

Rapid identification of material properties of the interface tissue in dental implant systems using reduced basis method

K.C. Hoang, B.C. Khoo, G.R. Liu, N.C. Nguyen & A.T. Patera

To cite this article: K.C. Hoang, B.C. Khoo, G.R. Liu, N.C. Nguyen & A.T. Patera (2013) Rapid identification of material properties of the interface tissue in dental implant systems using reduced basis method, *Inverse Problems in Science and Engineering*, 21:8, 1310-1334, DOI: [10.1080/17415977.2012.757315](https://doi.org/10.1080/17415977.2012.757315)

To link to this article: <https://doi.org/10.1080/17415977.2012.757315>



Published online: 21 Jan 2013.



Submit your article to this journal [↗](#)



Article views: 245



View related articles [↗](#)



Citing articles: 7 View citing articles [↗](#)

Rapid identification of material properties of the interface tissue in dental implant systems using reduced basis method

K.C. Hoang^{a*}, B.C. Khoo^a, G.R. Liu^b, N.C. Nguyen^c and A.T. Patera^d

^a*Singapore-MIT Alliance, National University of Singapore, Singapore, Singapore;* ^b*School of Aerospace System, University of Cincinnati, Cincinnati, OH, USA;* ^c*Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA;* ^d*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA*

(Received 11 September 2012; final version received 2 December 2012)

This paper proposes a rapid inverse analysis approach based on the reduced basis (RB) method and the Levenberg–Marquardt–Fletcher algorithm to identify the ‘unknown’ material properties: Young’s modulus and stiffness-proportional Rayleigh damping coefficient of the interfacial tissue between a dental implant and the surrounding bones. In the forward problem, a finite element approximation for a three-dimensional dental implant-bone model is first built. A RB approximation is then established by using a proper orthogonal decomposition – Greedy algorithm and the Galerkin projection to enable extremely fast and reliable computation of displacement responses for a range of material properties. In the inverse analysis, the RB approximation for the dental implant-bone model are incorporated in the Levenberg–Marquardt–Fletcher algorithm to enable rapid identification of the unknown material properties. Numerical results are presented to demonstrate the efficiency and robustness of the proposed method.

Keywords: second-order hyperbolic partial differential equations; reduced basis method; inverse analysis; Levenberg–Marquardt–Fletcher algorithm; material characterization; POD–Greedy algorithm

1. Introduction

Osseointegration is a slow process of structural and functional connection between living bone and dental implant surfaces.[1] In the osseointegration process, the conditions of implant-bone interfacial tissues are very important as they reflect the bone remodelling and the stability of dental implant-bone structures. Understanding these conditions allows clinicians to decide on effective treatments. From a mechanical viewpoint, the material properties of the interfacial tissues are the most important condition indicator as they determine the biomechanical behaviour and stability of implant-bone structures.

A number of methods have been proposed to identify the tissue properties of dental implant-bone structures with in vitro and in vivo studies.[2] Examples are the clinical percussion testing (impact testing),[3, 4] the radio-graphic observation method and the resonance frequency analysis (RFA).[5, 6] Among these methods, the RFA is adopted by most researchers and is extensively used in many dental implant researches to date. [6–9] In the area of non-destructive evaluation, there are other methods which were shown

*Corresponding author. Email: hoangkacchi@gmail.com; hoangk@cardiff.ac.uk

successful with some levels in identifying the tissues properties of dental implant-bone structures. An example of such methods is the inverse analysis method.[10–12] However, either the method was based on the finite element method (FEM) (which is time consuming) [10, 11] or has focused on the frequency-domain (which is not really real-time analysis).[12] Therefore, they are not convenient for clinicians.

The FEM has been widely employed to solve elasticity equations in dental implant studies (e.g. [9, 13, 14]). Although the FEM is a very useful and powerful tool in the inverse analysis context, it can be time consuming because the complexity of implant-bone structures requires a very large number of elements and because many forward problems need to be solved. The total CPU time using FEM can be so long that real-time identification is not possible. A fast forward solver is therefore essential to enable real-time inverse analysis, thereby providing clinicians with the immediate knowledge of the conditions of the implant-bone interfacial tissues.

The reduced basis (RB) method is a model order reduction framework for rapid and reliable evaluation of functional outputs of solutions of parametrized partial differential equations (PDEs). These PDEs depend on an input parameter vector that includes geometry parameters and/or material properties. The RB method was developed for elliptic PDEs,[15, 16] parabolic PDEs,[17] hyperbolic PDEs,[18] the viscous Burgers' equation [19] and the steady incompressible Navier-Stokes equation.[20] Recently, Liu *et al.* developed a RB method for elasticity problems based on a smooth Galerkin projection [21] which can provide an upper bound to the exact solution, while the original RB method provides a lower bound to the exact solution. The computational efficiency of the RB method was demonstrated significantly higher than that of the FEM in the inverse analysis context.[22]

Several methods have been proposed for solving inverse problems in non-destructive evaluation. They include global search algorithms such as the genetic algorithm, simulated annealing, neural network and local gradient-based algorithms such as the gradient descent, Gauss–Newton or Levenberg–Marquardt method. Applications of the neural network for dental implant inverse problems can be found in [10, 11]. Recently, Zaw *et al.* [12] developed a technique to determine non-invasively the material properties of implant-bone interfacial tissues by using the RB method in combination with the neural network in the frequency domain. However, time-domain applications have not yet been considered. In addition, the Levenberg–Marquardt algorithm has not been applied to the dental implant inverse problems, although this algorithm was widely used to solve other important inverse problems.[23–26]

In this paper, we introduce an inverse analysis approach for rapid identification of the material properties of the interfacial tissues. There are two main components in our approach: the RB method and the Levenberg–Marquardt–Fletcher algorithm. We first develop a RB approximation for linear elastodynamics that govern the structural responses of a dental implant-bone model. This is achieved by using a proper orthogonal decomposition (POD)–Greedy algorithm, the Galerkin projection and an offline–online computational procedure. The RB approximation provides extremely fast and reliable calculation of displacement responses for a range of material properties. We then incorporate the RB approximation into the Levenberg–Marquardt–Fletcher algorithm to enable rapid identification of the unknown material properties. Finally, the efficiency and robustness of the proposed method are demonstrated for a real *in vitro* model.

The paper is organized as follows. In Section 2, we introduce a real *in vitro* model and associated finite element approximations. In Section 3, we develop the RB approximation

and present some numerical results. In Section 4, we describe the proposed inverse analysis approach and present numerical results to demonstrate its efficiency and robustness. Finally, we provide some concluding remarks in Section 5.

2. Problem description and finite element approximation

2.1. Models and approximations

2.1.1. The real in vitro model

We consider a real in vitro model shown in Figure 1(a). The bone is made of the bovine rib of a mature specimen obtained commercially. The bone is composed of two subparts: the cortical bone and the cancellous bone. The thickness of the cortical bone is 2 mm. A cylindrical implant socket whose diameter of 6.5 mm and depth of 15 mm is drilled into the bone. A cylindrical dental implant whose diameter of 4 mm and length of 12 mm is inserted into the drilled hole. A layer of 2.5 mm thickness surrounding the dental implant is the interfacial tissue whose material properties need to be identified in the osseointegration process. Finally, a stainless steel screw is screwed tightly into the dental implant. The screw is modelled as a cylinder that has 1.5 mm in diameter and 12.5 mm in length, respectively.

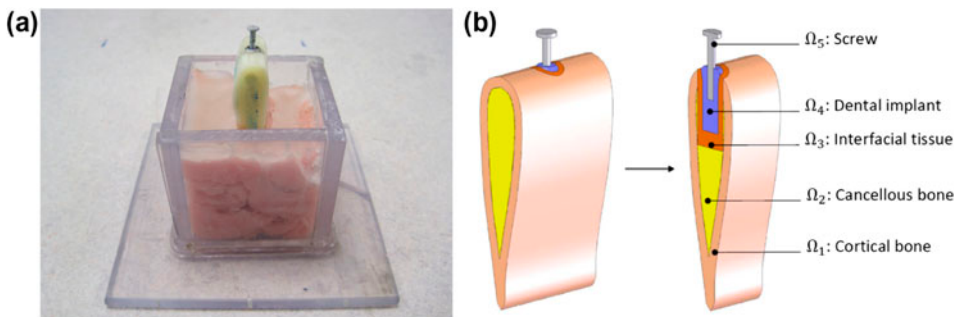


Figure 1. The real in vitro model (a) and the 3d simplified FEM model with sectional view (b).

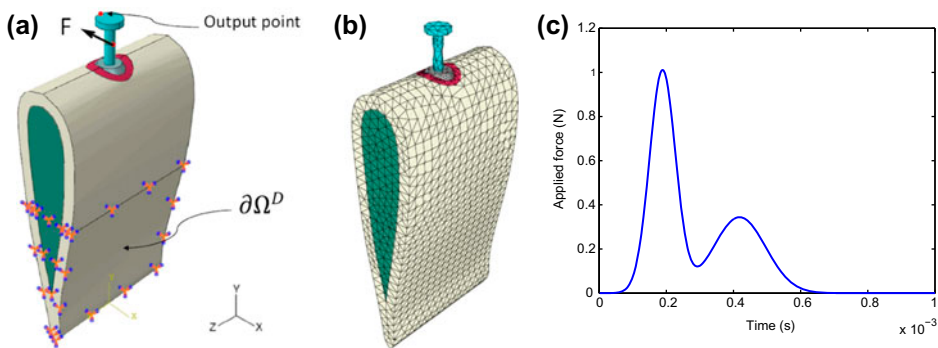


Figure 2. Output point, applied load and boundary conditions (a), meshed model in ABAQUS (b) and time history of load (c).

Table 1. Material properties of the dental implant-bone structure.

Domain	Layers	E (Pa)	ν	ρ (g 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}

2.1.2. The simplified 3d FEM model

Figure 1(b) presents a simplified 3D dental implant-bone model that simulates the real in vitro model shown in Figure 1(a). The geometry of the simplified dental implant-bone model is constructed by using SolidWorks 2005. 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 analysed in the software ABAQUS/CAE version 6.9-1. A dynamic force opposite to the x -direction is then applied to the body of the screw as shown in Figure 2(a). The time history of the applied load is also presented in Figure 2(c). The output of interest is defined as the displacement responses of a point on the head of the screw. The Dirichlet boundary condition ($\partial\Omega^D$) is specified in the bottom-half of the simplified model as illustrated in Figure 2(a). As shown in Figure 2(b) the finite element mesh consists of 9655 nodes and 52585 four-node tetrahedral solid elements. The coinciding nodes of the contact surfaces between different regions (the regions $\Omega_1, \Omega_2, \Omega_3, \Omega_4, \Omega_5$) 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 $\Omega_i, 1 \leq i \leq 5$, of the simplified model are homogeneous and isotropic¹. The material properties: the Young's moduli, Poisson's ratios and densities of these regions are presented in Table 1 [12, 27]. In order to simulate the damping of the system, the Rayleigh damping assumption [28] is used in our analysis. Each region shall have their own pair values of α_i – the mass-proportional damping coefficients which represent the contribution of the mass matrices into the damping matrices of interest and β_i ($i = 1, \dots, 5$) – the stiffness-proportional damping coefficients which represent the contribution of the stiffness matrices into the damping matrices, respectively. We conducted a sensitivity analysis for the values of $\alpha_i, 1 \leq i \leq 5$, to the displacement output and found that they do not affect the displacement output of our problem. This means that our current problem setting is stiffness dominated and the damping being used here, in essence, is Kelvin–Voigt damping. Based on this finding, $\beta_i, 1 \leq i \leq 5$, have the values as presented in the last column of Table 1 such that

$$\mathbf{C}_i = \beta_i \mathbf{A}_i, \quad i = 1, \dots, 5,$$

where \mathbf{C}_i and \mathbf{A}_i are the FEM damping and stiffness matrices of each region, respectively. We also note in Table 1 that the Young's modulus E and the stiffness-proportional Rayleigh damping coefficient β of the region Ω_3 (E_3, β_3) are 'unknown' material parameters that need to be identified.

The 3D simplified dental implant-bone problem is solved by taking two important considerations. Firstly, the loading applied to the head of the screw is extremely small. Hence, the deformation of the whole structure is small and shall be governed by linear elastodynamics.[11–13] Secondly, all layers except the tissue (i.e. the cortical bone, the cancellous bone, the implant and the screw) are very hard and the tissue is the only soft layer considered. Therefore, the output displacement responses of the system are mostly affected by the material properties of the tissue layer only.

Here, we aim to identify the ‘unknown’ material properties of the interfacial tissue, namely the Young’s modulus E and the stiffness-proportional Rayleigh’s damping coefficient β , from the displacement responses of the dental-implant bone structure due to the excitation force. Our analysis procedure consists of two parts: forward analysis and inverse analysis. In the forward analysis, the output displacement responses are determined for a range of input system parameters (E, β) for which we need to build a RB model. The inverse analysis determines $(E_{\text{true}}, \beta_{\text{true}})$ from a given measurement of output displacement response of the dental implant structure when it is excited by the applied load.

2.2. Finite element approximation

2.2.1. Formulations and definitions

We consider a spatial domain $\Omega \in \mathbb{R}^3$ with a boundary $\partial\Omega$. We denote the Dirichlet portion of the boundary by $\Gamma_i^D, 1 \leq i \leq 3$. We then introduce the Hilbert spaces

$$Y^e = \{v \equiv (v_1, v_2, v_3) \in (H^1(\Omega))^3 | v_i = 0 \text{ on } \Gamma_i^D, i = 1, 2, 3\}, \tag{1a}$$

$$X^e = (L^2(\Omega))^3. \tag{1b}$$

Here, $H^1(\Omega) = \{v \in L^2(\Omega) | \nabla v \in (L^2(\Omega))^3\}$ 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

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

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

where the summation over repeated indices is assumed.

We next define our parameter 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 , $m : X^e \times X^e \rightarrow \mathbb{R}$, $c : X^e \times X^e \times \mathcal{D} \rightarrow \mathbb{R}$, $f : X^e \rightarrow \mathbb{R}$ and $\ell : X^e \rightarrow \mathbb{R}$.

The ‘exact’ linear elasticity problem is stated follows: given a parameter $\mu \in \mathcal{D} \subset \mathbb{R}^P$, we evaluate the output of interest

$$s^e(\mu, t) = \ell(u^e(\mu, t)), \quad t \in [0, T], \tag{3}$$

where the field variable $u^e(\mu, t) \in Y^e$ satisfies the weak form of the μ -parametrized hyperbolic PDE

$$m\left(\frac{\partial^2 u^e(\mu, t)}{\partial t^2}, v\right) + c\left(\frac{\partial u^e(\mu, t)}{\partial t}, v; \mu\right) + a(u^e(\mu, t), v; \mu) = g(t)f(v), \quad \forall v \in Y^e, t \in [0, T], \quad (4)$$

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

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 $u(\mu, t) \in Y$ to the ‘exact’ problem² is stated as:

$$m\left(\frac{\partial^2 u(\mu, t)}{\partial t^2}, v\right) + c\left(\frac{\partial u(\mu, t)}{\partial t}, v; \mu\right) + a(u(\mu, t), v; \mu) = g(t)f(v), \quad \forall v \in Y, t \in [0, T], \quad (5)$$

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

$$s(\mu, t) = \ell(u(\mu, t)), \quad t \in [0, T]. \quad (6)$$

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

We shall assume that the bilinear forms $a(\cdot, \cdot; \mu)$ and $m(\cdot, \cdot; \mu)$ are continuous,

$$a(w, v; \mu) \leq \gamma \|w\|_Y \|v\|_Y \leq \gamma_0 \|w\|_Y \|v\|_Y, \quad \forall w, v \in Y, \forall \mu \in \mathcal{D}, \quad (7a)$$

$$m(w, v; \mu) \leq \rho \|w\|_X \|v\|_X \leq \rho_0 \|w\|_X \|v\|_X, \quad \forall w, v \in Y, \forall \mu \in \mathcal{D}, \quad (7b)$$

coercive,

$$0 \leq \alpha_0 \leq \alpha(\mu) \equiv \inf_{v \in Y} \frac{a(v, v; \mu)}{\|v\|_Y^2}, \quad \forall \mu \in \mathcal{D}, \quad (8a)$$

$$0 \leq \sigma_0 \leq \sigma(\mu) \equiv \inf_{v \in Y} \frac{m(v, v; \mu)}{\|v\|_X^2}, \quad \forall \mu \in \mathcal{D}; \quad (8b)$$

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} [17]). We also require that the linear forms $f(v) : Y \rightarrow \mathbb{R}$ and $\ell(v) : Y \rightarrow \mathbb{R}$ be bounded with respect to $\|\cdot\|_Y$ and $\|\cdot\|_X$, respectively.

With respect to our particular dental implant problem described in Section 2.1.2 the actual integral forms of the linear and bilinear forms are defined as:

$$m(w, v) = \sum_{r=1}^5 \int_{\Omega_r} \rho_r w_i v_i, \quad (9a)$$

$$a(w, v; \mu) = \sum_{r=1, r \neq 3}^5 \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l} + \mu_1 \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}, \quad (9b)$$

$$c(w, v; \mu) = \sum_{r=1, r \neq 3}^5 \beta_r \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l} + \mu_2 \mu_1 \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}, \tag{9c}$$

$$f(v) = \int_{\Gamma_n^N} v, \tag{9d}$$

for all $w, v \in Y, \mu \in \mathcal{D}$. Here, the ‘unknown’ parameter $\mu = (\mu_1, \mu_2) \equiv (E, \nu)$ 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_n^N is the point where the load is applied as shown in Figure 2(a). The material properties E_r and $\beta_r, 1 \leq r \leq 5, r \neq 3; \nu_r$ and $\rho_r, 1 \leq r \leq 5$ are defined as in Table 1.

From (9b) and (9c), we find that a and c depend affinely on the parameter μ and that they can be expressed as:

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

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

Here, 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 . Finally, we also require that all linear and bilinear forms be independent of time – the system is thus linear time-invariant.[17]

2.2.2. Time discretization

We shall use the Newmark’s scheme with coefficients $(\varphi = \frac{1}{2}, \psi = \frac{1}{4})$ [29] to approximate the time derivative terms of the ‘true’ statement (5). For time integration: we divide $[0, T]$ into K subintervals of equal lengths $\Delta t = \frac{T}{K}$, and define $t^k = k\Delta t, 0 \leq k \leq K$. Our finite element approximation is then given by:

$$\begin{aligned} & m(u(\mu, t^{k+1}), v) + \frac{1}{2} \Delta t c(u(\mu, t^{k+1}), v; \mu) + \frac{1}{4} \Delta t^2 a(u(\mu, t^{k+1}), v; \mu) \\ & = -m(u(\mu, t^{k-1}), v) + \frac{1}{2} \Delta t c(u(\mu, t^{k-1}), v; \mu) - \frac{1}{4} \Delta t^2 a(u(\mu, t^{k-1}), v; \mu) \\ & + 2m(u(\mu, t^k), v) - \frac{1}{2} \Delta t^2 a(u(\mu, t^k), v; \mu) \\ & + \Delta t^2 g^{eq}(t^k) f(v), \\ & \forall v \in Y, 1 \leq k \leq K - 1, \end{aligned} \tag{11}$$

with

$$g^{eq}(t^k) = \frac{1}{4} g(t^{k-1}) + \frac{1}{2} g(t^k) + \frac{1}{4} g(t^{k+1}), \quad 1 \leq k \leq K - 1. \tag{12}$$

In order to start the procedure (11), $u(\mu, t^1)$ is computed as on page 491 of [28]. With the assumptions of $u(\mu, t^0) = 0$, $\dot{u}(\mu, t^0) = 0$ and $g(t^0) = 0$; then $u(\mu, t^1)$ is computed from: $u(\mu, t^1) = \frac{1}{4} \Delta t^2 \ddot{u}(\mu, t^1)$, where the acceleration $\ddot{u}(\mu, t^1)$ is found from:

$$m(\ddot{u}(\mu, t^1), v) + \frac{1}{2} \Delta t c(\ddot{u}(\mu, t^1), v; \mu) + \frac{1}{4} \Delta t^2 a(\ddot{u}(\mu, t^1), v; \mu) = g(t^1) f(v), \quad \forall v \in Y. \quad (13)$$

We then evaluate the output from:

$$s(\mu, t^k) = \ell(u(\mu, t^k)), \quad 1 \leq k \leq K. \quad (14)$$

3. RB approximation

The RB method is a cooperative method that is constructed from the FEM. The idea of the RB method is that instead of solving a very time-consuming and expensive FEM system for each input parameter, we would rather proceed with two stages: an offline stage and an online stage. In the offline stage – performed only once, we pre-compute several FEM solutions corresponding to several sets of input parameters (the S^* set in the following sections) and use these FEM solutions as basis vectors of the RB spaces (the Y_N space in the following sections). Then, in the online stage – performed numerous times, we can compute very rapidly and accurately the solution/output of the RB system for a required input parameter by taking a proper linear combination (Galerkin projection) of the RB basis vectors. The computational time and computational cost of the online stage are very cheap because necessary operation counts depend only on the dimension N of the RB spaces rather than the dimension \mathcal{N} of the FEM space and $N \ll \mathcal{N}$. Obviously, we will evaluate the error between the RB solution and the FEM solution as well as the error between the RB output and the FEM output to assess the quality of the RB solution/output with respect to the FEM ones. This kind of error, which is called the ‘RB error’ because it is the error induced by the RB approximation, will be discussed in more detail in Section 3.3. One can refer to [30] for more information regarding the RB method and all related aspects.

3.1. RB method

We introduce the nested samples $S^* = \{\mu_1 \in \mathcal{D}, \mu_2 \in \mathcal{D}, \dots, \mu_N \in \mathcal{D}\}$, $1 \leq N \leq N_{\max}$, and associated nested Lagrangian RB spaces $Y_N = \text{span}\{\zeta_n, 1 \leq n \leq N\}$, $1 \leq N \leq N_{\max}$, where $\zeta_n \in Y_N$, $1 \leq n \leq N_{\max}$ are mutually $(\cdot, \cdot)_Y$ – orthonormal RB basis functions. The sets S^* and Y_N shall be constructed appropriately by the POD–Greedy algorithm described in Section 3.2.2 afterward. Our RB approximation $u_N(\mu, t)$ to $u(\mu, t)$ is then obtained by a standard Galerkin projection: given $\mu \in \mathcal{D}$, we now look for $u_N(\mu, t) \in Y_N$ satisfies

$$m\left(\frac{\partial^2 u_N(\mu, t)}{\partial t^2}, v\right) + c\left(\frac{\partial u_N(\mu, t)}{\partial t}, v; \mu\right) + a(u_N(\mu, t), v; \mu) = g(t) f(v), \quad \forall v \in Y_N, t \in [0, T]. \quad (15)$$

We evaluate the associated RB output, $s_N(\mu, t)$, from

$$s_N(\mu, t) = \ell(u_N(\mu, t)), \quad t \in [0, T]. \quad (16)$$

The discrete RB approximation equation of (15) is then given by:

$$\begin{aligned}
 & m(u_N(\mu, t^{k+1}), v) + \frac{1}{2} \Delta t c(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{4} \Delta t^2 a(u_N(\mu, t^{k+1}), v; \mu) \\
 &= -m(u_N(\mu, t^{k-1}), v) + \frac{1}{2} \Delta t c(u_N(\mu, t^{k-1}), v; \mu) - \frac{1}{4} \Delta t^2 a(u_N(\mu, t^{k-1}), v; \mu) \\
 &\quad + 2m(u_N(\mu, t^k), v) - \frac{1}{2} \Delta t^2 a(u_N(\mu, t^k), v; \mu) + \Delta t^2 g^{eq}(t^k) f(v), \\
 &\qquad \qquad \qquad \forall v \in Y_N, 1 \leq k \leq K - 1. \tag{17}
 \end{aligned}$$

Similar to (13), with zero initial conditions: $u_N(\mu, t^0) = 0, \dot{u}_N(\mu, t^0) = 0, g(t^0) = 0$; $u_N(\mu, t^1)$ is calculated from: $u_N(\mu, t^1) = \frac{1}{4} \Delta t^2 \ddot{u}_N(\mu, t^1)$, and the RB acceleration $\ddot{u}_N(\mu, t^1)$ is found from:

$$\begin{aligned}
 & m(\ddot{u}_N(\mu, t^1), v) + \frac{1}{2} \Delta t c(\ddot{u}_N(\mu, t^1), v; \mu) + \frac{1}{4} \Delta t^2 a(\ddot{u}_N(\mu, t^1), v; \mu) = g(t^1) f(v), \\
 &\qquad \qquad \qquad \forall v \in Y_N. \tag{18}
 \end{aligned}$$

Finally, the RB output is evaluated from:

$$s_N(\mu, t^k) = \ell(u_N(\mu, t^k)), \quad 1 \leq k \leq K. \tag{19}$$

3.2. POD–Greedy sampling procedure

In this section, we present briefly the POD method and then introduce our POD–Greedy sampling algorithm used in this work.

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 orthonormal 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:

$$W_M = \arg \min_{V_M \subset \text{span}\{\xi_1, \dots, \xi_{M_{\max}}\}} \left(\frac{1}{M_{\max}} \sum_{k=1}^{M_{\max}} \inf_{\alpha^k \in \mathbb{R}^M} \left\| \xi_k - \sum_{m=1}^M \alpha_m^k v_m \right\|^2 \right). \tag{20}$$

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 [31].

3.2.2. POD–Greedy algorithm

We now discuss our POD–Greedy algorithm [32] to construct the nested sets S^* and Y_N of interest. Let S^* denote the set of greedily selected parameters in \mathcal{D} . 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 algorithm is then defined as follows:

$$\begin{aligned}
 (21a) \quad & \text{Set } Y_N = 0. \\
 (21b) \quad & \text{Set } \mu^* = \mu_0^*. \\
 (21c) \quad & \text{While } N \leq N_{\text{max}} \\
 (21d) \quad & \mathcal{W} = \left\{ e_{\text{proj}}(\mu^*, t^k), 0 \leq k \leq K \right\}; \\
 (21e) \quad & Y_{N+M} \leftarrow Y_N \oplus \text{POD}(\mathcal{W}, M); \\
 (21f) \quad & N \leftarrow N + M; \\
 (21g) \quad & \mu^* = \arg \max_{\mu \in \Xi_{\text{train}}} \left\{ \frac{\sqrt{\sum_{k=1}^K \|R(v; \mu, t^k)\|_{Y'}^2}}{\|u_N(\mu, t^k)\|_Y} \right\}; \\
 (21h) \quad & S^* \leftarrow S^* \cup \{\mu^*\}; \\
 (21i) \quad & \text{end.}
 \end{aligned} \tag{21}$$

Here, M is the number of RB basis functions that are constructed from the set of snapshots \mathcal{W} at each POD-Greedy iteration, and Ξ_{train} is the training parameter set where $\Xi_{\text{train}} \in \mathcal{D}$. The term $\|R(v; \mu, t^k)\|_{Y'}, \forall v \in Y, 1 \leq k \leq K - 1$ is the dual norm of the residual, where the residual is defined from (17) as:

$$\begin{aligned}
 R(v; \mu, t^k) = & g^{eq}(t^k) f(v) - \frac{1}{\Delta t^2} \left(m(u_N(\mu, t^{k+1}), v) - 2m(u_N(\mu, t^k), v) \right. \\
 & \left. + m(u_N(\mu, t^{k-1}), v) \right) \\
 & - \frac{1}{\Delta t} \left(\frac{1}{2} c(u_N(\mu, t^{k+1}), v; \mu) - \frac{1}{2} c(u_N(\mu, t^{k-1}), v; \mu) \right) \\
 & - \left(\frac{1}{4} a(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{2} a(u_N(\mu, t^k), v; \mu) \right. \\
 & \left. + \frac{1}{4} a(u_N(\mu, t^{k-1}), v; \mu) \right).
 \end{aligned} \tag{22}$$

In addition, the dual norm of the residual is defined theoretically as:

$$\|R(v; \mu, t^k)\|_{Y'} \equiv \sup_{v \in Y} \frac{R(v; \mu, t^k)}{\|v\|_Y}, \quad \forall v \in Y, 1 \leq k \leq K - 1. \tag{23}$$

At the step (21g) of the algorithm (21), note that we use the dual norm of the residual as a surrogate for the RB error (define in the next section).

3.3. Errors

3.3.1. RB error

In order to evaluate the efficiency of the RB model relative to the FEM model, RB errors are used in this work. The RB error for the solution $u_N(\mu, t^k)$ is defined as:

$$e(\mu, t^k) = u(\mu, t^k) - u_N(\mu, t^k), \quad 1 \leq k \leq K, \quad (24)$$

where $u(\mu, t^k)$, $u_N(\mu, t^k)$, $1 \leq k \leq K$ are the FEM and RB solutions, respectively. The relative RB error of solution and relative RB error of output are defined as:

$$\epsilon_u(\mu, t^k) = \frac{\|e(\mu, t^k)\|_Y}{\|u_N(\mu, t^k)\|_Y}; \quad \epsilon_s(\mu, t^k) = \left| \frac{s(\mu, t^k) - s_N(\mu, t^k)}{s_N(\mu, t^k)} \right|, \quad 1 \leq k \leq K, \quad (25)$$

respectively. We note in (25) that the final time step t_K is usually preferred since the errors will grow up as time progresses.

3.3.2. Error indicator

Consider the POD–Greedy algorithm (21), we can use the RB error (24) as the error indicator in the step (21g). In that situation the computational time, computational effort and required storage will be huge because we need to solve and store all the FEM solutions of all $\mu \in \Xi_{\text{train}}$; hence the use of RB error is not feasible. Another choice for the error indicator (and also for the error evaluation) is the rigorous *a posteriori* error bound [15, 17]. Tan derived the *a posteriori* error bound for linear hyperbolic PDEs [18]; however, this bound is for the Newmark’s scheme³ ($\varphi = \frac{1}{2}$, $\psi = \frac{1}{2}$) and is thus not applicable for our work. Recently, Huynh *et al.* [33] used the Laplace transform technique to derive a new *a posteriori* error bound for linear hyperbolic PDEs. The technique improves the situation but also introduces much additional complication.

As analyzed above, we need to have another error indicator since both the error bound and RB error are not available for our particular problem. Therefore, in order to implement the POD–Greedy strategy we use the dual norm of the residual $\|R(v; \mu, t^k)\|_{Y'}$ as a surrogate. The dual norm of the residual is actually not rigorous because it does not include stability information, i.e. some temporal terms as present in the full error bound of [18]. However, there are three advantages in using the dual norm of the residual. Firstly, it is nearly the only remaining choice; secondly, it can evaluate relatively the accuracy of the RB solutions for various choices of μ ; and thirdly – most important, its calculation is very convenient: fast and efficient offline–online decomposition for many μ computations as required in the Greedy strategy. Furthermore, in the next section we will show that the operation counts to find the dual norm of the residual (22) for one particular μ is very cheap – roughly $O(KN^2Q^2 + KN^2Q + KNQ)$, where $Q = Q_a + Q_c$.

3.4. Offline–online computational procedure

In this section, we develop an offline–online computational procedure in order to fully exploit the dimension reduction of the problem. We first express $u_N(\mu, t^k)$ as:

$$u_N(\mu, t^k) = \sum_{n=1}^N u_{Nn}(\mu, t^k) \zeta_n, \quad \forall \zeta_n \in Y_N. \quad (26)$$

We then choose a test functions $v = \zeta_n, 1 \leq n \leq N$ for the discrete RB equation (17). It then follows from (17) that $\underline{u}_N(\mu, t^k) = [u_{N1}(\mu, t^k) \ u_{N2}(\mu, t^k) \ \dots \ u_{NN}(\mu, t^k)]^T \in \mathbb{R}^N$ satisfies

$$\begin{aligned} & \left(\mathbf{M}_N + \frac{1}{2} \Delta t \mathbf{C}_N(\mu) + \frac{1}{4} \Delta t^2 \mathbf{A}_N(\mu) \right) \underline{u}_N(\mu, t^{k+1}) \\ &= \left(-\mathbf{M}_N + \frac{1}{2} \Delta t \mathbf{C}_N(\mu) - \frac{1}{4} \Delta t^2 \mathbf{A}_N(\mu) \right) \underline{u}_N(\mu, t^{k-1}) \\ &+ \left(2\mathbf{M}_N - \frac{1}{2} \Delta t^2 \mathbf{A}_N(\mu) \right) \underline{u}_N(\mu, t^k) \\ &+ \Delta t^2 g^{eq}(t^k) \mathbf{F}_N, \end{aligned} \tag{27}$$

$1 \leq k \leq K - 1.$

The initial condition is treated similar to the treatment in (13) and (18). Here, $\mathbf{C}_N(\mu), \mathbf{A}_N(\mu), \mathbf{M}_N \in \mathbb{R}^{N \times N}$ are symmetric positive definite matrices with entries $\mathbf{C}_{Ni,j}(\mu) = c(\zeta_i, \zeta_j; \mu), \mathbf{A}_{Ni,j}(\mu) = a(\zeta_i, \zeta_j; \mu), \mathbf{M}_{Ni,j} = m(\zeta_i, \zeta_j), 1 \leq i, j \leq N$ and $\mathbf{F}_N \in \mathbb{R}^N$ is the RB load vector with entries $\mathbf{F}_{Ni} = f(\zeta_i), 1 \leq i \leq N.$

The RB output is then computed from:

$$s_N(\mu, t^k) = \mathbf{L}_N^T \underline{u}_N(\mu, t^k), \quad 1 \leq k \leq K. \tag{28}$$

Invoking the affine parameter dependence (10), we obtain

$$\mathbf{A}_{Ni,j}(\mu) = a(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(\zeta_i, \zeta_j), \tag{29a}$$

$$\mathbf{C}_{Ni,j}(\mu) = c(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_c} \Theta_c^q(\mu) c^q(\zeta_i, \zeta_j), \tag{29b}$$

which can be written as:

$$\mathbf{A}_{Ni,j}(\mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) \mathbf{A}_{Ni,j}^q, \quad \mathbf{C}_{Ni,j}(\mu) = \sum_{q=1}^{Q_c} \Theta_c^q(\mu) \mathbf{C}_{Ni,j}^q, \tag{30}$$

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

$$\mathbf{A}_{Ni,j}^q = a^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_a, \tag{31a}$$

$$\mathbf{C}_{Ni,j}^q = c^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_c, \tag{31b}$$

respectively.

The computational procedure is now clear with two stages: an offline stage and an online stage. In the offline stage – performed only once, we solve for the $\zeta_n, 1 \leq n \leq N_{\max};$ we then compute and store the μ -independent quantities in (31), (A6) and (A8) for the estimation of the output and the dual norm of the residual. We consider each POD–Greedy iteration (21) in more details. We first need to solve (11) for the ‘true’ FE solutions; then do the error projection in step (21d) and solve the POD/eigenvalue problem in step (21e). In addition, we have to compute $O(N^2Q)$ \mathcal{N} -inner products $(\cdot, \cdot)_Y$ in (31); $O(NQ + N)$

pseudo \mathcal{N} -solutions in (A6) and roughly $O(N^2 Q^2 + N^2 Q + N Q)$ \mathcal{N} -inner products $(\cdot, \cdot)_Y$ in (A8) for the dual norm estimations. Since there are totally $\frac{N_{\max}}{M}$ POD–Greedy iterations, the above calculations are thus multiplied by $\frac{N_{\max}}{M}$ times. In summary, for the offline stage, the operation counts depend on \mathcal{N} and hence, its computational cost is very expensive.

In the online stage—performed many times, for each new parameter μ —we first assemble the RB matrices in (29), this requires $O(N^2 Q)$ operations. We then solve the RB governing equation (27), the operation counts are $O(N^3 + K N^2)$ as the RB matrices are generally full. Finally, we evaluate the displacement output $s_N(\mu, t^k)$ from (28) at the cost of $O(K N)$. For the dual norm of the residual, the operation counts to gather all offline terms and calculate the norm as in (A7) are roughly $O(K N^2 Q^2 + K N^2 Q + K N Q)$. Therefore, as required in real-time context, the online complexity is 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.

3.5. Numerical results

We now turn to the 3D simplified FEM dental implant-bone model created in Section 2.1.2. The ‘true’ finite element approximation space is of dimension $\mathcal{N} = 26802$. For time integration, $T = 1 \times 10^{-3}$ s, $\Delta t = 2 \times 10^{-6}$ s, $K = \frac{T}{\Delta t} = 500$. The input parameter μ is defined by E and β : $\mu \equiv (E, \beta) \in \mathcal{D}$, where $\mathcal{D} = [1.0 \times 10^6, 15 \times 10^6] \text{Pa} \times [5 \times 10^{-6}, 5 \times 10^{-5}] \subset \mathbb{R}^{P=2}$. The $\|\cdot\|_Y$ used in this work is defined as $\|w\|_Y^2 = a(w, w; \bar{\mu}) + m(w, w; \bar{\mu})$, where $\bar{\mu} = (8 \times 10^6 \text{Pa}, 2.75 \times 10^{-5})$ is the arithmetic average of μ in \mathcal{D} ; $Q_a = 2$, $Q_c = 2$. To verify our computational code (performed in Matlab R2007a), we first compare the FEM outputs computed by ABAQUS and by our code with the test parameter $\mu_{\text{test}} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$. Figure 3 shows the output displacement responses in the x -, y - and z -directions vs. time at μ_{test} via ABAQUS and our code, respectively. Figure 3(a) and (c) demonstrate that the FEM results by our code match very well with the results computed by ABAQUS. However, due to machine errors, there are some small differences between the ABAQUS results and ours as shown in Figure 3(b). In our dental implant problem, since the applied load is opposite to the x -direction, the x -component of the output displacement response is the most important among the three components (i.e. x -, y - and z -components). Hence, for the remaining discussion the ‘output displacement response’ refers only to the x -component of the output displacement.

The POD–Greedy algorithm is then implemented to create the RB spaces $Y_N = \{\zeta_n, 1 \leq n \leq N\}$, $1 \leq N \leq N_{\max}$. The algorithm is actually the POD in the time space and Greedy in the parameter space. We choose $M = 5$ (in step (21e)) for each Greedy iteration and $N_{\max} = 60$ to terminate the iteration procedure⁴. A sample set Ξ_{train} is created randomly by a uniform distribution over \mathcal{D} with $n_{\text{train}} = 1000$ samples. The distribution of the sample set S^* is illustrated in Figure 4(a). We show, as a function of N : $\epsilon_u^{\max, \text{rel}}$ is the maximum over Ξ_{train} of $\epsilon_u(\mu, t^K)$ and $\epsilon_s^{\max, \text{rel}}$ is the maximum over Ξ_{train} of $\epsilon_s(\mu, t^K)$ in Figure 4(b). The comparison of $s(\mu_{\text{test}}, t)$ vs. $s_N(\mu_{\text{test}}, t)$ for various choices of N are presented in Figure 5, respectively. The numerical results demonstrate that our RB errors are acceptably small, and that the convergence rate is fast for a $N \ll \mathcal{N}$.

All computations were performed on a desktop with processor Intel(R) Core(TM)2 Duo CPU E8200 @2.66GHz 2.66GHz, RAM 3.25GB, 32-bit Operating System. The computation time for the RB forward solver ($t_{\text{RB(online)}}$), the CPU-time for the FEM forward solver by our code (t_{FEM}) and by ABAQUS (t_{ABAQUS}), and the CPU-time saving factor

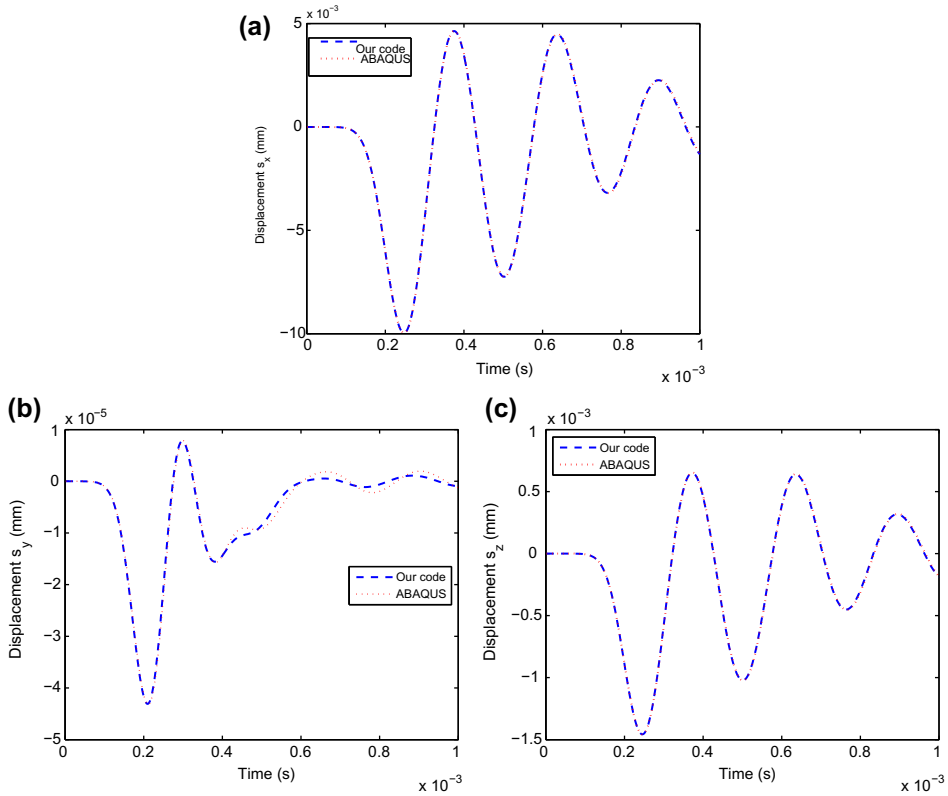


Figure 3. Comparison of the FEM output displacement responses computed by our code vs. by ABAQUS software with respect time in the x - (a), y - (b), and z -direction (c) with $\mu_{\text{test}} = (10 \times 10^6 \text{ Pa}, 1 \times 10^{-5})$.

$\kappa = t_{\text{FEM}}/t_{\text{RB(online)}}$ are listed on Table 2, respectively. We observe that while the FEM forward solvers (i.e. our code and ABAQUS) take a thousand of seconds to compute the output displacement responses, the RB solver (with various choices of N) takes less than 1 second to find the approximation displacement response with known accuracy. Thus, it is clear that the RB is very efficient and reliable for solving forward problems. Next,

Table 2. Comparison of the CPU-time for a FEM, RB and ABAQUS forward analysis.

N	$t_{\text{RB(online)}} \text{ (s)}$	$t_{\text{FEM}} \text{ (s)}$	$t_{\text{ABAQUS}} \text{ (s)}$	$\kappa = t_{\text{FEM}}/t_{\text{RB(online)}}$
10	0.1947	3750	2010	1.9260×10^4
20	0.2312	3750	2010	1.6220×10^4
30	0.2969	3750	2010	1.2631×10^4
40	0.3405	3750	2010	1.1013×10^4

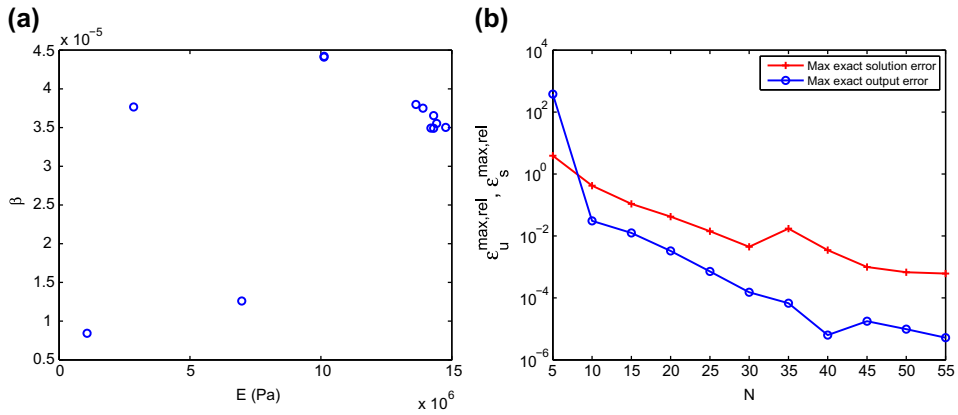


Figure 4. Distribution of sampling points with the POD–Greedy procedure (a) and the maximum relative RB error of the solution and the output as functions of N (b).

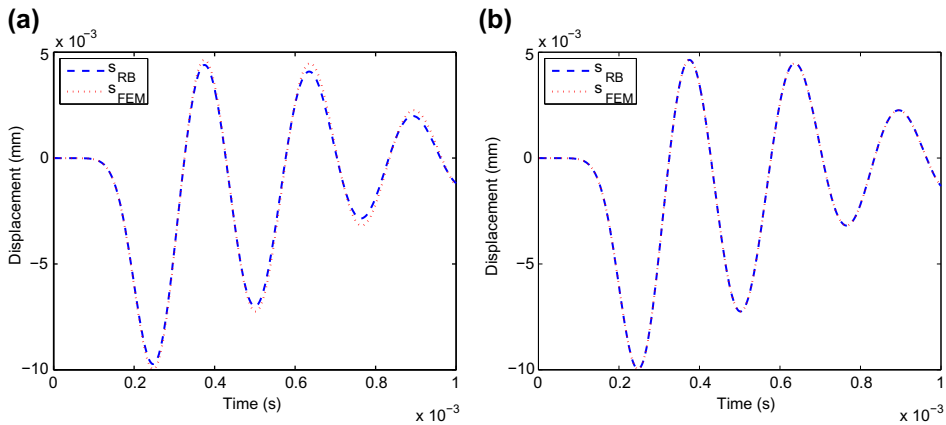


Figure 5. Comparison of output displacement responses by FEM and RB with $\mu_{\text{test}} = (10 \times 10^6 \text{ Pa}, 1 \times 10^{-5})$ with $N = 2$ (a) and $N = 10$ (b) basis functions.

our RB model is now ready to be utilized as an efficient forward solver in the inverse analysis.

4. Inverse procedure

Here, we establish an inverse procedure using our RB model in combination with the Levenberg–Marquardt–Fletcher algorithm to identify rapidly the elastic modulus E and the stiffness Rayleigh damping coefficient β of the interfacial tissue in our dental implant–bone structure.

4.1. The Levenberg–Marquardt–Fletcher (LMF) algorithm

The considered inverse problem is concerned with the simultaneous estimation of two parameter components: the Young’s modulus E and the stiffness Rayleigh damping coefficient β of the interfacial tissue from the ‘measured’ displacement response at the output point (Figure 2(a)). This inverse problem can be regarded as an optimization problem which aims at finding the unknown parameter⁵ $\mu^* = (E^*, \beta^*)$ that minimizes the following objective function or cost function

$$S(\mu) = \sum_{i=1}^K [s_{N,i}(\mu) - s_i^{measure}]^2 = \mathbf{r}^T \mathbf{r}, \tag{32}$$

where

$$r_i(\mu) = s_{N,i}(\mu) - s_i^{measure}. \tag{33}$$

Here, K is the total number of discrete time steps; $s_{N,i}(\mu)$ is the ‘computed’ RB displacement response defined in (16) at the time t_i with the parameter μ ; $s_i^{measure}$ is the simulated ‘measured’ displacement response at the time t_i and $\mu = (E, \beta)$ is the input material property parameter.

The parameter μ^* which minimizes the objective function S must satisfy the following set of non-linear algebraic equations:

$$\sum_{i=1}^K 2 \frac{\partial s_{N,i}}{\partial \mu_j} (s_{N,i} - s_i^{measure}) = 2 \frac{\partial \mathbf{r}^T}{\partial \mu_j} \mathbf{r} = 2 \mathbf{J}^T \mathbf{r} = 0, \quad j = 1, 2. \tag{34}$$

The set of equations (34) is obtained by differentiating (32) with respect to each component of the parameter μ and then setting those derivatives equal to zero. The matrix \mathbf{J} is called the Jacobian matrix whose entries are defined as: $J_{ij} = \frac{\partial r_i}{\partial \mu_j}$, $1 \leq i \leq K$, $j = 1, 2$. In order to solve the system (34), the Levenberg–Marquardt–Fletcher iterative method [34, 35] is used. The update equation of the parameter μ at $(l + 1)$ th iteration has the form:

$$\mu^{(l+1)} = \mu^{(l)} + \Delta \mu^{(l)}, \tag{35a}$$

$$\Delta \mu^{(l)} = -(\mathbf{J}^{(l)T} \mathbf{J}^{(l)} + \lambda^{(l)} \mathbf{D})^{-1} \mathbf{J}^{(l)T} \mathbf{r}^{(l)}. \tag{35b}$$

The solution of the inverse problem starts with a suitable guess $\mu^{(0)}$, and the iteration procedure is continued until

$$|\mu_j^{(l+1)} - \mu_j^{(l)}| < \varepsilon, \quad j = 1, 2, \tag{36}$$

where ε is a prescribed tolerance. The entries of the Jacobian matrix \mathbf{J} can be calculated from the following finite difference formula

$$\frac{\partial r_i(\mu)}{\partial \mu_j} \approx \frac{r_i(\mu + \epsilon U_j) - r_i(\mu)}{\epsilon}, \tag{37}$$

where $U_j = [\delta_{1j}, \delta_{2j}]^T$, δ denotes the Kronecker delta symbol and ϵ is a small number.

In order to check the uniqueness of solutions of the posed inverse problem, a study on local minima is carried out by plotting the function $S(\mu)$ vs. a wide range of both parameter components. Note that we use $N = 40$ basis functions in all computations related to our RB model in the inverse analysis (i.e. computations of $s_N(\mu)$ in (32)). We present on Figure 6

the plot of the logarithm of the function $S(\mu)$ vs. both parameter components E and β in the parameter domain \mathcal{D} . In these plots, we choose $\mu^{\text{measure}} \equiv (8 \times 10^6 \text{Pa}, 8 \times 10^{-6})$ for $S(\mu)$ as defined in (32). The parameter μ is taken in the parameter domain \mathcal{D} , whereby \mathcal{D} is discretized uniformly into a rectangular grid of (100×100) points. The plots confirm that with a particular $\mu^{\text{measure}} \equiv (8 \times 10^6 \text{Pa}, 8 \times 10^{-6})$ the function $S(\mu)$ has a unique global minimum point which is exactly the μ^{measure} itself. This also confirms the absence of other local minima (if any) of the function $S(\mu)$, and hence ensuring the uniqueness of solutions of our inverse problem.

In the remaining sections, the open-source code [35] with appropriate modifications is used to implement our RB-LMF algorithm.

4.2. Numerical results

4.2.1. Effects of E and β to output displacement s_N

We use $N = 40$ basis functions in all computations related to our RB model in this inverse analysis. The effects of both the Young's modulus E and the stiffness Rayleigh damping coefficient β to the displacement response s_N are plotted in Figure 7(a) and (b), respectively. As observed, the Young's modulus E dominates the width between two consecutive peaks of the displacement response curves (Figure 7(a)), while the coefficient β controls the height of these peaks (Figure 7(b)). It is shown that the output displacement responses are very sensitive to these two parameter components.

4.2.2. Synthetic data

To verify our RB-LMF procedure, the simulated 'measured' displacement responses are used as the input information. A simulated measured displacement response s^{measure} is generated by adding a Gaussian noise term to the displacement response $s_N(\mu^{\text{measure}})$ as

$$s^{\text{measure}} = s_N(\mu^{\text{measure}}) + \omega\sigma, \tag{38}$$

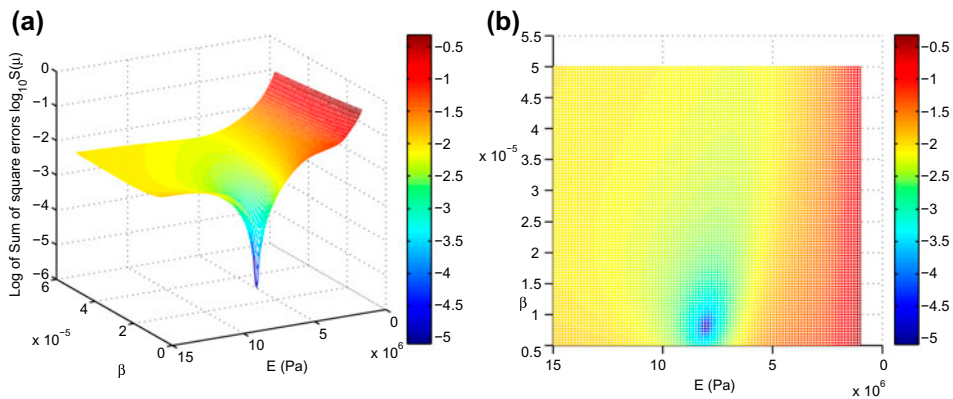


Figure 6. Logarithm of the function $S(\mu)$ over the parameter domain \mathcal{D} of parameter components E and β with (a) an overall 3D view and (b) a xy-view.

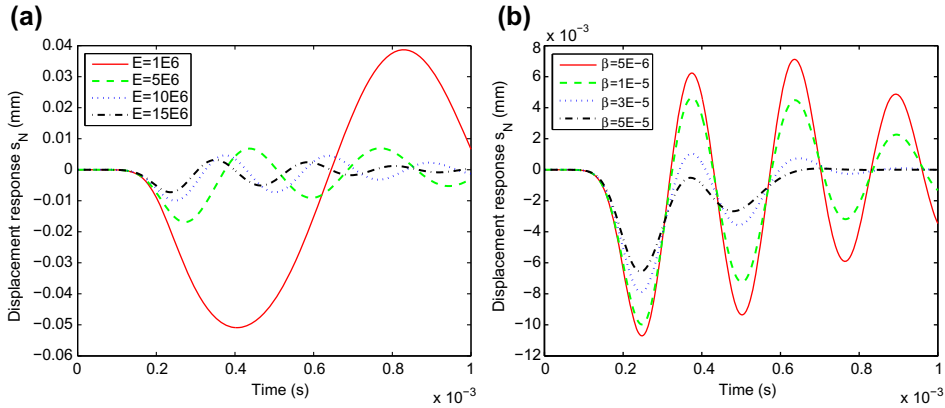


Figure 7. Effects of the Young’s modulus E (with $\beta = 1 \times 10^{-5}$) (a) and effects of the stiffness Rayleigh damping coefficient β (with $E = 10 \times 10^6\text{Pa}$) (b) on displacement responses.

where ω is a vector which contains random variables with zero mean and unit variance normal [24]; the standard deviation σ computed as in [36]

$$\sigma = p_e \left(\frac{1}{K - 1} \sum_{k=1}^K (s_N(\mu^{\text{measure}}, t^k))^2 \right)^{1/2}. \quad (39)$$

Here, p_e is the noise level (for example, $p_e = 0.05$ means a 5% noise level), K is the total number of time steps and s_N is the usual RB displacement response, respectively.

4.2.3. Parameter estimation

As an estimation example, we choose $\mu^{\text{measure}} = (8 \times 10^6\text{Pa}, 8 \times 10^{-6})$ to test our RB-LMF procedure. The lowest value of $\mu \in \mathcal{D}$: $\mu^{(0)} = (1 \times 10^6\text{Pa}, 5 \times 10^{-6})$ is chosen to be the initial guess – which is also independent of μ^{measure} . We first create a number of noisy (or contaminated) outputs $\{s^{\text{measure}}\}$ as defined in (38), then we use the LMF algorithm to find

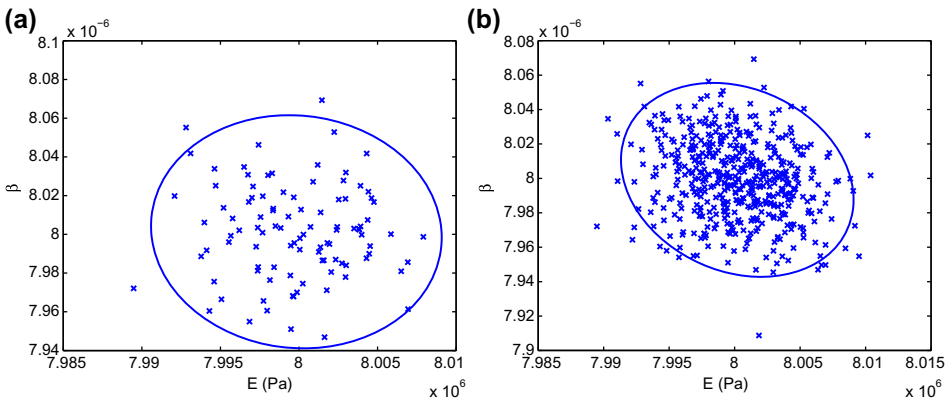


Figure 8. 95% confidence ellipses and computed parameter samples of (a) 100 random tests and (b) 500 random tests.

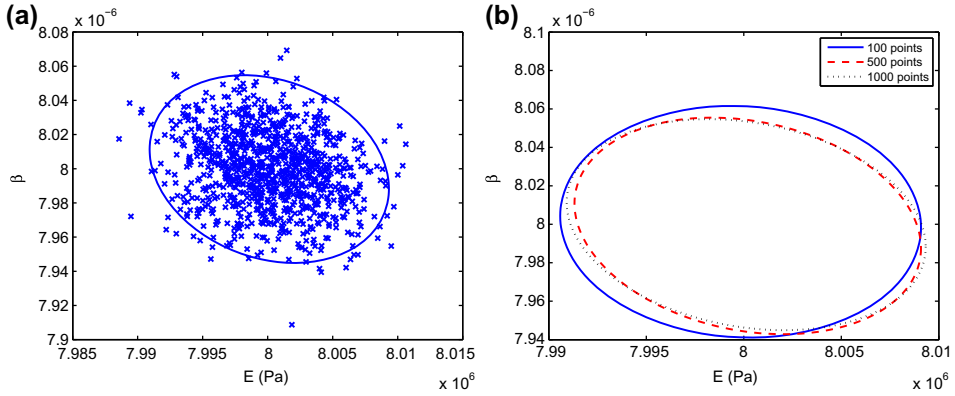


Figure 9. 95% confidence ellipse and computed parameter samples of 1000 random tests (a) and 95% confidence ellipses of all three cases.

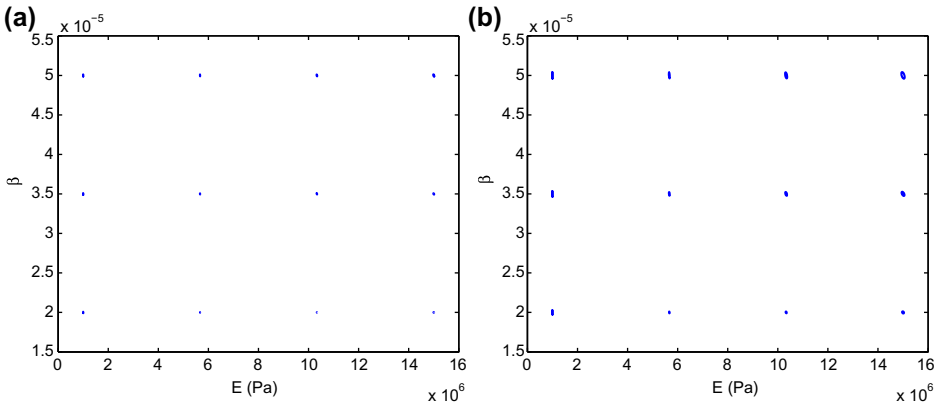


Figure 10. 95% confidence ellipses of the sample set S^{true} with $p_e = 1\%$ (a) and $p_e = 3\%$ (b) noise added.

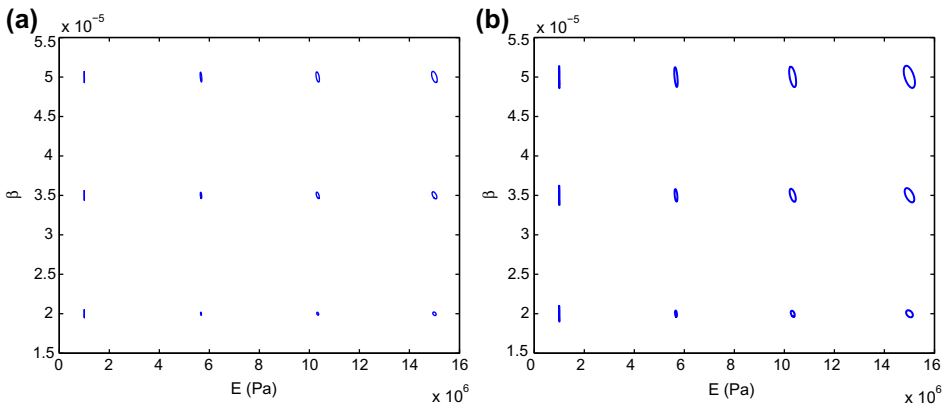


Figure 11. 95% confidence ellipses of the sample set S^{true} with $p_e = 5\%$ (a) and $p_e = 10\%$ (b) noise added.

Table 3. Total number of forward analyses required in the RB–LMF inverse analysis (for one particular μ^{measure}).

Average number of iterations	Number of RB calls in each iteration	Total RB calls
30	3	$m = 90$

Table 4. Comparison of computational time for the LMF model using FEM and RB as forward solvers (for one particular μ^{measure}).

Total RB calls	CPU time for each solver	Total computation time		
$m = 90$	t_{FEM}	3750 (s)	$m \times t_{\text{FEM}}$	93.75 (h)
	$t_{\text{RB(online)}}$	0.3405 (s)	$m \times t_{\text{RB(online)}}$	35.65 (s)

the corresponding set $\{\mu^{\text{compute}}\}$. This set of computed parameters is then covered by a 95% confidence ellipse which is drawn using the principal components analysis method.[37] For a $p_e = 3\%$ noise level, the estimation results which are the 95% confidence ellipses and the computed parameter samples of 100, 500 and 1000 random tests are plotted in Figures 8 and 9(a), respectively. These figures show that the computed parameter samples converge around the center point $\mu^{\text{measure}} = (8 \times 10^6 \text{Pa}, 8 \times 10^{-6})$ and that the ellipses' shapes are nearly similar as the number of random tests increase (Figure 9(b)). Hence, 500 random tests are sufficient to construct reliable 95% confidence ellipses for all test cases afterwards.

In order to test with many parameters, a sample set S^{true} of regular (4×3) grid pattern over $[1.0 \times 10^6, 15 \times 10^6] \text{Pa} \times [2 \times 10^{-5}, 5 \times 10^{-5}]$ is created. We then implement the RB–LMF procedure with the fixed initial guess $\mu^{(0)} = (1 \times 10^6 \text{Pa}, 5 \times 10^{-6})$ and 500 random tests for each true parameter point in the grid. We show the plots of 95% confidence ellipses of each true parameter point with added noise levels $p_e = 1\%, 3\%, 5\%$ and 10% in Figures 10 and 11, respectively. The accuracy and suitability of obtained results show that at lower noise of displacement responses, reliable estimates can be provided by this procedure.

To validate the efficiency of the RB–LMF procedure, total forward solver calls for the RB–LMF inverse analysis are given in Table 3; the total CPU time is recorded and provided in Table 4. It is found that the CPU time of the LMF model using the RB solver⁶ is significantly faster than the one using the FEM solver. Therefore, the proposed RB–LMF approach strongly reduces the computational time and cost.

5. Conclusion

In this paper, a rapid inverse procedure (RB–LMF) is established, which consists of two main stages: constructing a fast elastodynamic RB model and determining inversely material properties via the Levenberg–Marquardt–Fletcher algorithm. We applied the RB–LMF approach to a specific 3D simplified dental implant–bone structure. In the RB stage, the results show that the RB model is very efficient and reliable. In the inverse analysis, the identified results of the RB–LMF approach are very accurate and fast for all test cases: noise-free, contaminated noise, one true parameter, various true parameters. The results of

our example support our conclusion that the computational efficiency is greatly increased due to the use of the RB, and that the RB–LMF approach is able to model the non-linear relations between the structural parameters and the non-static responses of complex dental implant structures.

Acknowledgements

This work was supported by the Singapore–MIT Alliance. We thank Dr. Bernard Haasdonk for revising critically the manuscript. We also thank the reviewers for valuable comments which improve the quality of this paper.

Notes

1. This assumption would introduce modelling error and approximation error (due to sharp interfaces) and that functionally graded materials (FGM) would be a more realistic assumption. Since this paper is our first attempt on the development of the RB method for elastic wave equations and its application to the inverse dental implant problem, we consider this simplified 3D model to demonstrate the usefulness of the RB method for certain applications. The FGM assumption approach will be another subject for future work.
2. Generally, for our dental implant problem u and u^e (and hence s and s^e) are different due to various environmentally contaminated noise. The simulation treatment of this noise will be described in Section 4.2.2.
3. In this work, note that we use the Newmark's scheme ($\varphi = \frac{1}{2}$, $\psi = \frac{1}{4}$) as described in Section 2.2.2.
4. In fact, the POD–Greedy algorithm behaves unstably when $N \geq 60$ basis functions, i.e. the maximum error curves (in Figure 4(b)) will go up with $N \geq 60$. This is because of the high instability of the dual norm of the residual $\|R(v; \mu, t^k)\|_{Y'}$ and the ill conditions of the RB matrices A_N and C_N with larger N . However, as observed from Figure 4(b) we already obtain acceptably small RB error even with $N = 30$ basis functions and hence, we chose to terminate the procedure with $N_{\max} = 60$ basis functions.
5. Here, the vector μ should be typed in bold as $\boldsymbol{\mu}$, we use the mediumface italic for suitability with previous sections.
6. The work focuses on the real-time context with many online computations, the offline stage is done once and expensive: its computational time is approximately 16 h on the same computer described in Section 3.5.

References

- [1] Branemark PI, Zarb GA, Albrektsson T. Tissue-integrated prostheses: osseointegration in clinical dentistry. Chicago (IL): Quintessence; 1985.
- [2] Cowin SC. Bone mechanics handbook. Vol. 56. Boca Raton (FL): CRC Press; 2001.
- [3] Olivé J, Aparicio C. Periotest method as a measure of osseointegrated oral implant stability. Int. J. Oral Maxillofac. Implants. 1990;5:390–400.
- [4] Swain R, Faulkner G, Raboud J, Wolfaardt J. A dynamic analytical model for impact evaluation of percutaneous implants. J. Biomech. Eng. 2008;130:13–26.
- [5] Sennerby L, Meredith N. Implant stability measurements using resonance frequency analysis: biological and biomechanical aspects and clinical implications. Periodontology 2000. 2008;47:51–66.
- [6] Huang HM, Pan LC, Lee SY, Chiu CL, Fan KH, Ho KN. Assessing the implant/bone interface by using natural frequency analysis. Oral. Surg. Oral. Med. Oral. Pathol. Oral. Radiol. Endod. 2000;90:285–291.

- [7] Lachmann S, Jäger B, Axmann D, Gomez-Roman G, Groten M, Weber H. Resonance frequency analysis and damping capacity assessment. Part 1: an in vitro study on measurement reliability and a method of comparison in the determination of primary dental implant stability. *Clin. Oral Impl. Res.* 2006;17:75–79.
- [8] Lachmann S, Jäger B, Axmann D, Gomez-Roman G, Groten M, Weber H. Resonance frequency analysis and damping capacity assessment. Part 2: peri-implant bone loss follow-up. An in vitro study with the PeriotestTM and OsstellTM instruments. *Clin. Oral Impl. Res.* 2006;17:80–84.
- [9] Capek L, Simunek A, Slezak R, Dzan L. Influence of the orientation of the Osstell[®] transducer during measurement of dental implant stability using resonance frequency analysis: a numerical approach. *Med. Eng. Phys.* 2009;31:764–769.
- [10] Deng B, Han X, Liu GR, Tan KBC. Prediction of elastic properties of the maxillary bone. *Proceeding of WCCM VI in conjunction with APCOM'04, Computational Mechanics*; 2004.
- [11] Deng B, Tan KBC, Lu Y, Zaw K, Zhang J, Liu GR, Geng JP. Inverse identification of elastic modulus of dental implant-bone interfacial tissue using neural network and FEA model. *Inverse Probl. Sci. Eng.* 2009;17:1073–1083.
- [12] Zaw K, Liu GR, Deng B, Tan KBC. Rapid identification of elastic modulus of the interface tissue on dental implants surfaces using reduced-basis method and a neural network. *J. Biomech.* 2009;42:634–641.
- [13] Natali AN, Meroi EA, Williams KR, Calabrese L. Investigation of the integration process of dental implants by means of a numerical analysis of dynamic response. *Dent. Mater.* 1997;13:325–332.
- [14] Deng B, Tan KBC, Liu GR. Influence of osseointegration degree and pattern on resonance frequency in the assessment of dental implant stability using finite element analysis. *Int. J. Oral Maxillofac. Implants.* 2008;23:1082–1088.
- [15] Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for finely parametrized elliptic coercive partial differential equations - application to transport and continuum mechanics. *Arch. Comput. Methods Eng.* 2008;15:229–275.
- [16] Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for stress intensity factors. *Int. J. Numer. Methods Eng.* 2007;72:1219–1259.
- [17] Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM. Math. Model. Numer. Anal. (M2AN)*. 2005;39:157–181.
- [18] Tan AYK. Reduced basis methods for 2nd order wave equation: application to one dimensional seismic problem [Master thesis]. Singapore-MIT Alliance, National University of Singapore; 2006.
- [19] Nguyen NC, Rozza G, Patera AT. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo.* 2009;46:157–185.
- [20] Veroy K, Patera AT. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Methods Fluids.* 2005;47:773–788.
- [21] Liu GR, Zaw K, Wang YY, Deng B. A novel reduced-basis method with upper and lower bounds for real-time computation of linear elasticity problems. *Comput. Methods Appl. Mech. Eng.* 2008;198:269–279.
- [22] Liu GR, Lee JH, Patera AT, Yang ZL, Lam KY. Inverse identification of thermal parameters using reduced-basis method. *Comput. Methods Appl. Mech. Eng.* 2005;194:3090–3107.
- [23] Sawaf B, Özisik MN, Jarny Y. An inverse analysis to estimate linearly temperature dependent thermal conductivity components and heat capacity of an orthotropic medium. *Int. J. Heat Mass Transfer.* 1995;38:3005–3010.
- [24] Tang DW, Araki N. An inverse analysis to estimate relaxation parameters and thermal diffusivity with a universal heat conduction equation. *Int. J. Thermophys.* 2000;21:553–561.

- [25] Schnur DS, Zabarás N. An inverse method for determining elastic material properties and a material interface. *Int. J. Numer. Methods Eng.* 1992;33:2039–2057.
- [26] Zhao KM, Lee JK. Inverse estimation of material properties for sheet metals. *Commun. Numer. Methods Eng.* 2004;20:105–118.
- [27] Wang S, Liu GR, Hoang KC, Guo Y. Identifiable range of osseointegration of dental implants through resonance frequency analysis. *Med. Eng. Phys.* 2010;32:1094–1106.
- [28] Hughes TJR. *The finite element method: linear static and dynamic finite element analysis.* Englewood Cliffs (NJ): Prentice-Hall; 1987.
- [29] Daniel WJT. The subcycled Newmark algorithm. *Comput. Mech.* 1997;20:272–281.
- [30] Patera AT. RB website (augustine); 2007. Available from: <http://augustine.mit.edu/methodology.htm>
- [31] Nguyen NC. A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. *J. Comput. Phys.* 2008;227:9807–9822.
- [32] Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM. Math. Model. Numer. Anal. (M2AN)*. 2008;42:277–302.
- [33] Huynh DBP, Knezevic DJ, Patera AT. A Laplace transform certified reduced basis method; application to the heat equation and wave equation. *C. R. Math.* 2011;349:401–405.
- [34] Fletcher R. A modified Marquardt subroutine for nonlinear least squares. *Tech. Rep. AERE-R-6799.* Harwell (UK): Atomic Energy Research Establishment; 1971.
- [35] Balda M. An algorithm for nonlinear least squares. Institute of Thermomechanics, Academy of Sciences of the Czech Republic, v.v.i., 2010. Available from: http://dsp.vscht.cz/konference_matlab/MATLAB07/prispevky/balda_m/balda_m.pdf
- [36] Xu YG, Liu GR. Detection of flaws in composites from scattered elastic-wave field using an improved μ GA and a local optimizer. *Comput. Methods Appl. Mech. Eng.* 2002;191:3929–3946.
- [37] Jackson JE. *A User's Guide to Principal Components.* Hoboken (NJ): John Wiley & Sons; 1991.

Appendix. Calculation of the dual norm of the residual

In this section, we present explicitly the calculation of the dual norm of the residual associated with the discrete RB equation (17) discretized by the Newmark’s scheme ($\varphi = \frac{1}{2}, \psi = \frac{1}{4}$). We consider the residual defined in (22) and its dual norm given in (23). The dual norm of the residual can be computed alternatively as:

$$\begin{aligned} \|R(v; \mu, t^k)\|_{Y'} &\equiv \sup_{v \in Y} \frac{R(v; \mu, t^k)}{\|v\|_Y}, \\ &= \|\hat{e}(\mu, t^k)\|_Y, \quad 1 \leq k \leq K - 1, \end{aligned} \tag{A1}$$

where $\hat{e}(\mu, t^k) \in Y$ is given by the Riesz representation:

$$(\hat{e}(\mu, t^k), v)_Y = R(v; \mu, t^k), \quad \forall v \in Y, 1 \leq k \leq K - 1. \tag{A2}$$

From (22), (26) and the affine property (10) it thus follows that $\hat{e}(\mu, t^k)$ satisfies

$$\begin{aligned} (\hat{e}(\mu, t^k), v)_Y &= g^{eq}(t^k) f(v) \\ &+ \sum_{n=1}^N \frac{-u_{Nn}(\mu, t^{k+1}) + 2u_{Nn}(\mu, t^k) - u_{Nn}(\mu, t^{k-1})}{\Delta t^2} m(\zeta_n, v) \\ &+ \sum_{n=1}^N \sum_{q=1}^{Q_c} \Theta_c^q(\mu) \frac{-u_{Nn}(\mu, t^{k+1}) + u_{Nn}(\mu, t^{k-1})}{2\Delta t} c^q(\zeta_n, v) \\ &+ \sum_{n=1}^N \sum_{q=1}^{Q_a} \Theta_a^q(\mu) \frac{-u_{Nn}(\mu, t^{k+1}) - 2u_{Nn}(\mu, t^k) - u_{Nn}(\mu, t^{k-1})}{4} a^q(\zeta_n, v). \end{aligned} \tag{A3}$$

It is clear from linear superposition that we can express $\hat{e}(\mu, t^k)$ as:

$$\begin{aligned} \hat{e}(\mu, t^k) &= g^{eq}(t^k) \mathcal{F} + \sum_{n=1}^N \lambda_{m,n}(\mu, t^k) \mathcal{M}_n \\ &+ \sum_{n=1}^N \sum_{q=1}^{Q_c} \Theta_c^q(\mu) \lambda_{c,n}(\mu, t^k) \mathcal{C}_{q,n} + \sum_{n=1}^N \sum_{q=1}^{Q_a} \Theta_a^q(\mu) \lambda_{a,n}(\mu, t^k) \mathcal{A}_{q,n}. \end{aligned} \tag{A4}$$

Here,

$$\begin{aligned} \lambda_{m,n}(\mu, t^k) &= \frac{-u_{Nn}(\mu, t^{k+1}) + 2u_{Nn}(\mu, t^k) - u_{Nn}(\mu, t^{k-1})}{\Delta t^2}, \\ \lambda_{c,n}(\mu, t^k) &= \frac{-u_{Nn}(\mu, t^{k+1}) + u_{Nn}(\mu, t^{k-1})}{2\Delta t}, \\ \lambda_{a,n}(\mu, t^k) &= \frac{-u_{Nn}(\mu, t^{k+1}) - 2u_{Nn}(\mu, t^k) - u_{Nn}(\mu, t^{k-1})}{4}; \end{aligned} \tag{A5}$$

and the parameter-independent terms $\mathcal{F}, \mathcal{M}, \mathcal{C}, \mathcal{A}$ are calculated from:

$$\begin{aligned} \mathcal{F} \in Y &\quad \text{from} \quad (\mathcal{F}, v)_Y = f(v), \quad \forall v \in Y, \\ \mathcal{M}_n \in Y &\quad \text{from} \quad (\mathcal{M}_n, v)_Y = m(\zeta_n, v), \quad \forall v \in Y \text{ for } 1 \leq n \leq N, \\ \mathcal{C}_{q,n} \in Y &\quad \text{from} \quad (\mathcal{C}_{q,n}, v)_Y = c^q(\zeta_n, v), \quad \forall v \in Y \text{ for } 1 \leq q \leq Q_c, 1 \leq n \leq N, \\ \mathcal{A}_{q,n} \in Y &\quad \text{from} \quad (\mathcal{A}_{q,n}, v)_Y = a^q(\zeta_n, v), \quad \forall v \in Y \text{ for } 1 \leq q \leq Q_a, 1 \leq n \leq N. \end{aligned} \tag{A6}$$

From (A1) and (A4) it follows that

$$\begin{aligned}
 \|\hat{e}(\mu, t^k)\|_Y^2 &= (\hat{e}(\mu, t^k), \hat{e}(\mu, t^k))_Y \\
 &= g^{eq}(t^k)g^{eq}(t^k)\Lambda^{ff} \\
 &+ 2 \sum_{n=1}^N g^{eq}(t^k) \left\{ \lambda_{m,n}(\mu, t^k)\Lambda_n^{fm} + \sum_{q=1}^{Q_c} \Theta_c^q(\mu)\lambda_{c,n}(\mu, t^k)\Lambda_{qn}^{fc} \right. \\
 &\quad \left. + \sum_{q=1}^{Q_a} \Theta_a^q(\mu)\lambda_{a,n}(\mu, t^k)\Lambda_{qn}^{fa} \right\} \\
 &+ \sum_{n,n'=1}^N \left\{ \lambda_{m,n}(\mu, t^k)\lambda_{m,n'}(\mu, t^k)\Lambda_{nn'}^{mm} + 2 \sum_{q=1}^{Q_c} \Theta_c^q(\mu)\lambda_{c,n}(\mu, t^k)\lambda_{m,n'}(\mu, t^k)\Lambda_{qnn'}^{mc} \right. \\
 &\quad + 2 \sum_{q=1}^{Q_a} \Theta_a^q(\mu)\lambda_{a,n}(\mu, t^k)\lambda_{m,n'}(\mu, t^k)\Lambda_{qnn'}^{ma} \\
 &\quad + \sum_{q,q'=1}^{Q_c} \Theta_c^q(\mu)\Theta_c^{q'}(\mu)\lambda_{c,n}(\mu, t^k)\lambda_{c,n'}(\mu, t^k)\Lambda_{qmq'n'}^{cc} \\
 &\quad + 2 \sum_{q=1}^{Q_c} \sum_{q'=1}^{Q_a} \Theta_c^q(\mu)\lambda_{c,n}(\mu, t^k)\Theta_a^{q'}(\mu)\lambda_{a,n'}(\mu, t^k)\Lambda_{qmq'n'}^{ca} \\
 &\quad \left. + \sum_{q,q'=1}^{Q_a} \Theta_a^q(\mu)\lambda_{a,n}(\mu, t^k)\Theta_a^{q'}(\mu)\lambda_{a,n'}(\mu, t^k)\Lambda_{qmq'n'}^{aa} \right\}, \tag{A7}
 \end{aligned}$$

where the parameter-independent quantities Λ are defined as

$$\begin{aligned}
 \Lambda^{ff} &= (\mathcal{F}, \mathcal{F})_Y, \\
 \Lambda_n^{fm} &= (\mathcal{F}, \mathcal{M}_n)_Y, \quad 1 \leq n \leq N, \\
 \Lambda_{qn}^{fc} &= (\mathcal{F}, \mathcal{C}_{q,n})_Y, \quad 1 \leq q \leq Q_c, 1 \leq n \leq N, \\
 \Lambda_{qn}^{fa} &= (\mathcal{F}, \mathcal{A}_{q,n})_Y, \quad 1 \leq q \leq Q_a, 1 \leq n \leq N, \\
 \Lambda_{nn'}^{mm} &= (\mathcal{M}_n, \mathcal{M}_{n'})_Y, \quad 1 \leq n, n' \leq N, \\
 \Lambda_{qnn'}^{mc} &= (\mathcal{M}_{n'}, \mathcal{C}_{q,n})_Y, \quad 1 \leq q \leq Q_c, 1 \leq n, n' \leq N, \\
 \Lambda_{qnn'}^{ma} &= (\mathcal{M}_{n'}, \mathcal{A}_{q,n})_Y, \quad 1 \leq q \leq Q_a, 1 \leq n, n' \leq N, \\
 \Lambda_{qmq'n'}^{cc} &= (\mathcal{C}_{q,n}, \mathcal{C}_{q',n'})_Y, \quad 1 \leq q, q' \leq Q_c, 1 \leq n, n' \leq N, \\
 \Lambda_{qmq'n'}^{ca} &= (\mathcal{C}_{q,n}, \mathcal{A}_{q',n'})_Y, \quad 1 \leq q \leq Q_c, 1 \leq q' \leq Q_a, 1 \leq n, n' \leq N, \\
 \Lambda_{qmq'n'}^{aa} &= (\mathcal{A}_{q,n}, \mathcal{A}_{q',n'})_Y, \quad 1 \leq q, q' \leq Q_a, 1 \leq n, n' \leq N.
 \end{aligned} \tag{A8}$$