SANDIA REPORT

SAND2013-9172
Unlimited Release
Printed October, 2013

Parameterized Reduced-Order Models Using Hyper-Dual Numbers

Jeffrey A. Fike and Matthew R. Brake

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-9172
Unlimited Release
Printed October, 2013

Parameterized Reduced-Order Models Using Hyper-Dual Numbers

Jeffrey A. Fike and Matthew R. Brake
Component Science and Mechanics
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1070

Abstract

The goal of most computational simulations is to accurately predict the behavior of a real, physical system. Accurate predictions often require very computationally expensive analyses and so reduced order models (ROMs) are commonly used. ROMs aim to reduce the computational cost of the simulations while still providing accurate results by including all of the salient physics of the real system in the ROM. However, real, physical systems often deviate from the idealized models used in simulations due to variations in manufacturing or other factors. One approach to this issue is to create a parameterized model in order to characterize the effect of perturbations from the nominal model on the behavior of the system. This report presents a methodology for developing parameterized ROMs, which is based on Craig-Bampton component mode synthesis and the use of hyper-dual numbers to calculate the derivatives necessary for the parameterization.

Acknowledgment

This work was funded in part by the U.S. Department of Energy's Predictive Science Academic Alliance Program (PSAAP) Center at Stanford University. We would like to acknowledge the contributions of Sean Topping, who created the initial beam model in MATLAB and performed similar studies using finite-differences rather than the hyper-dual numbers used here. We would also like to thank Rich Field, Ryan Schultz, and Remi Dingreville for feedback on drafts of this report. Finally, we would like to thank Eliot Fang for his support of this research.

Contents

List of Figures

Acronyms

C-B Craig-Bampton

CMS Component Mode Synthesis

DOF Degrees of Freedom

FE Finite Element

PROM Parameterized Reduced Order Model

ROM Reduced Order Model

Chapter 1

Introduction

The goal of most computational simulations is to accurately predict the behavior of a real, physical system. High-fidelity computational simulations can often provide very accurate predictions but they have a very high computational cost. Reduced order models (ROMs) are often used as surrogates for a full-order model in order to decrease the computational expense of analysis. A ROM should therefore be much less expensive to evaluate while retaining a sufficient level of accuracy.

A real, physical system often deviates from the idealized model used for the simulations. These perturbations from the nominal model are often due to variations in manufacturing of the physical system. To model these perturbations without a systematic, efficient reduced order approach would be prohibitively difficult. For example, consider a scenario where it takes several weeks to develop a high quality mesh for one relatively simple component. To quantify the aleatoric uncertainty associated with manufacturing, thousands of perturbations of the ideal geometry are necessary, and each likely requires a new mesh. Even with factoring in time saved from some automation of the process, the number of man hours required to construct these meshes is on the order of 20 years. In addition, the computational time to analyze all of these models is on the order of several years assuming that an entire super computer can be dedicated to the analysis. Clearly, decades of time are infeasible constraints to be incorporated into a design cycle. One method of accounting for these perturbations is to create a parameterized reduced order model (PROM) of the system [1–4]. This allows the behavior of the system to be inexpensively predicted over a range of perturbations based on a few simulations distributed about the nominal design.

A standard approach to constructing ROMs for structural dynamics is Craig-Bampton (C-B) Component Mode Synthesis (CMS) [5]. Figure 1.1 shows two approaches that can be taken to analyze a system composed of several components. The left branch of the diagram shows the traditional approach of forming a full-order finite element (FE) model for the system, which is often prohibitively expensive for assessing aleatoric uncertainty. The right branch shows the steps when C-B CMS is used. To account for aleatoric uncertainty, PROMs can be utilized. Simple PROMs can be constructed from a finite Taylor series expansion; for instance, in computing some scalar quantity of interest, $f(x)$, as a function of some perturbation to the nominal design, Δx, $f(x)$ can be approximated as $\bar{f}(x)$, which is based on a Taylor series expansion. For the FE and CMS cases outlined in Fig. 1.1, $f(x)$ could be elements of the mass and stiffness matrices or it could be the result of the system analysis, such as displacements or eigenvalues. The perturbations to the nominal design, Δx, could be changes in material properties or geometric variations. A parameterized model can then be created for a quadratic expansion as

$$\bar{f}(x + \Delta x) = f(x) + (\Delta x) f'(x) + \frac{(\Delta x)^2}{2} f''(x). \tag{1.1}$$

Higher order models, such as a cubic model

$$\bar{f}(x + \Delta x) = f(x) + (\Delta x) f'(x) + \frac{(\Delta x)^2}{2} f''(x) + \frac{(\Delta x)^3}{6} f'''(x), \tag{1.2}$$

can be created by keeping higher-order terms in the expansion.

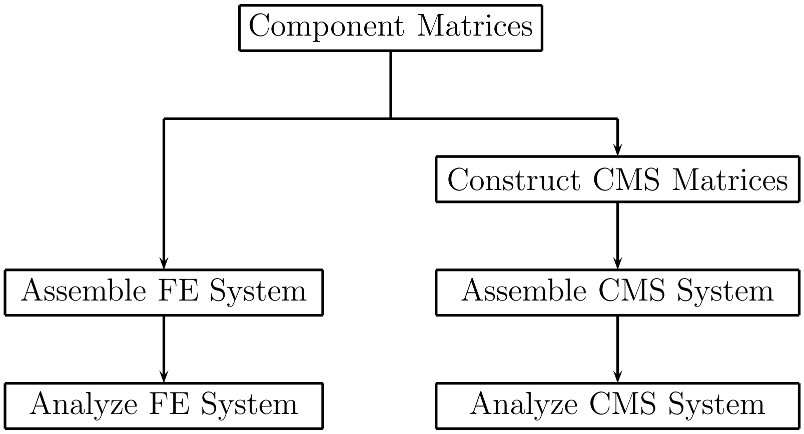


Figure 1.1. Two possible approaches for analyzing a system composed of several components.

The use of these parameterized models with FE analysis requires the calculation of derivatives at the nominal design. These derivatives can be computed in many ways, such as using finite-difference approximations. The approach proposed here is to compute the derivatives using hyper-dual numbers [6]. Hyper-dual numbers produce exact values of the derivatives, and only require evaluation at the nominal design. This is in contrast to finite-difference approximations which require evaluations at several perturbed designs and hence can require the creation of several meshes. The use of hyper-dual numbers only requires the nominal mesh, but the information on how the mesh would change due to the perturbations needs to be known or calculated. In practice, this can require the introduction of hyper-dual numbers into the meshing procedure, which is not likely to be practical for large commercial codes; however, this may be feasible for a research code such as SIERRA [7].

Parameterized models can be created at several levels in the analysis procedure outlined in Figure 1.1. The various possibilities are shown in Figure 1.2. One possibility is to construct parameterized models for the component mass and stiffness matrices. Another possibility is to parameterize the output of the system analysis, i.e. displacements or eigenvalues. When using CMS it is also possible to parameterize the reduced-order mass and stiffness matrices. One goal of this report is to assess the efficacy of developing PROMs at each level identified in Fig. 1.2.

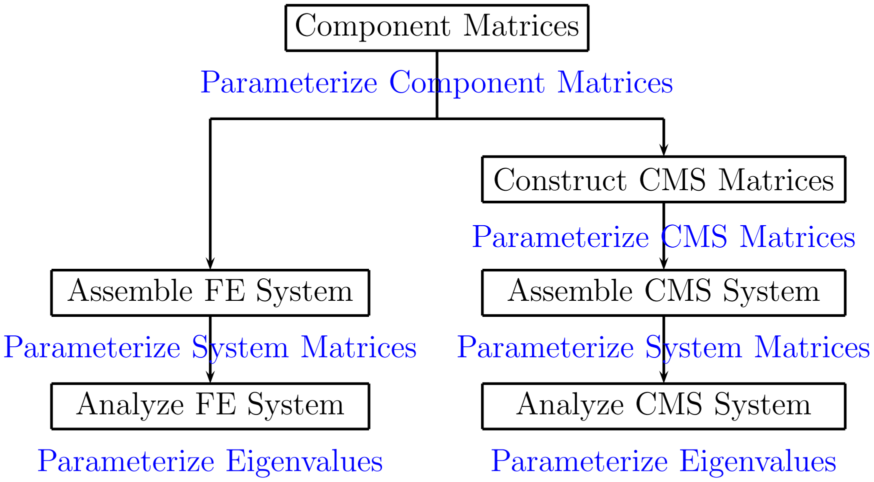


Figure 1.2. The levels at which parameterized models can be constructed.

This report discusses the implementation of C-B CMS in MATLAB in Chapter 2. Chapter 3 introduces hyper-dual numbers and discusses the development of PROMs, where the required derivatives are computed using hyper-dual numbers [6]. Chapter 4 assesses the efficacy of each potential level of developing a PROM, as indicated in Fig. 1.2, and conclusions are presented in 5.

Chapter 2

Craig-Bampton Component Mode Synthesis

Theory

The equation of motion for an unforced and undamped structure with mass matrix $[M]$, stiffness matrix $[K]$, and displacement $\{u\}$ is

$$[M]\{\ddot{u}\} + [K]\{u\} = \{0\}. \tag{2.1}$$

The C-B CMS [5] method is based on a substructuring of the component's degrees of freedom (DOFs) into boundary (u_b) DOFs and internal (u_i) DOFs as $\{u\} = \{u_b, u_i\}^T$. Boundary DOFs typically are defined as DOFs where excitations are applied or where output quantities are desired (such as the displacement of a particular flange), whereas internal DOFs are all non-boundary DOFs, which generally are inside of the component and thus have no applied loads. This substructuring is used to recast 2.1 as

$$\begin{bmatrix} M_{bb} & M_{bi} \\ M_{ib} & M_{ii} \end{bmatrix} \begin{Bmatrix} \ddot{u}_b \\ \ddot{u}_i \end{Bmatrix} + \begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{Bmatrix} u_b \\ u_i \end{Bmatrix} = \{0\}. \tag{2.2}$$

Modal analysis is performed on the internal degrees of freedom,

$$(K_{ii} - \lambda_\ell M_{ii})\,\phi_\ell = 0, \tag{2.3}$$

where λ_ℓ are individual eigenvalues and ϕ_ℓ are the corresponding eigenvectors of the fixed-interface normal modes. Constraint modes, ϕ_C, are a static deflection shape and are computed by imposing a unit displacement at each individual boundary DOF while holding the other boundary DOFs fixed at zero displacement. The constraint modes can therefore be computed as

$$\phi_C = -K_{ii}^{-1} K_{ib}. \tag{2.4}$$

Subsequently, the physical DOFs (u_b and u_i) are related to the hybrid set of physical and modal DOFs (u_b and q_m) by

$$\begin{Bmatrix} u_b \\ u_i \end{Bmatrix} = \begin{bmatrix} I & 0 \\ \phi_C & \phi_N \end{bmatrix} \begin{Bmatrix} u_b \\ q_m \end{Bmatrix}, \tag{2.5}$$

where I is the identity matrix and ϕ_N is a matrix of the eigenvectors ϕ_ℓ. This allows the equation of motion for each component to be written as

$$\begin{bmatrix} M_{CC} & M_{CN} \\ M_{NC} & I \end{bmatrix} \left\{ \begin{array}{c} \ddot{u}_b \\ \ddot{q}_m \end{array} \right\} + \begin{bmatrix} K_{CC} & 0 \\ 0 & \Lambda^2 \end{bmatrix} \left\{ \begin{array}{c} u_b \\ q_m \end{array} \right\} = \{0\}\,, \tag{2.6}$$

assuming the eigenvectors are normalized with respect to the mass matrix M_{ii}, and with

$$M_{CC} = M_{bb} + M_{bi}\phi_C + \phi_C^T + \phi_C^T M_{ii}\phi_C, \tag{2.7}$$

$$M_{CN} = M_{NC}^T = \left(M_{bi} + \phi_C^T M_{ii}\right)\phi_N, \tag{2.8}$$

$$K_{CC} = K_{bb} + M_{bi}\phi_C, \tag{2.9}$$

where Λ is a diagonal matrix of the eigenvalues λ_ℓ.

The number of degrees of freedom can be reduced by retaining only enough fixed-interface normal modes to capture the behavior of interest. In general, keeping more modes results in a more accurate approximation of the true behavior at the expense of an increased computational cost. The reduced mass and stiffness matrices for each component can be combined to form a ROM for the system. This ROM is less expensive to analyze and provides accurate results given that enough modes are kept to capture the behavior of interest.

Description of MATLAB Files

This section describes the MATLAB analysis routines created to analyze the stepped beam system described in Chapter 4.

AnalyzePerturbedBeam_FEM.m

This function is called to analyze the simple beam described in Figure 4.1 using the FE method for a full DOF model. This function takes as inputs the nominal design (component properties and geometry) as well as perturbations about the nominal design. This function analyzes the beam using the following procedure

1. ComputeComponentProperties_FEM.m determines the number of DOFs per component.
2. ComputeComponentMatrices.m computes the mass and stiffness matrices for the three components.
3. AssembleThreeComponents.m creates the mass and stiffness matrix for the assembled beam.
4. ComputeNormalModes.m computes the eigenvalues and eigenvectors.

AnalyzePerturbedBeam_CMS.m

This function is similar to AnalyzePerturbedBeam_FEM.m except that C-B CMS is performed and the resulting ROM for the system is analyzed. This function analyzes the beam using the following procedure

1. ComputeComponentProperties_FEM.m (same as above).
2. ComputeComponentProperties_CMS.m determines various properties for each component for use in the C-B CMS procedure.
3. ComputeComponentMatrices.m (same as above).
4. ComputeCraigBamptonMatrices.m computes the reduced mass and stiffness matrices, as shown in 2.6, and the transformation matrices that relate the modal displacements to the internal DOFs for each component.
5. AssembleCraigBamptonSystem.m combines the component matrices into a ROM for the system.
6. AnalyzeAssembledSystem.m computes the eigenvalues and eigenvectors for the ROM.

ComputeComponentProperties_FEM.m
This function takes in the geometry of the components and returns the number of DOFs for each component.

ComputeComponentProperties_CMS.m
This function computes the number of boundary conditions per component and type of boundary condition (global or attachment).

ComputeComponentMatrices.m
This function takes in the nominal design and perturbation and computes the mass and stiffness matrices for each component.

ComputeCraigBamptonMatrices.m
This function takes in the component mass and stiffness matrices and returns the reduced-order mass and stiffness matrices as shown in 2.6 and the transformation matrices given in 2.5.

AssembleThreeComponents.m
This function takes the full-order component mass and stiffness and assembles them into system matrices.

AssembleTwoComponents.m
This function takes in two sets of reduced mass and stiffness matrices, as well as the transformation matrices, and creates matrices for the joined components.

AssembleCraigBamptonSystem.m
This function uses AssembleTwoComponents.m to create reduced mass and stiffness matrices for the system.

ComputeNormalModes.m
This function computes eigenvalues and eigenvectors for the full-order system.

AnalyzeAssembledSystem.m
This function computes eigenvalues and eigenvectors for the ROM.

ApplyBounaryConditions.m

This function eliminates the global boundary DOFs from the full-order system before computing the eigenvalues and eigenvectors.

eig_Nsort.m

This function calls MATLAB's built in function *eig.m*. The purpose of this function is to sort the eigenvalues in increasing order and normalize the eigenvectors with respect to the mass matrix. These operations are sometimes carried out by *eig.m*, but are not guaranteed. This function returns the smallest N eigenvalues as a column vector and their corresponding eigenvectors. This function was created so that the hyper-dual number versions of the above routines avoid the high computational cost of computing derivatives of all the eigenvectors when only a few are actually kept.

Chapter 3

Parameterization Using Hyper-Dual Numbers

The PROM method proposed in the present work is derived using hyper dual numbers because hyper dual numbers allow for exact calculations of derivatives without needing multiple points at which the derivatives are evaluated. This is achievable due to the definition of a dual number. A dual number is defined as a class of generalized complex numbers where the non-real part is defined by the non-zero root of the number zero, as described in [6]. A hyper-dual number is a dual number defined in more than one dimension. For example, two dimensional hyper-dual numbers are defined to consist of one real part and three non-real parts, where the three non-real units ϵ_1, ϵ_2, and $\epsilon_1\epsilon_2$ have the properties that $\epsilon_1^2 = \epsilon_2^2 = (\epsilon_1\epsilon_2)^2 = 0$ but $\epsilon_1 \neq \epsilon_2 \neq \epsilon_1\epsilon_2 \neq 0$. Higher dimensional hyper-dual numbers can also be considered, such as described later in the text. The Taylor series for a real-valued function subjected to a hyper-dual perturbation of h_1 and h_2 truncates exactly at the second-derivative term

$$f(x + h_1\epsilon_1 + h_2\epsilon_2 + 0\epsilon_1\epsilon_2) = f(x) + h_1 f'(x)\epsilon_1 + h_2 f'(x)\epsilon_2 + h_1 h_2 f''(x)\epsilon_1\epsilon_2. \tag{3.1}$$

There is no truncation error because all the higher-order terms contain ϵ_1^2 or ϵ_2^2 or higher powers and are zero by definition. The first and second derivatives are the leading terms of the non-real parts, and these terms can be found by taking the individual non-real parts and dividing by the step size. There is no required difference operation, as in finite-difference approximations, which would lead to subtractive cancelation error. The first and second derivatives can thus be computed exactly, regardless of the step size.

Hyper-dual numbers have been implemented in C++, CUDA, and MATLAB for second-derivative calculations. For this parameterization study, higher-order derivatives are desired since many geometric terms enter into the stiffness matrix in a cubic manner. A hyper-dual implementation that produces exact third derivatives is created by including ϵ_3 terms with additional perturbation h_3. This yields a Taylor series that truncates exactly at the third derivative term

$$\begin{aligned} f(x &+ h_1\epsilon_1 + h_2\epsilon_2 + h_3\epsilon_3 + 0\epsilon_1\epsilon_2 + 0\epsilon_1\epsilon_3 + 0\epsilon_2\epsilon_3 + 0\epsilon_1\epsilon_2\epsilon_3) \\ &= f(x) + h_1 f'(x)\epsilon_1 + h_2 f'(x)\epsilon_2 + h_3 f'(x)\epsilon_3 + h_1 h_2 f''(x)\epsilon_1\epsilon_2 \\ &\quad + h_1 h_3 f''(x)\epsilon_1\epsilon_3 + h_2 h_3 f''(x)\epsilon_2\epsilon_3 + h_1 h_2 h_3 f'''(x)\epsilon_1\epsilon_2\epsilon_3. \end{aligned} \tag{3.2}$$

The use of hyper-dual numbers requires overloading all of the functions in the analysis code to operate on hyper-dual numbers instead of on real numbers (that is, creating a new method for a function that operates on hyper dual numbers instead of the originally intended data structure). However, there are often cases where functions are used for which the code is not available and therefore cannot be modified. In these situations it may still be possible to use hyper-dual numbers, if the effect of computing the derivatives can be achieved. One example of this is the solution of a linear system, $\mathbf{A}\mathbf{y} = \mathbf{b}$, where derivatives can be computed by several calls to the real-valued routine [8]. First derivatives of the solution of a linear system, $\mathbf{A}\mathbf{y} = \mathbf{b}$, can be computed by solving

$$\mathbf{A}\frac{\partial \mathbf{y}}{\partial x_i} = \frac{\partial \mathbf{b}}{\partial x_i} - \frac{\partial \mathbf{A}}{\partial x_i}\mathbf{y}. \tag{3.3}$$

Second derivatives can then be found by solving

$$\mathbf{A}\frac{\partial^2 \mathbf{y}}{\partial x_i \partial x_j} = \frac{\partial^2 \mathbf{b}}{\partial x_i \partial x_j} - \frac{\partial^2 \mathbf{A}}{\partial x_i \partial x_j}\mathbf{y} - \frac{\partial \mathbf{A}}{\partial x_i}\frac{\partial \mathbf{y}}{\partial x_j} - \frac{\partial \mathbf{A}}{\partial x_j}\frac{\partial \mathbf{y}}{\partial x_i}, \tag{3.4}$$

and third derivatives by solving

$$\begin{aligned}\mathbf{A}\frac{\partial^3 \mathbf{y}}{\partial x_i \partial x_j \partial x_k} = \frac{\partial^3 \mathbf{b}}{\partial x_i \partial x_j \partial x_k} - \frac{\partial^3 \mathbf{A}}{\partial x_i \partial x_j \partial x_k}\mathbf{y} - \frac{\partial^2 \mathbf{A}}{\partial x_i \partial x_j}\frac{\partial \mathbf{y}}{\partial x_k} - \frac{\partial^2 \mathbf{A}}{\partial x_i \partial x_k}\frac{\partial \mathbf{y}}{\partial x_j} \\ - \frac{\partial^2 \mathbf{A}}{\partial x_j \partial x_k}\frac{\partial \mathbf{y}}{\partial x_i} - \frac{\partial \mathbf{A}}{\partial x_i}\frac{\partial^2 \mathbf{y}}{\partial x_j \partial x_k} - \frac{\partial \mathbf{A}}{\partial x_j}\frac{\partial^2 \mathbf{y}}{\partial x_i \partial x_k} - \frac{\partial \mathbf{A}}{\partial x_k}\frac{\partial^2 \mathbf{y}}{\partial x_i \partial x_j}.\end{aligned} \tag{3.5}$$

Using hyper-dual numbers to compute derivatives for the parameterization of eigenvalues or CMS matrices requires a hyper-dual version of the eigenvalue calculation routine. As with the solution of a linear system, the effect of using hyper-dual numbers can be recreated without modifying the real-valued eigenvalue solver.

Derivatives of Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are solutions of

$$\left(K - \lambda_\ell M\right)\phi_\ell = F_\ell \phi_\ell = 0, \tag{3.6}$$

with $F_\ell = (K - \lambda_\ell M)$. This equation can be differentiated to give

$$\frac{\partial F_\ell}{\partial x_i}\phi_\ell + F_\ell \frac{\partial \phi_\ell}{\partial x_i} = 0. \tag{3.7}$$

Pre-multiplying this equation by the transpose of the eigenvector, and making use of the fact that $F_\ell \phi_\ell = 0$, yields

$$\phi_\ell^T \frac{\partial F_\ell}{\partial x_i}\phi_\ell = \phi_\ell^T \left(\frac{\partial K}{\partial x_i} - \lambda_\ell \frac{\partial M}{\partial x_i} - \frac{\partial \lambda_\ell}{\partial x_i} M\right)\phi_\ell. \tag{3.8}$$

The eigenvectors are orthonormal with respect to M, so $\phi_\ell^T M \phi_\ell = 1$, and (3.8) can be rearranged to give the first derivative of the eigenvalue

$$\frac{\partial \lambda_\ell}{\partial x_i} = \phi_\ell^T \left(\frac{\partial K}{\partial x_i} - \lambda_\ell \frac{\partial M}{\partial x_i} \right) \phi_\ell. \tag{3.9}$$

There are several methods for computing the first derivatives of eigenvectors, as summarized by Alvin [9]. The method of Nelson [10] is exact but can be computationally expensive since it involves solving a linear system for each derivative. The modal superposition method [11] represents the derivative of an eigenvector as a superposition of the other eigenvectors and is less computationally intensive than Neslon's method.

Figure 3.1 shows the first derivative of the second eigenvector with respect to the cross-sectional height computed using finite differences, Nelson's method, and modal superposition. Nelson's method and the finite-difference calculation are in good agreement. The first derivative of the eigenvector has sharp corners where the individual components join together. Modal superposition requires the use of all eigenvectors in order to be exact. Using a smaller subset results in an approximation. Figure 3.1 shows the result of modal superposition with 20 modes. The modal superposition method does not capture the sharp corners, and exhibits Gibb's phenomena, as would be expected of a method relying on a finite summation of modes. When the application is CMS, only a few eigenvectors are kept, so the computational cost of Nelson's method is not expected to be an issue and is preferred over modal superposition.

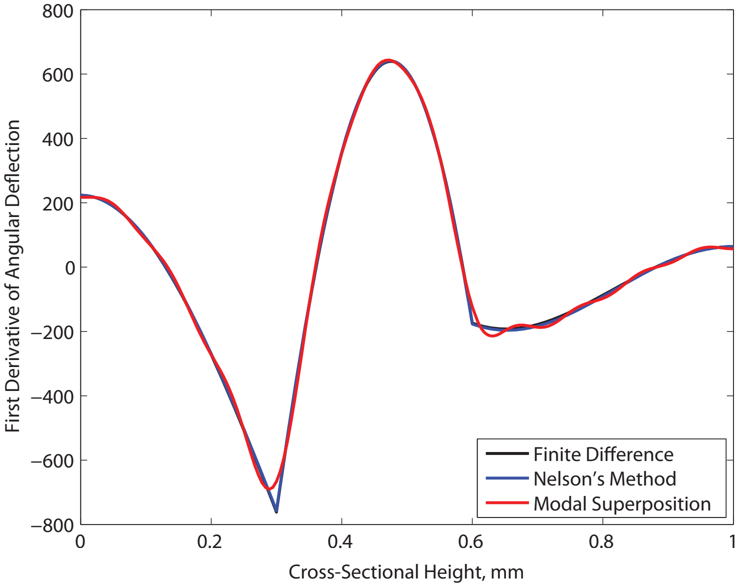


Figure 3.1. A comparison of three methods for computing derivatives of eigenvectors.

Nelson's method for computing the first derivative of an eigenvector is to represent it as a sum of two terms

$$\frac{\partial \phi_\ell}{\partial x_i} = z_i + c_i \phi_\ell. \tag{3.10}$$

The quantity z_i is found by solving (3.7) with $\frac{\partial \phi_\ell}{\partial x_i}$ replaced by z_i,

$$F_\ell z_i = -\frac{\partial F_\ell}{\partial x_i} \phi_\ell. \tag{3.11}$$

The contribution of ϕ_ℓ is added back in by multiplying by c_i, where c_i is found by differentiation the orthonormalization equation

$$\phi_j^T M \phi_i = \delta_{ij}, \tag{3.12}$$

where δ_{ij} is the Kronecker delta function. The first derivative of the orthonormalization equation is

$$\phi^T \frac{\partial M}{\partial x_i} \phi + 2\phi^T M \frac{\partial \phi_\ell}{\partial x_i} = 0. \tag{3.13}$$

Substituting Eq. 3.10 into this equation produces

$$\phi^T \frac{\partial M}{\partial x_i} \phi + 2\phi^T M \left(z_i + c_i \phi_\ell \right) = 0, \tag{3.14}$$

and c_i can then be computed as

$$c_i = -\frac{1}{2} \phi_\ell^T \frac{\partial M}{\partial x_i} \phi_\ell - \phi_\ell^T M z_i. \tag{3.15}$$

Expressions for second and third derivatives can be derived following a similar procedure. The second derivative of an eigenvalue can be computed as

$$\frac{\partial^2 \lambda_\ell}{\partial x_i \partial x_j} = \phi_\ell^T \left(\frac{\partial^2 K}{\partial x_i \partial x_j} - \frac{\partial \lambda_\ell}{\partial x_i} \frac{\partial M}{\partial x_j} - \frac{\partial \lambda_\ell}{\partial x_j} \frac{\partial M}{\partial x_i} - \lambda \frac{\partial^2 M}{\partial x_i \partial x_j} \right) \phi_\ell + \phi_\ell^T \frac{\partial F_\ell}{\partial x_i} \frac{\partial \phi_\ell}{\partial x_j} + \phi_\ell^T \frac{\partial F_\ell}{\partial x_j} \frac{\partial \phi_\ell}{\partial x_i}, \tag{3.16}$$

and the second derivative of the corresponding eigenvector can then be computed as

$$\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_j} = z_{ij} + c_{ij} \phi_\ell, \tag{3.17}$$

where

$$c_{ij} = -\frac{1}{2} \phi_\ell^T \frac{\partial^2 M}{\partial x_i \partial x_j} \phi_\ell - \phi_\ell^T \frac{\partial M}{\partial x_i} \frac{\partial \phi_\ell}{\partial x_j} - \phi_\ell^T \frac{\partial M}{\partial x_j} \frac{\partial \phi_\ell}{\partial x_i} - \frac{\partial \phi_\ell^T}{\partial x_j} M \frac{\partial \phi_\ell}{\partial x_i} - \phi_\ell^T M z_{ij}. \tag{3.18}$$

The term z_{ij} is found by solving

$$F_\ell z_{ij} = -\frac{\partial^2 F_\ell}{\partial x_i \partial x_j}\phi_\ell - \frac{\partial F_\ell}{\partial x_i}\frac{\partial \phi_\ell}{\partial x_j} - \frac{\partial F_\ell}{\partial x_j}\frac{\partial \phi_\ell}{\partial x_i}. \tag{3.19}$$

The expression for the third derivative of an eigenvalue is

$$\begin{aligned}\frac{\partial^3 \lambda_\ell}{\partial x_i \partial x_j \partial x_k} = \phi_\ell^T \Bigg(& \frac{\partial^3 K}{\partial x_i \partial x_j \partial x_k} - \frac{\partial^2 \lambda_\ell}{\partial x_j \partial x_k}\frac{\partial M}{\partial x_i} - \frac{\partial^2 \lambda_\ell}{\partial x_i \partial x_k}\frac{\partial M}{\partial x_j} - \frac{\partial^2 \lambda_\ell}{\partial x_i \partial x_j}\frac{\partial M}{\partial x_k} - \frac{\partial \lambda_\ell}{\partial x_i}\frac{\partial^2 M}{\partial x_j \partial x_k} \\ & - \frac{\partial \lambda_\ell}{\partial x_j}\frac{\partial^2 M}{\partial x_i \partial x_k} - \frac{\partial \lambda_\ell}{\partial x_k}\frac{\partial^2 M}{\partial x_i \partial x_j} - \lambda \frac{\partial^3 M}{\partial x_i \partial x_j \partial x_k}\Bigg)\phi_\ell + \phi_\ell^T \frac{\partial^2 F_\ell}{\partial x_j \partial x_k}\frac{\partial \phi_\ell}{\partial x_i} + \phi_\ell^T \frac{\partial^2 F_\ell}{\partial x_i \partial x_k}\frac{\partial \phi_\ell}{\partial x_j} \\ & + \phi_\ell^T \frac{\partial^2 F_\ell}{\partial x_i \partial x_j}\frac{\partial \phi_\ell}{\partial x_k} + \phi_\ell^T \frac{\partial F_\ell}{\partial x_i}\frac{\partial^2 \phi_\ell}{\partial x_j \partial x_k} + \phi_\ell^T \frac{\partial F_\ell}{\partial x_j}\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_k} + \phi_\ell^T \frac{\partial F_\ell}{\partial x_k}\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_j}. \end{aligned} \tag{3.20}$$

The third derivative of the corresponding eigenvector can then be computed as

$$\frac{\partial^3 \phi_\ell}{\partial x_i \partial x_j \partial x_k} = z_{ijk} + c_{ijk}\phi_\ell, \tag{3.21}$$

where

$$\begin{aligned} c_{ijk} = & -\frac{1}{2}\phi_\ell^T \frac{\partial^3 M}{\partial x_i \partial x_j \partial x_k}\phi_\ell - \phi_\ell^T \frac{\partial^2 M}{\partial x_i \partial x_j}\frac{\partial \phi_\ell}{\partial x_k} - \phi_\ell^T \frac{\partial^2 M}{\partial x_i \partial x_k}\frac{\partial \phi_\ell}{\partial x_j} - \phi_\ell^T \frac{\partial^2 M}{\partial x_j \partial x_k}\frac{\partial \phi_\ell}{\partial x_i} - \frac{\partial \phi_\ell^T}{\partial x_k}\frac{\partial M}{\partial x_i}\frac{\partial \phi_\ell}{\partial x_j} \\ & - \frac{\partial \phi_\ell^T}{\partial x_k}\frac{\partial M}{\partial x_j}\frac{\partial \phi_\ell}{\partial x_i} - \frac{\partial \phi_\ell^T}{\partial x_k} M \frac{\partial^2 \phi_\ell}{\partial x_i \partial x_j} - \frac{\partial \phi_\ell^T}{\partial x_j}\frac{\partial M}{\partial x_k}\frac{\partial \phi_\ell}{\partial x_i} - \frac{\partial^2 \phi_\ell^T}{\partial x_j \partial x_k} M \frac{\partial \phi_\ell}{\partial x_i} - \frac{\partial \phi_\ell^T}{\partial x_j} M \frac{\partial^2 \phi_\ell}{\partial x_i \partial x_k} \\ & - \phi_\ell^T \frac{\partial M}{\partial x_i}\frac{\partial^2 \phi_\ell}{\partial x_j \partial x_k} - \phi_\ell^T \frac{\partial M}{\partial x_j}\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_k} - \phi_\ell^T \frac{\partial M}{\partial x_k}\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_j} - \phi_\ell^T M z_{ijk}, \end{aligned} \tag{3.22}$$

and z_{ijk} is found by solving

$$\begin{aligned} F_\ell z_{ijk} = & -\frac{\partial^3 F_\ell}{\partial x_i \partial x_j \partial x_k}\phi_\ell - \frac{\partial^2 F_\ell}{\partial x_j \partial x_k}\frac{\partial \phi_\ell}{\partial x_i} - \frac{\partial^2 F_\ell}{\partial x_i \partial x_k}\frac{\partial \phi_\ell}{\partial x_j} - \frac{\partial^2 F_\ell}{\partial x_i \partial x_j}\frac{\partial \phi_\ell}{\partial x_k} \\ & - \frac{\partial F_\ell}{\partial x_i}\frac{\partial^2 \phi_\ell}{\partial x_j \partial x_k} - \frac{\partial F_\ell}{\partial x_j}\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_k} - \frac{\partial F_\ell}{\partial x_k}\frac{\partial^2 \phi_\ell}{\partial x_i \partial x_j}. \end{aligned} \tag{3.23}$$

Description of MATLAB Files

This section describes the MATLAB analysis routines used to investigate parameterization at the various levels shown in Figure 1.2.

@hyperdual2

Directory containing the class implementation of hyper-dual numbers for second-derivative calculations.

@hyperdual3

Directory containing the class implementation of hyper-dual numbers for third-derivative calculations.

ParameterSweep_FEM.m

This function performs a sweep through the parameter variations calling AnalyzePerturbedBeam_FEM.m for each perturbation. The results of this function are used to assess the accuracy of the parameterization approaches applied to the FEM branch of Figure 1.2.

ParameterSweep_CMS.m

This function performs a sweep through the parameter variations calling AnalyzePerturbedBeam_CMS.m for each perturbation. The results of this function are used to assess the accuracy of the parameterization approaches applied to the CMS branch of Figure 1.2.

ParameterizeEigenvalues_FEM.m

This function computes the eigenvalues and their derivatives for the nominal design, following the analysis procedure shown in the FEM branch of Figure 1.1. A parameterized model of the eigenvalues is then computed using the derivatives at the nominal design.

ParameterizeEigenvalues_CMS.m

This function computes the eigenvalues and their derivatives for the nominal design, following the analysis procedure shown in the CMS branch of Figure 1.1. A parameterized model of the eigenvalues is then computed using the derivatives at the nominal design.

ParameterizeComponentMatrices_FEM.m

This function computes parameterized versions of the component matrices using hyper-dual numbers to compute the derivatives. A sweep through the allowable variations is then performed using the parameterized component matrices.

ParameterizeComponentMatrices_CMS.m

This function computes parameterized versions of the component matrices using hyper-dual numbers to compute the derivatives. A sweep through the allowable variations is then performed using CMS with the parameterized component matrices.

ParameterizeCraigBamptonMatrices_CMS.m

This function computes parameterized versions of the C-B reduced matrices for the components using hyper-dual numbers to compute the derivatives. A sweep through the allowable variations is then performed using the parameterized CMS matrices.

ParameterizeSystemMatrices_FEM.m

This function computes parameterized versions of the system matrices using hyper-dual numbers to compute the derivatives. Eigenvalues are then computed for the allowable variations using the parameterized system matrices.

ParameterizeCraigBamptonSystem_CMS.m

This function computes parameterized versions of the C-B system matrices, after the reduced component matrices have been assembled, with hyper-dual numbers used to compute the derivatives. Eigenvalues are then computed for the allowable variations using the parameterized CMS system matrices.

Chapter 4

Discussion of Results

The example problem being considered is a simply supported beam composed of three components, as shown in Figure 4.1. The beam has elastic modulus E, density ρ, width B, cross-sectional height H, location of the center of the defect ℓ, and length of the defect W, with nominal values given in Table 4.1. The material properties and geometry of the center section are allowed to be perturbed from their nominal values.

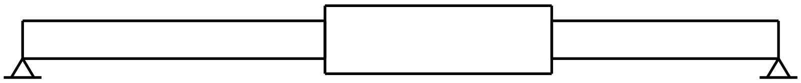


Figure 4.1. A simply supported beam composed of three components.

Property	Value
Property	Value
Density, ρ	2700 kg/m^3
Elastic modulus, E	68.9 GPa
Cross-sectional width, B	20 cm
Cross-sectional height, H	5 mm
Location of defect's center, ℓ	45 cm
Length of the defect, W	30 cm
Length of the beam, L	1 m

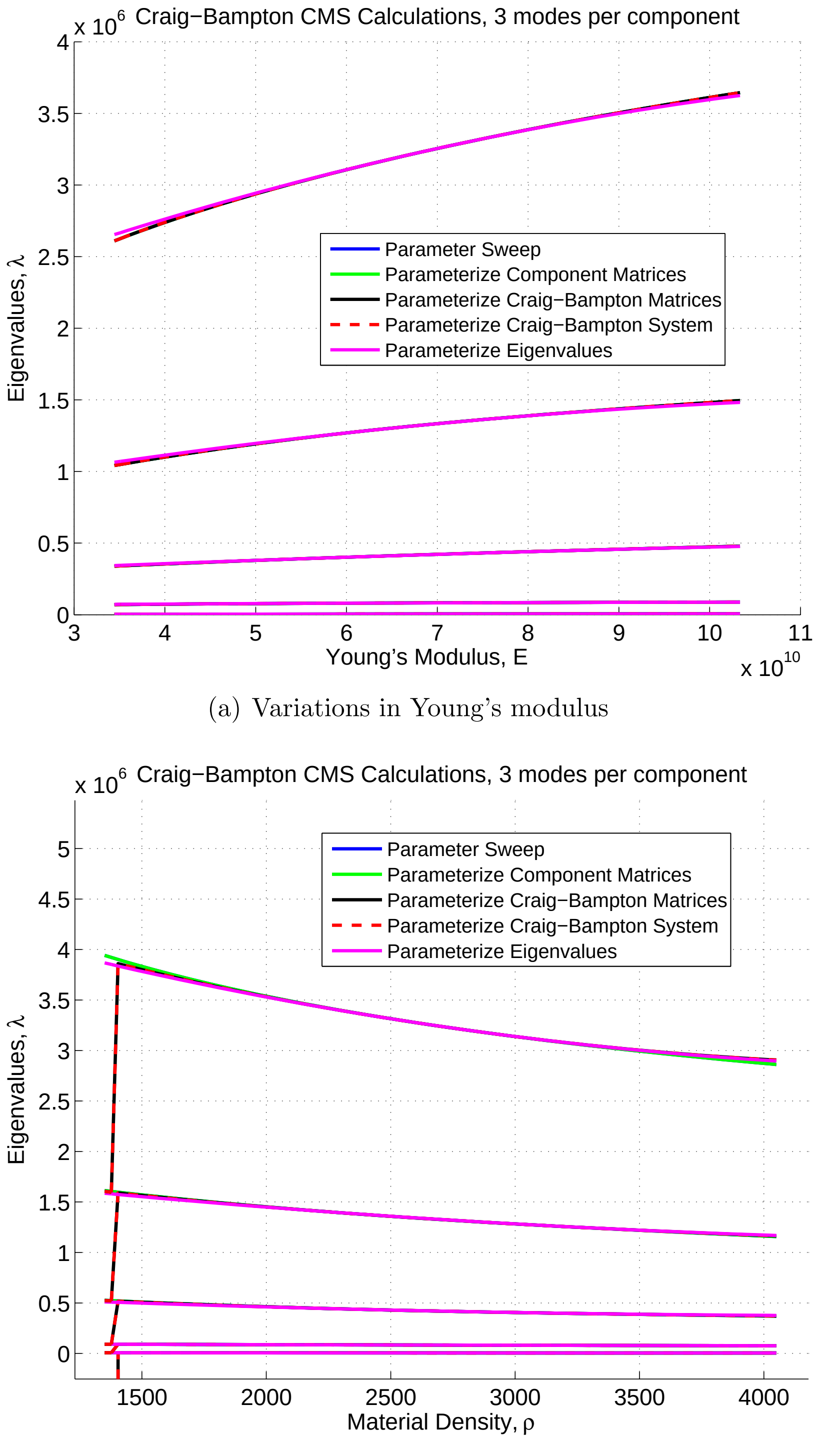
Table 4.1. Material and geometric properties for the beam.

Two types of parameterizations are considered: quadratic and cubic. These parameterizations are applied at all the levels shown in Figure 1.1, although only results from the CMS branch are presented here. A parameter sweep was run for each case to determine the true behavior in order to assess the accuracy of the parameterized models. The CMS calculations are performed by keeping only three modes per component. This seems to produce accurate results although more testing is needed on the effect of varying the number of modes kept. Parameter variations are considered for each of the variables in Table 4.1, except for L.

The quadratic parameterization given in Eq. 1.1 is used to produce the comparisons shown in Figs. 4.2-4.4. Detailed comparisons for each case are provided for the first three eigenvalues in Figs. 4.5-4.10. Quadratic parameterization produces fairly accurate results for variations in Young's modulus, material density and cross-sectional width and height. There are some issues with the CMS parameterizations at the extremes of the parameter space. For the quadratic parameterization, variations in Young's modulus, material density and cross-sectional width are exactly represented to machine precision by parameterizing the component matrices. Parameterizing the CMS matrices produces slightly less accurate results, and parameterizing the eigenvalues produces fairly accurate results. Quadratic parameterization of the full system matrices and CMS matrices is not able to accurately capture the effects of variations in cross-sectional height. Here, the system matrices are composed to represent bending stiffness (as opposed to the cross-sectional height). Bending stiffness relates to the cube of cross-sectional height, whereas the other parameters factor into the system matrices in a linear manner. For variations in location and length of the center component the parameterizations are only accurate in a small region around the nominal design. For these variations, parameterizing the eigenvalues is the most accurate, followed by parameterizing the CMS matrices, with parameterizing the component matrices the least accurate.

Cubic parameterizations, as given in 1.2, produce similar trends but are in general more accurate. The accuracy of the cubic parameterization is shown in Fig. **??**, with detailed comparisons for the first three eigenvalues of each case given in Figs. 4.14-4.19. The cubic parameterization applied to the component matrices is able to accurately represent the effect of varying the cross-sectional height, and also improves the accuracy of parameterization of the CMS matrices. The geometric variations, location and length of the center component, are accurate only in a small region around the nominal design but the accuracy is better than the quadratic parameterization. For the geometric variations, parameterizing the eigenvalues is most accurate, followed by parameterizing the CMS matrices. Parameterizing the component matrices is the least accurate approach for the geometric variations. These trends are consistent with those for using real-valued finite difference methods to construct the parameterizations instead of hyper-dual numbers.

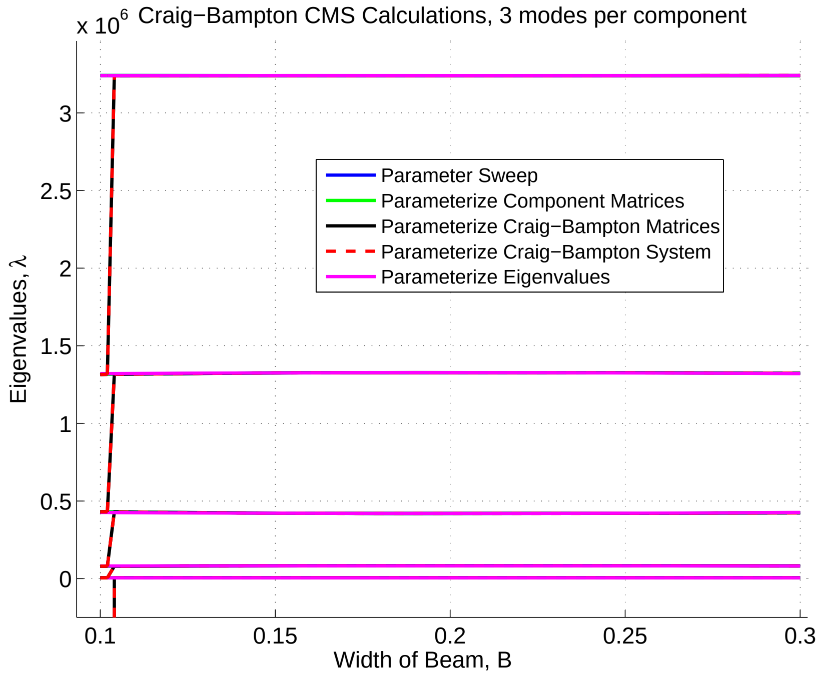
Parameters related to the geometric changes of the system, specifically the center location and width, are expected to be more difficult to model due to their nonlinear effects on the stiffness matrix as they are varied. Other parameters, that affect the system in only a bulk sense such as B or H, or that are material properties such as E or ρ, are expected to be easier to parameterize as varying them linearly varies the system matrices linearly as well.



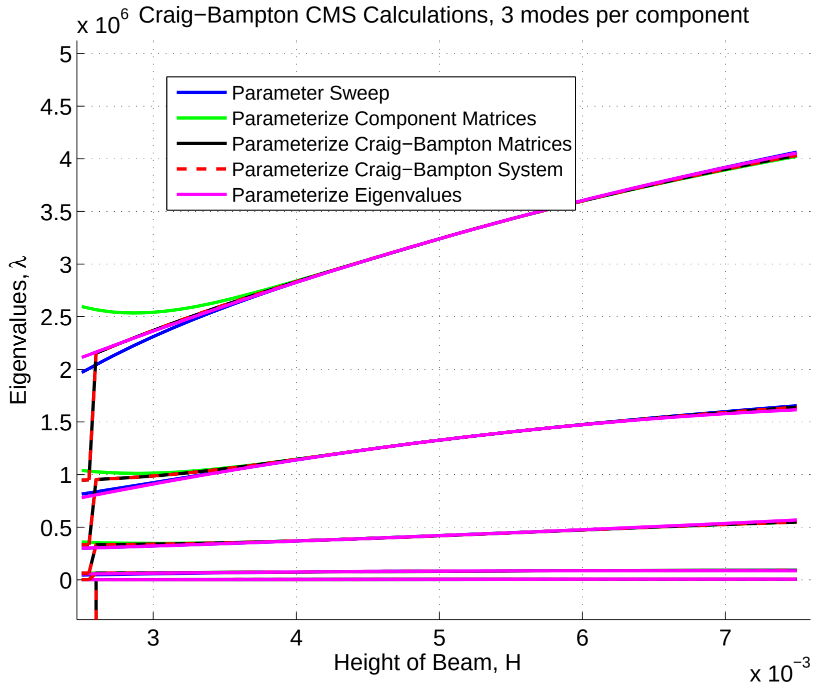
(a) Variations in Young's modulus

(b) Variations in material density

Figure 4.2. The effects on the first five eigenvalues of quadratic parameterizations for E and ρ.

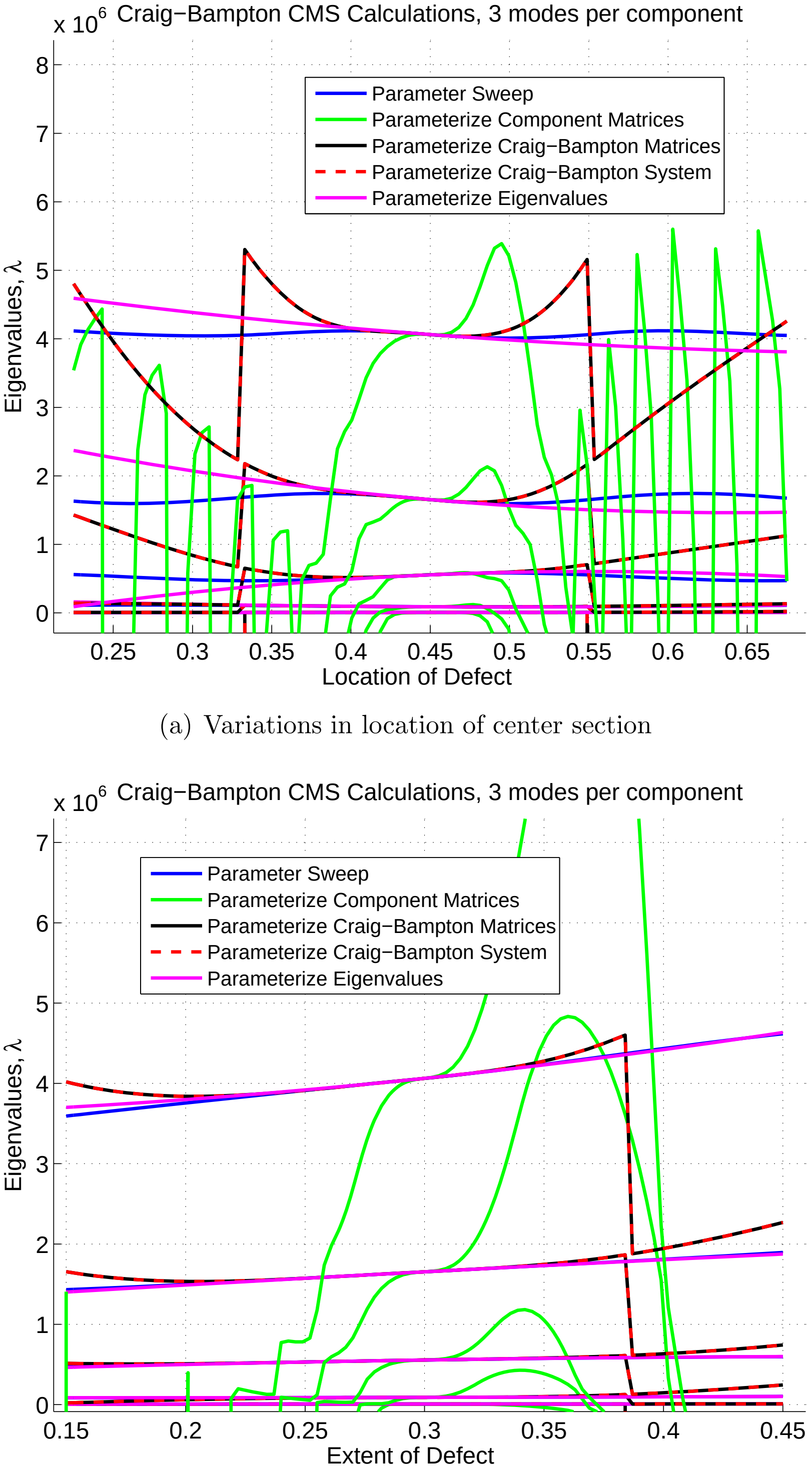


(a) Variations in cross-sectional width



(b) Variations in cross-sectional height

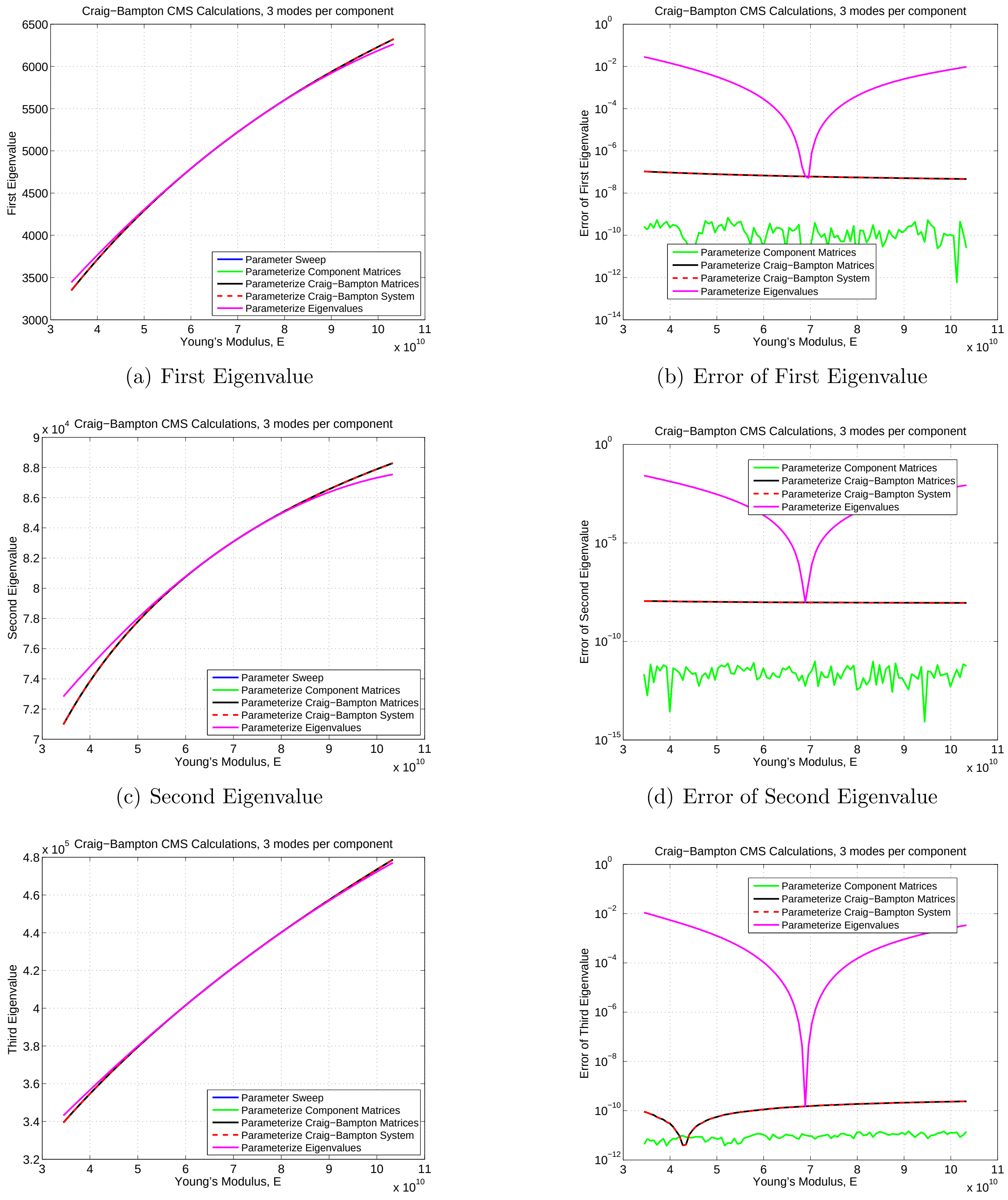
Figure 4.3. The effects on the first five eigenvalues of quadratic parameterizations for B and H.



(a) Variations in location of center section

(b) Variations in length of center section

Figure 4.4. The effects on the first five eigenvalues of quadratic parameterizations for ℓ and W.



(a) First Eigenvalue

(b) Error of First Eigenvalue

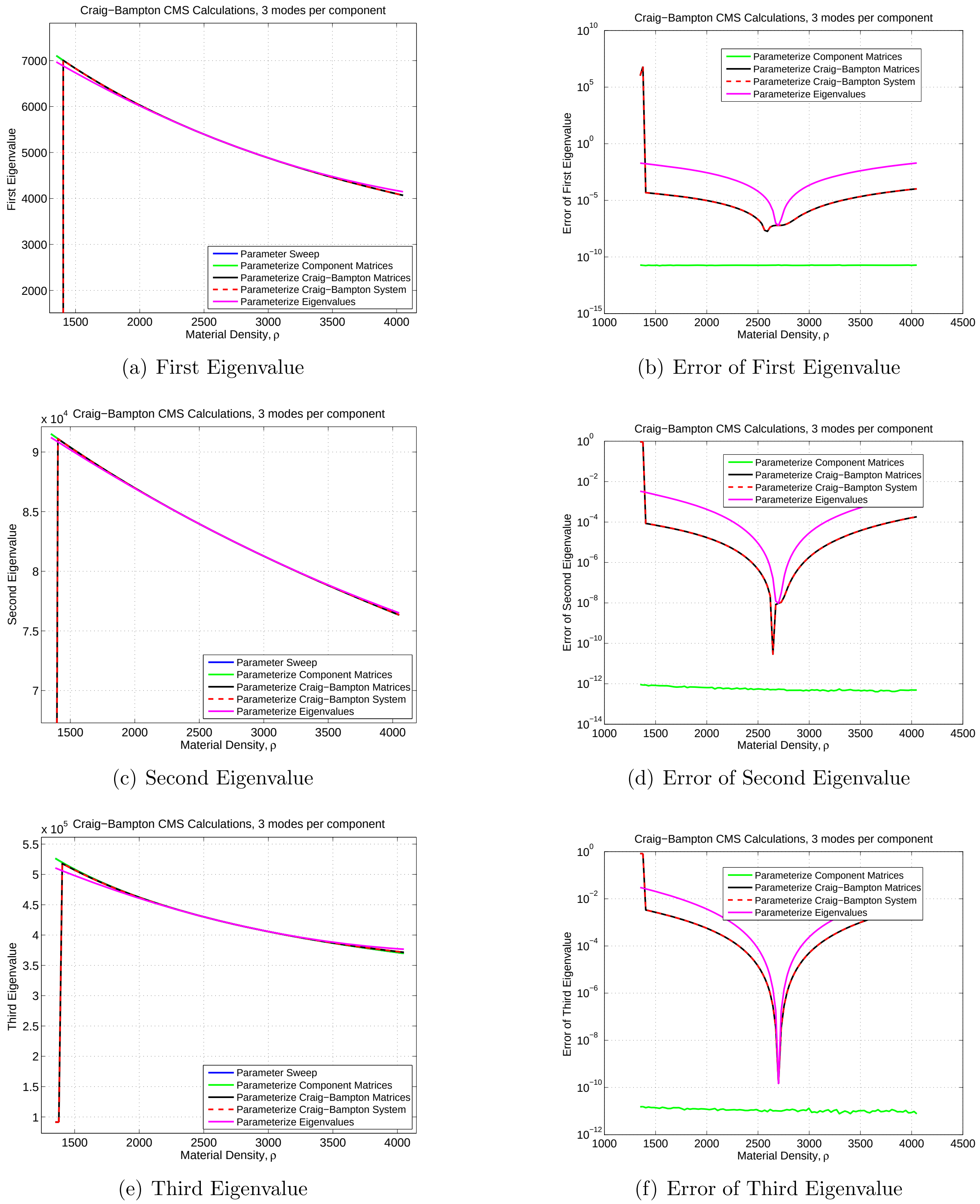
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.5. A comparison of quadratic parameterizations for variations in Young's modulus, E, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

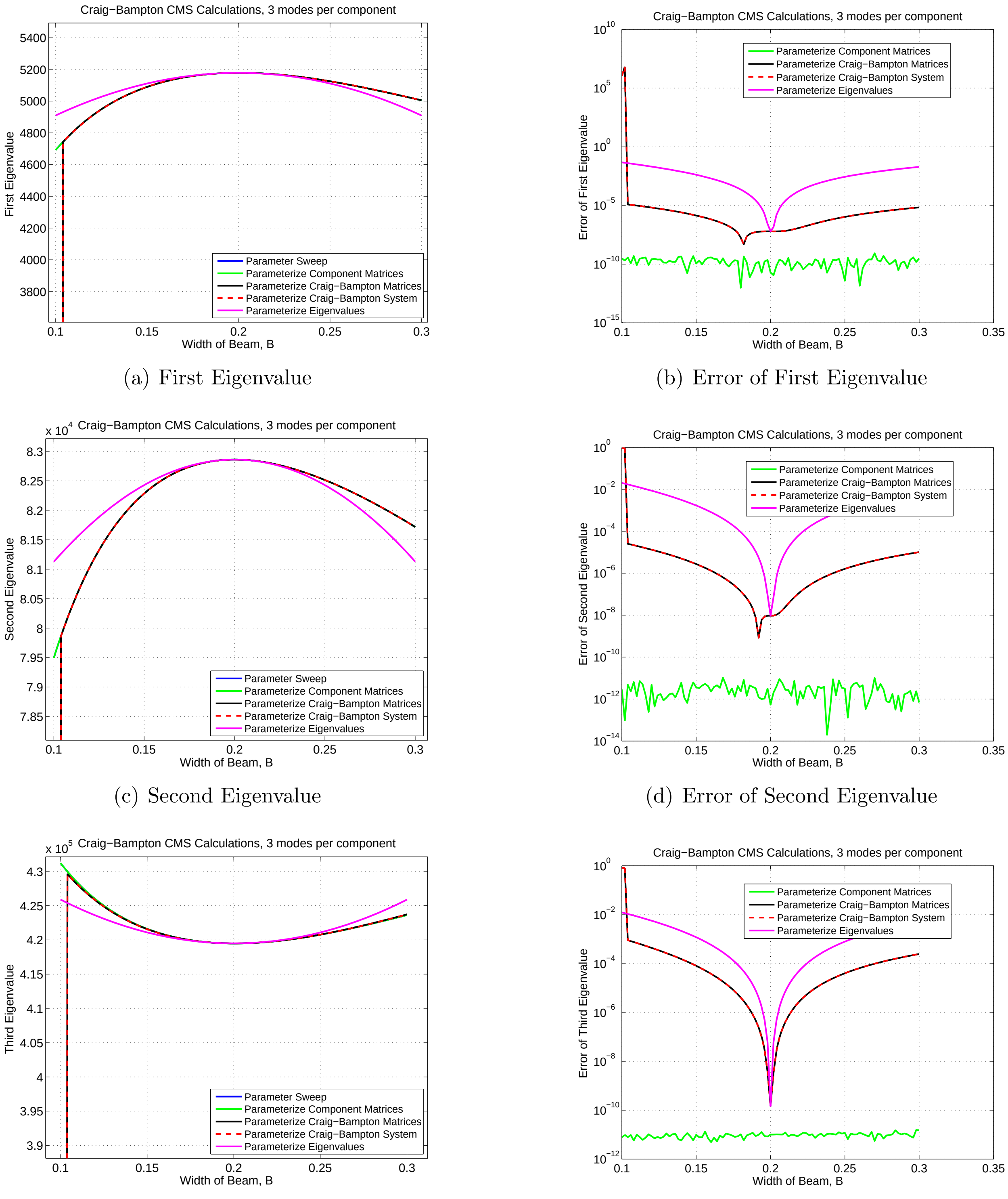
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.6. A comparison of quadratic parameterizations for variations in material density, ρ, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

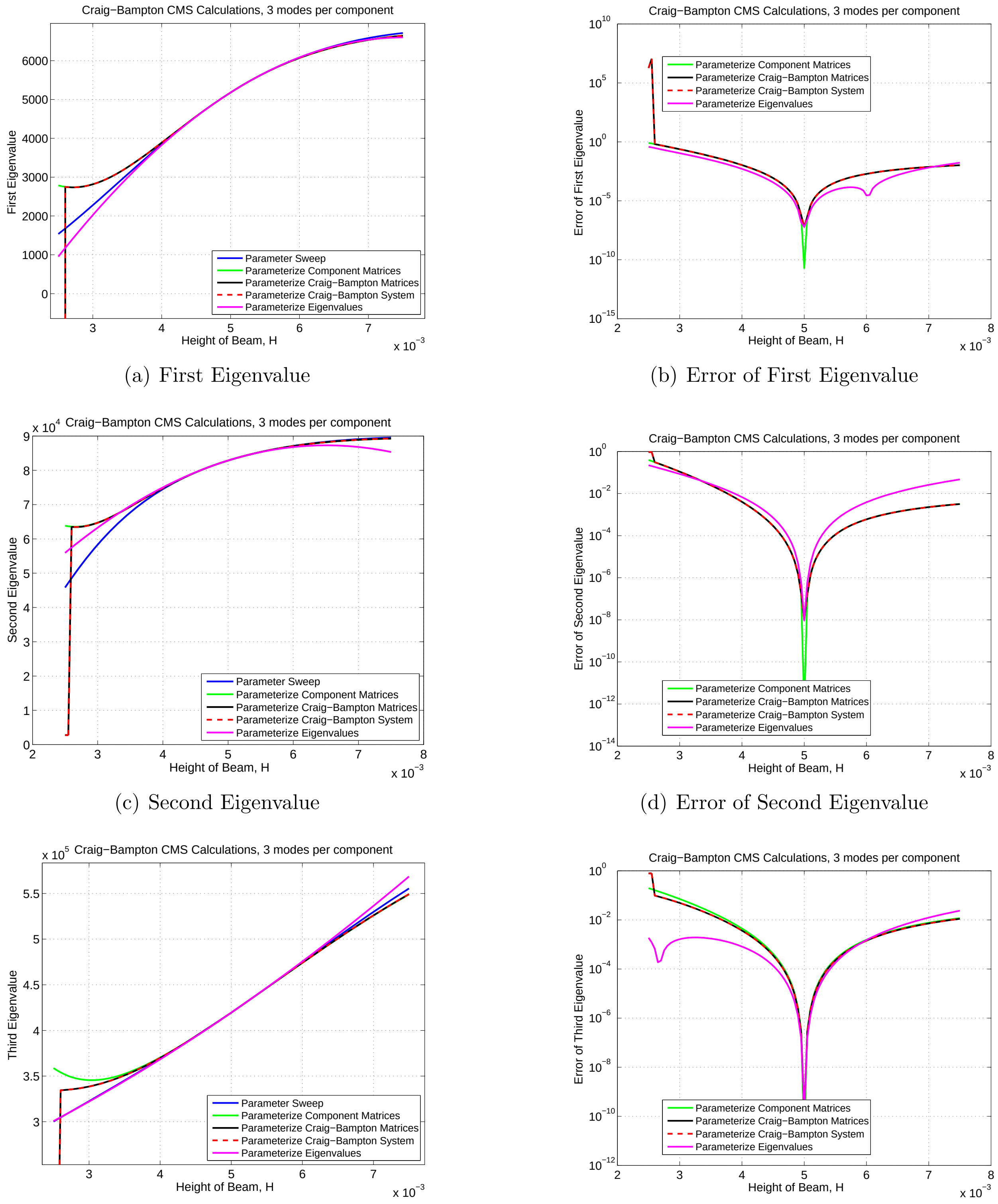
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.7. A comparison of quadratic parameterizations for variations in cross-sectional width, B, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

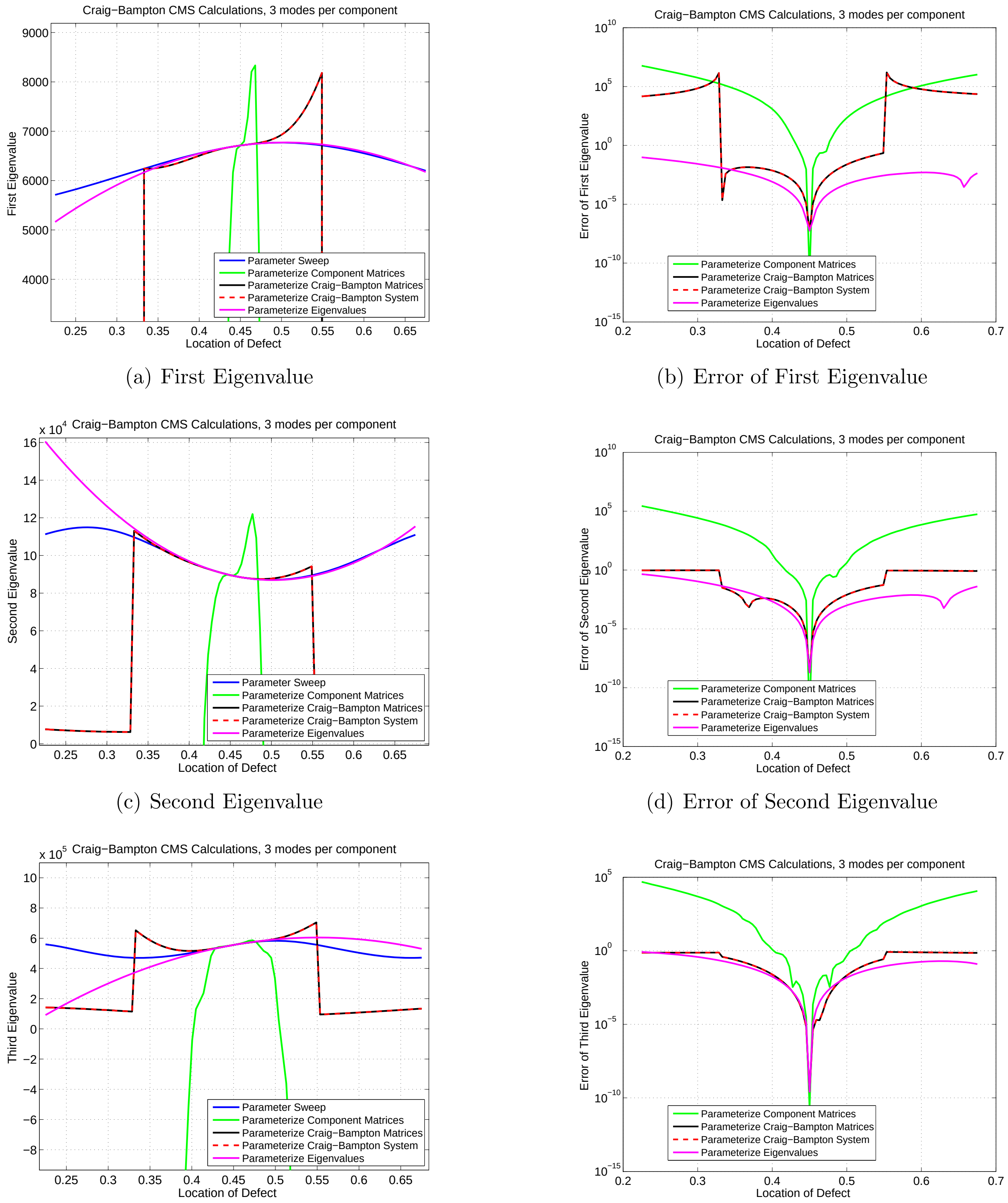
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.8. A comparison of quadratic parameterizations for variations in cross-sectional height, H, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

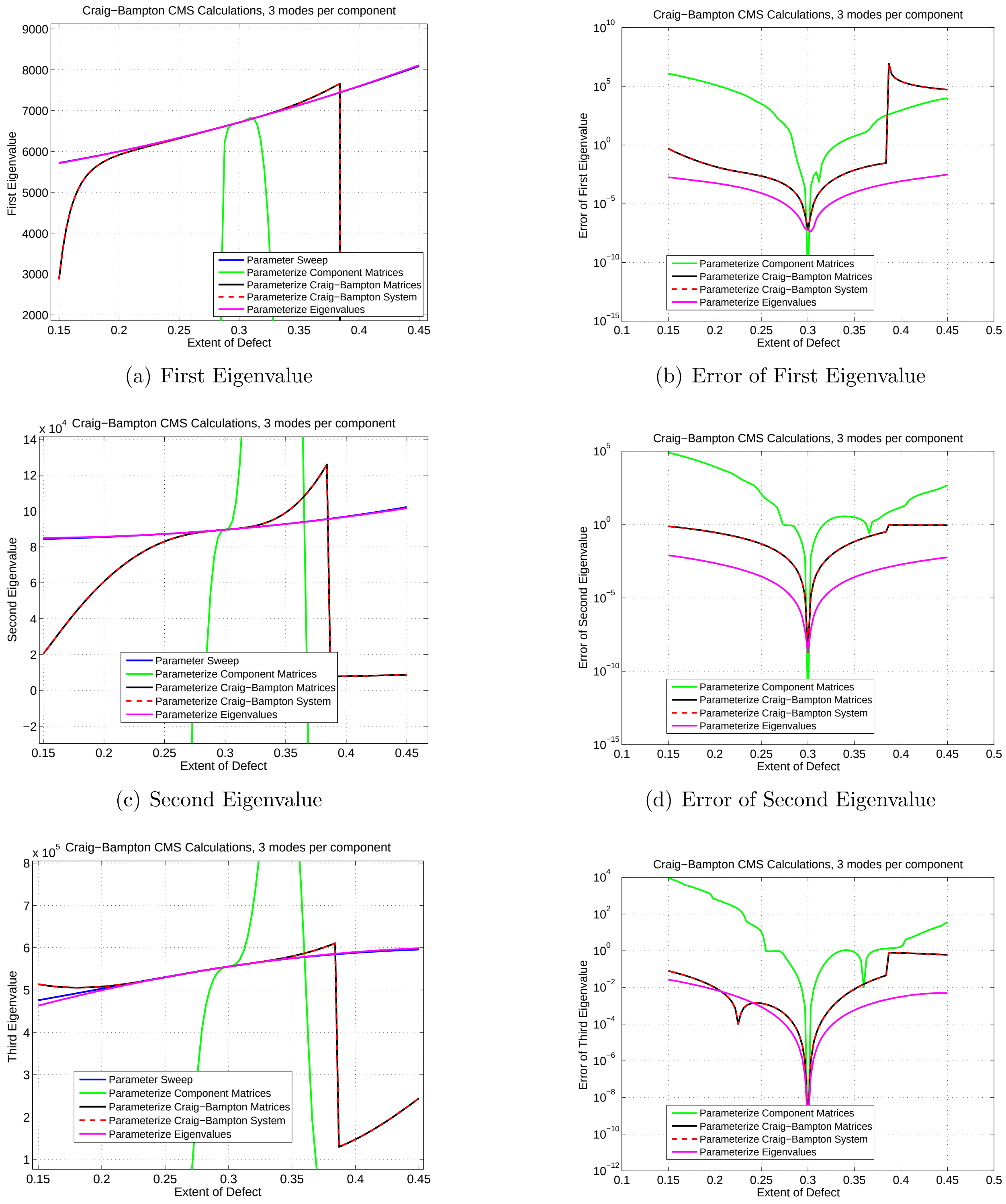
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.9. A comparison of quadratic parameterizations for variations in location of the center section, ℓ, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

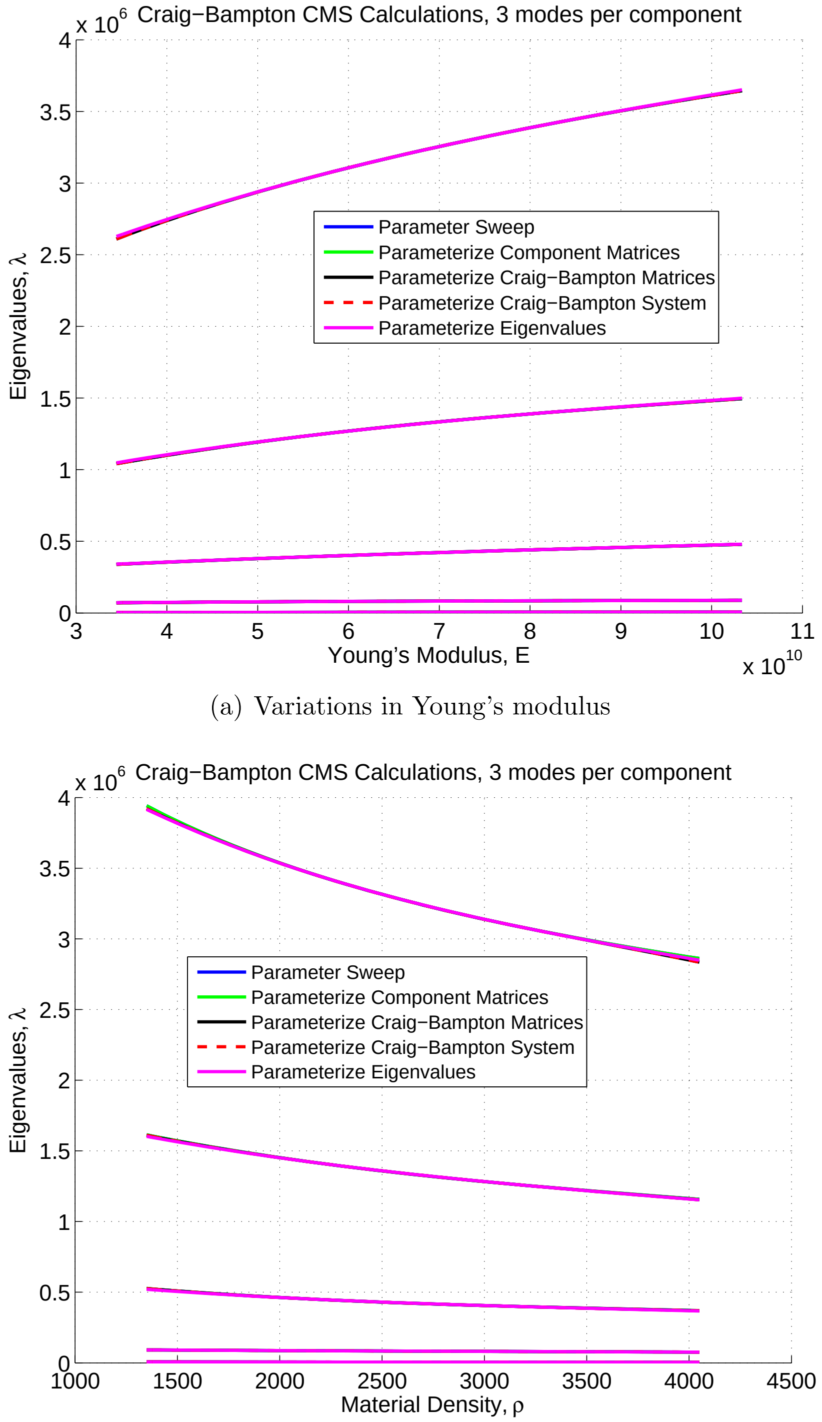
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

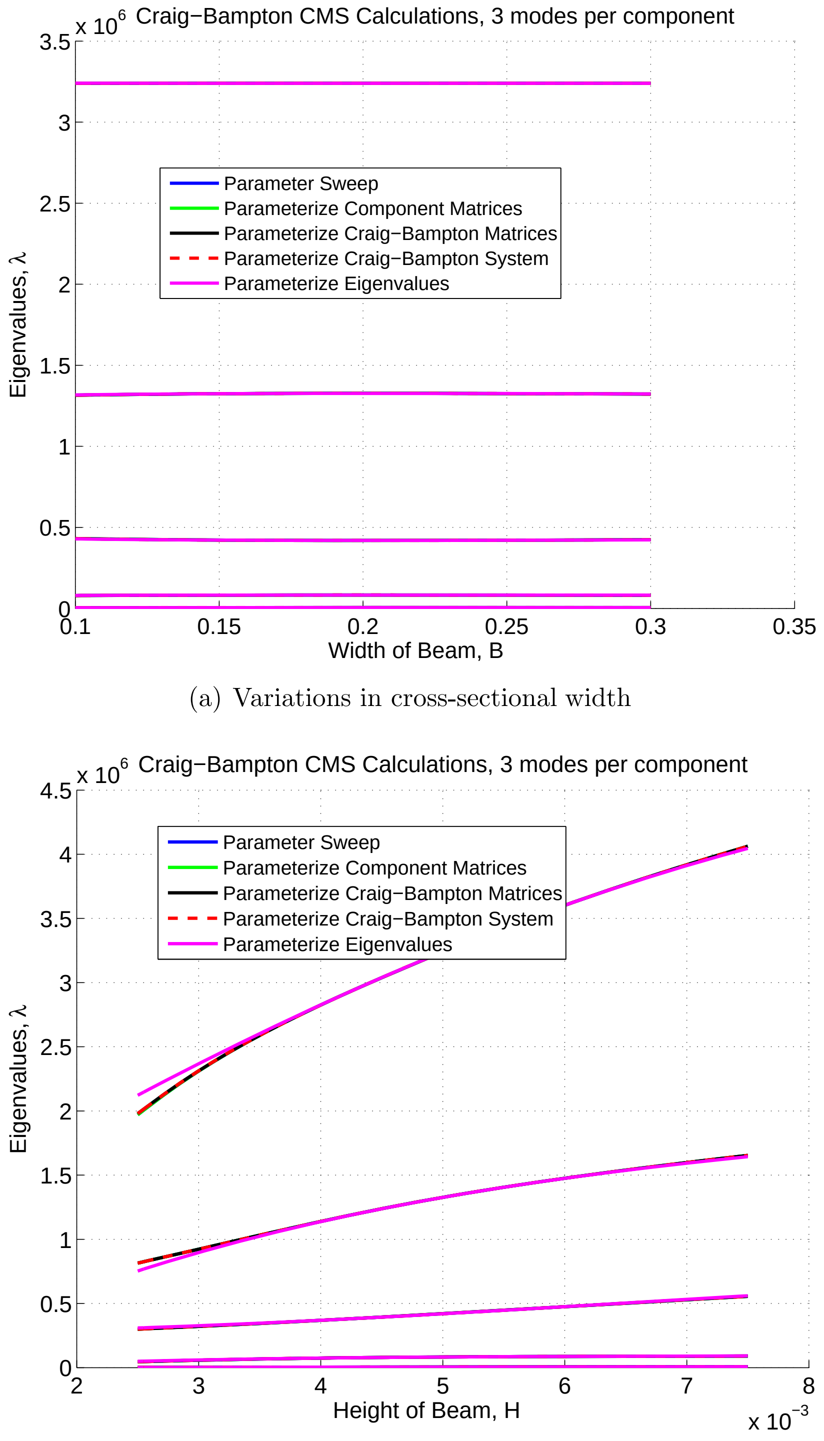
Figure 4.10. A comparison of quadratic parameterizations for variations in length of the center section, W, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) Variations in Young's modulus

(b) Variations in material density

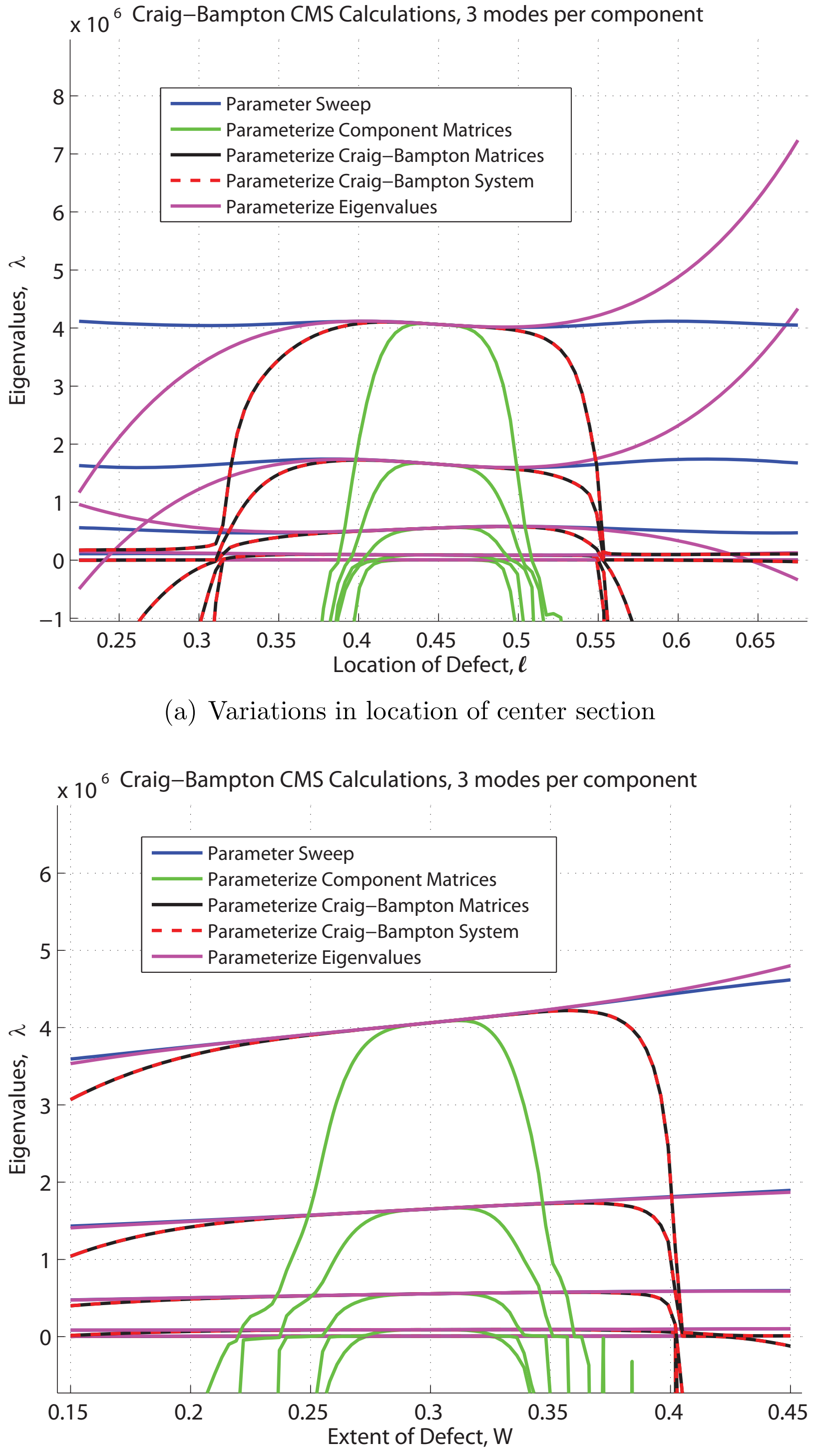
Figure 4.11. The effects on the first five eigenvalues of cubic parameterizations for E and ρ.



(a) Variations in cross-sectional width

(b) Variations in cross-sectional height

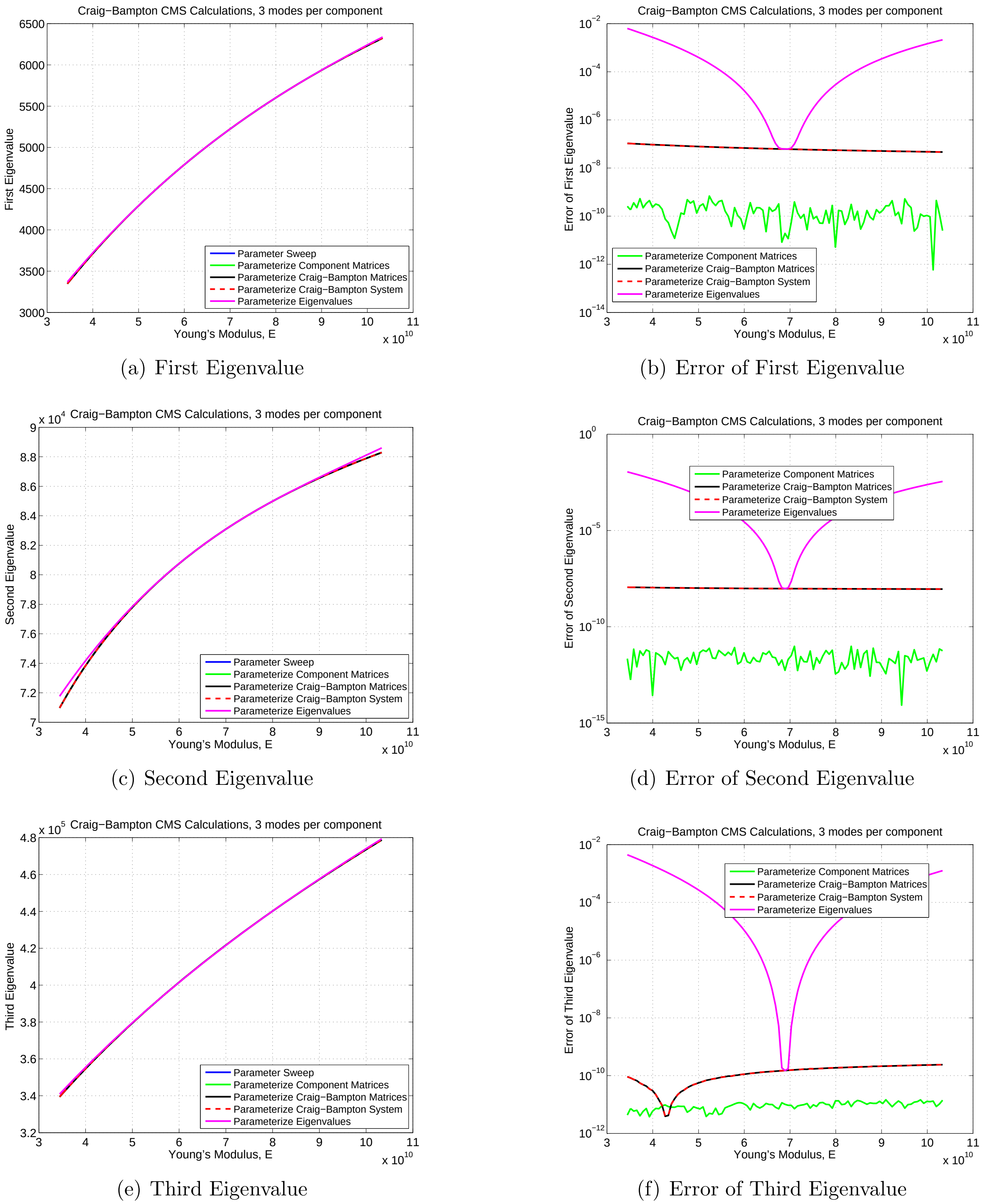
Figure 4.12. The effects on the first five eigenvalues of cubic parameterizations for B and H.



(a) Variations in location of center section

(b) Variations in length of center section

Figure 4.13. The effects on the first five eigenvalues of cubic parameterizations for ℓ and W.



(a) First Eigenvalue

(b) Error of First Eigenvalue

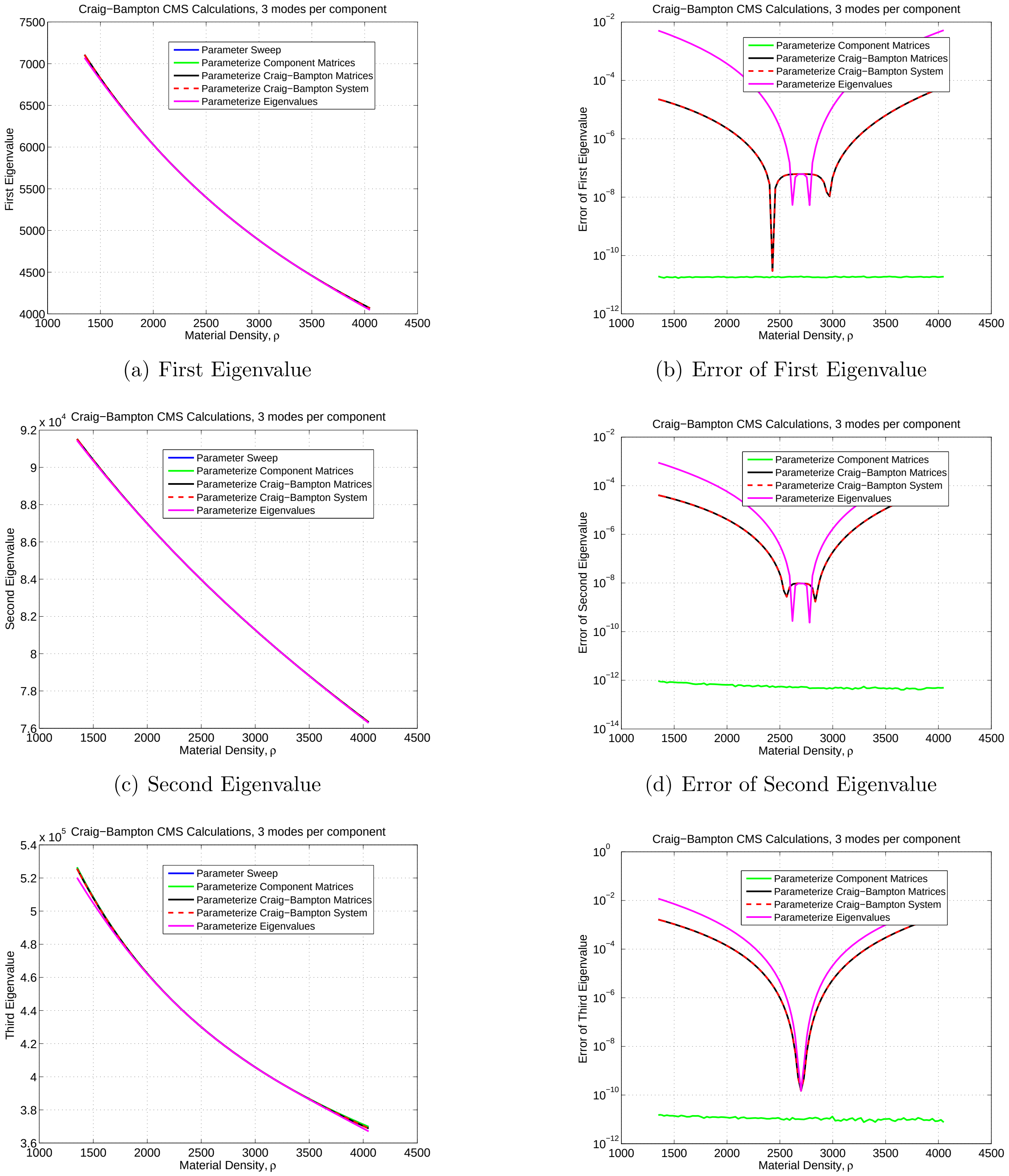
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.14. A comparison of cubic parameterizations for variations in Young's modulus, E, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

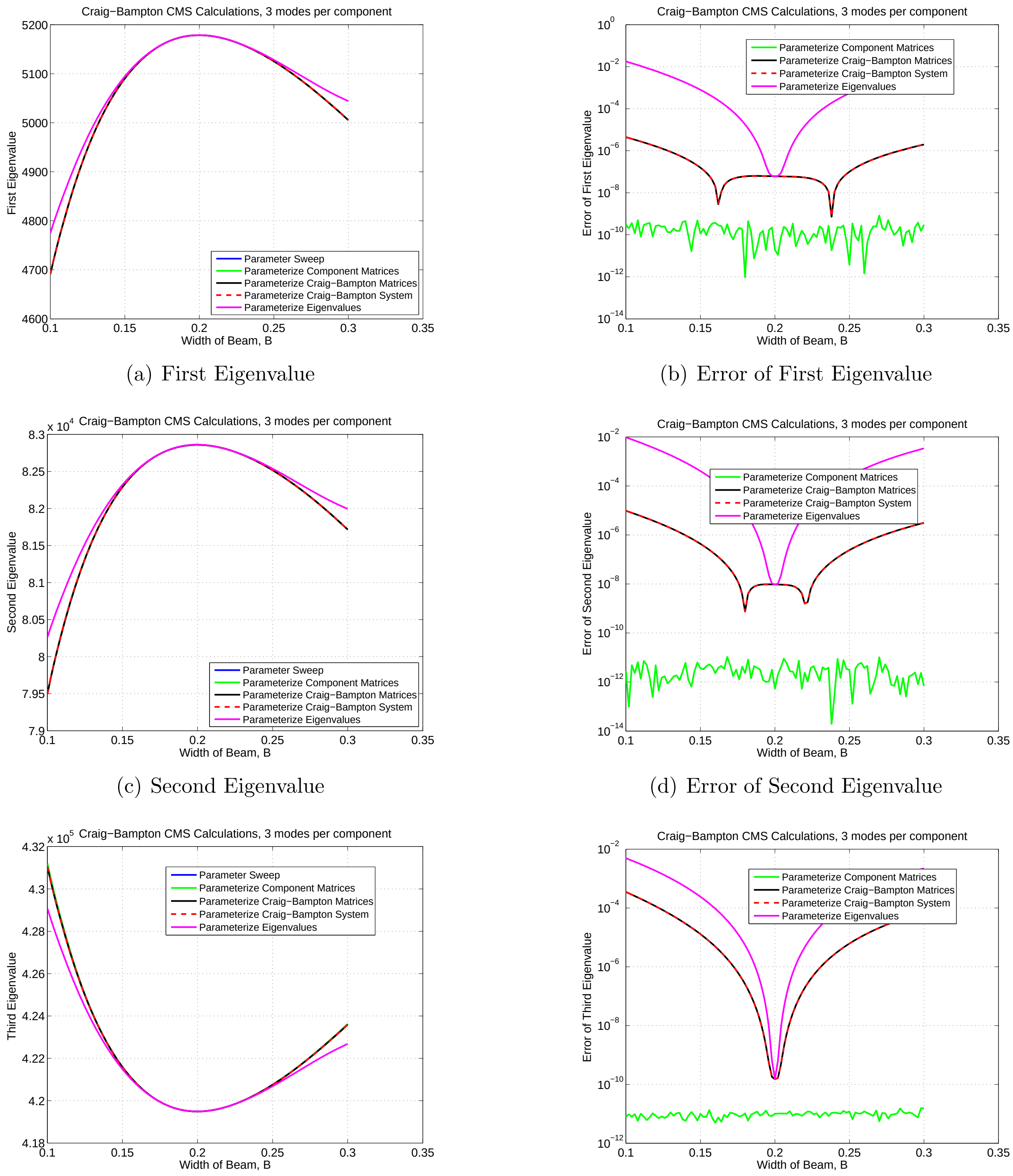
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.15. A comparison of cubic parameterizations for variations in material density, ρ, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

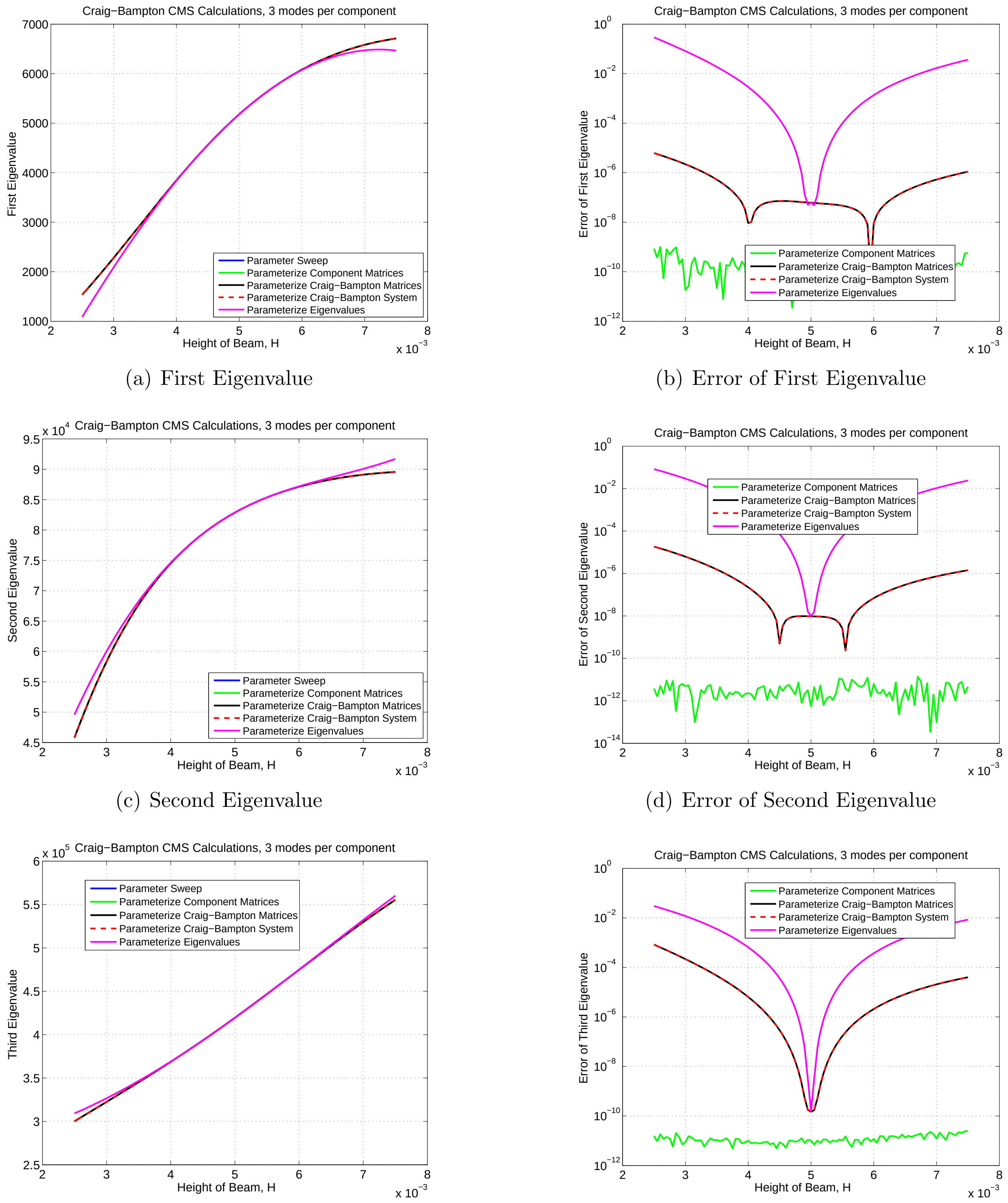
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.16. A comparison of cubic parameterizations for variations in cross-sectional width, B, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

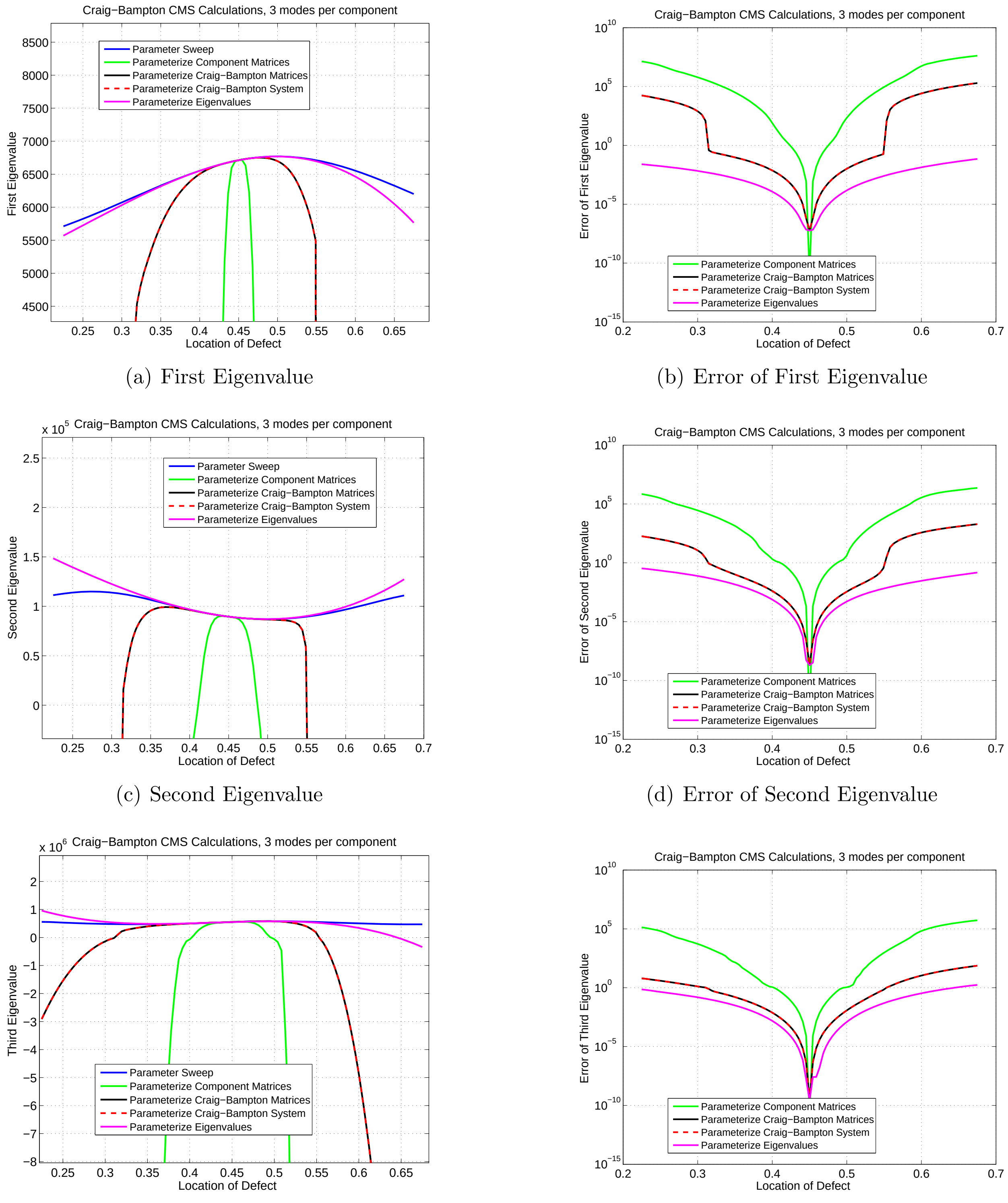
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.17. A comparison of cubic parameterizations for variations in cross-sectional height, H, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

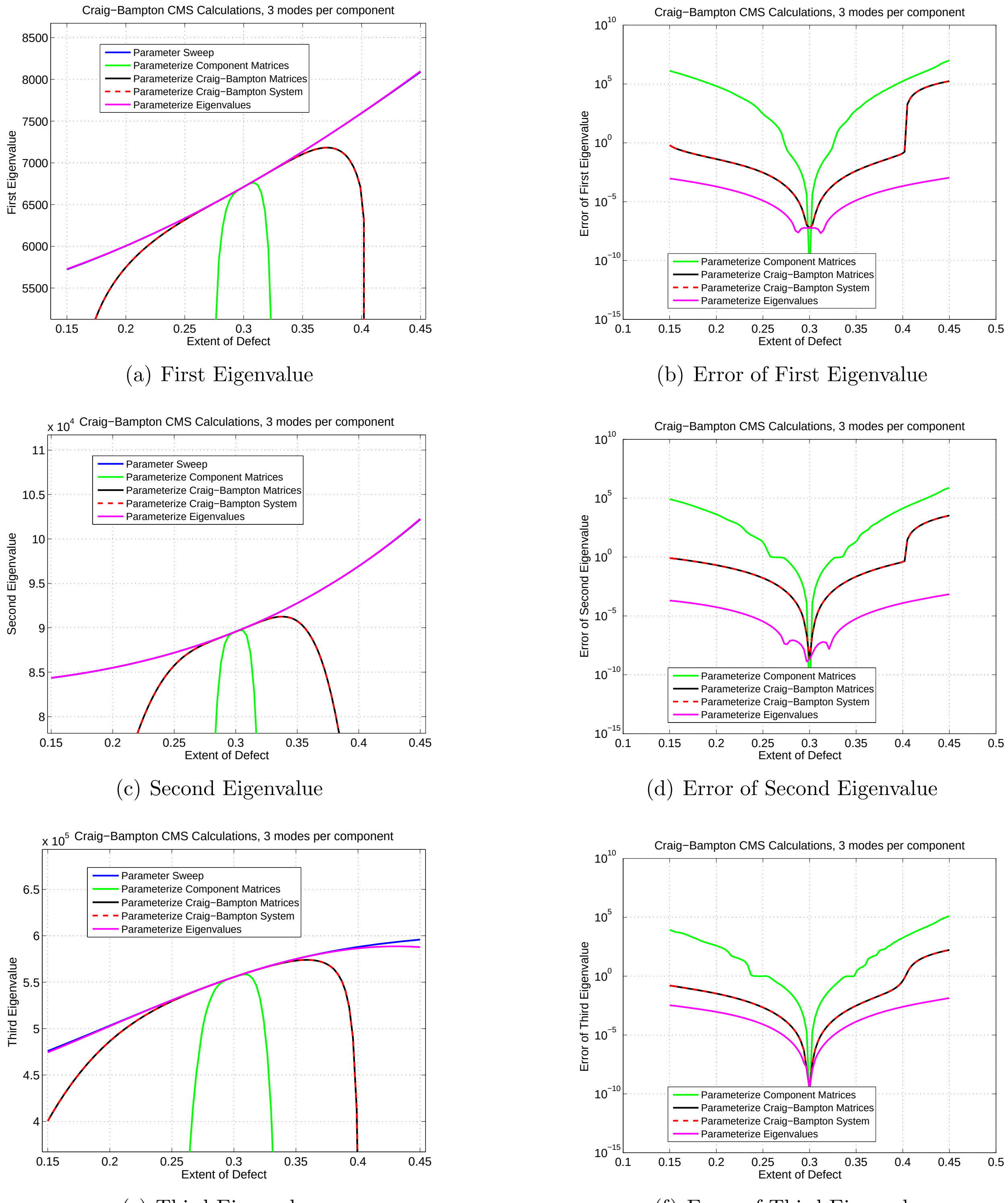
(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.18. A comparison of cubic parameterizations for variations in location of the center section, for the CMS branch of Figure 1.2 where three modes are kept per component.



(a) First Eigenvalue

(b) Error of First Eigenvalue

(c) Second Eigenvalue

(d) Error of Second Eigenvalue

(e) Third Eigenvalue

(f) Error of Third Eigenvalue

Figure 4.19. A comparison of cubic parameterizations for variations in length of the center section, for the CMS branch of Figure 1.2 where three modes are kept per component.

Chapter 5

Conclusions and Future Work

This work demonstrates that hyper-dual numbers can be used to construct PROMs both efficiently (in terms of the number of meshes required) and accurately. With these PROMs, the task of assessing the response of a real system with parametric (aleatoric) uncertainty due to manufacturing or other sources should now be feasible for the first time. Without these PROMs, the effort required to analyze or model aleatoric uncertainty in a real system would be prohibitively expensive. The derivatives necessary for constructing the PROMs are computed using only the nominal design, eliminating the need to construct more than one mesh as would be required by finite differencing. However, in order for this approach to be applied it is necessary to know how the mesh would change as the parameters of the design are varied. One approach would be to incorporate hyper-dual numbers into the mesh generation routine. This has been done for fairly simple 2D computational fluid dynamics meshes; however, this is not likely to be practical for large commercial codes, but it should be feasible for research codes such as SIERRA. More work therefore needs to be done to determine if hyper-dual numbers can be used for large scale finite element calculations.

Higher-order parameterizations should better capture the behavior for the geometric variations. This would require hyper-dual numbers capable of producing exact fourth (or higher) derivatives. It is fairly straightforward to extend the existing hyper-dual number formulation to higher derivatives, as evidenced by the extension to third derivatives. A more general approach, though, in which dual numbers are defined recursively would allow for easier extensions to arbitrary derivatives. This approach would define a hyper-dual number as a dual number with dual number components. Third derivatives would then be produced by going one level further. This approach should be fairly straightforward to implement, and has been done by others. The complication in the present example is determining how to handle the differentiation of the eigenvalue calculations and the linear system solve.

References

[1] Hong, S.-K., Epureanu, B. I., and Castanier, M. P., 2012. "Joining of components of complex structures for improved dynamic response". *Journal of Sound and Vibration,* ***331***(19), pp. 4285–4298.

[2] Hong, S.-K., Epureanu, B. I., and Castanier, M. P., 2013. "Next-generation parametric reduced-order models". *Mechanical Systems and Signal Processing,* ***37***(1-2), pp. 403–421.

[3] Kammer, D. C., and Nimityongskul, S., 2011. "Propagation of uncertainty in test-analysis correlation of substructured spacecraft". *Journal of Sound and Vibration,* ***330***, pp. 1211–1224.

[4] Kammer, D. C., and Krattiger, D., 2013. "Propagation of uncertainty in substructured spacecraft using frequency response". *AIAA Journal,* ***51***(2), pp. 353–361.

[5] Craig, R. R., and Bampton, M. C. C., 1968. "Coupling of substructures for dynamic analyses". *AIAA Journal,* ***6***(7), pp. 1313–1319.

[6] Fike, J. A., and Alonso, J. J., 2011. "The development of hyper-dual numbers for exact second-derivative calculations". In AIAA paper 2011-886, 49th AIAA Aerospace Sciences Meeting.

[7] Edwards, H. C., 2002. *Sierra Framework Version 3: Core Services Theory and Design.* SAND2002-3616. Sandia National Laboratories, Albuquerque, NM.

[8] Fike, J. A., Jongsma, S., Alonso, J. J., and van der Weide, E., 2011. "Optimization with gradient and Hessian information calculated using hyper-dual numbers". In AIAA paper 2011-3807, 29th AIAA Applied Aerodynamics Conference.

[9] Alvin, K. F., 1997. "Efficient computation of eigenvector sensitivities for structural dynamics". *AIAA Journal,* ***35***(11), pp. 1760–1766.

[10] Nelson, R. B., 1976. "Simplified calculation of eigenvector derivatives". *AIAA Journal,* ***14***(9), pp. 1201–1205.

[11] Fox, R. L., and Kapoor, M. P., 1968. "Rates of change of eigenvalues and eigenvectors". *AIAA Journal,* ***6***(12), pp. 2426–2429.

DISTRIBUTION:

1 Jeffrey A. Fike
Aerospace Design Lab
Stanford University
Stanford, CA 94305

1	MS 0346	Richard V. Field, Jr, 1526 (electronic)
1	MS 0346	Nolan McPeek-Bechtold, 1526 (electronic)
1	MS 0346	Diane E. Peebles, 1526 (electronic)
1	MS 0346	Ryan Schultz, 1523 (electronic)
1	MS 0346	D. Gregory Tipton, 1523 (electronic)
1	MS 0346	Douglas J. VanGoethem, 1526 (electronic)
1	MS 0557	C. Dennis Croessmann, 1520 (electronic)
1	MS 0557	Todd W. Simmermacher, 1523 (electronic)
1	MS 0678	Andre A. Claudet, 2617 (electronic)
1	MS 0747	Remi P. M. Dingreville, 6233 (electronic)
1	MS 0815	Justine E. Johannes, 1500 (electronic)
1	MS 0828	John R. Red-Horse, 1544 (electronic)
1	MS 0828	Michael Ross, 1523 (electronic)
1	MS 0840	H. Eliot Fang, 1524 (electronic)
1	MS 0845	Clark R. Dohrmann, 1542 (electronic)
1	MS 0845	Joseph Jung, 1542 (electronic)
1	MS 0845	Garth M. Reese, 1542 (electronic)
1	MS 1070	Matthew R. Brake, 1526 (electronic)
1	MS 1070	Jordan E. Massad, 1526 (electronic)
1	MS 9042	Daniel J. Segalman, 8259 (electronic)
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories