UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

Advanced Model Reduction Techniques for Structural Dynamics Simulations

Filippos Katsimalis

Supervisor: **Dr. Costas Papadimitriou**

Submitted in partial fulfillment of the requirements for the degree of Diploma

in Mechanical Engineering at the University of Thessaly

Volos, October 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Προηγμένες Τεχνικές Μείωσης Μοντέλων σε Προσομοιώσεις Δυναμικής των Κατασκευών

Φίλιππος Κατσιμαλής

Επιβλέπων: **Δρ. Κώστας Παπαδημητρίου**

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για

την απόκτηση του Διπλώματος Μηχανολόγου Μηχανικού

Βόλος, Οκτώβριος 2021

The approval of the Diploma Thesis by the Department of Mechanical Engineering of the University of Thessaly does not imply acceptance of the author’s opinions. (Law 5343/32, article 202, paragraph 2).

Approved by the Committee on Final Examination:

Advisor **Dr. Costas Papadimitriou**

Professor of Structural Dynamics, Department of Mechanical Engineering, University of Thessaly

Member **Dr. Konstantinos Ampountolas**

Associate Professor of Automatic Control and Optimization Theory, Department of Mechanical Engineering, University of Thessaly

Member **Dr. Spyros A. Karamanos**

Professor of Structural Mechanics, Department of Mechanical Engineering, University of Thessaly

Acknowledgements

I would like to especially thank my mother and father for supporting me all these years throughout my studies. It would not have been possible to pursue engineering without their continuous support and encouragement.

I express my sincere gratitude to my supervisor Dr. Costas Papadimitriou who patiently assisted me with every problem I encountered working on this thesis and - with his lectures - shaped my interest in the area of structural dynamics.

I would also like to thank the members of the committee Dr. Konstantinos Ampountolas and Dr. Spyros A. Karamanos for their time and valuable feedback on this work.

I am grateful to my colleague and friend Georgios Trousas for giving me precious advice in the early stages of writing and answering my many questions.

Finally, I must thank my friends Dimitris Petrolekas, Dimitris Sdrolias and Dimitris Tsogias who stood by me in times of difficulty from the very beginning. They helped me become a better version of myself and for that, I owe them a lot.

Advanced Model Reduction Techniques for Structural Dynamics Simulations

Filippos Katsimalis

Department of Mechanical Engineering, University of Thessaly

Supervisor: **Dr. Costas Papadimitriou**

Professor of Structural Dynamics, Department of Mechanical Engineering, University of Thessaly

Abstract

Component mode synthesis (CMS) is a well-known model reduction method usually applied on large and complex finite element models of hundreds of thousands or even million degrees of freedom. The resulting reduced-order model maintains the dynamic behavior of the original but requires significantly less time and resources to be analyzed. This thesis presents the classic Craig-Bampton CMS technique and two additional methods based on it that consider interface reduction at the global and local level. Standard CMS methods cannot be used directly in structural dynamics simulations due to the repetitive generation of the reduced-order model which can be computationally expensive. In this work, three parametrized CMS methods are introduced along with an efficient parametrization scheme and an interpolation method for approximating interface modes. A novel method for generating support points used in the interpolation scheme of interface modes is also presented. If the structure is parametrized using the proposed scheme, dramatic computational savings can be achieved since the parametrized CMS methods presented here do not require the re-assembling of the reduced-order system matrices during the simulation process. The efficiency and accuracy of the non-parameterized methods is examined using a high-fidelity finite element model of a highway bridge consisting of nearly one million degrees of freedom. The finite element model is constructed using COMSOL Multiphysics and all CMS methods are applied using MATLAB code originally developed for the purpose of this thesis.

Keywords: Model reduction, Component mode synthesis, Structural dynamics, Finite elements, Eigen-analysis, Substructuring, Craig-Bampton method, Interface reduction, Parametrization, Parametrized reduced order models, MATLAB, COMSOL Multiphysics

Προηγμένες Τεχνικές Μείωσης Μοντέλων σε Προσομοιώσεις Δυναμικής των Κατασκευών

Φίλιππος Κατσιμαλής

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Επιβλέπων: **Δρ. Κώστας Παπαδημητρίου**

Καθηγητής Δυναμικής των Κατασκευών, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Περίληψη

Η μέθοδος σύνθεσης ιδιομορφών (Component mode synthesis: CMS) είναι μια ευρέως διαδεδομένη τεχνική μείωσης μοντέλων που συνήθως εφαρμόζεται σε μεγάλα και σύνθετα μοντέλα πεπερασμένων στοιχείων εκατοντάδων χιλιάδων ή ακόμη και εκατομμυρίων βαθμών ελευθερίας. Το μοντέλο μειωμένης τάξης που προκύπτει διατηρεί τη δυναμική συμπεριφορά του αρχικού αλλά απαιτείται σημαντικά λιγότερος χρόνος και πόροι για την ανάλυσή του. Αυτή η εργασία παρουσιάζει την κλασική τεχνική Craig-Bampton και δύο επιπλέον μεθόδους που βασίζονται σε αυτήν οι οποίες πραγματοποιούν μείωση των συνόρων μεταξύ υπό-κατασκευών σε ολικό αλλά και τοπικό επίπεδο. Οι τυπικές μέθοδοι CMS δεν μπορούν να χρησιμοποιηθούν απευθείας σε προσομοιώσεις δυναμικής των κατασκευών λόγω της ανάγκης για επαναλαμβανόμενη δημιουργία του μοντέλου μειωμένης τάξης που μπορεί να είναι υπολογιστικά δαπανηρή. Σε αυτήν την εργασία, εισάγονται τρεις παραμετροποιημένες μέθοδοι CMS μαζί με ένα αποτελεσματικό σχήμα παραμετροποίησης της κατασκευής και μία μέθοδος παρεμβολής για την προσέγγιση των ιδιομορφών στα σύνορα μεταξύ υπό-κατασκευών. Παρουσιάζεται επίσης μια νέα μέθοδος για τη δημιουργία σημείων παρεμβολής που χρησιμοποιούνται στο προτεινόμενο σχήμα παρεμβολής των συνοριακών ιδιομορφών. Εάν η αρχική κατασκευή παραμετροποιηθεί χρησιμοποιώντας το προτεινόμενο σχήμα, μπορεί να επιτευχθεί μεγάλη μείωση του υπολογιστικού φόρτου αφού οι παραμετροποιημένες μέθοδοι CMS που παρουσιάζονται εδώ δεν απαιτούν την επαναλαμβανόμενη δημιουργία των μητρώων μειωμένης τάξης κατά τη διάρκεια προσομοίωσης της κατασκευής. Η αποτελεσματικότητα και η ακρίβεια των τυπικών (μη-παραμετροποιημένων) μεθόδων εξετάζονται χρησιμοποιώντας ένα αρκετά λεπτομερές μοντέλο πεπερασμένων στοιχείων μιας γέφυρας που αποτελείται από σχεδόν ένα εκατομμύριο βαθμούς ελευθερίας. Το μοντέλο πεπερασμένων στοιχείων αναπτύσσεται με τη χρήση του πακέτου COMSOL Multiphysics και όλες οι μέθοδοι CMS εφαρμόζονται χρησιμοποιώντας κώδικα MATLAB που αναπτύχθηκε ειδικά για τους σκοπούς της διπλωματικής εργασίας.

Λέξεις κλειδιά: Μείωση μοντέλου, Σύνθεση ιδιομορφών, Δυναμική των κατασκευών, Πεπερασμένα στοιχεία, Ιδιομορφική ανάλυση, Διαίρεση σε υπό-κατασκευές, Μέθοδος Craig-Bampton, Μείωση συνοριακών βαθμών ελευθερίας, Παραμετροποίηση, Παραμετροποιημένα μοντέλα μειωμένης τάξης, MATLAB, COMSOL Multiphysics

Table of Contents

List of Figures

List of Tables

Abbreviations

FE: Finite element

DOF: Degree(s) of freedom

CMS: Component mode synthesis

1 Introduction

The finite element (FE) method is widely used today in virtually all fields of engineering. It has proven especially useful in the analysis of structures, fluids, and solids. The capabilities and usefulness of the method is closely related to the computational power of the digital computer [1]. Today's computer hardware allows for large and complex engineering problems to be solved relatively quickly using commercial programs that utilize the FE method.

Such problems require a FE model to be developed which is characterized by – among others – the number of finite elements and degrees of freedom. The number of degrees of freedom of a FE model greatly impacts the computational demands and time needed to analyze it. For many modern, high-fidelity models with hundreds of thousands or millions of degrees of freedom, the direct implementation of the FE method is impractical.

When the model is so large or complex, that it is inefficient to apply the FE method directly on it, some form of model reduction must be employed [2]. With model reduction (or model order reduction), the number of degrees of freedom of the original FE model is greatly reduced and a new, reduced-order model is used in its place. The reduced-order model requires less computational resources while it retains the dynamic characteristics of the original model.

Component mode synthesis (CMS), also referred to as dynamic substructuring, is a very popular method of model reduction for large structural dynamics problems. It involves partitioning of the entire structure into several simpler substructures or components, obtaining reduced-order models of the substructures and then assembling a reduced-order model of the entire structure [3]. The essential idea is to derive the behavior of the entire assembly from its constituents [2].

Apart from the case of large or complex models, some occasions where the use of CMS for model reduction presents an attractive possibility are:

- In a structural dynamics simulation where a large number of dynamic re-analyses is required. Such problems are Bayesian uncertainty quantification, model updating, reliability analysis and so on. Typically, the time to perform a single analysis is large (in view of the large number of re-analyses necessary) and the computational effort can be excessive [3]. Using CMS greatly reduces the time of a single analysis, and consequently, the time-to-solution.
- In a design situation, where it is "natural" for different parts of a model to be designed by different teams independently [4]. For example, the different components of an aircraft are designed by different groups which – using CMS – could work only on their relevant part (wings, fuselage etc.) of the original structure (the aircraft).
- In a situation where parallel processing capabilities exist [4]. In such a setting, the substructures of the reduced-order model could be analyzed in parallel and great time savings could be made.
- In case experimentally obtained modal data are available for some components, CMS gives the possibility of combining modeled components with experimentally identified ones [5].

This work focuses on a specific CMS technique: the Craig-Bampton (or Hurty/Craig-Bampton) approach [6]. It is one of the most popular CMS techniques in industry and academia [7] and mathematically justified as the most “natural” CMS method [4][8]. It makes use of substructure eigenproperties (component modes and eigenvalues) to “capture” the dynamic behavior of each component and subsequently of the original structure.

The method is a modification of Hurty’s approach [9], one of the first CMS methods introduced in the mid-1960s. Hurty’s method uses three types of component modes: rigid body modes, constraint modes, and fixed-interface normal modes. Craig and Bampton essentially simplified Hurty’s method by showing that it was not necessary to separately consider the rigid body modes [4]. Other classic CMS methods resulting from Hurty’s work are that of Rubin [10] and MacNeal [11] in the 1970s.

Although capable of significant reduction in the number of a component’s internal DOF, the Craig-Bampton method does not consider reduction at the interface DOF. The number of interface DOF in the original FE model is determined by the FE mesh. If the mesh is fine in the interface regions, or if there are many substructures, the reduced-order model may be relatively large [3]. In this case, reduction of interface DOF – in addition to internal DOF – can be very appealing.

In this thesis, two techniques for reduction of interface DOF are presented: a method for global-level (or system-level) reduction proposed by Castanier et al. [12] and a method for local-level reduction based on the work of Hong et al. [13].

The global-level method of interface reduction treats all interface DOF together and the interface definition plays no role (hence it is called global-level reduction). An eigenvalue analysis is performed on the interface partition of the reduced-order matrices to obtain the characteristic constraint modes. The downside of such methods is that they generally require more support points – compared to local-level methods – to interpolate the interface modes in a structural dynamics simulation. This is because the (global) interface necessary depends on all model parameters.

Conversely, the local-level interface reduction method treats each interface independently. An eigen-analysis is performed on the partition of the reduced-order matrices corresponding to the DOF of each interface to obtain the local interface modes. The basic benefit of such methods is that – depending on the interface definition – few support points are needed to approximate the interface modes during a simulation. The reason is that every (local) interface usually depends on few (not all) model parameters.

All the above CMS methods produce non-parameterized reduced-order matrices that must be reassembled at every sample point of a simulation-based problem. To avoid this computationally intensive step, three advanced techniques of model reduction are presented here which give parameterized matrices that can be used efficiently in a structural dynamics simulation.

The first parametrized CMS method is based on the classic Craig-Bampton technique and does not consider reduction of interface DOF. The other two are based on the global- and local-level methods of interface reduction already mentioned. These methods make use of an interpolation scheme proposed by Goller et al. [14] for the approximation of interface modes to avoid a direct interface analysis during the simulation process.

The parametrized CMS methods makes use of the parametrization scheme proposed by Jensen and Papadimitriou [3] where the mass and stiffness matrix of a component depend only on one (or none) model parameter. This type of parametrization is often encountered in structural systems modeled by standard finite elements.

The organization of the thesis is as follows. Chapter 2 introduces some basic theory concerning the classic Craig-Bampton CMS method and an improved formulation that takes into account the residual normal modes of each component proposed by Jensen et al. [15]. In Chapter 3 the CMS methods that consider reduction of interface DOF are presented. Theory for parametrization of the classic Craig-Bampton method is presented in Chapter 4. In Chapter 5 the parametrization of the two methods that perform interface reduction is introduced. In Chapter 6 the accuracy and efficiency of all non-parameterized CMS methods is examined using a large FE model of nearly one million DOF. The conclusions and suggestions for further study are presented in Chapter 7. Finally, the Appendix contains a presentation of the MATLAB function *input.m* that controls most aspects of model reduction and a link to the GitHub repository that hosts the MATLAB code developed for applying all formulations presented here.

2 Basic Theory of the Craig-Bampton Method of Component Mode Synthesis

Among the most commonly used techniques for CMS is that developed by Craig and Bampton [2][6]. The Craig-Bampton method, which is used in the present work, is widely adopted because of its superior accuracy, its ease of implementation, and its efficient use of computer resources.

This chapter aims to introduce necessary theory of the Craig-Bampton method of component mode synthesis. The chapter is based on the formulations presented in chapter 1 of [3].

2.1 Structural Model

The models considered are structural dynamical systems with localized non-linearities that satisfy the equation of motion

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}_{NL}(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{y}(t)) + \mathbf{f}(t) \tag{2.1}$$

where $\mathbf{u}(t) \in \mathbb{R}^n$ denotes the displacement vector, $\dot{\mathbf{u}}(t)$ the velocity vector, $\ddot{\mathbf{u}}(t)$ the acceleration vector, $\mathbf{f}_{NL}(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{y}(t))$ the vector of nonlinear restoring forces, $\mathbf{y}(t)$ the vector of a set of variables that describes the state of the nonlinear components, and $\mathbf{f}(t)$ the external force vector. The matrices $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$, which are assumed to be symmetric, describe the mass, damping, and stiffness, respectively. The evolution of the set of variables $\mathbf{y}(t)$ is described through an appropriate nonlinear model that depends on the nature of the nonlinearity [3].

2.2 Substructure Modes

The term substructure modes is used to signify Ritz vectors, or assumed modes, that are used as basis vectors in describing the displacement of points within a substructure, or component [2].

Most applications of component-mode synthesis employ one of two approaches, which may be called fixed-interface-mode methods and free-interface-mode methods. The former employ fixed-interface normal modes and constraint modes. The latter employ free interface normal modes and attachment modes [2].

The Craig-Bampton method is a fixed interface technique. In this manner, the dynamic behavior of the linear components of the structural system is described by a set of normal modes (eigenvectors) of individual substructures along with a set of constraint modes that account for the coupling at each interface where the substructures are connected [3].

2.2.1 Fixed-Interface Normal Modes

Consider a linear substructure $s, s = 1,\dots,N_s$ having n_s degrees of freedom. The mass $\mathbf{M}^s \in \mathbb{R}^{n^s \times n^s}$ and stiffness $\mathbf{K}^s \in \mathbb{R}^{n^s \times n^s}$ matrix of the substructure are partitioned as follows[3]

$$\mathbf{M}^s = \begin{bmatrix} \mathbf{M}_{ii}^s & \mathbf{M}_{ib}^s \\ \mathbf{M}_{bi}^s & \mathbf{M}_{bb}^s \end{bmatrix} \tag{2.2}$$

$$\mathbf{K}^s = \begin{bmatrix} \mathbf{K}_{ii}^s & \mathbf{K}_{ib}^s \\ \mathbf{K}_{bi}^s & \mathbf{K}_{bb}^s \end{bmatrix} \tag{2.3}$$

where N_s is the total number of linear substructures, and the indices i and b are sets containing the internal and boundary DOF, respectively, of substructure s.

The internal degrees of freedom, which are not shared with any adjacent substructures, are kept in the vector $\mathbf{u}_i^s(t) \in \mathbb{R}^{n_i^s}$, while all boundary degrees of freedom are kept in the vector $\mathbf{u}_b^s(t) \in \mathbb{R}^{n_b^s}$. The boundary degrees of freedom include only those that are in common with the interface degrees of freedom of adjacent substructures. Note that the number of internal and boundary DOF sum up to the number of total DOF of the substructure $n^s = n_i^s + n_b^s$.

Component fixed-interface normal modes are obtained by restraining all boundary DOF and solving the following eigenproblem

$$\mathbf{K}_{ii}^s \boldsymbol{\Phi}_{ii}^s - \mathbf{M}_{ii}^s \boldsymbol{\Phi}_{ii}^s \boldsymbol{\Lambda}_{ii}^s = \mathbf{0}, \quad s = 1,\dots,N_s \tag{2.4}$$

where the matrix $\boldsymbol{\Phi}_{ii}^s$ contains the complete set of n_i^s fixed-interface normal modes in its columns and $\boldsymbol{\Lambda}_{ii}^s$ is the corresponding diagonal matrix containing the eigenvalues (squares of the natural frequencies). The fixed interface normal modes are normalized with respect to the mass matrix $\mathbf{M}_{ii}^s$, that is

$$\boldsymbol{\Phi}_{ii}^{sT} \mathbf{M}_{ii}^s \boldsymbol{\Phi}_{ii}^s = \mathbf{I}_{ii}^s \tag{2.5}$$

and

$$\boldsymbol{\Phi}_{ii}^{sT} \mathbf{K}_{ii}^s \boldsymbol{\Phi}_{ii}^s = \boldsymbol{\Lambda}_{ii}^s \tag{2.6}$$

where $\mathbf{I}_{ii}^{s} \in \mathbb{R}^{n_i^s \times n_i^s}$ is the identity matrix.

2.2.2 Interface Constraint Modes

The interface constraint modes are defined as the static deformation of the substructure when a unit displacement is applied at one coordinate of vector $\mathbf{u}_b^s(t)$ and zero displacement at the remaining interface degrees of freedom, while the internal degrees of freedom are force free [3][2]. Then, the interface constraint modes matrix is

$$\boldsymbol{\Psi}^s = \begin{bmatrix} \boldsymbol{\Psi}_{ib}^s \\ \mathbf{I}_{bb}^s \end{bmatrix} \in \mathbb{R}^{n^s \times n_b^s} \tag{2.7}$$

where $\boldsymbol{\Psi}_{ib}^s \in \mathbb{R}^{n_i^s \times n_b^s}$ is the interior partition of the interface constrained modes matrix and $\mathbf{I}_{bb}^s \in \mathbb{R}^{n_b^s \times n_b^s}$ is the identity matrix. The interface constrained modes matrix satisfy

$$\begin{bmatrix} \mathbf{K}_{ii}^s & \mathbf{K}_{ib}^s \\ \mathbf{K}_{bi}^s & \mathbf{K}_{bb}^s \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_{ib}^s \\ \mathbf{I}_{bb}^s \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{ib}^s \\ \mathbf{R}_{bb}^s \end{bmatrix} \tag{2.8}$$

where $\mathbf{0}_{ib}^s \in \mathbb{R}^{n_i^s \times n_b^s}$ is the null matrix, and $\mathbf{R}_{bb}^s \in \mathbb{R}^{n_b^s \times n_b^s}$ is the corresponding matrix of interface forces. The interior partition of the interface constrained modes matrix is calculated by solving the first block of Eq. (2.8)

$$\boldsymbol{\Psi}_{ib}^s = -\mathbf{K}_{ii}^{s^{-1}} \mathbf{K}_{ib}^s \tag{2.9}$$

as a result, the interface constraint modes matrix takes the form

$$\boldsymbol{\Psi}^s = \begin{bmatrix} \boldsymbol{\Psi}_{ib}^s \\ \mathbf{I}_{bb}^s \end{bmatrix} = \begin{bmatrix} -\mathbf{K}_{ii}^{s^{-1}} \mathbf{K}_{ib}^s \\ \mathbf{I}_{bb}^s \end{bmatrix} \tag{2.10}$$

2.3 Reduced-order model: Standard formulation

In the standard formulation of the Craig-Bampton method, the reduced-order model includes a fraction of the fixed-interface modal coordinates of each substructure and the physical interface coordinates. The effect of the residual fixed-interface modal coordinates is neglected in the analysis. This section aims to present the derivation of the corresponding reduced-order model.

2.3.1 Transformation Matrix

The displacement transformation of the Craig-Bampton method is performed through the matrix $\mathbf{T}_D$ (Craig-Bampton transformation matrix) which utilizes a combination of fixed-interface normal modes and interface constraint modes. The transformation matrix $\mathbf{T}_D$ relates the vector of physical coordinates of all substructures $\bar{\mathbf{u}}(t)$ to the vector of generalized coordinates $\mathbf{q}(t)$ as

$$\bar{\mathbf{u}}(t) = \mathbf{T}_D \mathbf{q}(t) \tag{2.11}$$

where

$$\bar{\mathbf{u}}(t) = \begin{Bmatrix} \mathbf{u}_i(t) \\ \mathbf{u}_I(t) \end{Bmatrix} \in \mathbb{R}^n \tag{2.12}$$

in which

$$\mathbf{u}_i(t) = \begin{Bmatrix} \mathbf{u}_i^1(t) \\ \vdots \\ \mathbf{u}_i^{N_s}(t) \end{Bmatrix} \in \mathbb{R}^{n_i}, \quad n_i = \sum_{s=1}^{N_s} n_i^s \tag{2.13}$$

is the vector of physical coordinates at the internal DOF of all substructures,

$$\mathbf{u}_I(t) = \begin{Bmatrix} \mathbf{u}_I^1(t) \\ \vdots \\ \mathbf{u}_I^{N_I}(t) \end{Bmatrix} \in \mathbb{R}^{n_I}, \quad n_I = \sum_{l=1}^{N_I} n_I^l \tag{2.14}$$

is the vector of physical coordinates at the N_I independent interfaces, where n_I^l is the number of DOF at the interface l,

$$\mathbf{T}_D = \begin{bmatrix} \left[\boldsymbol{\Phi}_{id}^1, \ldots, \boldsymbol{\Phi}_{id}^{N_s}\right] & \left[\boldsymbol{\Psi}_{ib}^1, \ldots, \boldsymbol{\Psi}_{ib}^{N_s}\right] \tilde{\mathbf{T}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{n \times n_D} \tag{2.15}$$

is the Craig-Bampton transformation matrix, where $\left[\cdot, \ldots, \cdot\right]$ denotes a block diagonal matrix having the matrices inside the square brackets as diagonal blocks, $\boldsymbol{\Phi}_{id}^s \in \mathbb{R}^{n_i^s \times n_{id}^s}$ is the matrix containing the n_{id}^s kept fixed-interface normal modes of substructure s. Note that due to the truncation of the complete set of eigenvectors $n_{id}^s \ll n_i^s$. The number of columns of $\mathbf{T}_D$ is given by $n_D = n_{id} + n_I$ in which $n_{id} = \sum_{s=1}^{N_s} n_{id}^s$. Also, $\mathbf{0} \in \mathbb{R}^{n_I \times n_{id}}$ is the null matrix, $\mathbf{I} \in \mathbb{R}^{n_I \times n_I}$ is the identity matrix and $\tilde{\mathbf{T}} \in \mathbb{R}^{n_b \times n_I}$ is a transformation matrix consisting of zeros and ones that maps the vector $\mathbf{u}_I(t)$ of independent interface coordinates to the vector of boundary coordinates of all substructures $\mathbf{u}_b(t)$, that is

$$\mathbf{u}_b(t) = \tilde{\mathbf{T}} \mathbf{u}_I(t) \tag{2.16}$$

where

$$\mathbf{u}_b(t) = \begin{Bmatrix} \mathbf{u}_b^1(t) \\ \vdots \\ \mathbf{u}_b^{N_s}(t) \end{Bmatrix} \in \mathbb{R}^{n_b}, \quad n_b = \sum_{s=1}^{N_s} n_b^s \tag{2.17}$$

and where the vector of generalized coordinates is defined as

$$\mathbf{q}(t) = \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \mathbf{u}_I(t) \end{Bmatrix} \in \mathbb{R}^{n_D} \tag{2.18}$$

where $\boldsymbol{\eta}(t)$ is the vector of kept fixed-interface modal coordinates of all substructures.

When applying the method using a FE mesh, the vector of boundary coordinates of all substructures $\mathbf{u}_b(t)$ can contain repeated entries of nodes which belong to the interface between two or more substructures.

On the other hand, the vector $\mathbf{u}_I(t)$ of independent interface coordinates does not contain repeated entries of nodes since each node is associated with a single independent interface. The particular structure of the transformation matrix $\tilde{\mathbf{T}}$ depends on the definition of the independent interface coordinates $\mathbf{u}_I^l(t), \quad l = 1,\ldots,N_I$.

The kept fixed-interface normal modes of each substructure $\boldsymbol{\Phi}_{id}^s$ are referred to as dominant fixed-interface normal modes.

2.3.2 Reduced-Order Matrices

Given the above formulations, the mass and stiffness matrices of the model referred to the vector $\overline{\mathbf{u}}(t)$ are given by [3]

$$\hat{\mathbf{M}} = \begin{bmatrix} \left[\mathbf{M}_{ii}^1,\ldots,\mathbf{M}_{ii}^{N_s}\right] & \left[\mathbf{M}_{ib}^1,\ldots,\mathbf{M}_{ib}^{N_s}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^T\left[\mathbf{M}_{ib}^{1^T},\ldots,\mathbf{M}_{ib}^{N_s^T}\right] & \tilde{\mathbf{T}}^T\left[\mathbf{M}_{bb}^1,\ldots,\mathbf{M}_{bb}^{N_s}\right]\tilde{\mathbf{T}} \end{bmatrix} \tag{2.19}$$

and

$$\hat{\mathbf{K}} = \begin{bmatrix} \left[\mathbf{K}_{ii}^1,\ldots,\mathbf{K}_{ii}^{N_s}\right] & \left[\mathbf{K}_{ib}^1,\ldots,\mathbf{K}_{ib}^{N_s}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^T\left[\mathbf{K}_{ib}^{1^T},\ldots,\mathbf{K}_{ib}^{N_s^T}\right] & \tilde{\mathbf{T}}^T\left[\mathbf{K}_{bb}^1,\ldots,\mathbf{K}_{bb}^{N_s}\right]\tilde{\mathbf{T}} \end{bmatrix} \tag{2.20}$$

The corresponding mass and stiffness matrices of the model referred to the generalized coordinates $\mathbf{q}(t)$ take the form

$$\hat{\mathbf{M}}_D = \mathbf{T}_D^T\hat{\mathbf{M}}\mathbf{T}_D \tag{2.21}$$

and

$$\hat{\mathbf{K}}_D = \mathbf{T}_D^T\hat{\mathbf{K}}\mathbf{T}_D \tag{2.22}$$

Executing the previous products yields

$$\hat{\mathbf{M}}_D = \begin{bmatrix} \mathbf{I} & \left[\hat{\mathbf{M}}_{ib}^1,\ldots,\hat{\mathbf{M}}_{ib}^{N_s}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^T\left[\hat{\mathbf{M}}_{ib}^{1^T},\ldots,\hat{\mathbf{M}}_{ib}^{N_s^T}\right] & \tilde{\mathbf{T}}^T\left[\hat{\mathbf{M}}_{bb}^1,\ldots,\hat{\mathbf{M}}_{bb}^{N_s}\right]\tilde{\mathbf{T}} \end{bmatrix} \in \mathbb{R}^{n_D \times n_D} \tag{2.23}$$

and

$$\hat{\mathbf{K}}_D = \begin{bmatrix} \left[\boldsymbol{\Lambda}_{id}^1,\ldots,\boldsymbol{\Lambda}_{id}^{N_s}\right] & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{T}}^T\left[\hat{\mathbf{K}}_{bb}^1,\ldots,\hat{\mathbf{K}}_{bb}^{N_s}\right]\tilde{\mathbf{T}} \end{bmatrix} \in \mathbb{R}^{n_D \times n_D} \tag{2.24}$$

with

$$\hat{\mathbf{M}}_{ib}^s = \boldsymbol{\Phi}_{id}^{s^T}\mathbf{M}_{ii}^s\boldsymbol{\Psi}_{ib}^s + \boldsymbol{\Phi}_{id}^{s^T}\mathbf{M}_{ib}^s \tag{2.25}$$

$$\hat{\mathbf{K}}_{bb}^s = \mathbf{K}_{ib}^{s^T}\boldsymbol{\Psi}_{ib}^s + \mathbf{K}_{bb}^s = -\mathbf{K}_{ib}^{s^T}\mathbf{K}_{ii}^{s^{-1}}\mathbf{K}_{ib}^s + \mathbf{K}_{bb}^s \tag{2.26}$$

and

$$\hat{\mathbf{M}}_{bb}^s = \left(\boldsymbol{\Psi}_{ib}^{s^T}\mathbf{M}_{ii}^s + \mathbf{M}_{ib}^{s^T}\right)\boldsymbol{\Psi}_{ib}^s + \boldsymbol{\Psi}_{ib}^{s^T}\mathbf{M}_{ib}^s + \mathbf{M}_{bb}^s\,, \quad s = 1,\ldots,N_s \tag{2.27}$$

where $\mathbf{I} \in \mathbb{R}^{n_{id} \times n_{id}}$ is the identity matrix and $\boldsymbol{\Lambda}_{id}^s, s = 1,\ldots,N_s$ are diagonal matrices containing the eigenvalues of the kept (dominant) fixed-interface normal modes for each substructure.

The essence of the method lies in the fact that the dimension of the reduced-order matrices can be substantially smaller than the dimension of the unreduced matrices, that is, $n_D \ll n$. The reason for this is the great reduction of the number of kept modes of the reduced model compared to the complete set of modes of the unreduced model which results in great computational savings.

2.4 Reduced-order model: Improved formulation

In the previous section the standard reduced order model of the Craig-Bampton method has been derived. According to the standard method, the vector of physical coordinates at the internal degrees of freedom of all substructures $\mathbf{u}_i(t)$ is approximated as

$$\mathbf{u}_i(t) = \left[\boldsymbol{\Phi}_{id}^1,\ldots,\boldsymbol{\Phi}_{id}^{N_s}\right]\boldsymbol{\eta}(t) + \left[\boldsymbol{\Psi}_{ib}^1,\ldots,\boldsymbol{\Psi}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\mathbf{u}_I(t) \tag{2.28}$$

It can be seen that the $n_i - n_{id}$ residual fixed-interface normal modes are not taken into account in the approximation.

In this improved formulation their effect is explicitly considered which results in more accurately constructed reduced-order matrices [3][16]. The derivation of the improved reduced-order model is shown below.

2.4.1 Improved Transformation Matrix

The static contribution of the residual fixed-interface normal modes to the response of the physical coordinates at the internal degrees of freedom of all substructures $\mathbf{u}_i(t)$ is approximated by using static correction [3][16][1] as

$$\mathbf{u}_i(t) = \left[\boldsymbol{\Phi}_{id}^1, \ldots, \boldsymbol{\Phi}_{id}^{N_s}\right]\boldsymbol{\eta}(t) + \left[\boldsymbol{\Psi}_{ib}^1, \ldots, \boldsymbol{\Psi}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\mathbf{u}_I(t) - \bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\ddot{\mathbf{u}}_I(t) \qquad (2.29)$$

where $\bar{\mathbf{F}}$ is a block diagonal matrix containing the residual flexibility matrix of all substructures

$$\bar{\mathbf{F}} = \left[\bar{\mathbf{F}}^1, \ldots, \bar{\mathbf{F}}^{N_s}\right] \qquad (2.30)$$

where for a substructure s the residual flexibility matrix corresponding to the fixed-interface normal modes problem is

$$\bar{\mathbf{F}}^s = \mathbf{K}_{ii}^{s^{-1}} - \boldsymbol{\Phi}_{id}^s \boldsymbol{\Lambda}_{id}^{s^{-1}} \boldsymbol{\Phi}_{id}^{s^T} \qquad (2.31)$$

and

$$\tilde{\mathbf{M}}_{ib} = \left[\tilde{\mathbf{M}}_{ib}^1, \ldots, \tilde{\mathbf{M}}_{ib}^{N_s}\right] \qquad (2.32)$$

with

$$\tilde{\mathbf{M}}_{ib}^s = \mathbf{M}_{ib}^s - \mathbf{M}_{ii}^s \mathbf{K}_{ii}^{s^{-1}} \mathbf{K}_{ib}^s \qquad (2.33)$$

Taking into consideration the improved approximation of $\mathbf{u}_i(t)$ in Eq. (2.29) and the definition of the vector of physical coordinates of all substructures $\bar{\mathbf{u}}(t)$ in Eq. (2.12), it follows that

$$\bar{\mathbf{u}}(t) = \begin{bmatrix} \left[\boldsymbol{\Phi}_{id}^1, \ldots, \boldsymbol{\Phi}_{id}^{N_s}\right] & \left[\boldsymbol{\Psi}_{ib}^1, \ldots, \boldsymbol{\Psi}_{ib}^{N_s}\right]\tilde{\mathbf{T}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \mathbf{u}_I(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{0} & -\bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{\eta}}(t) \\ \ddot{\mathbf{u}}_I(t) \end{Bmatrix} \qquad (2.34)$$

From the equation of motion of the undamped free vibration of the linear components of Eq. (2.1), the relation between the vector of generalized coordinates $\mathbf{q}(t)$ and its second derivative is

$$\begin{Bmatrix} \ddot{\boldsymbol{\eta}}(t) \\ \ddot{\mathbf{u}}_I(t) \end{Bmatrix} = -\hat{\mathbf{M}}_D^{-1}\hat{\mathbf{K}}_D \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \mathbf{u}_I(t) \end{Bmatrix} \qquad (2.35)$$

Using Eqs. (2.15) and (2.35), Eq. (2.34) can be rewritten in the form

$$\bar{\mathbf{u}}(t) = \left\{ \mathbf{T}_D + \mathbf{T}_R \right\} \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \mathbf{u}_I(t) \end{Bmatrix} \tag{2.36}$$

where

$$\mathbf{T}_R = \begin{bmatrix} \mathbf{0} & \bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \hat{\mathbf{M}}_D^{-1}\hat{\mathbf{K}}_D \tag{2.37}$$

is the transformation matrix that accounts for the contribution of the residual fixed-interface normal modes.

After performing the products, the transformation matrix $\mathbf{T}_R \in \mathbb{R}^{n \times n_D}$ can be expressed as [3]

$$\mathbf{T}_R = \begin{bmatrix} -\bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\left(\mathbf{M}_I - \mathbf{M}_{iI}^T\mathbf{M}_{iI}\right)^{-1}\mathbf{M}_{iI}^T\boldsymbol{\Lambda} & \bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\left(\mathbf{M}_I - \mathbf{M}_{iI}^T\mathbf{M}_{iI}\right)^{-1}\mathbf{K}_I \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.38}$$

where

$$\mathbf{M}_I = \tilde{\mathbf{T}}^T\left[\hat{\mathbf{M}}_{bb}^1, \ldots, \hat{\mathbf{M}}_{bb}^{N_s}\right]\tilde{\mathbf{T}} \tag{2.39}$$

$$\mathbf{M}_{iI} = \left[\hat{\mathbf{M}}_{ib}^1, \ldots, \hat{\mathbf{M}}_{ib}^{N_s}\right]\tilde{\mathbf{T}} \tag{2.40}$$

$$\boldsymbol{\Lambda} = \left[\boldsymbol{\Lambda}_{id}^1, \ldots, \boldsymbol{\Lambda}_{id}^{N_s}\right] \tag{2.41}$$

and

$$\mathbf{K}_I = \tilde{\mathbf{T}}^T\left[\hat{\mathbf{K}}_{bb}^1, \ldots, \hat{\mathbf{K}}_{bb}^{N_s}\right]\tilde{\mathbf{T}} \tag{2.42}$$

The matrix $\mathbf{T}_D + \mathbf{T}_R$ represents an improved transformation matrix that explicitly incorporates the contribution of the substructures' residual modes into the analysis.

2.4.2 Enhanced Reduced-Order Matrices

The enhanced reduced-order mass matrix $\hat{\mathbf{M}}_R \in \mathbb{R}^{n_D \times n_D}$ and stiffness matrix $\hat{\mathbf{K}}_R \in \mathbb{R}^{n_D \times n_D}$ are based on the transformation matrix $\mathbf{T}_D + \mathbf{T}_R$ and are defined as

$$\begin{aligned} \hat{\mathbf{M}}_R &= \left(\mathbf{T}_D + \mathbf{T}_R\right)^T \hat{\mathbf{M}}\left(\mathbf{T}_D + \mathbf{T}_R\right) \\ &= \hat{\mathbf{M}}_D + \mathbf{T}_R^T\hat{\mathbf{M}}\mathbf{T}_D + \mathbf{T}_D^T\hat{\mathbf{M}}\mathbf{T}_R + \mathbf{T}_R^T\hat{\mathbf{M}}\mathbf{T}_R \end{aligned} \tag{2.43}$$

and

$$\begin{aligned} \hat{\mathbf{K}}_R &= \left(\mathbf{T}_D + \mathbf{T}_R\right)^T \hat{\mathbf{K}}\left(\mathbf{T}_D + \mathbf{T}_R\right) \\ &= \hat{\mathbf{K}}_D + \mathbf{T}_R^T\hat{\mathbf{K}}\mathbf{T}_D + \mathbf{T}_D^T\hat{\mathbf{K}}\mathbf{T}_R + \mathbf{T}_R^T\hat{\mathbf{K}}\mathbf{T}_R \end{aligned} \tag{2.44}$$

Because of the explicit contribution of the residual fixed-interface normal modes through the matrix $\mathbf{T}_R$ it is expected that the enhanced reduced-order matrices $\hat{\mathbf{M}}_R$ and $\hat{\mathbf{K}}_R$ are more precisely constructed compared to the matrices obtained from the formulation based on dominant modes only.

2.4.3 Some Comments on the Use of Residual Fixed-Interface Normal Modes

The use of residual fixed-interface normal modes in the reduced-order model improves the approximation of the response [16][15] but not without an added computational cost. The residual flexibility matrix for each substructure $\overline{\mathbf{F}}^s$ defined in Eq. (2.31), which is needed in the enhanced formulation, has dimension $n_i^s \times n_i^s$ and is certainly full [1] (p.867). As a result, the computational effort required in the solution of Eq. (2.31) and the storage requirement for $\overline{\mathbf{F}}^s$ increase rapidly as the order of $\mathbf{K}_{ii}^s$ becomes large.

In other words, it can be very computationally expensive to construct the enhanced reduced-order matrices for big finite element models even though the dimensions of the enhanced reduced-order matrices are the same as those of the reduced-order matrices of the standard formulation, that is $n_D \times n_D$.

3 Theory for Interface Reduction in the Craig-Bampton Method

In the previous chapter, the basic theory of the standard Craig-Bampton CMS method was presented along with the enhanced formulation that incorporates residual fixed-interface normal modes.

Both the standard and improved formulations do not consider order reduction for the interface degrees of freedom. This is clear in Eq. (2.18) which shows that the vector of generalized coordinates $\mathbf{q}(t)$ contains all the physical interface DOF for all substructures in the vector $\mathbf{u}_I(t)$ of independent interface coordinates.

This makes the assembly of substructures into a reduced-order system model relatively simple but means that the reduced-order assembly will have as many interface degrees of freedom as the full model. When the full-model mesh is highly refined, and/or when the system is divided into many subcomponents, this can lead to an unacceptably large system of equations of motion. To overcome this, interface reduction methods aim to reduce the size of the Craig-Bampton model by reducing the number of interface degrees of freedom [7].

Most of the methods of reducing interface DOF are either global (system-level) or local (substructure-level) techniques [7].

This chapter aims to present one global technique and one local technique proposed in [3].

The global-level technique treats all interface DOF together by accessing the whole interface partition of the reduced-order matrices at once.

On the other hand, the local-level method considers each interface separately by accessing only the corresponding DOF in the interface partition of the reduced-order matrices sequentially (or in parallel).

3.1 Global-Level Interface Reduction

Global-level methods of reducing interface DOF are shown to be orders of magnitude more accurate compared to local-level methods because the coupling between all substructures has been properly considered [7]. They are also easier to formulate and implement since all interfaces are treated as one (global) interface and no interface-selection problems occur (as will be seen in the next section with local-level reduction).

On the other hand, because an eigenvalue analysis must be performed on the whole interface partition of the reduced-order matrices at once, the computational cost of such a method may be unacceptable for large scale FE models with millions of interface DOF. Another possible drawback of this type of interface reduction is found in parameterized FE models (see chapter 5) concerning the number of support points needed for accurate interpolation of interface modes.

In this section a global-level formulation proposed by [12] is presented which is based on derivations present in chapter 1 of [3].

3.1.1 Interface (Characteristic-Constraint) Modes

The reduced-order matrices $\hat{\mathbf{M}}_D$ and $\hat{\mathbf{K}}_D$ in Eqs. (2.23) and (2.24) can be rewritten equivalently as

$$\hat{\mathbf{M}}_D = \begin{bmatrix} \mathbf{I} & \mathbf{M}_{iI} \\ \mathbf{M}_{iI}^T & \mathbf{M}_I \end{bmatrix} \tag{3.1}$$

and

$$\hat{\mathbf{K}}_D = \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_I \end{bmatrix} \tag{3.2}$$

where all partitions of the reduced-order matrices are already defined.

The proposed global-level method of interface reduction uses the so-called characteristic constraint modes. They are the eigenvectors corresponding to an eigen-analysis of the matrices $\mathbf{M}_I$ and $\mathbf{K}_I$ (the constraint-mode partitions of the reduced-order matrices), that is,

$$\mathbf{K}_I \boldsymbol{Y}_I - \mathbf{M}_I \boldsymbol{Y}_I \boldsymbol{\Omega}_I = \mathbf{0} \tag{3.3}$$

where the matrix $\boldsymbol{Y}_I \in \mathbb{R}^{n_I \times n_{IR}}$ contains the truncated set of interface modes and $\boldsymbol{\Omega}_I \in \mathbb{R}^{n_{IR} \times n_{IR}}$ is the diagonal matrix that contains the corresponding eigenvalues.

The kept interface modes are normalized with respect to $\mathbf{M}_I$, satisfying

$$\boldsymbol{Y}_I^T \mathbf{M}_I \boldsymbol{Y}_I = \mathbf{I}_I \tag{3.4}$$

and

$$\boldsymbol{Y}_I^T \mathbf{K}_I \boldsymbol{Y}_I = \boldsymbol{\Omega}_I \tag{3.5}$$

where $\mathbf{I}_I \in \mathbb{R}^{n_{IR} \times n_{IR}}$ is the identity matrix. The number of kept modes can be small compared to the number of interface DOF of the unreduced model, $n_{IR} \ll n_I$, leading to a highly reduced model.

3.1.2 Reduced-Order Matrices Based on Dominant Fixed-Interface Modes and Global Interface Reduction

The truncated set of interface modes is used to approximate the vector of physical coordinates at the N_I independent interfaces $\mathbf{u}_I(t)$, that is,

$$\mathbf{u}_I(t) = \boldsymbol{Y}_I \boldsymbol{\gamma}(t) \tag{3.6}$$

where $\boldsymbol{\gamma}(t) \in \mathbb{R}^{n_{IR}}$ are the modal coordinates representing the interface DOF. Using Eq. (3.6), the vector of physical coordinates of all substructures $\bar{\mathbf{u}}(t)$ can be expressed as

$$\bar{\mathbf{u}}(t) = \mathbf{T}_D \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_I \end{bmatrix} \mathbf{q}_I(t) = \mathbf{T}_{DI}\mathbf{q}_I(t) \tag{3.7}$$

where $\mathbf{q}_I(t)$ is the vector of generalized coordinates

$$\mathbf{q}_I(t) = \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \boldsymbol{\gamma}(t) \end{Bmatrix} \in \mathbb{R}^{n_{DI}}, \quad n_{DI} = n_{id} + n_{IR} \tag{3.8}$$

and

$$\mathbf{T}_{DI} = \begin{bmatrix} \left[\boldsymbol{\Phi}_{id}^1, \ldots, \boldsymbol{\Phi}_{id}^{N_s}\right] & \left[\boldsymbol{\Psi}_{ib}^1, \ldots, \boldsymbol{\Psi}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_I \\ \mathbf{0} & \boldsymbol{Y}_I \end{bmatrix} \in \mathbb{R}^{n \times n_{DI}} \tag{3.9}$$

is the transformation matrix that considers the effect of the dominant fixed-interface normal modes and interface reduction.

The reduced-order mass and stiffness matrices corresponding to the vector of generalized coordinates $\mathbf{q}_I(t)$, are defined as

$$\begin{aligned} \hat{\mathbf{M}}_{DI} &= \mathbf{T}_{DI}^T \hat{\mathbf{M}} \mathbf{T}_{DI} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{M}_{iI}\boldsymbol{Y}_I \\ \boldsymbol{Y}_I^T \mathbf{M}_{iI}^T & \mathbf{I}_I \end{bmatrix} \in \mathbb{R}^{n_{DI} \times n_{DI}} \end{aligned} \tag{3.10}$$

and

$$\begin{aligned} \hat{\mathbf{K}}_{DI} &= \mathbf{T}_{DI}^T \hat{\mathbf{K}} \mathbf{T}_{DI} \\ &= \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_I \end{bmatrix} \in \mathbb{R}^{n_{DI} \times n_{DI}} \end{aligned} \tag{3.11}$$

The dominant fixed-interface normal modes and the characteristic-constraint modes can be used to define a reduced-order model with fewer generalized coordinates compared to the case without interface reduction, that is, $n_{DI} < n_D \ll n$.

3.1.3 Reduced-Order Matrices Based on Residual Fixed-Interface Modes and Global Interface Reduction

Taking into account Eq. (3.6) and Eq. (2.35) (using $\hat{\mathbf{M}}_{DI}$ and $\hat{\mathbf{K}}_{DI}$ in place of $\hat{\mathbf{M}}_D$ and $\hat{\mathbf{K}}_D$, respectively), Eq. (2.34) can be rewritten as,

$$\begin{aligned}\bar{\mathbf{u}}(t) &= \mathbf{T}_D \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_I \end{bmatrix} \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \boldsymbol{\gamma}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{0} & \bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_I \end{bmatrix} \hat{\mathbf{M}}_{DI}^{-1}\hat{\mathbf{K}}_{DI} \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \boldsymbol{\gamma}(t) \end{Bmatrix} \\ &= \mathbf{T}_{DI} \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \boldsymbol{\gamma}(t) \end{Bmatrix} + \mathbf{T}_{RI} \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \boldsymbol{\gamma}(t) \end{Bmatrix} \\ &= \left(\mathbf{T}_{DI} + \mathbf{T}_{RI}\right)\mathbf{q}_I(t) \end{aligned} \tag{3.12}$$

Carrying out the corresponding products, the transformation matrix $\mathbf{T}_{RI}$ is written, similar to Eq. (2.38), as

$$\mathbf{T}_{RI} = \\ \begin{bmatrix} -\bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\boldsymbol{Y}_I\left(\mathbf{I}_I - \mathbf{M}_{iIR}^T\mathbf{M}_{iIR}\right)^{-1}\mathbf{M}_{iIR}^T\boldsymbol{\Lambda} & \bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\boldsymbol{Y}_I\left(\mathbf{I}_I - \mathbf{M}_{iIR}^T\mathbf{M}_{iIR}\right)^{-1}\boldsymbol{\Omega}_I \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.13}$$

where

$$\mathbf{M}_{iIR} = \left[\hat{\mathbf{M}}_{ib}^{1}, \ldots, \hat{\mathbf{M}}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_I \tag{3.14}$$

The reduced-order mass matrix $\hat{\mathbf{M}}_{RI} \in \mathbb{R}^{n_{DI} \times n_{DI}}$ and stiffness matrix $\hat{\mathbf{K}}_{RI} \in \mathbb{R}^{n_{DI} \times n_{DI}}$ that consider residual fixed-interface normal modes and interface reduction are derived using Eq. (3.12) as

$$\begin{aligned}\hat{\mathbf{M}}_{RI} &= \left(\mathbf{T}_{DI} + \mathbf{T}_{RI}\right)^T \hat{\mathbf{M}}\left(\mathbf{T}_{DI} + \mathbf{T}_{RI}\right) \\ &= \hat{\mathbf{M}}_{DI} + \mathbf{T}_{RI}^T\hat{\mathbf{M}}\mathbf{T}_{DI} + \mathbf{T}_{DI}^T\hat{\mathbf{M}}\mathbf{T}_{RI} + \mathbf{T}_{RI}^T\hat{\mathbf{M}}\mathbf{T}_{RI}\end{aligned} \tag{3.15}$$

and

$$\begin{aligned}\hat{\mathbf{K}}_{RI} &= \left(\mathbf{T}_{DI} + \mathbf{T}_{RI}\right)^T \hat{\mathbf{K}}\left(\mathbf{T}_{DI} + \mathbf{T}_{RI}\right) \\ &= \hat{\mathbf{K}}_{DI} + \mathbf{T}_{RI}^T\hat{\mathbf{K}}\mathbf{T}_{DI} + \mathbf{T}_{DI}^T\hat{\mathbf{K}}\mathbf{T}_{RI} + \mathbf{T}_{RI}^T\hat{\mathbf{K}}\mathbf{T}_{RI}\end{aligned} \tag{3.16}$$

These reduced-order matrices are expected to be more accurately constructed relative to the corresponding matrices that consider only dominant fixed-interface normal modes. Concerning the computational cost of constructing them, the same comments apply as in section 2.4.3 concerning enhanced reduced-order matrices without reduction of interface DOF.

3.2 Local-Level Interface Reduction

As already mentioned, local-level interface reduction techniques do not perform an eigenvalue analysis on the whole interface partition of the reduced-order matrices. For large FE models where such an analysis would be prohibitively expensive computationally, local-level methods are more attractive than the corresponding global-level methods.

Apart from the case of large FE models, one might prefer to use local-level methods when working with parameterized FE models (e.g. in Bayesian model updating). It will be shown that

- in contrast to global-level methods - local-level methods generally require less support points in the parameter space to accurately interpolate interface modes in each sample point. This translates to less computational time per iteration compared to the global-level techniques.

This section introduces a local-level method inspired by [13] and transformed to fit in the Craig-Bampton framework presented in [3].

3.2.1 Definition of Interfaces

The local-level interface reduction method presented here treats each interface separately. Therefore the definition of interfaces plays an important role in the result of interface reduction.

The definition of interfaces must be made in a way that there occur no distinct interfaces that lie on boundaries which share one or more FE nodes. In other words, boundaries that share one or more FE nodes should be selected together to define a single interface. This way the coupling between interfaces is properly considered.

To make this point clear, **Fig. 3.1** shows a simple 2-D FE model with a coarse quadrilateral mesh. Nodes are designated with dots and elements are delimited with thin lines. The model is partitioned into four substructures S_1, S_2, S_3 and S_4 with boundaries represented by thick lines. It can be seen that boundaries 1 and 2 share a node.

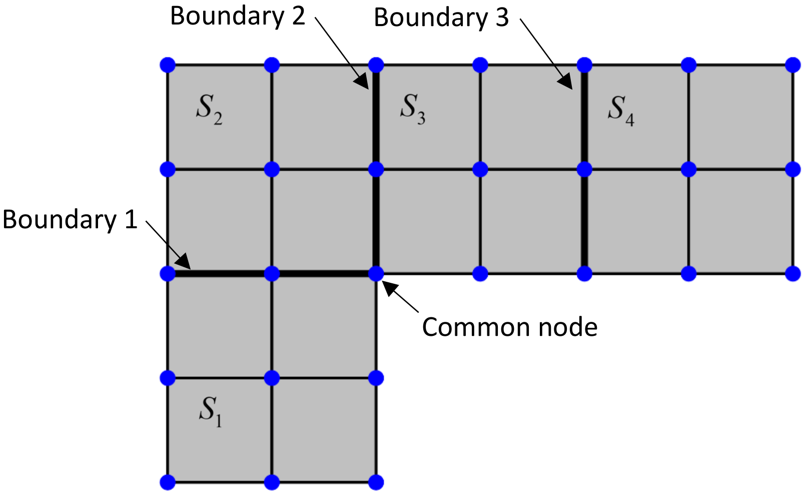

Fig. 3.1 Simple 2-D FE model used for clarification of correct interface definition.

In **Fig. 3.2**, two cases of interface definition of the FE model presented in **Fig. 3.1** are shown. The figure indicates the different nodes that are associated with the physical coordinates of the independent interfaces of the model in vector $\mathbf{u}_I^l$, where l is the interface number.

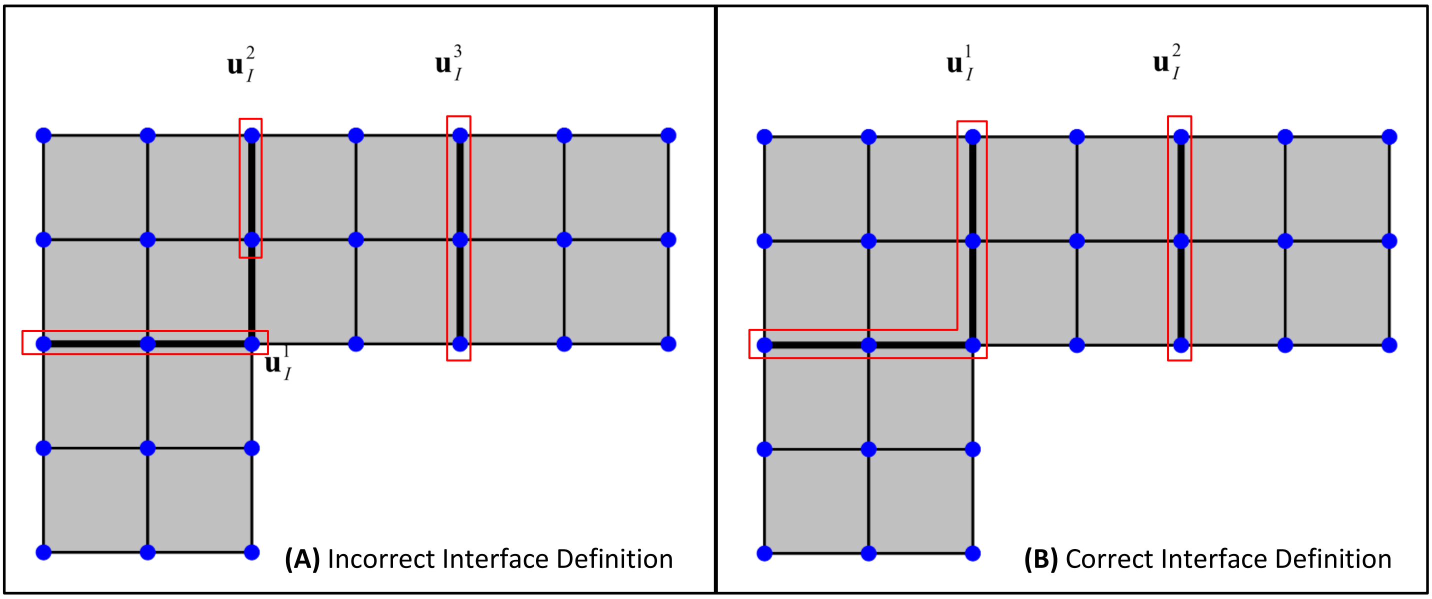

Fig. 3.2 (B) Correct and **(A)** incorrect definition of interfaces.

Case **(A)** represents an incorrect definition of interfaces for local-level reduction. Here, interfaces 1 and 2 are distinct interfaces that belong on boundaries 1 and 2, respectively, which share a node. This interface definition does not consider the coupling between interfaces 1 and 2 and leads to erroneous results.

Case **(B)** indicates the correct way to define interfaces. Now interfaces that lie on boundaries which share a node are selected together (interfaces 1 and 2 have been merged to interface 1).

It must be noted that another possible definition of interfaces would be to define a single interface containing all nodes that belong to model boundaries. Although correct, this case would yield the same result as using global-level interface reduction.

3.2.2 Local Interface Modes

After defining interfaces according to section 3.2.1, reduction of interface DOF can be considered at the local level.

Let $\mathbf{K}_{Ill} \in \mathbb{R}^{n_I^l \times n_I^l}$ and $\mathbf{M}_{Ill} \in \mathbb{R}^{n_I^l \times n_I^l}$ be the partitions of the interface matrices $\mathbf{K}_I$ and $\mathbf{M}_I$, respectively, associated with the physical coordinates at interface $l, \quad l = 1, \ldots, N_I$, i.e. $\mathbf{u}_I^l(t)$. The interface modes corresponding to interface l satisfy the eigenvalue problem

$$\mathbf{K}_{Ill}\boldsymbol{Y}_{Ill} - \mathbf{M}_{Ill}\boldsymbol{Y}_{Ill}\boldsymbol{\Omega}_{Ill} = \mathbf{0} \qquad (3.17)$$

where $\boldsymbol{Y}_{Ill} \in \mathbb{R}^{n_I^l \times n_{IR}^l}$ contains the kept n_{IR}^l local interface modes and $\boldsymbol{\Omega}_{Ill} \in \mathbb{R}^{n_{IR}^l \times n_{IR}^l}$ is the diagonal matrix that contains the corresponding eigenvalues. The local interface modes are mass normalized, that is

$$\boldsymbol{Y}_{Ill}^T \mathbf{M}_{Ill} \boldsymbol{Y}_{Ill} = \mathbf{I}_{Ill} \tag{3.18}$$

and

$$\boldsymbol{Y}_{Ill}^T \mathbf{K}_{Ill} \boldsymbol{Y}_{Ill} = \boldsymbol{\Omega}_{Ill} \tag{3.19}$$

where $\mathbf{I}_{Ill} \in \mathbb{R}^{n_{IR}^l \times n_{IR}^l}$ is the identity matrix. These modes are used to represent the vector of physical coordinates $\mathbf{u}_I^l(t)$ at interface l in terms of the local interface modal coordinates $\boldsymbol{\gamma}_l(t) \in \mathbb{R}^{n_{IR}^l}$ in the form

$$\mathbf{u}_I^l(t) = \boldsymbol{Y}_{Ill} \boldsymbol{\gamma}_l(t) \tag{3.20}$$

The kept local interface modes for each interface are used to define $\boldsymbol{Y}_{IL}$, a matrix similar to $\boldsymbol{Y}_I$ defined in Eq. (3.3) for global-level interface reduction, that is,

$$\boldsymbol{Y}_{IL} = \left[\boldsymbol{Y}_{I11}, \ldots, \boldsymbol{Y}_{IN_I N_I}\right] \in \mathbb{R}^{n_I \times n_{IRL}}, \quad n_{IRL} = \sum\nolimits_{l=1}^{N_I} n_{IR}^l \tag{3.21}$$

3.2.3 Reduced-Order Matrices Based on Dominant Fixed-Interface Modes and Local Interface Reduction

The vector of physical coordinates of all substructures can be approximated similarly to Eq. (3.7) as

$$\bar{\mathbf{u}}(t) = \mathbf{T}_D \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_{IL} \end{bmatrix} \mathbf{q}_{IL}(t) = \mathbf{T}_{DIL} \mathbf{q}_{IL}(t) \tag{3.22}$$

where $\mathbf{q}_{IL}(t)$ is the vector of generalized coordinates

$$\mathbf{q}_{IL}(t) = \begin{Bmatrix} \boldsymbol{\eta}(t) \\ \boldsymbol{\gamma}_L(t) \end{Bmatrix} \in \mathbb{R}^{n_{DIL}}, \quad n_{DIL} = n_{id} + n_{IRL}, \tag{3.23}$$

$\boldsymbol{\gamma}_L(t)$ is the vector of local interface modal coordinates of all independent interfaces

$$\boldsymbol{\gamma}_L(t) = \begin{Bmatrix} \boldsymbol{\gamma}_1(t) \\ \vdots \\ \boldsymbol{\gamma}_{N_I}(t) \end{Bmatrix} \in \mathbb{R}^{n_{IRL}}, \tag{3.24}$$

and

$$\mathbf{T}_{DIL} = \begin{bmatrix} \left[\boldsymbol{\Phi}_{id}^{1},\ldots,\boldsymbol{\Phi}_{id}^{N_s}\right] & \left[\boldsymbol{\Psi}_{ib}^{1},\ldots,\boldsymbol{\Psi}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL} \\ \mathbf{0} & \boldsymbol{Y}_{IL} \end{bmatrix} \in \mathbb{R}^{n\times n_{DIL}} \tag{3.25}$$

is the transformation matrix that accounts for the effect of the dominant fixed-interface normal modes and the local interface normal modes. The corresponding reduced-order matrices are defined as

$$\begin{aligned} \hat{\mathbf{M}}_{DIL} &= \mathbf{T}_{DIL}^{T}\hat{\mathbf{M}}\mathbf{T}_{DIL} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{M}_{iI}\boldsymbol{Y}_{IL} \\ \boldsymbol{Y}_{IL}^{T}\mathbf{M}_{iI}^{T} & \boldsymbol{Y}_{IL}^{T}\mathbf{M}_{I}\boldsymbol{Y}_{IL} \end{bmatrix} \in \mathbb{R}^{n_{DIL}\times n_{DIL}} \end{aligned} \tag{3.26}$$

and

$$\begin{aligned} \hat{\mathbf{K}}_{DIL} &= \mathbf{T}_{DIL}^{T}\hat{\mathbf{K}}\mathbf{T}_{DIL} \\ &= \begin{bmatrix} \boldsymbol{\Lambda} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_{IL}^{T}\mathbf{K}_{I}\boldsymbol{Y}_{IL} \end{bmatrix} \in \mathbb{R}^{n_{DIL}\times n_{DIL}} \end{aligned} \tag{3.27}$$

3.2.4 Reduced-Order Matrices Based on Residual Fixed-Interface Modes and Local Interface Reduction

In the case of local-level interface reduction, the transformation matrix that considers residual fixed-interface normal modes becomes

$$\mathbf{T}_{RIL} = \begin{bmatrix} \mathbf{T}_{RIL,1} & \mathbf{T}_{RIL,2} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n\times n_{DIL}} \tag{3.28}$$

where

$$\mathbf{T}_{RIL,1} = -\bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}\left(\boldsymbol{Y}_{IL}^{T}\mathbf{M}_{I}\boldsymbol{Y}_{IL} - \mathbf{M}_{iIRL}^{T}\mathbf{M}_{iIRL}\right)^{-1}\mathbf{M}_{iIRL}^{T}\boldsymbol{\Lambda} \tag{3.29}$$

and

$$\mathbf{T}_{RIL,2} = \bar{\mathbf{F}}\tilde{\mathbf{M}}_{ib}\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}\left(\boldsymbol{Y}_{IL}^{T}\mathbf{M}_{I}\boldsymbol{Y}_{IL} - \mathbf{M}_{iIRL}^{T}\mathbf{M}_{iIRL}\right)^{-1}\boldsymbol{Y}_{IL}^{T}\mathbf{K}_{I}\boldsymbol{Y}_{IL} \tag{3.30}$$

with

$$\mathbf{M}_{iIRL} = \left[\hat{\mathbf{M}}_{ib}^{1},\ldots,\hat{\mathbf{M}}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL} \tag{3.31}$$

Taking into consideration the effect of residual fixed-interface normal modes, Eq. (3.22) becomes

$$\bar{\mathbf{u}}(t) = \left(\mathbf{T}_{DIL} + \mathbf{T}_{RIL}\right)\mathbf{q}_{IL}(t) \tag{3.32}$$

The associated reduced-order mass matrix $\hat{\mathbf{M}}_{RIL} \in \mathbb{R}^{n_{DIL}\times n_{DIL}}$ and stiffness matrix $\hat{\mathbf{K}}_{RIL} \in \mathbb{R}^{n_{DIL}\times n_{DIL}}$ change to

$$\begin{aligned}\hat{\mathbf{M}}_{RIL} &= \left(\mathbf{T}_{DIL}+\mathbf{T}_{RIL}\right)^T \hat{\mathbf{M}}\left(\mathbf{T}_{DIL}+\mathbf{T}_{RIL}\right) \\ &= \hat{\mathbf{M}}_{DIL}+\mathbf{T}_{RIL}^T\hat{\mathbf{M}}\mathbf{T}_{DIL}+\mathbf{T}_{DIL}^T\hat{\mathbf{M}}\mathbf{T}_{RIL}+\mathbf{T}_{RIL}^T\hat{\mathbf{M}}\mathbf{T}_{RIL}\end{aligned} \tag{3.33}$$

and

$$\begin{aligned}\hat{\mathbf{K}}_{RIL} &= \left(\mathbf{T}_{DIL}+\mathbf{T}_{RIL}\right)^T \hat{\mathbf{K}}\left(\mathbf{T}_{DIL}+\mathbf{T}_{RIL}\right) \\ &= \hat{\mathbf{K}}_{DIL}+\mathbf{T}_{RIL}^T\hat{\mathbf{K}}\mathbf{T}_{DIL}+\mathbf{T}_{DIL}^T\hat{\mathbf{K}}\mathbf{T}_{RIL}+\mathbf{T}_{RIL}^T\hat{\mathbf{K}}\mathbf{T}_{RIL}\end{aligned} \tag{3.34}$$

The same comments apply as in section 2.4.3 concerning the computational cost of constructing the enhanced transformation matrix and consequently the corresponding reduced-order matrices

3.2.5 Some Comments on the definition of $\mathbf{u}_I(t)$ and $\boldsymbol{Y}_{IL}$

The presented method of local-level interface reduction works well if carefully applied.

A point that requires special attention is the correct definition of the vector $\mathbf{u}_I(t)$ of physical coordinates at the N_I independent interfaces in Eq. (2.14) and the matrix $\boldsymbol{Y}_{IL}$ containing the truncated set of local interface modes for each independent interface in Eq. (3.21).

The definition of $\mathbf{u}_I(t)$ influences the matrix $\tilde{\mathbf{T}}$ as can be seen in Eq. (2.16) which in turn affects the definition of the constraint-mode partitions of the reduced-order matrices $\mathbf{M}_I$ and $\mathbf{K}_I$ in Eqs. (2.39) and (2.42) respectively.

On the other hand, $\boldsymbol{Y}_{IL}$ is a block-diagonal matrix that contains the local interface modes of each interface. Since the physical coordinates of all independent interfaces are kept in $\mathbf{u}_I(t)$, each row of $\boldsymbol{Y}_{IL}$ is associated to the same interface DOF as in the corresponding row of $\mathbf{u}_I(t)$.

The definition of the reduced-order matrices involves the multiplication of $\boldsymbol{Y}_{IL}$ with $\mathbf{M}_I$ and $\mathbf{K}_I$ in Eqs. (3.26) and (3.27) respectively. If $\mathbf{u}_I(t)$ and $\boldsymbol{Y}_{IL}$ are not correctly defined (their rows do not correspond to the same interface DOF) this multiplication yields erroneous results, and the corresponding reduced-order matrices are wrong.

One way to check that $\mathbf{u}_I(t)$ and $\boldsymbol{Y}_{IL}$ are correctly constructed is to take into account the mass normalization of the interface modes in Eqs. (3.18) and (3.19) which should give

$$\boldsymbol{Y}_{IL}^T\mathbf{M}_I\boldsymbol{Y}_{IL} \approx \left[\mathbf{I}_{I11},\ldots,\mathbf{I}_{IN_IN_I}\right] \tag{3.35}$$

and

$$\boldsymbol{Y}_{IL}^T\mathbf{K}_I\boldsymbol{Y}_{IL} \approx \left[\boldsymbol{\Omega}_{I11},\ldots,\boldsymbol{\Omega}_{IN_IN_I}\right] \tag{3.36}$$

In case the definition of interfaces changes, one should be careful to update $\mathbf{u}_I(t)$ and $\boldsymbol{Y}_{IL}$ as needed.

4 Parametrization of Reduced-Order Models based on Fixed-Interface Normal Modes

The solution to many dynamic simulation-based problems involving uncertainty requires evaluating the response of the modeled system at a large number of samples in the uncertain parameter space (of the order of hundreds or thousands). This fact makes such problems computationally very demanding especially when the time of a single analysis is significant.

Obviously, model reduction techniques such as that presented in sections 2 and 3 alleviate part of the computational burden by projecting the model to a greatly reduced set of generalized coordinates. However, the time-consuming process of calculating the fixed-interface normal modes and interface constraint modes at each sample point remains. This is necessary due to the fact that changes in model parameter values affect the modal characteristics and the static response of the structure. This procedure greatly increases simulation times, due to the substantial computational overhead that arises at the substructure level.

To avoid reconstructing the reduced-order model at each sample point, an efficient parametrization scheme is presented in this chapter. When the reduced-order model is parametrized using this scheme, the calculation of the fixed-interface normal modes and interface constraint modes at each sample point is completely avoided. With such a parametrized model, it is necessary to compute these quantities only once. As a result, even greater computational savings can be achieved compared to the non-parametrized reduced-order model formulation.

Throughout this chapter, it is assumed that the FE model is parametrized by a set of uncertain parameters $\theta \in \Omega_\theta \subset \mathbb{R}^{n_\theta}$ which are modeled using a probability density function $q(\theta)$ that indicates the relative plausibility of the possible values of the parameters. The formulations presented next are based on chapter 2 of [3].

4.1 Parametrization Scheme

The original structure is parametrized assuming that the mass and stiffness matrices for each linear substructure s, $s = 1, \dots, N_s$, depend on only one (or none) of the model parameters.

4.1.1 Matrices of substructures that do not depend on model parameters

If a substructure s does not depend on any model parameter, it belongs to S_0 which is the set of substructures that do not depend on the vector of model parameters θ, that is $s \in S_0$. In this case, the substructure mass and stiffness matrices are written as

$$\mathbf{M}^s = \bar{\mathbf{M}}^s \tag{4.1}$$

and

$$\mathbf{K} = \bar{\mathbf{K}}^{s} \tag{4.2}$$

Since the mass and stiffness matrix of substructures $s \in S_0$ are independent of model parameters, their fixed-interface normal modes and their interface constraint modes are also parameter-independent. This means that the time-consuming process of solving the eigenproblem to compute $\boldsymbol{\Phi}_{ii}^{s}$ in Eq. (2.4) and the linear system to compute $\boldsymbol{\Psi}_{ib}^{s}$ in Eq. (2.9) are performed only once for these substructures.

4.1.2 Matrices of substructures that depend on the model parameter θ_j

Let S_j be the set of substructures that depend on the model parameter θ_j. For substructures $s \in S_j$, the substructure matrices take the general form

$$\mathbf{M}^{s} = \bar{\mathbf{M}}^{s} g^{j}(\theta_j) \tag{4.3}$$

and

$$\mathbf{K}^{s} = \bar{\mathbf{K}}^{s} h^{j}(\theta_j) \tag{4.4}$$

where $g^{j}(\theta_j)$ and $h^{j}(\theta_j)$ are linear or nonlinear functions of θ_j and the matrices $\bar{\mathbf{M}}^{s}$ and $\bar{\mathbf{K}}^{s}$ are independent of θ_j obtained from the reference model by setting $g^{j}(\theta_j) = 1$ and $h^{j}(\theta_j) = 1$. The partitions of the mass matrix $\mathbf{M}^{s}$ in Eq. (2.2) and the stiffness matrix $\mathbf{K}^{s}$ in Eq. (2.3) admit the same parametrization, that is

$$\begin{aligned} \mathbf{M}_{ii}^{s} &= \bar{\mathbf{M}}_{ii}^{s} g^{j}(\theta_j) \\ \mathbf{M}_{ib}^{s} &= \bar{\mathbf{M}}_{ib}^{s} g^{j}(\theta_j) \\ \mathbf{M}_{bi}^{s} &= \bar{\mathbf{M}}_{bi}^{s} g^{j}(\theta_j) \\ \mathbf{M}_{bb}^{s} &= \bar{\mathbf{M}}_{bb}^{s} g^{j}(\theta_j) \end{aligned} \tag{4.5}$$

and

$$\begin{aligned} \mathbf{K}_{ii}^{s} &= \bar{\mathbf{K}}_{ii}^{s} h^{j}(\theta_j) \\ \mathbf{K}_{ib}^{s} &= \bar{\mathbf{K}}_{ib}^{s} h^{j}(\theta_j) \\ \mathbf{K}_{bi}^{s} &= \bar{\mathbf{K}}_{bi}^{s} h^{j}(\theta_j) \\ \mathbf{K}_{bb}^{s} &= \bar{\mathbf{K}}_{bb}^{s} h^{j}(\theta_j) \end{aligned} \tag{4.6}$$

where all matrices with an over-bar are independent of θ_j.

The above parametrization is often encountered in structural FE systems [17][18].

4.1.3 Fixed-Interface Normal Modes and Interface Constraint Modes

Taking into account the parametrization of the mass matrix $\mathbf{M}_{ii}^{s}$ in Eq. (4.5) and the normalization of the fixed-interface normal modes in Eq. (2.5) with respect to that matrix, it follows that the matrix of fixed-interface normal modes $\boldsymbol{\Phi}_{ii}^{s},\, s \in S_j$ can be written as

$$\boldsymbol{\Phi}_{ii}^{s} = \bar{\boldsymbol{\Phi}}_{ii}^{s} \frac{1}{\sqrt{g^{j}(\theta_j)}} \tag{4.7}$$

where the matrix $\bar{\boldsymbol{\Phi}}_{ii}^{s}$ is independent of the model parameter θ_j.

Next, if the previous parametrization of $\mathbf{M}_{ii}^{s}$, $\mathbf{K}_{ii}^{s}$ and $\boldsymbol{\Phi}_{ii}^{s}$ in the eigenvalue problem (2.4) is considered, the diagonal matrix of the corresponding eigenvalues $\boldsymbol{\Lambda}_{ii}^{s},\, s \in S_j$ allows the parametrization

$$\boldsymbol{\Lambda}_{ii}^{s} = \bar{\boldsymbol{\Lambda}}_{ii}^{s} \frac{h^{j}(\theta_j)}{g^{j}(\theta_j)} \tag{4.8}$$

where the matrix $\bar{\boldsymbol{\Lambda}}_{ii}^{s}$ is independent of the model parameter θ_j. From the above parametrizations of $\boldsymbol{\Phi}_{ii}^{s}$ and $\boldsymbol{\Lambda}_{ii}^{s}$, it is evident that $\bar{\boldsymbol{\Phi}}_{ii}^{s}$ and $\bar{\boldsymbol{\Lambda}}_{ii}^{s}$ are computed from the eigenproblem

$$\bar{\mathbf{K}}_{ii}^{s} \bar{\boldsymbol{\Phi}}_{ii}^{s} - \bar{\mathbf{M}}_{ii}^{s} \bar{\boldsymbol{\Phi}}_{ii}^{s} \bar{\boldsymbol{\Lambda}}_{ii}^{s} = \mathbf{0} \tag{4.9}$$

where the mode shapes are mass normalized as

$$\bar{\boldsymbol{\Phi}}_{ii}^{s^{T}} \bar{\mathbf{M}}_{ii}^{s} \bar{\boldsymbol{\Phi}}_{ii}^{s} = \mathbf{I}_{ii}^{s} \tag{4.10}$$

and

$$\bar{\boldsymbol{\Phi}}_{ii}^{s^{T}} \bar{\mathbf{K}}_{ii}^{s} \bar{\boldsymbol{\Phi}}_{ii}^{s} = \bar{\boldsymbol{\Lambda}}_{ii}^{s} \tag{4.11}$$

Concerning the interface constraint modes $\boldsymbol{\Psi}_{ib}^{s},\, s \in S_j$ in Eq. (2.9), it holds that

$$\boldsymbol{\Psi}_{ib}^{s} = -\mathbf{K}_{ii}^{s^{-1}} \mathbf{K}_{ib}^{s} = -\bar{\mathbf{K}}_{ii}^{s^{-1}} h^{j^{-1}}(\theta_j) \bar{\mathbf{K}}_{ib}^{s} h^{j}(\theta_j) = -\bar{\mathbf{K}}_{ii}^{s^{-1}} \bar{\mathbf{K}}_{ib}^{s} = \bar{\boldsymbol{\Psi}}_{ib}^{s} \tag{4.12}$$

where the matrix $\bar{\boldsymbol{\Psi}}_{ib}^{s}$ is independent of the model parameter θ_j.

From Eqs. (4.7), (4.8) and (4.12) it is clear that the fixed-interface normal modes and interface constraint modes can be estimated for any value of the parameter vector $\boldsymbol{\theta}$ by solving the eigenproblem in Eq. (4.9) and the linear system in Eq. (4.12) once. This aspect of the proposed parametrization scheme greatly reduces the computational cost since it eliminates the need for estimating the above quantities at each step of the simulation process.

4.2 Parametrization of Reduced-Order Matrices Based on Dominant Fixed-Interface Modes

This section introduces the parametrization of the reduced-order mass and stiffness matrices based on dominant fixed-interface normal modes as well as the parametrization of the various matrices involved in their definition.

4.2.1 Unreduced Matrices $\hat{\mathbf{M}}(\boldsymbol{\theta})$ and $\hat{\mathbf{K}}(\boldsymbol{\theta})$

If the parametrization of $\mathbf{M}^s$ and $\mathbf{K}^s$ are taken into account, the mass and stiffness matrix of the unreduced model referring to the vector of physical coordinates of all substructures $\overline{\mathbf{u}}(t)$, given in Eqs. (2.19) and (2.20), respectively, take the form

$$\hat{\mathbf{M}}(\boldsymbol{\theta}) = \begin{bmatrix} \left[\overline{\mathbf{M}}_{ii}^{1}\delta_{10},\ldots,\overline{\mathbf{M}}_{ii}^{N_s}\delta_{N_s0}\right] & \left[\overline{\mathbf{M}}_{ib}^{1}\delta_{10},\ldots,\overline{\mathbf{M}}_{ib}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{M}}_{ib}^{1^T}\delta_{10},\ldots,\overline{\mathbf{M}}_{ib}^{N_s^T}\delta_{N_s0}\right] & \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{M}}_{bb}^{1}\delta_{10},\ldots,\overline{\mathbf{M}}_{bb}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \end{bmatrix}$$
$$+\sum_{j=1}^{n_\theta}\begin{bmatrix} \left[\overline{\mathbf{M}}_{ii}^{1}\delta_{1j},\ldots,\overline{\mathbf{M}}_{ii}^{N_s}\delta_{N_sj}\right] & \left[\overline{\mathbf{M}}_{ib}^{1}\delta_{1j},\ldots,\overline{\mathbf{M}}_{ib}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{M}}_{ib}^{1^T}\delta_{1j},\ldots,\overline{\mathbf{M}}_{ib}^{N_s^T}\delta_{N_sj}\right] & \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{M}}_{bb}^{1}\delta_{1j},\ldots,\overline{\mathbf{M}}_{bb}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \end{bmatrix} g^{j}(\theta_j) \tag{4.13}$$

and

$$\hat{\mathbf{K}}(\boldsymbol{\theta}) = \begin{bmatrix} \left[\overline{\mathbf{K}}_{ii}^{1}\delta_{10},\ldots,\overline{\mathbf{K}}_{ii}^{N_s}\delta_{N_s0}\right] & \left[\overline{\mathbf{K}}_{ib}^{1}\delta_{10},\ldots,\overline{\mathbf{K}}_{ib}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{K}}_{ib}^{1^T}\delta_{10},\ldots,\overline{\mathbf{K}}_{ib}^{N_s^T}\delta_{N_s0}\right] & \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{K}}_{bb}^{1}\delta_{10},\ldots,\overline{\mathbf{K}}_{bb}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \end{bmatrix}$$
$$+\sum_{j=1}^{n_\theta}\begin{bmatrix} \left[\overline{\mathbf{K}}_{ii}^{1}\delta_{1j},\ldots,\overline{\mathbf{K}}_{ii}^{N_s}\delta_{N_sj}\right] & \left[\overline{\mathbf{K}}_{ib}^{1}\delta_{1j},\ldots,\overline{\mathbf{K}}_{ib}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{K}}_{ib}^{1^T}\delta_{1j},\ldots,\overline{\mathbf{K}}_{ib}^{N_s^T}\delta_{N_sj}\right] & \tilde{\mathbf{T}}^{T}\left[\overline{\mathbf{K}}_{bb}^{1}\delta_{1j},\ldots,\overline{\mathbf{K}}_{bb}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \end{bmatrix} h^{j}(\theta_j) \tag{4.14}$$

where

$$\delta_{s0} = \begin{cases} 1 & \text{if } s \in S_0 \\ 0 & \text{otherwise} \end{cases}, \quad s = 1,\ldots,N_s \tag{4.15}$$

and

$$\delta_{sj} = \begin{cases} 1 & \text{if } s \in S_j \\ 0 & \text{otherwise} \end{cases}, \quad s = 1,\ldots,N_s \tag{4.16}$$

4.2.2 Transformation Matrix $\mathbf{T}_D(\boldsymbol{\theta})$

Considering the parametrization of the fixed-interface normal modes $\boldsymbol{\Phi}_{ii}^{s}$ and the independence of interface constraint modes $\boldsymbol{\Psi}_{ib}^{s}$ on model parameters, the transformation matrix $\mathbf{T}_D$ defined in Eq. (2.15) can be written as

$$\begin{aligned}\mathbf{T}_D(\boldsymbol{\theta}) = & \begin{bmatrix} \left[\bar{\boldsymbol{\Phi}}_{id}^{1}\delta_{10},\ldots,\bar{\boldsymbol{\Phi}}_{id}^{N_s}\delta_{N_s 0}\right] & \left[\bar{\boldsymbol{\Psi}}_{ib}^{1},\ldots,\bar{\boldsymbol{\Psi}}_{ib}^{N_s}\right]\tilde{\mathbf{T}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ & + \sum_{j=1}^{n_\theta} \begin{bmatrix} \left[\bar{\boldsymbol{\Phi}}_{id}^{1}\delta_{1j},\ldots,\bar{\boldsymbol{\Phi}}_{id}^{N_s}\delta_{N_s j}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{1}{\sqrt{g^{j}(\theta_j)}} \end{aligned} \tag{4.17}$$

4.2.3 Reduced-Order Matrices $\hat{\mathbf{M}}_D(\boldsymbol{\theta})$ and $\hat{\mathbf{K}}_D(\boldsymbol{\theta})$

Taking into account all previous parametrizations, the matrices $\hat{\mathbf{M}}_{ib}^{s}$, $\hat{\mathbf{M}}_{bb}^{s}$ and $\hat{\mathbf{K}}_{bb}^{s}$ for $s \in S_j$ which are used to define $\hat{\mathbf{M}}_D$ and $\hat{\mathbf{K}}_D$ in Eqs. (2.23) and (2.24) respectively, can be written as

$$\begin{aligned} \hat{\mathbf{M}}_{ib}^{s} &= \hat{\bar{\mathbf{M}}}_{ib}^{s}\sqrt{g^{j}(\theta_j)} \\ \hat{\mathbf{K}}_{bb}^{s} &= \hat{\bar{\mathbf{K}}}_{bb}^{s} h^{j}(\theta_j) \\ \hat{\mathbf{M}}_{bb}^{s} &= \hat{\bar{\mathbf{M}}}_{bb}^{s} g^{j}(\theta_j) \end{aligned} \tag{4.18}$$

where

$$\begin{aligned} \hat{\bar{\mathbf{M}}}_{ib}^{s} &= \bar{\boldsymbol{\Phi}}_{id}^{s^T}\bar{\mathbf{M}}_{ii}^{s}\bar{\boldsymbol{\Psi}}_{ib}^{s} + \bar{\boldsymbol{\Phi}}_{id}^{s^T}\bar{\mathbf{M}}_{ib}^{s} \\ \hat{\bar{\mathbf{K}}}_{bb}^{s} &= \bar{\mathbf{K}}_{ib}^{s^T}\bar{\boldsymbol{\Psi}}_{ib}^{s} + \bar{\mathbf{K}}_{bb}^{s} \\ \hat{\bar{\mathbf{M}}}_{bb}^{s} &= \left(\bar{\boldsymbol{\Psi}}_{ib}^{s^T}\bar{\mathbf{M}}_{ii}^{s} + \bar{\mathbf{M}}_{ib}^{s^T}\right)\bar{\boldsymbol{\Psi}}_{ib}^{s} + \bar{\boldsymbol{\Psi}}_{ib}^{s^T}\bar{\mathbf{M}}_{ib}^{s} + \bar{\mathbf{M}}_{bb}^{s} \end{aligned} \tag{4.19}$$

Considering the above expansions along with the parametrization of the matrices of eigenvalues, the reduced-order mass matrix and stiffness matrix can be expressed as

$$\hat{\mathbf{M}}_D(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{I} & \left[\hat{\bar{\mathbf{M}}}_{ib}^{1}\delta_{10},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{M}}}_{ib}^{1^T}\delta_{10},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s^T}\delta_{N_s0}\right] & \tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{M}}}_{bb}^{1}\delta_{10},\ldots,\hat{\bar{\mathbf{M}}}_{bb}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \end{bmatrix}$$

$$+\sum_{j=1}^{n_\theta}\left\{\begin{bmatrix} \mathbf{0} & \left[\hat{\bar{\mathbf{M}}}_{ib}^{1}\delta_{1j},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \\ \tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{M}}}_{ib}^{1^T}\delta_{1j},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s^T}\delta_{N_sj}\right] & \mathbf{0} \end{bmatrix}\sqrt{g^j(\theta_j)}\right. \tag{4.20}$$

$$\left.+\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{M}}}_{bb}^{1}\delta_{1j},\ldots,\hat{\bar{\mathbf{M}}}_{bb}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \end{bmatrix} g^j(\theta_j)\right\}$$

and

$$\hat{\mathbf{K}}_D(\boldsymbol{\theta}) = \begin{bmatrix} \left[\bar{\boldsymbol{\Lambda}}_{id}^{1}\delta_{10},\ldots,\bar{\boldsymbol{\Lambda}}_{id}^{N_s}\delta_{N_s0}\right] & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{K}}}_{bb}^{1}\delta_{10},\ldots,\hat{\bar{\mathbf{K}}}_{bb}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}} \end{bmatrix}$$

$$+\sum_{j=1}^{n_\theta}\left\{\begin{bmatrix} \left[\bar{\boldsymbol{\Lambda}}_{id}^{1}\delta_{1j},\ldots,\bar{\boldsymbol{\Lambda}}_{id}^{N_s}\delta_{N_sj}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\frac{h^j(\theta_j)}{g^j(\theta_j)}\right. \tag{4.21}$$

$$\left.+\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{K}}}_{bb}^{1}\delta_{1j},\ldots,\hat{\bar{\mathbf{K}}}_{bb}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}} \end{bmatrix} h^j(\theta_j)\right\}$$

where all terms have been previously defined.

4.3 Some Comments on the Proposed Parametrization Scheme

From Eqs. (4.20) and (4.21) it is evident that the reduced-order mass matrix $\hat{\mathbf{M}}_D(\boldsymbol{\theta})$ and stiffness matrix $\hat{\mathbf{K}}_D(\boldsymbol{\theta})$ are expressed explicitly in terms of the model parameter vector $\boldsymbol{\theta}$ and other constant matrices. These constant matrices are computed and assembled once and, therefore, there is no need this computation to be repeated during the iterations of the simulation process [17]. The same fact holds for the transformation matrix $\mathbf{T}_D(\boldsymbol{\theta})$ defined in Eq. (4.17).

Consequently, $\hat{\mathbf{M}}_D(\boldsymbol{\theta})$, $\hat{\mathbf{K}}_D(\boldsymbol{\theta})$ and $\mathbf{T}_D(\boldsymbol{\theta})$ - which are needed to estimate the dynamic response of the original unreduced system - can be estimated at each iteration without the need to reconstruct them at the substructure level which would require solving a computationally intensive eigenproblem and linear system.

Therefore, the presented parametrization scheme allows for substantial computational savings since it avoids (a) re-computing the fixed-interface and constrained modes for each component, and (b) assembling the reduced-order matrices from these components [17].

It must be stressed that the efficiency of the above formulation in terms of the number of substructure analyses required is based on the assumption that the stiffness and mass matrices of the substructures depend only on one (or none) model parameter. For a more general case, the normal and constraint modes have to be recomputed in each iteration of the simulation process [18].

5 Parametrization of Reduced-Order Models based on Fixed-Interface Normal Modes and Interface Reduction

In section 4 a parametrization scheme was presented that can be applied on reduced-order models constructed using the Craig-Bampton CMS method based on fixed-interface normal modes (theory in section 2). This parametrization scheme does not consider reduction of interface DOF which can be a problem for FE models with fine meshes and large numbers of interface DOF. In this case, it is possible that the interface partition of the vector of generalized coordinates dominates the reduced-order model.

As indicated in section 3, reduction of interface DOF can be considered at the global and at the local level. In this section, parametrization schemes that take into account both methods of interface reduction are proposed.

When the parametrized matrices are constructed considering interface reduction, additional computational savings can be achieved by reducing the size of the final model even further.

Firstly, parametrization based on global-level interface reduction is presented followed by the corresponding formulations based on local-level reduction.

5.1 Parametrization Based on Global-Level Interface Reduction

This section presents a parametrization scheme which is similar to that of section 4 but takes into account global-level interface reduction. The formulations are based on chapter 3 of [3].

5.1.1 Meta-Model for Global Interface Modes

The parametrization scheme presented in section 4 was based on the assumption that the substructure matrices depend on one (or none) model parameter. This assumption does not hold for the interface partition $\mathbf{M}_I$ and $\mathbf{K}_I$ of the reduced-order mass matrix $\hat{\mathbf{M}}_D$ and stiffness matrix $\hat{\mathbf{K}}_D$, respectively. In general, these interface matrices depend on multiple model parameters since interface DOF belong on multiple substructures. This means that a direct interface analysis should be performed at each iteration to reduce interface DOF.

To avoid this computationally costly procedure, an interpolation scheme is proposed that approximates the global interface modes at each sample point in terms of the model parameters.

5.1.1.1 Baseline Information

Initially, L support points are defined in the model parameter space ($\boldsymbol{\theta}^m, \quad m = 1,\ldots,L$) and the interface matrices $\mathbf{M}_I$ and $\mathbf{K}_I$ defined in Eqs. (2.39) and (2.42), respectively, are assembled at these points considering parametrizations in Eq. (4.18), that is

$$\mathbf{M}_I(\boldsymbol{\theta}^m) = \tilde{\mathbf{T}}^T \left[\hat{\tilde{\mathbf{M}}}_{bb}^1 \delta_{10}, \ldots, \hat{\tilde{\mathbf{M}}}_{bb}^{N_s} \delta_{N_s 0} \right] \tilde{\mathbf{T}} + \sum_{j=1}^{n_\theta} \tilde{\mathbf{T}}^T \left[\hat{\tilde{\mathbf{M}}}_{bb}^1 \delta_{1j}, \ldots, \hat{\tilde{\mathbf{M}}}_{bb}^{N_s} \delta_{N_s j} \right] \tilde{\mathbf{T}} g^j(\theta_j^m) \tag{5.1}$$

$$\mathbf{K}_I(\boldsymbol{\theta}^m) = \tilde{\mathbf{T}}^T \left[\hat{\tilde{\mathbf{K}}}_{bb}^1 \delta_{10}, \ldots, \hat{\tilde{\mathbf{K}}}_{bb}^{N_s} \delta_{N_s 0} \right] \tilde{\mathbf{T}} + \sum_{j=1}^{n_\theta} \tilde{\mathbf{T}}^T \left[\hat{\tilde{\mathbf{K}}}_{bb}^1 \delta_{1j}, \ldots, \hat{\tilde{\mathbf{K}}}_{bb}^{N_s} \delta_{N_s j} \right] \tilde{\mathbf{T}} h^j(\theta_j^m) \tag{5.2}$$

where θ_j^m is the jth component of the support point $\boldsymbol{\theta}^m$. It is assumed that the support points $\boldsymbol{\theta}^m, \quad m = 1,\ldots,L$ are distributed around the nominal point $\boldsymbol{\theta}^0$.

To compute the kept n_{IR} global interface modes $\boldsymbol{Y}_I(\boldsymbol{\theta}^m) \in \mathbb{R}^{n_I \times n_{IR}}$ at each support point $\boldsymbol{\theta}^m$, the associated eigenvalue problems

$$\mathbf{K}_I(\boldsymbol{\theta}^m)\boldsymbol{Y}_I(\boldsymbol{\theta}^m) - \mathbf{M}_I(\boldsymbol{\theta}^m)\boldsymbol{Y}_I(\boldsymbol{\theta}^m)\boldsymbol{\Omega}_I(\boldsymbol{\theta}^m) = \mathbf{0}, \quad m = 1,\ldots,L \tag{5.3}$$

are solved and the matrix of interface modes is mass-normalized, satisfying

$$\boldsymbol{Y}_I^T(\boldsymbol{\theta}^m)\mathbf{M}_I(\boldsymbol{\theta}^m)\boldsymbol{Y}_I(\boldsymbol{\theta}^m) = \mathbf{I}_I, \quad m = 1,\ldots,L \tag{5.4}$$

and

$$\boldsymbol{Y}_I^T(\boldsymbol{\theta}^m)\mathbf{K}_I(\boldsymbol{\theta}^m)\boldsymbol{Y}_I(\boldsymbol{\theta}^m) = \boldsymbol{\Omega}_I(\boldsymbol{\theta}^m), \quad m = 1,\ldots,L \tag{5.5}$$

where $\mathbf{I}_I \in \mathbb{R}^{n_{IR} \times n_{IR}}$ is the identity matrix and $\boldsymbol{\Omega}_I(\boldsymbol{\theta}^m) \in \mathbb{R}^{n_{IR} \times n_{IR}}$ is the matrix containing the corresponding eigenvalues. Additionally, the kept global interface modes $\boldsymbol{Y}_I(\boldsymbol{\theta}^0)$ at the nominal point $\boldsymbol{\theta}^0$ are computed.

5.1.1.2 Approximation of Global Interface Modes at a sample point $\boldsymbol{\theta}^k$

A linear interpolation of the interface modes $\boldsymbol{Y}_I(\boldsymbol{\theta}^m)$ at each support point $\boldsymbol{\theta}^m, \quad m = 1,\ldots,L$ yields the matrix $\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)$ evaluated at a sample point $\boldsymbol{\theta}^k$ as [14]

$$\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k) = (1 - \sum_{m=1}^{L} \xi_m^k)\boldsymbol{Y}_I(\boldsymbol{\theta}^0) + \sum_{m=1}^{L} \xi_m^k \boldsymbol{Y}_I(\boldsymbol{\theta}^m) \tag{5.6}$$

where the coefficient ξ_m^k represents the contribution of the support point $\boldsymbol{\theta}^m$ to the simulation point $\boldsymbol{\theta}^k$. In order to consider only interpolations, the simulation point $\boldsymbol{\theta}^k$ should belong to the n_θ-dimensional convex hull of the support points.

The approximate interface modes $\boldsymbol{Y}_I(\boldsymbol{\theta}^k)$ are defined as a linear combination of the vectors in the matrix $\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)$, that is

$$\boldsymbol{Y}_I(\boldsymbol{\theta}^k) = \hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)\mathbf{Q}(\boldsymbol{\theta}^k) \tag{5.7}$$

where $\mathbf{Q}(\boldsymbol{\theta}^k) \in \mathbb{R}^{n_{IR} \times n_{IR}}$ is an auxiliary transformation matrix obtained from the solution of the reduced eigenproblem

$$\left[\hat{\boldsymbol{Y}}_I^T(\boldsymbol{\theta}^k)\mathbf{K}_I(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)\right]\mathbf{Q}(\boldsymbol{\theta}^k) = \left[\hat{\boldsymbol{Y}}_I^T(\boldsymbol{\theta}^k)\mathbf{M}_I(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)\right]\mathbf{Q}(\boldsymbol{\theta}^k)\boldsymbol{\Omega}_I(\boldsymbol{\theta}^k) \tag{5.8}$$

where the matrices $\hat{\boldsymbol{Y}}_I^T(\boldsymbol{\theta}^k)\mathbf{K}_I(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)$ and $\hat{\boldsymbol{Y}}_I^T(\boldsymbol{\theta}^k)\mathbf{M}_I(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_I(\boldsymbol{\theta}^k)$ are of dimension equal to $n_{IR} \times n_{IR}$ which means that the solution of the eigenproblem in Eq. (5.8) generally requires minimal computational effort. Note that the interface matrices $\mathbf{M}_I(\boldsymbol{\theta}^k)$ and $\mathbf{K}_I(\boldsymbol{\theta}^k)$ evaluated at the sample point $\boldsymbol{\theta}^k$ can be computed directly from Eqs. (5.1) and (5.2), respectively.

The solution of the reduced eigenproblem along with Eq. (5.7) provide an approximation of the global interface modes $\boldsymbol{Y}_I(\boldsymbol{\theta}^k)$ at the sample point $\boldsymbol{\theta}^k$. Additionally, the reduced eigenproblem gives an approximation of the corresponding eigenvalues $\boldsymbol{\Omega}_I(\boldsymbol{\theta}^k)$.

5.1.1.3 Determination of Interpolation Coefficients

As mentioned in the previous section, the interpolation coefficients $\xi_m^k, \quad m = 1,\ldots,L$ represent the contribution of the support points to the new sample point. To obtain them, the norm of the difference between the support points $\boldsymbol{\theta}^m, \quad m = 1,\ldots,L$ and the simulation point $\boldsymbol{\theta}^k$ is first minimized, that is

$$\text{Min}_{m=1,\ldots,L} \left\| \boldsymbol{\theta}^m - \boldsymbol{\theta}^k \right\| \tag{5.9}$$

and the nearest simulation point to $\boldsymbol{\theta}^k$ is denoted by $\boldsymbol{\theta}^q, \quad q \in \{1,\ldots,L\}$. The corresponding interpolation coefficient ξ_q^k is obtained by projecting $\boldsymbol{\theta}^k - \boldsymbol{\theta}^0$ onto $\boldsymbol{\theta}^q - \boldsymbol{\theta}^0$, which yields

$$\xi_q^k = \frac{(\boldsymbol{\theta}^k - \boldsymbol{\theta}^0)^T(\boldsymbol{\theta}^q - \boldsymbol{\theta}^0)}{\left\|(\boldsymbol{\theta}^q - \boldsymbol{\theta}^0)\right\|^2} \tag{5.10}$$

The remaining part of the vector, which is perpendicular to $\boldsymbol{\theta}^q - \boldsymbol{\theta}^0$, is given by

$$\mathbf{v}^k = (\boldsymbol{\theta}^k - \boldsymbol{\theta}^0) - \xi_q^k(\boldsymbol{\theta}^q - \boldsymbol{\theta}^0) \tag{5.11}$$

This vector is represented as a linear combination of the remaining support points $\boldsymbol{\theta}^m, \quad m = 1,\ldots,L, \quad m \neq q$ through

$$\mathbf{v}^k = \left[\boldsymbol{\theta}^1 - \boldsymbol{\theta}^0,\ldots,\boldsymbol{\theta}^{q-1} - \boldsymbol{\theta}^0,\boldsymbol{\theta}^{q+1} - \boldsymbol{\theta}^0,\ldots,\boldsymbol{\theta}^L - \boldsymbol{\theta}^0\right]\boldsymbol{\tau}^k \tag{5.12}$$

where $\boldsymbol{\tau}^k$ is given by

$$\boldsymbol{\tau}^k = \begin{Bmatrix} \tau_1^k \\ \vdots \\ \tau_{q-1}^k \\ \tau_{q+1}^k \\ \vdots \\ \tau_L^k \end{Bmatrix} \in \mathbb{R}^{L-1} \tag{5.13}$$

The (unknown) components of the vector $\boldsymbol{\tau}^k$ are obtained as the solution of Eq. (5.12) using the singular value decomposition (SVD) method which can be applied to cases of under- and over-determined systems of equations. The solution for the coefficients $\xi_m^k, \quad m = 1, \ldots, L$, kept in the vector $\boldsymbol{\xi}^k$, are obtained by considering ξ_q^k in Eq. (5.10) and $\boldsymbol{\tau}^k$ in Eq. (5.13). This gives

$$\boldsymbol{\xi}^k = \begin{Bmatrix} \tau_1^k \\ \vdots \\ \tau_{q-1}^k \\ \xi_q^k \\ \tau_{q+1}^k \\ \vdots \\ \tau_L^k \end{Bmatrix} \in \mathbb{R}^{L} \tag{5.14}$$

The interpolation scheme above guarantees that the approximation is exact in each support point. Additionally, the potential time-consuming step of calculating the interface modes is performed only once for the support points and the nominal point at the beginning of the simulation process. This means that the approximation of interface modes using this method generally requires minimal computational time at each sample point.

If one wishes to increase the accuracy of the interpolation scheme, more support points can be added or higher-order interpolations (quadratic, cubic etc.) can be considered.

5.1.1.4 Comments on the Use of Support Points

In this section some general comments and limitations are presented concerning the use of support points with parametrized models based on global-level interface reduction.

5.1.1.4.1 General Remarks

It has been stated that support points $\boldsymbol{\theta}^m, \quad m = 1, \ldots, L$ are distributed around the nominal point $\boldsymbol{\theta}^0$ of the model. The nominal point may correspond to the reference model of the structure, or it can be chosen as the mean value of the uncertain model parameters. For the support points, there is not a unique selection method and one can use different approaches to generate them.

For example, the support points can be generated by a number of sampling methods, such as random sampling, Latin Hypercube sampling, orthogonal sampling, etc. Also, adaptive schemes can be considered, where the nominal point and the support points are updated during the simulation process to improve convergence and maintain accuracy.

In this work emphasis is given on a sampling method based on n-dimensional simplices.

5.1.1.4.2 Limitations Concerning Support Points

Regardless of the technique selected for the generation of support points, it is necessary – as stated in section 5.1.1.2 – that each sample point $\boldsymbol{\theta}^k$ lies in the convex hull of the support points to ensure that only interpolations are made.

If a sample point $\tilde{\boldsymbol{\theta}}^k$ does not belong to the convex hull of the $\boldsymbol{\theta}^m$, $m = 1,\ldots,L$ support points, the interface modes $\boldsymbol{Y}_I(\tilde{\boldsymbol{\theta}}^k)$ are computed directly at that sample point. Next, the support points are updated to contain the sample point $\tilde{\boldsymbol{\theta}}^k$ and the interface modes $\boldsymbol{Y}_I(\tilde{\boldsymbol{\theta}}^k)$ are added to the set of interface modes for the already defined L support points. This procedure increases the number of support points to $L+1$ and expands their convex hull so that the sample point $\tilde{\boldsymbol{\theta}}^k$ lies (marginally) inside it. An increase in the number of support points translates to an increase in the time required to perform a single interpolation of interface modes at a sample point.

The number of model parameters n_θ corresponds to the number of dimensions of the model parameter space. As n_θ increases, the volume (around the nominal point) that the support points have to sample increases exponentially as a result of the "curse of dimensionality".

Thus, as the number of model parameters n_θ increases, more support points are needed to create a sufficiently large convex hull so that few out-of-hull sample points occur. This approach can quickly increase the time required for the approximation of interface modes which is directly related to the number of support points used.

Alternatively, few (but enough to create a n_θ-dimensional simplex) and highly scattered support points around the nominal point could be generated. This strategy can result in poor approximations since the support points might be too far away from the nominal point to provide any meaningful accuracy.

The previous remarks make it clear that parametrization using global-level interface reduction might not be practical for models utilizing many uncertain parameters. For such a case, local-level reduction is a good alternative and is discussed in following sections.

5.1.2 Parametrization of Reduced-Order Matrices Based on Dominant Fixed-Interface Modes and Global-Level Interface Reduction

This section presents the parametrization of the transformation matrix and the reduced-order mass and stiffness matrices based on dominant modes and global-level interface reduction. It is assumed that the matrix $\boldsymbol{Y}_I(\boldsymbol{\theta})$ containing the global interface modes and $\boldsymbol{\Omega}_I(\boldsymbol{\theta})$ containing the corresponding eigenvalues are readily available. They can be approximated using the procedure described in the previous section or they can be computed directly through an interface analysis.

5.1.2.1 Transformation Matrix $\mathbf{T}_{DI}(\boldsymbol{\theta})$

The transformation matrix $\mathbf{T}_{DI}$ which takes into account the effect of the dominant fixed-interface normal modes and global-level interface reduction is defined in Eq. (3.9). In the context of parametrized reduced-order models it is defined similarly to Eq. (4.17) as

$$\begin{aligned}\mathbf{T}_{DI}(\boldsymbol{\theta}) = &\begin{bmatrix} \left[\bar{\boldsymbol{\Phi}}_{id}^{1}\delta_{10},\ldots,\bar{\boldsymbol{\Phi}}_{id}^{N_s}\delta_{N_s 0}\right] & \left[\bar{\boldsymbol{\Psi}}_{ib}^{1},\ldots,\bar{\boldsymbol{\Psi}}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_I(\boldsymbol{\theta}) \\ \mathbf{0} & \boldsymbol{Y}_I(\boldsymbol{\theta}) \end{bmatrix} \\ &+\sum_{j=1}^{n_\theta}\begin{bmatrix} \left[\bar{\boldsymbol{\Phi}}_{id}^{1}\delta_{1j},\ldots,\bar{\boldsymbol{\Phi}}_{id}^{N_s}\delta_{N_s j}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\frac{1}{\sqrt{g^j(\theta_j)}}\end{aligned} \tag{5.15}$$

5.1.2.2 Reduced-Order Matrices $\hat{\mathbf{M}}_{DI}(\boldsymbol{\theta})$ *and* $\hat{\mathbf{K}}_{DI}(\boldsymbol{\theta})$

The reduced-order mass and stiffness matrices based on dominant modes and global-level reduction are defined in Eqs. (3.10) and (3.11), respectively. They are parametrized similarly to Eqs. (4.20) and (4.21) as

$$\begin{aligned}\hat{\mathbf{M}}_{DI}(\boldsymbol{\theta}) = &\begin{bmatrix} \mathbf{I} & \left[\hat{\bar{\mathbf{M}}}_{ib}^{1}\delta_{10},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s}\delta_{N_s 0}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_I(\boldsymbol{\theta}) \\ \boldsymbol{Y}_I^T(\boldsymbol{\theta})\tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{M}}}_{ib}^{1^T}\delta_{10},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s^T}\delta_{N_s 0}\right] & \mathbf{I} \end{bmatrix} \\ +\sum_{j=1}^{n_\theta}&\begin{bmatrix} \mathbf{0} & \left[\hat{\bar{\mathbf{M}}}_{ib}^{1}\delta_{1j},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s}\delta_{N_s j}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_I(\boldsymbol{\theta}) \\ \boldsymbol{Y}_I^T(\boldsymbol{\theta})\tilde{\mathbf{T}}^T\left[\hat{\bar{\mathbf{M}}}_{ib}^{1^T}\delta_{1j},\ldots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s^T}\delta_{N_s j}\right] & \mathbf{0} \end{bmatrix}\sqrt{g^j(\theta_j)}\end{aligned} \tag{5.16}$$

and

$$\hat{\mathbf{K}}_{DI}(\boldsymbol{\theta}) = \begin{bmatrix} \left[\bar{\boldsymbol{\Lambda}}_{id}^{1}\delta_{10},\ldots,\bar{\boldsymbol{\Lambda}}_{id}^{N_s}\delta_{N_s 0}\right] & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_I(\boldsymbol{\theta}) \end{bmatrix} + \sum_{j=1}^{n_\theta} \begin{bmatrix} \left[\bar{\boldsymbol{\Lambda}}_{id}^{1}\delta_{1j},\ldots,\bar{\boldsymbol{\Lambda}}_{id}^{N_s}\delta_{N_s j}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{h^j(\theta_j)}{g^j(\theta_j)} \tag{5.17}$$

5.1.3 Comments on the Proposed Parametrization Scheme

From Eqs. (5.15), (5.16) and (5.17) it is clear that the matrices $\mathbf{T}_{DI}$, $\hat{\mathbf{M}}_{DI}$ and $\hat{\mathbf{K}}_{DI}$, respectively, can be computed directly at a sample point $\boldsymbol{\theta}^k$ in terms of some constant matrices (given that $\boldsymbol{Y}_I(\boldsymbol{\theta})$ and $\boldsymbol{\Omega}_I(\boldsymbol{\theta})$ are already defined) and the functions $g^j(\theta_j)$ and $h^j(\theta_j)$ at each component j of the sample point. Therefore, the time required to calculate these matrices in each step of the simulation process should be minimal – as was the case for the parametrization scheme without interface reduction presented in section 4.

Although efficient for a small to moderate number of model parameters n_θ, this parametrization scheme can be impractical when n_θ is large due to the large number of support points required to approximate interface modes at a sample point $\boldsymbol{\theta}^k$ (see section 5.1.1.4.2). To overcome this problem, another parametrization scheme based on local-level interface reduction is presented in the next sections.

5.2 Parametrization Based on Local-Level Interface Reduction

This section introduces a parametrization scheme based on local-level interface reduction. The formulations here are novel. They are motivated by section 3.5 of [3] and further developed and implemented by me.

5.2.1 Meta-Model for Local Interface Modes

Local interface modes need to be approximated at each sample point similarly to global modes in the previous sections. The main difference is that the modes for every independent interface are interpolated separately in contrast to the global modes which treat all interface DOF as a single interface.

Throughout the following sections it is assumed that the interfaces are defined according to guidelines presented in section 3.2.1 concerning local-level interface reduction.

5.2.1.1 Baseline Information

Initially, L support points are defined in the model parameter space ($\boldsymbol{\theta}^m, \quad m=1,\ldots,L$) and the interface matrices $\mathbf{K}_I(\boldsymbol{\theta}^m)$ and $\mathbf{M}_I(\boldsymbol{\theta}^m)$ are assembled at these points using Eqs. (5.2) and (5.1), respectively. It is assumed that the support points are distributed around the nominal point $\boldsymbol{\theta}^0$.

Let $\mathbf{K}_{Ill}(\boldsymbol{\theta}^m) \in \mathbb{R}^{n_I^l \times n_I^l}$ and $\mathbf{M}_{Ill}(\boldsymbol{\theta}^m) \in \mathbb{R}^{n_I^l \times n_I^l}$ be the partitions of $\mathbf{K}_I(\boldsymbol{\theta}^m)$ and $\mathbf{M}_I(\boldsymbol{\theta}^m)$, respectively, at support point $\boldsymbol{\theta}^m, \quad m=1,\ldots,L$ associated with the physical coordinates at interface $l, \quad l=1,\ldots,N_I$, i.e. $\mathbf{u}_I^l(t)$.

To compute the kept n_{IR}^l local interface modes $\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m) \in \mathbb{R}^{n_I^l \times n_{IR}^l}$ at each support point $\boldsymbol{\theta}^m$ and interface l, the associated eigenvalue problems

$$\mathbf{K}_{Ill}(\boldsymbol{\theta}^m)\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m) - \mathbf{M}_{Ill}(\boldsymbol{\theta}^m)\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m)\boldsymbol{\Omega}_{Ill}(\boldsymbol{\theta}^m) = \mathbf{0}, \quad m=1,\ldots,L \text{ and } l=1,\ldots,N_I \tag{5.18}$$

are solved and the matrix of local interface modes is mass-normalized, satisfying

$$\boldsymbol{Y}_{Ill}^T(\boldsymbol{\theta}^m)\mathbf{M}_{Ill}(\boldsymbol{\theta}^m)\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m) = \mathbf{I}_{Ill}, \quad m=1,\ldots,L \text{ and } l=1,\ldots,N_I \tag{5.19}$$

and

$$\boldsymbol{Y}_{Ill}^T(\boldsymbol{\theta}^m)\mathbf{K}_{Ill}(\boldsymbol{\theta}^m)\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m) = \boldsymbol{\Omega}_{Ill}(\boldsymbol{\theta}^m), \quad m=1,\ldots,L \text{ and } l=1,\ldots,N_I \tag{5.20}$$

where $\mathbf{I}_{Ill} \in \mathbb{R}^{n_{IR}^l \times n_{IR}^l}$ is the identity matrix and $\boldsymbol{\Omega}_{Ill}(\boldsymbol{\theta}^m) \in \mathbb{R}^{n_{IR}^l \times n_{IR}^l}$ is the matrix containing the corresponding eigenvalues. Additionally, the kept local interface modes $\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^0) \in \mathbb{R}^{n_I^l \times n_{IR}^l}$ at the nominal point $\boldsymbol{\theta}^0$ for each interface $l, \quad l=1,\ldots,N_I$ are computed.

5.2.1.2 Approximation of Local Interface Modes at a sample point $\boldsymbol{\theta}^k$

A linear interpolation of the interface modes $\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m)$ for a given interface l at each support point $\boldsymbol{\theta}^m, \quad m=1,\ldots,L$ yields the matrix $\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)$ for the given interface evaluated at a sample point $\boldsymbol{\theta}^k$ as

$$\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k) = (1 - \sum_{m=1}^{L} \xi_{m,l}^k)\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^0) + \sum_{m=1}^{L} \xi_{m,l}^k \boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^m), \quad l=1,\ldots,N_I \tag{5.21}$$

where the coefficient $\xi_{m,l}^k$ represents the contribution of the support point $\boldsymbol{\theta}^m$ to the simulation point $\boldsymbol{\theta}^k$ for the interface l.

In general, the number of parameters related to a given interface l, denoted as $n_{\theta,l}$, is much smaller than the total number of model parameters n_θ. Consequently, to consider only interpolations for the given interface l, only the $n_{\theta,l}$ components of $\boldsymbol{\theta}^k$ associated with that interface must lie in the $n_{\theta,l}$-dimensional convex hull of the corresponding components of the support points. Section 5.2.1.3 illustrates how to identify the parameters related to each interface.

The approximate interface modes $\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^k)$ at a sample point $\boldsymbol{\theta}^k$ for a given interface l are defined as a linear combination of the vectors in the matrix $\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)$, that is

$$\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^k) = \hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)\mathbf{Q}_{ll}(\boldsymbol{\theta}^k), \quad l = 1,\dots,N_I \tag{5.22}$$

where $\mathbf{Q}_{ll}(\boldsymbol{\theta}^k) \in \mathbb{R}^{n_{IR}^l \times n_{IR}^l}$ is an auxiliary transformation matrix corresponding to interface l obtained from the solution of the reduced eigenproblem

$$\begin{aligned}&\left[\hat{\boldsymbol{Y}}_{Ill}^T(\boldsymbol{\theta}^k)\mathbf{K}_{Ill}(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)\right]\mathbf{Q}_{ll}(\boldsymbol{\theta}^k) = \\ &\left[\hat{\boldsymbol{Y}}_{Ill}^T(\boldsymbol{\theta}^k)\mathbf{M}_{Ill}(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)\right]\mathbf{Q}_{ll}(\boldsymbol{\theta}^k)\boldsymbol{\Omega}_{Ill}(\boldsymbol{\theta}^k), \quad l = 1,\dots,N_I\end{aligned} \tag{5.23}$$

where the matrices $\hat{\boldsymbol{Y}}_{Ill}^T(\boldsymbol{\theta}^k)\mathbf{K}_{Ill}(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)$ and $\hat{\boldsymbol{Y}}_{Ill}^T(\boldsymbol{\theta}^k)\mathbf{M}_{Ill}(\boldsymbol{\theta}^k)\hat{\boldsymbol{Y}}_{Ill}(\boldsymbol{\theta}^k)$ are of dimension equal to $n_{IR}^l \times n_{IR}^l$.

Note that the interface matrices $\mathbf{M}_{Ill}(\boldsymbol{\theta}^k)$ and $\mathbf{K}_{Ill}(\boldsymbol{\theta}^k)$ for a given interface l evaluated at the sample point $\boldsymbol{\theta}^k$ are the partitions of $\mathbf{M}_I(\boldsymbol{\theta}^k)$ and $\mathbf{K}_I(\boldsymbol{\theta}^k)$, respectively, associated with the physical coordinates at interface l. The matrices $\mathbf{M}_I(\boldsymbol{\theta}^k)$ and $\mathbf{K}_I(\boldsymbol{\theta}^k)$ can be computed directly from Eqs. (5.1) and (5.2), respectively.

The approximated kept local interface modes $\boldsymbol{Y}_{Ill}(\boldsymbol{\theta}^k)$ at the sample point $\boldsymbol{\theta}^k$ for each interface $l, \quad l = 1,\dots,N_I$ – defined in Eq. (5.22) – are used to construct the matrix $\boldsymbol{Y}_{IL}(\boldsymbol{\theta}^k)$ as

$$\boldsymbol{Y}_{IL}(\boldsymbol{\theta}^k) = \left[\boldsymbol{Y}_{I11}(\boldsymbol{\theta}^k),\dots,\boldsymbol{Y}_{IN_IN_I}(\boldsymbol{\theta}^k)\right] \in \mathbb{R}^{n_I \times n_{IRL}} \tag{5.24}$$

The matrix $\boldsymbol{Y}_{IL}(\boldsymbol{\theta}^k)$ takes into account the kept modes of all independent interfaces and is similar to the matrix $\boldsymbol{Y}_I(\boldsymbol{\theta}^k)$ based on global-level interface reduction defined in Eq. (5.7).

The solution of the reduced eigenproblem in Eq. (5.23) along with Eq. (5.22) for each interface $l, \quad l = 1,\dots,N_I$ provide all the terms needed to define the matrix $\boldsymbol{Y}_{IL}(\boldsymbol{\theta}^k)$ in Eq. (5.24) which is used to approximate the local interface modes at the sample point $\boldsymbol{\theta}^k$.

5.2.1.3 Determination of Model Parameters Related to Each Interface

The basic difference between the interpolation method presented here and the one for global-level reduction in section 5.1.1.3, is that in this case only the $n_{\theta,l}$ model parameters associated with a given interface l are taken into account in the determination of interpolation coefficients for that interface.

The model parameters related to a given interface $l, \quad l=1,\ldots,N_I$ are those that the substructures connected to that interface depend on. They are used to define an index set p_l for each interface l that holds the $n_{\theta,l}$ associated model parameters. These index sets can be defined in set notation as

$$p_l = \left\{ j \in \{1,\ldots,n_\theta\} \mid \text{substructures connected to interface } l \text{ depend on parameter } j \right\}, \quad l=1,\ldots,N_I \tag{5.25}$$

They are used as subscripts of vectors in the following formulations denoting that only the vector elements associated with the parameters contained in the set are considered.

5.2.1.4 Determination of Interpolation Coefficients

As already mentioned, the interpolation coefficients $\xi^k_{m,l}, \quad m=1,\ldots,L \text{ and } l=1,\ldots,N_I$ represent the contribution of the support points to the new sample point for a given interface l.

To obtain the interpolation coefficients for a given interface l, the norm of the difference between the support points $\boldsymbol{\theta}^m_{p_l}, \quad m=1,\ldots,L$ and the simulation point $\boldsymbol{\theta}^m_{p_l}$ is first minimized, that is

$$\text{Min}_{m=1,\ldots,L} \left\| \boldsymbol{\theta}^m_{p_l} - \boldsymbol{\theta}^k_{p_l} \right\|, \quad l=1,\ldots,N_I \tag{5.26}$$

and the nearest simulation point to $\boldsymbol{\theta}^k_{p_l}$ is denoted by $\boldsymbol{\theta}^q_{p_l}, \quad q \in \{1,\ldots,L\}$. The corresponding interpolation coefficient $\xi^k_{q,l}$ is obtained by projecting $\boldsymbol{\theta}^k_{p_l} - \boldsymbol{\theta}^0_{p_l}$ onto $\boldsymbol{\theta}^q_{p_l} - \boldsymbol{\theta}^0_{p_l}$, which yields

$$\xi^k_{q,l} = \frac{(\boldsymbol{\theta}^k_{p_l} - \boldsymbol{\theta}^0_{p_l})^T (\boldsymbol{\theta}^q_{p_l} - \boldsymbol{\theta}^0_{p_l})}{\left\| (\boldsymbol{\theta}^q_{p_l} - \boldsymbol{\theta}^0_{p_l}) \right\|^2}, \quad l=1,\ldots,N_I \tag{5.27}$$

The remaining part of the vector, which is perpendicular to $\boldsymbol{\theta}^q_{p_l} - \boldsymbol{\theta}^0_{p_l}$, is given by

$$\mathbf{v}^k_l = (\boldsymbol{\theta}^k_{p_l} - \boldsymbol{\theta}^0_{p_l}) - \xi^k_{q,l}(\boldsymbol{\theta}^q_{p_l} - \boldsymbol{\theta}^0_{p_l}), \quad l=1,\ldots,N_I \tag{5.28}$$

This vector is represented as a linear combination of the remaining support points $\boldsymbol{\theta}^m_{p_l}, \quad m=1,\ldots,L, \quad m \neq q$ through

$$\mathbf{v}^k_l = \left[\boldsymbol{\theta}^1_{p_l} - \boldsymbol{\theta}^0_{p_l}, \ldots, \boldsymbol{\theta}^{q-1}_{p_l} - \boldsymbol{\theta}^0_{p_l}, \boldsymbol{\theta}^{q+1}_{p_l} - \boldsymbol{\theta}^0_{p_l}, \ldots, \boldsymbol{\theta}^L_{p_l} - \boldsymbol{\theta}^0_{p_l} \right] \boldsymbol{\tau}^k_l, \quad l=1,\ldots,N_I \tag{5.29}$$

where $\boldsymbol{\tau}^k_l$ is given by

$$\boldsymbol{\tau}_l^k = \begin{Bmatrix} \tau_{1,l}^k \\ \vdots \\ \tau_{q-1,l}^k \\ \tau_{q+1,l}^k \\ \vdots \\ \tau_{L,l}^k \end{Bmatrix} \in \mathbb{R}^{L-1}, \quad l = 1,\ldots,N_I \tag{5.30}$$

The (unknown) components of the vector $\boldsymbol{\tau}_l^k$ are obtained as the solution of Eq. (5.29) using the singular value decomposition technique. The interpolation coefficients $\xi_{m,l}^k, \quad m = 1,\ldots,L$ for the given interface l are kept in the vector $\boldsymbol{\xi}_l^k$ and are obtained by considering $\xi_{q,l}^k$ in Eq. (5.27) and $\boldsymbol{\tau}_l^k$ in Eq. (5.30), that is

$$\boldsymbol{\xi}_l^k = \begin{Bmatrix} \tau_{1,l}^k \\ \vdots \\ \tau_{q-1,l}^k \\ \xi_{q,l}^k \\ \tau_{q+1,l}^k \\ \vdots \\ \tau_{L,l}^k \end{Bmatrix} \in \mathbb{R}^{L}, \quad l = 1,\ldots,N_I \tag{5.31}$$

If an interface $\tilde{l}$ connects to substructures that do not depend on model parameters, that is $p_{\tilde{l}} = \varnothing$, the vectors associated with the above formulations become empty and the equations are ill-defined. In that case, the vector of interpolation coefficients for that interface is set to zero $\boldsymbol{\xi}_{\tilde{l}}^k = \langle 0,\ldots,0\rangle^T \in \mathbb{R}^L$. This is to ensure that Eq. (5.21) yields $\hat{\boldsymbol{Y}}_{I\tilde{l}}(\boldsymbol{\theta}^k) = \boldsymbol{Y}_{I\tilde{l}}(\boldsymbol{\theta}^0)$ which is exact since the modes of interface $\tilde{l}$ are constant and equal to those calculated at the nominal point in each sample point $\boldsymbol{\theta}^k$.

The same comments as in section 5.1.1.3 apply concerning the efficiency of the interpolation scheme and potential methods to increase its accuracy.

5.2.2 Parametrization of Reduced-Order Matrices Based on Dominant Fixed-Interface Modes and Local-Level Interface Reduction

This section presents the parameterized transformation matrix and reduced-order mass and stiffness matrices based on dominant fixed-interface normal modes and local-level interface reduction. It is assumed that the matrix $\boldsymbol{Y}_{IL}(\boldsymbol{\theta})$ containing the local interface modes is already available. It can be approximated using the proposed interpolation scheme in section 5.2.1 or it can be calculated directly at each sample point.

5.2.2.1 Transformation Matrix $\mathbf{T}_{DIL}(\boldsymbol{\theta})$

The transformation matrix $\mathbf{T}_{DIL}$ considers the effect of dominant fixed-interface normal modes and local-level interface reduction and is defined in Eq. (3.25). In view of the parametrization of normal modes in Eq. (4.7) and the definition of matrix $\boldsymbol{Y}_{IL}(\boldsymbol{\theta})$ in Eq. (5.24), the transformation matrix $\mathbf{T}_{DIL}(\boldsymbol{\theta})$ can be written similarly to $\mathbf{T}_{DI}(\boldsymbol{\theta})$ in Eq. (5.15) as

$$\mathbf{T}_{DIL}(\boldsymbol{\theta}) = \begin{bmatrix} \left[\bar{\boldsymbol{\Phi}}_{id}^{1}\delta_{10},\dots,\bar{\boldsymbol{\Phi}}_{id}^{N_s}\delta_{N_s0}\right] & \left[\bar{\boldsymbol{\Psi}}_{ib}^{1},\dots,\bar{\boldsymbol{\Psi}}_{ib}^{N_s}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \\ \mathbf{0} & \boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \end{bmatrix} + \sum_{j=1}^{n_\theta} \begin{bmatrix} \left[\bar{\boldsymbol{\Phi}}_{id}^{1}\delta_{1j},\dots,\bar{\boldsymbol{\Phi}}_{id}^{N_s}\delta_{N_sj}\right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{1}{\sqrt{g^j(\theta_j)}} \tag{5.32}$$

5.2.2.2 Reduced-Order Matrices $\hat{\mathbf{M}}_{DIL}(\boldsymbol{\theta})$ and $\hat{\mathbf{K}}_{DIL}(\boldsymbol{\theta})$

The reduced-order mass matrix $\hat{\mathbf{M}}_{DIL}$ and stiffness matrix $\hat{\mathbf{K}}_{DIL}$ that take into account dominant fixed-interface normal modes and interface reduction at the local level are defined in Eqs. (3.26) and (3.27), respectively. Considering the parametrizations defined in Eqs. (4.18), (4.8) and (5.24), they can be written as

$$\begin{aligned} \hat{\mathbf{M}}_{DIL}(\boldsymbol{\theta}) = & \begin{bmatrix} \mathbf{I} & \left[\hat{\bar{\mathbf{M}}}_{ib}^{1}\delta_{10},\dots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \\ \boldsymbol{Y}_{IL}^{T}(\boldsymbol{\theta})\tilde{\mathbf{T}}^{T}\left[\hat{\bar{\mathbf{M}}}_{ib}^{1^T}\delta_{10},\dots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s^T}\delta_{N_s0}\right] & \boldsymbol{Y}_{IL}^{T}(\boldsymbol{\theta})\tilde{\mathbf{T}}^{T}\left[\hat{\bar{\mathbf{M}}}_{bb}^{1}\delta_{10},\dots,\hat{\bar{\mathbf{M}}}_{bb}^{N_s}\delta_{N_s0}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \end{bmatrix} \\ & + \sum_{j=1}^{n_\theta}\left\{ \begin{bmatrix} \mathbf{0} & \left[\hat{\bar{\mathbf{M}}}_{ib}^{1}\delta_{1j},\dots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \\ \boldsymbol{Y}_{IL}^{T}(\boldsymbol{\theta})\tilde{\mathbf{T}}^{T}\left[\hat{\bar{\mathbf{M}}}_{ib}^{1^T}\delta_{1j},\dots,\hat{\bar{\mathbf{M}}}_{ib}^{N_s^T}\delta_{N_sj}\right] & \mathbf{0} \end{bmatrix}\sqrt{g^j(\theta_j)} \right. \\ & \left. + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_{IL}^{T}(\boldsymbol{\theta})\tilde{\mathbf{T}}^{T}\left[\hat{\bar{\mathbf{M}}}_{bb}^{1}\delta_{1j},\dots,\hat{\bar{\mathbf{M}}}_{bb}^{N_s}\delta_{N_sj}\right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \end{bmatrix} g^j(\theta_j) \right\} \end{aligned} \tag{5.33}$$

and

$$\hat{\mathbf{K}}_{DIL}(\boldsymbol{\theta}) = \begin{bmatrix} \left[\overline{\boldsymbol{\Lambda}}_{id}^{1}\delta_{10}, \ldots, \overline{\boldsymbol{\Lambda}}_{id}^{N_s}\delta_{N_s 0} \right] & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_{IL}^{T}(\boldsymbol{\theta})\tilde{\mathbf{T}}^{T}\left[\hat{\hat{\mathbf{K}}}_{bb}^{1}\delta_{10}, \ldots, \hat{\hat{\mathbf{K}}}_{bb}^{N_s}\delta_{N_s 0} \right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \end{bmatrix}$$
$$+\sum_{j=1}^{n_\theta}\left\{ \begin{bmatrix} \left[\overline{\boldsymbol{\Lambda}}_{id}^{1}\delta_{1j}, \ldots, \overline{\boldsymbol{\Lambda}}_{id}^{N_s}\delta_{N_s j} \right] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \frac{h^j(\theta_j)}{g^j(\theta_j)} \right. \tag{5.34}$$
$$\left. + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Y}_{IL}^{T}(\boldsymbol{\theta})\tilde{\mathbf{T}}^{T}\left[\hat{\hat{\mathbf{K}}}_{bb}^{1}\delta_{1j}, \ldots, \hat{\hat{\mathbf{K}}}_{bb}^{N_s}\delta_{N_s j} \right]\tilde{\mathbf{T}}\boldsymbol{Y}_{IL}(\boldsymbol{\theta}) \end{bmatrix} h^j(\theta_j) \right\}$$

5.2.3 Comments on the Proposed Parametrization Scheme

This parametrization scheme is similar to that presented in section 5.1 with the basic difference that interface reduction is performed at the local level. The desirable property resulting from local-level reduction is that only the model parameters associated with each interface are considered during the approximation of its modes (see section 5.2.1.3).

Since the number of related parameters for a given interface is generally much smaller than the total number of model parameters, far fewer support points are needed in this scheme compared to the one considering global-level reduction.

This point is made clear in section 5.4.

5.3 Support Points Based on n-dimensional Simplices

Both parametrization methods based on interface reduction require that (all or some of the elements of) each sample point $\boldsymbol{\theta}^k \in \mathbb{R}^{n_\theta}$ lie in the convex hull of the support points used to approximate the interface modes (global or local) in each iteration. Since using as few support points as possible is of prime interest, it is natural to consider generating support points based on n-dimensional simplices where $n \leq n_\theta$.

The formulations presented here are, to my knowledge, novel.

Initially, a regular simplex in $\mathbb{R}^n$ is created with the following properties: 1) it is inscribed in a unit hypersphere (the distance of each vertex from the centroid is one) and 2) its centroid is $\mathbf{0}$. The $\mathbf{t}_i \in \mathbb{R}^n, \quad i = 1, \ldots, n+1$ vertices of the simplex are given by [19]

$$\mathbf{t}_i = \sqrt{1+n^{-1}}\mathbf{e}_i - n^{-3/2}(\sqrt{n+1}-1)\langle 1, \ldots, 1\rangle^T, \quad i = 1, \ldots, n \tag{5.35}$$

and

$$\mathbf{t}_{n+1} = -n^{-1/2}\langle 1, \ldots, 1\rangle^T \tag{5.36}$$

where $\mathbf{e}_i$ is the unit vector in direction i. The corresponding support points $\boldsymbol{\theta}^m, \quad m=1,\ldots,n+1$ are centered on $\boldsymbol{\theta}^0$ and are scaled according to the scaling matrix $\mathbf{A}$, that is

$$\boldsymbol{\theta}^m = \mathbf{A}\mathbf{t}_m + \boldsymbol{\theta}^0, \quad m=1,\ldots,n+1 \tag{5.37}$$

where

$$\mathbf{A} = \left[a_1,\ldots,a_n\right] \tag{5.38}$$

is a diagonal matrix that multiplies the j element of $\mathbf{t}_m$ by a_j.

This method is guaranteed to give the minimum number of support points needed in $\mathbb{R}^n$ ($L=n+1$) since fewer support points would fail to create a n-dimensional simplex. It is not guaranteed, however, that the given support points will create a sufficiently large convex hull for each sample point $\boldsymbol{\theta}^k$ to lie in – as will be shown next.

5.4 Generation of Support Points for Parametrization Based on Global- and Local-Level Interface Reduction

This section illustrates the fact that parametrization based on local-level interface reduction generally requires less support points than that based on global-level interface reduction.

Initially, a simple multi-parameter model is introduced, and support points based on n-dimensional simplices are generated for both interface reduction methods.

Next, random sample points are generated, and - for each method and its corresponding support points - the frequency of interpolations is measured.

The interpolation frequency indicates the ability of each set of support points to create a sufficiently large convex hull.

5.4.1 Multi-Parameter Model

A simple multi-parameter model used to illustrate the above point is shown in **Fig. 5.1**.

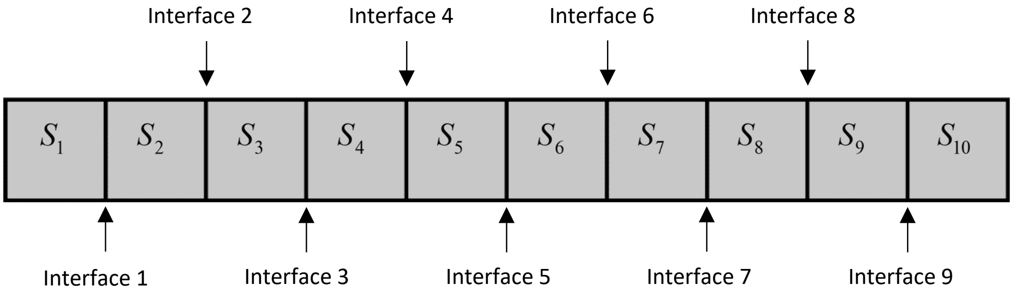

Fig. 5.1 Simple multi-parameter model with 10 components and 9 interfaces.

It is a rectangle divided into ten components with each component dependent on one model parameter. This model has ten parameters ($n_\theta = 10$), nine interfaces ($N_I = 9$) and the nominal point $\boldsymbol{\theta}^0$ is chosen at

$$\boldsymbol{\theta}^0 = \langle 1,\ldots,1 \rangle^T \in \mathbb{R}^{10} \tag{5.39}$$

5.4.2 Support Points Required for Global-Level Reduction

Parametrization based on dominant normal modes and global-level interface reduction requires that each sample point $\boldsymbol{\theta}^k \in \mathbb{R}^{n_\theta}$ lie in the convex hull of the L support points (see section 5.1.1.2). As such, the minimum number of support points is

$$L = n_\theta + 1 = 10 + 1 = 11 \tag{5.40}$$

which is equal to the number of vertices needed to create a 10-dimensional simplex.

Considering Eq. (5.37) with $\mathbf{A} = a\mathbf{I}$ and Eq. (5.39), the support points are

$$\boldsymbol{\theta}^m = a\mathbf{t}_m + \langle 1,\ldots,1 \rangle^T, \quad m = 1,\ldots,11 \tag{5.41}$$

where $\mathbf{t}_m, \quad m = 1,\ldots,11$ are the vertices of the 10-dimensional simplex defined in Eqs. (5.35) and (5.36) with $n = 10$. The parameter a scales the support points symmetrically around $\boldsymbol{\theta}^0$. As a increases, the convex hull of the support points becomes larger and more sample points are expected to lie in it.

For reference, setting $a = 1$ yields the following support points

$$\boldsymbol{\theta}^1 = \begin{bmatrix} 1.9756 \\ 0.9267 \\ 0.9267 \\ 0.9267 \\ \vdots \\ 0.9267 \end{bmatrix}, \quad \boldsymbol{\theta}^2 = \begin{bmatrix} 0.9267 \\ 1.9756 \\ 0.9267 \\ 0.9267 \\ \vdots \\ 0.9267 \end{bmatrix}, \quad \boldsymbol{\theta}^3 = \begin{bmatrix} 0.9267 \\ 0.9267 \\ 1.9756 \\ 0.9267 \\ \vdots \\ 0.9267 \end{bmatrix}, \ldots, \quad \boldsymbol{\theta}^{10} = \begin{bmatrix} 0.9267 \\ 0.9267 \\ 0.9267 \\ 0.9267 \\ \vdots \\ 1.9756 \end{bmatrix}, \quad \boldsymbol{\theta}^{11} = \begin{bmatrix} 0.6838 \\ 0.6838 \\ 0.6838 \\ 0.6838 \\ \vdots \\ 0.6838 \end{bmatrix}$$

5.4.3 Support Points Required for Local-Level Reduction

Parametrization based on local-level interface reduction requires that, for a given interface l $n_{\theta,l}$ model parameters must lie in the convex hull of the support points (see section 5.2.1.3). This property reduces the dimension of the problem from n_θ to $n_{\theta,l}$ for each interface $l, \quad l = 1, \ldots, N_I$. The steps taken to generate support points for the given model are illustrated below.

Interface 1

Interface 1 connects to components 1 and 2 that depend on parameters 1 and 2, respectively. Thus, interface 1 associates with parameters 1 and 2 which means that index set p_1 (see Eq. (5.25)) is defined as

$$p_1 = \{1, 2\} \tag{5.42}$$

and the number of associated parameters for interface 1 is

$$n_{\theta,1} = 2 \tag{5.43}$$

From interface 1, it is required that the elements of the support points defined by $p_1 = \{1, 2\}$ (rows 1 and 2) create a simplex of minimum dimension $n_{\theta,1} = 2$. So at least $n_{\theta,1} + 1 = 2 + 1 = 3$ support points are required for interface 1. The elements in rows 1 and 2 are generated based on a simplex in $\mathbb{R}^{n_{\theta,1}} = \mathbb{R}^2$. Considering Eq. (5.37) with $\mathbf{A} = a\mathbf{I}$ and Eq. (5.39), the elements in rows 1 and 2 of the support points are

$$\boldsymbol{\theta}_{p_1}^m = a\mathbf{t}_m + \langle 1, 1 \rangle^T, \quad m = 1, 2, 3 \tag{5.44}$$

where $\mathbf{t}_m, \quad m = 1, 2, 3$ are the vertices of the 2-dimensional simplex defined in Eqs. (5.35) and (5.36) with $n = 2$. The parameter a scales the elements in rows 1 and 2 of the support points symmetrically around $\boldsymbol{\theta}_{p_1}^0$.

For reference, setting $a = 1$ yields the following elements in rows 1 and 2 of the support points

$$\boldsymbol{\theta}^1 = \begin{bmatrix} 1.9659 \\ 0.7412 \\ ? \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^2 = \begin{bmatrix} 0.7412 \\ 1.9659 \\ ? \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^3 = \begin{bmatrix} 0.2929 \\ 0.2929 \\ ? \\ ? \\ \vdots \\ ? \end{bmatrix} \tag{5.45}$$

where question marks denote that information from more interfaces is needed to determine the value of the corresponding elements.

Interface 2

Interface 2 connects to components 2 and 3 that depend on parameters 2 and 3, respectively. Similarly to interface 1, for interface 2 the following hold

$$p_2 = \{2,3\} \tag{5.46}$$

and

$$n_{\theta,2} = 2 \tag{5.47}$$

Following the same process as before, the elements in rows 2 and 3 of the support points are defined as

$$\boldsymbol{\theta}_{p_2}^{m} = a\mathbf{t}_m + \langle 1,1 \rangle^T , \quad m = 1,2,3 \tag{5.48}$$

where $\mathbf{t}_m, \quad m = 1,2,3$ are the vertices of the 2-dimensional simplex defined in Eqs. (5.35) and (5.36) with $n = 2$. The parameter a scales the elements in rows 2 and 3 of the support points symmetrically around $\boldsymbol{\theta}_{p_2}^{0}$.

For reference, setting $a = 1$ yields the following elements in rows 2 and 3 of the support points

$$\boldsymbol{\theta}^1 = \begin{bmatrix} ? \\ 1.9659 \\ 0.7412 \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^2 = \begin{bmatrix} ? \\ 0.7412 \\ 1.9659 \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^3 = \begin{bmatrix} ? \\ 0.2929 \\ 0.2929 \\ ? \\ \vdots \\ ? \end{bmatrix} \tag{5.49}$$

From Eqs. (5.45) and (5.49) it can be seen that both interfaces 1 and 2 determine the value of the element in row 2 for the 3 support points. However, the values indicated by the two interfaces are different in row 2 for support points 1 ($0.7412 \neq 1.9659$) and 2 ($1.9659 \neq 0.7412$).

One way to solve this incompatibility is to add points 1 and 2 in Eq. (5.49) unchanged as points 4 and 5, increasing the total number of support points to 5. That way, the following support points are defined

$$\boldsymbol{\theta}^1 = \begin{bmatrix} 1.9659 \\ 0.7412 \\ ? \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^2 = \begin{bmatrix} 0.7412 \\ 1.9659 \\ ? \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^3 = \begin{bmatrix} 0.2929 \\ 0.2929 \\ 0.2929 \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^4 = \begin{bmatrix} ? \\ 1.9659 \\ 0.7412 \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^5 = \begin{bmatrix} ? \\ 0.7412 \\ 1.9659 \\ ? \\ \vdots \\ ? \end{bmatrix}$$

However, swapping support points 1 and 2 in Eq. (5.49) results in an equivalent set of points that is fully compatible to the one in Eq. (5.45) since the elements in row 2 of the corresponding points are the same. Consequently, the points from Eqs. (5.45) and (5.49) can be "merged" in row 2 producing the support points

$$\boldsymbol{\theta}^1 = \begin{bmatrix} 1.9659 \\ 0.7412 \\ 1.9659 \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^2 = \begin{bmatrix} 0.7412 \\ 1.9659 \\ 0.7412 \\ ? \\ \vdots \\ ? \end{bmatrix}, \quad \boldsymbol{\theta}^3 = \begin{bmatrix} 0.2929 \\ 0.2929 \\ 0.2929 \\ ? \\ \vdots \\ ? \end{bmatrix} \tag{5.50}$$

Eq. (5.50) shows that the number of support points remains 3 while interfaces 1 and 2 are both considered. The swap between points 1 and 2 in Eq. (5.49) is essential to avoid increasing the number of support points from 3 to 5.

The same approach is used in all subsequent interfaces. **Table 5.1** summarizes relevant information for some interfaces.

Interface l	**Index set** p_l	**Number of associated parameters** $n_{\theta,l}$	**Minimum number of support points for given interface** $n_{\theta,l}+1$
1	$\{1,2\}$	2	3
2	$\{2,3\}$	2	3
3	$\{3,4\}$	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
9	$\{9,10\}$	2	3

Table 5.1 Information concerning needed support points for each interface of given model.

After considering all interfaces and performing the necessary swaps, the final support points needed for local-level reduction (assuming $a=1$) are

$$\boldsymbol{\theta}^1 = \begin{bmatrix} 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \end{bmatrix}, \quad \boldsymbol{\theta}^2 = \begin{bmatrix} 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \\ 1.9659 \\ 0.7412 \end{bmatrix}, \quad \boldsymbol{\theta}^3 = \begin{bmatrix} 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \\ 0.2929 \end{bmatrix} \tag{5.51}$$

It can be seen that 3 support points are adequate for the approximation of interface modes in parametrized models based on local-level reduction. On the other hand, parametrization based on global-level reduction requires at least 11 support points (see section 5.4.2).

It must be noted that if the scaling parameter a was not the same for all model parameters (rows of the support points) the “merging” of rows would not be possible. In that case more support points would be added.

The ability to “merge” rows of support points is the big advantage of generating support points based on n-dimensional simplices. This property is only utilized by parametrization based on local-level interface reduction.

5.4.4 Frequency of Interpolations Between the Two Parametrization Methods

Sections 5.4.2 and 5.4.3 have demonstrated that parametrized models based on local-level reduction require less support points than a corresponding model based on global-level reduction. In the case of the 10-parameter beam model it was 3 and 11 points for local- and global-level reduction, respectively.

This section investigates the frequency at which sample points lie in the convex hull of the support points generated for the two parametrization methods. This is equal to the frequency of Interpolations, which can be made only when the sample points lie in the convex hull of the support points.

One hundred random sample points are generated. Each sample point $\boldsymbol{\theta}^k$ is normally distributed with mean $\boldsymbol{\theta}^0 = \langle 1,\dots,1\rangle^T$ and covariance $\sigma^2\mathbf{I} = 0.05^2\,\mathbf{I}$, that is

$$\boldsymbol{\theta}^k \sim \mathcal{N}(\boldsymbol{\theta}^0, \sigma^2\mathbf{I}), \quad k = 1,\dots,100 \tag{5.52}$$

For the parametrization based on global-level reduction, each sample point must lie in the convex hull of the 11 support points defined in Eq. (5.41).

For the parametrization based on local-level reduction, only the associated model parameters for each interface $l, \quad l = 1,\dots,9$ indicated in index set p_l (see **Table 5.1**) must lie in the convex hull of the 3 support points generated using the procedure in section 5.4.3.

Support points generated for both parametrizations depend on the scaling parameter a around the nominal point. The results in **Fig. 5.2** consider 10 sets of support points (for each parametrization) for 10 values of a: $a = 0.1, 0.2,\dots,1$.

Fig. 5.2 clearly shows that parametrization based on local-level interface reduction results in much fewer out-of-hull sample points – for the same value of a – compared to the parametrization based on global-level reduction.

Specifically, for $a \geq 0.3$, at least 95% of the sample points lie in the convex hull of the support points generated for local-level reduction. Conversely, a must be at least 0.8 for more than 50% of the sample points to lie in the convex hull of the support points generated for global-level reduction. All this while the former parametrization uses 3 support points, and the latter uses 11.

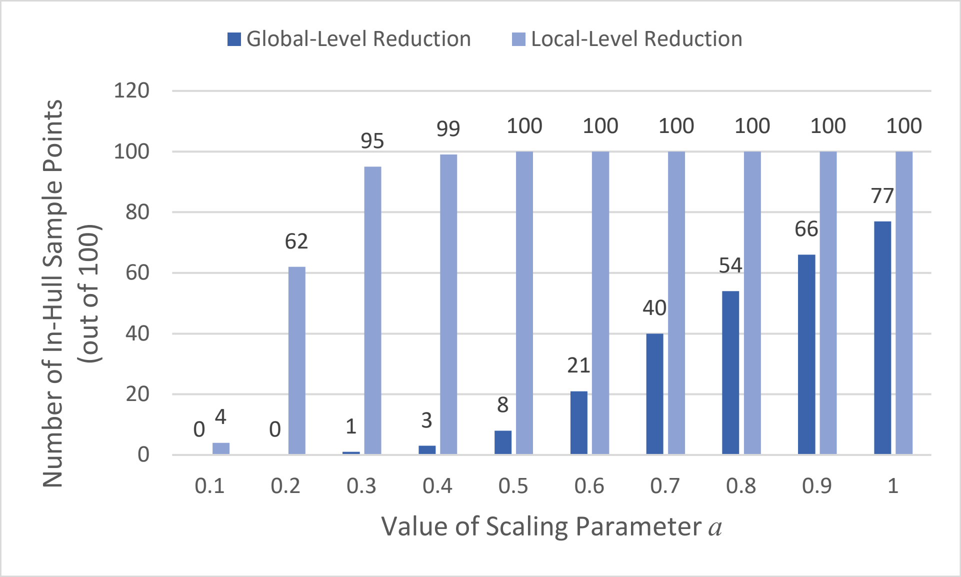

Fig. 5.2 Frequency of interpolations for both parametrizations and for increasing values of a

5.4.5 Comments on the Two Parametrization Methods

It has been shown that, for the given model, parametrization based on local-level reduction requires less support points than the one based on global-level reduction (see sections 5.4.2 and 5.4.3). At the same time, more sample points lie in the convex hull of the support points generated for the former method – even though it requires less support points than the latter one (see section 5.4.4).

It is clear, therefore, that for models with a moderate-to-large number of parameters, parametrization based on local-level interface reduction is superior to that based on global-level reduction in terms of number of needed support points and frequency of interpolations.

6 Application

This chapter investigates the computational efficiency and accuracy of the proposed non-parameterized model reduction techniques.

A demonstrative FE model is first introduced. It is developed using the commercial program COMSOL Multiphysics [20].

The CMS techniques presented in chapters 2 and 3 are then applied on the model. For this, MATLAB [21] code is used which was developed for the purpose of this thesis.

6.1 FE Model: The Metsovo Bridge

The model used to illustrate the proposed methodologies is based on the Metsovo bridge. The bridge is depicted in **Fig. 6.1** [17][22].

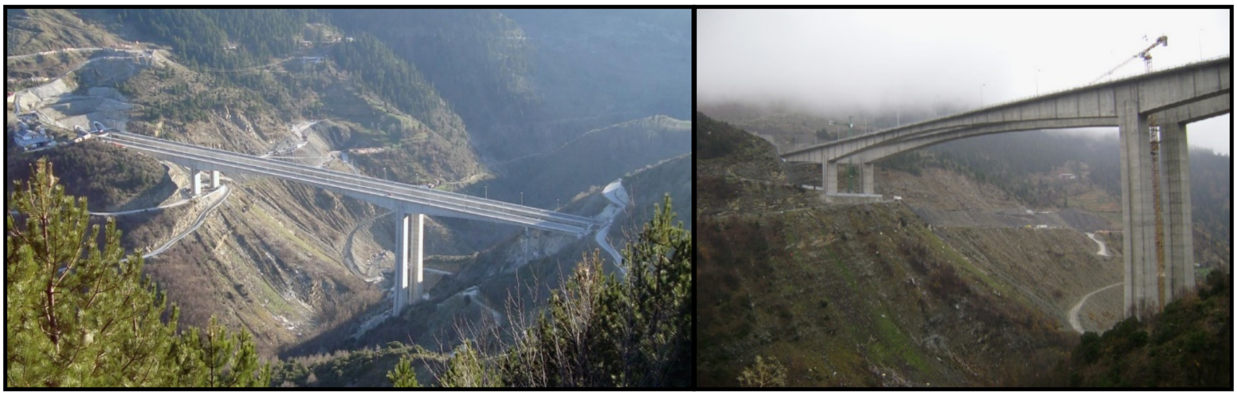

Fig. 6.1 Two views of the Metsovo bridge.

The Metsovo bridge is located at the section 3.3 of Egnatia Motorway, westwards of Metsovo village. It bridges the Metsovitikos river and the corresponding gorge between Metsovo and Anilio villages. The bridge has the longest span in Greece among cantilever bridges and one of the longest worldwide [22].

The total length of the bridge is 537 m. The bridge has 4 spans, of length 44.78 m, 117.87 m, 235.00 m, 140.00 m and three piers of which pier P1, 45 m high, supports the boxbeam superstructure through pot bearings (movable in both horizontal directions), while P2 and P3 piers (110 m and 35 m, respectively) connect monolithically to the superstructure. The total width of the deck is 13.95 m [17].

The superstructure is prestressed of single boxbeam section, of height varying from the maximum 13.5 m in its support to pier P2 to the minimum 4.00 m in key section. Piers P2 and P3 are founded on huge circular Ø12.0 m rock sockets in the steep slopes of the Metsovitikos river, in a depth of 25 m and 15 m, respectively [17].

A longitudinal view of the bridge can be seen in **Fig. 6.2** [22].

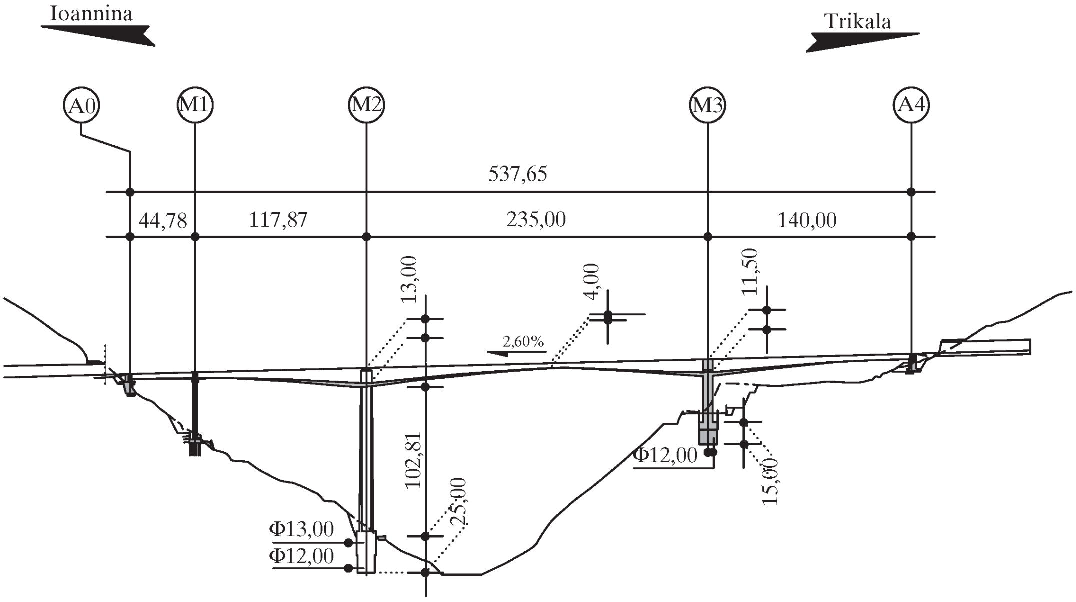

Fig. 6.2 Longitudinal view of the Metsovo bridge.

The FE model of the bridge is constructed using the commercial package COMSOL Multiphysics. For this, a CAD model of the bridge is initially made using the available design plans, the geometric details, and the material properties of the structure. This model is then imported in the COMSOL environment as the geometry of the bridge.

To simplify the model, the pot bearings are treated as linear structures. To simulate the effect of the soil, the foundations and bearings are attached to large soil blocks with fixed boundaries. The FE model of the bridge and its division into components is presented in **Fig. 6.3**. For demonstration purposes, it is divided into 22 component.

The nominal values of the relevant material properties are presented in **Table 6.1**.

Material property	Deck and bearings	Piers and foundations	Soil
Young's modulus, E [GPa]	37	34	10^{11}
Poisson's ratio, ν	0.2	0.2	0.3
Density, ρ [kg/m^3]	2548	2548	1800

Table 6.1 Nominal values of the material properties of the FE model.

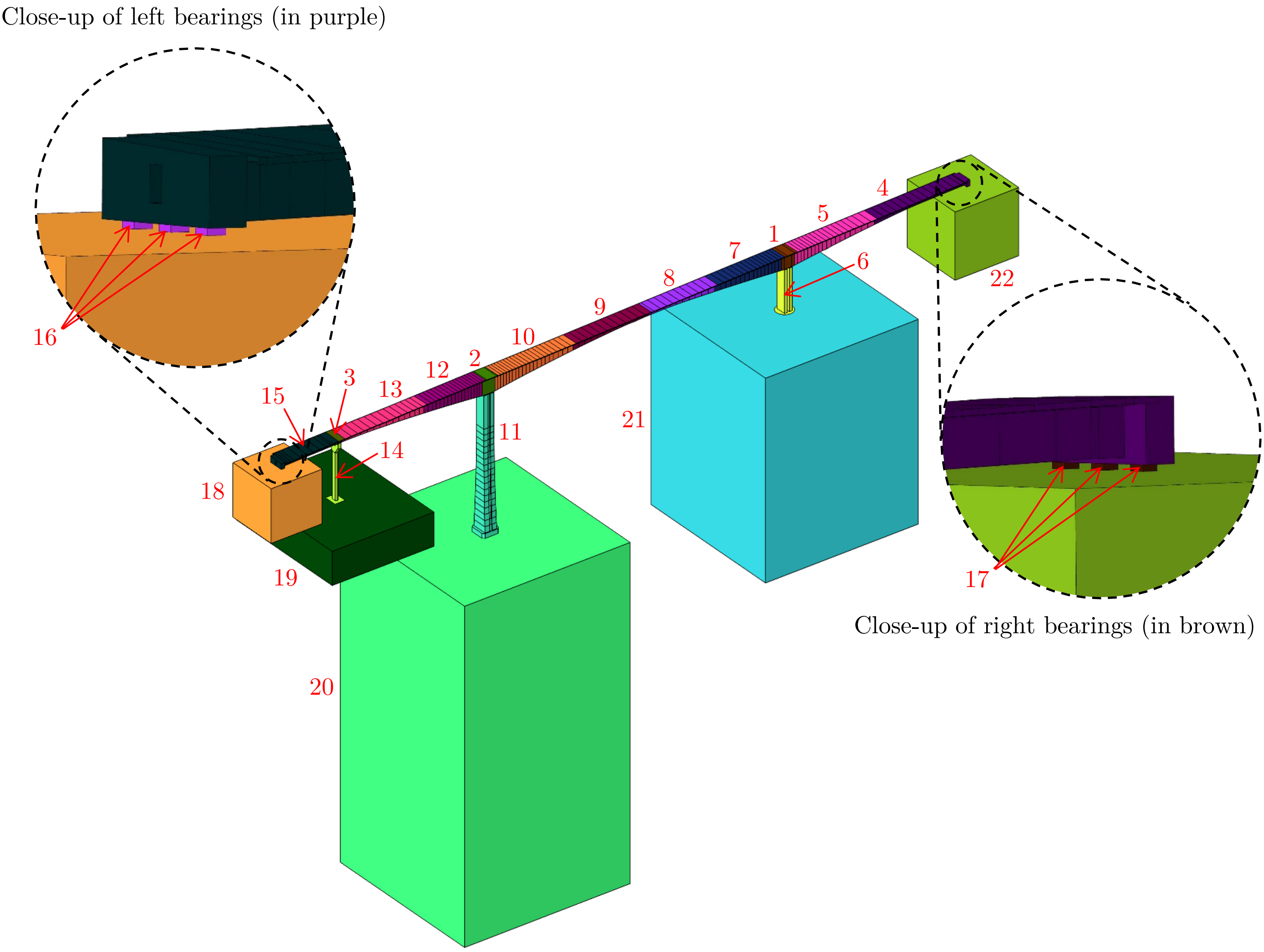
Close-up of left bearings (in purple)
16
15
3
13
12
2
10
9
8
7
1
5
4
6
22
11
14
18
19
21
20
17
Close-up of right bearings (in brown)

Fig. 6.3 FE model of the Metsovo bridge and its components.

The mesh consists of three-dimensional tetrahedral Lagrange finite elements with quadratic discretization. It is created using the "normal" settings concerning element size. This results in a model with 168,008 finite elements and 944,613 DOF. Model reduction is well suited for this FE model since it consists of nearly one million DOF.

A typical pier and deck section with their FE mesh can be seen in **Fig. 6.4**.

All necessary model data (division into components, mesh connectivity etc.) are then transferred to MATLAB environment using the LiveLink module of COMSOL Multiphysics.

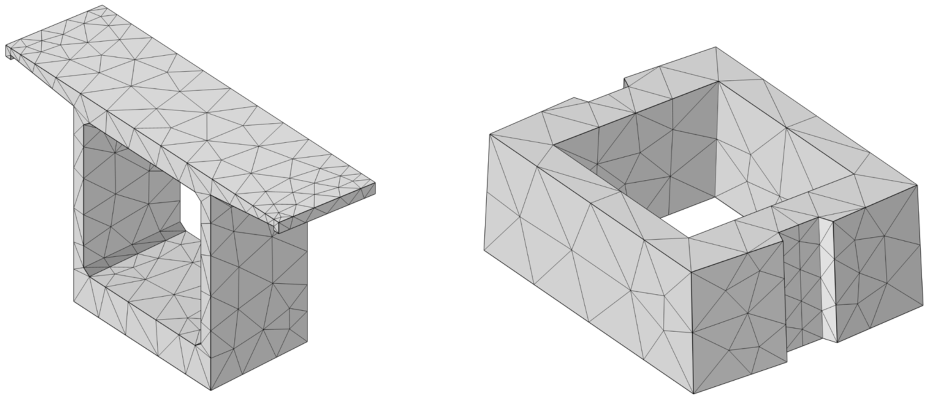

Fig. 6.4 Typical deck section (left) and section of highest pier (right) with their FE mesh.

6.2 Reduced-Order FE Model: No Interface Reduction

In this section, the standard formulation of the Craig-Bampton CMS method is applied to the FE model of the bridge (see section 2.3). The reduction is performed only on the internal DOF of each component while the interface DOF are kept unreduced.

6.2.1 Number of Kept Fixed-Interface Normal Modes for Each Component

The highest modal frequency that is of interest is denoted as the cutoff frequency ω_c . It is selected to be the 20th modal frequency of the bridge. The lowest 20 modal frequencies computed from the unreduced mass and stiffness matrix of the FE model are presented in **Fig. 6.5**.

It can be seen that ω_c is approximately 4.5 Hz.

The kept fixed-interface normal modes of each component (computed using Eq. (2.4)) are all those that have a modal frequency less than $\omega_{\max} = \rho\omega_c$, a multiple of the cut-off frequency ω_c . The multiplication factor ρ controls the number of kept modes per component. Larger values of ρ lead to more retained fixed modes per component.

The objective is to find the value of ρ that leads to a maximum fractional error between the modal frequencies computed using the complete FE model and the modal frequencies computed using the reduced-order model of the order of approximately 10^{-2} (1%). After trial and error, the value of ρ is found to be $\rho = 2.5$.

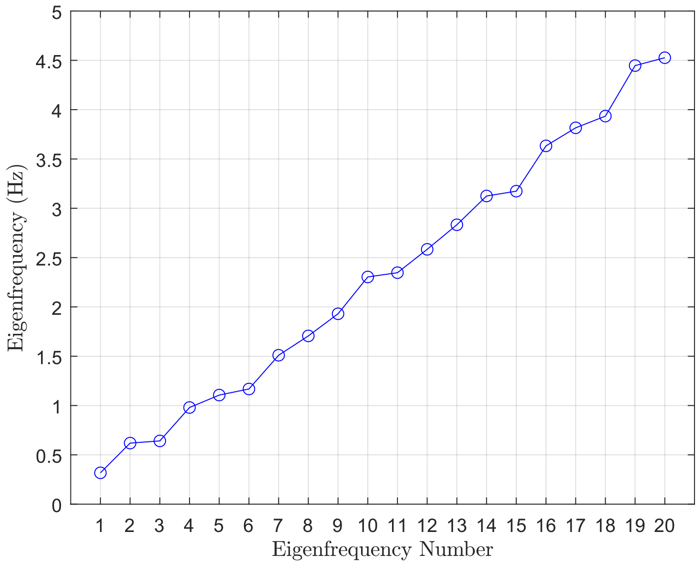

Fig. 6.5 Lowest 20 eigenfrequencies of the original (unreduced) FE model of the Metsovo bridge.

6.2.2 Results

Fig. 6.6 shows the fractional error for the first 20 modal frequencies for $\rho = 2.5$.

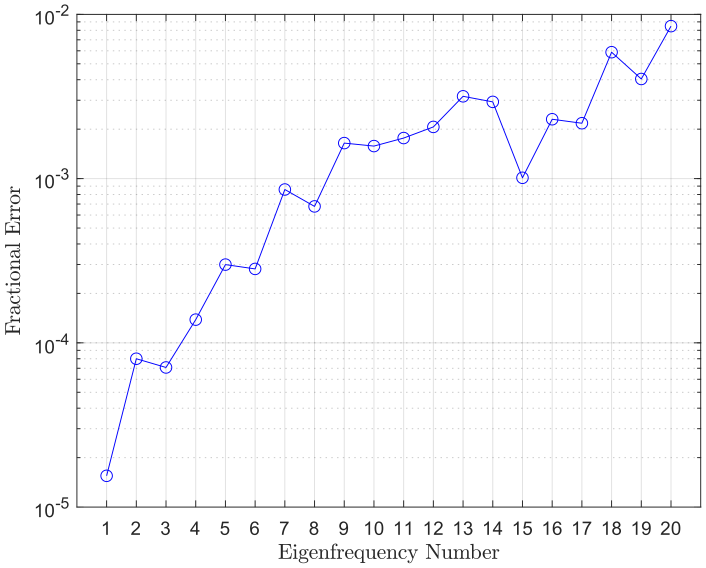

Fig. 6.6 Fractional modal frequency error - as a function of eigenmode number - between the predictions of the full model and the reduced-order model without interface reduction and $\rho = 2.5$.

It is clear that, for $\rho = 2.5$, the maximum fractional error is of the order of 10^{-2} (for the 20th eigenfrequency).

Fig. 6.7 shows the number of internal DOF per component for the full model and for the reduced model with $\rho = 2.5$. The number of internal DOF is reduced more than three orders of magnitude in some components.

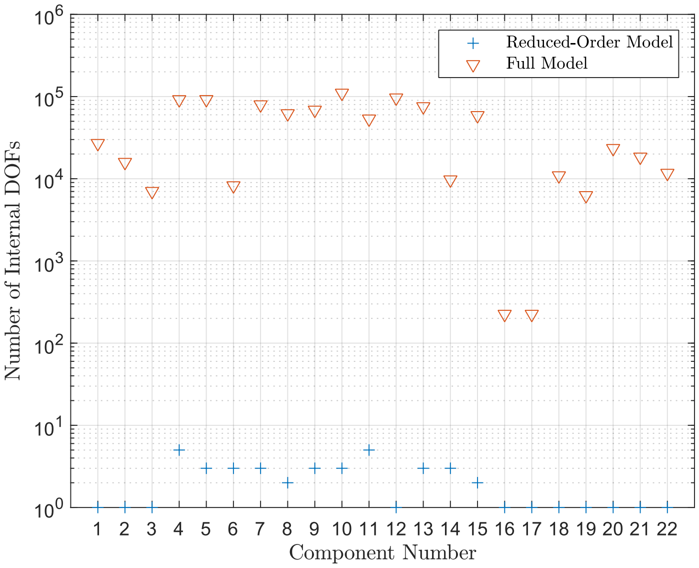

Fig. 6.7 Number of DOF per component of the full and reduced FE model of Metsovo bridge for $\rho = 2.5$ and no interface reduction.

6.3 Reduced-Order FE Model: Global-Level Interface Reduction

Here, reduction of internal as well as interface DOF is considered. The interface reduction is performed at the global level where all interfaces are treated as one (see section 3.1).

The number of kept normal modes for each component is calculated the same way as in the previous section, that is, with $\rho = 2.5$.

6.3.1 Number of Kept Global Interface Modes

The number of kept interface modes is computed similarly to the number of retained fixed-interface normal modes for each component (see section 6.2.1).

The kept interface modes (computed using Eq. (3.3)) are all those that have a modal frequency less than $\omega_{\max} = \nu\omega_c$, a multiple of the cut-off frequency. The multiplication factor ν controls the number of kept interface modes.

The objective is, again, to find the value of ν that leads to a maximum fractional error between the modal frequencies computed using the complete FE model and the modal frequencies computed using the reduced-order model of the order of approximately 10^{-2} (1%). After trial and error, the value of ν is found to be $\nu = 2.4$.

6.3.2 Results

For $\nu = 2.4$, the fractional error for the first 20 modal frequencies is presented in **Fig. 6.8**. The maximum fractional error is of the order of 10^{-2} for the 20th eigenfrequency.

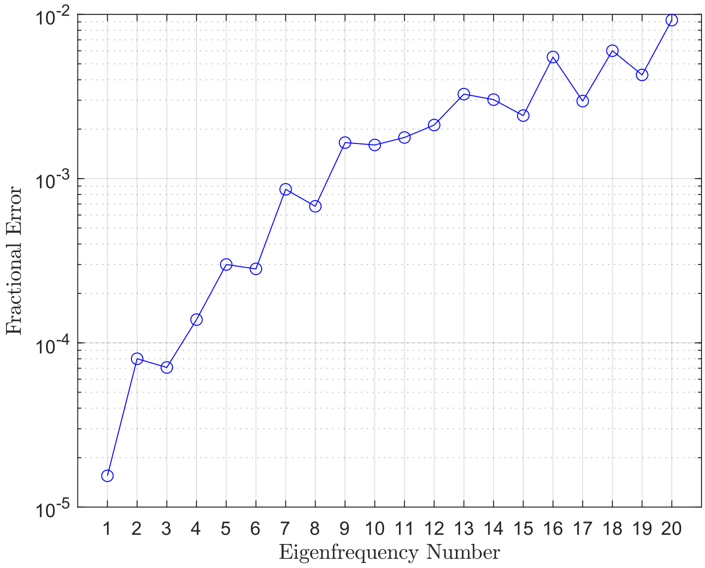

Fig. 6.8 Fractional modal frequency error - as a function of eigenmode number - between the predictions of the full model and the reduced-order model with global-level interface reduction, $\rho = 2.5$ and $\nu = 2.4$.

6.4 Reduced-Order FE Model: Local-Level Interface Reduction

In this section, interface reduction at the local level is applied (see section 3.2). In local-level reduction, each interface is treated separately.

The number of kept normal modes for each component is calculated the same way as in the previous two sections, that is, with $\rho = 2.5$.

6.4.1 Selection of Interfaces

Since every interface is treated separately, the selection of interfaces impacts the computational efficiency of the model reduction technique. There is no “correct” way to select interfaces, but some ways are more efficient than others. That is, they require less modes per interface to give accurate results.

Various interface definitions were examined – always taking into consideration the guidelines presented in section 3.2.1. The chosen interface selection consists of thirteen interfaces and is shown in **Fig. 6.9**. It arises “normally” since boundaries that are close together are selected to form a single interface. Information about adjacent components and the number of DOF for each interface is presented in columns two and three of **Table 6.3**.

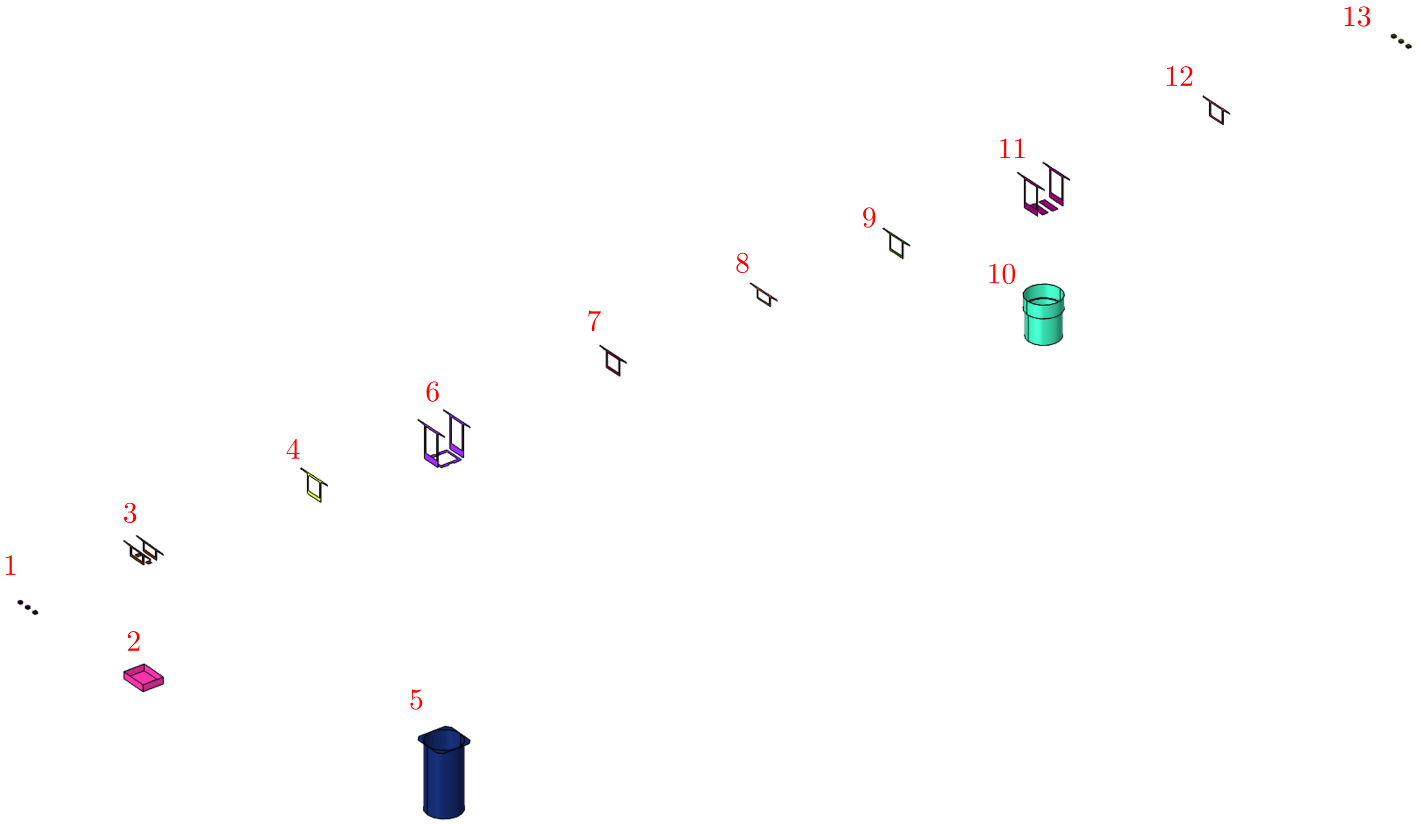

Fig. 6.9 Selection of interfaces for local-level reduction of the FE model of the Metsovo bridge.

6.4.2 Number of Kept Local Interface Modes for Each Interface

Two methods are examined to calculate the number of retained modes for each interface. One is based on the cutoff frequency ω_c and the other is based on solving an optimization problem.

6.4.2.1 Number of kept interface modes based on the cutoff frequency ω_c

This method is similar to that used to calculate the number of kept global interface modes in global-level reduction. The kept local interface modes for each interface (computed using Eq. (3.17)) are all those that have a modal frequency less than $\omega_{\max} = \mu\omega_c$, a multiple of the cut-off frequency (4.5 Hz).

The value of the multiplication factor μ must be such, that the maximum fractional error between the modal frequencies computed using the complete FE model and the modal frequencies computed using the reduced-order model is of the order of approximately 10^{-2} (1%). After trial and error, the value of μ is found to be $\mu = 70.5$.

The number of retained modes for each interface using this method is presented in the last column of **Table 6.3**.

6.4.2.2 Number of kept interface modes as a solution to an optimization problem

This method of selecting the number of retained modes per interface uses optimization to find a solution.

The optimization problem consists of thirteen independent variables $\theta_i, \quad i = 1, 2, ..., 13$. Each variable is associated with an interface and indicates the number of kept modes for that interface.

The variables can take only integer values between one (the minimum number of kept modes for an interface) and an arbitrary upper bound. The upper bound is selected at 100 modes for every variable.

The objective function to be minimized is the sum of the squares of the independent variables, that is

$$\min_{\boldsymbol{\theta} \in \mathbb{Z}^+} f(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|^2 \tag{6.1}$$

where

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_{13}]^T \tag{6.2}$$

is the vector of independent variables.

The constraint is that the maximum fractional error between the modal frequencies computed using the complete FE model and the modal frequencies computed using the reduced-order model is no larger than 10^{-2} (1%).

To solve this integer optimization problem, MATLAB's "ga" function with integer constraints was used [23]. It is part of the global optimization toolbox and uses genetic algorithms to solve the optimization problem.

Four runs of the algorithm were performed on a computer with a 64-thread CPU running at 3.7 GHz and 128 GB of RAM. Because the genetic algorithm is stochastic, the results differed a little each time.

The number of generations and the time for the algorithm to finish for each run are presented in **Table 6.2**. The results of each run are shown in columns 4 through 7 of **Table 6.3**.

	Run 1	**Run 2**	**Run 3**	**Run 4**
Number of generations	249	212	281	269
Time for completion [hours]	13	11	25	24

Table 6.2 Number of generations and time for completion for each run of the genetic algorithm.

Interface number	**Adjacent components**	**Number of DOF**	**Number of kept modes**				
			Using optimization				Using a cutoff frequency with $\mu = 70.5$
			Run 1	Run 2	Run 3	Run 4	
1	15-16-18	450	5	10	10	9	11
2	14-19	561	1	1	6	8	1
3	3-13-14-15	1524	28	33	28	28	63
4	12-13	663	38	38	38	39	37
5	11-20	3123	1	2	8	12	1
6	2-10-11-12	2805	59	59	60	59	92
7	9-10	603	37	37	37	37	36
8	8-9	603	33	33	33	33	33
9	7-8	612	23	19	19	21	37
10	6-21	2259	11	3	1	5	1
11	1-5-6-7	2112	37	40	40	37	78
12	4-5	648	22	7	23	11	38
13	4-17-22	450	4	9	5	5	11
Total	-	16413	299	291	308	304	439

Table 6.3 Information for each interface involved in local-level reduction along with the number of kept interface DOF resulting from optimization and use of a cutoff frequency.

6.4.3 Results

The fractional error for the first 20 modal frequencies and the two methods of determining the number of kept modes per interface is presented in **Fig. 6.10**.

It can be seen that the four optimization runs give very similar results. The cutoff frequency method gives smaller fractional errors for the medium modal frequencies but performs the same as the other method for the 1^{st} and 20^{th} eigenfrequencies. That is, the maximum fractional error for both methods is of the order of 10^{-2}.

The total number of kept interface modes using optimization (≈300 kept modes) is about 46% smaller than that resulting from the use of a cutoff frequency (439 kept modes). This justifies the use of optimization as a method to compute the number of retained modes for each interface.

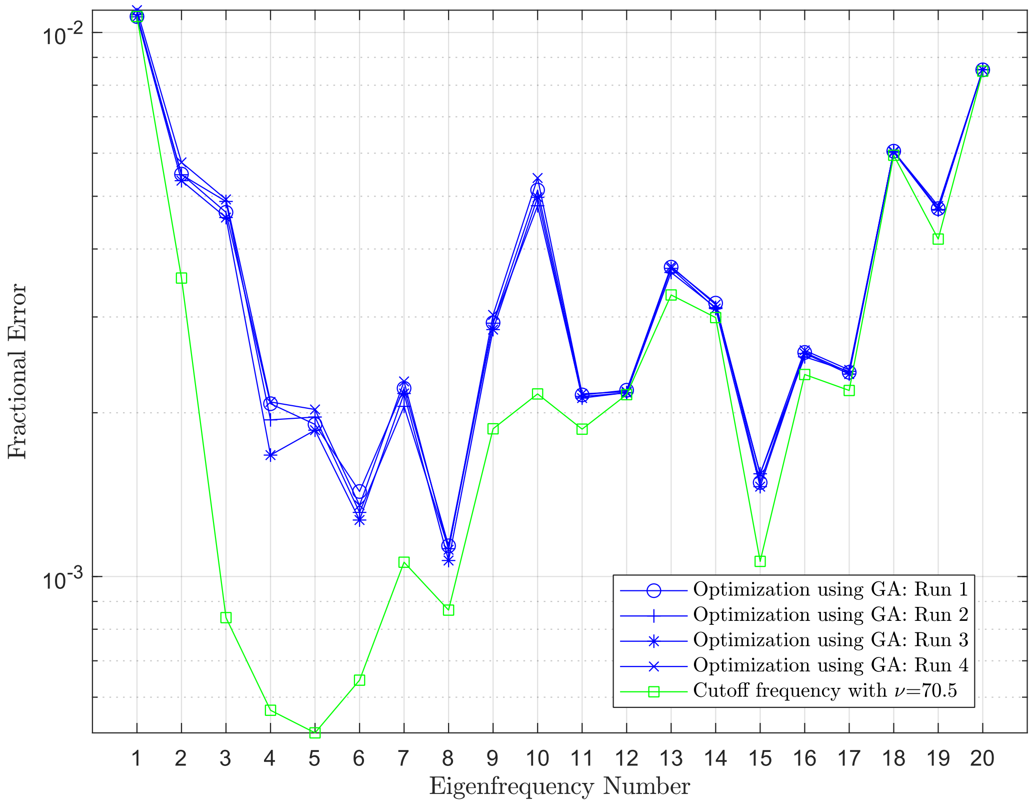

Fig. 6.10 Fractional modal frequency error - as a function of eigenmode number - between the predictions of the full model and the reduced-order model with local-level interface reduction, $\rho = 2.5$ and two methods of determining the number of kept modes per interface.

6.5 Comparison of methods

The number of DOF associated with the full FE model and the three reduced-order models already presented are shown in **Table 6.4**. When interface reduction is not considered, the total number of DOF is reduced by one order of magnitude. With global- and local-level interface reduction, the total DOF are reduced by four and three orders of magnitude, respectively.

The last row of the same table shows the time required for the calculation of the 20 lowest interface modes for the full model and the reduced-order models. The eigenproblems were solved using MATLAB on a computer equipped with a 64-thread CPU running at 3.7 GHz and 128 GB of RAM. Reduction of internal DOF only, results in a model that can be solved about 10 times faster than the full model. When interface reduction is considered at the global and local level, the time is reduced by approximately four and three orders of magnitude, respectively.

	Full model	**Reduced-order model: No interface reduction with** $\rho = 2.5$	**Reduced-order model: Global-level interface reduction with** $\rho = 2.5$ **and** $\nu = 2.4$	**Reduced-order model: Local-level interface reduction with** $\rho = 2.5$ **and kept interface modes given by run 2 of optimization**
Total Internal DOF	928,200	46	46	46
Total Interface DOF	16,413	16,413	36	291
Total DOF	944,613	16,459	82	337
Time to calculate the 20 lowest modal frequencies [sec]	174	17	<10^{-2}	<10^{-1}

Table 6.4 Number of DOF and time to perform an eigenvalue analysis for the full FE model and the three reduced-order models of the Metsovo bridge.

Fig. 6.11 shows the fractional error for the first 20 modal frequencies between the full FE model of the bridge and the three reduced-order models in **Table 6.4**.

When global-level interface reduction is considered, the model behaves very similarly to that created without reducing the interface DOF. Both of these methods demonstrate the smallest fractional error at the 1st modal frequency. As the eigenfrequency number increases, the error also increases, and the maximum fractional error occurs at the 20th modal frequency. The close performance of these two methods is observed in [7] where numerical experiments have been performed in a cantilevered-plate model, a w-bracket model and a can-beam model. The authors attribute the good performance of the global-level method to the proper consideration of the coupling between all substructures.

On the other hand, for local-level reduction, the maximum fractional error occurs at the 1st eigenfrequency. It then decreases until the 6th modal frequency and performs similarly to the other two CMS methods for the rest modal frequencies. Analogous behavior has been observed in [7] for the applications mentioned above and in [17] where the same bridge has been analyzed. High accuracy was not expected of the method. As already mentioned, the benefits of such a CMS method are:

- Only the partitions of the reduced-order matrices that correspond to the DOF of each interface are analyzed sequentially or in parallel (see Eq. (3.17)). This can be helpful for very large FE models with millions of interface DOF.
- They generally require less support points to approximate interface modes than global-level methods (see section 5.4).

From the above discussion it is clear that each CMS method has its advantages and disadvantages. The final decision lies on the user that has to decide which method is better suited for the problem at hand.

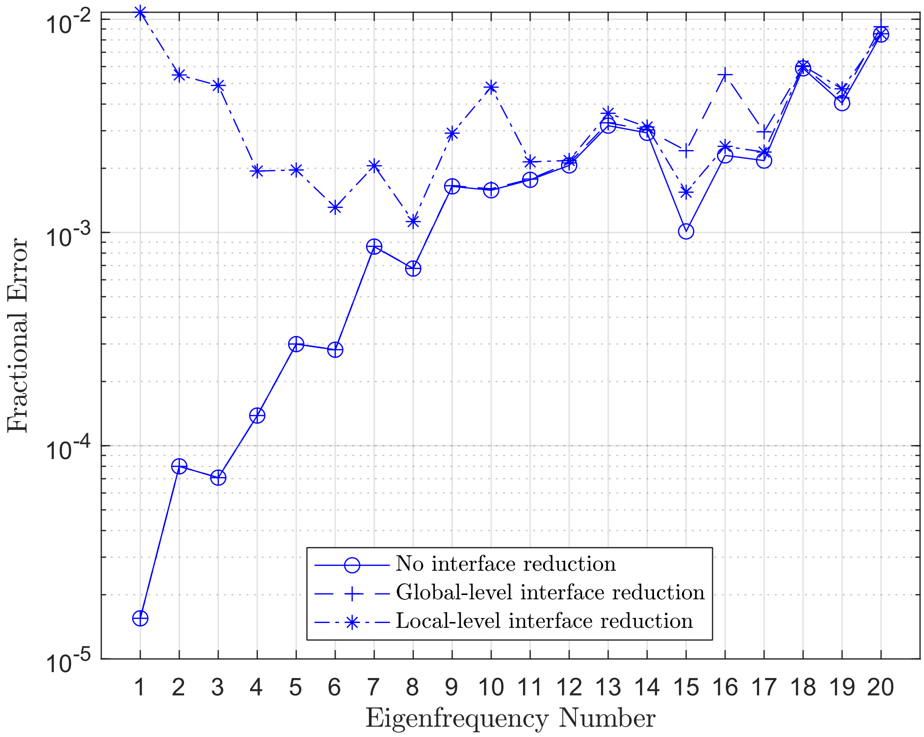

Fig. 6.11 Fractional modal frequency error - as a function of eigenmode number - between the predictions of the full model and the three reduced-order models presented in **Table 6.4**.

7 Conclusions

This work provided the theoretical basis for three non-parameterized CMS methods:

1. The classic Craig-Bampton method without interface reduction
2. A global-level interface reduction method based on the Craig-Bampton technique
3. A local-level interface reduction method based on the Craig-Bampton technique

An improved formulation of the classic Craig-Bampton method has also been presented. It takes into consideration the residual normal modes of each component and is expected to produce more accurate models compared to the standard formulation of the method. Due to excessive computational requirements, this improved formulation has not been presented in the application section.

Moreover, this thesis proposed three parametrized CMS methods that can be used for performing structural dynamics simulations. They are the parametrized equivalents of the above non-parameterized methods.

These methods are well suited for simulation-based problems because it has been shown that the resulting parametrized reduced-order matrices needed to estimate the dynamic response of the original system can be expressed explicitly in terms of model parameters. Therefore, the computationally-intensive process of reconstructing them at the substructure level at each sample of the simulation process is completely avoided.

For the two parametrized CMS methods using interface reduction, an efficient interpolation scheme has been presented. It involves support points in the parameter space which are used to approximate the interface modes at each sample point. Using such a scheme accelerates the simulation process since there is no need for a direct interface analysis.

A novel method for generating support points based on n-dimensional simplices has been developed. It proved especially useful in minimizing the number of support points for the parametrized CMS method using local-level interface reduction. Also, most of the formulations concerning that method are novel since no relevant work on the specific technique could be found.

The three non-parametrized methods have been applied on a large FE model of nearly one million DOF. For the first two CMS methods, the number of kept normal modes and interface modes was determined using a cutoff frequency. For the third method, optimization using a genetic algorithm with integer constraints proved more efficient than using a cutoff frequency. The results concerning fractional modal frequency error between the predictions of the full model and the reduced-order models created with the three CMS methods seemed to agree with similar studies in bibliography.

There are certainly some interesting open areas that were not covered by this thesis. The assessment of the computational efficiency and accuracy of the parametrized CMS methods is strongly suggested for a future work. The inclusion of non-linear components in the formulations is another interesting topic. In the special case where a structure only exhibits localized non-linearities, significant computational efficiency can be obtained using CMS due to the restriction of the iterative process to the DOF of the nonlinear substructures. Finally, the performance of the improved formulation of the Craig-Bampton method could be examined if more powerful equipment is available or more efficient formulations are developed.

8 References

[1] K. J. Bathe, *Finite Element Procedures*. 1996.

[2] R. R. Craig and A. J. Kurdila, *Fundamentals of Structural Dynamics*. Wiley, 2006.

[3] H. Jensen and C. Papadimitriou, *Sub-structure Coupling for Dynamic Analysis*. Springer, 2019.

[4] P. Seshu, "Substructuring and component mode synthesis," *Shock Vib.*, vol. 4, no. 3, pp. 199–210, 1997, doi: 10.1155/1997/147513.

[5] D. De Klerk, D. J. Rixen, and S. N. Voormeeren, "General framework for dynamic substructuring: History, review, and classification of techniques," *AIAA J.*, vol. 46, no. 5, pp. 1169–1181, 2008, doi: 10.2514/1.33274.

[6] R. R. Craig and M. C. C. Bampton, "Coupling of substructures for dynamic analyses," *AIAA J.*, vol. 6, no. 7, pp. 1313–1319, 1968, doi: 10.2514/3.4741.

[7] D. Krattiger *et al.*, "Interface reduction for Hurty/Craig-Bampton substructured models: Review and improvements," *Mech. Syst. Signal Process.*, vol. 114, pp. 579–603, 2019, doi: 10.1016/j.ymssp.2018.05.031.

[8] C. Farhat and M. Geradin, "On a component mode synthesis method and its application to incompatible substructures," *Comput. Struct.*, vol. 51, no. 5, pp. 459–473, 1994, doi: 10.1016/0045-7949(94)90053-1.

[9] W. C. HURTY, "Dynamic analysis of structural systems using component modes," *AIAA J.*, vol. 3, no. 4, pp. 678–685, Apr. 1965, doi: 10.2514/3.2947.

[10] S. Rubin, "Improved Component-Mode Representation for Structural Dynamic Analysis," *AIAA J.*, vol. 13, no. 8, pp. 995–1006, Aug. 1975, doi: 10.2514/3.60497.

[11] H. Richard, "A hybrid method of component mode synthesis," *Comput. Struct.*, vol. 1, no. 4, pp. 581–601, 1971.

[12] M. P. Castanier, Y. C. Tan, and C. Pierre, "Characteristic constraint modes for component mode synthesis," *AIAA J.*, vol. 39, no. 6, pp. 1182–1187, 2001, doi: 10.2514/2.1433.

[13] S. K. Hong, B. I. Epureanu, and M. P. Castanier, "Next-generation parametric reduced-order models," *Mech. Syst. Signal Process.*, vol. 37, no. 1–2, pp. 403–421, 2013, doi: 10.1016/j.ymssp.2012.12.012.

[14] B. Goller, H. J. Pradlwarter, and G. I. Schuëller, "An interpolation scheme for the approximation of dynamical systems," *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 1–4, pp. 414–423, 2011, doi: 10.1016/j.cma.2010.09.005.

[15] H. A. Jensen, A. Muñoz, C. Papadimitriou, and C. Vergara, "An enhanced substructure coupling technique for dynamic re-analyses: Application to simulation-based problems," *Comput. Methods Appl. Mech. Eng.*, vol. 307, pp. 215–234, 2016, doi: 10.1016/j.cma.2016.04.011.

[16] J.-G. Kim and P.-S. Lee, "An enhanced Craig–Bampton method," *Int. J. Numer. Meth. Engng*, vol. 103, pp. 79–93, 2015, doi: 10.1002/nme.

[17] C. Papadimitriou and D. C. Papadioti, "Component mode synthesis techniques for finite

element model updating," *Comput. Struct.*, vol. 126, no. 1, pp. 15–28, 2013, doi: 10.1016/j.compstruc.2012.10.018.

[18] H. A. Jensen, E. Millas, D. Kusanovic, and C. Papadimitriou, "Model-reduction techniques for Bayesian finite element model updating using dynamic response data," *Comput. Methods Appl. Mech. Eng.*, vol. 279, pp. 301–324, 2014, doi: 10.1016/j.cma.2014.06.032.

[19] "Cartesian coordinates for a regular n-dimensional simplex." https://en.wikipedia.org/wiki/Simplex.

[20] "COMSOL Multiphysics 5.5 User's Guide." https://doc.comsol.com/5.5/docserver/.

[21] "MATLAB R2020a User's Guide." https://www.mathworks.com/help/matlab/.

[22] S. Stathopoulos *et al.*, "The Metsovo bridge, Greece," *Struct. Eng. Int. J. Int. Assoc. Bridg. Struct. Eng.*, vol. 20, no. 1, pp. 49–53, 2010, doi: 10.2749/101686610791555568.

[23] "MATLAB R2020a Genetic Algorithm Documentation." https://www.mathworks.com/help/gads/genetic-algorithm.html.

9 Appendix

Great effort has been put in developing MATLAB code that can be used to apply every formulation presented in this thesis. Currently, the code is compatible with FE models created with COMSOL Multiphysics. In the future, it may be modified to accept models from other commercial FE packages as well.

In the time of writing, the code is hosted on a GitHub repository under a non-commercial license. It can be found using the following link:

https://github.com/FK-MAD/CMS

If for some reason the link does not work or the code is removed, anyone interested can contact me on fkatsimalis@gmail.com.

What follows is a presentation of the function *input.m* that controls most aspects of the code. The comments are self-explanatory.

```
%% input


% ---parallelization settings---
indata.num_workers=6; % number of workers to use
% ---


% ---optimization settings---
indata.num_modes=20; % number of first modes to use in error term between unreduced and reduced model
% ---


% ---matrix assembly settings---
% method of building the block-diagonal matrices. It might help with very
% big matrices that may not fit in RAM.
% 1 -> use files mats_S_k.mat, read variables in a for-loop and build the block-diagonal incrementally
% 2 -> use files mats_S_k.mat, read variables in a for-loop and build the block-diagonal once
% 3 -> use file mats_S.mat and build the block-diagonal once
indata.blkdiag_method=2;
% ---


% ---reduction method---
% without parametrization: 1=no interface reduction | 2=global interface reduction | 3=local interface reduction
% with parametrization:    4=no interface reduction | 5=global interface reduction | 6=local interface reduction
indata.reduction_I=1;
% ---

% ---use of static correction---
% 0=without static correction | 1=with static correction
indata.static=0;
% ---


% ---kept modes for component groups---
% method of calculating the kept modes
% 0=explicitly using n_id_S | 1=until the target eigenfrequency for each group is reached
indata.eigf.group.method=1;

% all vectors have:
% rows=1
% columns>=number of component groups (will run normally if more columns than component groups exist)

% this is used if method=0
indata.n_id_S=50*ones(1,100); % kept fixed-interface normal modes for each group of components

% this is used if method=1
r=2.5*ones(1,100); % multiplication constant used to define the target frequency
indata.eigf.group.multiplier=r;
indata.eigf.group.target=r*4.5; % target eigenfrequency (Hz) for each group of components

% this controls the way modes are searched
indata.eigf.group.max=500*ones(1,100); % maximum allowed number of modes
indata.eigf.group.step=50*ones(1,100); % increase in the number of calculated modes if target is not reached
indata.eigf.group.init=50*ones(1,100); % initial number of calculated modes
% ---
% ---stored modes for component groups---
% they are computed once and used when updating matrices during
% optimization of r. They should be enough to avoid solving the eigenproblem
% during optimization.
% If you don't want any stored modes select:
% indata.eigf.group.method_store=0 and
% indata.n_id_S_store=0*ones(1,100)

% everything here works similarly to kept modes (same logic)

indata.eigf.group.method_store=1; % 0 or 1

% this is used if method_store=0
indata.n_id_S_store=50*ones(1,100); % large values

% this is used if method_store=1
indata.eigf.group.target_store=20*4.5*ones(1,100); % large cutoff frequency

% this controls the way modes are searched
indata.eigf.group.max_store=500*ones(1,100); % large values
indata.eigf.group.step_store=50*ones(1,100); % large values
indata.eigf.group.init_store=50*ones(1,100); % large values
% ---


% ---kept modes for interfaces---
% method of calculating the kept modes
% 0=explicitly using n_IR (for global reduction) or n_IR_l (for local reduction) | 1=until the↙
target eigenfrequency for each interface is reached
indata.eigf.interface.method=0;

% all vectors have:
% rows=1
% columns>=number of interfaces (will run normally if more columns than interfaces exist)

% if global interface reduction is selected:
% only the first element of the target, max, step and init vectors is used
% (there is only one interface)

% this is used if method=0 and global reduction is selected
indata.n_IR=36; % kept interface modes for all interfaces (global reduction)


% this is used if method=0 and local reduction is selected
indata.n_IR_l=[10,1,33,38,2,59,37,33,19,3,40,7,9]; % kept interface modes for each interface (local reduction)

% this is used if method=1
v=70.5*ones(1,100); % multiplication constant used to define the target frequency
indata.eigf.interface.multiplier=v;
indata.eigf.interface.target=v*4.5; % target eigenfrequency (Hz) for each interface

% this controls the way modes are searched
indata.eigf.interface.max=1000*ones(1,100); % maximum allowed number of modes
indata.eigf.interface.step=100*ones(1,100); % increase in the number of calculated modes if target is not reached
indata.eigf.interface.init=100*ones(1,100); % initial number of calculated modes
% ---


% ---stored modes for interfaces---
% they are computed once and used when updating matrices during
% optimization of v. They should be enough to avoid solving the eigenproblem
% during optimization.
% If you don't want any stored modes select:
% indata.eigf.interface.method_store=0 and indata.n_IR_l_store=0*ones(1,100)

% everything here works similarly to kept modes (same logic)

indata.eigf.interface.method_store=0; % 0 or 1

% this is used if method_store=0 and global reduction is selected
indata.n_IR_store=50; % for global reduction, large value

% this is used if method_store=0 and local reduction is selected
indata.n_IR_l_store=100*ones(1,100); % for local reduction, large values

% this is used if method_store=1
indata.eigf.interface.target_store=80*4.5*ones(1,100); % large cutoff frequency

% this controls the way modes are searched
indata.eigf.interface.max_store=500*ones(1,100); % large values
indata.eigf.interface.step_store=100*ones(1,100); % large values
indata.eigf.interface.init_store=100*ones(1,100); % large values
% ---


% ---material properties---
% all vectors have:
% rows>=number of component groups (will run normally if more columns than component groups exist)
% columns=1

indata.E=37*10^9*ones(22,1); % Young's modulus [Pa] for each group of components. most groups are deck components -> 37 GPa
indata.E([6,11,14])=34*10^9; % groups 6, 11 and 14 are piers -> 34 GPa
indata.E(18:22)=10^20; % groups 18 through 22 are soil -> 10^11 GPa

indata.nu=.2*ones(22,1); % Poisson's ratio. for deck and piers -> 0.2
indata.nu(18:22)=.3; % groups 18 through 22 are soil -> 0.3
indata.rho=2548*ones(22,1); % density [kg/m^3]. for deck and piers -> 2548 [kg/m^3]
indata.rho(18:22)=1800; % groups 18 through 22 are soil -> 1800 [kg/m^3]
% ---


% ---general parametrization settings---
% interpolation scheme used in interpolation of interface modes (global or local reduction)
indata.quad_interp=0; % 0=linear interpolation | 1=quadratic interpolation

% static correction method
indata.invariant=0; % 0=full static correction | 1=invariant assumption


% functions of the model parameters -> one entry for each model parameter
% func_g applies on mass matrix (see Eq. (2.3))
indata.func_g=repmat({@(x) 1},1,22);

% func_h applies on stiffnes matrix (see Eq. (2.4))
indata.func_h=repmat({@(x) x},1,22);


% vectors have:
% rows=number of model parameters
% columns=1

% sample point where reduced matrices are calculated
% This is used to test the code. Normally, every sample point is generated
% during the stochastic simulation process.
indata.theta_k=ones(22,1);

% nominal point used in the invariant assumption of static correction. See
% page 42 of Book.
```

```
indata.theta_nom=ones(22,1);

% nominal point used in interpolation of interface modes (global or local
% reduction). See page 50 of Book.
indata.theta_0=ones(22,1);
% ---


% ---settings concerning support points (if parametrization is used)---
% used in interpolation of interface modes (global or local reduction)

% 'scatter_theta_l' is the fraction of theta_0 that the support points are
% scattered around theta_0 (can be different for each parameter)
% e.g. for a model with 2 parameters and theta_0=[1;1]:
% scatter_theta_l(1)=.1 -> support of parameter 1=[.9,1.1]
% scatter_theta_l(2)=.2 -> support of parameter 2=[.8,1.2]
indata.scatter_theta_l=1*ones(length(indata.theta_0),1);

% 'simplex' -> for dimension n there are needed n+1 vertices to create the convex hull
% 1D -> 2 points -> line segment
% 2D -> 3 points -> triangle
% 3D -> 4 points -> tetrahedron
% 4D -> 5 points -> 5-cell (4-simplex)
% ...

% 'hypercube' -> for dimension n there are needed 2^n vertices to create the convex hull
% 1D -> 2 points -> line segment
% 2D -> 4 points -> square
% 3D -> 8 points -> cube
% 4D -> 16 points -> 4-cube (hypercube,tesseract)
% ...

% both methods provide as few support points as possible using smart
% merging rules
indata.method_theta_l='simplex'; % 'simplex' or 'hypercube'
% ---


%% pass additional input data to structure "indata"

indata.filename=filename;
indata.save_dir=save_dir;

indata.S_0=S_0; % groups that are independent of model parameters
indata.S_j=S_j; % groups that depend of model parameters. cell 1,2,... contains the groups that depend on parameter 1,2,...
indata.n_theta=length(indata.S_j); % number of parameters. n_theta=length(func_g);

indata.group_S=group_S; % grouping of geometrical domains. cell 1,2,... contains the domains that make up group 1,2,...
indata.n_id=sum(indata.n_id_S); % total number of kept fixed-interface normal modes
indata.N_S=length(indata.group_S); % number of groups of components
indata.n_IRL=sum(indata.n_IR_l); % total number of kept interface modes using local reduction

indata.n_DIL=indata.n_id+indata.n_IRL; % dimension of reduced matrices if local reduction is used
indata.n_DI=indata.n_id+indata.n_IR; % dimension of reduced matrices if global reduction is used
```