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

Citation

As Published
http://dx.doi.org/10.1080/17415977.2012.757315

Publisher
Taylor & Francis

Version
Author’s final manuscript

Citable link
http://hdl.handle.net/1721.1/79067

Terms of Use
Creative Commons Attribution-Noncommercial-Share Alike 3.0

Detailed Terms
http://creativecommons.org/licenses/by-nc-sa/3.0/
RESEARCH ARTICLE

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 and A. T. Patera

\( ^{a} \) National University of Singapore, Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576
\( ^{b} \) School of Aerospace System, University of Cincinnati, Cincinnati, OH 45221-0070, USA
\( ^{c} \) Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, Room 37-471, 77 Mass. Ave., Cambridge, MA02139, USA
\( ^{d} \) Massachusetts Institute of Technology, Dept. of Mechanical Engineering, Room 3-266, 77 Mass. Ave., Cambridge, MA02139, USA

(Received 00 Month 200x; final version received 00 Month 200x)

This paper proposes a rapid inverse analysis approach based on the reduced basis 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 reduced basis approximation is then established by using a Proper Orthogonal Decomposition (POD)–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 reduced basis 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 the living bone and dental implant surface [1]. In the osseointegration process, conditions of implant-bone interfacial tissues are very important as they reflect the bone remodelling and the stability of the dental implant-bone structure. Understanding the conditions allows the clinician to decide on an effective treatment. From the mechanical viewpoint, the material properties of the interfacial tissues are the most important condition indicator as they determine the biomechanical behavior and stability of the implant-bone structures.

A number of methods have been proposed to identify tissue properties of dental implant-bone structure with in vitro and in vivo studies [2]. Examples are 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 research to date [6–9]. In the area of nondestructive evaluation, there are other methods which have been shown 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 has been based on the finite element method (which is time-consuming) [10, 11] or has focused on the frequency-domain (which is not really a real-time analysis) [12]. Therefore, they have not been convenient for the clinician.

The finite element method (FEM) has been widely employed to solve elasticity equations in dental implant (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 the clinician with an 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 solution of parametrized partial differential equations (PDEs). These PDEs depend on an input parameter vector that include geometry parameters and/or material properties. The RB method has been developed for elliptic PDEs [15, 16], parabolic PDEs [17], hyperbolic PDEs [18], viscous Burgers’ equation [19], and steady-state and time-dependent incompressible Navier-Stokes equations [20]. Recently, Liu et al. developed a reduced basis 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 has been demonstrated significantly higher than that of the FEM in the inverse analysis context [22].

Several methods have been proposed for solving inverse problems in nondestructive evaluation. They include direct search algorithms, gradient-base algorithms, genetic algorithms, neural network. Applications of neural network for the dental implant inverse problem can be found in [10, 11]. Recently, Zaw et al. [12] have developed a technique to determine noninvasively the material properties of implant-bone interfacial tissues by using the RB method in combination with the neural network in 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 problem, although this algorithm has been widely used to solve other 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 reduced basis approximation for linear elastodynamics that governs the structural response of the 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 approximation. In Section 3, we develop the reduced basis 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 Fig.1a. 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 2mm. A cylindrical implant socket of \( \phi_1 6.5 \text{mm} \times 15 \text{mm} \) is drilled into the bone. A cylindrical dental implant of \( \phi_2 4 \text{mm} \times 12 \text{mm} \) is inserted into the drilled hole. A layer of 2.5mm 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 modeled as a cylinder of \( \phi_3 1.5 \text{mm} \times 12.5 \text{mm} \).

2.1.2. The simplified 3d FEM model

Fig.1b presents a simplified 3D dental implant-bone model that simulates the real in vitro model shown in Fig.1a. The geometry of the simplified dental implant-bone model is constructed by using SolidWorks 2005. The physical domain \( \Omega \) consists of five regions: the outermost cortical bone \( \Omega_1 \), the cancellous bone \( \Omega_2 \), the interfacial tissue \( \Omega_3 \), the dental implant \( \Omega_4 \) and the stainless steel screw \( \Omega_5 \). The 3D simplified model is then meshed and analyzed in the software ABAQUS/CAE version 6.9-1. A dynamic force opposite to x-direction is then applied to the body of the screw as shown in Fig.2a. The time history of the applied load is also presented in Fig.2c. The output of interest is defined as displacement 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 Fig.2a. As shown in Fig.2b 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 \( x-, y- \) 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 isotropic. The material properties: 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, Rayleigh damping [28] is used in our analysis. Each region shall have their own pair values of \( \alpha_i \) and \( \beta_i (i = 1, \ldots, 5) \): the mass-proportional and stiffness-proportional Rayleigh damping coefficients. We have conducted a sensitivity analysis 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. Based on this finding, \( \beta_i, 1 \leq i \leq 5 \), have values as presented in the last column of Table 1 such that

\[
C_i = \beta_i A_i, \quad i = 1, \ldots, 5,
\]

where \( C_i \) and \( A_i \) are the FEM damping and stiffness matrices of each region,
respectively. We also note that in Table 1, the Young’s modulus $E$ and the stiffness-proportional Rayleigh damping coefficient $\beta$ of the region 3 ($E_3, \beta_3$) 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 structure is small and governed by linear elastodynamics [11–13]. Secondly, all layers except the tissue (i.e., the cortical, the cancellous, the implant and the screw) are very hard and the tissue is the only soft layer considered. Therefore, the response displacement output is mostly affected by the material properties of the tissue layer.

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 $\beta$, 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 of system parameter ($E$, $\beta$) 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 boundary $\partial \Omega$. We denote the Dirichlet portion of the boundary by $\Gamma^D_i, 1 \leq i \leq 3$. We then introduce the Hilbert spaces

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

$$X^e = (L^2(\Omega))^3. \quad (1b)$$

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 $\Omega$. 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

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

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

where summation over repeated component indices is assumed.

We next define our parameter set $D \in \mathbb{R}^p$, a typical point in which shall be denoted $\mu \equiv (\mu_1, \ldots, \mu_p)$. We then define the parametrized bilinear forms $a$ in $Y^e$, $a : Y^e \times Y^e \times D \to \mathbb{R}$; $m, c, f, \ell$ are $X^e$– continuous bilinear and linear forms in $X^e$, $m : X^e \times X^e \to \mathbb{R}$, $c : X^e \times X^e \times D \to \mathbb{R}$, $f : X^e \to \mathbb{R}$ and $\ell : X^e \to \mathbb{R}$.

The “exact” linear elasticity problem is stated follow: given a parameter $\mu \in D \subset \mathbb{R}^p$, we evaluate the output of interest
\[
\hat{s}(\mu, t) = \ell(\hat{u}(\mu, t)),
\]

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

\[
m \left( \frac{\partial^2 \hat{u}(\mu, t)}{\partial t^2}, v \right) + c \left( \frac{\partial \hat{u}(\mu, t)}{\partial t}, v; \mu \right) + a (\hat{u}(\mu, t), v; \mu) = g(t)f(v),
\]

\( \forall v \in Y^e, t \in [0, T], \)

with initial condition \( \hat{u}(\mu, t^0) = 0, \frac{\partial \hat{u}(\mu, t^0)}{\partial t} = 0. \)

We next introduce a reference finite element approximation space \( Y \subset Y^e(\subset X_e) \) of dimension \( N \); 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 “truth” finite element approximation \( u(\mu, t) \in Y \) to the “exact” problem is stated as:

\[
m \left( \frac{\partial^2 u(\mu, t)}{\partial t^2}, v \right) + c \left( \frac{\partial u(\mu, t)}{\partial t}, v; \mu \right) + a (u(\mu, t), v; \mu) = g(t)f(v),
\]

\( \forall v \in Y, t \in [0, T], \)

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

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

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:

\[
m(w, v) = \sum_{r=1}^{5} \int_{\Omega_r} \rho_r w_i v_i,
\]

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

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

\[
f(v) = \int_{\Gamma^N} v,
\]
In the “unknown” parameter \( \mu = (\mu_1, \mu_2) \equiv (E, \beta) \) belongs to region \( \Omega_3. \) \( C_{ijkl} \) is the constitutive elasticity tensor for isotropic material and it is expressed in terms of the Young’s modulus \( E \) and Poisson’s ratio \( \nu \) of each region, respectively. \( \Gamma^N \) is the point where the load is applied as shown in Fig.2a. 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 (7b) and (7c), we find that \( a \) and \( c \) depend affinely on parameter \( \mu \) and they can be expressed as:

\[
a(w, v; \mu) = \sum_{q=1}^{Q_\mu} \Theta_a^q(\mu)a^q(w, v), \quad \forall w, v \in Y, \mu \in \mathcal{D}, \tag{8a}
\]

\[
c(w, v; \mu) = \sum_{q=1}^{Q_\mu} \Theta_c^q(\mu)c^q(w, v), \quad \forall w, v \in Y, \mu \in \mathcal{D}. \tag{8b}
\]

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 \( \mu. \) But the bilinear forms \( a^1(w, v) = \sum_{r=1, r \neq 3}^5 \int_{\Omega_\mu} \frac{\partial w}{\partial x_i} C_{ijkl} \frac{\partial v}{\partial x_j}, a^2(w, v) = \int_{\Omega_\mu} \frac{\partial w}{\partial x_i} C_{ijkl} \frac{\partial v}{\partial x_j}; c^1(w, v) = \sum_{r=1, r \neq 3}^5 \beta_r \int_{\Omega_\mu} \frac{\partial w}{\partial x_i} C_{ijkl} \frac{\partial v}{\partial x_j} \) and \( c^2(w, v) = \int_{\Omega_\mu} \frac{\partial w}{\partial x_i} C_{ijkl} \frac{\partial v}{\partial x_j}, \) do not depend on \( \mu. \) We also require that all linear and bilinear forms are independent of time – the system is thus linear time-invariant (LTI) [17].

### 2.2.2. Time discretization

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

\[
m(u(\mu, t^{k+1}), v) + \frac{1}{2}\Delta t c(u(\mu, t^{k+1}), v; \mu) + \frac{1}{4}\Delta t^2 a(u(\mu, t^{k+1}), v; \mu)
\]

\[
= -m(u(\mu, t^{k-1}), v) + \frac{1}{2}\Delta t c(u(\mu, t^{k-1}), v; \mu) - \frac{1}{4}\Delta t^2 a(u(\mu, t^{k-1}), v; \mu)
\]

\[
+ 2m(u(\mu, t^k), v) - \frac{1}{2}\Delta t^2 a(u(\mu, t^k), v; \mu) + \Delta t^2 g^e(t^k) f(v), \quad \forall v \in Y, 1 \leq k \leq K-1,
\]

with

\[
g^e(t^k) = \frac{1}{4}g(t^{k-1}) + \frac{1}{2}g(t^k) + \frac{1}{4}g(t^{k+1}), \quad 1 \leq k \leq K - 1. \quad \tag{10}
\]

The initial solutions are computed as in [28], \( u(\mu, t^0) = 0, u(\mu, t^1) \) is computed from: \( u(\mu, t^1) = \frac{1}{4}\Delta t^2 \ddot{u}(\mu, t^1), \) where the initial acceleration \( \ddot{u}(\mu, t^1) \) is found from\(^1\):

\(^1\)This initial solutions treatment is only true with zero initial condition: \( u(\mu, t^0) = \ddot{u}(\mu, t^0) = \dot{u}(\mu, t^0) = 0.\)
satisfies by a standard Galerkin projection: given \( n \leq m \)

\[
m\left(\frac{\partial^2 u_N(\mu, t^1)}{\partial t^2}, v\right) + c\left(\frac{\partial u_N(\mu, t)}{\partial t}, v; \mu\right) + a(u_N(\mu, t), v; \mu) = g(t) f(v), \quad \forall v \in Y.
\]

(11)

We then evaluate the output from:

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

(12)

3. Reduced Basis Approximation

3.1. Reduced basis method

We are given a set of mutually \((\cdot, \cdot)_Y\) orthonormal basis functions \( \zeta_n \in Y, 1 \leq n \leq N_{\text{max}} \), the reduced basis spaces are given by \( Y_N = \text{span} \{ \zeta_n, 1 \leq n \leq N \}, 1 \leq N \leq N_{\text{max}} \). Our reduced basis approximation \( u_N(\mu, t) \) to \( u(\mu, t) \) is then obtained by a standard Galerkin projection: given \( \mu \in D \), we now look for \( u_N(\mu, t) \in Y_N \) satisfies

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

(13)

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

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

(14)

The discrete RB approximation equation of (13) is then given by

\[
m(u_N(\mu, t^{k+1}), v) + \frac{1}{2} \Delta t c(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{4} \Delta t^2 a(u_N(\mu, t^{k+1}), v; \mu) \\
= -m(u_N(\mu, t^{k-1}), v) + \frac{1}{2} \Delta t c(u_N(\mu, t^{k-1}), v; \mu) - \frac{1}{4} \Delta t^2 a(u_N(\mu, t^{k-1}), v; \mu) \\
+ 2m(u_N(\mu, t^k), v) - \frac{1}{2} \Delta t^2 a(u_N(\mu, t^k), v; \mu) + \Delta t^2 g^a(t^k) f(v), \quad \forall v \in Y_N, 1 \leq k \leq K-1.
\]

(15)

The initial condition is calculated as: \( u_N(\mu, t^0) = 0, u_N(\mu, t^1) \) is computed from: \( u_N(\mu, t^1) = \frac{1}{4} \Delta t^2 \bar{u}_N(\mu, t^1) \), and \( \bar{u}_N(\mu, t^1) \) is found from:

\[
m(\bar{u}_N(\mu, t^1), v) + \frac{1}{2} \Delta t c(\bar{u}_N(\mu, t^1), v; \mu) + \frac{1}{4} \Delta t^2 a(\bar{u}_N(\mu, t^1), v; \mu) = g(t) f(v), \quad \forall v \in Y_N.
\]

(16)

Finally, the RB output is evaluated from:

\[
s_N(\mu, t^k) = \ell(u_N(\mu, t^k)), \quad 1 \leq k \leq K.
\]

(17)
3.2. POD–Greedy sampling procedure

In this section, we present briefly the Proper Orthogonal Decomposition (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 \( M_{\text{max}} \geq M \) snapshots \( \{ \xi_k \}_{k=1}^{M_{\text{max}}} \). To do this, let \( V_M = \text{span}\{ v_1, \ldots, v_M \} \subset \text{span}\{ \xi_1, \ldots, \xi_{M_{\text{max}}} \} \) 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 \( \delta_{nm} \) is the Kronecker delta symbol). The POD space, \( W_M = \text{span}\{ \zeta_1, \ldots, \zeta_M \} \) is defined as

\[
W_M = \arg \min_{V_M \subset \text{span}\{ \xi_1, \ldots, \xi_{M_{\text{max}}} \}} \left( \frac{1}{M_{\text{max}}} \sum_{k=1}^{M_{\text{max}}} \inf_{\alpha \in \mathbb{R}^M} \| \xi_k - \sum_{m=1}^{M} \alpha_m^k v_m \|_2^2 \right). \tag{18}
\]

The POD space \( W_M \) which is extracted from the given set of snapshots \( \{ \xi_k \}_{k=1}^{M_{\text{max}}} \) is the space that best approximate this given set of snapshots and can be written as \( W_M = \text{POD}(\{ \xi_1, \ldots, \xi_{M_{\text{max}}} \}, M) \). We can construct this POD space by using the method of snapshots which is presented concisely in the Appendix of [30].

3.2.2. POD–Greedy algorithm

We now discuss our POD–Greedy algorithm [31]. Let \( S^* \) denote the set of greedily selected parameters in \( D \). Initialize \( S^* = \{ \mu_0^* \} \), where \( \mu_0^* \) 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:

\[
\begin{align*}
(19a) & \quad \text{Set } Y_N = 0. \\
(19b) & \quad \text{Set } \mu^* = \mu_0^*. \\
(19c) & \quad \text{While } N \leq N_{\text{max}} \\
(19d) & \quad \mathcal{W} = \left\{ e_{\text{proj}}(\mu^*, t^k), 0 \leq k \leq K \right\}; \\
(19e) & \quad Y_{N+M} \leftarrow Y_N \bigoplus \text{POD}(\mathcal{W}, M); \\
(19f) & \quad N \leftarrow N + M; \\
(19g) & \quad \mu^* = \arg \max_{\mu \in D} \left\{ \sqrt{\sum_{k=1}^{K} \| R(v; \mu, t^k) \|_{Y^*}^2} \right\}; \\
(19h) & \quad S^* \leftarrow S^* \cup \{ \mu^* \}; \\
(19i) & \quad \text{end.}
\end{align*}
\]

Here, \( M \) is the number of RB basis functions that are constructed from the set of snapshots \( \mathcal{W} \) at each POD–Greedy iteration. 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. The residual here is defined as
\[ R(v; \mu, t^k) = g^{eq}(t^k) f(v) - \frac{1}{\Delta t^2} \left( m(u_N(\mu, t^{k+1}), v) - 2m(u_N(\mu, t^k), v) + m(u_N(\mu, t^{k-1}), v) \right) \]
\[ - \frac{1}{\Delta t} \left( \frac{1}{2} c(u_N(\mu, t^{k+1}), v; \mu) - \frac{1}{2} c(u_N(\mu, t^{k-1}), v; \mu) \right) \]
\[ - \left( \frac{1}{4} a(u_N(\mu, t^{k+1}), v; \mu) + \frac{1}{2} a(u_N(\mu, t^k), v; \mu) + \frac{1}{4} a(u_N(\mu, t^{k-1}), v; \mu) \right). \]

(20)

The dual norm of the residual is defined as

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

(21)

In algorithm (19), note that we use the dual norm of the residual (in (19g)) as a surrogate for the exact error (define in next Section).

3.3. Errors

3.3.1. Exact error

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

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

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

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

(23)

3.3.2. Error indicator

Consider the POD–Greedy algorithm (19), we can use the exact error (22) as error indicator in step (19g). In that case the computational time, computational effort and required storage would be huge because we would need to solve and store all FEM solutions of all \( \mu \in D \), hence not feasible. Another choice for the error indicator (and also for the error evaluation) would be the rigorous a Posteriori error bound [15, 17] but for the hyperbolic case these bounds [18] are not sharp. (Laplace Transform techniques can improve the situation but also introduce additional complications [32].)

In order to implement the POD–Greedy strategy for our particular problem, we use the dual norm of residual \( \|R(v; \mu, t^k)\|_{Y'} \) as the error indicator. The residual dual norm is actually not rigorous because it does not include stability information (in fact, some temporal terms) present in the full error bound of [18]. However, main advantages of the dual norm of residual are rigorous calculations of the dual norm,
and fast-efficient offline-online decomposition for many \( \mu \) computations required in the Greedy strategy. Furthermore, in the next section we will show that the operation count to find the dual norm of residual (20) for one particular \( \mu \) is very cheap – roughly \( O(N^2(K + Q_a + Q_c + 1)) \).

### 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. We first express \( u_N(\mu, t^k) \) as

\[
  u_N(\mu, t^k) = \sum_{n=1}^{N} u_{Nn}(\mu, t^k) \zeta_n, \quad \forall \zeta_n \in Y_N. \tag{24}
\]

We then choose a test functions \( v = \zeta_n, 1 \leq n \leq N \) for the discrete RB equation (15). It then follows from (15) that

\[
  u_N(\mu, t^k) = \begin{bmatrix} u_{N1}(\mu, t^k) & u_{N2}(\mu, t^k) & \ldots & u_{NN}(\mu, t^k) \end{bmatrix}^T \in \mathbb{R}^N
\]

satisfies

\[
\begin{aligned}
  \left( M_N + \frac{1}{2} \Delta t C_N(\mu) + \frac{1}{4} \Delta t^2 A_N(\mu) \right) u_N(\mu, t^{k+1}) \\
  = \left( -M_N + \frac{1}{2} \Delta t C_N(\mu) - \frac{1}{4} \Delta t^2 A_N(\mu) \right) u_N(\mu, t^{k-1}) \\
  + \left( 2M_N - \frac{1}{2} \Delta t^2 A_N(\mu) \right) u_N(\mu, t^k) + \Delta t^2 g^{eq}(t^k) F_N, \quad 1 \leq k \leq K - 1. \tag{25}
\end{aligned}
\]

The initial condition is treated similar to the treatments in (11) and (16). Here, \( C_N(\mu), A_N(\mu), M_N \in \mathbb{R}^{N \times N} \) are SPD matrices with entries \( C_{Ni,j}(\mu) = c(\zeta_i, \zeta_j; \mu), A_{Ni,j}(\mu) = a(\zeta_i, \zeta_j; \mu), M_{Ni,j} = m(\zeta_i, \zeta_j), 1 \leq i, j \leq N \) and \( F_N \in \mathbb{R}^N \) is the RB load vector with entries \( F_N_i = f(\zeta_i), 1 \leq i \leq N \).

The RB output is then computed from

\[
  s_N(\mu, t^k) = L_N^T u_N(\mu, t^k), \quad 1 \leq k \leq K. \tag{26}
\]

Invoking the affine decomposition in (8), we obtain

\[
A_{Ni,j}(\mu) = a(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_a} \Theta_{q,N}(\mu) a^q(\zeta_i, \zeta_j), \tag{27a}
\]

\[
C_{Ni,j}(\mu) = c(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_c} \Theta_{q,N}(\mu) c^q(\zeta_i, \zeta_j), \tag{27b}
\]

which can be written as
given by

\[ A_{N_{i,j}}(\mu) = \sum_{q=1}^{Q_a} \Theta^q_{\mu}(\mu) A^q_{N_{i,j}}, \quad C_{N_{i,j}}(\mu) = \sum_{q=1}^{Q_c} \Theta^q_{\mu}(\mu) C^q_{N_{i,j}}, \quad (28) \]

where the parameter independent quantities \( A^q_{N_{i,j}} \in \mathbb{R}^{N \times N} \) and \( C^q_{N_{i,j}} \in \mathbb{R}^{N \times N} \) are given by

\[ A^q_{N_{i,j}} = a^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\text{max}}, \quad 1 \leq q \leq Q_a, \quad (29a) \]

\[ C^q_{N_{i,j}} = c^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\text{max}}, \quad 1 \leq q \leq Q_c, \quad (29b) \]

respectively.

The computational procedure is now clear with two stages: the offline and online stages. In the offline stage – performed only once, we solve for the \( \zeta_n, 1 \leq n \leq N_{\text{max}} \); we then compute and store the \( \mu \)-independent quantities in (29), (A6) and (A8) for the output and the dual norm of residual estimate. We shall write \((m - v)\text{prod}\) stands for an \( \mathcal{N} \)-matrix-vector product, and \((v - v)\text{prod}\) stands for an \( \mathcal{N} \)-vector-vector product. Consider algorithm (19), at each POD–Greedy iteration the computational cost is: \( O(K) \) solutions of the underlying \( \mathcal{N} \)-dimensional “truth” FE approximation \((9)\); \( O(K(m - v)\text{prods} + KN(v - v)\text{prods}) \) for the error projection step (19d); \( O(K(m - v)\text{prods} + K^2(v - v)\text{prods}) \) plus one eigenvalues problem for the POD step (19c); and roughly \( O((NQ^2 + NQ)(m - v)\text{prods} + (N^2Q^2 + N^2Q + NQ)(v - v)\text{prods}) \) (with \( Q = Q_a + Q_c + 1 \)) for all necessary offline quantities in (29), (A6) and (A8). Finally, we have totally \( N_{\text{max}}/M \) iterations for the POD–Greedy procedure, thus the offline computational cost is expensive.

In the online stage – performed many times, for each new parameter \( \mu \) – we first assemble the reduced basis matrices in (27), this requires \( O(N^2Q) \) operations. We then solve the RB governing equation (25), the operation count is \( O(N^3 + K^2N^2) \) as the RB matrices are in general full. Finally, we evaluate the displacement output \( s_N(\mu, t^k) \) from (26) at a cost of \( O(KN) \). For the dual norm of residual, the operation count to gather all offline terms and then, calculate the norm as in (A7) is roughly \( O(N^2(K + Q)) \). Thus, 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 savings in the online stage relative to the classical FE approaches.

### 3.5. Numerical results

We now turn to our 3D simplified FEM dental implant-bone model created in section 2.1.2. Our “truth” finite element approximation space is of dimension \( \mathcal{N} = 26802 \). For time integration, \( T = 1 \times 10^{-3} \text{s}, \Delta t = 2 \times 10^{-6} \text{s}, K = \frac{T}{2\Delta t} = 500 \). The input parameter \( \mu \) is defined by \( E \) and \( \beta \): \( \mu = (E, \beta) \in \mathcal{D} \), 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 \( \|w\|_Y^2 = a(w, w; \tilde{\mu}) + m(w, w; \tilde{\mu}) \), where \( \tilde{\mu} = (8 \times 10^6 \text{Pa}, 2.75 \times 10^{-5}) \) is the arithmetic average of \( \mu \) in \( \mathcal{D} \); \( Q_a = 2, Q_c = 2 \). To verify our computation code (performed in Matlab R2007a), we first compare the FEM outputs computed by ABAQUS and by our code with the test parameter \( \mu_{\text{test}} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5}) \). Fig.3 shows the output displacement responses in \( x-, y- \) and \( z- \)direction versus time at \( \mu_{\text{test}} \) via ABAQUS and our code. Fig.3a and Fig.3b demonstrate that our
FEM results 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 Fig.3c. In our dental implant problem, since the applied load is opposite to the $x-$direction, the $x-$component of the output displacement response is most important among the three components (i.e. $x-$, $y-$ and $z-$component). Hence, for the remaining discussion the “output displacement response” refers only to the $x-$component of the displacement of the output.

The POD–Greedy algorithm is then implemented to create the RB spaces $Y_N = \{\zeta_n, 1 \leq n \leq N\}, 1 \leq N \leq N_{\text{max}}$. The algorithm is actually the POD in time and Greedy in parameter space. We choose $M = 5$ (in step (19e)) for each Greedy iteration and $N_{\text{max}} = 60$ to terminate the iteration procedure. A sample set $\Xi_{\text{train}}$ is created randomly with uniform distribution over $D$ with $n_{\text{train}} = 100$ samples. Sample points distribution of $S^*$ is illustrated in Fig.4a. We show, as a function of $N$: $c_u^{\text{max,rel}}$ is the maximum over $\Xi_{\text{train}}$ of $c_u(\mu, t^K)$ and $c_s^{\text{max,rel}}$ is the maximum over $\Xi_{\text{train}}$ of $c_s(\mu, t^K)$ in Fig.4b. The comparison of $s(\mu_{\text{test}}, t)$ versus $s_N(\mu_{\text{test}}, t)$ with various $N$ are presented in Fig.5. The numerical results demonstrate that our exact errors are acceptably small, and the convergence rate is fast for a $N \ll N_{\text{max}}$.

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. Computation time for the RB forward solver ($t_{\text{RB(online)}}$), CPU-time for the FEM forward solver by our code ($t_{\text{FEM}}$) and by ABAQUS ($t_{\text{ABAQUS}}$), and the CPU-time saving factor $\kappa = t_{\text{FEM}}/t_{\text{RB(online)}}$ are listed on Table 2, respectively. We observe that while the original forward solver FEM (i.e. our code and ABAQUS) take thousands of seconds to compute the displacement outputs, the RB model (with various choices of $N$) takes less than 1 second to find that with known accuracy. Thus, it is clear that the RB is very efficient and reliable for solving forward problems. Next, 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 $\beta$ of the interfacial tissue in our dental implant-bone structure.

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

The inverse problem considered is concerned with the simultaneous estimation of the two parameters: Young’s modulus $E$ and stiffness Rayleigh damping coefficient $\beta$ of the interfacial tissue from the “measured” displacement response at output point (Fig.2a). This inverse problem can be regarded as an optimization problem which aims at finding the unknown parameters $\mu = (E, \beta)$ that minimizes the following sum of the squares function:

$$S(\mu) = \sum_{i=1}^{K} [s_{N,i}(\mu) - s_{i}^{\text{measure}}]^2 = r^T r, \quad (30)$$

\footnote{Here, vector $\mu$ should be typed in bold as $\mu$, we use the mediumface italic for suitability with previous sections.}
where

$$r_i(\mu) = s_{N,i}(\mu) - s_i^{\text{measure}}.$$  \hfill (31)

Here, $K$ is the total number of discrete time steps. $s_{N,i}(\mu)$ is the “computed” RB output displacement defined in (14) at time $t_i$ with parameter $\mu$. $s_i^{\text{measure}}$ is the “measured” simulated displacement at time $t_i$, and $\mu = (E, \beta)$ is the unknown material properties that we aim to find.

The parameters $\mu$ which minimize the function $S$ defined by equation (30) must satisfy the following set of nonlinear algebraic equations:

$$\sum_{i=1}^{K} 2 \frac{\partial s_{N,i}}{\partial \mu_j} (s_{N,i} - s_i^{\text{measure}}) = 2 \frac{\partial r^T}{\partial \mu_j} r = 2 J^T r = 0, \quad j = 1, 2.$$  \hfill (32)

The set of equations (32) are obtained by differentiating equation (30) with respect to each component of the parameter vector $\mu$ and then setting these derivatives equal to zero. The matrix $J$ is called Jacobian matrix with entries are defined as: $J_{ij} = \frac{\partial r}{\partial \mu_j}$. In order to solve the system of algebraic equations (32), the Levenberg–Marquardt–Fletcher iterative method [33, 34] is used. The update equation of parameter $\mu$ at iteration $l + 1$ has the form:

$$\mu^{(l+1)} = \mu^{(l)} + \Delta \mu^{(l)},$$  \hfill (33a)

$$\Delta \mu^{(l)} = -(J^{(l)T}J^{(l)} + \lambda^{(l)}D)^{-1}J^{(l)T}r^{(l)}.$$  \hfill (33b)

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

$$|\mu_j^{(l+1)} - \mu_j^{(l)}| < \varepsilon, \quad j = 1, 2,$$  \hfill (34)

where $\varepsilon$ is a small number. The entries of the Jacobian matrix $J$ can be calculated from the following finite difference formula

$$\frac{\partial r_i(\mu)}{\partial \mu_j} \approx \frac{r_i(\mu + \varepsilon U_j) - r_i(\mu)}{\varepsilon},$$  \hfill (35)

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

In the remain sections, the open-source code [34] with appropriate modifications is used to implement our RB–LMF algorithm.

4.2. **Numerical results**

4.2.1. **Effects of $E$ and $\beta$ to output displacement $s_N$**

We use $N = 40$ basis functions in all the computations that relate to our RB model in this inverse analysis. Effects of Young’s modulus $E$ and stiffness Rayleigh
damping coefficient $\beta$ to the displacement response $s_N$ are plotted in Fig.6a and Fig.6b, respectively. As observed, Young’s modulus $E$ dominates the width between the peaks of the displacement response curves while coefficient $\beta$ controls the height of these peaks. It is shown that the output displacement responses are very sensitive to these two parameters.

4.2.2. Synthetic data

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

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

where $\omega$ lies in the range $-2.576 \leq \omega \leq 2.576$ if a 99% confidence interval is assumed for the data. $\omega$ takes a random value from a normal distribution [24] with the standard deviation $\sigma$ computed as in [35]

$$\sigma = p_e \left( \frac{1}{n_s - 1} \sum_{j=1}^{n_s} (s^j_N)^2 \right)^{1/2}. \quad (37)$$

Here, $p_e$ is the noise level (for example, $p_e = 0.05$ means a 5% noise level), $n_s$ is the total number of sampling points $\mu^j$ and $s^j_N$ is the usual RB displacement response at sampling point $\mu^j$.

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 $\mu \in D$: $\mu^{(0)} = (1 \times 10^6 \text{Pa}, 5 \times 10^{-6})$ is chosen to be the initial guess - which is also independent of $\mu^\text{measure}$. For the case $p_e = 5\%$ noise level, the simulated “measured” displacement versus time is plotted in Fig.7a. The final “computed” displacement versus simulated “measured” displacement are presented in Fig.7b. In addition, we show the iteration history of each computed parameter component and that of the sum of the squares errors in Fig.8. For the cases of various noise levels, we list the computed parameters, relative errors and the corresponding number of iterations in Table 3, respectively.

In order to test with many parameters, a sample set $S^\text{true}$ of regular $(5 \times 5)$ grid pattern over $D$ is created. We then implement the RB–LMF procedure with fixed initial guess $\mu^{(0)} = (1 \times 10^6 \text{Pa}, 5 \times 10^{-6})$. We show the plots of the set $S^\text{true}$ ($\geq \mu^\text{true}$) together with the final computed set $S^\text{compute}$ ($\geq \mu^\text{compute}$) with added noise levels $p_e = 5\%$ and $p_e = 10\%$ in Fig.9, respectively. Comparisons with the exact values show that at lower noise of displacement response, reliable estimates can be provided by this procedure.

To validate the efficiency of the RB–LMF procedure, the total forward solver calls for a RB–LMF inverse analysis are given in Table 4; the total CPU time is recorded and provided in Table 5. It is found that CPU time for a LMF model using RB solver$^1$ is significantly faster than that of using the FEM solver. Therefore, the proposed RB–LMF approach strongly reduces the computational time and cost.

$^1$The work is focused in real-time context with many online computations, the offline stage is done once and expensive; hence that computation time would not be mentioned here.
Table 1. Material properties of dental implant-bone structure.

<table>
<thead>
<tr>
<th>Layers</th>
<th>E (Pa)</th>
<th>$\nu$</th>
<th>$\rho$ (g/mm$^3$)</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortical bone</td>
<td>$2.3162 \times 10^{10}$</td>
<td>0.371</td>
<td>$1.8601 \times 10^{-3}$</td>
<td>$3.38 \times 10^{-6}$</td>
</tr>
<tr>
<td>Cancellous bone</td>
<td>$8.2345 \times 10^8$</td>
<td>0.3136</td>
<td>$7.1195 \times 10^{-4}$</td>
<td>$6.76 \times 10^{-6}$</td>
</tr>
<tr>
<td>Tissue</td>
<td>$E = 3.155 \times 10^{10}$</td>
<td>0.32</td>
<td>$4.52 \times 10^{-3}$</td>
<td>$5.1791 \times 10^{-10}$</td>
</tr>
<tr>
<td>Titan implant</td>
<td>$1.05 \times 10^{11}$</td>
<td>0.305</td>
<td>$8.927 \times 10^{-3}$</td>
<td>$2.5685 \times 10^{-8}$</td>
</tr>
<tr>
<td>Stainless steel</td>
<td>$1.93 \times 10^{11}$</td>
<td>0.305</td>
<td>$8.927 \times 10^{-3}$</td>
<td>$2.5685 \times 10^{-8}$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>N</th>
<th>$t_{RB(online)}$ (sec)</th>
<th>$t_{FEM}$ (sec)</th>
<th>$t_{ABAQUS}$ (sec)</th>
<th>$\kappa = t_{FEM}/t_{RB(online)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.1947</td>
<td>3750</td>
<td>2010</td>
<td>$1.9260 \times 10^{4}$</td>
</tr>
<tr>
<td>20</td>
<td>0.2312</td>
<td>3750</td>
<td>2010</td>
<td>$1.6220 \times 10^{4}$</td>
</tr>
<tr>
<td>30</td>
<td>0.2969</td>
<td>3750</td>
<td>2010</td>
<td>$1.2631 \times 10^{4}$</td>
</tr>
<tr>
<td>40</td>
<td>0.3405</td>
<td>3750</td>
<td>2010</td>
<td>$1.1013 \times 10^{4}$</td>
</tr>
</tbody>
</table>

Table 3. Computed results with case $\mu^{measure} = (8 \times 10^6 \text{Pa}, 8 \times 10^{-6})$ for various noise level.

| $p_e$ | $E^{compute}$ (Pa) | $\beta^{compute}$ | $|E^{comp} - E^{mea}|/E^{mea}$ (%) | $|\beta^{comp} - \beta^{mea}|/\beta^{mea}$ (%) | No. of iterations |
|-------|-------------------|-------------------|---------------------------------|---------------------------------|-----------------|
| 0     | $8.0000E+06$      | 0.0000            | 0.0000                          | 0.0000                          | 28              |
| 0.01  | $8.0007E+06$      | 7.9969            | 0.0085                          | 0.0390                          | 35              |
| 0.05  | $8.0232E+06$      | 7.9290            | 0.2930                          | 0.8879                          | 33              |
| 0.1   | $7.9819E+06$      | 7.8484            | 0.2791                          | 2.3714                          | 41              |
| 0.15  | $8.0233E+06$      | 7.8103            | 0.2791                          | 2.3714                          | 41              |
| 0.2   | $8.0778E+06$      | 8.0333            | 0.9723                          | 0.4166                          | 34              |

Table 4. Total number of forward analyses required in a RB–LMF inverse analysis (for one particular $\mu^{measure}$).

<table>
<thead>
<tr>
<th>Average number of iterations</th>
<th>Number of RB calls in each iteration</th>
<th>Total RB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>$m = 105$</td>
</tr>
</tbody>
</table>

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, noise contaminated, one parameter, many 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 RB–LMF approach is able to model non-linear relation between structural parameter and non-static response of complex dental implant structures.

Acknowledgements

This work was supported by the Singapore–MIT Alliance.

Table 5. Comparison of computational time for a LMF model using FEM and RB as forward solvers (for one particular $\mu^{measure}$).

<table>
<thead>
<tr>
<th>Total RB calls</th>
<th>CPU time for each solver</th>
<th>Total computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 105$</td>
<td>$t_{FEM}$ (sec)</td>
<td>$m \times t_{FEM}$</td>
</tr>
<tr>
<td>$t_{RB(online)}$</td>
<td>0.3405 (sec)</td>
<td>$m \times t_{RB(online)}$</td>
</tr>
<tr>
<td></td>
<td>3750 (sec)</td>
<td>109.375 (hrs)</td>
</tr>
<tr>
<td></td>
<td>3750 (sec)</td>
<td>35.7525 (sec)</td>
</tr>
</tbody>
</table>
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).

Figure 3. Comparison of FEM output displacement responses by using our code versus by using ABAQUS software with respect time in $x-$ (a), $y-$ (b), and $z-$direction (c) with $\mu_{text} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$.
Figure 4. Distribution of sampling points of the POD–Greedy procedure (a) and the maximum relative error of solution and output as a function of $N$ (b).

Figure 5. Comparison of output displacement responses by FEM and RB with $\mu_{test_1} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$ with $N = 3$ (a), $N = 5$ (b) and $N = 10$ (c) basis functions.

Figure 6. Effects of Young’s modulus $E$ (with case $\beta = 1 \times 10^{-5}$) (a) and effects of stiffness Rayleigh damping coefficient $\beta$ (with case $E = 10 \times 10^6 \text{Pa}$) (b) on displacement responses.
Figure 7. Simulated “measured” displacements (white Gaussian noise added) with respect time (a) and comparison between “measured” with “computed” displacements with respect time (b).

Figure 8. Iteration history of Young’s modulus $E$ (a), of stiffness Rayleigh coefficient $\beta$ (b) and of the sum of square errors $S(\mu)$ (c).

Figure 9. Comparison of sample set $S_{\text{true}}$ and $S_{\text{compute}}$ with noise added $p_e = 5\%$ (a) and $p_e = 10\%$ (b).
References


30 REFERENCES

Appendix A. Calculation of the dual norm of the residual

In this section, we discuss the calculation of the dual norm of the residual. We consider the residual as defined in (20), its dual norm is given by (21). The dual norm of residual can be defined alternatively as

\[ \| R(v; \mu, t^k) \|_{Y'} \equiv \sup_{v \in Y} \frac{R(v; \mu, t^k)}{\| v \|_Y}, \]

(A1)

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

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

(A2)

From (20), (24) and the affine property (8) it thus follows that \( \hat{e}(\mu, t^k) \) satisfies

\[
(\hat{e}(\mu, t^k), v)_Y = g^{eq}(t^k) f(v) + \sum_{n=1}^{N} \frac{-u_{N,n}(\mu, t^{k+1}) + 2u_{N,n}(\mu, t^{k}) - u_{N,n}(\mu, t^{k-1})}{\Delta t^2} m(\zeta_n, v) \\
+ \sum_{n=1}^{N} \frac{Q_c}{\sum_{q=1}^{\Theta_c(\mu)}} \frac{-u_{N,n}(\mu, t^{k+1}) + u_{N,n}(\mu, t^{k-1})}{2\Delta t} \Theta_c(\mu) \\
+ \sum_{n=1}^{N} \frac{Q_a}{\sum_{q=1}^{\Theta_a(\mu)}} \frac{-u_{N,n}(\mu, t^{k+1}) - 2u_{N,n}(\mu, t^{k}) - u_{N,n}(\mu, t^{k-1})}{4} \Theta_a(\mu). \]

(A3)

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

\[
\hat{e}(\mu, t^k) = g^{eq}(t^k) f + \sum_{n=1}^{N} \lambda_m,n(\mu, t^k) M_n \\
+ \sum_{n=1}^{N} \frac{Q_c}{\sum_{q=1}^{\Theta_c(\mu)}} \lambda_c,n(\mu, t^k) C_{q,n} \\
+ \sum_{n=1}^{N} \frac{Q_a}{\sum_{q=1}^{\Theta_a(\mu)}} \lambda_a,n(\mu, t^k) A_{q,n}. \]

(A4)
Where

\[
\begin{align*}
\lambda_{m,n}(\mu, t^k) &= \frac{-u_N n(\mu, t^{k+1}) + 2u_N n(\mu, t^k) - u_N n(\mu, t^{k-1})}{\Delta t^2}, \\
\lambda_{c,n}(\mu, t^k) &= \frac{-u_N n(\mu, t^{k+1}) + u_N n(\mu, t^{k-1})}{2\Delta t}, \\
\lambda_{a,n}(\mu, t^k) &= \frac{-u_N n(\mu, t^{k+1}) - 2u_N n(\mu, t^k) - u_N n(\mu, t^{k-1})}{4};
\end{align*}
\]  

(A5)

and we calculate

\[
\mathcal{F} \in Y \quad \text{from} \quad (\mathcal{F}, v)_Y = f(v), \quad \forall v \in Y,
\]

\[
\mathcal{M}_n \in Y \quad \text{from} \quad (\mathcal{M}_n, v)_Y = m(\zeta_n, v), \quad \forall v \in Y \text{ for } 1 \leq n \leq N,
\]

\[
\mathcal{C}_{q,n} \in Y \quad \text{from} \quad (\mathcal{C}_{q,n}, v)_Y = c^q(\zeta_n, v), \quad \forall v \in Y \text{ for } 1 \leq q \leq Q_c, 1 \leq n \leq N,
\]

\[
\mathcal{A}_{q,n} \in Y \quad \text{from} \quad (\mathcal{A}_{q,n}, v)_Y = a^q(\zeta_n, v), \quad \forall v \in Y \text{ for } 1 \leq q \leq Q_a, 1 \leq n \leq N,
\]

note \(\mathcal{F}, \mathcal{M}, \mathcal{C}, \mathcal{A}\) are parameter independent.

From (A1) and (A4) it follows that

\[
\begin{align*}
\|\hat{\epsilon}(\mu, t^k)\|_Y^2 &= (\hat{\epsilon}(\mu, t^k), \hat{\epsilon}(\mu, t^k))_Y \\
&= g^eq(t^k)g^q(t^k)\Delta_t f + \\
&\quad + 2 \sum_{n=1}^{N} g^eq(t^k) \left\{ \lambda_{m,n}(\mu, t^k)\Lambda_{n}^{fm} + \sum_{q=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\lambda_{c,n}(\mu, t^k)\Lambda_{qn}^{fc} \right. \\
&\quad \left. + \sum_{q=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\lambda_{a,n}(\mu, t^k)\Lambda_{qn}^{fa} \right\} \\
&\quad + \sum_{n,n'=1}^{N} \left\{ \lambda_{m,n}(\mu, t^k)\lambda_{m,n'}(\mu, t^k)\Lambda_{nn'}^{mm} + 2 \sum_{q=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\lambda_{c,n}(\mu, t^k)\lambda_{m,n'}(\mu, t^k)\Lambda_{nn'}^{mc} \right. \\
&\quad \left. + 2 \sum_{q=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\lambda_{a,n}(\mu, t^k)\lambda_{m,n'}(\mu, t^k)\Lambda_{nn'}^{ma} \right\} \\
&\quad + \sum_{q,q'=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\Theta_{q',\zeta}^{q'}(\mu)\lambda_{c,n}(\mu, t^k)\lambda_{c,n'}(\mu, t^k)\Lambda_{nn'}^{cc} \\
&\quad + \sum_{q,q'=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\lambda_{a,n}(\mu, t^k)\Theta_{q',\zeta}^{q'}(\mu)\lambda_{a,n'}(\mu, t^k)\Lambda_{nn'}^{ca} \\
&\quad + \sum_{q,q'=1}^{Q_c} \Theta_{q,\zeta}^q(\mu)\lambda_{a,n}(\mu, t^k)\Theta_{q',\zeta}^{q'}(\mu)\lambda_{a,n'}(\mu, t^k)\Lambda_{nn'}^{aa} \right\}.
\end{align*}
\]  

(A7)

where the parameter-independent quantities \(\Lambda\) are defined as

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