta_i, \zeta_j; \mu)$, $\mathbf{M}_{Nij}(\mu) = m(\zeta_i, \zeta_j; \mu)$, $1 \leq i, j \leq N$ and $\mathbf{F}_N \in \mathbb{R}^N$ is the RB load vector with entries $\mathbf{F}_{Ni} = f(\zeta_i)$, $1 \leq i \leq N$, respectively.

The RB output is then computed from

$$s_N(\mu) = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \mathbf{L}_N^T \underline{u}_N(\mu, t) dt. \quad (22)$$

Invoking the affine parameter dependence (10), we obtain

$$\mathbf{M}_{Nij}(\mu) = m(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_m} \Theta_m^q(\mu) m^q(\zeta_i, \zeta_j), \quad (23a)$$

$$\mathbf{C}_{Nij}(\mu) = c(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_c} \Theta_c^q(\mu) c^q(\zeta_i, \zeta_j), \quad (23b)$$

$$\mathbf{A}_{Nij}(\mu) = a(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(\zeta_i, \zeta_j), \quad (23c)$$

$$\mathbf{F}_{Ni}(\mu) = f(\zeta_i; \mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f^q(\zeta_i), \quad (23d)$$

which can be written as

$$\begin{aligned} \mathbf{M}_{Nij}(\mu) &= \sum_{q=1}^{Q_m} \Theta_m^q(\mu) \mathbf{M}_{Nij}^q, & \mathbf{C}_{Nij}(\mu) &= \sum_{q=1}^{Q_c} \Theta_c^q(\mu) \mathbf{C}_{Nij}^q, \\ \mathbf{A}_{Nij}(\mu) &= \sum_{q=1}^{Q_a} \Theta_a^q(\mu) \mathbf{A}_{Nij}^q, & \mathbf{F}_{Ni}(\mu) &= \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathbf{F}_{Ni}^q, \end{aligned} \quad (24)$$

where the *parameter independent* quantities $\mathbf{M}_N^q, \mathbf{C}_N^q, \mathbf{A}_N^q \in \mathbb{R}^{N \times N}$, and $\mathbf{F}_N^q \in \mathbb{R}^N$ are given by

$$\mathbf{M}_{N i,j}^q = m^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_m, \quad (25a)$$

$$\mathbf{C}_{N i,j}^q = c^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_c, \quad (25b)$$

$$\mathbf{A}_{N i,j}^q = a^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_a, \quad (25c)$$

$$\mathbf{F}_{N i}^q = f^q(\zeta_i), \quad 1 \leq i \leq N_{\max}, \quad 1 \leq q \leq Q_f, \quad (25d)$$

respectively.

The offline-online computational procedure is now described as follows. In the offline stage – performed only once, we first implement the standard POD–Greedy algorithm [3]: we solve for the $\zeta_n^{\text{st}}, 1 \leq n \leq N_{\max}$; then compute and store the μ -independent quantities in (25) for the estimation of the RB solution and output³. Once these RB solution and output are available, we can now implement the goal-oriented POD–Greedy algorithm. We consider each goal-oriented POD–Greedy iteration (Table 1) in more details. We first need to solve (6) for the “true” FE solutions; then do the error projection in step (T1d) and solve the POD/eigenvalue problem in step (T1e). In addition, we have to compute $O(N^2Q)$ \mathcal{N} -inner products in (25) (we denote $Q = Q_m + Q_c + Q_a$). By approximating $s(\mu)$ by the enriched approximation $s_{2N}^{\text{st}}(\mu)$ in (T1j) through the standard algorithm, we can now do a cheaply exhaustive/wide search over Ξ_{train} to look for optimal μ in each Greedy iteration. In summary, the offline stage of the goal-oriented algorithm also includes the offline stage of the standard algorithm, and therefore it is more expensive than that of the standard one.

The online stage of the goal-oriented algorithm is completely similar to that of the standard algorithm [3]. In the online stage – performed many times, for each new parameter μ – we first assemble the RB matrices in (23), this requires $O(N^2Q)$ operations. We then solve the RB governing equation (21), the operation counts are $O(N^3 + KN^2)$ as the RB matrices are generally full. Finally, we evaluate the displacement output $s_N(\mu)$ from (22) at the cost of $O(KN)$. For the online asymptotic error (i.e., $s_{2N}^{\text{go}}(\mu) - s_N^{\text{go}}(\mu)$), there is nothing more than computing one more output $s_{2N}^{\text{go}}(\mu)$ and then performing the associated subtraction; the cost is $O(2KN)$. Therefore, as required in real-time context, the online complexity to compute the output and its associated asymptotic error are independent of \mathcal{N} , and since $N \ll \mathcal{N}$ we can expect significant computational saving in the online stage relative to the classical FE approach.

4 Numerical example

In this section, we will verify both POD–Greedy algorithms by investigating an numerical example which is a three-dimensional dental implant model problem in the time domain. This model problem is similar to that in the work of Hoang et al. [3]. The details are described in the following.

4.1 A 3D dental implant model problem

We consider a simplified 3D dental implant-bone model in Fig.1(a). The geometry of the simplified dental implant-bone model is constructed by using SolidWorks 2010. The physical domain Ω consists of five regions: the outermost cortical bone Ω_1 , the cancellous bone Ω_2 , the interfacial tissue Ω_3 , the dental implant Ω_4 and the stainless steel screw Ω_5 . The 3D simplified model is then meshed and analyzed in the software ABAQUS/CAE version 6.10-1 (Fig.1(b)). A dynamic force opposite to the x -direction is then applied to a prescribed area on the body of the screw as shown in Fig.2(a). As mentioned in Section 2.2, all computations and simulations will be performed for the unit input loading case, since the solutions associated with other input loading cases can be easily inferred from the Duhamel’s convolution principle (11). For example, Fig.2(b) shows the time history of such

³There are still several terms related to the computation of dual norm of the residual, we do not show them here for simplicity. We refer the readers to [3] for all related details.

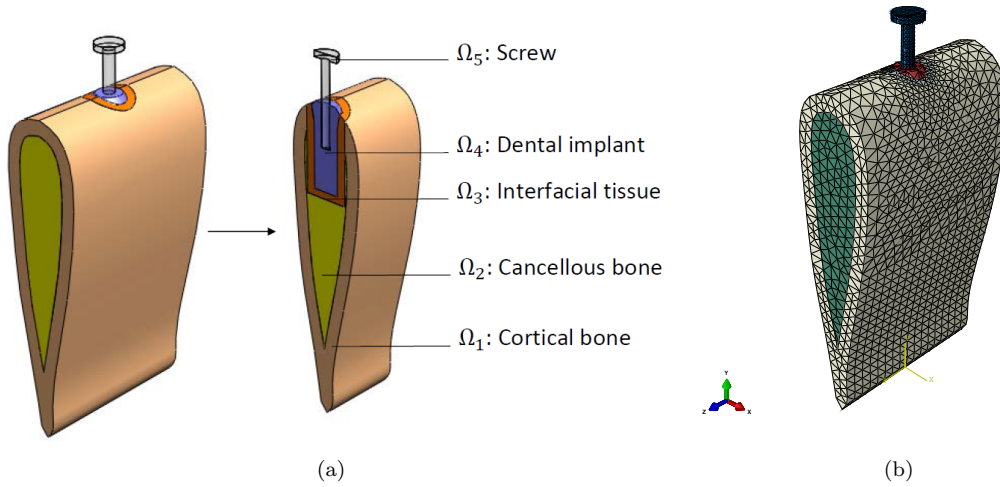


Figure 1: (a) The 3D simplified FEM model with sectional view, and (b) meshing in ABAQUS.

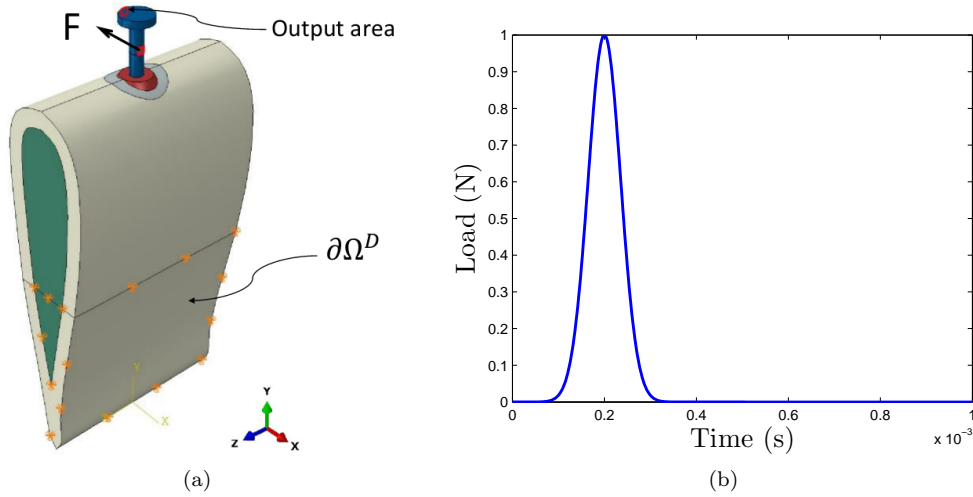


Figure 2: (a) Output area, applied load \mathbf{F} and boundary condition, and (b) time history of a particular load.

“arbitrary” input loading case. The output of interest is defined as the average displacement responses of a prescribed area on the head of the screw (Fig.2(a)). The Dirichlet boundary condition ($\partial\Omega^D$) is specified in the bottom-half of the simplified model as illustrated in Fig.2(a). The finite element mesh consists of 9479 nodes and 50388 four-node tetrahedral solid elements. The coinciding nodes of the contact surfaces between different regions (the regions Ω_1 , Ω_2 , Ω_3 , Ω_4 , Ω_5) are assumed to be rigidly fixed, i.e. the displacements in the x -, y - and z -directions are all set to be the same for the same coinciding nodes.

We assume that the regions Ω_i , $1 \leq i \leq 5$, of the simplified model are homogeneous and isotropic. The material properties: the Young’s moduli, Poisson’s ratios and densities of these regions are presented in Table 2 [33]. As similar to [3], we still use Rayleigh damping with stiffness-proportional damping coefficient β_i , $1 \leq i \leq 5$ (Table 2) such that $\mathbf{C}_i = \beta_i \mathbf{A}_i$, $1 \leq i \leq 5$, where \mathbf{C}_i and \mathbf{A}_i are the FEM damping and stiffness matrices of each region, respectively. We also note in Table 2 that (E, β)

– the Young’s modulus and Rayleigh damping coefficient associated with the interfacial tissue are our sole parameters.

Domain	Layers	E (Pa)	ν	$\rho(\text{g}/\text{mm}^3)$	β
Ω_1	Cortical bone	2.3162×10^{10}	0.371	1.8601×10^{-3}	3.38×10^{-6}
Ω_2	Cancellous bone	8.2345×10^8	0.3136	7.1195×10^{-4}	6.76×10^{-6}
Ω_3	Tissue	E	0.3155	1.055×10^{-3}	β
Ω_4	Titan implant	1.05×10^{11}	0.32	4.52×10^{-3}	5.1791×10^{-10}
Ω_5	Stainless steel screw	1.93×10^{11}	0.305	8.027×10^{-3}	2.5685×10^{-8}

Table 2: Material properties of the dental implant-bone structure.

Finally, with respect to our particular dental implant problem, the actual integral forms of the linear and bilinear forms are defined as:

$$m(w, v) = \sum_{r=1}^5 \int_{\Omega_r} \rho_r w_i v_i, \quad (26a)$$

$$a(w, v; \mu) = \sum_{r=1, r \neq 3}^5 \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l} + \mu_1 \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}, \quad (26b)$$

$$c(w, v; \mu) = \sum_{r=1, r \neq 3}^5 \beta_r \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l} + \mu_2 \mu_1 \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}, \quad (26c)$$

$$f(v) = \int_{\Gamma_1} v, \quad (26d)$$

$$\ell(v) = \frac{1}{|\Gamma_o|} \int_{\Gamma_o} v, \quad (26e)$$

for all $w, v \in Y$, $\mu \in \mathcal{D}$. Here, the parameter $\mu = (\mu_1, \mu_2) \equiv (E, \beta)$ belongs to the region Ω_3 . C_{ijkl}^r is the constitutive elasticity tensor for isotropic materials and it is expressed in terms of the Young’s modulus E and Poisson’s ratio ν of each region $\Omega_r, 1 \leq r \leq 5$, respectively. Γ_1 is the prescribed loading area (surface traction) and Γ_o is the prescribed output area as shown in Fig.2(a), respectively. It is obvious from (10) and (26) that the smooth functions $\Theta_a^1(\mu) = 1$, $\Theta_a^2(\mu) = \mu_1$; $\Theta_c^1(\mu) = 1$, $\Theta_c^2(\mu) = \mu_1 \mu_2$ depend on μ – but the bilinear forms $a^1(w, v) = \sum_{r=1, r \neq 3}^5 \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l}$, $a^2(w, v) = \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}$; $c^1(w, v) = \sum_{r=1, r \neq 3}^5 \beta_r \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l}$ and $c^2(w, v) = \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}$ do *not* depend on μ .

4.2 Numerical results

The FE space to approximate the 3D dental implant-bone problem is of dimension $\mathcal{N} = 26343$. For time integration, $T = 1 \times 10^{-3}\text{s}$, $\Delta t = 2 \times 10^{-6}\text{s}$, $K = \frac{T}{\Delta t} = 500$. The input parameter $\mu \equiv (E, \beta) \in \mathcal{D}$, where the parameter domain $\mathcal{D} \equiv [1 \times 10^6, 25 \times 10^6]\text{Pa} \times [5 \times 10^{-6}, 5 \times 10^{-5}] \subset \mathbb{R}^{P=2}$. (Note that the range of E of this parameter domain is nearly two times larger than that of [3].) The $\|\cdot\|_Y$ norm used in this work is defined as $\|w\|_Y^2 = a(w, w; \bar{\mu}) + m(w, w; \bar{\mu})$, where $\bar{\mu} = (13 \times 10^6\text{Pa}, 2.75 \times 10^{-5})$ is the arithmetic average of μ in \mathcal{D} ; $Q_a = 2$, $Q_c = 2$. To verify our computational code (performed in Matlab R2012b), we first compare the FEM outputs computed by ABAQUS and by our code with the test parameter $\mu_{\text{test}} = (10 \times 10^6\text{Pa}, 1 \times 10^{-5})$. Fig.3 shows the “unit” output displacements (i.e., under the unit loading case) in the x -, y - and z -directions versus time at μ_{test} via ABAQUS and our code, respectively. Fig.3 demonstrates that the FEM results by our code match very well with the

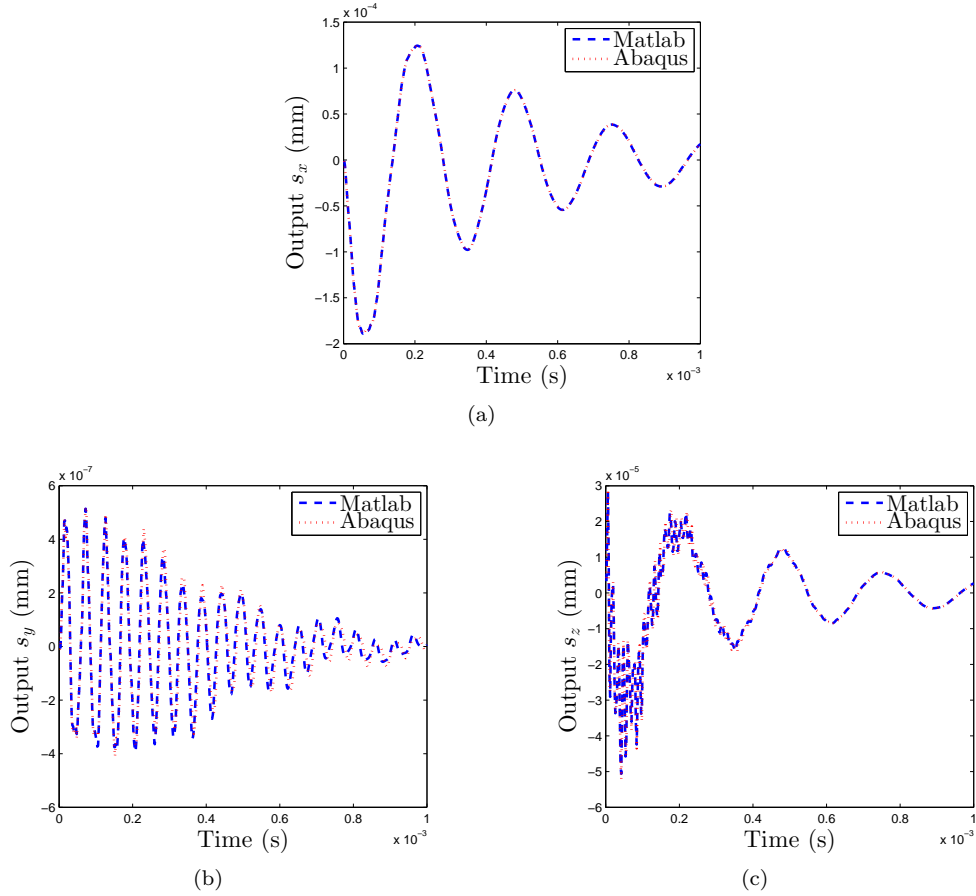
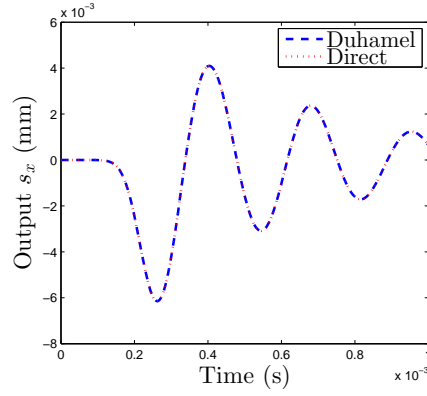


Figure 3: Comparison of the “unit” FEM output displacements computed by our code versus by ABAQUS software with respect time in the x - (a), y - (b), and z -direction (c) with $\mu_{\text{test}} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$.

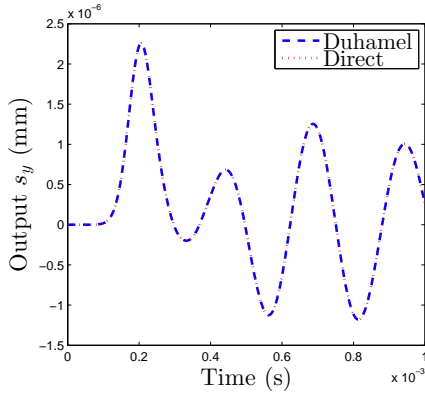
results computed by ABAQUS. Next, we show in Fig.4 the FEM output displacements versus time under the loading history in Fig.2(b) by direct FEM computation and by Duhamel’s convolution. It is observed that these two results match perfectly well with each other.

We now discuss the POD–Greedy algorithms of interest. As shown in Fig.5, a sample set Ξ_{train} is created by a uniform distribution over \mathcal{D} with $n_{\text{train}} (= 30 \times 30) = 900$ samples. Note that we use $M = 1$ and $N_{\text{max}} = 120$ (as in Table 1) to terminate the iteration procedures.

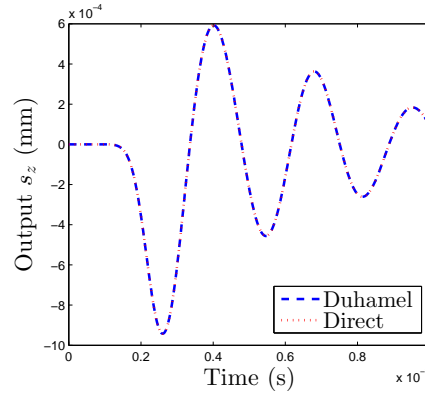
- First, we implement the standard POD–Greedy algorithm (i.e., the left column of Table 1) [3]. The results are presented in Fig.6: Fig.6(a) shows the maximum error indicator $\Delta_u^{\text{max}} = \max_{\mu \in \Xi_{\text{train}}} \Delta_u(\mu)$ as a function of N ; and Fig.6(b) shows the distribution of the sample set S_*^{st} , respectively.
- Once the sets S_*^{st} and Y_N^{st} are available, the term $s_{2N}^{\text{st}}(\mu)$ (of $\Delta_s(\mu)$ in (T1j)) is now computable. Thus, we can now implement the goal-oriented POD–Greedy algorithm (i.e., the right column of Table 1). Consider the term $s_{2N}^{\text{st}}(\mu)$, the subscript “ $2N$ ” means that we approximate the FE output $s(\mu)$ by the (standard algorithm) enriched RB approximation $s_{2N}^{\text{st}}(\mu)$ with $2N$ basis functions corresponding to the current N^{th} -Greedy iteration of the goal-oriented algorithm:



(a)



(b)



(c)

Figure 4: Comparison of “arbitrary” FEM output displacements (with the applied load in Fig.2(b)) computed by Duhamel’s convolution and direct computation with respect time in the x - (a), y - (b), and z -direction (c) with $\mu_{\text{test}} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$.

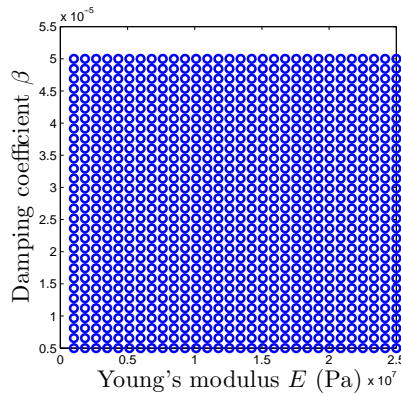


Figure 5: The training set Ξ_{train} with $n_{\text{train}} = 900$ samples.

$N^{\text{st}} = 2N^{\text{go}}$. Since $N_{\text{max}}^{\text{st}} = 120$, then we set

$$N^{\text{st}} = \begin{cases} 2N^{\text{go}} & \text{if } 1 \leq N^{\text{go}} \leq 60, \\ 120 & \text{if } 60 < N^{\text{go}} \leq 120, \end{cases} \quad (27)$$

to compute the term $s_{2N}^{\text{st}}(\mu)$. Implementing the goal-oriented algorithm, the results are presented in Fig.7: Fig.7(a) shows the maximum error indicator $\Delta_s^{\text{max}} = \max_{\mu \in \Xi_{\text{train}}} \Delta_s(\mu)$ as a function of N ; and Fig.7(b) shows the distribution of the sample set S_*^{go} , respectively.

- In order to compare the performances of the two algorithms, we create the test sample set $\Xi_{\text{test}} \subset \mathcal{D}$ which is a coarse subset of \mathcal{D} and then test the two algorithms based on this set. The set is defined as follows: $\Xi_{\text{test}} \equiv [1 \times 10^6, 25 \times 10^6] \text{Pa} \times [5 \times 10^{-6}, 5 \times 10^{-5}]$ where $n_{\text{test}} = 100$ sample points distributed uniformly. We show, as functions of N : $e_u^{\text{max}} = \max_{\mu \in \Xi_{\text{test}}} e_u(\mu)$ (defined in (16)) and $e_s^{\text{max}} = \max_{\mu \in \Xi_{\text{test}}} e_s(\mu)$ (defined in (17)) in Fig.8, respectively. From Fig.8(a), we see that the solution errors by the standard algorithm is a bit smaller than that of the goal-oriented one (with $N \geq 65$); while from Fig.8(b), the goal-oriented algorithm completely prevail over the the standard algorithm. The difference in the RB true output error by the two algorithms can be up to 10 times.
- Lastly, as regards error estimation, we show the plots of $\hat{e}_s^{\text{off,max}} = \max_{\mu \in \Xi_{\text{test}}} \hat{e}_s^{\text{off}}(\mu)$ (defined in (18a)), $\hat{e}_s^{\text{onl,max}} = \max_{\mu \in \Xi_{\text{test}}} \hat{e}_s^{\text{onl}}(\mu)$ (defined in (18b)), and $e_s^{\text{go,max}} = \max_{\mu \in \Xi_{\text{test}}} e_s^{\text{go}}(\mu)$ (defined in (17b)) as functions of N together in Fig.9(a). The maximum associated effectivities of these offline/online errors (i.e., $\eta_s^{\text{off,max}} = \max_{\mu \in \Xi_{\text{test}}} \eta_s^{\text{off}}(\mu)$ and $\eta_s^{\text{onl,max}} = \max_{\mu \in \Xi_{\text{test}}} \eta_s^{\text{onl}}(\mu)$) are also shown in Fig.9(b), respectively. As observed from Fig.9, we see that the online asymptotic output error (18b) is very close to the true output error (17b) and that its maximum effectivity is around the range [10 – 100] which is acceptable for a 3D time-dependent dynamic problem.

Finally, regarding computational time, all computations were performed on a desktop Intel(R) Core(TM) i7-3930K CPU @3.20GHz 3.20GHz, RAM 32GB, 64-bit Operating System. The computation time for the RB solver⁴ ($t_{\text{RB(online)}}$), the CPU-time for the FEM solver by our code (t_{FEM}) and the CPU-time saving factor $\kappa = t_{\text{FEM}}/t_{\text{RB(online)}}$ are listed on Table 3, respectively. We see that the RB solver is approximately $O(1000)$ times faster than the FEM solver; and thus it is clear that the RB is very efficient and reliable for solving time-dependent dynamic problems.

Table 3: Comparison of the CPU-time for a FEM and RB analysis.

N	$t_{\text{RB(online)}} \text{ (sec)}$	$t_{\text{FEM}} \text{ (sec)}$	$\kappa = t_{\text{FEM}}/t_{\text{RB(online)}}$
10	0.0072	29	4027
20	0.0081	29	3580
30	0.0106	29	2736
40	0.0140	29	2070
50	0.0243	29	1193
60	0.0300	29	966

⁴The work focuses on real-time context with many online computations, the offline stage is done once and expensive: the total computational time is approximately 2 weeks (including all FEM solutions/outputs and RB true errors of the standard and goal-oriented algorithms) on this computer.

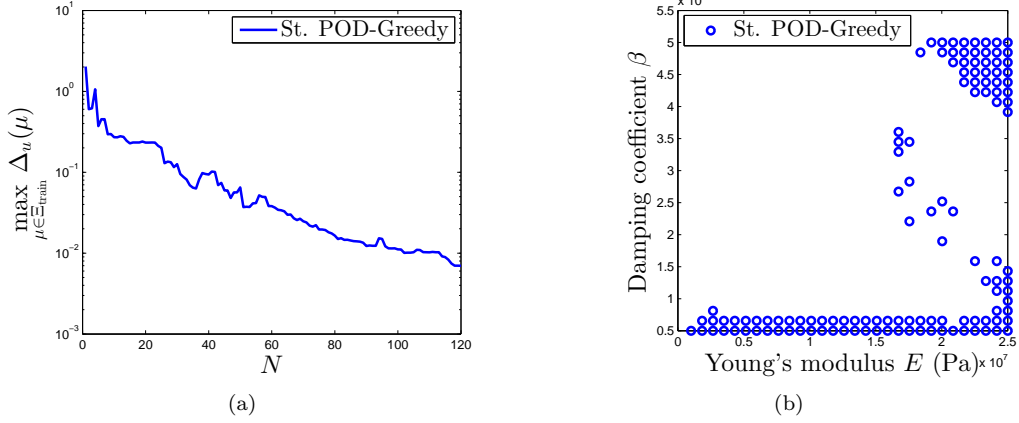


Figure 6: (a) Maximum of error indicator $\Delta_u(\mu)$ over Ξ_{train} as a function of N , and (b) distribution of sampling points by the standard POD-Greedy algorithm.

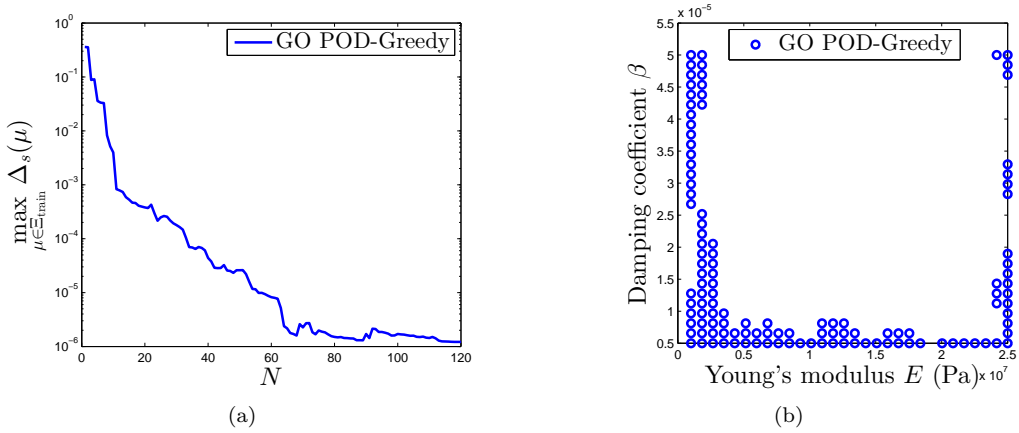


Figure 7: (a) Maximum of error indicator $\Delta_s(\mu)$ over Ξ_{train} as a function of N , and (b) distribution of sampling points by the goal-oriented POD-Greedy algorithm.

5 Conclusion

A new goal-oriented POD-Greedy sampling algorithm was proposed. The proposed algorithm cooperates and improves further the standard POD-Greedy algorithm by using the asymptotic output error rather than the dual norm of residual as error indicator in the Greedy iterations. It is demonstrated that this type of error indicator will guide the Greedy iterations to select the parameter samples/points to optimize the true output error. The proposed algorithm is then verified by investigating a 3D dental implant problem in the time domain. In comparison with the standard algorithm, we conclude that our proposed algorithm performs better – in terms of output’s accuracy, and quite similar – in terms of solution’s accuracy. The proposed algorithm is applicable to any (regular) output functional and is thus very suitable within the goal-oriented RB approximation context.

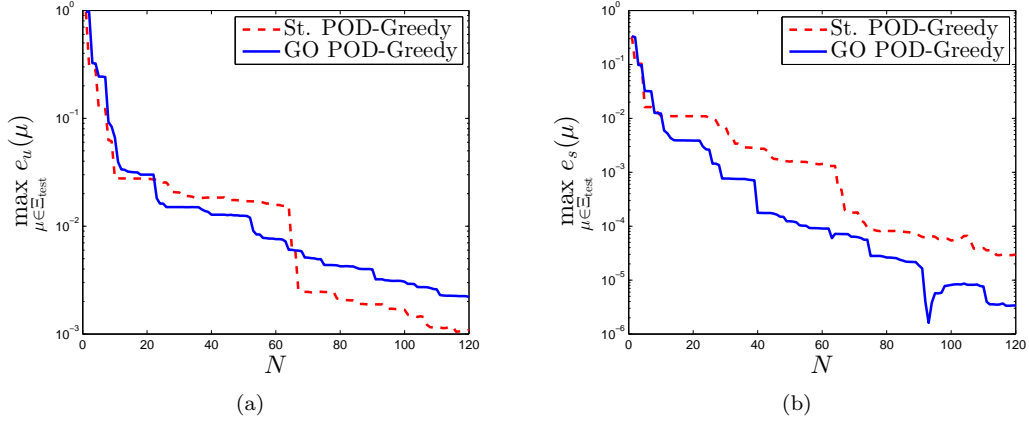


Figure 8: Comparison of maximum RB true errors by standard and goal-oriented POD-Greedy algorithms: (a) solution and (b) output over Ξ_{test} .

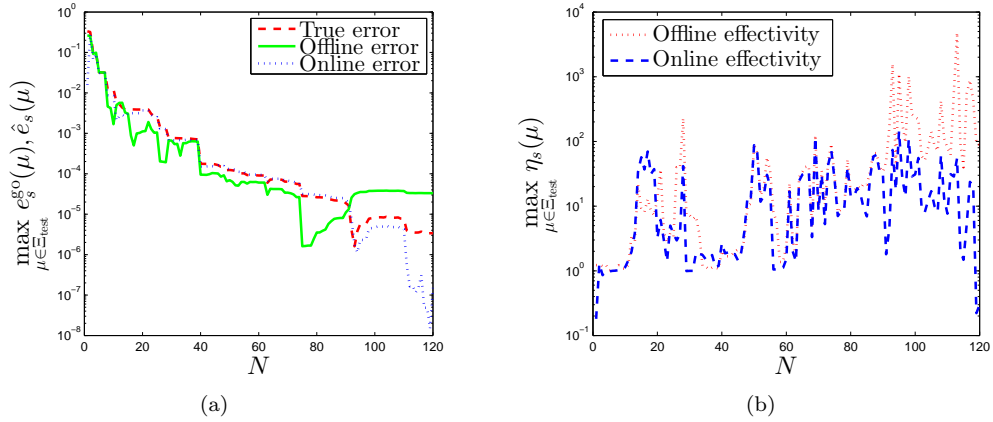


Figure 9: (a) Comparison of max true output error versus max asymptotic output errors, and (b) their corresponding max effectivities over Ξ_{test} .

Acknowledgements

We are sincerely grateful for the financial support of the European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled “Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery”.

References

- [1] Martin A Grepl and Anthony T Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(01):157–181, 2005.

- [2] Gianluigi Rozza, DBP Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.
- [3] KC Hoang, BC Khoo, GR Liu, NC Nguyen, and AT Patera. Rapid identification of material properties of the interface tissue in dental implant systems using reduced basis method. *Inverse Problems in Science and Engineering*, (ahead-of-print):1–25, 2013.
- [4] Anthony T. Patera. RB website (augustine). <http://augustine.mit.edu/methodology.htm>.
- [5] Alex YK Tan. Reduced basis method for 2nd order wave equation: Application to one-dimensional seismic problem. Master’s thesis, Massachusetts Institute of Technology, 2007.
- [6] DB Huynh, David J Knezevic, and Anthony T Patera. A Laplace transform certified reduced basis method; application to the heat equation and wave equation. *Comptes Rendus Mathematique*, 349(7):401–405, 2011.
- [7] K. C. Hoang. *Reduced Basis Approximation and Inverse Analyses for Dental Implant Problems*. PhD thesis, Singapore–MIT Alliance, National University of Singapore, 2012.
- [8] Thomas Grätsch and Klaus-Jürgen Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & structures*, 83(4):235–265, 2005.
- [9] Wolfgang Bangerth and Rolf Rannacher. Finite element approximation of the acoustic wave equation: Error control and mesh adaptation. *East West Journal of Numerical Mathematics*, 7(4):263–282, 1999.
- [10] Wolfgang Bangerth and Rolf Rannacher. Adaptive finite element techniques for the acoustic wave equation. *Journal of Computational Acoustics*, 9(02):575–591, 2001.
- [11] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta numerica*, 10(1):1–102, 2001.
- [12] Wolfgang Bangerth, Michael Geiger, and Rolf Rannacher. Adaptive Galerkin finite element methods for the wave equation. *Computational Methods in Applied Mathematics*, 10:3–48, 2010.
- [13] Marcus Meyer and Hermann G Matthies. Efficient model reduction in non-linear dynamics using the Karhunen-Loeve expansion and dual-weighted-residual methods. *Computational Mechanics*, 31(1):179–191, 2003.
- [14] K. C. Hoang, P. Kerfriden, and S. P. A. Bordas. A goal-oriented reduced basis method for the wave equation in inverse analysis. In *Proceedings of the International Conference on Computational Mechanics (CM13)*, Durham, UK, 25-27 March 2013.
- [15] Lawrence Sirovich. Turbulence and the dynamics of coherent structures, part 1: Coherent structures. *Quarterly of applied mathematics*, 45:561–571, 1987.
- [16] Pierre Kerfriden, Pierre Gosselet, Sondipon Adhikari, Stéphane Bordas, and Jean-Charles Passieux. POD-based model order reduction for the simulation of strong nonlinear evolutions in structures: application to damage propagation. In *IOP Conference Series: Materials Science and Engineering*, volume 10, page 012165. IOP Publishing, 2010.
- [17] Pierre Kerfriden, Jean-Charles Passieux, and Stephane Pierre-Alain Bordas. Local/global model order reduction strategy for the simulation of quasi-brittle fracture. *International Journal for Numerical Methods in Engineering*, 89(2):154–179, 2011.

- [18] Pierre Kerfriden, Pierre Gosselet, Sondipon Adhikari, and Stephane Pierre-Alain Bordas. Bridging proper orthogonal decomposition methods and augmented Newton–Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200(5):850–866, 2011.
- [19] Pierre Kerfriden, Olivier Goury, Timon Rabczuk, and Stephane Pierre-Alain Bordas. A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics. *Computer methods in applied mechanics and engineering*, 2012.
- [20] K Willcox, O Ghattas, B van Bloemen Waanders, and B Bader. An optimization framework for goal-oriented, model-based reduction of large-scale systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 2265–2271. IEEE, 2005.
- [21] T Bui-Thanh, K Willcox, O Ghattas, and B van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896, 2007.
- [22] Jorge Nocedal and Stephen J Wright. *Numerical optimization*, volume 2. Springer New York, 1999.
- [23] M Heinkenschloss. Numerical solution of implicitly constrained optimization problems. *Rice University Department of Computational and Applied Mathematics Tech. Rep*, 2008.
- [24] Bernard Haasdonk and Mario Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(02):277–302, 2008.
- [25] J Knezevic David, Ngoc-Cuong Nguyen, and T Patera Anthony. Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations. *Mathematical Models and Methods in Applied Sciences*, 21(07):1415–1442, 2011.
- [26] WJT Daniel. The subcycled Newmark algorithm. *Computational Mechanics*, 20(3):272–281, 1997.
- [27] Thomas JR Hughes. *The finite element method: Linear static and dynamic finite element analysis*. 1987.
- [28] Khin Zaw, GR Liu, B Deng, and KBC Tan. Rapid identification of elastic modulus of the interface tissue on dental implants surfaces using reduced-basis method and a neural network. *Journal of biomechanics*, 42(5):634–641, 2009.
- [29] NC Nguyen. A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. *Journal of Computational Physics*, 227(23):9807–9822, 2008.
- [30] Ngoc-Cuong Nguyen, Gianluigi Rozza, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers equation. *Calcolo*, 46(3):157–185, 2009.
- [31] Masayuki Yano, Anthony T Patera, and Karsten Urban. A space-time certified reduced basis method for Burgers equation. *Math. Mod. Meth. Appl. S.*, *submitted*, 2012.
- [32] Masayuki Yano. A space-time Petrov-Galerkin certified reduced basis method: Application to the Boussinesq equations. *Submitted to SIAM Journal on Scientific Computing*, 2012.
- [33] S Wang, GR Liu, KC Hoang, and Y Guo. Identifiable range of osseointegration of dental implants through resonance frequency analysis. *Medical engineering & physics*, 32(10):1094–1106, 2010.