Department of Precision and Microsystems Engineering

MODEL ORDER REDUCTION FOR NON-LINEAR STRUCTURAL DYNAMICS

SHOBHIT JAIN

Report no	: EM 2015.020
Coach	: Dr. ir P. Tiso
Professor	: Prof. dr. ir. A. van Keulen
Specialisation	: Engineeing Mechanics
Type of report	: MSc Thesis
Date	: 23-09-2015

TU Delft Delft University of Technology

Challenge the future

Model Order Reduction for Non-linear Structural Dynamics

by

Shobhit Jain
4324358

in partial fulfillment of the requirements for the degrees of

Master of Science

in Mechanical Engineering

&

Master of Science

in Applied Mathematics

at the Delft University of Technology,

to be defended publicly on Wednesday September 23, 2015 at 14:00 hrs for Mechanical Engineering and Thursday September 24, 2015 at 09:30 hrs for Applied Mathematics.

Supervisors:	Dr. ir. P. Tiso	ETH Zürich
	Dr. ir. M.B. van Gijzen	TU Delft
Thesis Committee 3ME:	Prof. dr. ir. A. van Keulen	3ME, TU Delft
	Dr. ir. P. Tiso	ETH Zürich
	Dr. ir. M.B. van Gijzen	AM, TU Delft
	Ir. L. Wu	3ME, TU Delft
Thesis Committee AM:	Dr. ir. M.B. van Gijzen	AM, TU Delft
	Prof. dr. ir. C. Vuik	AM, TU Delft
	Prof. dr. ir. A.W. Heemink	AM, TU Delft
	Dr. ir. P. Tiso	ETH Zürich

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft
Delft University of Technology

Abstract

Model Order Reduction is an important tool to solve dynamical problems in a short amount of time, which would otherwise seem computationally infeasible. General reduction techniques for nonlinear systems require a *full* solution run in order to construct a reduced order model. Such models are optimal only for the given input (load). This makes such techniques computationally expensive on their own. In the context of thin-walled structural dynamics based on finite elements and characterized by geometric nonlinearities, this work has studied and developed techniques for effective reduction without the need of a full solution run. By taking into account the nature of nonlinearities, the work progressed in two different directions. In the first case, a new nonlinear mapping (*quadratic manifold*) based reduction technique is proposed. This uses tensors to make the reduced order model thus produced, highly compact and effective - both in terms of accuracy and speed. The *speed-ups* in this case are shown to increase exponentially with the full system size. The other case addresses the field of *hyper-reduction*. Novel ways to construct training vectors required for ECSW (a recently introduced method in the field) are proposed here. These training vectors generally come from a full solution run but the new method proposes the use of quadratic manifold for training set generation. This leads to negligible *offline* costs and is shown to be very effective. All techniques have been tested on a range of examples, and the results support consistent and desirable conlusions - in terms of both speed and accuracy.

Preface

Mathematical and computational modelling plays a pivotal role in modern engineering research - whether in making predictions better, faster and/or optimizing existing designs. I envision myself working at this boundary between engineering and mathematics and thus opted to pursue education in both the disciplines. This work is a part of a double degree in Mechanical Engineering and Applied Mathematics. It addresses an issue of practical and engineering significance at heart. At mechanical engineering, novel techniques were proposed to address the issue by extension of existing techniques in a mathematically intuitive manner. After considerable efforts in implementation of these ideas, the results thus obtained showed great potential. But in order to take further steps, one needs to know the limits of applicability of these new ideas, how they would compare to the existing state of the art methods which are well-established in the mathematical community, and finally how effective these techniques would be in achieving their goal if they're tested on problems of general size. This is where mathematics came into the picture, helped in addressing these questions, and resulted in some even newer ideas (which proved to be even better than the original ones in some sense). The work had to be limited at some point but the scope could not be.

Both departments have made their distinct contributions in shaping this work and it is formally given the weightage of 60 ECTS. However, it should still be seen as a whole since the ideas from both the sides produced a much amalgamated outcome. It is for this reason that the work was not split into two separate reports, and efforts have been made to report the findings in the most coherent way possible.

Acknowledgements

Projects with multiple supervisors are usually not expected to flow very smoothly, especially when one of them addresses an engineering field and the other, mathematics. But fortunately in my case, the supervisors have been a great source of inspiration and guidance. Paolo was always available for a discussion to any level of detail when things seemed tough for me. This included everything, from derivation of equations afresh to even looking into my code at times. And Martin kept me from never losing sight of the bigger picture. Discussions with him set many problems in the right perspective. On a more personal level, his thoughts about the viewing things from the "scale of life" made interactions with him very pleasant.

Apart from my supervisors, many people have contributed in shaping this work. Interactions with Long Wu were helpful in providing critical insights about my work. Samee Rehman's knowledge about the Linux cluster and help with parallel computing on it, have been of great value. I thank him. Discussions with fellow MSc students - Anoop and Cees, who were working in closely related areas have also been very fruitful.

Friends and family have been an integral part of all my endeavours. Words are not enough to acknowledge such relations and I shall not try. But as an exception and upon his own request, I would like to thank my friend Panos for the delicious rice-waffel-peanut-butter sandwiches which kept me going.

Delft, 2015 Shobhit Jain

Contents

Nomenclature

List of Abbreviations

DEIM	Discrete Empirical Interpolation Method
ECSW	Energy Conserving Sampling and Weighting
FE	Finite Element
GRE	Global Relative Error
IVP	Initial Value Problem
LM	Linear Manifold
LMT	Linear Manifold Tensorial
MD	Modal Derivative
MMI	Maximum Modal Interaction
MMI	Modal Virtual Work
NNLS	Non-Negative Least Squares
NR	Newton-Raphson
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
QM	Quadratic Manifold
QMT	Quadratic Manifold Tensorial
ROB	Reduced order basis
ROM	Reduced order model
sNNLS	sparse Non-Negative Least Squares
VK	Von Kármán
VM	Vibration Mode

List of Symbols

$\boldsymbol{\sigma}$	Cauchy stress tensor
$\mathbf{D}$	Material matrix
$\mathbf{V}$	General notation for a reduction basis
ν	Poisson's ratio
ρ	Density
E	set of Elements sampled by ECSW
k	number of vectors in the reduction basis used for ECSW
m	number of vibration modes used in the mapping (linear manifold or quadratic manifold)

m_s total number of vectors in a Linear Manifold Basis which includes m VMs and either all or a selected few MDs

n Total number of DOFs in the system

N_e Number of degrees of freedom of element e

n_e Total number of elements in a finite element model

n_{MD} total number of MDs used in a Linear Manifold Basis ()either all or a selected few MDs)

t thickness of shell elements

Y Young's modulus

1

Introduction

Background

The size of finite element (FE) models used in industrial and research applications is steadily growing over the past decades. Although Moore's law shows the trends of increasing ability of computer architecture for handling problems of larger size more efficiently, still the stringent requirements of a more detailed and complex analysis surpass these trends. Nonetheless, this increase in computational power and maturity of the commercial FEM packages has also led to an increased interest in non-linear behaviour of structures even during the early stages of design. This leads to FE models repeatedly being solved to optimize or test various load cases. However, the repeated solution in time of large non-linear systems of equations to obtain the static and dynamic behaviour of a general structure, is still a computationally heavy task. The need of reducing the number degrees of freedom of a given model (Model Order Reduction) is thus of great importance. Efficient reduced order models are very much in need during the early design phase when the designer is typically faced with "what if" questions during analysis or when, in some cases, optimization routines are used to generate the best design.

1.1. General summary of existing methods

Model Order reduction (MOR) refers to reduction in the number of unknowns in a large model to facilitate faster solution. MOR for large nonlinear structural dynamic systems is a topic of ongoing research. Literature is replete with techniques for MOR. One particular class of structures which are very attractive for use in the aircraft/ defence industry,especially due to their load carrying mechanism, is the thin-walled structures . These are characterized by the so called geometric nonlinearities (Section 2.2) which are triggered by even small out of plane displacements of the structure and is of much practical importance. The material however, tends to remain in the linear range (where by Hooke's law is still applicable). Many approximation theories have been developed to efficiently model kinematics causing the geometric nonlinearities in thin walled structures [1]. One of such models is used in the Finite-Elements context.

MOR for linear systems is well established. Techniques such as modal superposition, modal acceleration, Guyan reduction, dynamic substructuring, component mode synthesis such as Craig-Bamption method, Rubin's method etc. are all very successful methods for linear structural dynamics [2]. Reduced Order Models (ROMs) are generally characterized by projecting the large system of equations onto a smaller subspace (constant or variable) along with a mapping (which could be nonlinear) from a high dimensional space of solution to a lower dimensional space of reduced unknowns.

For nonlinear systems, there are largely two class of reduction techniques: the ones that require the full solution in order to generate a ROM and the others which exploit the nature of nonlinearities to quickly predict the solution, without the knowledge of the full solution. Techniques such as Proper Orthogonal Decomposition (POD) come under the first category. Though the ROMs produced using such methods are generally applicable for a particular loading case in the given time span, they are very versatile with respect to nonlinearities and find use especially in running optimization routines on large systems. Projection of the equations can be done onto a Reduced Order Basis (ROB) carefully chosen offline by taking into account the nature of nonlinearities and type of loading (without the knowledge of solution), a ROB iteratively updated by monitoring the residual, or a ROB whose variation is known a priori. All these techniques fall under latter of the categories. Here the ROB is typically formed to capture the nonlinear behaviour by looking into the structure of nonlinearities, and is problem dependent. An example would be, augmenting the linear vibration mode basis with modal derivatives to capture nonlinearities. Such methods are very problem specific but can be used for a variety of load cases (in the right range) and don't require the calculation of the full solution beforehand.

The evaluation of nonlinearity at each time step and projection thereafter is a costly affair. Discrete Empirical Interpolation Method (DEIM)[3], its variant UDEIM [4] and ECSW [5] attempt to alleviate these problems by evaluating and interpolating the nonlinearity (or projected nonlinearity in case of ECSW) on a carefully chosen set of nodes (or elements) . They fall under the class of so-called *Hyper-reduction* [6] techniques. Typically these methods require training vectors which come from the full solution and might be thought of as methods in the first category, but these training sets could also come from a reduced solution using a second category method, in which case hyper-reduction can result in huge speed up factors.

1.2. Research Outline

This research mainly focusses on the finite element based Model Order Reduction of thin-walled structures under certain kinematic assumptions (Section 2.2). Most of the existing reduction techniques for nonlinear dynamical systems involve the solution of full systems first. This requirement of a full solution run can be computationally expensive, or even unaffordable at a preliminary design stage when a variety of load scenarios, geometric layouts, material choices need to be explored. Each of these cases would need a time integration of its own, which is undesirable. Thus, the research attempts to find answer to the following basic research question:

What are suitable techniques for Model Order Reduction of Dynamics of thin-walled structures characterized by geometrical nonlinearities without the need of a full solution run ?

This work starts with the review of Galerkin projection and the use of Vibration Modes (VMs) and Modal Derivatives (MDs) to compose a constant reduction basis which is shown to be effective [7, 8] for reduction of nonlinear systems excited by narrow spectrum loading (Section 3.3). Then it moves towards formulation of new techniques for Optimal selection of Modal Derivatives cheaply (Section 3.3.2) which finds use in reducing the basis size. This size tends to increase quadratically with the number of vibration modes used for calculating the modal derivatives.

Traditional methods in MOR (like POD, MDs) suffer from being based on linear models. In the more recent past, many new methods have been developed in the field of nonlinear dimensionality reduction, also called *manifold* learning, which has become a hot topic of research [9, 10]. In the context of using VMs in reduced order modelling and augmenting the basis with MDs to capture nonlinear response, a Taylor expansion based nonlinear mapping is proposed here which inherently reduces the number of unknowns for reduced modelling.

As mentioned earlier, reduction methods mainly aim to reduce time integration cost of large nonlinear systems. This is generally done by avoiding large linear system solves which need to be performed during iterative solution of nonlinear algebraic equations during each time step. These large systems are avoided done by introduction of a mapping onto a smaller subspace and then projecting onto a suitable subspace to obtain a smaller set of equations. But the linear system solve is not the only expensive task duing nonlinear time integration. Evaluation of nonlinearity and projection are also heavy tasks. The geometric nonlinearities studied in the current context are polynomial (up to cubic) in nature and in such a case the linear as well nonlinear projection based techniques can be implemented using *tensors*. The reduced system of nonlinear equations, corresponding residual and its Jacobian can be directly reproduced using tensors, without the need of first evaluation of their *full* counterparts though mapping and then projection onto the reduction basis. This makes the system highly compact by eliminating the need to touch the physical space for evaluation of nonlinearity during time integration. Hence it can lead to huge speed-up factors but this comes with an extra cost of evaluation of these tensors offline. This work involves such tensor implementations as well (Sections 3.3.3 and 3.4.1). This leads to the following research sub-question

How effective is the tensor framework in case of polynomial nonlinearities for model order reduction of nonlinear structural dynamics ?

In a more general setting (when nonlinearities may not be polynomial), tensors would not be of much help. Then the hyper-reduction class of techniques, which try to approximate the role of tensors in some sense by cheap evaluation of the nonlinearities at only selected locations in physical domain, find use. Traditionally, these methods require a full solution run to generate training sets. This leads to huge speed-up factors during time integration but they disguise enormous offline costs. Along with implementation of such hyper-reduction techniques, this work proposes new methods for very cheap construction of training vectors for use in a finite element based hyper-reduction technique (ECSW) (Section 4.3). This aims to provide the answer to the following research sub-question.

Can the hyper-reduction techniques be effectively used for model order reduction of nonlinear structural dynamical systems without the need of a full solution run ?

Finally, all the proposed reduction techniques are applied on thin walled structure examples and results are compared (Chapter 5). Three structures are considered with different types of excitation to invoke the nonlinearities under consideration. The first two structures (Sections 5.1 and 5.2) are simple, characterized by a small number of Degrees of Freedom (DOFs) and of academic interest. The final example is that of a more industrial and relatively complex structure characterized by realistically large number of DOFs. This example provides a platform for fair comparison of the reduction techniques where the merits and demerits of each can be debated. Chapter 6 discusses the complexity of operations and savings in computational time that can be expected from a system of general size when a given technique for reduction is used on it. It is remarkable that this leads to more qualitative conclusions which would be completely missed if results from merely a few applications are studied.

1.3. Notation

A major part of this work involves the use of tensors which is relevant due to the polynomial nature of nonlinearities. Tensors are multidimensional array objects. Apart from the famous Einstein summation convention which is very handy for compact notation of tensors and addresses elements of tensors using indices, another notation is proposed. This (bold) notation addresses the tensor as a whole instead of a particular of its elements using too many indices which can sometimes be confusing for an inexperienced user of Einstein convention. It also tends to be more compact as the order of tensors increases.

Bold Notation

- In quite the usual manner, the bold alphabets in lower case represent a vector (a first order tensor) and those in upper case represent a Matrix (a second order tensor).
- This notation is extended to higher order tensors by introducing a left subscript which denotes the order of the tensor for the tensors having order higher than 2. Scalars are represented in a non-bold font.
- Example : $a \in \mathbb{R}$ is a scalar, $\mathbf{a} \in \mathbb{R}^m$ represents a vector, $\mathbf{A} \in \mathbb{R}^{m \times n}$ represents a second order tensors and ${}_3\mathbf{A} \in \mathbb{R}^{m \times n \times p}$ represents a third order tensor.
- Note that a similar notation is used in literature whereby the order of the tensors is indicated by the number of wiggles

Indicial Notation

- In the Indicial notation, an element of a tensor is addressed using indices which are placed on the right subscript e.g. a_i refers to the i^{th} element of $\mathbf{a}$, A_{ij} represents the element in the i^{th}

component of first dimension and j^{th} component of the second dimension of $\mathbf{A}$, A_{ijk} represents the corresponding element of ${}_3\mathbf{A}$.

- Note that the number of indices in the right subscript indicates the order of the tensor in the indicial notation. Hence the number in the left subscript of the Bold notation is not necessary in the Indicial notation.

Apart from that, the superscripts (both left and right) are used either to differentiate two symbols or as an exponent (only right).The left superscript is generally used only when it is numerical so as to avoid confusion of it being an exponent.

Dot Product / Contraction

- For vectors the usual notion of dot product holds whereby $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T\mathbf{b}$. More generally, a dot product leads to contraction (reduction of dimensions in the resulting tensor). This is extended to tensors to allow for a contraction of m^{th} dimension of a tensor(say ${}_p\mathbf{A}$) with n^{th} dimension of another tensor (say ${}_q\mathbf{B}$) using operation $(\bullet) \cdot_{mn} (\bullet)$ as follows

$$ {}_{(p-1)+(q-1)}\mathbf{C} = {}_p\mathbf{A} \cdot_{mn} {}_q\mathbf{B} \tag{1.1} $$

This notation is also compatible with the tool [11] used for implementation of tensors in this work. It can be noticed that this operation takes away one dimension each from ${}_p\mathbf{A}$ and ${}_q\mathbf{B}$ resulting in a tensor of order $(p-1)+(q-1)$. Also, for this operation, ${}_p\mathbf{A}$ and ${}_q\mathbf{B}$ should have the same number of elements in the m^{th} and n^{th} dimension respectively (just like a usual dot product can be performed on vectors of same length). In the indicial notation Equation (1.1) is

$$ {}_3\mathbf{P} = {}_3\mathbf{Q} \cdot_{21} {}_4\mathbf{R} \tag{1.2} $$

$$ \iff P_{IJKLM} = \sum_i Q_{IiJ} R_{iKLM} \tag{1.3} $$

In the Einstein summation convention, the summation is implied over repeated indices in a term[1]. So Equation (1.3) becomes

$$ P_{IJKLM} = Q_{IiJ} R_{iKLM} \tag{1.4} $$

Einstein summation convention is always assumed hereafter unless explicitly specified.

- $(\bullet) \cdot_n (\bullet)$ would mean that one of the operating tensors is first order (i.e a vector) which is being contracted with the n^{th} dimension of the other tensor e.g.

$$ {}_3\mathbf{D} = {}_3\mathbf{E} \cdot_2 \mathbf{b} \tag{1.5} $$

$$ \iff D_{IJ} = E_{IiJ} b_i \tag{1.6} $$

- $(\bullet) \cdot (\bullet)$ for tensors simply represents the contraction of the last dimension of first tensor with the first dimension of the second tensor. Then $\mathbf{c} = \mathbf{A} \cdot \mathbf{b}$ for a matrix $\mathbf{A}$ and a vector $\mathbf{b}$ of suitable dimensions, is nothing but the matrix-vector product $\mathbf{Ab}$ (or $c_I = A_{Ii} b_i$ using Einstein convention)

$$ {}_5\mathbf{F} = {}_3\mathbf{G} \cdot {}_4\mathbf{H} \tag{1.7} $$

$$ \iff F_{IJKLM} = G_{IJi} H_{iKLM} \tag{1.8} $$

$$ c = \mathbf{a} \cdot \mathbf{b} \tag{1.9} $$

$$ \iff c = \sum_i a_i b_i = a_i b_i \tag{1.10} $$

- The $(\bullet) : (\bullet)$ operation represents contraction over two dimensions i.e. summation over two indices. By default, the last two dimensions of the first tensor are contracted with the first two dimensions

[1] Further details about tensors and Einstein convention can be found in for e.g. [12]

of the second tensor. If $\mathbf{A} \in \mathbb{R}^{m\times m}$ and $\mathbf{B} \in \mathbb{R}^{m\times m}$ are second order tensors and ${}_3\mathbf{C} \in \mathbb{R}^{n\times m\times m}$ is a third order tensor, then $\mathbf{A} : \mathbf{B}$ returns a scalar and ${}_3\mathbf{C} : \mathbf{B}$ returns a vector:

$$c = \mathbf{A} : \mathbf{B} = \sum_{i=1}^{m}\sum_{j=1}^{m} A_{ij}B_{ij} \iff A_{ij}B_{ij}, \tag{1.11}$$

$$\mathbf{a} = {}_3\mathbf{C} : \mathbf{B}, \tag{1.12}$$

$$\iff a_I = ({}_3\mathbf{C} : \mathbf{B})_I = \sum_{i=1}^{m}\sum_{j=1}^{m} C_{Iij}B_{ij} = C_{Iij}B_{ij} \tag{1.13}$$

- The $(\bullet) : (\bullet)$ operation can also be escalated to $(\bullet)\vdots(\bullet)$ and so on in case of higher order tensors.

$$d = {}_3\mathbf{K}\vdots{}_3\mathbf{L} = K_{ijk}L_{ijk}, \tag{1.14}$$

$${}_3\mathbf{M} = {}_4\mathbf{N} \vdots {}_5\mathbf{S}, \tag{1.15}$$

$$\iff M_{IJK} = N_{Iijk}B_{ijkJK} \tag{1.16}$$

- In this work, a *free* index is denoted by alphabet in upper case and a summation (or *dummy*) index is denoted in lower case.

Dyadic product/Tensor Product

- The kronecker or dyadic product is denoted by the operation $(\bullet) \otimes (\bullet)$. $(\mathbf{q} \otimes \mathbf{q})$ signifies the matrix(second order tensor) $\mathbf{q}\mathbf{q}^T$.

$$(\mathbf{q} \otimes \mathbf{q})_{IJ} = q_I q_J \qquad \forall I, J \in \{1, \ldots, m\} \tag{1.17}$$

- Not just vectors but the Kronecker product is applicable for tensors as well and the dyadic product of two tensors of orders p and q respectively, results in a tensor of order $p + q$. For, e.g.

$${}_9\mathbf{T} = {}_4\mathbf{U} \otimes {}_3\mathbf{V} \otimes \mathbf{W} \tag{1.18}$$

$$\iff T_{IJKLMNPQR} = U_{IJKL}V_{MNP}W_{QR} \tag{1.19}$$

Note that both tensor and dot products are not commutative in general since the resulting tensors can have dimensions which are permuted with respect to each other, thereby making them different. Special cases which make them commutative are when the result of such operations is a scalar and/or symmetries are involved in the arguments. These concepts and notations have been used primarily in Sections 3.3.3 and 3.4.1, and expressions have been written in both notations for clarity.

2

Nonlinear Finite Element formulation

This chapter considers the formulation of the finite element method in the context of thin walled structures. These structures are characterized by the so called 'shell' elements. Thin shells as structural elements occupy a leadership position in engineering and, in particular, in civil, mechanical, architectural, aeronautical, and marine engineering due to various mechanical advantages [1]. As possible for any general structure (continuum), here the continuous momentum balance partial differential equations (PDEs) are discretized using finite elements. The result is a large system of nonlinear equations for simulating system response to loading. First the governing equations are presented, then the finite elements are formulated for thin walled structures.

2.1. Equilibrium and Weak Form

The balance of momentum for a body V with surface boundary S being acted upon by body forces $\mathbf{b}$ (such as gravity) and surface traction $\mathbf{t}$, is given by (See for e.g. [12])

$$\int_V \rho\ddot{\mathbf{u}}\ \mathrm{d}V = \int_V \rho\mathbf{b}\ \mathrm{d}V + \int_S \mathbf{t}\ \mathrm{d}S \tag{2.1}$$

ρ being the mass density and $(\dot{\bullet})$ denoting time derivative. Using the stress tensor $\boldsymbol{\sigma}$, Equation (2.1) can be written as

$$\int_V \rho\ddot{\mathbf{u}}\ \mathrm{d}V = \int_V \rho\mathbf{b}\ \mathrm{d}V + \int_S \boldsymbol{\sigma}\cdot\mathbf{n}\ \mathrm{d}S \tag{2.2}$$

Applying Gauß' divergence[1] theorem,

$$\int_V \left(\nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{b} - \rho\ddot{\mathbf{u}}\right)\ \mathrm{d}V = \mathbf{0} \tag{2.3}$$

Since this balance should be applicable for any arbitrary V, the following strong form of governing equations can be obtained.

$$\nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{b} - \rho\ddot{\mathbf{u}} = \mathbf{0} \tag{2.4}$$

Due to the symmetry of the stress tensor $\boldsymbol{\sigma}$ [12], only 6 components are needed and the Voigt notation can be used to represent the tensor as a vector [13]. Then the strong form in Equation (2.4) can be rewritten as

$$\mathbf{L}^T\boldsymbol{\sigma} + \rho\mathbf{b} = \rho\ddot{\mathbf{u}}, \tag{2.5}$$

[1] A generalised notion of divergence has been used here $(\nabla\cdot\boldsymbol{\sigma})$, which is extended to tensor fields from the well known case of vector fields. See e.g. [12].

where

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} \qquad \mathbf{L}^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x}. \end{bmatrix}$$

The weak form can be obtained by projecting Equation (2.5) to an admissible virtual displacement field $\delta\mathbf{u}$ and integrating over the domain V, resulting in the principle of virtual work as follows

$$\int_V \delta\mathbf{u}^T \left(\mathbf{L}^T\boldsymbol{\sigma} + \rho\mathbf{b} - \rho\ddot{\mathbf{u}}\right) \,\mathrm{d}V = \mathbf{0}. \tag{2.6}$$

With prescribed boundary conditions

$$\begin{aligned} \boldsymbol{\sigma}\mathbf{n} = \mathbf{t} \qquad & \text{on} \;\; S_{natural} \\ \mathbf{u} = \mathbf{u}_p, \quad \delta\mathbf{u} = \mathbf{0} \qquad & \text{on} \;\; S_{essential} \end{aligned}$$

such that $S_{natural} \cup S_{essential} = S$ and $S_{natural} \cap S_{essential} = \emptyset$, and the initial conditions

$$\begin{aligned} \mathbf{u}(t_0) = \mathbf{u}_0 \qquad & \text{in} \;\; V \\ \dot{\mathbf{u}}(t_0) = \dot{\mathbf{u}}_0 \qquad & \text{in} \;\; V, \end{aligned}$$

and using integration by parts on the first term of Equation (2.6) and subsequently applying Gauß' divergence theorem yields the weak form

$$\int_V \left(\delta\mathbf{u}^T\rho\ddot{\mathbf{u}} + (\mathbf{L}\delta\mathbf{u})^T\boldsymbol{\sigma}\right) \,\mathrm{d}V = \int_V \delta\mathbf{u}^T\rho\mathbf{b} \,\mathrm{d}V + \int_S \delta\mathbf{u}^T\mathbf{t} \,\mathrm{d}S, \tag{2.7}$$

where $\mathbf{L}\delta\mathbf{u}$ represents the gradient : $\nabla\delta\mathbf{u}$ [2]. Equation (2.7) is also known as the principle of virtual work in literature, with LHS representing the virtual work done by inertial and internal forces, and the RHS representing the virtual work by the external forces. In a quasi-static case, the time derivatives (inertial effects) disappear and the principle of virtual work is written as

$$\delta W_{int} = \delta W_{ext}, \tag{2.8}$$

$$\text{with} \qquad \delta W_{int} = \int_V (\mathbf{L}\delta\mathbf{u})^T\boldsymbol{\sigma} \,\mathrm{d}V, \;\text{ and} \tag{2.9}$$

$$\delta W_{ext} = \int_V \delta\mathbf{u}^T\rho\mathbf{b} \,\mathrm{d}V + \int_S \delta\mathbf{u}^T\mathbf{t} \,\mathrm{d}S. \tag{2.10}$$

Note that the strain tensor $\boldsymbol{\varepsilon}$ is energetically conjugate to the Cauchy stress $\boldsymbol{\sigma}$ and must satisfy

$$\delta W_{int} = \int_V \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} \,\mathrm{d}V, \tag{2.11}$$

and thus it follows that $\delta\mathbf{u} = \delta\boldsymbol{\varepsilon}$. The weak form can then also be written as

$$\int_V \left(\delta\mathbf{u}^T\rho\ddot{\mathbf{u}} + \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma}\right) \,\mathrm{d}V = \int_V \delta\mathbf{u}^T\rho\mathbf{b} \,\mathrm{d}V + \int_S \delta\mathbf{u}^T\mathbf{t} \,\mathrm{d}S. \tag{2.12}$$

Up to this point, the formulation is completely general. In a displacement based formulation, we use the nodal displacements as fundamental unknowns, and approximate the displacement field inside an element as follows

$$\mathbf{u}_e = \mathbf{H}(\boldsymbol{\xi})\mathbf{a}_e \tag{2.13}$$

[2] A generalised notion of gradient ($\nabla\delta\mathbf{u}$) has been used here. The operation is extented to vector fields from the well known case of scalars. See e.g. [12]

where $\mathbf{H}(\boldsymbol{\xi})$ are the interpolation (shape) functions, N_e is the number of elemental DOFs, $\boldsymbol{\xi}$ can be the isoparametric general coordinates and $\mathbf{a}_e$ are the unknown coefficients for the interpolation functions at element level. These elemental unknowns are related to the global nodal unknowns as

$$\mathbf{a}_e = \mathbf{Z}_e \mathbf{a}, \tag{2.14}$$

where $\mathbf{Z}_e \in \mathbb{R}^{3N_e \times n}$ is typically known for every element by the topology of the structure (also sometimes referred to as the *assembly operator*), n being the number of global degrees of freedom. Then the following semi discrete equations can be extracted from the weak form Equation (2.7) (taking into account that it is applicable for any arbitrary admissible displacement $\delta\mathbf{u}$)

$$\mathbf{M}\ddot{\mathbf{a}} = \mathbf{f}_{ext} - \mathbf{f}_{int}, \tag{2.15}$$

where

$$\mathbf{M} = \sum_{e=1}^{n_e} \mathbf{Z}_e^T \underbrace{\int_{V_e} \rho \mathbf{H}^T \mathbf{H} \, \mathrm{d}V}_{\mathbf{M}_e} \mathbf{Z}_e, \tag{2.16}$$

$$\mathbf{f}_{ext} = \sum_{e=1}^{n_e} \mathbf{Z}_e^T \left(\int_{V_e} \rho \mathbf{H}^T \mathbf{b} \, \mathrm{d}V + \int_{S_e} \mathbf{H}^T \mathbf{t} \, \mathrm{d}S \right), \text{and} \tag{2.17}$$

$$\mathbf{f}_{int} = \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{V_e} \underbrace{(\mathbf{LH})^T}_{\mathbf{B}^T} \boldsymbol{\sigma} \, \mathrm{d}V. \tag{2.18}$$

Till this point, no assumption on material behavior has been made. The role of geometrical nonlinearities shall be shown next.

2.2. Geometric Nonlinearities and Von-Karman kinematics

As stated before, the thin walled structures are characterised by geometric nonlinearities due to finite, out of plane displacements while the material still stays in the linear range. Then, a linear constitutive relation exists between the stresses and strains.

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{2.19}$$

$$\tag{2.20}$$

For the Green Lagrange strain definition [12], the strain is composed of linear and quadratic components in $\nabla\mathbf{u}$ ($\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{u}^T + \nabla\mathbf{u} + \nabla\mathbf{u}^T \cdot \nabla\mathbf{u})$). These quadratic components constitute the so called geometric nonlinearities. Geometric nonlinearities in the most general form make the equations cumbersome to implement, and may not provide the best results (e.g. in context of postbuckling curvature [14]). In the context of thin walled structures, scientist have continuously developed approximate theories to capture nonlinear behavior. In 1850, Kirchhoff developed the *linear* plate bending theory in which he stated two basic independent assumptions (commonly referred to as the "Kirchhoff's hypothesis") where by the the deflection of the mid-plane is small compared with the thickness of the plate ($w \leq 0.2t$) and sections which are straight and normal to mid-plane before bending remain so after bending also.This resulted in reduction of 3D plate problem into 2D one (See for e.g. [1]).

The Kirchhoff's linear theory ignores strains in the mid-surface of the plate and corresponding in-plane stresses are neglected. However, if the magnitude of out of plane displacement increases beyond a certain level ($w \geq 0.3t$), the mid-surface starts stretching and producing in-plane stresses (membrane forces). As the $|w/t|$ ratio increases, the membrane action increases and as $|w/t| \approx 1$, the membrane action becomes comparable to that of bending beyond which it predominates bending. Thus, while constructing large deflection plate theory, the assumption of absence of membrane deformations is dropped [1].

Different kinematic models result in shell theories of varying degrees of accuracy to capture nonlinear behavior of plates. Generally, large deflection theories assume that deflections can be comparable to the plate thickness but still relatively smaller than other dimensions. Large displacement theory of thin plates and shells is based on the Von-Karman kinematic model in which terms $u,_x^2$, $v,_y^2$ are assumed to be negligible. This completely neglects the nonlinear in-plane rotation terms which could be important for assembly of flat plates or curved shells if not flat plates. Tiso [14] showed a way to simplify the Green-Lagrange strain tensor which captures the in-plane nonlinear effects for plates effectively (in the context of perturbation expansion for post buckling analysis)

$$\begin{aligned} \varepsilon_x &= u,_x + \frac{1}{2}(v,_x^2 + w,_x^2), \\ \varepsilon_y &= v,_y + \frac{1}{2}(u,_y^2 + w,_y^2), \\ \varepsilon_{xy} &= \frac{1}{2}(v,_x + u,_y) + \frac{1}{2}(w,_x w,_y). \end{aligned} \tag{2.21}$$

2.3. Shell Finite Elements

For thin plates/arches, shell elements are used to model the structure. Considerable savings in computer time can be gained by resorting to shell elements because a depth (thickness) integration is done directly instead of numerical integration in 3D, thereby reducing the degrees of freedom. The resulting formulation employs the membrane strain ($\boldsymbol{\varepsilon}_l$) and the curvature ($\boldsymbol{\chi}$) along with the corresponding stress resultants, **N** and **M** as shown in [13].

Tiso [14] implemented a triangular 3-noded flat shell element based on the material stiffness matrix for the membrane and bending element, formulated by Allman in [15] and [16] respectively. This formulation shall be used in this work. The element degrees of freedom (or generalized displacements) are defined as

$$\mathbf{a}_e = [\mathbf{a}_1^e \quad \mathbf{a}_2^e \quad \mathbf{a}_3^e], \text{ with} \tag{2.22}$$

$$\mathbf{a}_i^e = [u_i \quad v_i \quad w_i \quad \theta_{x_i} \quad \theta_{y_i} \quad \theta_{z_i}] \quad \forall i \in \{1, 2, 3\}, \tag{2.23}$$

where θ represents the rotational degree of freedom about the subscripted axis at node i of the element e. The nonlinear strain displacement relation within an element is written as

$$\boldsymbol{\varepsilon}_e = \mathbf{B}_L \mathbf{a}_e + \frac{1}{2}\mathbf{B}_{NL}(\mathbf{a}_e)\mathbf{a}_e, \tag{2.24}$$

such that $\mathbf{B}_L$ is nothing but **LH** and $\mathbf{B}_{NL}(\mathbf{u})$ is linear in $\mathbf{u}$. Following the details given in [14], the element level internal force as given in Appendix A.1 can be obtained. It can be seen that this force has upto third order contribution of displacement unknowns.

3

Model Order Reduction

After the spatial discretization of the governing PDEs using finite elements as shown in the previous chapter, a system of second order ODEs is obtained which can be written in the form of following IVP

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{f}(\mathbf{u}(t)) &= \mathbf{g}(t)\\ \mathbf{u}(t_0) &= \mathbf{u}_0\\ \dot{\mathbf{u}}(t_0) &= \mathbf{v}_0,\end{aligned} \tag{3.1}$$

where the solution $\mathbf{u}(t) \in \mathbb{R}^n$ is a high dimensional generalised displacement vector, $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the mass matrix, $\mathbf{C} \in \mathbb{R}^{n\times n}$ is the damping matrix, $\mathbf{f}(\mathbf{u}) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the nonlinear internal force and $\mathbf{g}(t) \in \mathbb{R}^n$ is the time dependent external load vector. Note that the damping term $\mathbf{C}\dot{\mathbf{u}}$ was not shown while describing the finite element formulation. Only the nonlinear elastic internal forces were considered there. In general, structural damping is characterized by such a linear damping term. These equations are time discretized using suitable time integration scheme (to be explained later) resulting in high dimensional fully discrete equations which are very expensive to solve. Model order reduction aims to reduce dimension of the model to reduce solution time.

3.1. Galerkin Projection

In structural dynamics especially for narrow spectrum loading, the response of high dimensional system is (approximately) contained in a low dimensional subspace (say $\mathcal{V}$). The solution can be written as a linear combination of vectors spanning the subspace which reduces the number of unknowns to the number of vectors in subspace. These vectors form a basis (ROB) for $\mathcal{V}$. In projection based MOR, displacement field $\mathbf{u}$ is projected onto a suitable ROB $\mathbf{V}$ of time independent vectors as:

$$\begin{aligned}\mathbf{V} &= [\mathbf{v}_1 \;\; \mathbf{v}_2 \ldots \mathbf{v}_m], & \mathbf{v}_i &\in \mathbb{R}^n \;\forall\; i \in \{1, 2, \ldots, m\}\\ \mathbf{u}(t) &\approx \mathbf{V}\mathbf{q}(t) & \mathbf{V} &\in \mathbb{R}^{n\times m},\;\; \mathbf{q}(t) \in \mathbb{R}^m\end{aligned}$$

where $\mathbf{q}(t)$ is the new time dependent vector of unknowns which has size $m \ll n$. Introducing this mapping creates an approximation to the exact solution and when substituted in the governing Equation (3.1) results in a residual error $\mathbf{r}$

$$\mathbf{M}\mathbf{V}\ddot{\mathbf{q}}(t) + \mathbf{C}\mathbf{V}\dot{\mathbf{q}}(t) + \mathbf{f}(\mathbf{V}\mathbf{q}(t)) + \mathbf{r}(t) = \mathbf{g}(t) \tag{3.2}$$

The residual is then constrained to be orthogonal to a subspace ($\mathcal{T}$) spanned by the basis $\mathbf{T} \in \mathbb{R}^{n\times m}$:

$$\mathbf{T}^T\mathbf{r}(t) = \mathbf{0} \tag{3.3}$$

Such a projection of the equations where $\mathbf{T} \neq \mathbf{V}$ leads the so called *Petrov-Galerkin* projection where one obtains the reduced set of equations

$$\mathbf{T}^T\mathbf{M}\mathbf{V}\ddot{\mathbf{q}}(t) + \mathbf{T}^T\mathbf{C}\mathbf{V}\dot{\mathbf{q}}(t) + \mathbf{T}^T\mathbf{f}(\mathbf{V}\mathbf{q}(t)) = \mathbf{T}^T\mathbf{g}(t) \tag{3.4}$$

But if the projection basis is taken to be the same as the mapping basis i.e. $\mathbf{T} = \mathbf{V}$, it leads to the *Bubnov-Galerkin* or simply the *Galerkin Projection.*

$$\underbrace{\mathbf{V}^T\mathbf{M}\mathbf{V}}_{\tilde{\mathbf{M}}}\ddot{\mathbf{q}}(t) + \underbrace{\mathbf{V}^T\mathbf{C}\mathbf{V}}_{\tilde{\mathbf{C}}}\dot{\mathbf{q}}(t) + \underbrace{\mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{q}(t))}_{\tilde{\mathbf{f}}(\mathbf{q}(t))} = \mathbf{V}^T\mathbf{g}(t), \tag{3.5}$$

where $\tilde{\mathbf{M}}$, $\tilde{\mathbf{C}} \in \mathbb{R}^{m\times m}$ are the reduced mass and damping matrices respectively. Sometimes the nonlinear internal force $\mathbf{f}(\mathbf{u})$ is explicitly broken in linear and nonlinear contributions, to obtain a reduced stiffness matrix as well.

$$\underbrace{\mathbf{V}^T\mathbf{M}\mathbf{V}}_{\tilde{\mathbf{M}}}\ddot{\mathbf{q}}(t) + \underbrace{\mathbf{V}^T\mathbf{C}\mathbf{V}}_{\tilde{\mathbf{C}}}\dot{\mathbf{q}}(t) + \underbrace{\mathbf{V}^T\mathbf{K}\mathbf{V}}_{\tilde{\mathbf{K}}(\mathbf{q}(t))}\mathbf{q}(t) + \mathbf{V}^T\mathbf{f}^{nl}(\mathbf{V}\mathbf{q}(t)) = \mathbf{V}^T\mathbf{g}(t), \tag{3.6}$$

It is easy to see that the nonlinearity evaluation $\mathbf{f}^{nl}(\mathbf{V}\mathbf{q}(t)$ and the tangent stiffness $\mathbf{K}$ assembly need to be performed at each iteration during time integration of these equations. The sparse assembly of these matrices (or vectors) is a significant cost and redundant especially when all that is needed is the reduced version of these matrices (or vectors). This can be done in an efficient manner as follows and is referred to as the *regular* approach in this work.

$$\tilde{\mathbf{f}}(\mathbf{q}) = \mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{q}) = \sum_{e=1}^{n_e}\mathbf{V}_e^T\mathbf{f}_e(\mathbf{V}_e\mathbf{q}), \tag{3.7}$$

$$\tilde{\mathbf{K}}(\mathbf{q}) = \mathbf{V}^T\mathbf{K}(\mathbf{V}\mathbf{q})\mathbf{V} = \sum_{e=1}^{n_e}\mathbf{V}_e^T\mathbf{K}_e(\mathbf{V}_e\mathbf{q})\mathbf{V}_e, \tag{3.8}$$

where $\mathbf{f}_e(\mathbf{u_e}) \in \mathbb{R}^{N_e}$ and $\mathbf{K}_e(\mathbf{u}_e)\mathbb{R}^{N_e\times N_e}$ are the contributions of the element e towards the vector $\mathbf{f}(\mathbf{u})$ and the matrix $\mathbf{K}(\mathbf{u})$ respectively, and $\mathbf{V}_e$ is the restriction of $\mathbf{V}$ to the rows indexed by DOFs corresponding to element e which contains N_e DOFs.

The choice of projection basis $\mathbf{V}$ is very critical in determining the accuracy of the reduced solution. The size of the basis is important in determining the speed-up in computation time.

3.2. POD

Proper Orthogonal Decomposition(POD) is a very effective method to construct a low dimensional subspace. The method is known by many names such as Karhunen Loeve Transform (KLT) in digital signal processing, Principal Component Analysis (PCA) in statistics or the more familiar linear algebra term - Singular Value Decomposition (SVD). Essentially, if the full nonlinear response of a system is known for a given timespan, then a low dimensional subspace in which the solution lies, can be constructed by taking an ensemble of solution vectors at different time instants and choosing the most significant singular vectors of the ensemble after doing an SVD.

Let $\mathbf{U} = [\mathbf{u}_1 \;\; \mathbf{u}_2 \ldots \mathbf{u}_{n_s}] \in \mathbb{R}^{n\times n_s}$ be the ensemble of snapshots of rank $r(< n)$ which is obtained from the full solution. A lower dimensional basis $\mathbf{V} = [\mathbf{v}_1 \;\; \mathbf{v}_2 \ldots \mathbf{v}_m] \in \mathbb{R}^{n\times m}$ containing $m \ll n_s$ orthogonal vectors which 'best' span the vectors in this ensemble can be obtained by the solution to the following minimization problem.

$$\min_{\mathbf{v}_i\in\mathbb{R}^n}\sum_{j=1}^{n_s}\left\|\mathbf{u}_j - \sum_{i=1}^{m}(\mathbf{u}_j^T\mathbf{v}_i)\mathbf{v}_i\right\|_2^2 \tag{3.9}$$

This is a least squares problem and the vectors in $\mathbf{V}$ are nothing but the left singular vectors of $\mathbf{U}$ (eigenvectors of $\mathbf{U}\mathbf{U}^T$). The following SVD problem can be solved.

$$\begin{aligned} &\mathbf{U} = \mathbf{A}\mathbf{\Sigma}\mathbf{B}^T && \mathbf{A} = [\mathbf{a}_1 \;\; \mathbf{a}_2 \ldots \mathbf{a}_n] \in \mathbb{R}^{n\times n},\; \mathbf{B} \in \mathbb{R}^{n_s\times n_s} \\ &\mathbf{\Sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1^2 & & \emptyset & \\ & \ddots & & \\ & & \boldsymbol{\sigma}_r^2 & \\ & \emptyset & & \emptyset \end{bmatrix} \in \mathbb{R}^{n\times n_s}, && \boldsymbol{\sigma}_1^2 \geq \boldsymbol{\sigma}_2^2 \geq \cdots \geq \boldsymbol{\sigma}_r^2 > 0 \end{aligned} \tag{3.10}$$

The vectors in the reduction basis of $\mathbf{V}$ can be obtained by the selection of first $m < r$ column vectors from $\mathbf{A}$. The singular values $(\boldsymbol{\sigma}_1^2, \ldots, \boldsymbol{\sigma}_r^2)$ of $\mathbf{U}$ give represent the relative importance of corresponding vectors of $\mathbf{A}$ in forming the ROB $\mathbf{V}$. It can be shown that

$$\sum_{j=1}^{n_s} \left\| \mathbf{u}_j - \sum_{i=1}^{m} (\mathbf{u}_j^T \mathbf{a}_i)\mathbf{a}_i \right\|_2^2 = \sum_{i=m+1}^{r} \boldsymbol{\sigma}_i^2 \tag{3.11}$$

Thus POD gives an important tool to select the most significant vectors for construction of ROB to be used during Galerkin projection. POD is a very versatile method applicable applicable for general nonlinear problems. However, one of the disadvantages of such a ROB is that it is applicable only for a solution which is characteristic of the applied loading and a new basis would be required to take into account other types of loading.

3.3. Linear Manifold

Versatility of POD or hyper-reduction (DEIM/ECSW) kind methods (as would be discussed in Chapter 4) is quite large since they're independent of the nature of non-linearities involved in the system. However, one of the disadvantages of these methods (at least when applied in the conventional way) is the need for training snapshots of solution vectors which are obtained from a full non-linear run. One could argue that it is pointless to form a ROM for a system whose full solution is available. Nonetheless, such ROMs are still useful for running optimization routines for a given load (or similar) load case.

One of the question that arises then is that if it's possible to exploit the structure of nonlinearities underlying in the governing equations to obtain a ROM without the need of a full nonlinear run. It would be interesting if such a ROM could accurately provide response for different load cases which are in the range of applicability of the underlying theory (Von-Karman Kinematics in this case).

Vibration Modes: It is well known that for a linear system, such as,

$$\mathbf{M}\ddot{\mathbf{y}}(t) + \mathbf{K}\mathbf{y}(t) = \mathbf{g}(t), \tag{3.12}$$

$$\mathbf{y}(0) = \mathbf{y}_0 \tag{3.13}$$

$$\dot{\mathbf{y}}(0) = \mathbf{v}_0 \tag{3.14}$$

the system response can be written as a linear combination of constant eigenvectors(also referred to as the eigenmodes or Vibration modes in the structural dynamics context) which form a basis of $\mathbb{R}^n$ as follows.

$$\mathbf{y}(t) = \sum_{i=1}^{n} \boldsymbol{\phi}_i q_i(t) \tag{3.15}$$

where the eigenmodes $\boldsymbol{\phi}_i \in \mathbb{R}^n$ are found by solution to the generalized eigenvalue problem

$$(\mathbf{K} - \omega_i^2 \mathbf{M})\boldsymbol{\phi}_i = \mathbf{0}, \tag{3.16}$$

and ω_i^2 is the eigenvalue (or eigenfrequency squared). This concept of expressing the solution $\mathbf{y}(t)$ in terms of a basis of eigenvectors is referred to as the principle of *linear modal superposition*. However, if one is considering the slowly varying dynamics of the system, then it can be shown that the response can be very accurately approximated by a few low frequency modes and a modal truncation can be obtained [2].

$$\mathbf{y}(t) \approx \sum_{i=1}^{m} \boldsymbol{\phi}_i q_i(t) = \boldsymbol{\Phi}\mathbf{q}(t), \tag{3.17}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n \times m}$, $\mathbf{q}(t) = [q_1(t)\ q_2(t)\ \cdots q_m(t)]^T \in \mathbb{R}^m, m \ll n$. Thus in doing so, we introduce a mapping $\mathbf{y} : \mathbb{R}^n \longmapsto \mathbb{R}^m$ such that $\mathbf{y} = \mathbf{y}(\mathbf{q}) = \boldsymbol{\Phi}\mathbf{q}$. Since $m \ll n$, this reduces the number of unknowns in the system and an effective ROM is obtained for linear systems. The time dependency in variables omitted for clarity reasons from here onwards.

Modal Derivatives: When nonlinearities are present such as in

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{f}(\mathbf{u}) = \mathbf{g}(t) \tag{3.18}$$

then a linearized system might me constructed around the equilibrium position ($\mathbf{u} = \mathbf{u}_{eq} := \mathbf{0}$) to obtain,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \underbrace{\left.\frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{0}}}_{\text{Tangent Stiffness } \mathbf{K}(\mathbf{u}=\mathbf{0})} \mathbf{u} = \mathbf{g}(t). \tag{3.19}$$

This linearised system can be a good approximation to the original system in Equation (3.18) for small displacements from the linearisation point. The deviation from linearisation point can then be expressed using a modal superposition [1]

$$\mathbf{u}(\mathbf{q}) : \mathbb{R}^n \mapsto \mathbb{R}^m \tag{3.20}$$

$$\mathbf{u} \approx \mathbf{u}_{eq} + \boldsymbol{\Phi}|_{eq}\mathbf{q}, \tag{3.21}$$

where $\boldsymbol{\Phi}_{eq}$ is the matrix containing the relevant eigenmodes evaluated at the equilibrium/linearisation point [2]. However, when nonlinear effects start increasing, the corresponding basis of eigenmodes is not effective any more due to departure from linear behaviour. Equation (3.21) can be assumed to address only first order effects in the nonlinear regime such that $\left.\frac{\partial \mathbf{u}}{\partial \mathbf{q}}\right|_{eq} = \boldsymbol{\Phi}|_{eq}$. Higher order effects can then be considered as shown in [7] and [8] and the lower frequency modes (or the basis $\boldsymbol{\Phi}$) can be assumed to be dependent on displacement along other modes also and

$$\mathbf{u} \approx \mathbf{u}_{eq} + \boldsymbol{\Phi}(\mathbf{q})\mathbf{q} \tag{3.22}$$

such that

$$\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}} = \boldsymbol{\Phi}(\mathbf{q}) \tag{3.23}$$

Assuming such an implicit dependence on modal amplitudes $\mathbf{q}$, a Taylor expansion around equilibrium position i.e. $\mathbf{q} = \mathbf{0}$ can be applied,

$$\mathbf{u}(\mathbf{q}) = \cancelto{\mathbf{u}_{eq}}{\mathbf{u}(\mathbf{0})} + \left.\frac{\partial \mathbf{u}}{\partial \mathbf{q}}\right|_{\mathbf{q}=\mathbf{0}} \cdot \mathbf{q} + \frac{1}{2!}\left.\frac{\partial^2 \mathbf{u}}{\partial \mathbf{q}\partial \mathbf{q}}\right|_{\mathbf{q}=\mathbf{0}} : (\mathbf{q} \otimes \mathbf{q}) + \dots \tag{3.24}$$

The partial derivatives of $\mathbf{u}(\mathbf{q})$ can be computed from Equation (3.23).

$$\frac{\partial \mathbf{u}}{\partial q_j} = \boldsymbol{\phi}_j \tag{3.25}$$

$$\implies \frac{\partial^2 \mathbf{u}}{\partial q_i \partial q_j} = \frac{\partial \boldsymbol{\phi}_j}{\partial q_i} =: \boldsymbol{\theta}_{ij} \tag{3.26}$$

where $\boldsymbol{\theta}_{ij} \in \mathbb{R}^n$, and evaluating these at equilibrium position means

$$\left.\frac{\partial \mathbf{u}}{\partial q_j}\right|_{\mathbf{q}=\mathbf{0}} = \boldsymbol{\phi}_j|_{eq} \tag{3.27}$$

$$\left.\frac{\partial^2 \mathbf{u}}{\partial q_i \partial q_j}\right|_{\mathbf{q}=\mathbf{0}} = \left.\frac{\partial \boldsymbol{\phi}_j}{\partial q_i}\right|_{eq} = \boldsymbol{\theta}_{ij}|_{eq} \tag{3.28}$$

Then from Equations (3.24) and (3.27) , its easy to see that the Vibration Modes (VMs) contribute towards the linear part of solution ($\mathbf{u}$) and Equations (3.24) and (3.28) imply that the Modal Derivatives (MDs) (i.e. $\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}$) capture the second order non-linear effects in $\mathbf{q}$. Physically, the MD $\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}$ represent the change in the VM $\boldsymbol{\phi}_i$ corresponding to a displacement given in the direction of VM $\boldsymbol{\phi}_j$ (i.e. $\mathbf{u} = q_j \boldsymbol{\phi}_j$). Thus as suggested in [7], the MDs form a nice augmentation of the reduction basis and the augmented basis can be formed as

$$\boldsymbol{\Psi} = [\boldsymbol{\phi}_1|_{eq}\ \boldsymbol{\phi}_2|_{eq}\ \dots\ \boldsymbol{\phi}_m|_{eq}\ \dots\ \boldsymbol{\theta}_{ij}|_{eq} \dots] \tag{3.29}$$

It is easy to see from Equation (3.26) that $\boldsymbol{\theta}_{ij} = \boldsymbol{\theta}_{ji}$. Thus only unique components of $\boldsymbol{\theta}_{ij}$ should be used in augmenting the basis of VMs and a basis $\boldsymbol{\Psi} \in \mathbb{R}^{n \times M}$ can be obtained, where $M = m + \frac{m(m+1)}{2}$.

[1] Note that for damped linear systems with low damping or *modal/Rayleigh* damping as explained in [2], the eigenvectors for an undamped system still form a good basis for linear modal superposition. Such damping is very popular in structural dynamics and shall be used here.

[2] eigenmodes become configuration dependent for a nonlinear system since the elastic stiffness matrix ($\mathbf{K}$) which is the Jacobian of the nonlinear internal force, becomes configuration dependent in general

3.3.1. Calculation of Modal Derivatives

The calculation of MDs is not trivial and is done by differentiating Equation (3.16) w.r.t. modal amplitude q_j (assuming $\mathbf{M}$ to be a constant mass matrix)

$$(\mathbf{K} - \omega_i^2 \mathbf{M}) \frac{\partial \boldsymbol{\phi}_i}{\partial q_j} + \left(\frac{\partial \mathbf{K}}{\partial q_j} - \frac{\partial \omega_i^2}{\partial q_j} \mathbf{M} \right) \boldsymbol{\phi}_i = \mathbf{0} \tag{3.30}$$

This equation is then evaluated at equilibrium (i.e. $\mathbf{q} = \mathbf{0}$).

$$(\mathbf{K}|_{eq} - \omega_i^2|_{eq} \mathbf{M}) \left.\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}\right|_{eq} + \left(\left.\frac{\partial \mathbf{K}}{\partial q_j}\right|_{eq} - \left.\frac{\partial \omega_i^2}{\partial q_j}\right|_{eq} \mathbf{M} \right) \boldsymbol{\phi}_i|_{eq} = \mathbf{0} \tag{3.31}$$

Here the tangent stiffness matrix derivative w.r.t. q_j is obtained by giving the structure a displacement in the direction of $\boldsymbol{\phi}_j$ i.e.

$$\left.\frac{\partial \mathbf{K}}{\partial q_j}\right|_{eq} = \left.\frac{\partial \mathbf{K}(\mathbf{u} = q_j \boldsymbol{\phi}_j))}{\partial q_j}\right|_{q_j=0} \tag{3.32}$$

It is easy to see that $\left.\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}\right|_{eq}$ cannot be trivially obtained from Equation (3.31) since the coefficient matrix is singular by definition (Equation (3.16)) . This singularity can be dealt by imposing a normalization condition for the eigenmodes. [17] covers an extensive account of different solution techniques and introduces a generalised approach to find eigenvector derivatives for different kinds of normalizations. The popular mass normalization has been adopted here i.e.

$$\boldsymbol{\phi}_i^T \mathbf{M} \boldsymbol{\phi}_i = 1 \qquad \forall i \in \{1, 2, \ldots, m\}. \tag{3.33}$$

Differentiating the above equation w.r.t. the modal amplitude results in

$$\boldsymbol{\phi}_i^T \mathbf{M} \frac{\partial \boldsymbol{\phi}_i}{\partial q_j} + \boldsymbol{\phi}_i^T \mathbf{M}^T \frac{\partial \boldsymbol{\phi}_i}{\partial q_j} = 0 \qquad \forall i, j \in \{1, 2, \ldots, m\}. \tag{3.34}$$

Exploiting the symmetry of $\mathbf{M}$ and subsequent evaluation at the equilibrium position results in the following relation

$$\boldsymbol{\phi}_i^T|_{eq} \mathbf{M} \left.\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}\right|_{eq} = 0 \qquad \forall i, j \in \{1, 2, \ldots, m\}. \tag{3.35}$$

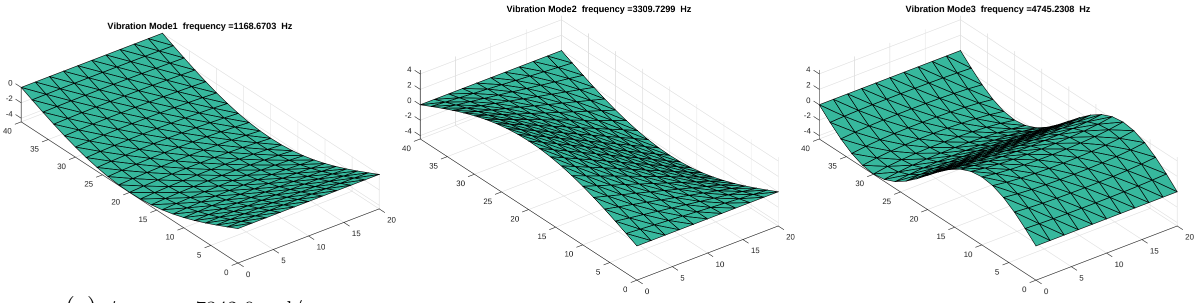
The following **direct** approach to calculate the MDs can then be formulated using Equations (3.31) and (3.35)

$$\begin{bmatrix} [\mathbf{K}|_{eq} - \omega_i^2|_{eq} \mathbf{M}]_{n \times n} & -[\mathbf{M} \boldsymbol{\phi}_i|_{eq}]_{n \times 1} \\ -[\mathbf{M} \boldsymbol{\phi}_i|_{eq}]^T_{1 \times n} & 0_{1 \times 1} \end{bmatrix} \begin{bmatrix} \left.\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}\right|_{eq} \\ \left.\frac{\partial \omega_i^2}{\partial q_j}\right|_{eq} \end{bmatrix} = \begin{bmatrix} -\left.\frac{\partial \mathbf{K}}{\partial q_j}\right|_{eq} \boldsymbol{\phi}_i|_{eq} \\ 0 \end{bmatrix}$$

The above non-singular system can be used to solve for modal derivatives. This approach is not very attractive since it destroys the band structure of the original system. Nonetheless, it is rigorous and accurate and has been used here. Apart from this direct approach, the Nelson's method [18] is also popular in literature which preserves the band structure of the matrices.

Though the above mentioned techniques are exact ways to calculate the modal derivatives, they're costly since a high dimensional matrix needs to be factorized. [8] discusses a way to approximate these derivatives by neglecting the inertial contribution simply as

$$\mathbf{K}|_{eq} \left.\frac{\partial \boldsymbol{\phi}_i}{\partial q_j}\right|_{eq} = -\left.\frac{\partial \mathbf{K}}{\partial q_j}\right|_{eq} \boldsymbol{\phi}_i|_{eq} \tag{3.36}$$

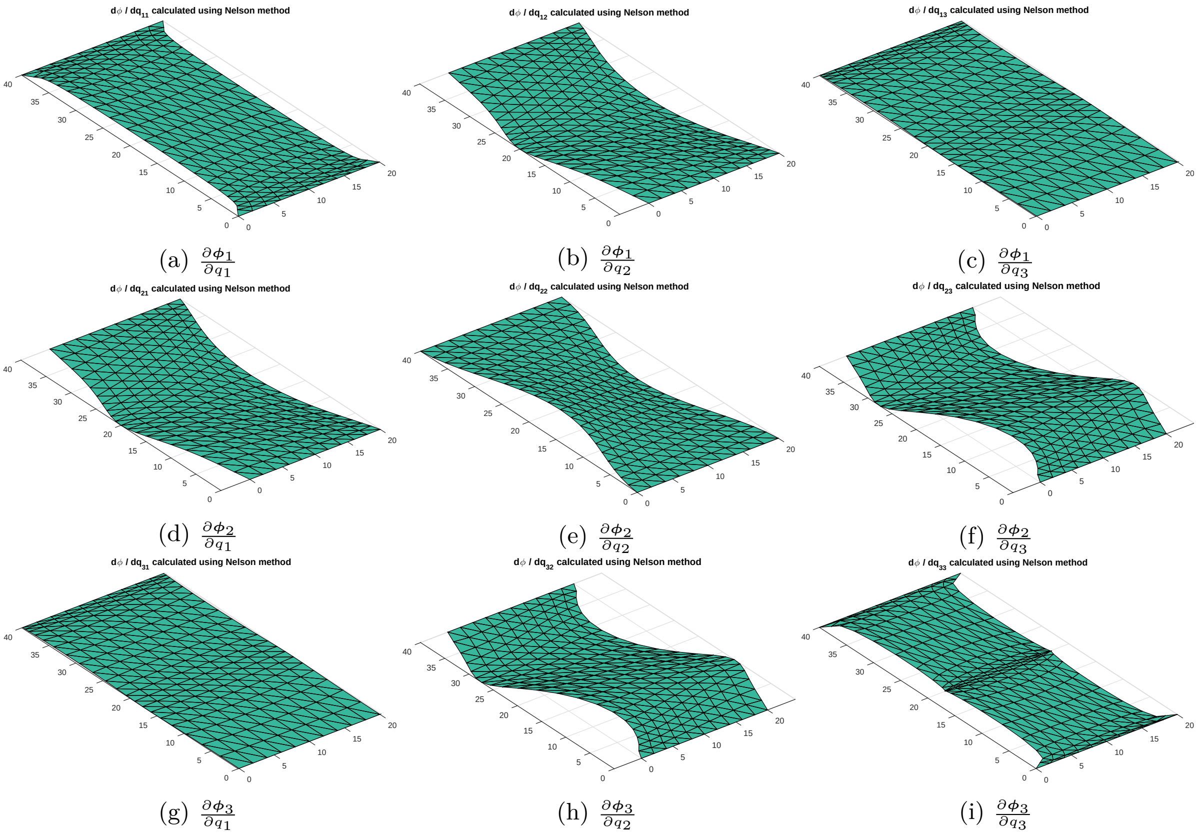


(a) $\boldsymbol{\phi}_1$, $\omega_1 = 7342.9$ rad/sec

(b) $\boldsymbol{\phi}_1$, $\omega_1 = 20795.6$ rad/sec

(c) $\boldsymbol{\phi}_1$, $\omega_1 = 29815.1$ rad/sec

Figure 3.1: The first three vibration modes (VMs) for a rectangular plate simply supported on two opposite sides.



(a) $\frac{\partial \boldsymbol{\phi}_1}{\partial q_1}$

(b) $\frac{\partial \boldsymbol{\phi}_1}{\partial q_2}$

(c) $\frac{\partial \boldsymbol{\phi}_1}{\partial q_3}$

(d) $\frac{\partial \boldsymbol{\phi}_2}{\partial q_1}$

(e) $\frac{\partial \boldsymbol{\phi}_2}{\partial q_2}$

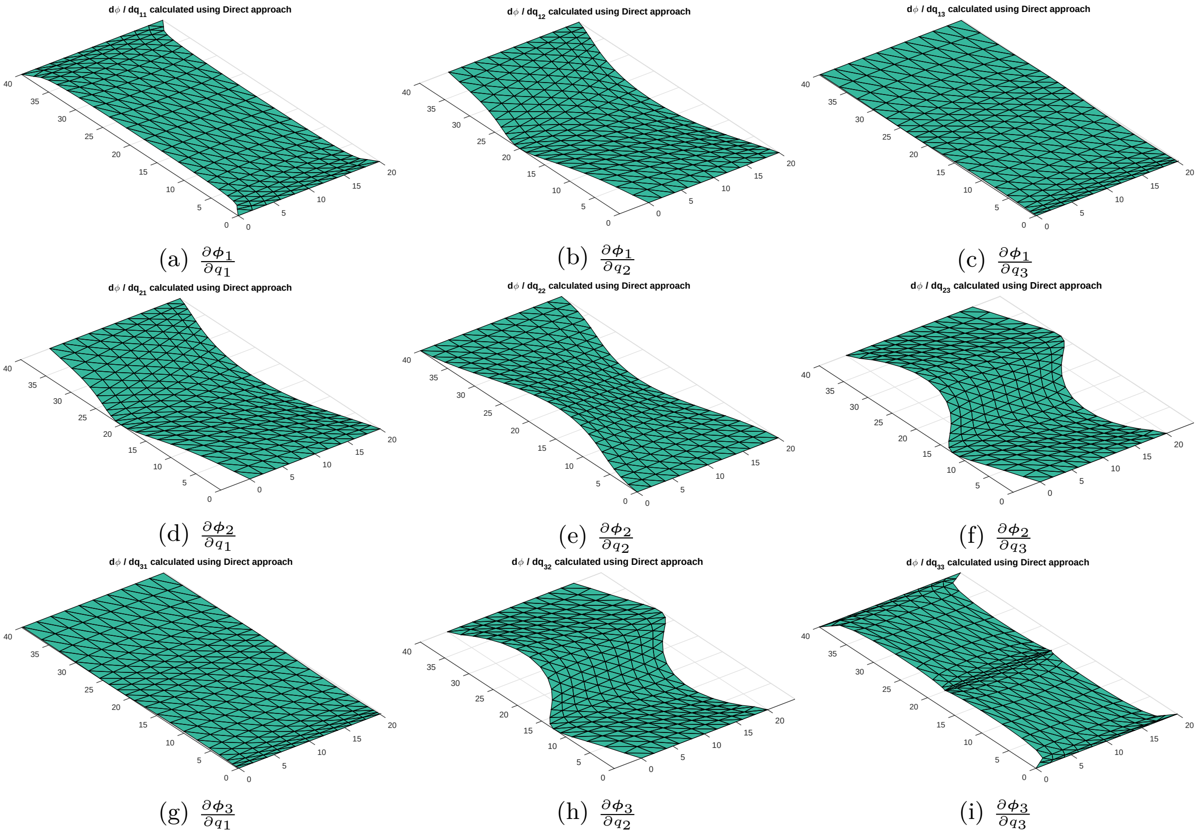
(f) $\frac{\partial \boldsymbol{\phi}_2}{\partial q_3}$

(g) $\frac{\partial \boldsymbol{\phi}_3}{\partial q_1}$

(h) $\frac{\partial \boldsymbol{\phi}_3}{\partial q_2}$

(i) $\frac{\partial \boldsymbol{\phi}_3}{\partial q_3}$

Figure 3.2: The modal derivatives (MDs) for the plate example calculated using the Nelson's method [18]. The VMs are out-of-plane modes, featuring bending and torsion. For this flat plate application, the MDs are in-plane only. Note the symmetry of the MDs, i.e $\frac{\partial \boldsymbol{\phi}_i}{\partial q_j} = \frac{\partial \boldsymbol{\phi}_j}{\partial q_i}$.


dϕ / dq11 calculated using Direct approach
dϕ / dq12 calculated using Direct approach
dϕ / dq13 calculated using Direct approach
dϕ / dq21 calculated using Direct approach
dϕ / dq22 calculated using Direct approach
dϕ / dq23 calculated using Direct approach
dϕ / dq31 calculated using Direct approach
dϕ / dq32 calculated using Direct approach
dϕ / dq33 calculated using Direct approach
0 5 10 15 20 25 30 35 40
0 5 10 15 20

(a) $\frac{\partial \phi_1}{\partial q_1}$

(b) $\frac{\partial \phi_1}{\partial q_2}$

(c) $\frac{\partial \phi_1}{\partial q_3}$

(d) $\frac{\partial \phi_2}{\partial q_1}$

(e) $\frac{\partial \phi_2}{\partial q_2}$

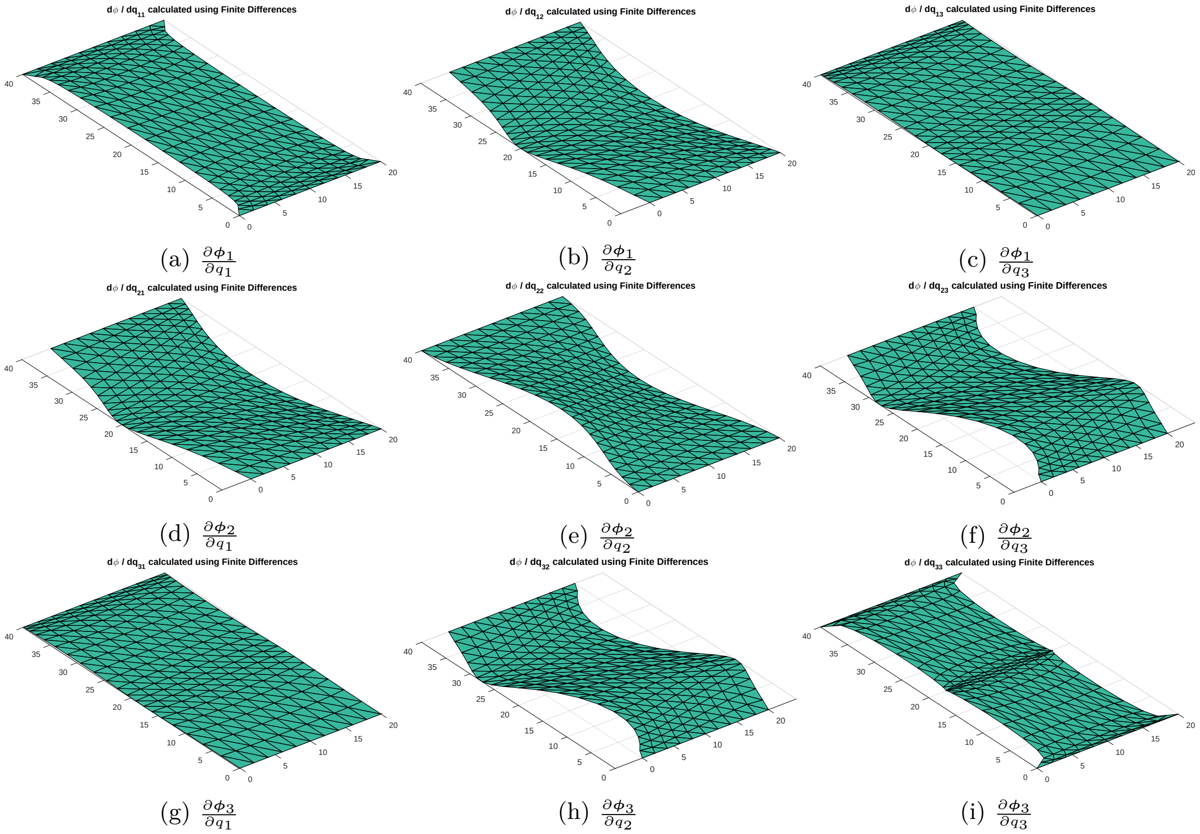
(f) $\frac{\partial \phi_2}{\partial q_3}$

(g) $\frac{\partial \phi_3}{\partial q_1}$

(h) $\frac{\partial \phi_3}{\partial q_2}$

(i) $\frac{\partial \phi_3}{\partial q_3}$

Figure 3.3: The modal derivatives (MDs) for the plate example calculated using the Direct approach. The VMs are out-of-plane modes, featuring bending and torsion. For this flat plate application, the MDs are in-plane only. Note the symmetry of the MDs, i.e $\frac{\partial \phi_i}{\partial q_j} = \frac{\partial \phi_j}{\partial q_i}$.



(a) $\frac{\partial \phi_1}{\partial q_1}$

(b) $\frac{\partial \phi_1}{\partial q_2}$

(c) $\frac{\partial \phi_1}{\partial q_3}$

(d) $\frac{\partial \phi_2}{\partial q_1}$

(e) $\frac{\partial \phi_2}{\partial q_2}$

(f) $\frac{\partial \phi_2}{\partial q_3}$

(g) $\frac{\partial \phi_3}{\partial q_1}$

(h) $\frac{\partial \phi_3}{\partial q_2}$

(i) $\frac{\partial \phi_3}{\partial q_3}$

Figure 3.4: The modal derivatives (MDs) for the plate example calculated using the Finite differences. The VMs are out-of-plane modes, featuring bending and torsion. For this flat plate application, the MDs are in-plane only. Note the symmetry of the MDs, i.e $\frac{\partial \phi_i}{\partial q_j} = \frac{\partial \phi_j}{\partial q_i}$.

3.3.2. Optimal MDs basis selection

As seen above, the MDs capture the essential second order non-linearities of the system. But if all the MDs corresponding to a given set of VMs, are used in order to augment the basis, then the size of the basis increases with $\mathcal{O}(m^2)$. This is undesirable and in practise only a few MDs could be selected to capture the nonlinear response of the system. Apart from already existing ones, different methods of basis selection are proposed here.

Maximum Modal Interaction (MMI)

Drawn from [19], the basic idea of this method is to calculate the modal interaction between different modes during a linear run and use the MDs corresponding to the maximum interaction in augmenting the basis. A weightage matrix $\mathbf{W}$ can be made to, rank the MDs in order of preference

$$W_{ij} = |\max_{t \in T} \eta_i(t)\eta_j(t)|, \tag{3.37}$$

where W_{ij} represents the weightage of the MD $\boldsymbol{\theta}_{ij}$ and $\eta_i(t)$ represents the time varying amplitude of the i^{th} mode in linear modal superposition run in time span T. This weightage matrix is obviously cheap to obtain. By looking at the maximum of the product of two modal amplitudes, one obtains the interaction between the two modes in the sense that if the relevant weightage becomes high, then the corresponding nonlinearity may get triggered and that MD becomes important.

Modal Virtual Work (MVW)

The basic idea is to compute the Virtual work done by the nonlinear elastic forces from one mode upon another mode. The maximum amplitudes generated by Linear modal superposition are multiplied by the corresponding modes while calculating the internal forces. These projection are collected in the matrix $\mathbf{P} \in \mathbb{R}^{M \times M}$. Then the MD weightage is collected in the matrix $\mathbf{W}$,

$$t_{max} = \arg\max_{t \in T} |\eta_j(t))|, \tag{3.38}$$

$$P_{ij} = |\boldsymbol{\phi}_i^T \mathbf{F}_{int}(\eta_j(t_{max})\boldsymbol{\phi}_j)|, \tag{3.39}$$

$$\mathbf{W} = \frac{1}{2}(\mathbf{P} + \mathbf{P}^T), \tag{3.40}$$

where $\mathbf{W}$ is made symmetric using the projection matrix $\mathbf{P}$. Physically, this also represents the interaction between two modes, thereby establishing importance of the corresponding MD.

3.3.3. An Approach using Tensors

The benefit of solving a reduced set of nonlinear equations (Equation (3.4)) obtained after Galerkin projection of the full system on to a ROB generally lies only in the time saved in the solution of a much smaller linearised system at every NR iteration during time integration at each step. However, as remarked earlier the evaluation of nonlinearity and subsequent projection onto a ROB is also a very expensive task. The hyper-reduction methods which cheaply approximate this step as discussed in the previous sections, indeed are very useful in this regard. Exact evaluation of nonlinearity and the tangent stiffness (Jacobian) at every NR iteration is done by the assembly of element level contributions in the physical domain which is a costly procedure. Hyper-reduction tries to minimize this cost by evaluation of the nonlinearity (or the projected nonlinearity) at a few elements(or points). But in case of a polynomial nonlinearity, the projection can be evaluated cheaply and exactly using tensors which can be precomputed offline.

Using tensors, the computation of projected residual doesn't involve evaluation of nonlinearity in the physical domain. In this manner, the online time for time integration is independent of the size of model and depends only on the size of the ROB $\mathbf{V}$. While the ROM obtained using hyper-reduction requires an offline cost for every load case, the offline cost related to the Tensorial approach is one time and only dependent on the ROB and the physical model.The use of tensors for offline evaluation of polynomial nonlinearity not new [20]. In the present context, the nonlinearity is cubic in displacements and can be expressed in the form of the following tensor relationship

$$f_I = K_{Ii}u_i + {}^2K_{Iij}u_iu_j + {}^3K_{Iijk}u_iu_ju_k. \tag{3.41}$$

or

$$\mathbf{f}(\mathbf{u}) = \mathbf{K}\cdot\mathbf{u} + {}^2_3\mathbf{K} : (\mathbf{u}\otimes\mathbf{u}) + {}^3_4\mathbf{K}\vdots(\mathbf{u}\otimes\mathbf{u}\otimes\mathbf{u}) \tag{3.42}$$

Upon substitution of this internal force into Equation (3.4), the following can be obtained

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}\dot{\mathbf{q}} + \mathbf{V}^T(\mathbf{K}\cdot\mathbf{V}\mathbf{q} + {}^2_3\mathbf{K} : (\mathbf{V}\mathbf{q}\otimes\mathbf{V}\mathbf{q}) + {}^3_4\mathbf{K}\vdots(\mathbf{V}\mathbf{q}\otimes\mathbf{V}\mathbf{q}\otimes\mathbf{V}\mathbf{q})) = \mathbf{V}^T\mathbf{g}(t). \tag{3.43}$$

Grouping the constant terms, together into tensors, the following system of reduced equations can be formed.

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}\dot{\mathbf{q}} + \tilde{\mathbf{K}}\cdot\mathbf{q} + {}^2_3\tilde{\mathbf{K}} : (\mathbf{q}\otimes\mathbf{q}) + {}^3_4\tilde{\mathbf{K}}\vdots(\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q})) = \mathbf{V}^T\mathbf{g}(t), \tag{3.44}$$

where

$$\tilde{\mathbf{M}} = \mathbf{V}^T\mathbf{M}\mathbf{V} \qquad \in \mathbb{R}^{m\times m} \tag{3.45}$$

$$\tilde{\mathbf{C}} = \mathbf{V}^T\mathbf{C}\mathbf{V} \qquad \in \mathbb{R}^{m\times m} \tag{3.46}$$

$$\tilde{\mathbf{K}} = \mathbf{V}^T\mathbf{K}\mathbf{V} \qquad \in \mathbb{R}^{m\times m} \tag{3.47}$$

$${}^2_3\tilde{\mathbf{K}} = ((\mathbf{V}^T\cdot{}^2_3\mathbf{K})\cdot_{21}\mathbf{V})\cdot_{21}\mathbf{V} \qquad \in \mathbb{R}^{m\times m\times m} \tag{3.48}$$

$${}^3_4\tilde{\mathbf{K}} = (((\mathbf{V}^T\cdot{}^3_4\mathbf{K})\cdot_{21}\mathbf{V})\cdot_{21}\mathbf{V})\cdot_{21}\mathbf{V} \qquad \in \mathbb{R}^{m\times m\times m\times m} \tag{3.49}$$

or

$$\tilde{M}_{IJ} = V_{iI}M_{ij}V_{jJ} \tag{3.50}$$

$$\tilde{C}_{IJ} = V_{iI}C_{ij}V_{jJ} \tag{3.51}$$

$$\tilde{K}_{IJ} = V_{iI}K_{ij}V_{jJ} \tag{3.52}$$

$${}^2\tilde{K}_{IJK} = V_{iI}\ {}^2K_{ijk}V_{jJ}V_{kK} \tag{3.53}$$

$${}^3\tilde{K}_{IJKL} = V_{iI}\ {}^2K_{ijkl}V_{jJ}V_{kK}V_{lL} \tag{3.54}$$

Jacobian for Time Integration

The reduced set of equations (3.44) is solved using implicit Newmark scheme resulting in a nonlinear set of algebraic equations which is iteratively solved using NR method. In doing so, the Jacobian can always be assembled in the reduced space ($\mathbf{q}$) using tensors instead of the physical space as follows.

$$\mathbf{r}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = \tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}\dot{\mathbf{q}} + \tilde{\mathbf{K}}\cdot\mathbf{q} + {}^2_3\tilde{\mathbf{K}} : (\mathbf{q}\otimes\mathbf{q}) + {}^3_4\tilde{\mathbf{K}}\vdots(\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q})) - \mathbf{V}^T\mathbf{g}(t) = \mathbf{0} \tag{3.55}$$

Using Newmark's scheme, the Jacobian can be calculated as follows:

$$\mathbf{S}(\mathbf{q}) = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{q}} = \underbrace{\frac{\partial\mathbf{r}}{\partial\mathbf{q}}}_{\mathbf{K}^t} + \frac{\gamma}{\beta h}\underbrace{\frac{\partial\mathbf{r}}{\partial\dot{\mathbf{q}}}}_{\mathbf{C}^t} + \frac{1}{\beta h^2}\underbrace{\frac{\partial\mathbf{r}}{\partial\ddot{\mathbf{q}}}}_{\mathbf{M}^t} \tag{3.56}$$

where,

$$\begin{gathered}\mathbf{M}^t = \tilde{\mathbf{M}},\\ \mathbf{C}^t = \tilde{\mathbf{C}},\\ K^t_{IJ} = \tilde{K}_{IJ} + \left({}^2\tilde{K}_{IJj} + {}^2\tilde{K}_{IjJ}\right)q_j + \left({}^3\tilde{K}_{IJij} + {}^3\tilde{K}_{IiJj} + {}^3\tilde{K}_{IijJ}\right)q_iq_j\end{gathered} \tag{3.57}$$

Implementation of Tensors

Here, the reduced matrices $\tilde{\mathbf{M}}$, $\tilde{\mathbf{C}}$, $\tilde{\mathbf{K}}$, ${}^2_3\tilde{\mathbf{K}}$ and ${}^3_4\tilde{\mathbf{K}}$ can be efficiently computed by summing over the element level contributions of the full tensors such that

$$\mathbf{f}^e(\mathbf{u}^e) = \mathbf{K}^e \cdot \mathbf{u}^e + {}^2_3\mathbf{K}^e : (\mathbf{u}^e \otimes \mathbf{u}^e) + {}^3_4\mathbf{K}^e \vdots (\mathbf{u}^e \otimes \mathbf{u}^e \otimes \mathbf{u}^e) \tag{3.58}$$

$$\tilde{\mathbf{M}} = \sum_{e=1}^{n_e} (\mathbf{V}^e)^T \mathbf{M}^e \mathbf{V}^e \tag{3.59}$$

$$\tilde{\mathbf{C}} = \sum_{e=1}^{n_e} (\mathbf{V}^e)^T \mathbf{C}^e \mathbf{V}^e \tag{3.60}$$

$$\tilde{\mathbf{K}} = \sum_{e=1}^{n_e} (\mathbf{V}^e)^T \mathbf{K}^e \mathbf{V}^e \tag{3.61}$$

$${}^2_3\tilde{\mathbf{K}} = \sum_{e=1}^{n_e} (((\mathbf{V}^e)^T \cdot {}^2_3\mathbf{K}^e) \cdot_{21} \mathbf{V}^e) \cdot_{21} \mathbf{V}^e \tag{3.62}$$

$${}^3_4\tilde{\mathbf{K}} = \sum_{e=1}^{n_e} ((((\mathbf{V}^e)^T \cdot {}^3_4\mathbf{K}^e) \cdot_{21} \mathbf{V}^e) \cdot_{21} \mathbf{V}^e) \cdot_{21} \mathbf{V}^e \tag{3.63}$$

It is worth mentioning that tensors calculated in this manner do not require the full tensors ${}^2_3\mathbf{K} \in \mathbb{R}^{n \times n \times n}$ and ${}^3_4\mathbf{K} \in \mathbb{R}^{n \times n \times n \times n}$ which are sparse but can be huge in size depending upon the system. Due to element level summation, the amount of *offline* time required to calculate ${}^2_3\tilde{\mathbf{K}}$ and ${}^3_4\tilde{\mathbf{K}}$ scales linearly with the total number of elements. See Appendix A.1 for formulation of element tensors ${}^2_3\mathbf{K}^e$ and ${}^3_4\mathbf{K}^e$.

3.4. Quadratic Manifold

As explained above, the linear modal superposition is a good technique to obtain the reduced solution of a linear system. However when the nonlinearities become significant, the modal basis needs to be updated to capture the nonlinearities. Solution of an eigenvalue problem *online* can be a very expensive task, sometimes even leaving the idea of such a reduction redundant. However, second order effects in nonlinearities can be captured by enriching the basis with MDs as discussed in the previous section. The idea was based on a truncated Taylor expansion around equilibrium position. Since the size of the basis can significantly increased upon inclusion of MDs, techniques for selection of a few important MDs in a cheap manner was studied (MMI) and proposed (MVW) in Section 3.3.2.

During reduction, one essentially introduces a mapping $\mathbf{u}(\mathbf{q}) : \mathbb{R}^n \mapsto \mathbb{R}^M$, M being the number of free coordinates in a reduced manifold in an n - dimensional space. So far, this mapping was linear in the above reduction methods and thus required more degrees of freedom to capture the nonlinear response. However, a nonlinear (such as quadratic) mapping can also be proposed which has the same no. DOFs as that in linear modal superposition. A quadratic manifolds can be proposed by including terms up to second order in the Taylor expansion 3.24.

A quadratic Mapping would then be given by

$$\mathbf{u} = \mathbf{u}(\mathbf{q}) := \boldsymbol{\Phi} \cdot \mathbf{q} + \frac{1}{2}\boldsymbol{\Omega} : (\mathbf{q} \otimes \mathbf{q}), \tag{3.64}$$

Notation: Here $\mathbf{q} \in \mathbb{R}^m$, $\boldsymbol{\Phi} \in \mathbb{R}^{n \times m}$, $\boldsymbol{\Omega} \in \mathbb{R}^{n \times m \times m}$ is a third order tensor and the Kronecker or dyadic product $(\mathbf{q} \otimes \mathbf{q})$ signifies the matrix(second order tensor) $\mathbf{q}\mathbf{q}^T$. The $(\bullet) : (\bullet)$ operation represents contraction i.e. summation over two indices which results in a first order tensor (a vector) in this case. More clearly, the Equation (3.64) can be written using indices as

$$u_I = \sum_{i=1}^{m} \Phi_{Ii} q_i + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \Omega_{Iij} q_i q_j \qquad \forall I \in \{1, \dots, n\} \tag{3.65}$$

The three indices in Ω are characteristic of a third order tensor, 2 indices in Φ that of a second order tensor (matrix) and the single index in u of a first order tensor (vector). Zeroth order tensor are index-less scalars. (See notation in Section 1.3)

and using the indicial notation with Einstein summation convention, the mapping 3.64 can be written as

$$u_I = \Phi_{Ii} q_i + \frac{1}{2} \Omega_{Iij} q_i q_j. \tag{3.66}$$

The velocity and acceleration are then expressed as functions of the modal coordinates q_I as

$$\dot{u}_I = \Phi_{Ii} \dot{q}_i + \frac{1}{2} \Omega_{Iij} \dot{q}_i q_j + \frac{1}{2} \Omega_{Iij} \dot{q}_j q_i \tag{3.67}$$

$$= \Phi_{Ii} \dot{q}_i + \underbrace{\frac{1}{2} \left(\Omega_{Iij} + \Omega_{Iji} \right)}_{\Theta_{Iij}} \dot{q}_i q_j \tag{3.68}$$

or

$$\dot{\mathbf{u}} = \boldsymbol{\Phi} \cdot \dot{\mathbf{q}} + \boldsymbol{\Theta} : (\dot{\mathbf{q}} \otimes \mathbf{q}), \tag{3.69}$$

and

$$\ddot{u}_I = \Phi_{Ii} \ddot{q}_i + \Theta_{Iij} \ddot{q}_i q_j + \Theta_{Iij} \dot{q}_i \dot{q}_j, \tag{3.70}$$

or

$$\ddot{\mathbf{u}} = \boldsymbol{\Phi} \cdot \ddot{\mathbf{q}} + \boldsymbol{\Theta} : (\ddot{\mathbf{q}} \otimes \mathbf{q}) + \boldsymbol{\Theta} : (\dot{\mathbf{q}} \otimes \dot{\mathbf{q}}). \tag{3.71}$$

This mapping is then inserted into the governing Equation (3.1) to obtain

$$\mathbf{M}\ddot{\mathbf{u}}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) + \mathbf{C}\dot{\mathbf{u}}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{f}(\mathbf{u}(\mathbf{q})) = \mathbf{g}(t). \tag{3.72}$$

The set of n equations in 3.72 can then be projected onto a tangent basis to ensure that the error of mapping is orthogonal to this tangent subspace. The subspace tangent to the quadratic manifold would be defined as

$$\begin{aligned} \partial U_{IJ} = \frac{\partial u_I}{\partial q_J} &= \Phi_{IJ} + \frac{1}{2}\Omega_{IJj}q_j + \frac{1}{2}\Omega_{IiJ}q_i && (3.73) \\ &= \Phi_{IJ} + \Theta_{IJj}q_j && (3.74) \end{aligned}$$

or simply,

$$\underbrace{\partial \mathbf{U}(\mathbf{q})}_{\in \mathbb{R}^{n\times m}} = \frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}} = \mathbf{\Phi} + \mathbf{\Theta}\cdot\mathbf{q}. \tag{3.75}$$

Then the reduced equations can be obtained as

$$\partial \mathbf{U}^T(\mathbf{M}\ddot{\mathbf{u}}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) + \mathbf{C}\dot{\mathbf{u}}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{f}(\mathbf{u}(\mathbf{q}))) = \partial \mathbf{U}^T\mathbf{g}(t). \tag{3.76}$$

Now Equation (3.76) is a set of m equations in m unknowns i.e. $\mathbf{q} \in \mathbb{R}^m$. In indicial notation, the I^{th} equation can be written as

$$\partial U_{iI}(M_{ij}\ddot{u}_j + C_{ij}\dot{u}_j + f_i) = \partial U_{iI}g_i(t). \tag{3.77}$$

This reduced system of equations can be solved using time integration schemes. Generally implicit Newmark scheme is used in structural dynamics. This leads to a system of nonlinear algebraic equations at every time step, which needs to be solved iteratively (usually using Newton-Raphson iterations).

Newmark Scheme for nonlinear systems: Explicit schemes tend to impose a very low CFL limit for time step size which is necessarily required for numerical stability of the scheme. Even though the system is simpler in explicit case, it still becomes extremely slow due to the inhibitive timestep size. An implicit scheme is thus preferred over an explicit one. For a general system of nonlinear equations of the form

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{f}(\mathbf{a}) = \mathbf{g}(t), \tag{3.78}$$

one can rewrite them in terms of displacements at time level t_{k+1} with the introduction of a residual vector

$$\mathbf{r}(\mathbf{a},\dot{\mathbf{a}},\ddot{\mathbf{a}}) = \mathbf{M}\ddot{\mathbf{a}} + \mathbf{C}\dot{\mathbf{a}} + \mathbf{f}(\mathbf{a}) - \mathbf{g} = \mathbf{0}. \tag{3.79}$$

For time step size h ($t_{k+1} = t_k + h$), accelerations velocities and displacements are related using Newmark's method in the following manner (see e.g. [2]).

$$\begin{aligned} \dot{\mathbf{a}}_{k+1} &= \dot{\mathbf{a}}_k + (1-\gamma)h\ddot{\mathbf{a}}_k + \gamma\ddot{\mathbf{a}}_{k+1}, \\ \mathbf{a}_{k+1} &= \mathbf{a}_k + h\dot{\mathbf{a}}_k + h^2(\frac{1}{2} - \beta)\ddot{\mathbf{a}}_k + h^2\beta\ddot{\mathbf{a}}_{k+1}, \end{aligned} \tag{3.80}$$

where the γ and β are constant parameters associated with the quadrature scheme. The time integration relations are then inverted in the following manner.

$$\begin{aligned} \ddot{\mathbf{a}}_{k+1} &= \frac{1}{\beta h^2}(\mathbf{a}_{k+1} - \mathbf{a}^*_{k+1}), \\ \dot{\mathbf{a}}_{k+1} &= \dot{\mathbf{a}}^*_{k+1} + \frac{\gamma}{\beta h}(\mathbf{a}_{k+1} - \mathbf{a}^*_{k+1}), \end{aligned} \tag{3.81}$$

where the predictors $\mathbf{a}^*_{k+1}$ and $\dot{\mathbf{a}}^*_{k+1}$ are obtained by setting $\ddot{\mathbf{a}}_{k+1} = \mathbf{0}$ in Equation (3.80) :

$$\begin{aligned}\dot{\mathbf{a}}^*_{k+1} &= \dot{\mathbf{a}}_k + (1-\gamma)h\ddot{\mathbf{a}}_k,\\ \mathbf{a}^*_{k+1} &= \mathbf{a}_k + h\dot{\mathbf{a}}_k + h^2(\frac{1}{2} - \beta)\ddot{\mathbf{a}}_k.\end{aligned} \tag{3.82}$$

Equations (3.81) and (3.82) can be substituted in Equation (3.79) to nonlinear residual equation only in terms of $\mathbf{a}_{k+1}$

$$\mathbf{r}(\mathbf{a}_{k+1}) = \mathbf{0}. \tag{3.83}$$

The nonlinear algebraic equations is solved using linearisation. If $\mathbf{a}^p_{k+1}$ as an approximation to $\mathbf{a}_{k+1}$ resulting from iteration k. Then the following system can be iteratively solved to determine increment $\Delta\mathbf{a}^p$ at each iteration.

$$\mathbf{r}(\mathbf{a}^{p+1}_{k+1}) \approx \mathbf{r}(\mathbf{a}^p_{k+1}) + \left.\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{a}}\right|_{\mathbf{a}^p_{k+1}} \Delta\mathbf{a}^p = \mathbf{0}, \tag{3.84}$$

$$\mathbf{S}(\mathbf{a}) = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{a}} = \frac{\partial\mathbf{r}}{\partial\mathbf{a}} + \frac{\partial\mathbf{r}}{\partial\dot{\mathbf{a}}}\frac{\partial\dot{\mathbf{a}}}{\partial\mathbf{a}} + \frac{\partial\mathbf{r}}{\partial\ddot{\mathbf{a}}}\frac{\partial\ddot{\mathbf{a}}}{\partial\mathbf{a}}, \tag{3.85}$$

where making use of relations 3.81,

$$\frac{\partial\ddot{\mathbf{a}}}{\partial\mathbf{a}} = \frac{1}{\beta h^2}\mathbf{I}, \qquad \frac{\partial\dot{\mathbf{a}}}{\partial\mathbf{a}} = \frac{\gamma}{\beta h}\mathbf{I}. \tag{3.86}$$

For a general nonlinear system, the evaluation of the tangent stiffness and internal forces required during formation of the Jacobian ($\mathbf{S}$) and the residual ($\mathbf{r}$) respectively, is done by element level assembly during each iteration. As discussed earlier, this is an expensive online cost apart from the linear system solution. The linear system solution cost is mitigated by projection onto a ROB but then the mapping, nonlinearity evaluation and projection become dominant in taking the CPU time during the time integration as the system becomes larger. An effective way to deal with this is the evaluation of nonlinearities offline using tensors, thereby making time integration independent of the system size.

3.4.1. An Approach using Tensors

The system under consideration employing the Von Karman kinematic model, has up to cubic geometric nonlinearities which makes the internal force vector a polynomial in terms of physical displacements. Since the nonlinearities are cubic in nature, tensors up to 4th order are required to express those nonlinearities as follows

$$f_I = K_{Ii}u_i + {}^2K_{Iij}u_iu_j + {}^3K_{Iijk}u_iu_ju_k, \tag{3.87}$$

or

$$\mathbf{f}(\mathbf{u}) = \mathbf{K}\cdot\mathbf{u} + {}^2_3\mathbf{K} : (\mathbf{u}\otimes\mathbf{u}) + {}^3_4\mathbf{K}\vdots(\mathbf{u}\otimes\mathbf{u}\otimes\mathbf{u}), \tag{3.88}$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$, ${}^2_3\mathbf{K} \in \mathbb{R}^{n\times n\times n}$ and ${}^3_4\mathbf{K} \in \mathbb{R}^{n\times n\times n\times n}$. See Appendix A.1 for element level implementation of ${}^2_3\mathbf{K}$ and ${}^3_4\mathbf{K}$.

Notation: The left subscript refers to the order of tensor for tensors with order higher than two. For e.g. ${}_4\mathbf{K}$ is a fourth order tensor.

The projection of the linear, quadratic and cubic term can be considered separately.

The inertial forces $f^{in}_I = M_{Ii}\ddot{u}$ projected onto the tangential subspace ∂U_{IJ} are written as

$$\hat{f}^{in}_I = \partial U_{iI}M_{ij}\ddot{u}_j = (\Phi_{iI} + \Theta_{iIj}q_j)M_{ik}(\Phi_{kl}\ddot{q}_l + \Theta_{klp}\ddot{q}_lq_p + \Theta_{klp}\dot{q}_l\dot{q}_p). \tag{3.89}$$

The forces 3.89 can be simplified as

$$\hat{f}_I^{in} = \ddot{q}_I + \mathcal{M}_{Iijk}^{\Theta\Theta}(q_i\ddot{q}_j q_k + q_i\dot{q}_j\dot{q}_k) + \mathcal{M}_{Iij}^{\Phi\Theta}(\ddot{q}_i q_j + \dot{q}_i\dot{q}_j) + \mathcal{M}_{Iij}^{\Theta\Phi}\ddot{q}_j q_i, \tag{3.90}$$

or

$$\hat{\mathbf{f}}^{in} = \partial\mathbf{U}^T\mathbf{M}\ddot{\mathbf{u}} = \ddot{\mathbf{q}} + {}_3\boldsymbol{\mathcal{M}}^{\Phi\Theta} : (\ddot{\mathbf{q}}\otimes\mathbf{q} + \dot{\mathbf{q}}\otimes\dot{\mathbf{q}}) + {}_3\boldsymbol{\mathcal{M}}^{\Theta\Phi} : (\mathbf{q}\otimes\ddot{\mathbf{q}}) + {}_4\boldsymbol{\mathcal{M}}^{\Theta\Theta}\vdots(\mathbf{q}\otimes\ddot{\mathbf{q}}\otimes\mathbf{q} + \mathbf{q}\otimes\dot{\mathbf{q}}\otimes\dot{\mathbf{q}}), \tag{3.91}$$

where

$$\begin{aligned} \mathcal{M}_{IJ}^{\Phi\Phi} &= \Phi_{iI}M_{ij}\Phi_{jJ}, \\ \mathcal{M}_{IJK}^{\Phi\Theta} &= \Phi_{iI}M_{ij}\Theta_{jJK}, \\ \mathcal{M}_{IJK}^{\Theta\Phi} &= \Theta_{iIJ}M_{ij}\Phi_{jK}, \\ \mathcal{M}_{IJKL}^{\Theta\Theta} &= \Theta_{iIJ}M_{ij}\Theta_{jKL}. \end{aligned} \tag{3.92}$$

A unit modal mass normalization for Φ_{IJ} has been adopted here i.e.$\mathcal{M}_{IJ}^{\Phi\Phi} = \delta_{IJ}$, where δ_{IJ} is the Kroncker-delta and represents the Identity matrix. The nonlinear constraint (3.64) between the modal and the physical coordinates introduce state dependent inertial forces. This is formally analogous to multi-body dynamics, where the nonlinear constraints introduced by joints give rise to state dependent inertial forces. Likewise, the reduced damping forces $\hat{f}_I^{dam} \in \mathbb{R}^m$ are

$$\hat{f}_I^{dam} = \partial U_{iI}C_{ij}\dot{u}_j = (\Phi_{iI} + \Theta_{iIj}q_j)C_{ik}(\Phi_{kl}\dot{q}_l + \Theta_{klp}\dot{q}_l q_p). \tag{3.93}$$

and upon simplification, we get:

$$\hat{f}_I^{dam} = \mathcal{C}_{Ii}^{\Phi\Phi}\dot{q}_i + \mathcal{C}_{Iij}^{\Phi\Theta}\dot{q}_i q_j + \mathcal{C}_{Iij}^{\Theta\Phi}q_i\dot{q}_j + \mathcal{C}_{Iijk}^{\Theta\Theta}q_i\dot{q}_j q_k, \tag{3.94}$$

or

$$\hat{\mathbf{f}}^{dam} = \partial\mathbf{U}^T\mathbf{C}\dot{\mathbf{u}} = \boldsymbol{\mathcal{C}}^{\Phi\Phi}\cdot\dot{\mathbf{q}} + {}_3\boldsymbol{\mathcal{C}}^{\Phi\Theta} : (\dot{\mathbf{q}}\otimes\mathbf{q}) + {}_3\boldsymbol{\mathcal{C}}^{\Theta\Phi}(\mathbf{q}\otimes\dot{\mathbf{q}}) + {}_4\boldsymbol{\mathcal{C}}^{\Theta\Theta}\vdots(\mathbf{q}\otimes\dot{\mathbf{q}}\otimes\mathbf{q}), \tag{3.95}$$

where

$$\begin{aligned} \mathcal{C}_{IJ}^{\Phi\Phi} &= \Phi_{iI}C_{ij}\Phi_{jJ}, \\ \mathcal{C}_{IJK}^{\Phi\Theta} &= \Phi_{iI}C_{ij}\Theta_{jJK}, \\ \mathcal{C}_{IJK}^{\Theta\Phi} &= \Theta_{iIJ}C_{ij}\Phi_{jK}, \\ \mathcal{C}_{IJKL}^{\Theta\Theta} &= \Theta_{iIJ}C_{ij}\Theta_{jKL}. \end{aligned} \tag{3.96}$$

Analogously, the projected linear elastic forces $\hat{f}_I^{lin} \in \mathbb{R}^m$ are written as

$$\hat{f}_I^{lin} = \partial U_{iI}K_{ij}u_j = (\Phi_{iI} + \Theta_{iIj}q_j)K_{ik}(\Phi_{kl}q_l + \frac{1}{2}\Omega_{klp}q_l q_p), \tag{3.97}$$

and, collecting the tensorial quantities, we obtain

$$\hat{f}_I^{lin} = \omega_I^2 q_I + \frac{1}{2}\mathcal{K}_{Iijk}^{\Theta\Omega}q_i q_j q_k + \frac{1}{2}\mathcal{K}_{Iij}^{\Phi\Omega}q_i q_j + \mathcal{K}_{Iij}^{\Theta\Phi}q_i q_j, \tag{3.98}$$

or

$$\hat{\mathbf{f}}^{lin} = \partial\mathbf{U}^T\mathbf{K}\mathbf{u} = \boldsymbol{\Lambda}^2\cdot\mathbf{q} + \frac{1}{2}{}_4\boldsymbol{\mathcal{K}}^{\Theta\Omega}\vdots(\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q}) + \frac{1}{2}{}_3\boldsymbol{\mathcal{K}}^{\Phi\Omega} : (\mathbf{q}\otimes\mathbf{q}) + {}_3\boldsymbol{\mathcal{K}}^{\Theta\Phi} : (\mathbf{q}\otimes\mathbf{q}), \tag{3.99}$$

with

$$\begin{aligned} \mathcal{K}_{IJK}^{\Phi\Omega} &= \Phi_{iI}K_{ij}^{lin}\Omega_{jJK}, \\ \mathcal{K}_{IJK}^{\Theta\Phi} &= \Theta_{iIJ}K_{ij}^{lin}\Phi_{jK}, \\ \mathcal{K}_{IJKL}^{\Theta\Omega} &= \Theta_{iIJ}K_{ij}^{lin}\Omega_{jKL}. \end{aligned} \tag{3.100}$$

The unit mass normalization on the VMs results in the diagonal matrix $\boldsymbol{\Lambda}^2$, where ω_i^2 is the i^{th} eigenvalue of Equation (3.16).

The projected quadratic forces are written as:

$${}^2\hat{f}_I = (\Phi_{iI} + \Theta_{iIj}q_j)^2 K_{ikl}(\Phi_{kp}q_p + \frac{1}{2}\Omega_{kpr}q_p q_r)(\Phi_{ls}q_s + \frac{1}{2}\Omega_{lst}q_s q_t), \tag{3.101}$$

or

$$^{2}\hat{\mathbf{f}} = \partial\mathbf{U}^{T}\left[{}^{2}_{3}\mathbf{K} : (\mathbf{u}\otimes\mathbf{u})\right], \tag{3.102}$$

Likewise, the reduced cubic force term is given by

$$^{3}\hat{f}_{I} = (\Phi_{iI} + \Theta_{iIj}q_{j})^{3}K_{iklp}(\Phi_{ku}q_{u} + \frac{1}{2}\Omega_{kuv}q_{u}q_{v})(\Phi_{lw}q_{w} + \frac{1}{2}\Omega_{lwz}q_{w}q_{z})(\Phi_{px}q_{x} + \frac{1}{2}\Omega_{pxy}q_{x}q_{y}), \tag{3.103}$$

or

$$^{3}\hat{\mathbf{f}} = \partial\mathbf{U}^{T}\left[{}^{3}_{4}\mathbf{K} : (\mathbf{u}\otimes\mathbf{u}\otimes\mathbf{u})\right], \tag{3.104}$$

The external force $\mathbf{g}(t)$ can also be projected to the variable basis as follows:

$$\hat{f}^{ext}_{I} = \partial U_{iI}g_{i}(t) = \Phi_{iI}g_{i} + \Theta_{iIj}q_{j}p_{i} \tag{3.105}$$

or

$$\hat{\mathbf{f}}^{ext} = \partial\mathbf{U}^{T}\mathbf{g}(t) = \boldsymbol{\Phi}^{T}\mathbf{g}(t) + (\boldsymbol{\Theta}\cdot\mathbf{q})^{T}\cdot\mathbf{g}(t) \tag{3.106}$$

Finally, The projected equations of motion can be then rewritten as

$$\begin{aligned}
&\ddot{q}_{I} + \mathcal{M}^{\Theta\Theta}_{Iijk}(q_{i}\ddot{q}_{j}q_{k} + q_{i}\dot{q}_{j}\dot{q}_{k}) + \mathcal{M}^{\Phi\Theta}_{Iij}(\ddot{q}_{i}q_{j} + \dot{q}_{i}\dot{q}_{j}) + \mathcal{M}^{\Theta\Phi}_{Iij}\ddot{q}_{j}q_{i}+\\
&\mathcal{C}^{\Phi\Phi}_{Ii}\dot{q}_{i} + \mathcal{C}^{\Phi\Theta}_{Iij}\dot{q}_{i}q_{j} + \mathcal{C}^{\Phi\Theta}_{Iij}q_{i}\dot{q}_{j} + \mathcal{C}^{\Theta\Theta}_{Iijk}q_{i}\dot{q}_{j}q_{k}+\\
&\omega^{2}_{I}q_{I} + {}^{2}\mathcal{K}_{Iij}q_{i}q_{j} + {}^{3}\mathcal{K}_{Iijk}q_{i}q_{j}q_{k} + {}^{4}\mathcal{K}_{Iijkl}q_{i}q_{j}q_{k}q_{l}+\\
&{}^{5}\mathcal{K}_{Iijklp}q_{i}q_{j}q_{k}q_{l}q_{p} + {}^{6}\mathcal{K}_{Iijklpr}q_{i}q_{j}q_{k}q_{l}q_{p}q_{r} + {}^{7}\mathcal{K}_{Iijklprs}q_{i}q_{j}q_{k}q_{l}q_{p}q_{r}q_{s} =\\
&\Phi_{iI}p_{i} + \Theta_{iIj}q_{j}p_{i},
\end{aligned} \tag{3.107}$$

or using Equations (3.76), (3.88), (3.95), (3.99), (3.99), (3.102), (3.104) and (3.106)

$$\hat{\mathbf{f}}^{in} + \hat{\mathbf{f}}^{dam} + \hat{\mathbf{f}}^{lin} + {}^{2}\hat{\mathbf{f}} + {}^{3}\hat{\mathbf{f}} = \hat{\mathbf{f}}^{ext} \tag{3.108}$$

$$\begin{array}{r|l}
\text{inertial} \rightarrow & \ddot{\mathbf{q}} + {}_{3}\boldsymbol{\mathcal{M}}^{\Phi\Theta} : (\ddot{\mathbf{q}}\otimes\mathbf{q} + \dot{\mathbf{q}}\otimes\dot{\mathbf{q}}) + {}_{3}\boldsymbol{\mathcal{M}}^{\Theta\Phi} : (\mathbf{q}\otimes\ddot{\mathbf{q}}) + {}_{4}\boldsymbol{\mathcal{M}}^{\Theta\Theta}\vdots(\mathbf{q}\otimes\ddot{\mathbf{q}}\otimes\mathbf{q} + \mathbf{q}\otimes\dot{\mathbf{q}}\otimes\dot{\mathbf{q}})+\\
\text{damping} \rightarrow & \boldsymbol{\mathcal{C}}^{\Phi\Phi}\cdot\dot{\mathbf{q}} + {}_{3}\boldsymbol{\mathcal{C}}^{\Phi\Theta} : (\dot{\mathbf{q}}\otimes\mathbf{q}) + {}_{3}\boldsymbol{\mathcal{C}}^{\Theta\Phi} : (\mathbf{q}\otimes\dot{\mathbf{q}}) + {}_{4}\boldsymbol{\mathcal{C}}^{\Theta\Theta}\vdots(\mathbf{q}\otimes\dot{\mathbf{q}}\otimes\mathbf{q})+\\
\text{elastic} \rightarrow & \boldsymbol{\Lambda}^{2}\cdot\mathbf{q} + {}^{2}_{3}\boldsymbol{\mathcal{K}} : (\mathbf{q}\otimes\mathbf{q}) + {}^{3}_{4}\boldsymbol{\mathcal{K}}\vdots(\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q}) + \cdots + {}^{7}_{8}\boldsymbol{\mathcal{K}}\vdots(\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q})\\
\text{external} \rightarrow & = \boldsymbol{\Phi}^{T}\cdot\mathbf{g}(t) + (\boldsymbol{\Theta}\cdot\mathbf{q})^{T}\cdot\mathbf{g}(t)
\end{array} \tag{3.109}$$

where the operators for the nonlinear elastic terms are all tensorial quantities given in Equation (3.117) that can be computed *offline*

Derivation of Tensor Expressions in reduced equations: The expansion of the reduced quadratic internal forces 3.101: is

$$\begin{aligned}
{}^{2}\hat{f}_{I} =& \Phi_{iI}{}^{2}K_{ikl}\Phi_{kp}\Phi_{ls}q_{p}q_{s}+\\
&\frac{1}{2}\Phi_{iI}{}^{2}K_{ikl}\Phi_{kp}\Omega_{lst}q_{p}q_{s}q_{t}+\\
&\frac{1}{2}\Phi_{iI}{}^{2}K_{ikl}\Omega_{kpr}\Phi_{ls}q_{p}q_{r}q_{s}+\\
&\frac{1}{4}\Phi_{iI}{}^{2}K_{ikl}\Omega_{kpr}\Omega_{lst}q_{p}q_{r}q_{s}q_{t}+\\
&\Theta_{iIj}{}^{2}K_{ikl}\Phi_{kp}\Phi_{ls}q_{j}q_{p}q_{s}+\\
&\frac{1}{2}\Theta_{iIj}{}^{2}K_{ikl}\Phi_{kp}\Omega_{lst}q_{j}q_{p}q_{s}q_{t}+\\
&\frac{1}{2}\Theta_{iIj}{}^{2}K_{ikl}\Omega_{kpr}\Phi_{ls}q_{j}q_{p}q_{r}q_{s}+\\
&\frac{1}{4}\Theta_{iIj}{}^{2}K_{ikl}\Omega_{kpr}\Omega_{lst}q_{j}q_{p}q_{r}q_{s}q_{t}.
\end{aligned} \tag{3.110}$$

The reduced quadratic forces (3.110) can be simplified to the following expression

$$\begin{aligned}{}^2\hat{f}_I =& {}^2\mathcal{K}^{\Phi\Phi\Phi}_{Iij} q_i q_j + \left(\frac{1}{2}{}^2\mathcal{K}^{\Phi\Omega\Phi}_{Iijk} + \frac{1}{2}{}^2\mathcal{K}^{\Phi\Phi\Omega}_{Iijk} + {}^2\mathcal{K}^{\Theta\Phi\Phi}_{Iijk}\right) q_i q_j q_k + \\ & \left(\frac{1}{4}{}^2\mathcal{K}^{\Phi\Omega\Theta}_{Iijkl} + \frac{1}{2}{}^2\mathcal{K}^{\Theta\Omega\Phi}_{Iijkl} + \frac{1}{2}{}^2\mathcal{K}^{\Theta\Phi\Omega}_{Iijkl}\right) q_i q_j q_k q_l + \frac{1}{4}{}^2\mathcal{K}^{\Theta\Omega\Omega}_{Ijprst} q_j q_p q_r q_s q_t,\end{aligned} \tag{3.111}$$

where

$$\begin{aligned}{}^2\mathcal{K}^{\Phi\Phi\Phi}_{IJK} &= \Phi_{iI}{}^2K_{ijk}\Phi_{jJ}\Phi_{kK} \\ {}^2\mathcal{K}^{\Phi\Omega\Phi}_{IJKL} &= \Phi_{iI}{}^2K_{ijk}\Phi_{jJ}\Omega_{kKL} \\ {}^2\mathcal{K}^{\Phi\Phi\Omega}_{IJKL} &= \Phi_{iI}{}^2K_{ijk}\Omega_{jJK}\Phi_{kK} \\ {}^2\mathcal{K}^{\Theta\Phi\Phi}_{IJKL} &= \Theta_{iIJ}{}^2K_{ijk}\Phi_{jK}\Phi_{kL} \\ {}^2\mathcal{K}^{\Phi\Omega\Omega}_{IJKLP} &= \Phi_{iI}{}^2Kikl\Omega_{kJK}\Omega_{lLP} \\ {}^2\mathcal{K}^{\Theta\Omega\Omega}_{IJKLPR} &= \Theta_{iIJ}{}^2K_{ijk}\Omega_{jKL}\Omega_{kPR} \\ {}^2\mathcal{K}^{\Theta\Phi\Omega}_{IJKLP} &= \Theta_{iIJ}{}^2K_{ijk}\Phi_{jK}\Omega_{kLP} \\ {}^2\mathcal{K}^{\Theta\Omega\Phi}_{IJKLP} &= \Theta_{iIJ}{}^2K_{ijk}\Omega_{jKL}\Phi_{kP}.\end{aligned} \tag{3.112}$$

Likewise, expanding the cubic forces 3.103 results in

$$\begin{aligned}{}^3\hat{f}_I = \quad & \Phi_{iI}{}^3K_{iklp}\Phi_{ku}\Phi_{lw}\Phi_{px}q_uq_wq_x + \\ & \frac{1}{2}\Phi_{iI}K^3_{iklp}\Phi_{ku}\Omega_{lwz}\Phi_{px}q_uq_wq_zq_x + \\ & \frac{1}{2}\Phi_{iI}{}^3K_{iklp}\Omega_{kuv}\Phi_{lw}\Phi_{px}q_uq_vq_wq_x + \\ & \frac{1}{4}\Phi_{iI}{}^3K_{iklp}\Omega_{kuv}\Omega_{lwz}\Phi_{px}q_uq_vq_wq_zq_x + \\ & \Theta_{iIj}{}^3K_{iklp}\Phi_{ku}\Phi_{lw}\Phi_{px}q_jq_uq_wq_x + \\ & \frac{1}{2}\Theta_{iIj}{}^3K_{iklp}\Phi_{ku}\Omega_{lwz}\Phi_{px}q_jq_uq_wq_zq_x + \\ & \frac{1}{2}\Theta_{iIj}{}^3K_{iklp}\Omega_{kuv}\Phi_{lw}\Phi_{px}q_jq_uq_vq_wq_x + \\ & \frac{1}{4}\Theta_{iIj}{}^3K_{iklp}\Omega_{kuv}\Omega_{lwz}\Phi_{px}q_jq_uq_vq_wq_zq_x + \\ & \frac{1}{2}\Phi_{iI}{}^3K_{iklp}\Phi_{ku}\Phi_{lw}\Omega_{pxy}q_uq_wq_xq_y + \\ & \frac{1}{4}\Phi_{iI}{}^3K_{iklp}\Phi_{ku}\Omega_{lwz}\Omega_{pxy}q_uq_wq_zq_xq_y + \\ & \frac{1}{4}\Phi_{iI}{}^3K_{iklp}\Omega_{kuv}\Phi_{lw}\Omega_{pxy}q_uq_vq_wq_xq_y + \\ & \frac{1}{8}\Phi_{iI}{}^3K_{iklp}\Theta_{kuv}\Omega_{lwz}\Omega_{pxy}q_uq_vq_wq_zq_xq_y + \\ & \frac{1}{2}\Theta_{iIj}{}^3K_{iklp}\Phi_{ku}\Phi_{lw}\Omega_{pxy}q_jq_uq_wq_xq_y + \\ & \frac{1}{4}\Theta_{iIj}{}^3K_{iklp}\Phi_{ku}\Omega_{lwz}\Omega_{pxy}q_jq_uq_wq_zq_xq_y + \\ & \frac{1}{4}\Theta_{iIj}{}^3K_{iklp}\Omega_{kuv}\Phi_{lw}\Omega_{pxy}q_jq_uq_vq_wq_xq_y + \\ & \frac{1}{8}\Theta_{iIj}{}^3K_{iklp}\Omega_{kuv}\Omega_{lwz}\Omega_{pxy}q_jq_uq_vq_wq_zq_xq_y.\end{aligned} \tag{3.113}$$

The reduced cubic forces (3.113) can be simplified to the following expression

$$\begin{gathered}
{}^3\hat{f}_I = {}^3\mathcal{K}^{\Phi\Phi\Phi\Phi}_{Iuwx} q_u q_w q_x + \left(\frac{1}{2}{}^3\mathcal{K}^{\Phi\Phi\Omega\Phi}_{Iuwxz} + \frac{1}{2}{}^3\mathcal{K}^{\Phi\Omega\Phi\Phi}_{Iuwxz} + \frac{1}{2}{}^3\mathcal{K}^{\Phi\Phi\Phi\Omega}_{Iuwxz} + {}^3\mathcal{K}^{\Theta\Phi\Phi\Phi}_{Iuwxz}\right) q_u q_w q_x q_z + \\
\left(\frac{1}{4}{}^3\mathcal{K}^{\Phi\Omega\Omega\Phi}_{Iuvwxz} + \frac{1}{4}{}^3\mathcal{K}^{\Phi\Omega\Phi\Omega}_{Iuvwxz} + \frac{1}{4}{}^3\mathcal{K}^{\Phi\Phi\Omega\Omega}_{Iuvwxz} + \frac{1}{2}{}^3\mathcal{K}^{\Theta\Omega\Phi\Phi}_{Iuvwxz} + \frac{1}{2}{}^3\mathcal{K}^{\Theta\Phi\Omega\Phi}_{Iuvwxz} + \frac{1}{2}{}^3\mathcal{K}^{\Theta\Phi\Omega\Phi}_{Iuvwxz} + \right) q_u q_v q_w q_x q_z + \\
\left(\frac{1}{4}{}^3\mathcal{K}^{\Theta\Omega\Omega\Phi}_{Iuvwxzp} + \frac{1}{4}{}^3\mathcal{K}^{\Theta\Omega\Phi\Omega}_{Iuvwxzp} + \frac{1}{4}{}^3\mathcal{K}^{\Theta\Phi\Omega\Omega}_{Iuvwxzp} + \frac{1}{8}{}^3\mathcal{K}^{\Phi\Omega\Omega\Omega}_{Iuvwxzp}\right) q_u q_v q_w q_x q_z q_p + \\
\frac{1}{8}{}^3\mathcal{K}^{\Theta\Omega\Omega\Omega}_{Iuvwxzpr} q_u q_v q_w q_x q_z q_p q_r
\end{gathered} \tag{3.114}$$

where,

$$\begin{aligned}
{}^3\mathcal{K}^{\Phi\Phi\Phi\Phi}_{IJKL} &= \Phi_{iI}{}^3K_{ijkl}\Phi_{jJ}\Phi_{kK}\Phi_{lL} \\
\circ \quad {}^3\mathcal{K}^{\Phi\Phi\Omega\Phi}_{IJKLM} &= \Phi_{iI}{}^3K_{ijkl}\Phi_{jJ}\Omega_{kKL}\Phi_{lM} \\
{}^3\mathcal{K}^{\Phi\Omega\Phi\Phi}_{IJKLM} &= \Phi_{iI}{}^3K_{ijkl}\Omega_{jJK}\Phi_{kL}\Phi_{lM} \\
\circ \quad {}^3\mathcal{K}^{\Phi\Phi\Phi\Omega}_{IJKLM} &= \Phi_{iI}{}^3K_{ijkl}\Phi_{jJ}\Phi_{kK}\Omega_{lLM} \\
{}^3\mathcal{K}^{\Theta\Phi\Phi\Phi}_{IJKLM} &= \Theta_{iIJ}{}^3K_{ijkl}\Phi_{jK}\Phi_{kL}\Phi_{lM} \\
\star \quad {}^3\mathcal{K}^{\Phi\Omega\Omega\Phi}_{IJKLMN} &= \Phi_{iI}{}^3K_{ijkl}\Omega_{jJK}\Omega_{kLM}\Phi_{lN} \\
\star \quad {}^3\mathcal{K}^{\Phi\Omega\Phi\Omega}_{IJKLMN} &= \Phi_{iI}{}^3K_{ijkl}\Omega_{jJK}\Phi_{kL}\Omega_{lMN} \\
{}^3\mathcal{K}^{\Phi\Phi\Omega\Omega}_{IJKLMN} &= \Phi_{iI}{}^3K_{ijkl}\Phi_{jJ}\Omega_{kKL}\Omega_{lMN} \\
{}^3\mathcal{K}^{\Theta\Omega\Phi\Phi}_{IJKLMN} &= \Theta_{iIJ}{}^3K_{ijkl}\Omega_{jKL}\Phi_{kM}\Phi_{lN} \\
\diamond \quad {}^3\mathcal{K}^{\Theta\Phi\Omega\Phi}_{IJKLMN} &= \Theta_{iIJ}{}^3K_{ijkl}\Phi_{jJ}\Omega_{kKL}\Phi_{lMN} \\
\diamond \quad {}^3\mathcal{K}^{\Theta\Phi\Phi\Omega}_{IJKLMN} &= \Theta_{iIJ}{}^3K_{ijkl}\Phi_{jK}\Phi_{kL}\Omega_{lMN} \\
\dagger \quad {}^3\mathcal{K}^{\Theta\Omega\Omega\Phi}_{IJKLMNP} &= \Theta_{iIJ}{}^3K_{ijkl}\Omega_{jKL}\Omega_{kMN}\Phi_{lP} \\
{}^3\mathcal{K}^{\Theta\Phi\Omega\Omega}_{IJKLMNP} &= \Theta_{iIJ}{}^3K_{ijkl}\Phi_{jK}\Omega_{kLM}\Omega_{lNP} \\
\dagger \quad {}^3\mathcal{K}^{\Theta\Omega\Phi\Omega}_{IJKLMNP} &= \Theta_{iIJ}{}^3K_{ijkl}\Omega_{jKL}\Phi_{kM}\Omega_{lNP} \\
{}^3\mathcal{K}^{\Phi\Omega\Omega\Omega}_{IJKLMNP} &= \Phi_{iI}{}^3K_{ijkl}\Omega_{jJK}\Omega_{kLM}\Omega_{lNP}. \\
{}^3\mathcal{K}^{\Theta\Omega\Omega\Omega}_{IJKLMNPR} &= \Theta_{iIJ}{}^3K_{ijkl}\Omega_{jKL}\Omega_{kMN}\Omega_{lPR}.
\end{aligned} \tag{3.115}$$

Since ${}^3K_{IJKL} = {}^3K_{IJLK} \forall L, K$, it is easy to see that $\circ$ ${}^3_5\boldsymbol{\mathcal{K}}^{\Phi\Phi\Omega\Phi} = {}^3_5\boldsymbol{\mathcal{K}}^{\Phi\Phi\Phi\Omega}$, $\star$ ${}^3_6\boldsymbol{\mathcal{K}}^{\Phi\Omega\Omega\Phi} = {}^3_6\boldsymbol{\mathcal{K}}^{\Phi\Omega\Phi\Omega}$, $\diamond$ ${}^3_6\boldsymbol{\mathcal{K}}^{\Theta\Phi\Omega\Phi} = {}^3_6\boldsymbol{\mathcal{K}}^{\Theta\Phi\Phi\Omega}$, $\dagger$ ${}^3_7\boldsymbol{\mathcal{K}}^{\Theta\Omega\Omega\Phi} = {}^3_7\boldsymbol{\mathcal{K}}^{\Theta\Omega\Phi\Omega}$.

Finally, the elastic term operators of Eqn. 3.107 are given by

$$\begin{aligned}
{}^2\mathcal{K}_{Iij} &= {}^2\mathcal{K}^{\Phi\Phi\Phi}_{Iij} + \mathcal{K}^{\Theta\Phi}_{Iij} + \frac{1}{2}\mathcal{K}^{\Phi\Omega}_{Iij} \\
{}^3\mathcal{K}_{Iijk} &= \frac{1}{2}\mathcal{K}^{\Theta\Omega}_{Iijk} + \left(\frac{1}{2}{}^2\mathcal{K}^{\Phi\Omega\Phi}_{Iijk} + \frac{1}{2}{}^2\mathcal{K}^{\Phi\Phi\Omega}_{Iijk} + {}^2\mathcal{K}^{\Theta\Phi\Phi}_{Iijk}\right) + {}^3\mathcal{K}^{\Phi\Phi\Phi\Phi}_{Iijk} \\
{}^4\mathcal{K}_{Iijkl} &= \left(\frac{1}{4}{}^2\mathcal{K}^{\Phi\Omega\Omega}_{Iijkl} + \frac{1}{2}{}^2\mathcal{K}^{\Theta\Omega\Phi}_{Iijkl} + \frac{1}{2}{}^2\mathcal{K}^{\Theta\Phi\Omega}_{Iijkl}\right) + \left(\frac{1}{2}{}^3\mathcal{K}^{\Phi\Omega\Phi\Phi}_{Iijkl} + {}^3\mathcal{K}^{\Phi\Phi\Omega\Phi}_{Iijkl} + {}^3\mathcal{K}^{\Theta\Phi\Phi\Phi}_{Iijkl}\right) \\
{}^5\mathcal{K}_{Iijklp} &= \frac{1}{4}{}^2\mathcal{K}^{\Theta\Omega\Omega}_{Iijklp} + \frac{1}{2}\left({}^3\mathcal{K}^{\Phi\Omega\Omega\Phi}_{Iijklp} + \frac{1}{2}{}^3\mathcal{K}^{\Phi\Phi\Omega\Omega}_{Iijklp} + {}^3\mathcal{K}^{\Theta\Omega\Phi\Phi}_{Iijklp} + 2{}^3\mathcal{K}^{\Theta\Phi\Omega\Phi}_{Iijklp}\right) \\
{}^6\mathcal{K}_{Iijklpr} &= \left(\frac{1}{2}{}^3\mathcal{K}^{\Theta\Omega\Omega\Phi}_{Iijklpr} + \frac{1}{4}{}^3\mathcal{K}^{\Theta\Phi\Omega\Omega}_{Iijklpr} + \frac{1}{8}{}^3\mathcal{K}^{\Phi\Omega\Omega\Omega}_{Iijklpr}\right) \\
{}^7\mathcal{K}_{Iijklprs} &= \frac{1}{8}{}^3\mathcal{K}^{\Theta\Omega\Omega\Omega}_{Iijklprs}
\end{aligned} \tag{3.116}$$

or,

$$
\begin{aligned}
{}^{2}_{3}\boldsymbol{\mathcal{K}} &= {}^{2}_{3}\boldsymbol{\mathcal{K}}^{\Phi\Phi\Phi} + {}_{3}\boldsymbol{\mathcal{K}}^{\Theta\Phi} + \frac{1}{2}{}_{3}\boldsymbol{\mathcal{K}}^{\Phi\Omega} \\
{}^{3}_{4}\boldsymbol{\mathcal{K}} &= \frac{1}{2}{}_{4}\boldsymbol{\mathcal{K}}^{\Theta\Omega} + \left(\frac{1}{2}{}^{2}_{4}\boldsymbol{\mathcal{K}}^{\Phi\Omega\Phi} + \frac{1}{2}{}^{2}_{4}\boldsymbol{\mathcal{K}}^{\Phi\Phi\Omega} + {}^{2}_{4}\boldsymbol{\mathcal{K}}^{\Theta\Phi\Phi}\right) + {}^{3}_{4}\boldsymbol{\mathcal{K}}^{\Phi\Phi\Phi\Phi} \\
{}^{4}_{5}\boldsymbol{\mathcal{K}} &= \left(\frac{1}{4}{}^{2}_{5}\boldsymbol{\mathcal{K}}^{\Phi\Omega\Omega} + \frac{1}{2}{}^{2}_{5}\boldsymbol{\mathcal{K}}^{\Theta\Omega\Phi} + \frac{1}{2}{}^{2}_{5}\boldsymbol{\mathcal{K}}^{\Theta\Phi\Omega}\right) + \left(\frac{1}{2}{}^{3}_{5}\boldsymbol{\mathcal{K}}^{\Phi\Omega\Phi\Phi} + {}^{3}_{5}\boldsymbol{\mathcal{K}}^{\Phi\Phi\Omega\Phi} + {}^{3}_{5}\boldsymbol{\mathcal{K}}^{\Theta\Phi\Phi\Phi}\right) \\
{}^{5}_{6}\boldsymbol{\mathcal{K}} &= \frac{1}{4}{}^{2}_{6}\boldsymbol{\mathcal{K}}^{\Theta\Omega\Omega} + \frac{1}{2}\left({}^{3}_{6}\boldsymbol{\mathcal{K}}^{\Phi\Omega\Omega\Phi} + \frac{1}{2}{}^{3}_{6}\boldsymbol{\mathcal{K}}^{\Phi\Phi\Omega\Omega} + {}^{3}_{6}\boldsymbol{\mathcal{K}}^{\Theta\Omega\Phi\Phi} + 2{}^{3}_{6}\boldsymbol{\mathcal{K}}^{\Theta\Phi\Omega\Phi}\right) \\
{}^{6}_{7}\boldsymbol{\mathcal{K}} &= \left(\frac{1}{2}{}^{3}_{7}\boldsymbol{\mathcal{K}}^{\Theta\Omega\Omega\Phi} + \frac{1}{4}{}^{3}_{7}\boldsymbol{\mathcal{K}}^{\Theta\Phi\Omega\Omega} + \frac{1}{8}{}^{3}_{7}\boldsymbol{\mathcal{K}}^{\Phi\Omega\Omega\Omega}\right) \\
{}^{7}_{8}\boldsymbol{\mathcal{K}} &= \frac{1}{8}{}^{3}_{8}\boldsymbol{\mathcal{K}}^{\Theta\Omega\Omega\Omega}
\end{aligned}
\tag{3.117}
$$

Jacobian for Time Integration

The Newmark time integration can be applied to reduced equations in the tensorial form. In doing so, the Jacobian can always be assembled in the modal space ($\mathbf{q}$) using tensors instead of the physical space like in the regular approach.

$$
\mathbf{r}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = \hat{\mathbf{f}}^{in} + \hat{\mathbf{f}}^{dam} + \hat{\mathbf{f}}^{lin} + {}^{2}\hat{\mathbf{f}} + {}^{3}\hat{\mathbf{f}} - \hat{\mathbf{f}}^{ext} = \mathbf{0} \tag{3.118}
$$

Using Newmark's scheme, the Jacobian can be calculated as follows:

$$
\mathbf{S}(\mathbf{q}) = \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{q}} = \underbrace{\frac{\partial \mathbf{r}}{\partial \mathbf{q}}}_{\boldsymbol{\mathcal{K}}^{t}} + \frac{\gamma}{\beta h}\underbrace{\frac{\partial \mathbf{r}}{\partial \dot{\mathbf{q}}}}_{\boldsymbol{\mathcal{C}}^{t}} + \frac{1}{\beta h^{2}}\underbrace{\frac{\partial \mathbf{r}}{\partial \ddot{\mathbf{q}}}}_{\boldsymbol{\mathcal{M}}^{t}} \tag{3.119}
$$

where,

$$
\left.
\begin{aligned}
\mathcal{M}^{t}_{IJ} =& \delta_{IJ} + \mathcal{M}^{\Theta\Theta}_{IiJk}q_i q_k + \mathcal{M}^{\Phi\Theta}_{IJk}q_k + \mathcal{M}^{\Theta\Phi}_{IiJ}q_i \\
\mathcal{C}^{t}_{IJ} =& \mathcal{M}^{\Theta\Theta}_{IiJk}q_i\dot{q}_k + \mathcal{M}^{\Theta\Theta}_{IikJ}q_i\dot{q}_k + \mathcal{M}^{\Phi\Theta}_{IJk}\dot{q}_k + \mathcal{M}^{\Phi\Theta}_{IkJ}\dot{q}_k + \mathcal{C}^{\Phi\Phi}_{IJ} + \mathcal{C}^{\Phi\Theta}_{IJj}q_j + \\
& \mathcal{C}^{\Theta\Phi}_{IjJ}q_j + \mathcal{C}^{\Theta\Theta}_{IjJk}q_j q_k \\
=& \underbrace{2\mathcal{M}^{\Theta\Theta}_{IiJk}q_i\dot{q}_k + 2\mathcal{M}^{\Phi\Theta}_{IJk}\dot{q}_k}_{\text{inertial terms}} + \underbrace{\mathcal{C}^{\Phi\Phi}_{IJ} + \mathcal{C}^{\Phi\Theta}_{IJj}q_j + \mathcal{C}^{\Theta\Phi}_{IjJ}q_j + \mathcal{C}^{\Theta\Theta}_{IjJk}q_j q_k}_{\text{damping terms}} \\
\mathcal{K}^{t}_{IJ} =& (\mathcal{M}^{\Theta\Theta}_{IkjJ} + \mathcal{M}^{\Theta\Theta}_{IJjk})\ddot{q}_j q_k + \mathcal{M}^{\Theta\Theta}_{IJjk}\dot{q}_j\dot{q}_k + (\mathcal{M}^{\Phi\Theta}_{IiJ} + \mathcal{M}^{\Theta\Phi}_{IJi})\ddot{q}_i + \\
& (\mathcal{C}^{\Phi\Theta}_{IiJ} + \mathcal{C}^{\Theta\Phi}_{IJi})\dot{q}_i + (\mathcal{C}^{\Theta\Theta}_{IJjk} + \mathcal{C}^{\Theta\Theta}_{IkjJ})\dot{q}_j q_k \\
& \underbrace{(\delta_{IJ})(\omega_I^2)}_{\text{no summation}} + \left({}^{2}\mathcal{K}_{IJj} + {}^{2}\mathcal{K}_{IjJ}\right)q_j + \left({}^{3}\mathcal{K}_{IJjk} + {}^{3}\mathcal{K}_{IjJk} + {}^{3}\mathcal{K}_{IjkJ}\right)q_j q_k + \\
& \left({}^{4}\mathcal{K}_{IJjkl} + {}^{4}\mathcal{K}_{IjJkl} + {}^{4}\mathcal{K}_{IjkJl} + {}^{4}\mathcal{K}_{IjklJ}\right)q_j q_k q_l + \\
& \left({}^{5}\mathcal{K}_{IJjklm} + {}^{5}\mathcal{K}_{IjJklm} + {}^{5}\mathcal{K}_{IjkJlm} + {}^{5}\mathcal{K}_{IjklJm} + {}^{5}\mathcal{K}_{IjklmJ}\right)q_j q_k q_l q_m + \\
& \big({}^{6}\mathcal{K}_{IJjklmn} + {}^{6}\mathcal{K}_{IjJklmn} + {}^{6}\mathcal{K}_{IjkJlmn} + {}^{6}\mathcal{K}_{IjklJmn} + {}^{6}\mathcal{K}_{IjklmJn} + \\
& {}^{6}\mathcal{K}_{IjklmnJ}\big)q_j q_k q_l q_m q_n + \\
& \big({}^{7}\mathcal{K}_{IJjklmnp} + {}^{7}\mathcal{K}_{IjJklmnp} + {}^{7}\mathcal{K}_{IjkJlmnp} + {}^{7}\mathcal{K}_{IjklJmnp} + {}^{7}\mathcal{K}_{IjklmJnp} + \\
& {}^{7}\mathcal{K}_{IjklmnJp} + {}^{7}\mathcal{K}_{IjklmnpJ}\big)q_j q_k q_l q_m q_n q_p - \Theta_{iIJ}p_i
\end{aligned}
\right\}
\tag{3.120}
$$

Implementation of Tensors

The implementation of tensors especially in a finite element setting can be done in an efficient manner at the element level. The higher order tensors ${}^{2}_{3}\mathbf{K}$ and ${}^{3}_{4}\mathbf{K}$ (Equation (3.88)) though sparse in nature are still huge in size. From Equations (3.112), (3.115) and (3.116), it seems that the higher order tensors required for the formation of reduced equations 3.107 need the full tensors ${}^2_3\mathbf{K}$ and ${}^3_4\mathbf{K}$ which would be expensive not only in terms of time, but also memory. But in practice, they can be obtained by just summing over the element level contributions similar to as explained in Section 3.3.3. For example the third order tensor ${}^2_3\boldsymbol{\mathcal{K}}^{\Phi\Phi\Phi}$ required for the calculation of ${}^2_3\boldsymbol{\mathcal{K}}$ can be calculated at the element level as follows

$$
{}^2_3\boldsymbol{\mathcal{K}}^{\Phi\Phi\Phi} = ((\boldsymbol{\Phi}^T \cdot {}^2_3\mathbf{K}) \cdot \boldsymbol{\Phi}) \cdot \boldsymbol{\Phi} = \sum_{e=1}^{n_e} (((\boldsymbol{\Phi}^{\boldsymbol{e}})^T \cdot {}^2_3\mathbf{K}^{\mathbf{e}}) \cdot_{21} \boldsymbol{\Phi}^{\boldsymbol{e}}) \cdot_{21} \boldsymbol{\Phi}^{\boldsymbol{e}}, \tag{3.121}
$$

where $\boldsymbol{\Phi}^{\boldsymbol{e}} \in \mathbb{R}^{N_e \times m}$ represents the restriction of the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n\times m}$ to the rows indexed by the DOFs corresponding to element e, N_e being the numbers of DOFs that e shares in the physical domain; ${}^2_3\mathbf{K}^{\mathbf{e}} \in \mathbb{R}^{N_e \times N_e \times N_e}$ represents the element level contribution of ${}^2_3\mathbf{K}$ towards the internal force such that

$$
\mathbf{f}^e(\mathbf{u}^e) = \mathbf{K}^e \cdot \mathbf{u}^e + {}^2_3\mathbf{K}^e : (\mathbf{u}^e \otimes \mathbf{u}^e) + {}^3_4\mathbf{K}^e \vdots (\mathbf{u}^e \otimes \mathbf{u}^e \otimes \mathbf{u}^e) \tag{3.122}
$$

See Appendix A.1 for formulation of element tensors ${}^2_3\mathbf{K}^e$ and ${}^3_4\mathbf{K}^e$.

3.4.2. Difference between Regular & Tensorial Approaches

The reduced equations can be concisely written as follows

$$
\left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right]^T \left(\mathbf{M}\frac{d^2\mathbf{u}(\mathbf{q})}{dt^2} + \mathbf{f}(\mathbf{u}(\mathbf{q}))\right) = \left[\frac{\partial \mathbf{u}}{\partial \mathbf{q}}\right]^T \mathbf{g}(t)
$$

In the regular approach, the Jacobian for N-R iterations is approximated by projecting the Tangent stiffness $\mathbf{K}(\mathbf{u}) = \dfrac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}$, on to the tangent subspace of the manifold as follows :

$$
\begin{aligned}
\mathbf{S}(\mathbf{q}) &= \left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right]^T \mathbf{K}(\mathbf{u}(\mathbf{q})) \left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right] && (3.123)\\
&= \left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right]^T \left.\frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}(\mathbf{q})} \left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right] && (3.124)\\
&= \left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right]^T \frac{\partial \mathbf{f}(\mathbf{u}(\mathbf{q}))}{\partial \mathbf{q}} && (3.125)
\end{aligned}
$$

But in the tensorial approach, the Jacobian is calculated accurately as :

$$
\begin{aligned}
\mathbf{r}(\mathbf{q}) &= \left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right]^T \left(\mathbf{M}\frac{d^2\mathbf{u}(\mathbf{q})}{dt^2} + \mathbf{f}(\mathbf{u}(\mathbf{q}))\right) - \left[\frac{\partial \mathbf{u}}{\partial \mathbf{q}}\right]^T \mathbf{g}(t) = \mathbf{0}\\
\mathbf{S}(\mathbf{q}) &= \frac{\partial \mathbf{r}(\mathbf{q})}{\partial \mathbf{q}}
\end{aligned}
$$

Still, if the convective terms (arising from the inertial terms in Equation (3.18)) and the external load terms in the Jacobian are neglected, the following approximation to the Jacobian can be obtained:

$$
\mathbf{S}(\mathbf{q}) = \frac{\partial}{\partial \mathbf{q}} \left(\left[\frac{\partial \mathbf{u}(\mathbf{q})}{\partial \mathbf{q}}\right]^T \mathbf{f}(\mathbf{u}(\mathbf{q}))\right) \tag{3.126}
$$

It is easy to see that the two Jacobian approximations in Equations (3.125) and (3.126) are completely different, the former being far from accurate. Using chain rule, it can be seen that it captures non-linearities only partially compared to that in Equation (3.126).

3.4.3. Mapping Error : Linear vs Quadratic

Upon constraining a high dimensional solution to a low dimensional subspace through a mapping (linear or nonlinear), usually an error is generated in the reduced solution compared to the high dimensional one. It is easy to see that this error in case of quadratic manifold would be lower bounded (not strictly) by that in the Linear manifold for the same number of VMs used.

For a linear manifold containing VMs and all the corresponding MDs, the mapping can be written as

$$\mathbf{u}(\mathbf{q},\boldsymbol{\eta}) = \boldsymbol{\Phi}\mathbf{q} + \boldsymbol{\Pi}\boldsymbol{\eta}, \tag{3.127}$$

where $\mathbf{q} \in \mathbb{R}^m$ is the unknown amplitude for the VMs in $\boldsymbol{\Phi} \in \mathbb{R}^{n\times m}$, $\boldsymbol{\eta} \in \mathbb{R}^{m(m+1)/2}$ are the unknown amplitudes for the MDs contained in the matrix $\boldsymbol{\Pi} \in \mathbb{R}^{n\times m(m+1)/2}$ and for a quadratic manifold, the mapping can be written as:

$$\mathbf{u}(\mathbf{q}) = \boldsymbol{\Phi}\mathbf{q} + \boldsymbol{\Xi}(\mathbf{q})\mathbf{q}, \tag{3.128}$$

where $\mathbf{q}, \boldsymbol{\Phi}$ have the same definitions as in the linear case, but the second part is nonlinear in $\mathbf{q}$ instead of being independent of it as in the linear case. Here $\boldsymbol{\Xi}(\mathbf{q})$ is nothing but $\frac{1}{2}\boldsymbol{\Omega}.\mathbf{q}$ from Equation (3.64). Note that $\boldsymbol{\Xi}(\mathbf{q})$ contains the same MDs which are present in $\boldsymbol{\Pi}$. This implies that the amplitudes for MDs becomes constrained to the values of $\mathbf{q}$. This has its benefits since this mapping is based on Taylor expansion and tries to capture the same nonlinear behavior with lesser number of unknowns but in a more general sense it has limited the freedom to reduce the error in a larger subspace. Thus, it is safe to say that the error generated by a linear mapping (of VMs enriched with all corresponding MDs) would act as a lower bound for that by a quadratic mapping with same number of VMs.

4

Hyper-Reduction

Previous chapter discussed the projection based model-order reduction techniques and their implementation in the "regular" and the "tensorial" sense. One of the reasons why tensorial approach seems useful is because the evaluation of non-linearity and Jacobian which become inhibitive for regular approach as system size becomes bigger, can be evaluated offline using tensors and online operations become independent of model size. The tensorial approaches on the other hand involve significant offline costs and thus in the case of frequent basis changes (arising due to design/load change) might lose their importance. They're also limited in applicability to polynomial non-linearities. In this context, the *hyper-reduction* class of techniques seems useful whereby the non-linearity and Jacobian evaluation is very cheaply approximated by evaluation at only a few nodes and elements. In a broad sense, the offline costs involved hyper-reduction are affordable except for the evaluation of training sets which might require a full solution run. Thus, if these training sets can be made available in a cheap manner, then it would be a very desirable balance between *regular* and *tensorial* approaches in terms of offline and online costs.

Some recent hyper-reduction methods which are relevant in the current (finite element) have been discussed here. Then some methods for the cheap evaluation of training sets required for these hype-reduction methods are proposed. Thus, this chapter tries to address the latter of the two sub research questions i.e.

Can the hyper-reduction techniques be effectively used for model order reduction of nonlinear structural dynamical system without the need of a full solution run ?

4.1. DEIM

It is to be realised that the Galerkin projection of nonlinear term ($\mathbf{f}(\mathbf{u})$) as in Equation (3.4), introduces an expensive *online* cost. It involves two basic steps. One is the evaluation of vector valued nonlinear function $\mathbf{f}(\mathbf{u})$ and the other is projection of the evaluated force on to the reduction basis.

$$\underbrace{\mathbf{f}_r}_{\in\mathbb{R}^m} = \underbrace{\mathbf{V}^T}_{\in\mathbb{R}^{m\times n}} \underbrace{\mathbf{f}(\mathbf{V}\mathbf{q})}_{\in\mathbb{R}^n} \tag{4.1}$$

The evaluation of nonlinear term doesn't happen in the reduced space but in the physical space whereby internal force vector has to be assembled for every element during each iteration *online*. Same holds true for the tangent stiffness matrix given by the Jacobian of $\mathbf{f}(\mathbf{u})$ which is used during the iterative solution of reduced equations.

$$\mathbf{K}(\mathbf{u}) = \frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}} \in \mathbb{R}^{n\times n} \tag{4.2}$$

$$\mathbf{K}_r(\mathbf{q}) = \mathbf{V}^T\mathbf{K}(\mathbf{V}\mathbf{q})\mathbf{V} \in \mathbb{R}^{m\times m} \tag{4.3}$$

The Discrete Empirical Interpolation Method was proposed in [3] as an effective way to approximate the evaluation of nonlinear vector $\mathbf{f}(\mathbf{u})$ in a speedy manner. The method reduces the computational effort by simply evaluating the internal force at selected points of interest and interpolating the force at other points.

The essential idea is that the nonlinear function $\mathbf{f}(\mathbf{u})$ is first projected onto a subspace spanned by a basis $\mathbf{D}$ of dimension $k \ll n$ that approximates the space generated by $\mathbf{f}(\mathbf{u})$. One of the choices of the basis can be the POD basis obtained from the SVD of ensemble of nonlinear function snapshots $([\mathbf{f}(\mathbf{u}(t_1)), \ldots, \mathbf{f}(\mathbf{u}(t_{n_s}))])$ generated from the nonlinear solution of original full system. Then this approximation can be written as

$$\mathbf{f}(\mathbf{u}(t)) \approx \mathbf{D}\mathbf{c}(t) \tag{4.4}$$

where $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{n \times k}$ and $\mathbf{c}(t) \in \mathbb{R}^k$ is the coefficient vector. To solve for $\mathbf{c}(t)$, k distinct rows can be chosen from the overdetermined system 4.4. This can be accomplished with a boolean matrix $\mathbf{E} = [\mathbf{e}_{\rho_1}, \ldots, \mathbf{e}_{\rho_k}] \in \mathbb{R}^{n \times k}$ where $\mathbf{e}_{\rho_i}$ represents the ρ_i^{th} column of the identity matrix $\mathbf{I}_n$. If $\mathbf{E}^T\mathbf{V}$ is not singular, then the system can be solved as

$$\mathbf{E}^T\mathbf{f}(\mathbf{u}(t)) = (\mathbf{E}^T\mathbf{D})\mathbf{c}(t) \tag{4.5}$$

$$\implies \mathbf{c}(t) = (\mathbf{E}^T\mathbf{D})^{-1}\mathbf{E}^T\mathbf{f}(\mathbf{u}(t)) \tag{4.6}$$

From this solution, the approximation in Equation (4.4) becomes

$$\mathbf{f}(\mathbf{u}(t)) \approx \mathbf{D}\mathbf{c}(t) = \underbrace{\mathbf{D}(\mathbf{E}^T\mathbf{D})^{-1}}_{precomputed} \quad \underbrace{\mathbf{E}^T\mathbf{f}(\mathbf{u}(t))}_{selective\ evaluation} \tag{4.7}$$

The term $\mathbf{E}^T\mathbf{f}(\mathbf{u}(t))$ indicates that the nonlinear vector $\mathbf{f}(\mathbf{u}(t))$ needs to be evaluated at only a few locations as specified by the boolean matrix $\mathbf{E}^T$ and the term $\mathbf{D}(\mathbf{E}^T\mathbf{D})^{-1}$ can be computed *offline* and provides the interpolation to other locations. The two basic requirements for this technique are the reduction basis for the internal force $\mathbf{D}$ (which as explained above can be obtained by POD from the SVD of nonlinear function snapshots) and the boolean matrix $\mathbf{E}$ which gives the sparse selection points on which the nonlinear function has to be evaluated. This selection of points is obtained from Algorithm 1.

Algorithm 1 DEIM point selection

```
Input: ${\mathbf{d}_i\}_{i=1}^k \subset \mathbb{R}^n$
Output: $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_k]^T \in \mathbb{R}^k$, $\mathbf{B} \in \mathbb{R}^{n \times k}$

$[|\varrho|, \rho_1] = \max |\mathbf{d}_1|$        ▷ max: returns maximum value in a vector followed by its location (index)
$\mathbf{D} = [\mathbf{d}_1]$, $\mathbf{E} = [\mathbf{e}_{\rho_1}]$, $\boldsymbol{\rho} = [\rho_1]$
for $i \leftarrow 2$ to $k$ do
    Solve $\mathbf{E}^T\mathbf{d}_i = (\mathbf{E}^T\mathbf{D})\mathbf{c}$ for $\mathbf{c}$
    $\mathbf{r} = \mathbf{d}_i - \mathbf{D}\mathbf{c}$
    $[|\varrho|, \rho_i] = \max |\mathbf{r}|$
    $\mathbf{D} \leftarrow [\mathbf{D} \;\; \mathbf{d}_i]$, $\mathbf{E} \leftarrow [\mathbf{E} \;\; \mathbf{e}_{\rho_i}]$, $\boldsymbol{\rho} \leftarrow \begin{bmatrix} \boldsymbol{\rho} \\ \rho_i \end{bmatrix}$
end for
```

Inefficiencies in DEIM: DEIM application were shown in [3] for systems discretized using finite differences and where the nonlinear function $\mathbf{f}(\mathbf{u})$ is applied component wise to the argument $\mathbf{u}$ i.e. $f_i = f_i(u_i)$. This results in $\mathbf{E}^T\mathbf{f}(\mathbf{u}) = \mathbf{f}(\mathbf{E}^T\mathbf{u})$ and evaluation of $\mathbf{f}(\mathbf{u})$ at a few indices requires only as many evaluations of the nonlinear function. In the finite element context, the DEIM is particularly inefficient due to the fact that the nonlinear function $\mathbf{f}(\mathbf{u})$ evaluation at a few DOFs(nodes) requires it to be evaluated at all the elements connected to the corresponding nodes. This happens because entries in the $\mathbf{f}$ vector are not component wise related to those in $\mathbf{u}$. In principle if all elements in a model are connected to a single node (or DOF), then the cost of evaluation of nonlinear function at that particular node is the same as that of evaluating it at all the nodes. This is because the boolean matrix cannot be brought inside the function argument i.e.

$$\mathbf{E}^T\mathbf{f}(\mathbf{u}) \neq \underbrace{\mathbf{f}(\mathbf{E}^T\mathbf{u})}_{\text{sparse evaluation}} \tag{4.8}$$

As also shown in [4], this can be explained with the help of the Figure 4.1. Alternative formulations for DEIM such as Unassembled DEIM (UDEIM), Surrogate DEIM (SDEIM) and Surrogate Unassembled DEIM (SUDEIM) were proposed in [4] which are particularly efficient in the nonlinear finite element structural dynamics context. Here the authors propose to use the unassembled version of nonlinear internal forces in constructing the ROB and the Boolean matrix. By using unassembled forces, the sizes of **D** and **E** increases thereby increasing the offline cost, but the nonlinear force needs to be calculated in only one element per column of **E**.

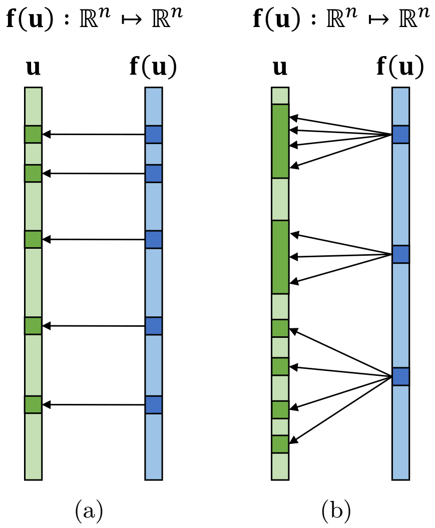

$\mathbf{f}(\mathbf{u}) : \mathbb{R}^n \mapsto \mathbb{R}^n$
$\mathbf{u}$
$\mathbf{f}(\mathbf{u})$
$\mathbf{f}(\mathbf{u}) : \mathbb{R}^n \mapsto \mathbb{R}^n$
$\mathbf{u}$
$\mathbf{f}(\mathbf{u})$
(a)
(b)

Figure 4.1: (a) shows a mapping in which $\mathbf{f}(\mathbf{u})$ is a nonlinear function which acts only component-wise on the argument $\mathbf{u}$. This makes the DEIM efficient since evaluation of $\mathbf{f}(\mathbf{u})$ at a few indices requires only as many function evaluations. (b) shows a general nonlinear mapping where the components of $\mathbf{f}(\mathbf{u})$ could be dependent on any of the components of $\mathbf{u}$. This would make the DEIM inefficient since the evaluation of $\mathbf{f}(\mathbf{u})$ at even a few indices might require many function evaluations.

4.2. ECSW

As mentioned earlier, the Galerkin projection of the nonlinear term $\mathbf{f}(\mathbf{u})$ as in Equation (3.4) involves two sources of inefficiency : the evaluation of the of the nonlinear vector function and the projection onto the subspace. The DEIM [3] (or UDEIM[4] in finite element context) is an efficient hyper-reduction method to tackle the former of these sources of inefficiencies by cheap evaluation of nonlinearity at selected points (or elements). However, the latter still provides room for improvement and the Energy Conserving Sampling and Weighting (ECSW) method recently proposed in [5] is remarkable in this context.

Unlike the other hyper-reduction techniques, the ECSW method directly approximates the projection of residuals and/or Jacobians instead of first approximating these vectors and then projecting the approximated nonlinear function onto the relevant subspace. Furthermore, it has been shown in [21] that the ECSW distinguished itself through its preservation of the Lagrangian structure which leads to numerical stability in case of second order dynamical systems. The DEIM and its variants on the other hand have been shown to be numerically unstable also in [21].

Assuming a precomputed basis (such as a POD basis), the ECSW is applicable in the finite element context by expressing the projected nonlinear force as sum of contributions at element level as follows.

$$\hat{\mathbf{f}}(\mathbf{q}) = \mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{q}) = \sum_{e=1}^{n_e} \mathbf{V}_e^T \mathbf{f}_e(\mathbf{V}_e\mathbf{q}), \tag{4.9}$$

where $\mathbf{V}_e \in \mathbb{R}^{N_e\times m}$ is the restriction of the ROB $\mathbf{V} \in \mathbb{R}^{n\times m}$ to the rows indexed by the DOFs corresponding to element e, n_e being the total number of elements in the model, N_e being the number of DOFs associated with element e and $\mathbf{f}_e \in \mathbb{R}^{N_e}$ is the element level internal force vector for that element. Physically, if each column of $\mathbf{V}$ represents a virtual displacement, then corresponding row of $\hat{\mathbf{f}}(\mathbf{q})$ represents the virtual work done by the internal force. The essence of ECSW is to identify a small set of elements E of size $|E| \ll n_e$ equipped with corresponding positive weights $\boldsymbol{\xi}^\star \in \mathbb{R}^{|E|}$ such that

$$\begin{gathered}\hat{\mathbf{f}}(\mathbf{q}) \approx \sum_{e\in E} \xi_e \mathbf{V}_e^T \mathbf{f}_e(\mathbf{V}_e\mathbf{q}),\\ E = \{e : e \in \{1,\dots,n_e\}, \xi_e^\star > 0\}.\end{gathered} \tag{4.10}$$

Thus, the ECSW aims to preserve the virtual work done by internal force on the set of vectors in the ROB by *sampling* and *weighting* elements. The selected elements and weights are determined to approximate virtual work over chosen training sets which generally come from full solution run(s). If there are n_t training vectors in the set with $\mathbf{u}^{(i)}$ representing the i^{th} vector, then corresponding reduced unknowns $\mathbf{q}^{(i)}$ can be easily calculated using least squares as

$$\mathbf{q}^{(i)} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{u}^{(i)}, \tag{4.11}$$

and element level contribution of projected internal force for each of the training vectors can be assembled in a matrix $\mathbf{G}$ as follows

$$\begin{gathered}\mathbf{G} = \begin{bmatrix} \mathbf{g}_{11} & \cdots & \mathbf{g}_{1n_e} \\ \vdots & \ddots & \vdots \\ \mathbf{g}_{n_t1} & \cdots & \mathbf{g}_{n_tn_e} \end{bmatrix} \in \mathbb{R}^{mn_t\times n_e}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{n_t} \end{bmatrix} \in \mathbb{R}^{mn_t}\\ \mathbf{g}_{ie} = \mathbf{V}_e^T\mathbf{f}_e(\mathbf{V}_e\mathbf{q}^{(i)}), \qquad \mathbf{b}_i = \sum_{e=1}^{n_e} \mathbf{g}_{ie}\end{gathered} \tag{4.12}$$

Here $\mathbf{b}_i$ represents the total projected internal force for the i^{th} training vector. The set of elements and weights can then be chosen by a sparse solution to the following problem

$$\Gamma = \{\boldsymbol{\xi} \in \mathbb{R}^{n_e} : \|\mathbf{G}\boldsymbol{\xi} - \mathbf{b}\|_2 \leq \tau\|\mathbf{b}\|_2,\ \boldsymbol{\xi} \geq \mathbf{0}\}, \tag{4.13}$$

where $\boldsymbol{\xi}$ represents a sparse vector containing the non negative weights $\boldsymbol{\xi}^\star$ at corresponding indices given by E and zeros elsewhere. $\tau \in (0,\ 1)$ is a pre-defined tolerance for controlling accuracy of nonlinear approximation. It is easy to see that $\tau = 0$ would result in $\boldsymbol{\xi} = \boldsymbol{\xi}^\star = \mathbf{1}$ and $E = \{1, 2, \ldots, n_e\}$ whereby all elements are selected. A more practical range for τ such as $[0.01, 0.1]$ gives a possibility of a sparse solution for $\boldsymbol{\xi}$ where $|E| \ll n_e$. The optimum solution would be the one which contains least number of non-zero elements in the vector $\boldsymbol{\xi}$. In other words, involves the solution of a non-negative least squares (NNLS) problem

$$(P1) \quad \min_{\boldsymbol{\xi} \geq \mathbf{0}} \|\mathbf{G}\boldsymbol{\xi} - \mathbf{b}\|_2^2, \tag{4.14}$$

but in a sparse manner. A variant of the classical active-set NNLS algorithm [22] called sparse NNLS (sNNLS) was proposed in [23] and has been used in formulation of ECSW, whereby instead of placing a limit on sparsity of $\boldsymbol{\xi}$ as in sNNLS, the algorithm is tuned to reach the user defined tolerance (τ). The algorithm is shown in Algorithm 2, where $\boldsymbol{\zeta} \in \mathbb{R}^{n_e}$ and $\boldsymbol{\zeta}_E$ and $\mathbf{G}_E$ denote respectively, the restriction of $\boldsymbol{\zeta}$ and column-wise restriction of $\mathbf{G}$ to the elements in the *active* subset E. The Z is the disjoint *inactive* subset which contains the zero entries of $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$.

Algorithm 2 Sparse NNLS for ECSW

Input: $\mathbf{G}, \mathbf{b}$
Output: $\boldsymbol{\xi} \in \mathbb{R}^{n_e}$ sparse, $E \subset \{1, \ldots, n_e\}$

```
$E \leftarrow \emptyset$, $Z \leftarrow \{1, \ldots, n_e\}$, $\boldsymbol{\xi} \leftarrow \mathbf{0} \in \mathbb{R}^{n_e}$
while $\|\mathbf{G}\boldsymbol{\xi} - \mathbf{b}\|_2 > \tau\|\mathbf{b}\|_2$ do
    $\boldsymbol{\mu} \leftarrow \mathbf{G}^T(\mathbf{b} - \mathbf{G}\boldsymbol{\xi})$
    $[|\nu|, e] = \max \boldsymbol{\mu}$        ▷ max: returns maximum value in a vector followed by its location (index)
    $E \leftarrow E \cup \{e\}$, $Z \leftarrow Z \leftarrow Z \backslash \{e\}$
    while true  do
        $\boldsymbol{\zeta}_E \leftarrow \mathbf{G}_E^\dagger \mathbf{b}$        ▷ † represents pseudo-inverse
        $\boldsymbol{\zeta}_Z \leftarrow \mathbf{0}$
        if $\boldsymbol{\zeta}_E > \mathbf{0}$ then
            $\boldsymbol{\xi} \leftarrow \boldsymbol{\zeta}$
            break
        end if
        $\eta = \min_{k \in E} \xi_k / (\xi_k - \zeta_k)$
        $\boldsymbol{\xi} \leftarrow \boldsymbol{\xi} + \eta(\boldsymbol{\zeta} - \boldsymbol{\xi})$
        $Z \leftarrow \{i | \xi_i = 0\}$
        $E \leftarrow \{1, \ldots, n_e\} \backslash Z$
    end while
end while
```

During the implicit time integration of reduced equations, the nonlinear set of algebraic equations is linearised using Newton-Raphson procedure at each time step. This requires the Jacobian of the projected internal force vector or the reduced tangent stiffness matrix. This matrix which is also a nonlinear function of unknowns $\mathbf{q}$, needs to be assembled at each element at every iteration and hence poses high cost. But using the same technique, the reduced tangent stiffness matrix $\hat{\mathbf{K}}(\mathbf{q})$ can be approximated to be evaluated only at the elements in E and weighted by $\boldsymbol{\xi}^\star$ as follows.

$$\hat{\mathbf{K}}(\mathbf{q}) = \frac{\partial \hat{\mathbf{f}}(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial \mathbf{V}^T \mathbf{f}(\mathbf{V}\mathbf{q})}{\partial \mathbf{q}} = \mathbf{V}^T \mathbf{K}(\mathbf{V}\mathbf{q})\mathbf{V} = \sum_{e=1}^{n_e} \mathbf{V}_e^T \mathbf{K}_e(\mathbf{V}_e\mathbf{q})\mathbf{V}_e \tag{4.15}$$

$$\implies \hat{\mathbf{K}}(\mathbf{q}) \approx \sum_{e \in E} \xi_e^\star \mathbf{V}_e^T \mathbf{K}_e(\mathbf{V}_e\mathbf{q})\mathbf{V}_e, \tag{4.16}$$

where $\mathbf{K}(\mathbf{u}) = \dfrac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}$ is the Jacobian of the full internal force vector and $\mathbf{K}_e(\mathbf{u}_e) \in \mathbb{R}^{N_e \times N_e}$ denotes the element matrix contribution of element e to the Jacobian $\mathbf{K}(\mathbf{u})$.

4.3. Hyper-Reduction using Quadratic Manifold

The Hyper-reduction methods such as DEIM, UDEIM or ECSW are nice when it comes to achieving a good speed up for fast computations. The only factor which makes them unfavourable is the requirement of a full solution run which can be computationally expensive, or even unaffordable at a preliminary design stage when a variety of load scenarios, geometric layouts, material choices need to be explored. Each of these cases needs a time integration of its own. One of the focus areas of the research is to avoid such prohibitive *offline* costs which makes the use of such hyper-reduction techniques unfavourable.

In finite element context, the ECSW has indeed emerged as a leader among the hyper-reduction methods. ECSW requires training vectors of displacements to solve a sparse non-negative least squares problem whereby the virtual work done by the nonlinear internal forces on the reduction basis is attempted to be reproduced by attaching weights to selected few elements. It is desirable to not choose these training vectors from a full nonlinear/reduced run since they are relatively expensive.

It is well known that the VMs and MDs form a nice basis to represent the nonlinear displacement field for the range of application. Essentially, if a "smart" combination of VMs and MDs is made to obtain the training sets so that the displacements are in the right range, then it should be able to capture the corresponding nonlinear forces. Finally, the sNNLS algorithm (2) can be applied to find out the elements(in a sparse sense) and corresponding weights such that the virtual work done by the training forces on the basis is accurately approximated. This sample of elements and weights shall then be valid for the nonlinear run in consideration and produce the projected internal forces with good accuracy, hence returning good results.

A variety of ways in which ECSW can be used or hype-reduction are proposed below.

1. **ECSW-I:** Here, a POD basis(of size k) obtained by SVD of the snapshots of the reference nonlinear solution is used as the reduction basis. For comparison purposes, the basis size k is taken to be the same as that of linear manifold. n_t training vectors are chosen from Reference nonlinear solution snapshots at equally spaced time instants. This method would be a typical way in which the ECSW would be used and requires the full nonlinear run.

2. **ECSW-II:** For the nonlinearities being considered, the Linear Manifold (LM) basis is a good basis for reduction. This reduction basis is composed of VMs and all MDs. A reduced solution run can then be obtained. This reduced solution is expected to be a very good approximation to the full solution and training sets can be generated by taking n_t snapshots of this reduced solution at equal spaced intervals. The ECSW applied in this manner would attempt to reproduce the LM reduced solution. This replaces the offline cost of evaluation of the full solution by that of the reduced solution and should make the hyper-reduction cheaper at the expense of marginal loss of accuracy.

3. **ECSW-III:** Though the reduced solution run is cheap compared to a full run, it is still expensive compared to a hyper-reduced run or a linear run. Choosing the same LM basis as in ECSW-II, it is proposed to obtain the training forces in the following manner:

   - perform a linear modal superposition run with m modes. Generally, the corresponding modal amplitudes would lack the in-plane components. These components are related to the bending stretching coupling which is triggered during nonlinear behavior.

   - A set of n_t equally spaced training modal amplitudes $\mathbf{q}^{(i)}, i \in \{1, \dots, n_t\}$ is chosen from the Linear modal solution.

   - The training modal amplitudes are then substituted into the mapping (3.64) to obtain physical displacements $\mathbf{u}(\mathbf{q}^{(i)})$. This is expected to reproduce in a very broad sense, the coupled in-plane behavior associated to the bending already captured by modes.

   - The obtained set of n_t physical displacements are used as training displacements to obtain the internal force projected onto the LM basis and apply sNNLS for sampling and weighting of elements.

   ECSW applied in this ways would lead to minimal offline cost.

4. **ECSW-IV:** ECSW-III provides a very cheap method to construct training vectors by using information obtained from a linear modal run and enriching it with the quadratic manifold. Using the same training vectors, the basis used for reduction can be reduced in size by techniques of MD selection proposed in Section 3.3.2. With the help of the linear modal solution already performed for training set construction, the MD selection using MMI technique becomes extremely cheap. Instead of the $\frac{m(m+1)}{2}$ MDs originally present in the LM basis, it is proposed to select the m most significant MDs using MMI technique and use them alongside the m vibration modes to construct a basis of size $2m$. The training internal forces are projected onto this reduced basis at the element and the element sampling and weighting is done using sNNLS.

The ECSW-III and ECSW-IV of the above methods are expected to be extremely cheap in terms of offline effort and avoid the calculation of full solution to obtain training sets completely. This cheap evaluation of training vectors using the quadratic manifold is derived from the nature of underlying nonlinearities and is novel. It seems to be applicable in the current context. The limits of application however need to be tested rigorously and would be a topic of further research.

5 Applications and Results

The reduction techniques are tested and compared on examples. Three models are considered, each being a thin-walled structure with varying complexity. All structures are models using triangular shell elements featuring 6 degrees of freedom (DOFs) per node or $N_e = 18$ DOFs per element. Rayleigh damping is used as structural damping in all models [1]. Due to the large size of some of the test cases and for fair comparison between various techniques, all the run are performed on the TU-Delft Linux cluster using a single core and 128 Gigabyte RAM.

The accuracy of the results will be compared to this reference nonlinear solutions in each of the models and the global relative error measure mentioned in [5] shall be used, which is defined as

$$GRE_\star = \frac{\sqrt{\sum\limits_{t\in S}(u_\star(t)-\tilde{u}_\star(t))^T(u_\star(t)-\tilde{u}_\star(t))}}{\sqrt{\sum\limits_{t\in S} u_\star(t)^T(u_\star(t)}} \times 100\% \tag{5.1}$$

where the $\star$ subscript designates the x, y or z the corresponding angles of the global reference frame, $u_\star(t) \in \mathbb{R}^{n_p}$ is the vector of $\star$ displacements at the time t obtained from the reference nonlinear solution, $\tilde{u}_\star(t) \in \mathbb{R}^{n_p}$ is the solution based on the hyper-reduced model, and S is the set of time instants at which error is recorded.

5.1. Flat Structures

A flat plate is simply supported on all sides is considered. The Model (henceforth referred to as Model-I) sketch and parameters are shown in Figure 5.1.

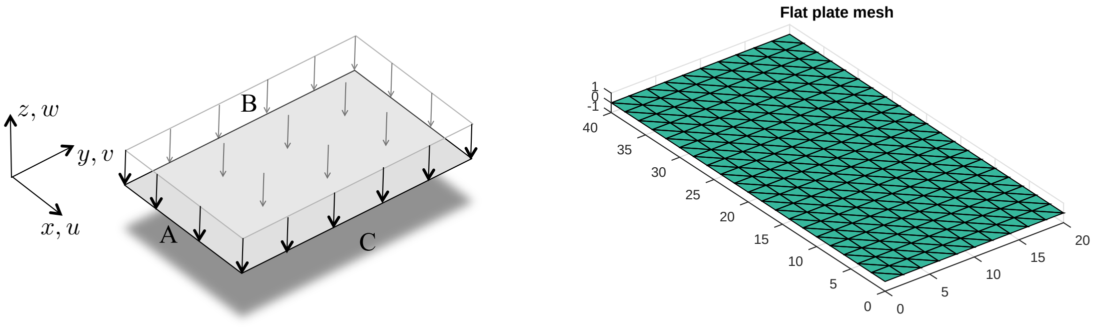
A uniform pressure distribution is chosen normal to the plate surface as the external load. A time varying amplitude (load factor) is used given by

$$\mathbf{g}(t) = p(t)\mathbf{l} \tag{5.2}$$

$$p(t) = p_0[\sin(\omega t) + \sin(\pi\omega t)] \tag{5.3}$$

where $\mathbf{l}$ is a characteristic load vector corresponding to a uniform pressure distribution which is constant and calculated *offline*. Here, $p(t)$ can be termed as the *dynamic load factor* which determines the time-dependency of the external load. For a small model(in terms of number of DOFs) such as Model-I, a wide

[1] A modal damping assumption is used to create a so called *diagonal damping* matrix using weighted sum of Mass and stiffness matrices ($\mathbf{M}$ and $\mathbf{K}$ respectively). A modal damping factor of 0.2% for first two modes is used to determine the weights. (See [2] for details about this implementation). This low value is realistic and is chosen to make sure the VMs of undamped system can be used in reduction



(a) Structure sketch

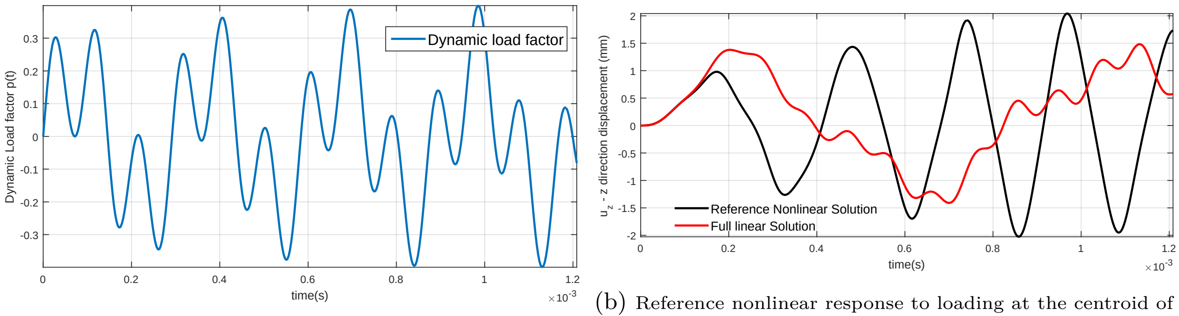
(b) mesh containing 1386 DOFs

Figure 5.1: **Model - I:** Flat plate example. The plate is $L = 40$ mm long, $H = 20$ mm wide, $t = 0.8$ mm thick. The Young Modulus is $E = 70$ GPa, the Poisson's ratio is $\nu = 0.33$, and the density is $\rho = 2700$ Kg/mm^3. All sides are simply supported, i.e. $u = v = w = 0$. A uniform pressure is applied on the plate, according to the time history $p(t) = P[\sin(\omega t) + \sin(\pi\omega t)]$, where $P = 50$ N/mm^2 and $\omega = 2.097 \times 10^4$ rad/s.

variety of loads capturing low-frequency dynamics were tested and results were obtained. Here, results are shown for a choice of $p(t)$ which is a combination of two harmonic signals such that the resulting signal is not (trivially) periodic. ω is a typical loading frequency chosen as the second eigenfrequency of the structure (in equilibrium configuration). The amplitude of loading is kept large enough to trigger nonlinear behaviour. The magnitude of p_0 is chosen such that the displacements are in the order of thickness. This makes VK kinematics applicable and the assumption of linear elasticity of material realistic.

The *full* nonlinear solution was computed by updating the Jacobian of the residual at each N-R iteration (within every time step). No reduction is involved here and thus the linearised system solve of full size is a costly procedure. This also involves the element level calculation and assembly of tangent-stiffness matrix at each N-R iteration which further adds to the *online* cost (at least for large systems). Different levels of mesh refinement were considered to reach an optimum number of degrees of freedom in terms of accuracy. The resulting mesh containing 1386 DOFs and $n_e = 400$ elements, is shown in Figure 5.1b.

The response to the dynamic loading with uniform pressure and load factor given by Equation (5.3) is shown in Figure 5.2. For the shown timespan, the solution is computed over $n_h = 400$ time steps of equal size for all the techniques. The nonlinear solution was compared to the linear system solution with stiffness matrix evaluated at equilibrium (See the comparison in Figure 5.2b). These are clearly different indicating that the given loading is large enough to introduce nonlinear behaviour. The range of the displacements is comparable to the thickness of the structure, thereby making the VK kinematics



(a) dynamic load factor $p(t)$ variation with time

(b) Reference nonlinear response to loading at the centroid of the structure (out of plane displacement (in mm))

Figure 5.2: Dynamic load factor and the corresponding full nonlinear solution for a flat plate simply supported on all sides (model-I).

assumption applicable. This nonlinear solution is used as a reference for comparing the various reduction techniques. All comparison plots are shown for the out of plane displacement at centroid of the structure.

5.1.1. Linear Manifold

Model-I (flat plate) is a rather simplistic example and linear modal superposition using $m = 5$ first VMs is able to accurately reproduce the linear solution but a basis containing these modes is not good enough for capturing the nonlinear response. The Linear Manifold (LM) reduction contains the linear VM basis augmented by all or some of the Modal derivatives (MDs) as explained in Section 3.3. Using 5 VMs, 15 modal derivatives can be obtained which together constitute a basis of 20 vectors. This reduced basis was used for integration and the results (Figure 5.3) are very accurate.

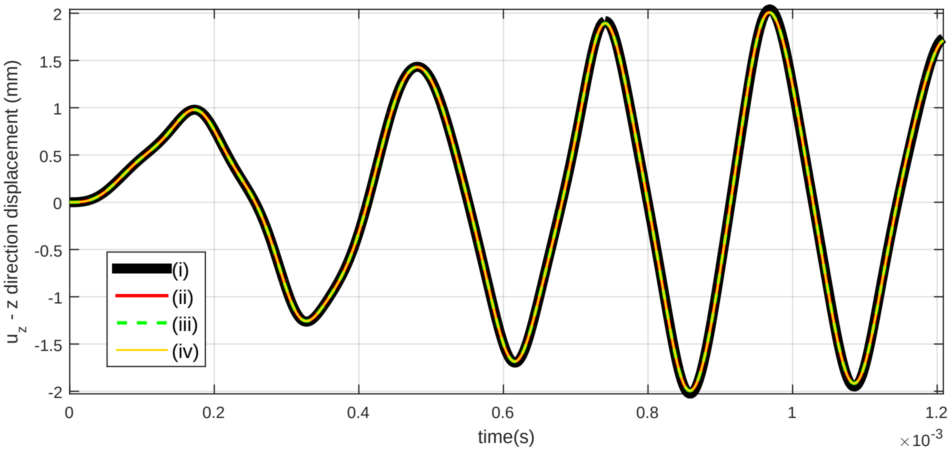


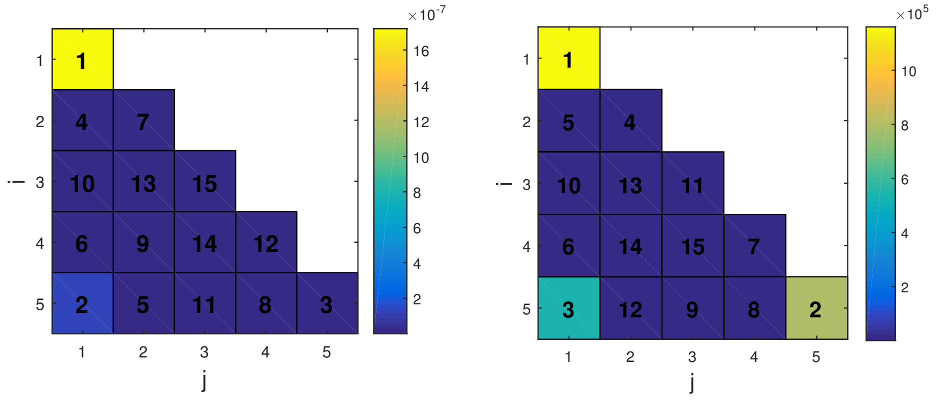
Figure 5.3: The Reference nonlinear solution (i) compared to the linear manifold solution containing 5 VMs and *All* (15) MDs (ii), selected 5 MDs using *MMI* method (iii) and selected 5 MDs using *MVW* method (iv)

.

MD Selection As remarked before that since m VMs correspond to $\frac{m(m+1)}{2}$ MDs, the reduction basis size for linear manifold increases quadratically ($\mathcal{O}(m^2)$) with the number of VMs m in basis. However, only a few of these MDs might be important for capturing the nonlinear behaviour. Section 3.3.2 describes 2 techniques to rank the MDs to be considered in the basis. The ranking and the weights obtained for all MDs using each of the techniques are shown in Figure 5.4. It can be observed that both the techniques (MMI and MVW) result in ranking which are fairly similar (Top 3 ranks contain same group of vectors using both techniques). Although only 3 MDs were sufficient to obtain results with a good accuracy, $n_{MD} = 5$ best ranked MDs have been chosen to highlight the difference in accuracy obtained using the two techniques. Also, as a rule of thumb a total of $n_{MD} = m$ MDs has been chosen in this work for a LM basis containing m VMs, thus making the basis size $2m$. This is done for a fair comparison and keeping the basis size linear with m.

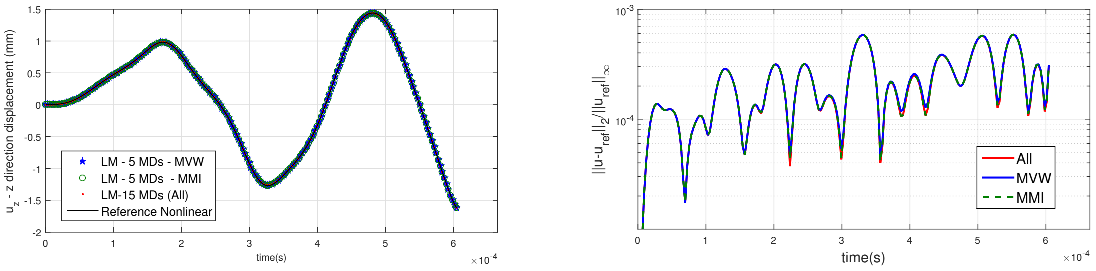
Using a basis of size 8 (5 VMs + 3 selected MDs), both techniques give results with same accuracy. However, when basis is augmented with 2 more MDs, then the accuracy of MVW method is slighted degraded as shown in Figure 5.5b. Global relative errors are listed in a comparison in Table 5.1

Note: The linear manifold can be also implemented using tensor as shown in Section 3.3.3. This leads to offline calculation of tensors and online cost while time integration reduces drastically since the physical space is not touched for calculating Jacobian of the residual during NR iterations. It can be easily shown that the Jacobian obtained in this *tensorial* approach would be exactly the same as that in the regular approach (unlike the Quadratic Manifold case). Thus, the two approaches are identical as far as accuracy and convergence is concerned. The tensorial approach was also implemented to monitor the speed-up of reduction, the plots have not been shown for the clarity reasons. Error statistics and Speed-up observed can be found in Table 5.1.



(a) (b)

Figure 5.4: The ranking and the weights obtained for all MDs using 5 VMs in Model-I. (a) shows the colour plot for weights found using Maximum Modal Interaction (MMI) technique. Cell (i,j) contains weight for MD_{ij}, (b) shows the colour plot for ranking found using Modal Virtual Work (MVW) technique. Cell (i,j) contains weight for MD_{ij}. (Note : part above diagonal not shown due to symmetry)



(a) (b)

Figure 5.5: (a) shows the Linear Manifold reduced solution for Model-I using basis containing 5 VMs and (*i*) 5 selected MDs using MVW, (*ii*) 5 selected MDs using MMI, and (*iii*) All 15 MDs. (b) shows the relative error norm $\left(e(t) = \frac{\|\mathbf{u}-\mathbf{u}_{ref}\|_2}{\max_t \|\mathbf{u}_{ref}(t)\|_2} \right)$ of the 3 methods. Note that when All MDs are used, error is the least (as expected), MMI gives similar error and MVW gives error higher than both of them.

5.1.2. Quadratic Manifold

A quadratic manifold is used with $m = 5$ modes. Thus the size of reduced system is 5 instead of 20 as in the linear Manifold consisting of all MDs. As mentioned in Chapter 3, the reduced time integration with the quadratic manifold can be performed in two ways. One solved using the exact Jacobian of the residual w.r.t. the reduced state in the *tensorial* approach when the physical space is never touched upon, and the other solved using an approximation to the Jacobian where by a reduced tangent stiffness matrix is calculated in the *regular* approach.

It was observed that both the methods give practically the same results for flat structures(e.g. Table 5.1). However, sometimes the *regular* approach took more NR iteration during a time step in order to converge. This is reasonable since the Jacobian is approximated in this approach, which could cause more iterations towards convergence. The results are shown in Figure 5.6

It is remarkable that the higher order tensors are expensive to compute (adding to the *offline* cost) and also lead to costly dot products to obtain reduced internal forces at each NR iteration (See Chapter 6). It is worth investigating the effect of neglecting the higher order tensors on the solution quality. Time integration were performed by neglecting ${}^7_8\mathcal{K}$, ${}^6_7\mathcal{K}$, ${}^5_6\mathcal{K}$, ${}^4_5\mathcal{K}$, ${}^3_4\mathcal{K}$ one by one and the results are shown in Figure 5.7

Figure 5.7 shows that for flat structures, neglecting the higher order tensors (${}^4_5\mathcal{K}$ onwards) has negligible

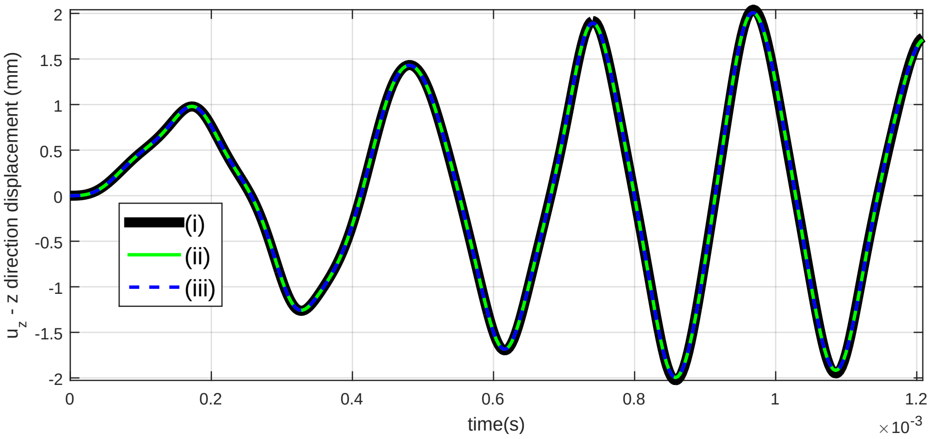


Figure 5.6: The Reference nonlinear solution (i) compared with the Quadratic manifold reduction with size of the reduced system $m = 5$. The two techniques for numerical solution *regular* (ii) and *tensorial* (iii) are compared.

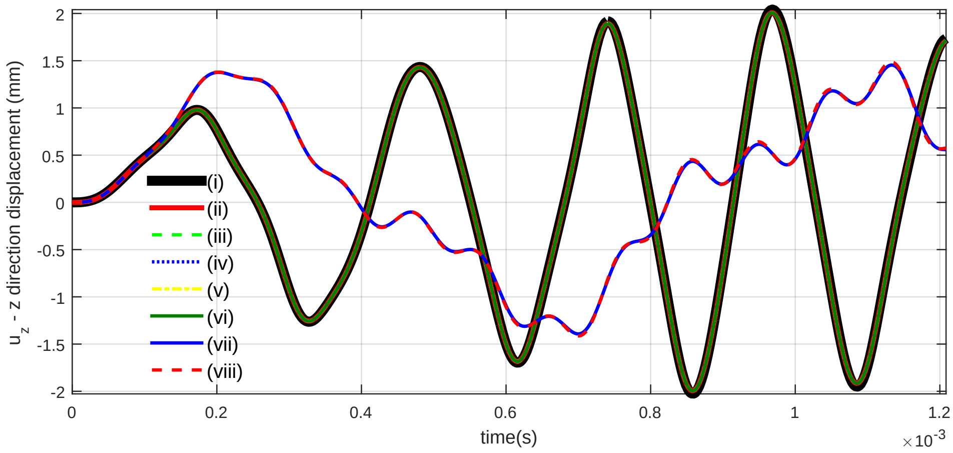


Figure 5.7: shows the Full nonlinear solution (i), QM reduced solution using *tensorial* approach with all tensors (ii), neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ (iii) , neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ onwards (iv), neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards (v), neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards (vi) and finally neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards (vii). Note that (vii) tends towards the linear solution ($viii$) . This implies ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ is essential for capturing nonlinear behaviour and other higher order tensors have negligible effect on the response of (at least) flat structures.

effect on accuracy. It can also be seen that in the case of VMs being out of plane and MDs being in-plane and vice versa (as is the case for flat structures), the tensors ${}^{4}\mathcal{K}$ onwards are zero. These would lead to huge time savings. However, neglecting ${}^{3}\mathcal{K}$ also deteriorates the solution to the point of making it converge towards the linear solution. This can be explained as follows.

It is easy to see that for a flat structure, the quadratic tensor ${}^{2}_{3}\mathbf{K}$ of Equation (3.88) would be identically zero since a flat structure would experience internal forces of equal magnitude for out of plane displacements in both directions. Force being an odd function would require the quadratic parts to be zero due to symmetry. Hence, when one neglects all the tensors ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards in the Equation (3.109), only the linear parts of the internal force are left behind in the reduced equations. Thus, the corresponding run return a solution which would be same as the linear modal superposition solution.

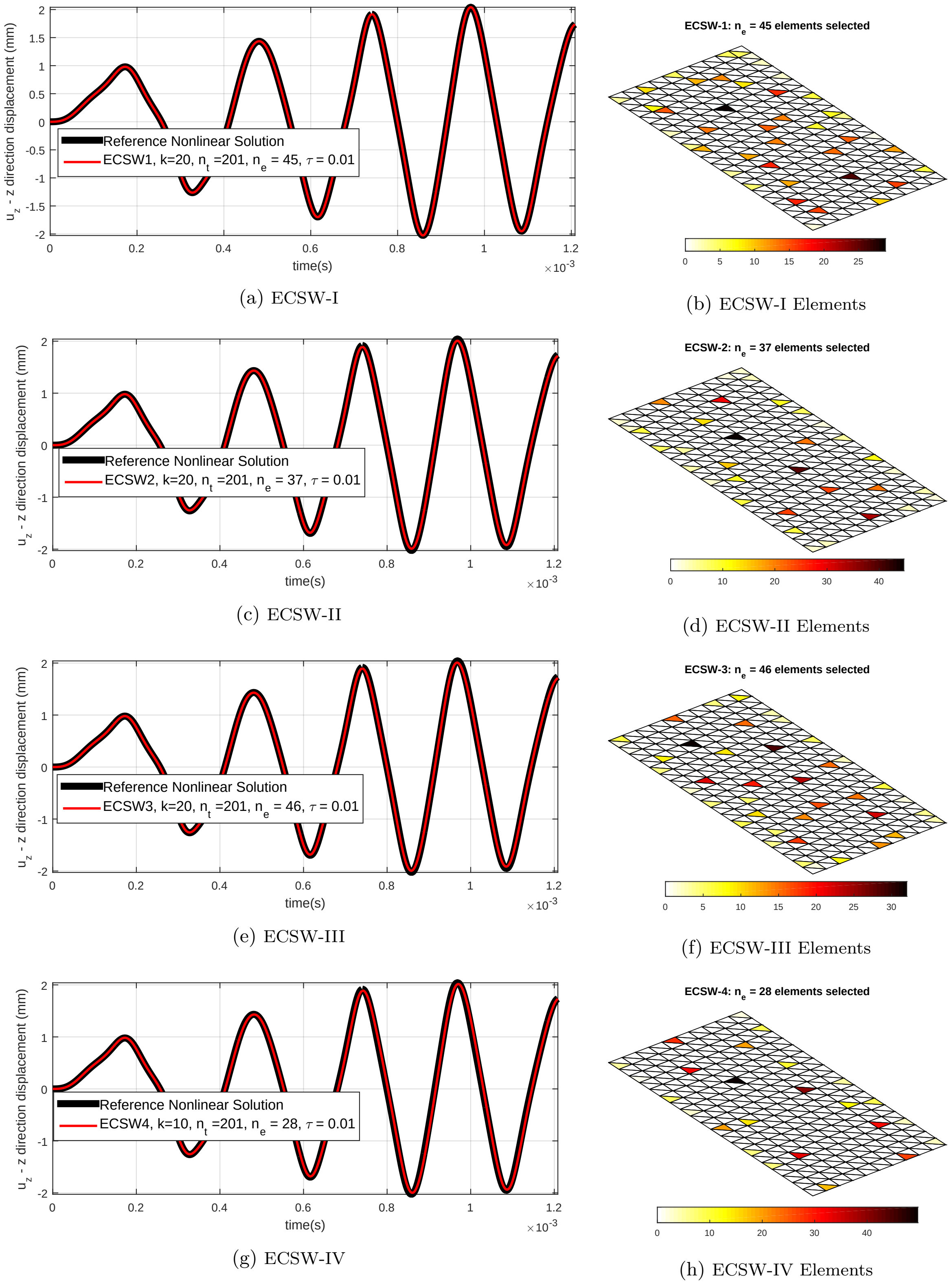
5.1.3. ECSW

ECSW requires training vectors of displacements to solve a sparse non-negative least squares problem whereby the virtual work done by the nonlinear internal forces on the reduction basis is attempted to be reproduced by attaching weights to selected few elements. It is desirable to not choose these training vectors from a full nonlinear/reduced run since they are relatively expensive. Some techniques to cheaply construct training sets for hyper-reduction were proposed in Section 4.3.

Hyper-reduction is performed in the following ways on Model-I keeping number of training vector $n_t = 200$, consistent among all the methods:

1. **ECSW-I -** POD basis obtained by SVD of the snapshots of the reference nonlinear solution. 20 most significant vectors were chosen ($k = 20$). Training vectors chosen from Reference nonlinear solution snapshots at equally spaced time instants.
2. **ECSW-II -** The reduction basis used in linear manifold (composed of VMs and all MDs) containing $k = 20$ vectors. Training vectors were chosen from reduced solution snapshots at 10 equally spaced time instants.
3. **ECSW-III -** The reduction basis used in linear manifold (composed of VMs and all MDs) containing $k = 20$ vectors. Training vectors were chosen by taking the modal amplitude snapshots (equally spaced) from the linear modal superposition solution with $m = 5$ modes and these amplitudes were fed to the quadratic manifold to obtain training vectors for the ECSW.
4. **ECSW-IV -** The reduction basis used in linear manifold (composed of VMs and selected MDs using MMI technique (Section 3.3.2) containing $k = m + n_{MD}$ vectors, where m is the number of VMs used and $n_{MD} = 5$ is the number of MDs selected using appropriate technique. Training vectors were chosen were chosen in the same way as in ECSW-III.

The structure contains 400 elements. Setting a τ value of 0.01 results in the selection of less than 50 elements. Thus, an online speed-up by a factor of at least $400/50 = 8$ can be expected. The results are shown in Figure 5.8 and table 5.1.



(a) ECSW-I

(b) ECSW-I Elements

(c) ECSW-II

(d) ECSW-II Elements

(e) ECSW-III

(f) ECSW-III Elements

(g) ECSW-IV

(h) ECSW-IV Elements

Figure 5.8: Hyper reduction using ECSW for different methods (I-IV), Solution comparison (left) and elements selected during sNNLS along with corresponding weights (right).

5.1.4. Error Statistics

The Global Relative Error (See 5.1) statistics are calculated for all reduction techniques in Model-I. The table also shows an online speed-up defined as follows:

$$S^{\star} = \frac{T_{full}}{T^{\star}}, \tag{5.4}$$

where T_{full} and $T^{\star}$ represent the CPU time taken during the time integration of *full* and reduced solution respectively. The $\star$ superscript denotes the reduction technique being used. Note that the speed-up defined in this manner only takes into account the *online* costs.

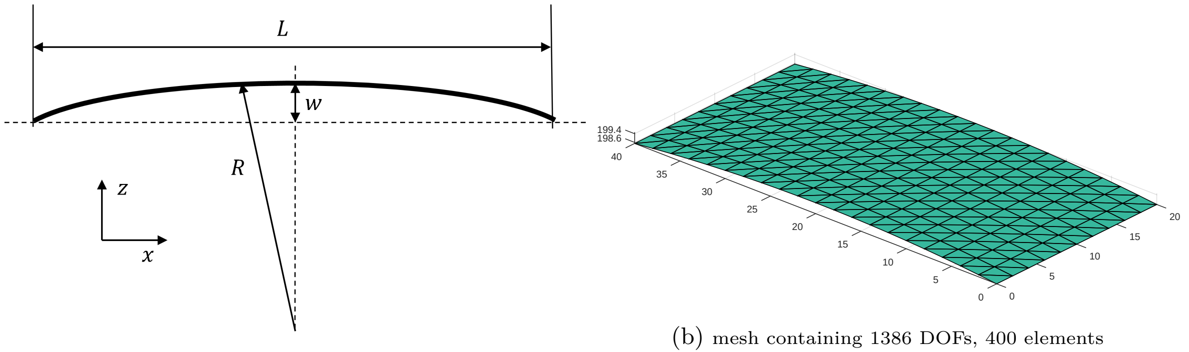
Table 5.1: Global Relative Error in Flat structure model (Model-I) for different reduction techniques. Total time for a full nonlinear run was 89.7 seconds. The speed is calculated w.r.t. this time.

Reduction Technique	GRE_x (%)	GRE_y (%)	GRE_z (%)	Speed-up ($S^{\star}$)
LM Regular (All MDs)	2.43	2.90	0.95	2.16
LM Tensorial (All MDs)	2.43	2.90	0.95	13.01
LM Regular (Selected MDs - MMI)	2.38	2.90	0.95	2.35
LM Tensorial (Selected MDs - MMI)	2.38	2.90	0.95	49.01
LM Regular (Selected MDs - MVW)	2.50	2.89	0.95	2.33
LM Tensorial (Selected MDs - MVW)	2.50	2.89	0.95	50.72
Quadratic Manifold (Regular)	2.34	2.90	0.94	2.08
Quadratic Manifold (Tensorial)	2.37	2.93	0.95	6.94
QMT neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	2.37	2.93	0.95	7.41
QMT neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	2.37	2.93	0.95	8.16
QMT neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards	2.31	2.93	0.92	8.62
QMT neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards	2.31	2.93	0.92	9.15
QMT neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards	>100	>100	>100	9.68
POD	0.33	0.09	0.02	2.25
ECSW - I	1.10	0.63	0.25	16.68
ECSW - II	2.55	3.26	0.99	19.88
ECSW - III	2.65	3.03	1.13	13.59
ECSW - IV	6.51	6.39	4.36	26.35

For the sake of comparison, a POD basis was created containing same number of vectors (20 in case of Model-I) as the LM basis (containing all MDs). This basis was used in hyper-reduction using ECSW as well as in a *regular* manner using Galerkin projection and it is not surprising from Table 5.1 that it returns the least amount of error among all selected techniques. The GRE estimates for Linear Manifold based techniques are very much comparable to the Quadratic manifold based ones. The GRE estimates for ECSW-IV are the highest but are still acceptable. Moreover, it can be seen that even for such a small model, a significantly . All the tensorial approaches show a significant speed-up but they disguise high offline costs as well (See Table 6.3)

5.2. Slightly curved Structure

To test the techniques on a less trivial structure, a slight curvature was added to the flat structure (Figure 5.1a), such that Radius of curvature $R \gg L$ (Figure 5.9). Then the structure becomes a part of a cylinder with its axis parallel to the x axis. Similar loading as flat case was applied with uniform pressure on the surface. Here the pressure force being normal to the surface is not identically aligned with the z direction as in case of the flat structure Model-I. Again different levels of mesh refinement were considered to reach an optimum number of degrees of freedom in terms of accuracy and computation time. The resulting mesh for the model (referred **Model-II** hereafter) containing 1386 DOFs and 400 elements, is shown in Figure 5.1b.

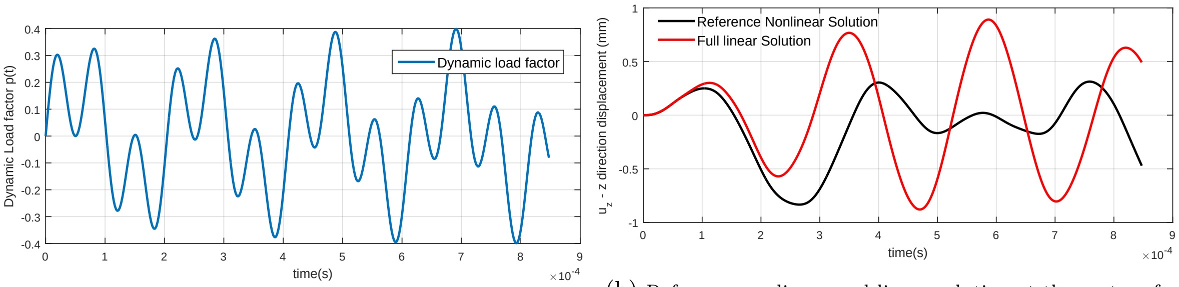


(a) Structure sketch

(b) mesh containing 1386 DOFs, 400 elements

Figure 5.9: **Model - II:** Slightly curved plate example. The plate is $L = 40$ mm long, $H = 20$ mm wide, $t = 0.8$ mm thick and curved with a Radius $R = 200.5$ mm and w = 1 mm. The Young Modulus is $E = 70$ GPa, the Poisson's ratio is $\nu = 0.33$, and the density is $\rho = 2700$ Kg/mm^3. All sides are simply supported, i.e. $u = v = w = 0$. A uniform pressure is applied on the plate, according to the time history $p(t) = P[\sin(\omega t) + \sin(\pi\omega t)]$, where $P = 20$ N/mm^2 and $\omega = 2.965 \times 10^4$ rad/s.

The time signature for loading in the form of the dynamic load factor $p(t)$ is given by Equation (5.3). Again the full nonlinear solution is first obtained by iterative solution of the linearised system and compared with the linear solution. For the shown timespan, the solution is computed over $n_h = 400$ time steps of equal size for all the techniques. The clear difference between linear and nonlinear response (see Figure 5.10b) suggests the system behaviour is nonlinear and the range of displacement being comparable to the structure thickness conforms to the kinematic assumptions of the FE simulation. The structure becomes less stiff in the nonlinear case (when compared to the linear one) because of the bending stretching coupling. The pressure loading pushes the structure inwards producing axial effects and making the bending behaviour softer (notice the first negative peaks of linear and nonlinear solutions in Figure 5.10b).



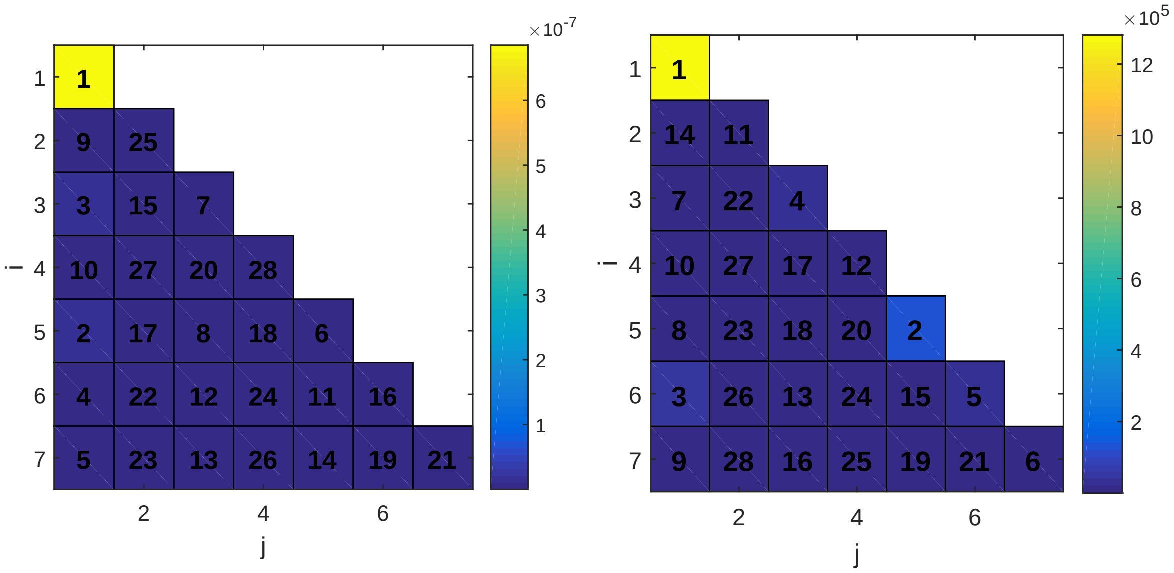
(a) dynamic load factor $p(t)$ variation with time

(b) Reference nonlinear and linear solution at the centre of the structure (out of plane displacement (in mm)). Note the clear difference with the linear solution.

Figure 5.10: Dynamic load factor and the corresponding full nonlinear solution for a rectangular plate with slight curvature simply supported on all sides (Model-II).

5.2.1. Linear Manifold

For the slightly curved plate, the linear manifold was constructed using $m = 7$ first VMs and corresponding MDs. A total of $\frac{m(m+1)}{2} = 28$ MDs can be obtained. If all are included then basis size becomes 35. However, all the MDs are not significant and once again $n_{MD} = m = 7$ MDs are chosen using the MMI and MVW methods (Section 3.3.2). It can be seen in Figure 5.11 that the MD ranking using the two techniques are quite different. Results are shown in Figure 5.12



i	1	2	3	4	5	6	7
1	1						
2	9	25					
3	3	15	7				
4	10	27	20	28			
5	2	17	8	18	6		
6	4	22	12	24	11	16	
7	5	23	13	26	14	19	21

(a)

i	1	2	3	4	5	6	7
1	1						
2	14	11					
3	7	22	4				
4	10	27	17	12			
5	8	23	18	20	2		
6	3	26	13	24	15	5	
7	9	28	16	25	19	21	6

(b)

Figure 5.11: The ranking and the weights obtained for all MDs using 7 VMs in Model-II. (a) shows the colour plot for weights found using Maximum Modal Interaction (MMI) technique,cell (i,j) contains rank(number) and weight(colour) for MD_{ij}. (b) shows the colour plot for ranking found using Modal Virtual Work (MVW) technique, Cell (i,j) contains rank(number) and weight(colour) for MD_{ij}. (Note : part above diagonal not shown due to symmetry)

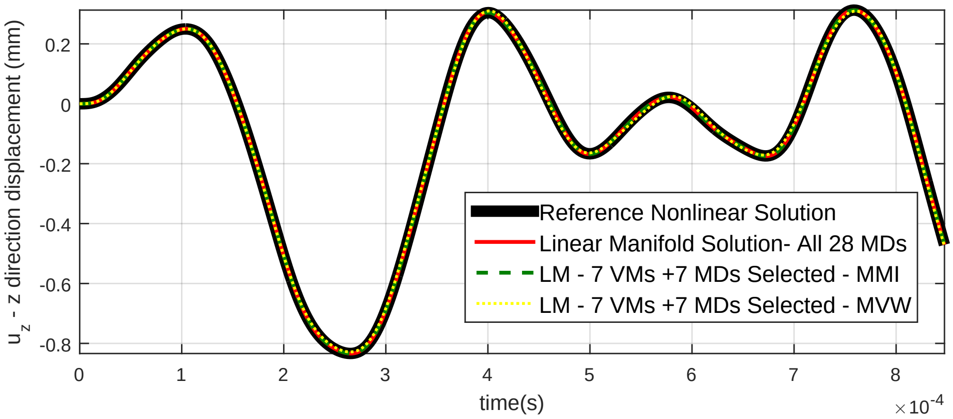


Figure 5.12: The Reference nonlinear solution (*i*) compared to the linear manifold solution containing 7 VMs and *All* (28) MDs (*ii*), selected 7 MDs using *MMI* method (*iii*) and selected 7 MDs using *MVW* method (*iv*)

.

Taking 7 MDs in the selected basis using MMI and MVW techniques gives good results. However, if only 5 are chosen then the error comparison in Figure 5.13 shows that MMI is a very good method for MD selection in this case and the error for MVW increases significantly as time proceeds.

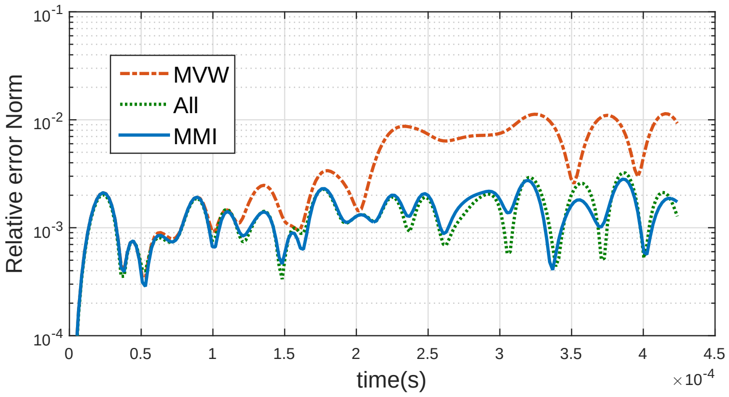


Figure 5.13: The relative error norm $\left(e(t) = \frac{\|\mathbf{u}-\mathbf{u}_{full}\|_2}{\max_t \|\mathbf{u}_{full}(t)\|_2}\right)$ of the 3 methods. Note that when All MDs are used, error is the least (as expected), MMI gives similar error and MVW gives an error higher than both of them.

5.2.2. Quadratic Manifold

A quadratic manifold is used with $m = 7$ modes reducing the number of unknowns to 7 instead of 35 as in case of LM. Results for the *regular* as well as *tensorial* approach are shown in Figure 5.14. It can be seen that the reduced solution using quadratic manifold is not as accurate as that using the linear manifold (Also see Table 5.2).

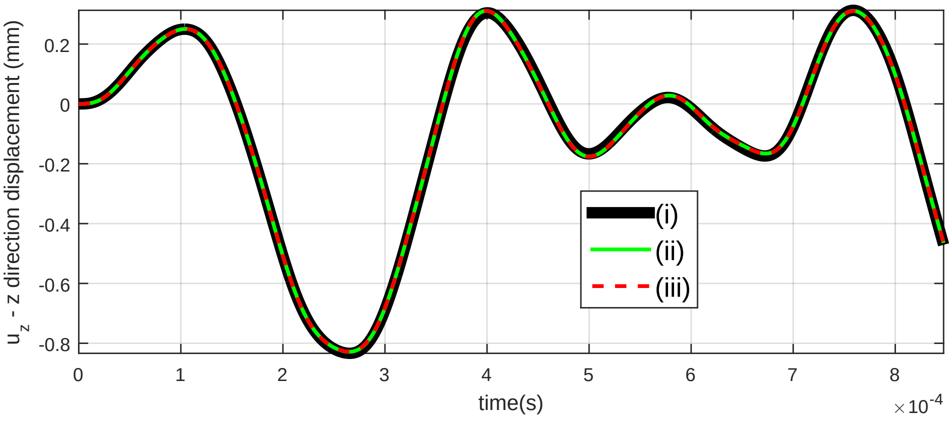


Figure 5.14: The Reference nonlinear solution (i) compared with the Quadratic manifold reduction with size of the reduced system $m = 7$. The two techniques for numerical solution *regular* (ii) and *tensorial* (iii) are compared.

It is interesting again to study the effect of neglecting the higher order tensors. As with the flat structure case, if all the tensors higher than and including ${}^{3}_{4}\mathcal{K}$ are neglected, then the nonlinear response is not captured. However, the behaviour for other tensors is not same. It is observed that the quality of the solution starts degrading as tensors are neglecting from highest to lowest order. It was observed that this effect becomes more pronounced as the curvature of the structure increases. As can be seen in Figure 5.15, the effect of neglecting ${}^{6}_{7}\mathcal{K}$ and ${}^{7}_{8}\mathcal{K}$ for e.g is not as high as that of ${}^{5}_{6}\mathcal{K}$ and ${}^{4}_{5}\mathcal{K}$. Still, this is beneficial since the highest order tensors (which are are most expensive *offline* as well as *online*) are the ones which have least effect on accuracy. Also, upon neglecting the tensors ${}^{3}_{4}\mathcal{K}$ onwards, the solution doesn't tend towards the linear solution as in the flat case of Model-I, and blows up. This is indeed understandable since this structure is not flat and similar reasoning doesn't apply here (See Section 5.1.2).

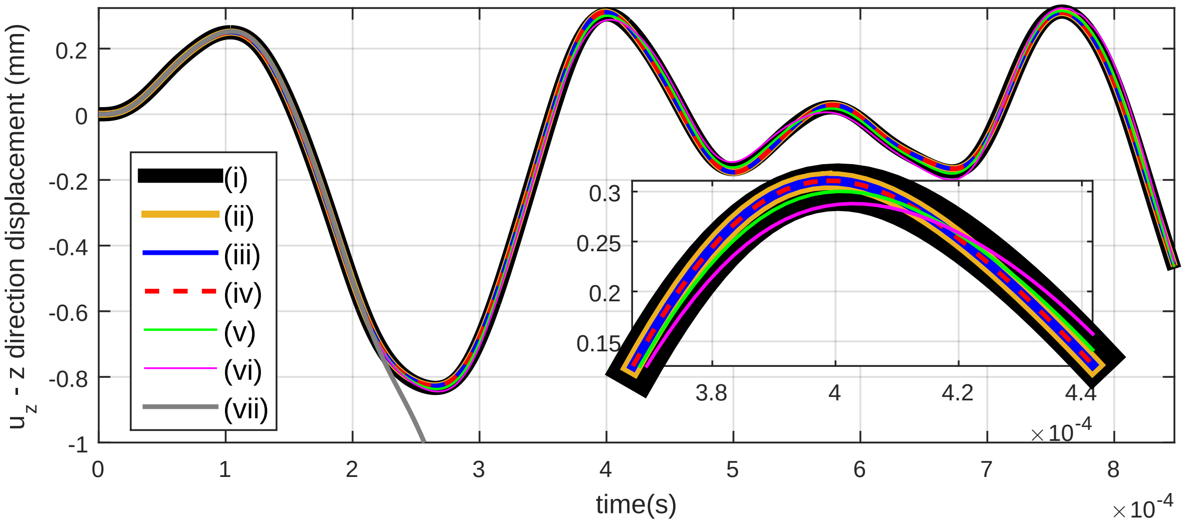


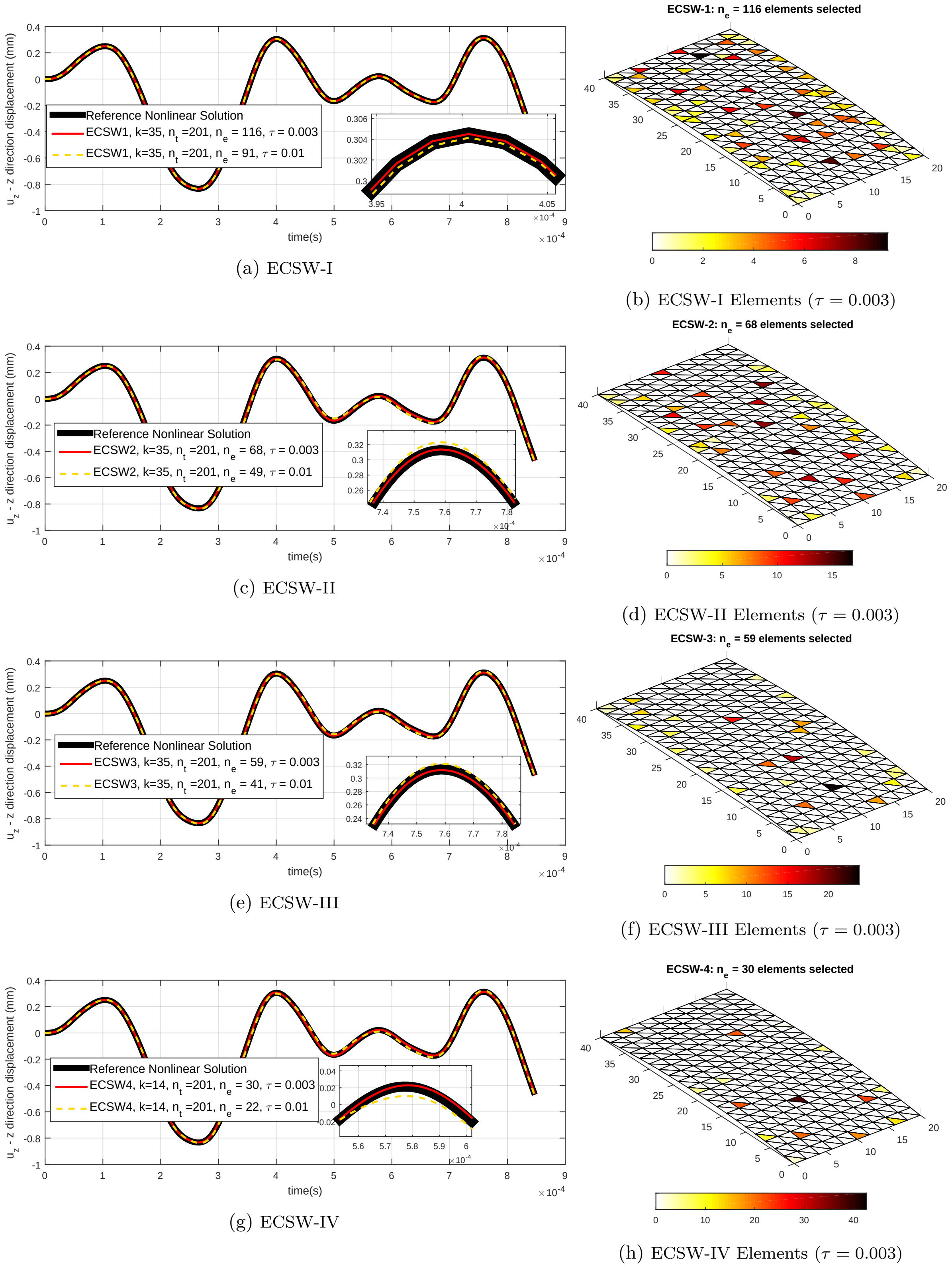
Figure 5.15: shows the Full nonlinear solution (i), QM reduced solution $(m = 7)$ using *tensorial* approach with all tensors (ii), neglecting ${}^{7}_{8}\mathcal{K}$ (iii) , neglecting ${}^{6}_{7}\mathcal{K}$ onwards (iv), neglecting ${}^{5}_{6}\mathcal{K}$ onwards (v), neglecting ${}^{4}_{5}\mathcal{K}$ onwards (vi) and finally neglecting ${}^{3}_{4}\mathcal{K}$ onwards (vii). The effect of neglecting ${}^{6}_{7}\mathcal{K}$ and ${}^{7}_{8}\mathcal{K}$ is negligible compared to that of ${}^{5}_{6}\mathcal{K}$ and ${}^{4}_{5}\mathcal{K}$. Neglecting ${}^{3}_{4}\mathcal{K}$ leads to blow up of solution.

5.2.3. ECSW

As in the case of Model-I, Hyper-reduction of Model-II is done using techniques to cheaply construct training sets proposed in Section 4.3. Hyper-reduction is performed in the following ways on Model-II keeping number of training vectors $n_t = 200$, consistent among all the methods:

1. **ECSW-I -** POD basis obtained by SVD of the snapshots of the reference nonlinear solution. 35 most significant vectors were chosen ($k = 35$). Training vectors chosen from Reference nonlinear solution snapshots at equally spaced time instants.
2. **ECSW-II -** The reduction basis used in linear manifold (composed of VMs and all MDs) containing $k = 35$ vectors. Training vectors were chosen from reduced solution snapshots at 10 equally spaced time instants.
3. **ECSW-III -** The reduction basis same as that in ECSW-II. The modal amplitude snapshots (equally spaced) obtained from the linear modal superposition solution with $m = 7$ modes were fed to the quadratic manifold mapping to obtain training vectors for the ECSW.
4. **ECSW-IV -** The reduction basis used in linear manifold (composed of VMs and selected MDs using MMI technique (Section 3.3.2) containing $k = m + n_{MD}$ vectors, where $m = 7$ is the number of VMs used and $n_{MD} = m = 7$ is the number of MDs selected using appropriate technique. Training vectors were chosen were chosen in the same way as in ECSW-III.

The structure contains 400 elements. Setting a τ value of 0.01 doesn't result in a solution with good accuracy (GRE $> 10\%$ in some cases). Apparently, by setting a lower tolerance of $\tau = 0.003$ on the sNNLS solution leads to much more accurate results (See Table 5.2 also) by addition of only a few elements into the selection set E. The results are portrayed in Figure 5.16. It is interesting to see that the sNNLS algorithm returns a much higher sample of elements for ECSW-I compared to the other three alternatives. It results in a more accurate solution indeed but the others are also of acceptable accuracy.



(a) ECSW-I

(b) ECSW-I Elements ($\tau = 0.003$)

(c) ECSW-II

(d) ECSW-II Elements ($\tau = 0.003$)

(e) ECSW-III

(f) ECSW-III Elements ($\tau = 0.003$)

(g) ECSW-IV

(h) ECSW-IV Elements ($\tau = 0.003$)

Figure 5.16: showing Hyper reduction using ECSW for different methods (I-IV) for two values of the tolerance factor $\tau = 0.01, 0.003$. Solution comparison (left) and elements selected during sNNLS along with corresponding weights (right) using $\tau = 0.003$.

5.2.4. Error Statistics

The results summary for all the reduction techniques performed on Model-II is shown in Table 5.2. For this small model, the speed-ups are not spectacular. It is also interesting to note that the full system size for this model was same as that of Model-I and the speed-ups factors have reduced significantly especially for the Linear manifold and Quadratic manifold based approaches . This is reasonable since the basis size for these techniques has increased due to an increase in m from 5 to 7.

Table 5.2: Global Relative Error in Slightly Curved structure model (Model-II) for different reduction techniques. The solution time for a full nonlinear run was 91.3 seconds. The speed-up factor is calculated w.r.t. this time.

Reduction Technique	GRE_x (%)	GRE_y (%)	GRE_z (%)	Speed-up ($S^\star$)
LM Regular (All MDs)	0.62	0.90	0.24	1.79
LM Tensorial (All MDs)	0.62	0.90	0.24	4.09
LM Regular (Selected MDs - MMI)	1.67	1.14	0.29	1.84
LM Tensorial (Selected MDs - MMI)	1.67	1.14	0.29	9.26
LM Regular (Selected MDs - MVW)	3.69	6.42	1.72	1.89
LM Tensorial (Selected MDs - MVW)	3.69	6.42	1.72	8.37
Quadratic Manifold (Regular)	6.08	4.46	1.70	1.34
Quadratic Manifold (Tensorial)	6.77	4.44	1.81	2.08
QMT neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	6.77	4.43	1.80	3.03
QMT neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	6.77	4.42	1.80	4.24
QMT neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards	7.37	16.62	3.03	4.59
QMT neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards	19.34	28.04	7.47	4.93
QMT neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards	>100	>100	>100	4.37
POD	0.14	0.03	0.01	2.20
ECSW - I	0.26	0.12	0.06	2.61
ECSW - II	5.54	1.12	0.49	3.41
ECSW - III	2.03	3.41	0.71	3.62
ECSW - IV	2.69	2.16	0.86	16.31

A structure which is not as academically simple and has realistically high number of DOFs is required to check the robustness and speed gains of these reduction techniques. Such a structure is developed for testing in the next section.

5.3. Doubly Curved Stiffened Panel

In context of a more realistic application, a thin-walled structure with curvature and stiffeners was considered. The mesh for this structural model (referred to as Model-III hereafter) contains realistically high number of DOFs so that the accuracy and speed up factors of reduction methods can be compared and appreciated. The structure is curved in both directions (can be considered to be a part of a large sphere), with stiffeners present along the length and width. A pressure load is applied locally on the structure at an area shown in Figure 5.17

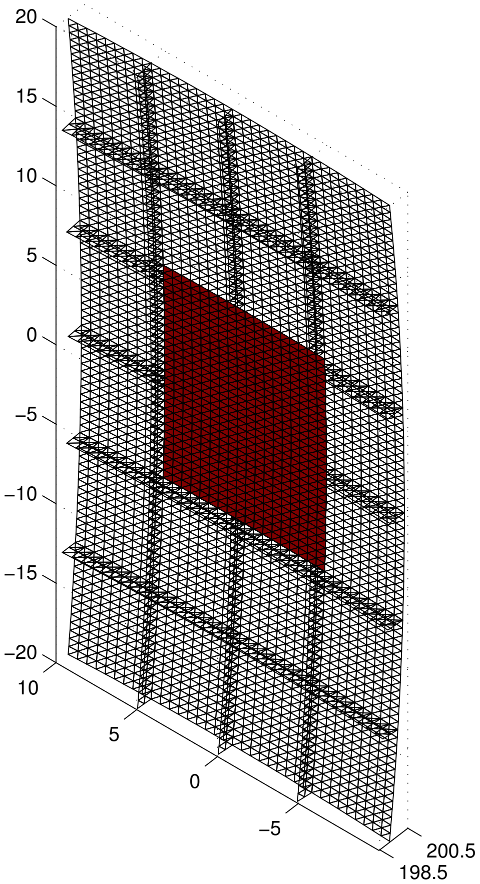


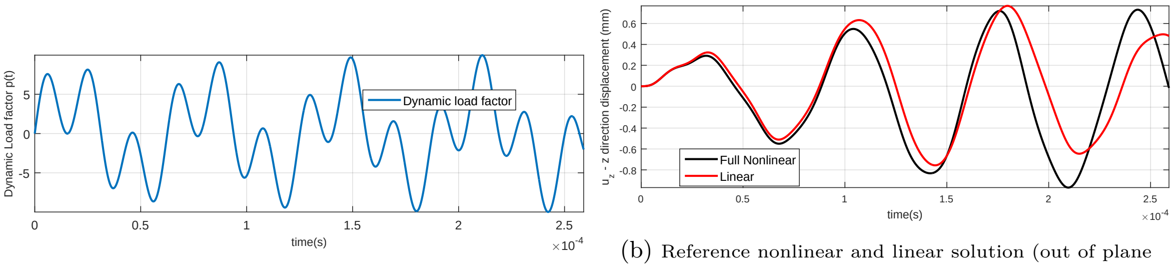
Figure 5.17: **Model - III:** A doubly curved multi-bay stiffened panel structure (length(L) = 40 mm and width (H) = 20 mm) which is a part of a large sphere with radius 200.5 mm and is stiffened with strips of the same material. The highlight area shows the pressure application region.The Young Modulus is $E = 70$ GPa, the Poisson's ratio is $\nu = 0.33$, and the density is $\rho = 2700$ Kg/mm^3. All sides are supported with $u = v = w = 0$ conditions. Pressure is applied on the highlighted area uniformly according to different load cases. Mesh contains $n = 235170$ DOFs, $n_{el} = 77712$ elements.

Different levels of refinement result in a mesh with $n = 235170$ DOFs and $n_e = 77712$ elements such that further refinement doesn't practically change the solution. Due to the small size of the previous models, the speed up observed during reduction seemed redundant (at least when offline costs are also taken into account) but such an application shows the importance of the reduction techniques proposed.

To check robustness of the reduction techniques proposed, three different types of excitation are used. Keeping the spatial component of the load vector same, the time signature is varied according to biharmonic aperiodic excitation with two different amplitudes, and a low frequency pulse. Response to an impulse (in broad spectrum) is not considered here. For reduction of linear systems in realistic load scenarios, only the first few VMs are generally considered and thus the corresponding modal derivatives would be helpful only in capturing slowly varying dynamics for the nonlinear systems in the current context. For all three cases, the responses are graphically shown for a randomly selected node in the middle near the middle-bays with approximate coordinates as $[-1.67,\ 1.34,\ 200.49]$ (in mm).

5.3.1. Load-1a

Here the dynamic load factor is given as in Equation (5.3) with ω chosen as the second natural frequency of vibration ($\approx 9.702 \times 10^4$ rad/s). The applied load and response are shown in 5.18. For the shown timespan, the solution is computed over $n_h = 800$ time steps of equal size for all the techniques. This case shall be referred to as **Model-III L1a** from here onwards. The full nonlinear solution is computed using implicit Newmark time integration such that each full linearized system is solved at each N-R iteration making the time integration extremely expensive. The linear response is close to the nonlinear response in the beginning and becomes significantly different towards the end of time span shown. The system could be called *mildly* nonlinear for this loading.



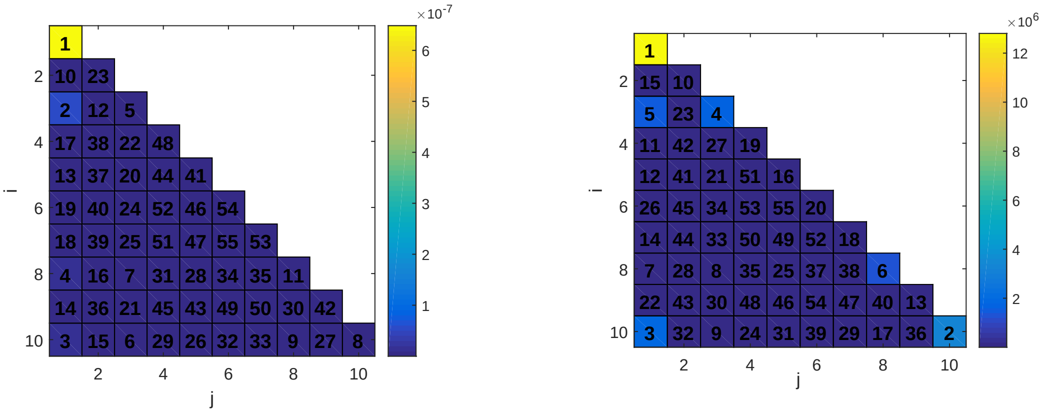
(a) dynamic load factor $p(t)$ variation with time

(b) Reference nonlinear and linear solution (out of plane displacement (in mm)) at a randomly selected location near the middle bays of the structure (coordinates in mm $\approx [-1.67,\ 1.34,\ 200.49,])$. Note the difference in linear solution.

Figure 5.18: Model-III L1a : Dynamic load factor and the corresponding full nonlinear solution for Model-III L1a.

Linear Manifold

A linear manifold is constructed with $m = 10$ first VMs and the corresponding MDs. A total of $\frac{m(m+1)}{2} = 55$ MDs can be obtained from 10 VMs, which would make the ROB size 65 if all the MDs are considered. However, as discussed in 3.3.2, the MD selection techniques (MMI & MVW) are also tested to include only $n_{MD} = m = 10$ MDs in the basis instead of all. The results are shown in Figure 5.20. It can be seen that all the 3 methods are able to reproduce the nonlinear response with very good accuracy. Unlike the previous models, the MVW technique produces results with almost the same error estimates as the MMI technique (See Table 5.3). Also it can be seen from Figure 5.19 that though the ranking generated for the MDs using the two techniques are different, 9 out of the selected top 10 MDs are common between them.



1									
10	23								
2	12	5							
17	38	22	48						
13	37	20	44	41					
19	40	24	52	46	54				
18	39	25	51	47	55	53			
4	16	7	31	28	34	35	11		
14	36	21	45	43	49	50	30	42	
3	15	6	29	26	32	33	9	27	8

(a)

1									
15	10								
5	23	4							
11	42	27	19						
12	41	21	51	16					
26	45	34	53	55	20				
14	44	33	50	49	52	18			
7	28	8	35	25	37	38	6		
22	43	30	48	46	54	47	40	13	
3	32	9	24	31	39	29	17	36	2

(b)

Figure 5.19: The ranking and the weights obtained for all (55) MDs using 10 VMs in Model-II. (a) shows the colour plot for weights found using Maximum Modal Interaction (MMI) technique,cell (i,j) contains rank(number) and weight(colour) for MD_{ij}. (b) shows the colour plot for ranking found using Modal Virtual Work (MVW) technique, Cell (i,j) contains rank(number) and weight(colour) for MD_{ij}. (Note : part above diagonal not shown due to symmetry)

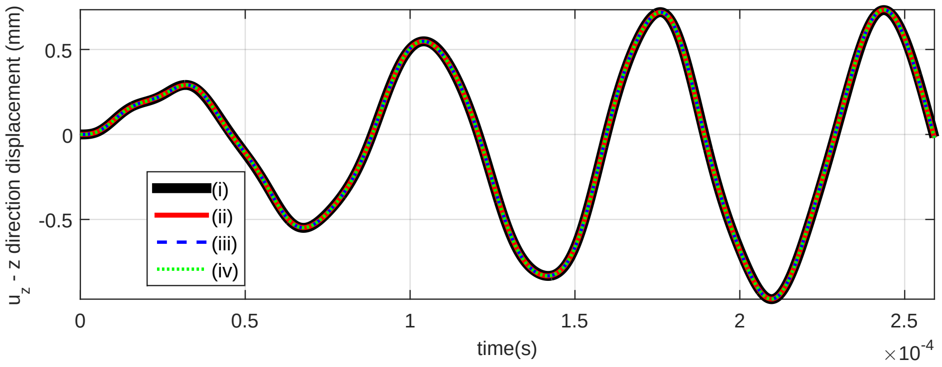


Figure 5.20: The Reference nonlinear solution (*i*) compared to the linear manifold solution containing 10 VMs and *All* (55) MDs (*ii*), selected 10 MDs using *MMI* method (*iii*) and selected 10 MDs using *MVW* method (*iv*)

.

Quadratic Manifold

A quadratic manifold is used with $m = 10$ first VMs, reducing the number of unknowns to 10 instead of 65 as in case of LM with all corresponding MDs. Results for the *regular* as well as *tensorial* approach are shown in Figure 5.21. It can be seen that the reduced solution using quadratic manifold is not as accurate as that using the linear manifold (Also see Table 5.3).

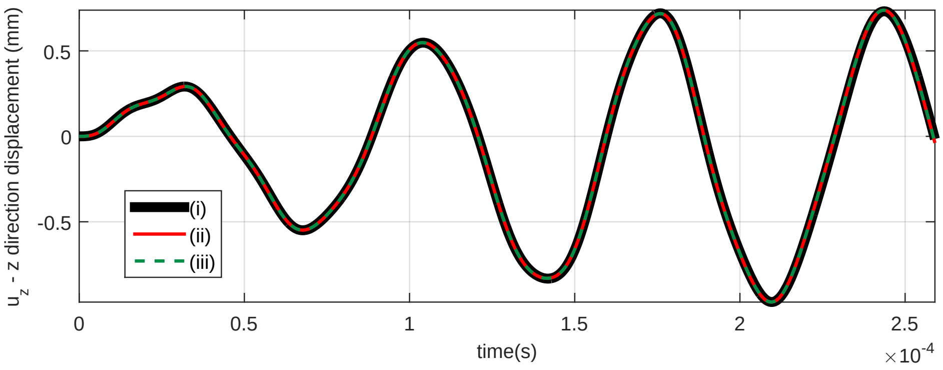


Figure 5.21: The Reference nonlinear solution (*i*) compared with the Quadratic manifold reduction with size of the reduced system $m = 7$. The two techniques for numerical solution *regular* (*ii*) and *tensorial* (*iii*) are compared.

It is interesting again to study the effect of neglecting the higher order tensors. As with the previous cases, if all the tensors higher than and including ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ are neglected, then the nonlinear response is not captured. It can be seen (in Figure 5.22 and Table 5.3) that the quality of the solution starts degrading as tensors are neglecting from highest to lowest order. The effect of neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$, ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ for e.g is not as high as that of neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$. Again, this is good since the higher order tensors are the ones which are most expensive to calculate and if these are not calculated, then there can be huge savings in offline cost. The online costs for calculation of projected residual and Jacobian of reduced equations also involved many dot products(contractions) which can be avoided to a large extent if higher order tensors are neglected.

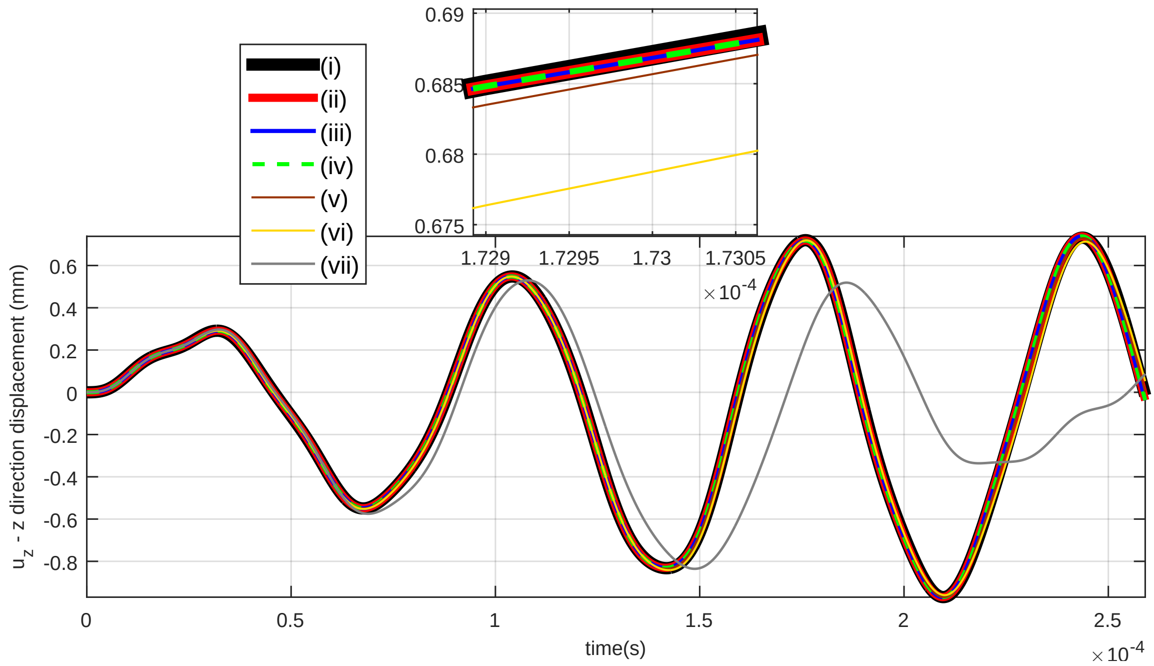


Figure 5.22: shows the Full nonlinear solution for Model-III-L1a (i), QM reduced solution $(m = 10)$ using *tensorial* approach with all tensors (ii), neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ (iii) , neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ onwards (iv), neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards (v), neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards (vi) and finally neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards (vii). The effect of neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ is negligible compared to that of ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{4}_{5}\boldsymbol{\mathcal{K}}$. Neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ leads to an irrelevant solution.

ECSW

As in the case of Model-I and II, Hyper-reduction of Model-III is done using techniques to cheaply construct training sets proposed in Section 4.3. Hyper-reduction is performed in the following ways on Model-III keeping number of training vectors $n_t = 100$ chosen from equally space snapshots, consistent among all the methods:

1. **ECSW-I -** POD basis obtained by SVD of the snapshots of the reference nonlinear solution. 65 most significant vectors were chosen ($k = 65$). Training vectors taken from full nonlinear solution.
2. **ECSW-II -** The reduction basis used in linear manifold (composed of VMs and all MDs) containing $k = 65$ vectors. Training vectors taken from LM reduced solution snapshots.
3. **ECSW-III -** The reduction basis same as ECSW-II. The modal amplitude snapshots (equally spaced) obtained from the linear modal superposition solution with $m = 10$ modes were fed to the quadratic manifold mapping to obtain training vectors for the ECSW.
4. **ECSW-IV -** The reduction basis used in linear manifold (composed of VMs and selected MDs using MMI technique (Section 3.3.2) containing $k = m + n_{MD}$ vectors, where $m = 10$ is the number of VMs used and $n_{MD} = m = 10$ is the number of MDs selected using appropriate technique. Training vectors same as that in ECSW-III.

As can be seen in Figure 5.23, the sNNLS returns an element sample set of a very small size (less than 500 elements). The complete structure contains 77712 elements and this would imply that nonlinearity is evaluated in less than 1% of the structure. It is interesting to see that the hyper-reduction techniques with minimal offline cost (i.e. ECSW-III and IV) generate a smaller element sample and produce results with similar accuracy. These properties make the proposed hyper-reduction techniques highly favourable and result in spectacular speed-ups. (See Table 5.3).

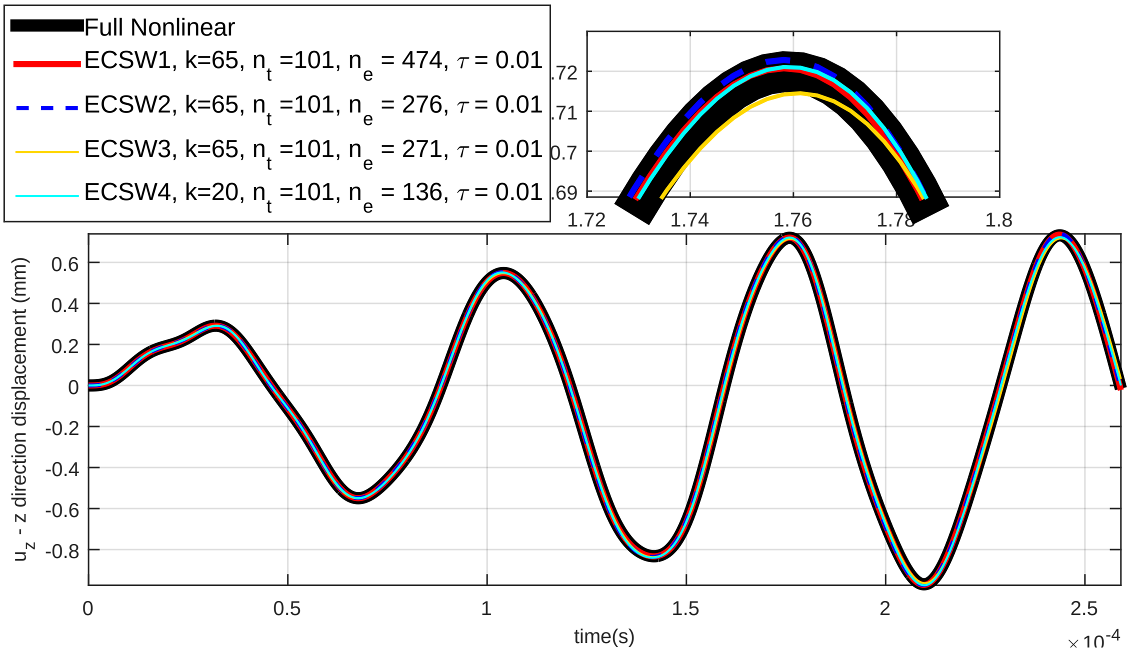


Figure 5.23: showing Hyper reduction using ECSW for different methods (I-IV). The full system contain 77712 elements, where sNNLS returns less than 500 elements to be sampled and weighted for evaluation and projection of nonlinearity. This gives huge speed-ups.

Error Statistics

The GRE (Equation (5.1)) and Speed-up (Equation (5.4)) statistics for all the techniques tried on Model-III L1a are shown in Table 5.3. The following observations can be made for Model-I-L1a:

- Projection based techniques implemented in *regular* manner return Speed-ups greater than 1 but still less than 2 even on a realistically large model. This confirms the need for tensor based techniques and hyper-reduction.

- The error computed using all th techniques seem acceptable (especially considering the fact that most of the techniques result in a phase error and are able to qualitatively follow the reference solution) except for the 'QM Tensorial neglecting ${}^3_4\mathcal{K}$ onwards'. As, also mentioned before, this is expected since it leads to ignoring the cubic part of the nonlinearities (Equation (3.88)) completely.

- The speed-ups obtained in the tensorial techniques seems spectacular (≈ 5000 for LM Tensorial with selected MDs!). This is expected since the reduced order model is independent of *full* systems size and even higher speed-ups can be expected for larger system provided m is kept the same. However, it should be realised that these high speed-ups disguise significant offline costs (further discussion in Chapter 6).

- The speed-ups for quadratic manifold based tensor approaches is high but still lower than that for LM based ones, even after higher order tensors are neglected. This is contrary to the expected behaviour and is studied further in Chapter 6.

- The hyper-reduction techniques based on ECSW result in high speed-ups. It can be seen that ECSW-I distinguishes itself from the others (II, III & IV) in terms of error but hides expensive full solution run in terms of offline cost. In this context, ECSW-II and III prove useful since the error estimated seem acceptable, they result in much higher speed-ups and carry minimal offline costs(Table 6.3).

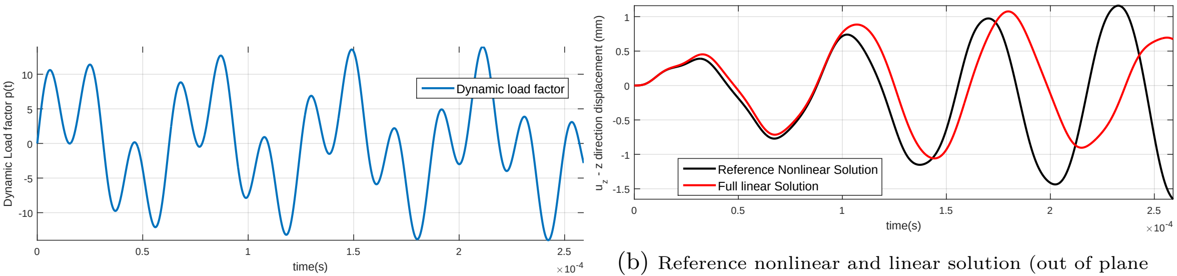
- It is interesting that ECSW-IV reaches much higher speed-ups than ECSW-III with very similar error margins. One of the reasons for higher speed is the smaller size of reduction basis. But a second and more interesting reason is that it has a much smaller elements sample for evaluation of nonlinearity than ECSW-III, even though both have the same training vectors. It is worth investigating if this is a pattern more general than this example.

Table 5.3: Global Relative Error in Model-III-L1a for different reduction techniques.The solution time for a full nonlinear run was ≈ 3.767×10^4 seconds. The speed-up factor is calculated w.r.t. this time.

Reduction Technique	GRE_x (%)	GRE_y (%)	GRE_z (%)	Speed-up ($S^\star$)
LM Regular (All MDs)	1.66	0.63	2.34	1.44
LM Tensorial (All MDs)	1.66	0.63	2.34	601.90
LM Regular (Selected MDs - MMI)	1.91	0.72	2.57	2.09
LM Tensorial (Selected MDs - MMI)	1.91	0.72	2.57	4966.00
LM Regular (Selected MDs - MVW)	1.89	0.71	2.58	2.11
LM Tensorial (Selected MDs - MVW)	1.89	0.71	2.58	5039.00
Quadratic Manifold (Regular)	3.07	1.53	3.91	1.39
Quadratic Manifold (Tensorial)	3.05	1.51	3.92	99.46
QMT neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	3.05	1.51	3.92	169.46
QMT neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	3.04	1.48	3.92	251.64
QMT neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards	2.90	1.21	4.54	254.70
QMT neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards	5.48	4.70	8.29	265.66
QMT neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards	76.32	76.13	74.10	268.50
POD	0.01	0.00	0.02	1.77
ECSW - I	0.55	0.51	0.82	180.20
ECSW - II	4.00	1.49	6.63	262.40
ECSW - III	4.99	2.58	8.01	261.90
ECSW - IV	4.01	2.56	8.51	923.80

5.3.2. Load-1b

This case varies from Model-III L1a only in terms of amplitude of loading. The loading is 40% higher than that in previous case (Model-II-L1a). As can be seen from the load response in Figure 5.29, the nonlinear response is significantly different from the linear ones (at least when compared to Figure 5.29). For the shown timespan, the solution is computed over $n_h = 800$ time steps of equal size for all the techniques. This case is referred to as **Model-III L1b** hereafter.



(a) dynamic load factor $p(t)$ variation with time

(b) Reference nonlinear and linear solution (out of plane displacement (in mm)) at a randomly selected location near the middle bays of the structure (coordinates in mm $\approx [-1.67,\ 1.34,\ 200.49,])$. Note the difference in linear solution.

Figure 5.24: Model-III L1b : Dynamic load factor and the corresponding full nonlinear solution for Model-III L1b.

Linear Manifold

A linear manifold is constructed with 10 VMs and corresponding MDs. A total of 55 MDs can be obtained from 10 VMs which makes the ROB size 65 if all the MDs are considered. However, as discussed in 3.3.2, the MD selection techniques (MMI & MVW) are also tested. The results are shown in Figure 5.25. It can be seen that all the 3 methods are able to reproduce the nonlinear response with very good accuracy.

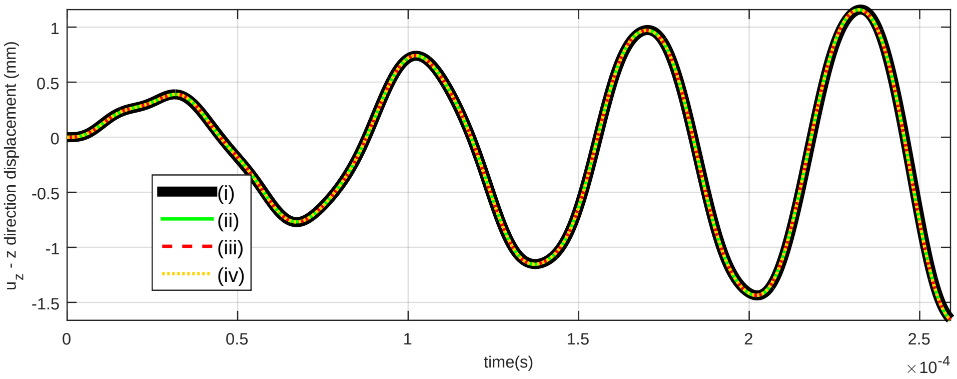


Figure 5.25: The Reference nonlinear solution (i) compared to the linear manifold solution containing 10 VMs and *All* (55) MDs (ii), selected 10 MDs using *MMI* method (iii) and selected 10 MDs using *MVW* method (iv)

.

Quadratic Manifold

A quadratic manifold is used with $m = 10$ modes reducing the number of unknowns to 10 instead of 65 as in case of LM. Results for the *regular* as well as *tensorial* approach are shown in Figure 5.26. It can be again seen that the reduced solution using quadratic manifold is not as accurate as that using the linear manifold (Also see Table 5.4).

It is interesting again to study the effect of neglecting the higher order tensors. As with the previous cases, if all the tensors higher than and including ${}^3_4\mathcal{K}$ are neglected, then the nonlinear response is not captured. It can be seen (in Figure 5.27 and Table 5.4) that the quality of the solution starts degrading

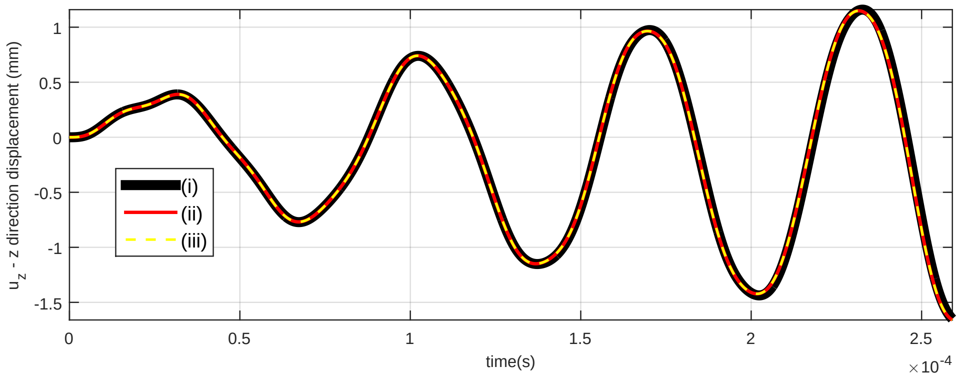


Figure 5.26: The Reference nonlinear solution (i) compared with the Quadratic manifold reduction with size of the reduced system $m = 7$. The two techniques for numerical solution *regular* (ii) and *tensorial* (iii) are compared.

as tensors are neglecting from highest to lowest order. The effect of neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ for e.g is not as high as that of ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{4}_{5}\boldsymbol{\mathcal{K}}$. This is good since the higher order tensors are the ones which are most expensive to calculate and if higher order tensors are not calculated, then there can be huge savings in offline as well as online operations. But compared to Model-III L1a, the results seem to be significantly deteriorated on neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{4}_{5}\boldsymbol{\mathcal{K}}$. This could be linked to the fact that this model shows a more nonlinear behaviour when compared to Model-III L1a.

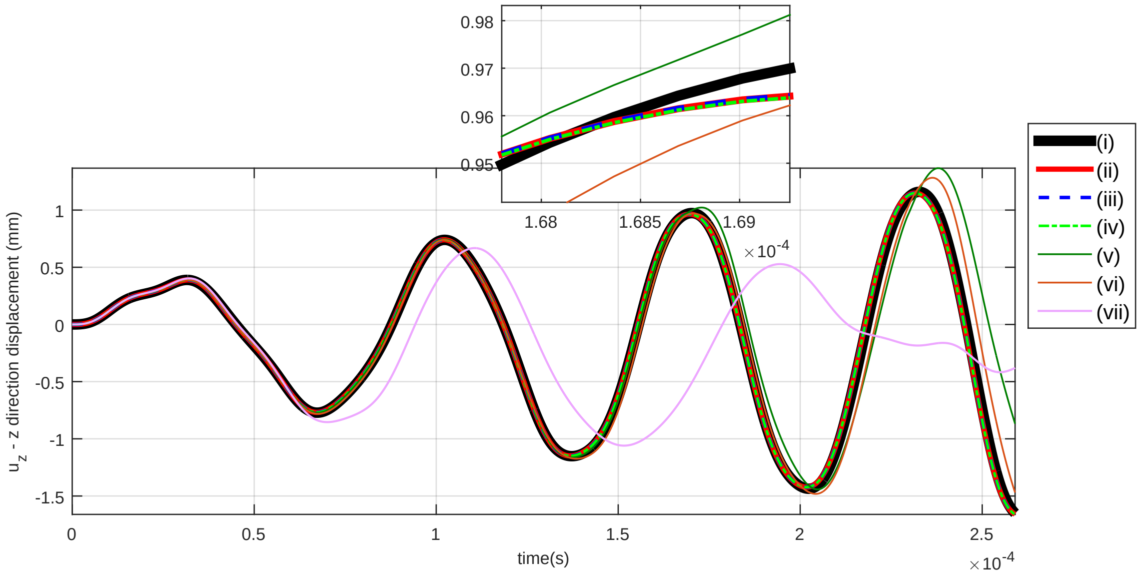


Figure 5.27: shows the Full nonlinear solution (i), QM reduced solution $(m = 10)$ using *tensorial* approach with all tensors (ii), neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ (iii) , neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ onwards (iv), neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards (v), neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards (vi) and finally neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards (vii). The effect of neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ is negligible compared to that of ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{4}_{5}\boldsymbol{\mathcal{K}}$.

ECSW

As in case of Model-III L1a, hyper-reduction on Model-III L1b is performed in the following ways keeping number of training vectors $n_t = 200$, consistent among all the methods:

1. **ECSW-I -** POD basis obtained by SVD of the snapshots of the reference nonlinear solution. 65 most significant vectors were chosen ($k = 65$). Training vectors taken from full nonlinear solution.
2. **ECSW-II -** The reduction basis used in linear manifold (composed of VMs and all MDs) containing $k = 65$ vectors. Training vectors taken from LM reduced solution snapshots.

3. **ECSW-III -** The reduction basis same as ECSW-II. The modal amplitude snapshots (equally spaced) obtained from the linear modal superposition solution with $m = 10$ modes were fed to the quadratic manifold mapping to obtain training vectors for the ECSW.

4. **ECSW-IV -** The reduction basis used in linear manifold (composed of VMs and selected MDs using MMI technique (Section 3.3.2) containing $k = m + n_{MD}$ vectors, where $m = 10$ is the number of VMs used and $n_{MD} = m = 10$ is the number of MDs selected using appropriate technique. Training vectors same as that in ECSW-III.

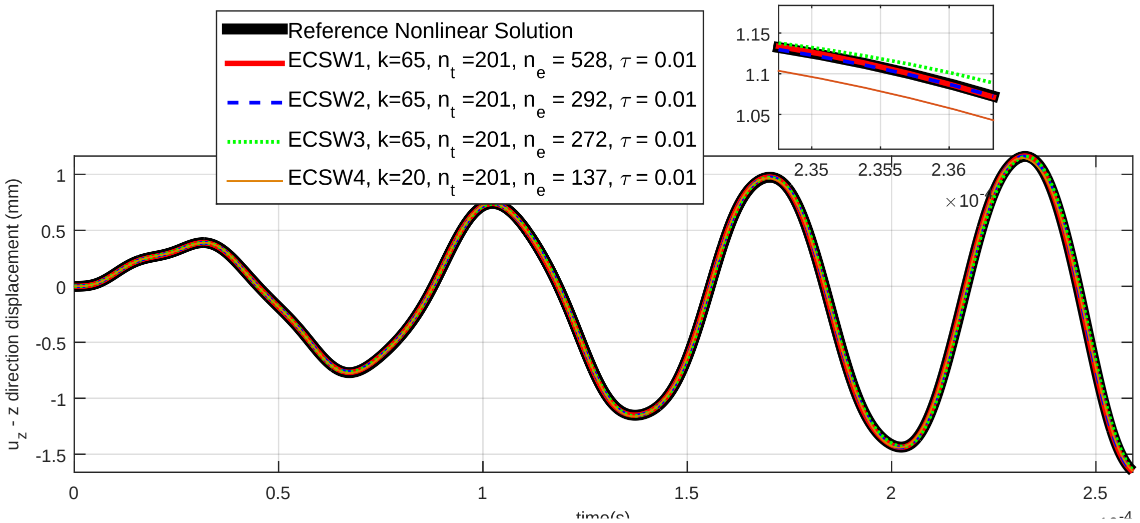


Figure 5.28: showing Hyper reduction using ECSW for different methods (I-IV). The full system contain 77712 elements, where sNNLS returns a sample of ≈ 500 or less elements to be weighted for evaluation and projection of nonlinearity. This gives huge speed up.

This model shows similar sampling of elements as in case of Model-III L1a, with nonlinearity being evaluated on less than 1 % of the mesh leading to high speed-ups as shown in Table 5.4.

Error Statistics

The GRE (Equation (5.1)) and Speed-up (Equation (5.4)) statistics for all the techniques tried on Model-III L1b are shown in Table 5.3. Apart from the general observations made for error statistics of Model-III L1a in Section 5.3.1, a few more observations can be made:

- As the nonlinearity in the solution has increased, the Linear Manifold based techniques perform better than QM based techniques in terms of GRE. This might be because the nonlinearities cannot be as accurately captured by the second order effects described in nonlinear mapping Equation (3.64). It is also known that LM in general provides a lower bound for error committed during QM based techniques (Section 3.4.3).
- The solution deteriorates much more significantly upon neglecting the tensors ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ and ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ as compared to that in Model-III L1a. Unacceptable errors are produced.
- It is intriguing that the proposed hyper-reduction techniques show the same interesting trends of producing smaller element sampling (resulting in speed-ups without compromising on error) as in case of Model-III L1a.

Table 5.4: Global Relative Error in Model-III-L1b for different reduction techniques.The solution time for a full nonlinear run was ≈ 3.895×10^4 seconds. The speed-up factor is calculated w.r.t. this time.

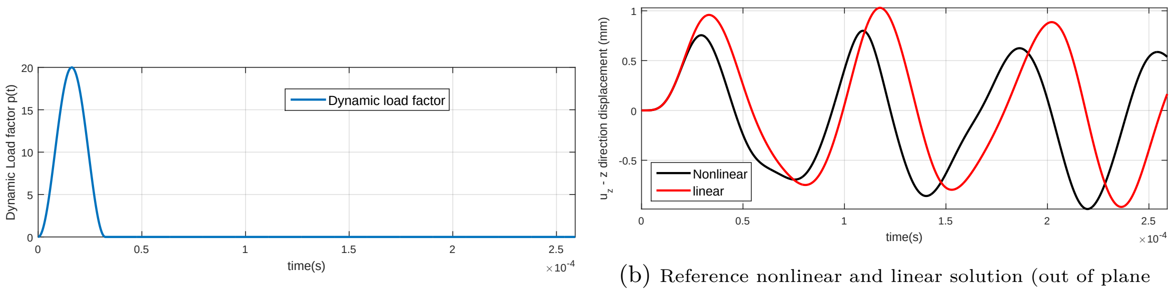
Reduction Technique	GRE_x (%)	GRE_y (%)	GRE_z (%)	Speed-up ($S^\star$)
LM Regular (All MDs)	2.67	1.35	3.49	1.81
LM Tensorial (All MDs)	2.67	1.35	3.49	1145.00
LM Regular (Selected MDs - MMI)	2.93	1.51	3.58	2.22
LM Tensorial (Selected MDs - MMI)	2.93	1.51	3.58	6931.00
LM Regular (Selected MDs - MVW)	2.90	1.49	3.54	2.27
LM Tensorial (Selected MDs - MVW)	2.90	1.49	3.54	4472.00
Quadratic Manifold (Regular)	7.59	4.52	8.27	1.69
Quadratic Manifold (Tensorial)	7.58	4.50	8.27	159.00
QMT neglecting ${}^{7}_{8}\mathcal{K}$	7.57	4.49	8.27	211.00
QMT neglecting ${}^{6}_{7}\mathcal{K}$ and ${}^{7}_{8}\mathcal{K}$	7.52	4.44	8.22	239.40
QMT neglecting ${}^{5}_{6}\mathcal{K}$ onwards	41.92	39.08	74.39	239.69
QMT neglecting ${}^{4}_{5}\mathcal{K}$ onwards	24.91	22.49	43.91	243.59
QMT neglecting ${}^{3}_{4}\mathcal{K}$ onwards	>100	>100	>100	304.10
POD	0.01	0.00	0.05	1.28
ECSW - I	1.15	1.04	1.69	105.90
ECSW - II	4.56	1.38	6.55	283.30
ECSW - III	7.18	4.71	10.44	305.90
ECSW - IV	4.71	3.65	12.64	800.80

5.3.3. Load-2

A more simplistic but an important kind of load is a pulse load. The pulse load is applied as a pressure uniformly on the area highlighted in Figure 5.17. The dynamic load factor is given as

$$p(t) = A\sin^2(\omega t)\left[H(t) - H\left(\frac{\pi}{\omega} - t\right)\right], \tag{5.5}$$

where $H(t)$ is the heaviside function and ω chosen as the second natural frequency of vibration ($\approx 9.702 \times 10^4$ rad/s). The load response is shown in Figure 5.24 and it can be seen that the nonlinear response is significantly different from the linear counterpart. For the shown timespan, the solution is computed over $n_h = 800$ time steps of equal size for all the techniques.



(a) dynamic load factor $p(t)$ variation with time

(b) Reference nonlinear and linear solution (out of plane displacement (in mm)) at a randomly selected location near the middle bays of the structure (coordinates in mm $\approx [-1.67,\ 1.34,\ 200.49,])$. Note the difference in linear solution.

Figure 5.29: Model-III L2 : Dynamic load factor for pulse loading and the corresponding full nonlinear solution for a flat plate simply supported on all sides.

Linear Manifold

A linear manifold is constructed with 10 VMs and corresponding MDs. A total of 55 MDs can be obtained from 10 VMs which makes the ROB size 65 if all the MDs are considered. However, as discussed in 3.3.2, the MD selection techniques (MMI & MVW) are also tested. The results are shown in Figure 5.30. It can be seen that all the 3 methods are able to reproduce the nonlinear response with very good accuracy. MD selection results in 9 out top 10 MDs common between the two techniques (Figure 5.19).

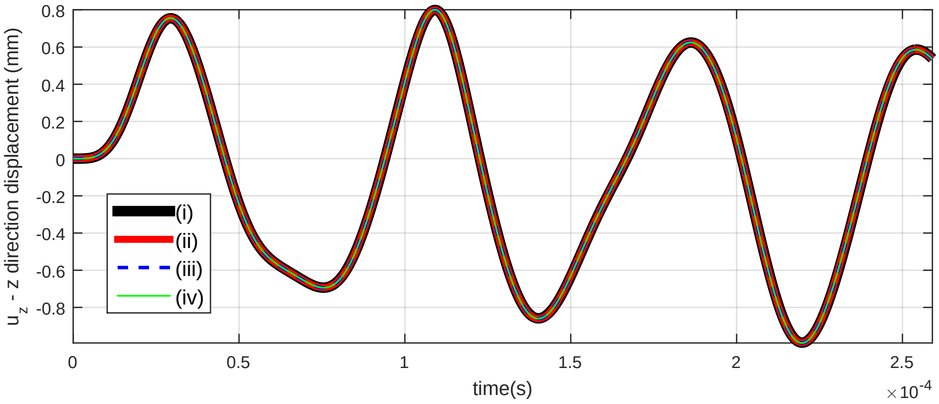


Figure 5.30: The Reference nonlinear solution (i) compared to the linear manifold solution containing 10 VMs and *All* (55) MDs (ii), selected 10 MDs using *MMI* method (iii) and selected 10 MDs using *MVW* method (iv)

.

Quadratic Manifold

A quadratic manifold is used with $m = 10$ modes reducing the number of unknowns to 10 instead of 65 as in case of LM. Results for the *regular* as well as *tensorial* approach are shown in Figure 5.14. It can be seen that the reduced solution using quadratic manifold is not as accurate as that using the linear manifold (Also see Table 5.5).

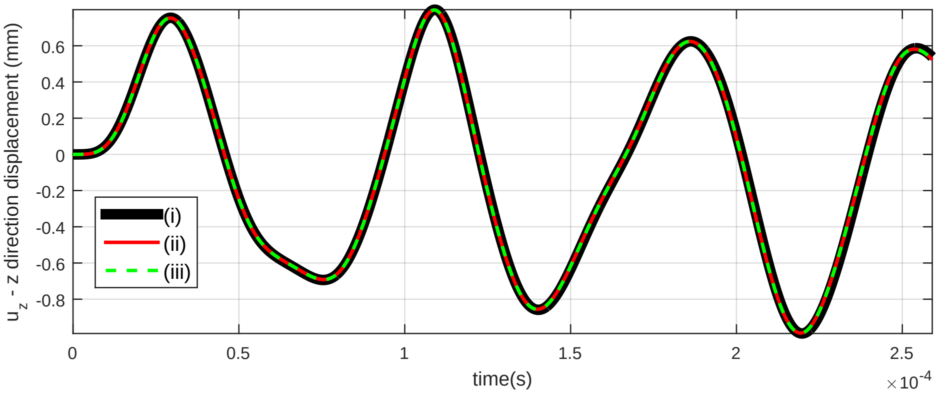


Figure 5.31: The Reference nonlinear solution (i) compared with the Quadratic manifold reduction with size of the reduced system $m = 7$. The two techniques for numerical solution *regular* (ii) and *tensorial* (iii) are compared.

Effect of neglecting the higher order tensors: As with the previous cases, if all the elastic tensors with order higher than and including ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ are neglected, then the nonlinear response is not captured. The structure loses stiffness in this particular example as seen by the higher amplitudes. It can also be seen (in Figure 5.32 and Table 5.5) that the quality of the solution starts degrading as tensors are neglecting from highest to lowest order. As observed in Model-III L1b, the effect of neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ negligible compared to that of neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{4}_{5}\boldsymbol{\mathcal{K}}$. Since the operations involving higher order tensors are expensive online as well as offline, it is of interest that neglecting them has consistently *not* hampered the accuracy of results.

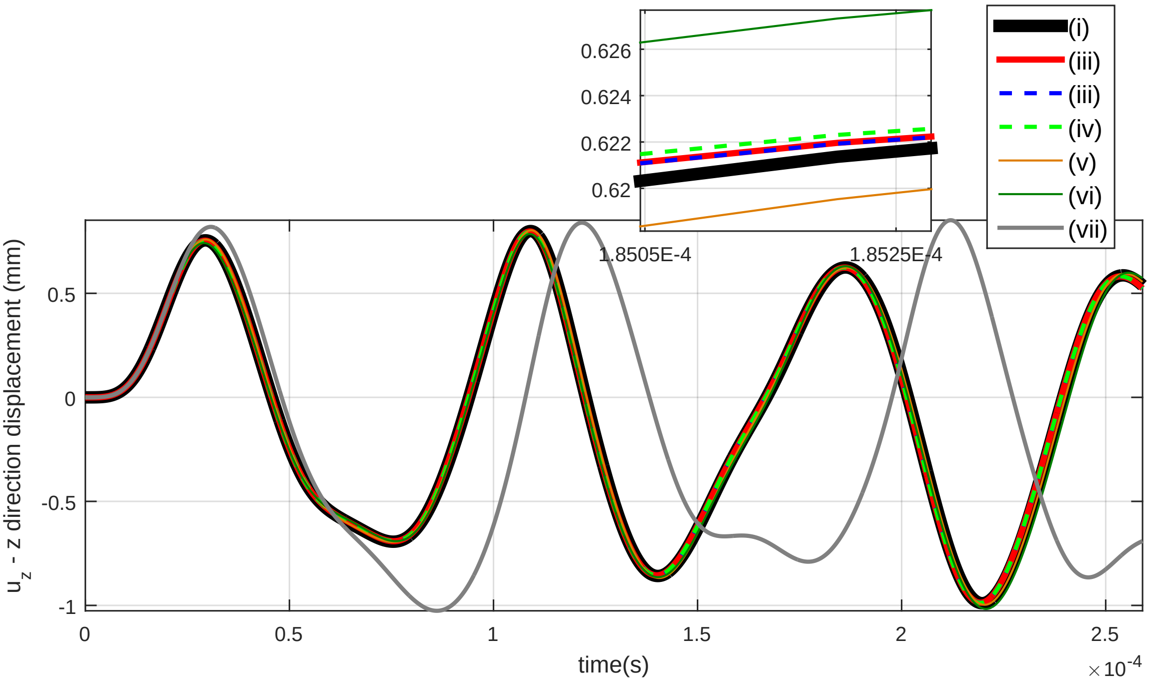


Figure 5.32: shows the Full nonlinear solution (i), QM reduced solution $(m = 10)$ using *tensorial* approach with all tensors (ii), neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ (iii) , neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ onwards (iv), neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards (v), neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards (vi) and finally neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards (vii). The effect of neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ is negligible compared to that of ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ and ${}^{4}_{5}\boldsymbol{\mathcal{K}}$. Neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ leads to blow up of solution.

ECSW

Hyper-reduction is performed in the following ways on Model-III L2 keeping number of training vectors $n_t = 100$, consistent among all the methods:

1. **ECSW-I -** POD basis obtained by SVD of the snapshots of the reference nonlinear solution. 65 most significant vectors were chosen ($k = 65$). Training vectors taken from full nonlinear solution.
2. **ECSW-II -** The reduction basis used in linear manifold (composed of VMs and all MDs) containing $k = 65$ vectors. Training vectors taken from LM reduced solution snapshots.
3. **ECSW-III -** The reduction basis same as ECSW-II. The modal amplitude snapshots (equally spaced) obtained from the linear modal superposition solution with $m = 10$ modes were fed to the quadratic manifold mapping to obtain training vectors for the ECSW.
4. **ECSW-IV -** The reduction basis used in linear manifold (composed of VMs and selected MDs using MMI technique (Section 3.3.2) containing $k = m + n_{MD}$ vectors, where $m = 10$ is the number of VMs used and $n_{MD} = m = 10$ is the number of MDs selected using appropriate technique. Training vectors same as that in ECSW-III.

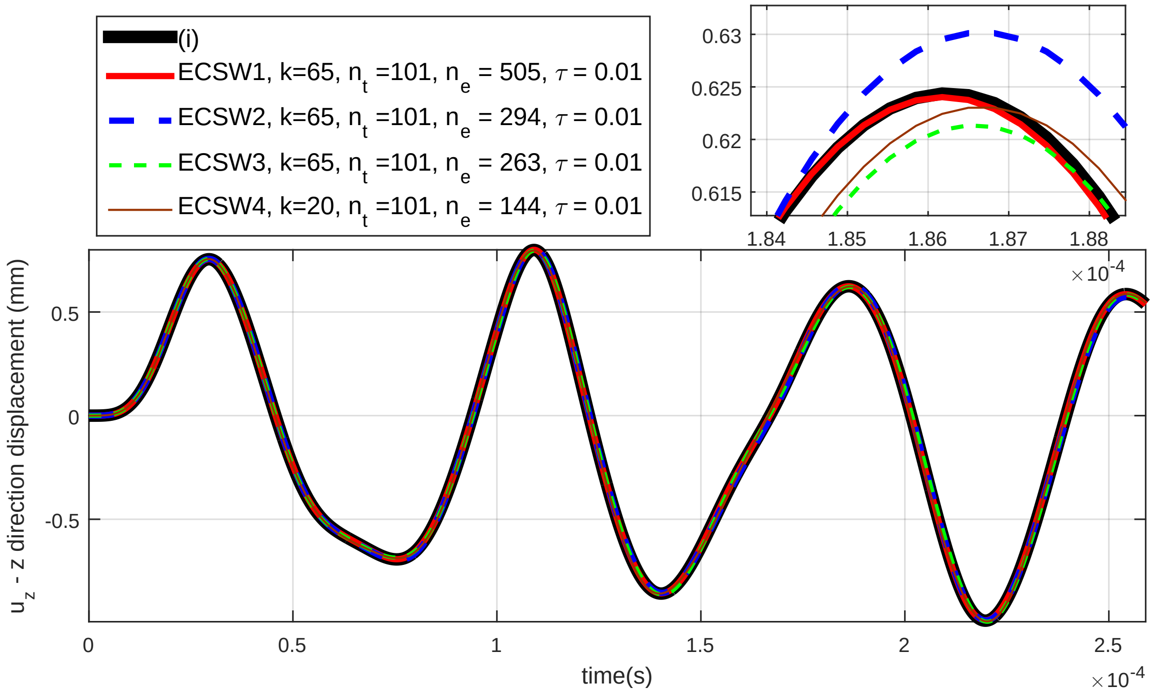


Figure 5.33: showing Hyper reduction using ECSW for different methods (I-IV). The full system contain 77712 elements, where sNNLS returns less than 500 elements to be sampled and weighted for evaluation and projection of nonlinearity. This gives huge speed up.

It is interesting to see that the proposed ECSW based hyper-reduction techniques have shown consistent results among all the tested examples.

Error Statistics

The GRE (Equation (5.1)) and Speed-up (Equation (5.4)) statistics for all the techniques tried on Model-III L2 are shown in Table 5.3. It is interesting to see that the general observations made for error statistics of Model-III L1a and Model-III L1b in Section 5.3.1 and Section 5.3.2 respectively are applicable for a different type of loading in Model-III L2 as well.

Table 5.5: Global Relative Error in Model-III-L2 for different reduction techniques.The solution time for a full nonlinear run was $\approx 3.743 \times 10^4$ seconds. The speed-up factor is calculated w.r.t. this time.

Reduction Technique	GRE_x (%)	GRE_y (%)	GRE_z (%)	Speed-up ($S^\star$)
LM Regular (All MDs)	1.20	0.53	1.39	1.40
LM Tensorial (All MDs)	1.20	0.53	1.39	884.00
LM Regular (Selected MDs - MMI)	1.46	0.68	1.62	2.04
LM Tensorial (Selected MDs - MMI)	1.46	0.68	1.62	4790.00
LM Regular (Selected MDs - MVW)	1.45	0.67	1.62	2.04
LM Tensorial (Selected MDs - MVW)	1.45	0.67	1.62	4784.00
Quadratic Manifold (Regular)	3.40	2.37	5.68	1.44
Quadratic Manifold (Tensorial)	3.39	2.36	5.71	118.00
QMT neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	3.38	2.33	5.68	148.20
QMT neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	3.45	2.44	5.79	249.90
QMT neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards	3.93	2.84	6.90	268.10
QMT neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards	6.28	5.47	9.99	275.22
QMT neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards	>100	>100	>100	278.29
POD	0.004	0.001	0.01	1.69
ECSW - I	0.64	0.31	0.98	163.70
ECSW - II	5.00	2.32	5.36	255.20
ECSW - III	5.13	2.39	7.01	265.60
ECSW - IV	4.05	2.49	6.58	883.00

All the reduction techniques which have been considered in this work were applied and tested in this chapter on examples which varied from being academically simple to realistically complex. These application have given a good overview of the relative behaviour and accuracy of the results. General claims about accuracy of these techniques require more in-depth research by taking an abstract and analytical approach. This is left for future work. General claims about speed and computational complexity however, are fairly straight forward to make. This shall be addressed in the next chapter.

6

Computational Complexity

The accuracy of the methods and respective savings in computational time have been established with the few applications discussed in the previous chapter. However, this doesn't give a clear overview as to what kind of speed-ups shall be expected in general models of arbitrary size using a given reduction method and in a more qualitative sense which method should be applied given the fact that some of the reduction techniques require significant offline effort (so much so, that the benefits of time savings during the online time integration becomes questionable). To address this, it is needed to know how the operations performed during these techniques qualitatively scale when the system size increases and larger bases are used for reduction, both offline as well as online. The complexity comparisons would be split into the operations which are done offline and those done during the time integration online.

6.1. Online Costs

Here, it is attempted to evaluate the complexity of operations performed during the time integration of reduced order models obtained using different reduction techniques. The Newmark time integration scheme is used for this purpose. The system of nonlinear equations (whether reduced or full) can be generalized in the following manner

$$\begin{aligned} \mathbf{r}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) &= \mathbf{0} \\ \mathbf{q}(t_0) &= \mathbf{q}_0 \\ \dot{\mathbf{q}}(t_0) &= \dot{\mathbf{q}}_0 \end{aligned}$$

The Newmark time iteration scheme can be coarsely summarized by the following basic steps

1. `INITIALIZATION`
   Given $\mathbf{q}_0$, $\dot{\mathbf{q}}_0$, Solve for $\ddot{\mathbf{q}}_0$

$$\mathbf{r}(\mathbf{q}_0, \dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0, t_0) = \mathbf{0}$$

2. `TIME STEP INCREMENT` (up to final time)

$$t_{p+1} = t_p + h_p$$

3. `PREDICTION`

$$\begin{aligned} &\dot{\mathbf{q}}_{p+1} \leftarrow \dot{\mathbf{q}}_p + (1-\gamma) h_p \ddot{\mathbf{q}}_p \\ &\mathbf{q}_{p+1} \leftarrow \mathbf{q}_p + h_p \dot{\mathbf{q}}_p + (0.5-\beta) h_p^2 \ddot{\mathbf{q}}_p \\ &\ddot{\mathbf{q}}_{p+1} \leftarrow \mathbf{0} \end{aligned}$$

```
4. (REDUCED)RESIDUAL EVALUATION
       $\mathbf{r}_{p+1} = \mathbf{r}(\mathbf{q}_{p+1}, \dot{\mathbf{q}}_{p+1}, \ddot{\mathbf{q}}_{p+1}, t_{p+1})$

5. CONVERGENCE CHECK
       if SOLUTION CONVERGED (e.g. $\|\mathbf{r}_{p+1}\| < \epsilon\|\mathbf{l}_{p+1}\|$)
       goto step 2
       endif

6. CALCULATE (REDUCED)JACOBIAN
       $\mathbf{S}(\mathbf{q}_{p+1}) = \frac{\partial \mathbf{r}_{p+1}}{\partial \mathbf{q}_{p+1}} + \frac{\gamma}{\beta h}\frac{\partial \mathbf{r}_{p+1}}{\partial \dot{\mathbf{q}}_{p+1}} + \frac{1}{\beta h^2}\frac{\partial \mathbf{r}_{p+1}}{\partial \ddot{\mathbf{q}}_{p+1}}$

7. INCREMENT EVALUATION
   Solve for $\Delta\mathbf{q}$,
       $\mathbf{S}(\mathbf{q}_{p+1})\Delta\mathbf{q} = -\mathbf{r}_{p+1}$

8. CORRECTION
       $\mathbf{q}_{p+1} \leftarrow \mathbf{q}_{p+1} + \Delta\mathbf{q}$
       $\dot{\mathbf{q}}_{p+1} \leftarrow \dot{\mathbf{q}}_{p+1} + \frac{\gamma}{\beta h_p}\Delta\mathbf{q}$
       $\ddot{\mathbf{q}}_{p+1} \leftarrow \ddot{\mathbf{q}}_{p+1} + \frac{1}{\beta h_p^2}\Delta\mathbf{q}$

9. LOOP, GOTO step 4
```

Out of these steps, 4,5,6,7 and 8 are repeated in every N-R iteration. In particular, steps 4,6 and 7 are the most significant(bottleneck) in determining the complexity of the time integration. In case of reduced equations, some reduction techniques require the evaluation of Jacobian and residual in the physical domain. In that case, the mapping of reduced coordinates to full system, calculation of full residual/Jacobian in physical domain, and thereafter projection onto appropriate subspace are the steps involved. This cost shall be included in step 4 and 5 itself, in which case the reduced residual/Jacobian are evaluated.

This complexity of the factorization of a dense $n \times n$ system has been optimized over the past decades to the fastest of $\mathcal{O}(n^{2.38})$ operations (flops) by the famous algorithm in [24]. Nonetheless, the naïve factorization estimates of $\mathcal{O}(n^3)$ operations (flops) have been used here for complexity estimation. In case of sparse systems, the solution depends a lot on the nature of sparseness. In the current context sparse matrices are generated for equations in 2 dimensions and its upper and lower bandwidth can be expected to be $\approx \sqrt{n}$, and for such a matrix linear solve using LU decomposition would require only $\mathcal{O}(n^2)$ flops [25]. In a general way, we represent the complexity of solution of a sparse $n \times n$ system by $\mathcal{O}(s(n))$. Similarly the cost of a matrix vector multiplication which is $\mathcal{O}(n^2)$ for a dense matrix becomes $\mathcal{O}(an)$ for a sparse one, where a is a constant dependent on the bandwidth of the matrix. Therefore, in the current context the this cost would take $\mathcal{O}(n^{3/2})$ flops.

In case of the (reduced)residual and/or the (reduced)Jacobian computation, the techniques which don't use tensors (such as LM Regular, QM Regular, ECSW, LM with selected MDs) require a mapping from reduced coordinates to full coordinates. Then the reduced residual/Jacobian is evaluated without the need for sparse assembly of the full tangent stiffness matrix (which can be expensive) as follows

$$\mathbf{r}(\mathbf{q}) = \mathbf{V}^T\mathbf{r}^{full}(\mathbf{u}(\mathbf{q})) = \sum_{e=1}^{n_e} \mathbf{V}_e^T\mathbf{r}_e^{full}(\mathbf{u}_e(\mathbf{q})), \tag{6.1}$$

$$\mathbf{S}(\mathbf{q}) = \frac{\mathrm{d}\mathbf{r}(\mathbf{q})}{\mathrm{d}\mathbf{q}} \approxeq \mathbf{V}^T\mathbf{S}^{full}(\mathbf{u}(\mathbf{q}))\mathbf{V} = \sum_{e=1}^{n_e} \mathbf{V}_e^T\mathbf{S}_e^{full}(\mathbf{u}_e(\mathbf{q}))\mathbf{V}_e, \tag{6.2}$$

where $\mathbf{V}$ represents the tangent subspace $\frac{\partial\mathbf{u}(\mathbf{q})}{\partial\mathbf{q}}$ which can be constant or variable depending on the mapping $\mathbf{u}(\mathbf{q})$ being linear or nonlinear (QM) respectively, $\mathbf{r}^{full}(\mathbf{u}) \in \mathbb{R}^n$ is the full residual, $\mathbf{S}^{full} \in \mathbb{R}^{n\times n}$ is the Jacobian $\frac{\mathrm{d}\mathbf{r}^{full}}{\mathrm{d}\mathbf{u}}$ $\mathbf{r}_e^{full}(\mathbf{u_e}) \in \mathbb{R}^{N_e}$ is the contribution of element e towards $\mathbf{r}^{full}$, $\mathbf{u}_e$ is the restriction of $\mathbf{u}$ to the rows indexed by DOFs corresponding to element e which contains N_e DOFs, $\mathbf{S}_e^{full}(\mathbf{u}_e)\mathbb{R}^{N_e\times N_e}$ is contribution of the element e towards the matrix $\mathbf{S}^{full}$. Note that the Jacobian $\mathbf{S}(\mathbf{q})$ is treated as $\mathbf{V}^T\mathbf{S}^{full}(\mathbf{u}(\mathbf{q}))\mathbf{V}$ which can be exact or an approximation depending on the mapping $\mathbf{u}(\mathbf{q})$ being linear or nonlinear as shown in Section 3.4.2. The number of flops in these operations scales linearly with the total number of elements n_e (or $|E|$ in case of ECSW). Though the number of elements can be related to the number of DOFs n depending on the topology of the domain, for sake of generality and ease of understanding the n_e has been used anyway in the estimation of the complexity of such operations. The complexity is estimated for some crucial operations in Table 6.1.

Table 6.1: Cost estimation for the Newmark implicit time integration during various full nonlinear time integration reduction schemes. n_e = total number of elements, n = total number of DOFs in full system, m = number of VMs used in mapping (Note the size of LM basis containing m VMs and all corresponding MDs would be $\mathcal{O}(m^2)$), $m_s = m + n_{MD}$ is the total number of vectors in basis containing selected MDs, k number of vectors in reduction basis of ECSW, E subset containing the elements sampled by ECSW

Operation	Mapping	(Reduced) residual	(Reduced)	Increment
			Jacobian	Evaluation
Full NL	-	$\mathcal{O}(an)$[a] $+\mathcal{O}(n_e)$[b]	$\mathcal{O}(n_e)$[b]	$\mathcal{O}(s(n))$[c]
LM Regular (All MDs)	$\mathcal{O}(nm^2)$[d]	$\mathcal{O}(m^4)$[e] $+\mathcal{O}(n_em^2)$[f]	$\mathcal{O}(n_em^4)$[g]	$\mathcal{O}(m^6)$
LM Tensorial (All MDs)	-	$\mathcal{O}(m^8)$[h]	$\mathcal{O}(m^6)$[i]	$\mathcal{O}(m^6)$
LM Regular (Selected MDs)	$\mathcal{O}(nm_s)$[d]	$\mathcal{O}(m_s^2)$[e] $+\mathcal{O}(n_em_s)$[f]	$n_e\mathcal{O}(n_em_s^2)$[g]	$\mathcal{O}(m_s^3)$
LM Tensorial (Selected MDs)	-	$\mathcal{O}(m_s^4)$[h]	$\mathcal{O}(m_s^3)$[i]	$\mathcal{O}(m_s^3)$
QM Regular	$\mathcal{O}(nm^2)$[j]	$\mathcal{O}(anm)$[k] $+\mathcal{O}(n_em)$[f]	$\mathcal{O}(n_em^2)$[g]	$\mathcal{O}(m^3)$
QM Tensorial	-	$\mathcal{O}(m^8)$	$\mathcal{O}(m^7)$	$\mathcal{O}(m^3)$
QMT neglecting ${}^7_8\boldsymbol{\mathcal{K}}$	-	$\mathcal{O}(m^7)$	$\mathcal{O}(m^6)$	$\mathcal{O}(m^3)$
QMT neglecting ${}^6_7\boldsymbol{\mathcal{K}}$ and ${}^7_8\boldsymbol{\mathcal{K}}$	-	$\mathcal{O}(m^6)$	$\mathcal{O}(m^5)$	$\mathcal{O}(m^3)$
QMT neglecting ${}^5_6\boldsymbol{\mathcal{K}}$ onwards	-	$\mathcal{O}(m^5)$	$\mathcal{O}(m^4)$	$\mathcal{O}(m^3)$
QMT neglecting ${}^4_5\boldsymbol{\mathcal{K}}$ onwards	-	$\mathcal{O}(m^4)$	$\mathcal{O}(m^3)$	$\mathcal{O}(m^3)$
QMT neglecting ${}^3_4\boldsymbol{\mathcal{K}}$ onwards	-	$\mathcal{O}(m^3)$	$\mathcal{O}(m^2)$	$\mathcal{O}(m^3)$
ECSW	$\mathcal{O}(nk)$[d]	$\mathcal{O}(k^2)$[e]$+\mathcal{O}(\|E\|k)$[l]	$n_e\mathcal{O}(\|E\|k^2)$[m]	$\mathcal{O}(k^3)$

[a] sparse Matrix-vector multiplication $\mathbf{M\ddot{u}}$. $\mathcal{O}(n^{3/2})$ in current contest of sparsity.
[b] element Internal force / tangent stiffness calculation for all elements
[c] sparse linear solve. $\mathcal{O}(n^2)$ in current context of sparsity
[d] matrix multiplication $\mathbf{u} = \mathbf{Vq}$
[e] projected Inertial force calculation $\hat{\mathbf{M}}\ddot{\mathbf{q}}$ ($\hat{\mathbf{M}} = \mathbf{V}^T\mathbf{MV}$ is dense calculated offline)
[f] projected internal force calculated for all elements $\sum_{e=1}^{n_e} \mathbf{V}_e^T\mathbf{f}_e(\mathbf{u}_e(\mathbf{q}))$
[g] reduced Jacobian $\sum_{e=1}^{n_e} \mathbf{V}_e^T\mathbf{S}_e(\mathbf{u}_e(\mathbf{q}))\mathbf{V}_e$

[h] bottleneck : tensor product ${}^3_4\hat{\mathbf{K}}\vdots(\mathbf{q}\otimes\mathbf{q}\otimes\mathbf{q})$
[i] tangent cost is included in residual cost
[j] bottle neck tensor product $\boldsymbol{\Theta} : (\mathbf{q}\otimes\mathbf{q})$, tangent subspace automatically evaluated in this process
[k] cost of multiplication $\partial\mathbf{U}^T\mathbf{M\ddot{u}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$
[l] projected internal force calculated for sampled elements $\sum_{e\in E} \mathbf{V}_e^T\mathbf{f}_e(\mathbf{u}_e(\mathbf{q}))$
[m] reduced Jacobian $\sum_{e\in E} \mathbf{V}_e^T\mathbf{S}_e(\mathbf{u}_e(\mathbf{q}))\mathbf{V}_e$

Following important remarks can be made after performing numerical experiments on meshes of different sizes and using different number of modes in projection. (See Figures 6.1 to 6.4)

- Among the *regular* approaches (such as Linear Manifold Regular, Quadratic Manifold Regular, Linear Manifold with selected MDs, POD) which involve the computation of residual in the physical domain (by mapping, assembly and projection) the online effort keeps increasing with the increase in system size (n_e). As the basis (or mapping) size (m) increases, there comes a point where the reduced time integration becomes more expensive (due to extra projection and mapping operations) than solving the model in its full form.

- Among the *tensorial approaches*, where the reduced nonlinearlity is evaluated offline in the form of tensors, the online computational effort becomes *independent* of the full system size and order of magnitudes cheaper than the regular approaches (for "small" m). However, this computational effort increases exponentially with the increase in m to the point when its becomes redundant to use it. This limit for m is system dependent. A larger system would lead to larger limit.

- The Linear Manifold Tensorial(LMT) approach with all MDs, scales as fast with m as the Ouadratic Manifold Tensorial (QMT). This can be concluded from Table 6.1 and is also supported by the numerical experiment results in Figures 6.1 to 6.4. It is reasonable that the LMT is cheaper than QMT by a constant factor since it exploits the symmetry of the MDs in the current implementation.

- All of the test examples showed that the Linear manifold tensorial approach was faster than the Quadratic Manifold tensorial even after neglecting the higher order tensors. However, the complexity analysis in Table 6.1 shows that the LMT loses to the QM Tensorial if even just ${}^{7}_{8}\boldsymbol{\mathcal{K}}$ is neglected. Same is supported by numerical experiment results e.g. in Figure 6.1 whereby this is true for $m \geq 13$.

- Linear Manifold Tensorial with selected MDs remains cheap since the basis size has been chosen as $m_s = 2m$, and thus increases only scaled linearly m instead of quadratically in case when all MDs are selected. However, it should be noted that this MD selection is load dependent and would require separate computation of corresponding tensors once other MDs are triggered by the load and selection basis needs to be updated.

- All the computations for tensorial operations are inhibitive in terms of memory usage. Although m is a small number but the memory required for tensor storage increases exponentially with m. And thus tensor manipulations become impossible due to memory shortage after a certain m. It is easy to see that the storage space for a r^{th} order tensor with m elements in each dimension scales with $\mathcal{O}(m^r)$. Thus, similar trends for memory usage can be expected as those for computational complexity in the tensorial cases. The limits of m for which the storage becomes impossible with the available resources for tensors in a particular tensorial technique are as follows:

Tensorial technique	Limiting value of m for available memory
LM Tensorial - All MDs	24
LM Tensorial - m Selected MDs (MMI or MVW)	>30
QM Tensorial	17
QM Tensorial neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	25
QM Tensorial neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$	>30

- The online costs during the time integration using the ECSW are sensitive to the training vectors & reduction basis used, and the set of elements thus sampled. Thus, numerical experiments which generalize its behaviour cannot be conducted. However, it is easy to see that the online costs would be would be a fraction $|E|/n_e$ of the online costs involved in a *regular* projection based MOR technique with a ROB of the same size. The sparser the solution of the sNNLS problem 4.14, the lesser the online costs.

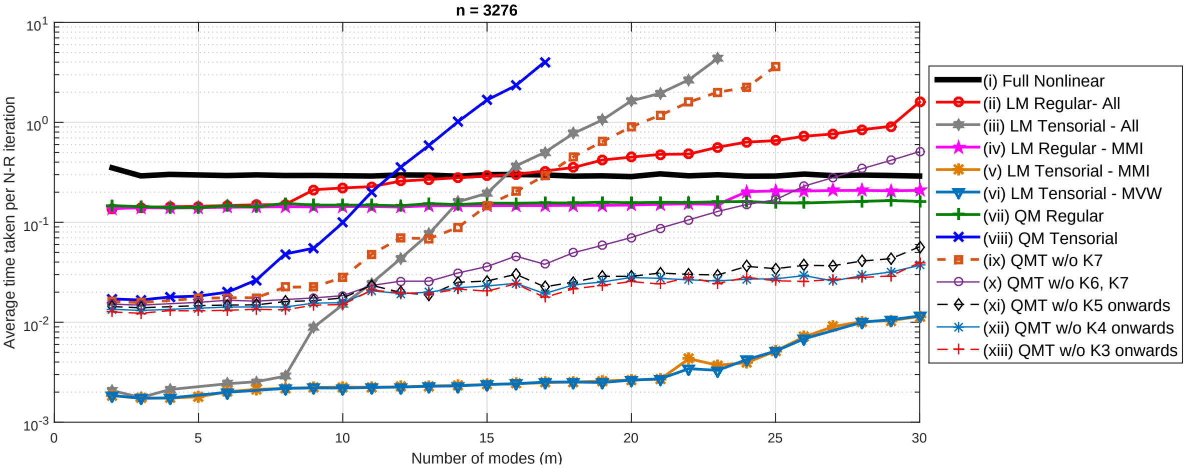


Figure 6.1: **Complexity Comparisons-I:** A small model with $n = 3276$ DOFs is tested for average cost(time) of a N-R iteration performed while implicit Newmark time integration. m (x axis) represents the number of modes which govern the reduction techniques. It can seen that for such small models as m increases, most of the reduction techniques seem redundant.

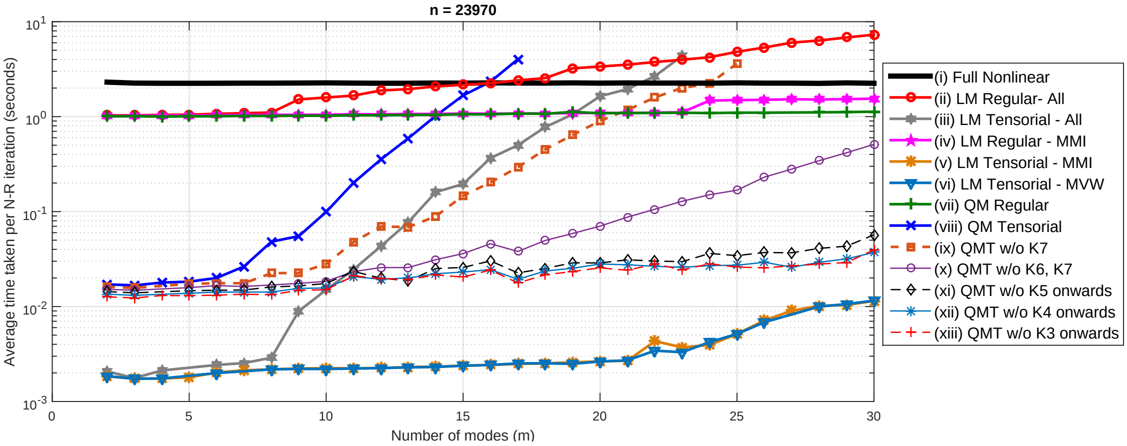


Figure 6.2: **Complexity Comparisons-II:** A medium sized model with $n = 23970$ DOFs is tested for average cost(time) of a N-R iteration performed while implicit Newmark time integration. m (x axis) represents the number of modes which govern the reduction techniques. The *regular* techniques take higher time and the *tensorial* techniques become more feasible compared to the smaller model of Figure 6.1

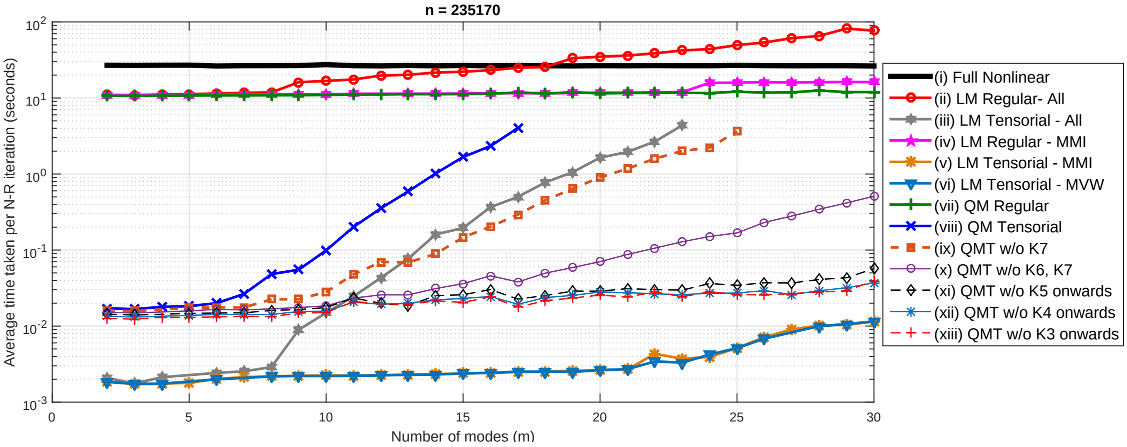


Figure 6.3: **Complexity Comparison-III:** A slightly large model with $n = 235170$ DOFs is tested for average cost(time) of a N-R iteration performed while implicit Newmark time integration. m (x axis) represents the number of modes which govern the reduction techniques. (*viii*) showing the QM Tensorial approach with all tensors becomes infeasible due to constraints of available memory resources for $m \geq 18$.

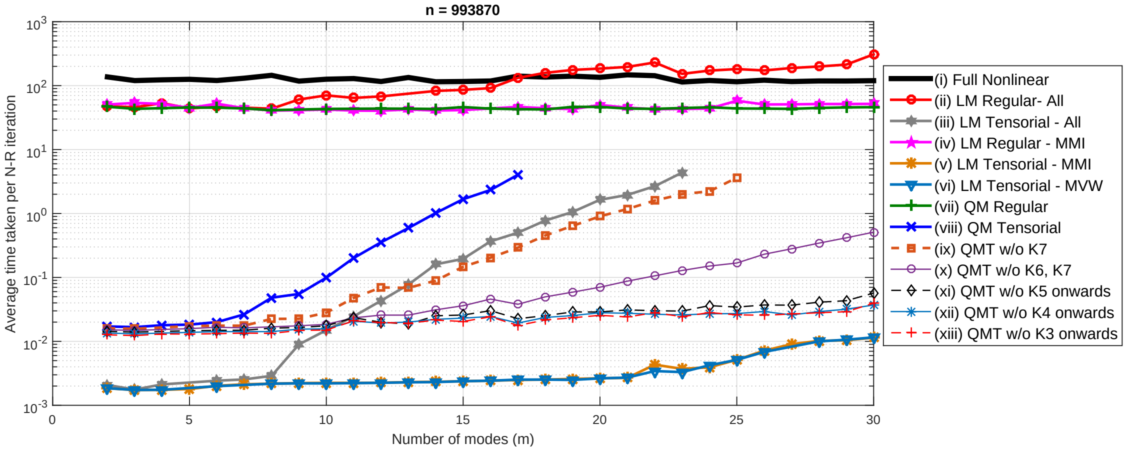


Figure 6.4: **Complexity Comparison-IV:** A realistically large model with almost a million DOFs is tested for average cost(time) of a N-R iteration performed while implicit Newmark time integration. m (x axis) represents the number of modes which govern the reduction techniques. The tensorial techniques (though constrained by available memory) are very beneficial in this case.

6.2. Offline Costs

The evaluation of offline costs for different methods is also important for fair comparison of the reduction techniques. The offline cost of a full nonlinear solution is considered zero. Thus, the time required to build the model from scratch is considered zero for all the reduction techniques since it is something common to all. Apart from that, any other offline cost which is not a part of the offline cost for full solution is treated as an offline cost for the any reduction technique. Note that the term *cost* is used in the context of CPU time / number of floating point operations here.

These offline costs depend on the techniques being used. Among all the offline costs, the solution of the generalized eigenvalue problem Equation (3.16) for the eigenmodes and eigenvalues is the most important and common. For a dense system, eigenvalue problem solution takes $\mathcal{O}(n^3)$ flops using methods such as QR and QZ (See for example [26]). These methods compute all the eigenvectors and require the matrices to be specified in explicit form which can require a large amount of memory $\mathcal{O}(n^2)$. However, in practise only a few eigenmodes are needed and the matrices **K** & **M** are sparse. Eigensolvers for large sparse systems which compute m eigenmodes are denoted here to require $\mathcal{O}(p(m,n))$ flops (could be Krylov subspace based methods, power iteration, orthogonal deflation etc.).

The majority of the methods also requires the computation of Modal Derivatives. These are obtained from the solution of the sensitivity problem Equation (3.31). Neglecting symmetry of MDs, m modes require the solution of the derivative problem m^2 times. However, the coefficient matrix $(\mathbf{K}-\omega_i^2\mathbf{M})$ needs to be factorized only m times. The RHS however needs to be evaluated m^2 times which involves sparse matrix vector products $(\frac{\partial \mathbf{K}}{\partial q_j}\boldsymbol{\phi}_i)$ between the stiffness sensitivity and the eigenmode. The computation of the stiffness sensitivity $\frac{\partial \mathbf{K}}{\partial q_j}$ has to be performed in total m times (1 for each mode), each of which take time similar to the assembly of a tangent stiffness matrix and is relatively cheap in comparison to factorization cost.

The proposed techniques which involve selection of MDs to reduce basis size, require the computation of linear system using modal superposition. The time step size h for all the schemes has been kept constant here for easier comparison. The total number of time steps is denoted as n_h. The linear modal superposition based time integration for an undamped system can be performed using a recurrent matrix formulation as explained in [2]. This leads to m uncoupled second order ODEs which can be easily solved in $\mathcal{O}(mn_h)$ flops for a constant time step. Damped time integration can be performed using Galerkin projection of the linear system onto a ROB composed of VMs. This leads to requirement of $\mathcal{O}(m^2n_h)$ flops which is still cheap for an offline cost.

The MD weightage cost for MMI selection technique is $\mathcal{O}(mn_h) + \mathcal{O}(m^2)$ and sorting the weights would cost $\mathcal{O}(m^2 \log m^2)$ flops in general (see e.g. [27]). The MVW technique, however involves $\mathcal{O}(mn_h) + m^2\mathcal{O}(nn_e)$ flops for weightage (when calculating the nonlinear internal force projected on the VMs) and same $\mathcal{O}(m^2 \log m^2)$ flops for sorting the weights and is thus expected to be expensive compared to MMI.

The tensor based methods also involve computation of higher order tensors and the complexity depends on the number of unknowns in the mapping (whether linear or nonlinear), and the total number of elements in the structure n_e. The complexity varies linearly with n_e. It is remarkable that the offline cost for tensor calculation is highly parallelizable. Tensors can be computed for different parts of a mesh on different processors and then simply added to obtain the tensors for the complete structure.

Some of the methods like POD, ECSW-I (which is POD based) also require the computation of the full nonlinear solution for generation of training sets. In such a case, the reference nonlinear solution time cost becomes an offline cost. The ECSW involves 3 basic offline costs :

1. Generating Training vectors.
2. Construction of **G** matrix
3. Solving sNNLS problem to obtain element sampling and weighting.

Generation of training vectors is largely dependent on the method used. On each of the training vectors, the element wise contribution of the nonlinear internal force is evaluated and then projected onto the element-wise contribution of the reduction basis. This results in the **G** matrix and would require

$n_t\mathcal{O}(n_e m)$ flops. Finally, the computational complexity of the sNNLS algorithm is bounded from above by that of the active set NNLS complexity [22], which is shown to converge but can be slow for large systems. This is due to the solution of a linear least squares problem in each iteration of the algorithm which is generally done via QR decomposition and carries $\mathcal{O}(rs^2)$ time-complexity for a $r \times s$ matrix $(r > s)$. Thus, if it takes l iterations to converge the sNNLS algorithm 2 can be crudely estimated to be of $\mathcal{O}(ln_e(kn_t)^2)$ complexity. It is worth mentioning that a more efficient method to solve the NNLS problem using parallel computing is developed in [28].

Keeping all these factors in mind, similar qualitative estimations for offline operations for various reduction techniques can be made as that for the computations done online. These are shown in Table 6.2.

Table 6.2: Estimation for the *offline* costs required before beginning the Newmark implicit time integration. n_e = total number of elements, n = total number of DOFs in full system, m = number of VMs used in mapping (Note the size of LM basis containing m VMs and all corresponding MDs would be $\mathcal{O}(m^2)$), $m_s = m + n_{MD}$ is the total number of vectors in basis containing selected MDs, k number of vectors in reduction basis of ECSW, E subset containing the elements sampled by ECSW, n_t = number of training vectors used in ECSW, n_h = number of time steps during time integration.

Operation	Offline Cost Complexity
Full NL	0
LM Regular (All MDs) (C_1)	$\mathcal{O}(p(m,n))$[a] $+\, m\mathcal{O}(s(n))$[b] $+\, m^2\mathcal{O}(an)$[c] $+\, m\mathcal{O}(n_e)$[d]
LM Tensorial (All MDs)	$C_1 + \mathcal{O}(n_e m^8)$[e]
LM Regular (Selected MDs) (C_2)	$C_1 + \mathcal{O}(n_h m^2)$[f] $+\, \mathcal{O}(m^2)$[g] $+\, \mathcal{O}(m^2 \log m^2)$ [h]
LM Tensorial (Selected MDs)	$C_2 + \mathcal{O}(n_e m_s^4)$
QM Regular	C_1
QM Tensorial	$C_1 + \mathcal{O}(n_e m^8)$[i]
QMT neglecting $^7_8\boldsymbol{\mathcal{K}}$	$C_1 + \mathcal{O}(n_e m^7)$ [j]
QMT neglecting $^6_7\boldsymbol{\mathcal{K}}$ and $^7_8\boldsymbol{\mathcal{K}}$	$C_1 + \mathcal{O}(n_e m^6)$[k]
QMT neglecting $^5_6\boldsymbol{\mathcal{K}}$ onwards	$C_1 + \mathcal{O}(n_e m^5)$[l]
QMT neglecting $^4_5\boldsymbol{\mathcal{K}}$ onwards	$C_1 + \mathcal{O}(n_e m^4)$[m]
QMT neglecting $^3_4\boldsymbol{\mathcal{K}}$ onwards	$C_1 + \mathcal{O}(n_e m^4)$ [n]
ECSW	C_{n_t} [o] $+\, n_e n_t \mathcal{O}(m)$[p] $+\, C_{SNNLS}$ [q]

[a] Cost of solution of Eigenvalue problem Equation (3.16) for m modes

[b] Cost of solution of m sparse $n \times n$ systems given by Equation (3.31). $\mathcal{O}(n^2)$ in current context of sparsity.

[c] Cost of m^2 sparse matrix vector multiplications $\frac{\partial \mathbf{K}}{\partial q_j}\phi_i$

[d] Cost of Stiffness sensitivity $\frac{\partial \mathbf{K}}{\partial q_j}$ assembly w.r.t. to m modes. Each cost same as tangent stiffness assembly.

[e] Cost of tensor assembly Equations (3.62) and (3.63)

[f] Cost of linear modal damped time integration

[g] Cost of finding the weightage Equation (3.37)

[h] Cost of sorting weights

[i] cost of element wise assembly of all tensors in Equation (3.109)

[j] cost of element wise assembly of all tensors in Equation (3.109) except $^7_8\boldsymbol{\mathcal{K}}$

[k] cost of element wise assembly of all tensors in Equation (3.109) except $^7_8\boldsymbol{\mathcal{K}}, {^6_7\boldsymbol{\mathcal{K}}}$

[l] cost of element wise assembly of all tensors in Equation (3.109) except $^7_8\boldsymbol{\mathcal{K}}, {^6_7\boldsymbol{\mathcal{K}}}, {^5_6\boldsymbol{\mathcal{K}}}$

[m] cost of element wise assembly of all tensors in Equation (3.109) except $^7_8\boldsymbol{\mathcal{K}}, {^6_7\boldsymbol{\mathcal{K}}}, {^5_6\boldsymbol{\mathcal{K}}}, {^4_5\boldsymbol{\mathcal{K}}}$

[n] cost of element wise assembly of all tensors in Equation (3.109) except $^7_8\boldsymbol{\mathcal{K}}, {^6_7\boldsymbol{\mathcal{K}}}, {^5_6\boldsymbol{\mathcal{K}}}, {^4_5\boldsymbol{\mathcal{K}}}, {^3_4\boldsymbol{\mathcal{K}}}$ (There still exist fourth order tensor computations from damping and inertia.)

[o] Cost of generation of *training* vectors for ECSW

[p]Cost of construction of $\mathbf{G}$ and $\mathbf{b}$ (Equation (4.12))

[q] Cost of solution of sNNLS problem (Algorithm 2)

The following general remarks/observations can be made after studying complexity of offline operations involved in various reduction techniques.

- The Eigenvalue problem Equation (3.16) and Modal derivative problem have been an important part of most of the reduction techniques. These do not pose a very high offline cost (at least when compared to the their counterparts in POD). Numerical experiments were performed to check the time spent on solution of these problems for problems of different sizes for first m modes (See Figure 6.5).

- As explained before, the offline cost for calculation the tensors scales linearly with the number of elements in the system. Thus, the offline cost involved in calculation of Tensors depends only on m and can be calculated in a "per element" fashion (at least when all elements in the mesh contain the same number of DOFs). These results are depicted in Figure 6.6

- As for the online costs, the offline costs for tensor calculation requires additional storage in the CPU memory. This is space required is independent of the model size and only depends on m. The limiting values of m above which tensor computation becomes constrained by available memory is as follows

Tensorial technique	Limiting value of m for available memory
LM Tensorial - All MDs	21
LM Tensorial - m Selected MDs (MMI or MVW)	>30
QM Tensorial	16
QM Tensorial neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	22
QM Tensorial neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$	>30

- Due to the element level implementation of tensors, this calculation is massively parallelizable and tests show that the use of s processors in parallel reduces the offline cost by a factor of $\approx s$. Same holds true for all the tensor based approaches.

- Among other factors, the offline costs involved in ECSW depend on the method used to obtain the training vectors, which can be as large as the solution time of a full nonlinear solution (ECSW-I) or as small as that of a linear modal superposition run (ECSW-III & IV).

- After the training sets have been obtained for ECSW, it has been observed that the construction of $\mathbf{G}$ matrix contributing to the ECSW offline costs, is an expensive task both in terms of time and memory. It should be noted that $\mathbf{G}$ is a dense matrix with large dimensions and for large systems with many training vectors, its storage becomes inhibitive.

- The offline costs involved in solution of sNNLS problem 4.14 are case dependent but it has been observed that this cost is negligible compared to the other offline costs (even for large systems).

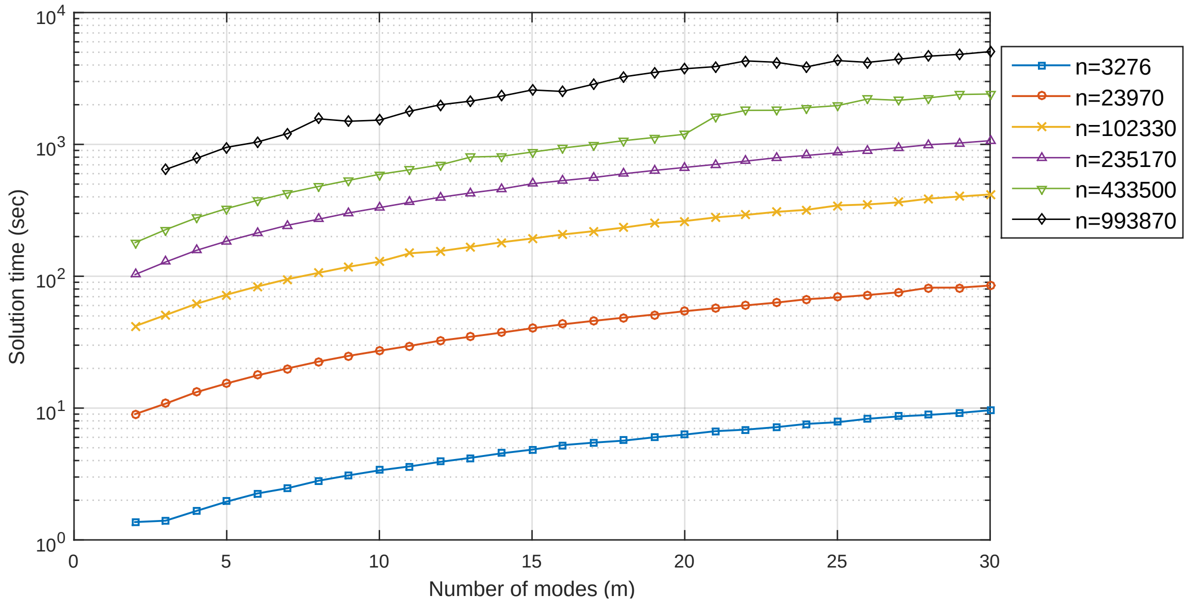


Figure 6.5: **Eigenmode and Modal Derivative computation time:** Models containing different number (n) of DOFs were tested for time required for computation (y axis) of first m eigenmodes (x axis) and all corresponding m^2 MDs as given by Equations (3.16) and (3.31) respectively. It is known that these are very important constituents for Reductions techniques discussed however, they're not very expensive to compute (at least when compared to a full solution run)

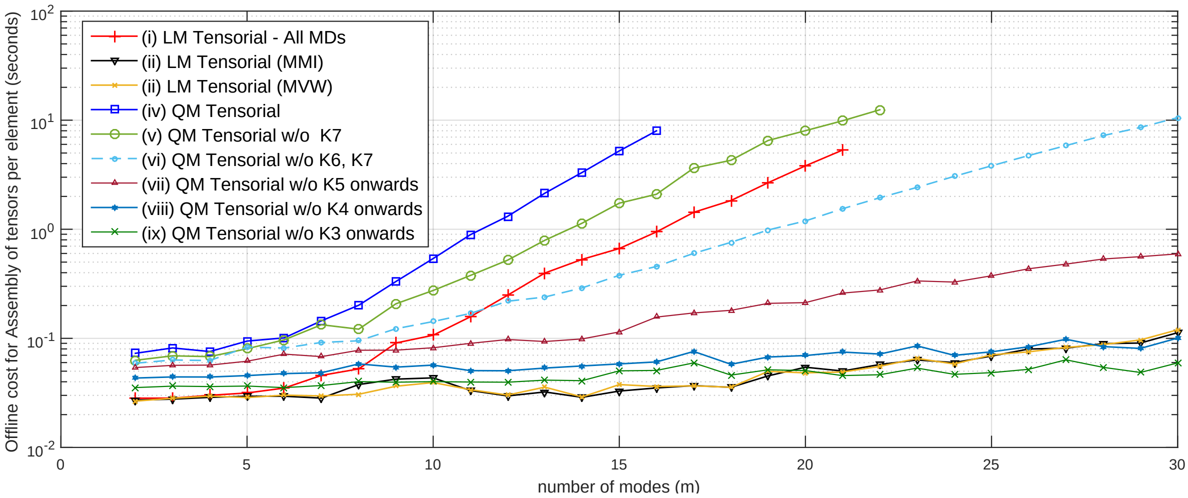


Figure 6.6: **Tensor Computation costs:** Time required for computation of higher order tensors depends linearly on the number of elements of a structure. All meshes contain similar elements with $N_e = 18$ DOFs per element. This cost is shown for computation of tensors in different techniques in a "per element" sense (y axis) which makes it independent of the model size. This cost increases exponentially with the size of tensors which is related to m (x axis). Notice that the available memory becomes a constraint as the tensor size increases (where data not available for higher values of m)

So far, qualitative and semi-quantitative estimates about the scaling of the important offline costs involved in various methods have been attempted. To take into account the missing details and put things into a better perspective, the offline costs involved in all the test examples were computed. These are reported in Table 6.3.

Table 6.3: **Offline Costs:** Results for the *offline* costs required before beginning the Newmark implicit time integration for different Models using various reduction methods. Note that for Model-III, MOR was shown for 3 different load cases(L1a, L1b, L2). Some of the techniques have offline costs which are load dependent and others don't, in which case a single number is presented instead of three.

Technique	Model-I	Model-II		Model-III	
	(s)	(s)	L1a(s)	L1b(s)	L2(s)
LM Regular (All MDs)	0.70	1.42		360.20	
LM Tensorial (All MDs)	60.57	50.42		9230.00	
LM Regular (Selected MDs - MMI)	0.85	1.59	386.10	386.40	386.10
LM Tensorial (Selected MDs - MMI)	51.14	101.30	2972.00	2918.00	2972.00
LM Regular (Selected MDs - MVW)	1.31	2.57	617.90	594.20	604.30
LM Tensorial (Selected MDs - MVW)	54.62	103.30	3125.00	3127.00	3111.00
Quadratic Manifold (Regular)	0.70	1.42		360.20	
Quadratic Manifold (Tensorial)	258.20	158.70		43430.00	
QMT neglecting ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	261.70	126.90		21430.00	
QMT neglecting ${}^{6}_{7}\boldsymbol{\mathcal{K}}$ and ${}^{7}_{8}\boldsymbol{\mathcal{K}}$	175.30	98.45		11150.00	
QMT neglecting ${}^{5}_{6}\boldsymbol{\mathcal{K}}$ onwards	107.50	66.71		6739.00	
QMT neglecting ${}^{4}_{5}\boldsymbol{\mathcal{K}}$ onwards	60.63	39.50		4449.00	
QMT neglecting ${}^{3}_{4}\boldsymbol{\mathcal{K}}$ onwards	41.82	28.60		3535.00	
POD	89.81	91.33	37680.00	39340.00	37440.00
ECSW - I	101.00	134.60	39410.00	43020.00	39450.00
ECSW - II	51.85	72.24	27680.00	29030.00	28160.00
ECSW - III	12.29	23.72	2103.00	2986.00	1815.00
ECSW - IV	10.53	15.38	1174.00	2140.00	1175.00

It can be seen from Table 6.3 that even for large models (Model-III), the offline cost for the Quadratic Manifold Tensorial approach is almost as high as the cost of a full solution run in this case (at least when all the tensors are calculated). However, it should be noted that this cost is load independent. Thus whereas a POD based method or hyper-reduction (such as ECSW-I) uses the full solution snapshots, it very much load dependent and every case requires a separate run to generate these snapshots.

6.3. Effective Speed-Up

The *regular* techniques, whether using a quadratic manifold or a linear manifold have minimal offline costs. However, as shown in Figures 6.1 to 6.4, the online cost for these techniques become inhibitive as the basis size increases thereby making them unusable. The tensorial techniques find use in use a situation where the online cost is independent of model size. But with all the benefits, these techniques are taxing in terms of memory and offline costs.

The hyper-reduction using ECSW tries to minimize the online cost incurred during the *regular* linear projection techniques by sampling elements on which nonlinearity is evaluated and high speed ups have been observed for it(See Tables 5.3 to 5.5). But conventionally (ECSW-I,II) it involves a full solution run for training set generation which is highly undesirable in the context of this research and leads to inhibitive offline costs.

However, the proposed hyper-reduction with cheap generation of training sets (ECSW-III & IV) somehow finds a balance between the two type of techniques and leads to affordable offline costs with very high online speed-ups. A crude estimate of an *effective speed-up* which gives weights to offline and online efforts can be defined in the following manner

$$S^{\star}_{effective} = \frac{c_{on} T_{full}}{c_{off} T^{\star}_{offline} + c_{on} T^{\star}_{online}}, \tag{6.3}$$

where T_{full} is the computational time for full nonlinear solution run. $T^{\star}_{offline}$ and $T^{\star}_{online}$ represent the computational time spent *offline* and *online* respectively, on the solution of the system using a given reduction technique, $c_{on}, c_{off} \in [0, 1]$ represent the user defined relative weights to be given to the online and offline costs respectively such that $c_{on} + c_{off} = 1$. As mentioned before, the full solution run is assumed to carry zero offline costs and thus $S_{effective} = 1$ for a full nonlinear run. A $S_{effective} < 1$ would imply that the reduction technique is not favourable in an overall sense and a $S_{effective} > 1$ would make it favourable with a higher value of $S_{effective}$ making the technique more desirable.This effective speed-up can be used to compare various reduction technique in a overall sense by taking into account the offline as well as online efforts.

The effective speed-up was computed for various reduction techniques by very conservatively giving *equal* weightage to offline and online costs i.e. $c_{on} = c_{off} = 0.5$. The results are shown in Table 6.4.

Table 6.4: Effective speed-up factor obtained after taking into account the offline as well as online cost for computations.

Technique	Model-I	Model-II	Model-III		
	(s)	(s)	Load1a	Load1b	Load2
LM Regular (All MDs)	2.12	1.74	1.42	1.78	1.38
LM Tensorial (All MDs)	1.33	1.25	4.05	4.19	4.04
LM Regular (Selected MDs - MMI)	2.09	1.78	2.05	2.17	2.00
LM Tensorial (Selected MDs - MMI)	1.69	0.82	12.64	12.11	12.56
LM Regular (Selected MDs - MVW)	2.07	1.80	2.04	2.19	1.98
LM Tensorial (Selected MDs - MVW)	1.59	0.80	12.03	11.55	12.00
Quadratic Manifold (Regular)	2.05	1.32	1.37	1.66	1.42
Quadratic Manifold (Tensorial)	0.33	0.45	0.86	0.89	0.86
QMT neglecting $\frac{7}{8}\mathcal{K}$	0.33	0.58	1.68	1.80	1.73
QMT neglecting $\frac{6}{7}\mathcal{K}$ and $\frac{7}{8}\mathcal{K}$	0.48	0.76	3.33	3.44	3.31
QMT neglecting $\frac{5}{6}\mathcal{K}$ onwards	0.76	1.05	5.47	5.65	5.44
QMT neglecting $\frac{4}{5}\mathcal{K}$ onwards	1.27	1.57	8.21	8.45	8.17
QMT neglecting $\frac{3}{4}\mathcal{K}$ onwards	1.76	1.84	10.25	10.63	10.20
POD	0.69	0.69	0.64	0.56	0.63
ECSW - I	0.84	0.54	0.95	0.93	0.94
ECSW - II	1.59	0.92	1.35	1.70	1.32
ECSW - III	4.75	1.86	16.77	22.27	19.14
ECSW - IV	6.44	4.35	**31.01**	**32.39**	**30.75**

The effective speed-up obtained using $c_{on} = c_{off} = 0.5$ is on the conservative side because one expects the offline costs to be useful for many more online runs. Giving equal weightage to them would mean offline cost are effective being repeated for every response simulation. This weightage chosen to merely to give an overview of the worst case scenario. Actual weights are a subject of application at hand and would require user based decisions depending upon the method of reduction, loading etc. Still, Table 6.4 shows that hyper-reduction based on quadratic manifold are leading the chart. It can also be noted that Linear Manifold Tensorial approaches with selected MDs, also return high effective speed-ups. However, it should be kept in mind that MD selection is sensitive to applied loading. A change in MD basis would result in recomputation of the tensors, thus repeating the huge offline costs. Details conlusions about the comparison of the techniques follow in the next chapter.

7 Conclusions

The field of Model order reduction is one of ongoing research. In this work, the emphasis was laid on a general class of techniques in MOR namely the projection based reduction techniques. In the context of thin walled structural dynamics characterized by Von Kármán kinematics, the main focus of the research was to study/develop/propose and implement methods for Model order reduction which essentially don't require a full non-linear solution run. With this focus, the work branched into two sub-focal areas. The first of these concluded with the proposal of linear and non-linear mapping using Modal derivatives to reduce the number of unknowns in a reduced model. Due to polynomial nature of non-linearities, tensor based approaches were also proposed and implemented for these mappings. The second branch dealt with hyper-reduction using ECSW and in the context of this research, a very cheap and effective method to produce training vectors for ECSW was proposed. Both these branches have showed their own advantages, disadvantages and have addressed the two sub-research questions mentioned in Section 1.2 respectively. The conclusions-cum-recommendations have been split accordingly as follows.

I

Linear Manifold vs the Quadratic Manifold : The Linear Manifold consisting of a few vibration modes and Modal derivatives proved to be an effective basis for reduction in the current context. The quadratically increasing size of basis led to : 1) a non-linear mapping which inherently reduces the number of unknowns in the reduced equations by exploiting the relation of modal derivatives to the Taylor expansion.

- **Accuracy:** It is easy to see that the amplitudes connected to the MDs in the linear manifold are unknowns in the corresponding reduced equations and thus are free to choose any value, where as in the quadratic manifold they are inherently constrained by the amplitudes of VMs. Thus, the former gives more freedom for reduction and provides a lower bound for error during the latter (Section 3.4.3). This is also shown by the results in all the examples in the Chapter 5 whereby the GRE estimates of the approaches using quadratic manifold are always greater than or equal to that during the ($GRE_{QM} \geq GRE_{LM}$) .
- **Speed:** Even after selection of important MDs, it is easy to see that the number of unknowns is greater in Linear Manifold than the quadratic manifold. Thus the reduced system is smaller in case of a quadratic manifold. Also, as seen in the *regular* implementation of these techniques, the quadratic manifold as always faster than the linear manifold (Figures 6.1 to 6.4 Table 6.1) .

MD Selection: The search for selection criteria of MDs to reduce the Linear manifold basis size, resulted in two possible candidates (MMI, MVW Section 3.3.2). Both of them give results of comparable accuracy. However, the MVW technique did not perform better than MMI on any of the examples in terms accuracy as well as offline costs (Tables 5.1 to 5.5 and 6.3). It should be noted that all the selection techniques provided a way to rank the MDs in the order of relative importance, but these techniques did not indicate how many of these ranked MDs should be used to accurately reproduce the solution obtained using all the MDs. Thus, there is scope for further work along this course. As a rule of thumb, a total of $n_{MD} = m$ MDs was chosen in this work in a linear manifold basis containing m VMs. This was done for fair comparison and keeping the basis size linear with m.

Tensorial approaches: Tensor based approaches were implemented for both linear and quadratic manifold. The main advantages of these approaches was the offline evaluation of nonlinearity resulting in a reduced model whose online solution time is independent of the size of the full physical system. Following general conclusions can be made from the work.

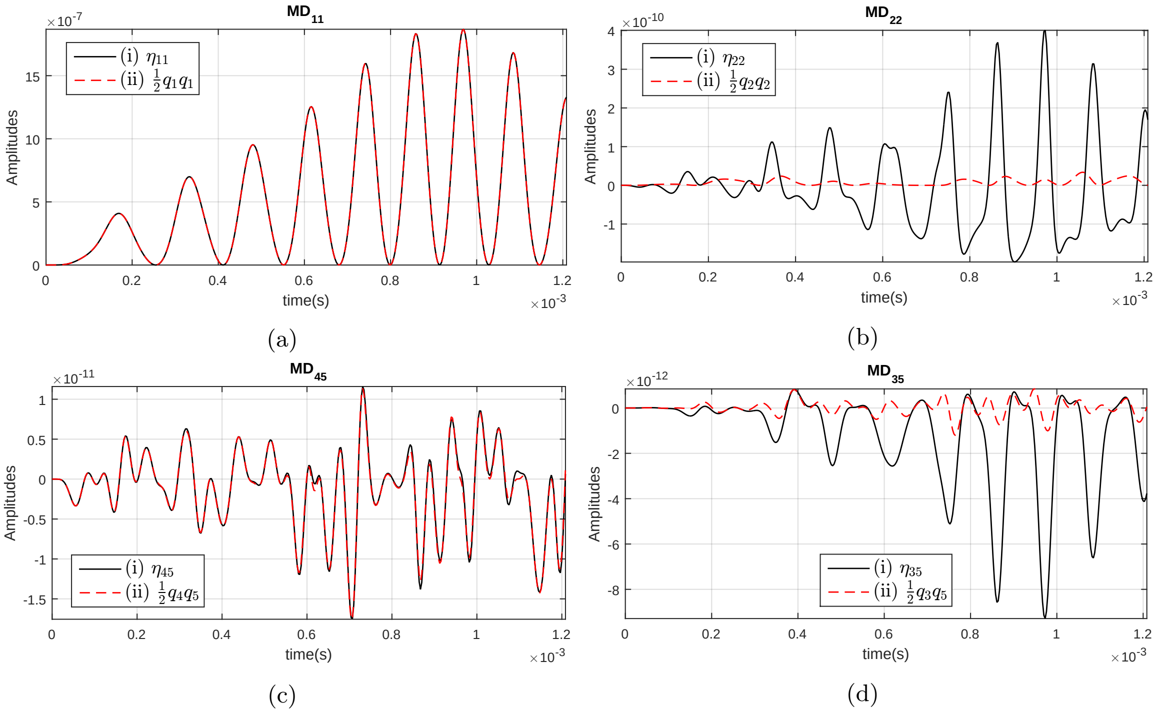
- The effect of neglecting higher order tensors (${}^{7}_{8}\boldsymbol{\mathcal{K}}$ and ${}^{6}_{7}\boldsymbol{\mathcal{K}}$) had a marginal effect on the accuracy of solution (Tables 5.1 to 5.5). It's interesting that these tensors are the most expensive in terms of computation time (both offline and online) as well as memory, and neglecting them makes the QM tensorial approach even more efficient.
- It was also shown that beyond a certain m, neglecting these tensors significantly boosts the speed up while plummeting the memory usage to the extent that they become faster and more affordable for the available memory, than the linear Manifold(including all MDs) (Figures 6.1 to 6.4).
- It was observed that an increase in structure curvature leads to a more pronounced deterioration of the solution accuracy when higher order tensors are neglected. Since neglecting the higher order tensors is of so much interest, it is thus recommended to establish analytical bounds for the structure curvature under which it is safe to neglect the higher order tensors.
- A tensorial approach involves significant offline costs but this cost can be split between a number of processors working in parallel on different parts of the mesh making the approach very efficient.
- It is expected that high offline costs would be required to construct a tensor based ROM which can be used for a variety of load scenarios (e.g. QM Tensorial or LM tensorial with all MDs). Thus, although the Linear manifold tensorial approach with selected few MDs returns extremely large online speed-ups, the MD selection is sensitive to load cases and a change in basis would involve re-computation of tensors, which might not be desirable.

Convergence of QM regular approach: For a Linear Manifold, the *regular* and *tensorial* approaches are identical as far as the solution convergence/accuracy is concerned. However in the QM case , the Jacobian used during time integration of reduced equations differ between the two approaches, with the former being far from accurate. This leads to convergence issues during the iterative solution of nonlinear algebraic equations while performing time integration. Sometimes the system takes higher number of iterations to converge to the required tolerance and sometimes it might not even converge.

Convergence of QM Tensorial approach: Using the tensors, the Jacobian for the N-R iterations during time integration can be constructed in an accurate manner. This leads to convergence in all the tested examples. Whether this can be generally claimed is recommended as work for future research.

It was sometimes observed that after following reference solution up to a certain point in time, the system converged to a different solution using the QM Tensorial approach. It is also remarkable that the solution using linear manifold (including all MDs) quickly converged to the reference solution in such cases. This could be linked to the fact that in such cases, some of the MDs don't end up following the amplitudes constrained (by the quadratic mapping) to that of the corresponding VMs (as depicted in Figure 7.1). This shows that sometimes, the quadratic mapping assumption indeed is not able to capture the nonlinearities as effectively as the Linear manifold containing MDs.

As a further step towards model order reduction in this context, if the MDs which follow the QM assumption can somehow be distinguished, then a hybrid approach between linear and quadratic manifold could be established. Such MDs could then be a part of the nonlinear mapping and other important MDs should then be included in the linear mapping.



(a)

(b)

(c)

(d)

Figure 7.1: shows the how some of the MD amplitudes which are constrained by the quadratic manifold (to the VM amplitudes) are not able to capture the nonlinear response in a typical run. In all figures, (i) η_{jk} shows the amplitude of the MD_{jk} and q_j corresponds the amplitude of VM_j in a Linear Manifold run. Thus (ii) $\frac{1}{2}q_jq_k$ represents the amplitude that a quadratic manifold run should return i.e. in an ideal situation where the quadratic mapping is able to capture the response effectively, $\eta_{jk} = \frac{1}{2}q_jq_k$. The two figures in left show that the quadratic mapping is successful for the corresponding MDs, whereas the two on the right show that the quadratic mapping has constrained the MD amplitudes to a much smaller magnitude than desirable.

II

Hyper-Reduction: The ECSW [5] was established as an ideal choice for hyper-reduction of finite element based applications. It was implemented and tested successfully. The results using snapshots of reference nonlinear solution as training vectors, and a POD basis for reduction were exceptional but the need for a full solution run made it an undesirable choice.

Tensors vs. ECSW: Tensorial approaches are limited to systems with polynomial nonlinearities and as such involve large offline costs. These approaches pre-compute the nonlinearities in the reduced equations exactly, thereby resulting in huge online speed-ups. The ECSW on the other hand is a generic tool for MOR of finite element based nonlinear dynamics applications and its structure preserving and stability properties make it a very attractive choice in the context of this work. Apart from the training set generation, the offline costs are minimal and the online speed-ups are large since nonlinearity is very cheaply and effectively approximated by calculating it over a set predetermined elements. The online speed-up factors though high, were still less than that of the tensorial approaches (Tables 5.3 to 5.5). However, the ECSW seemed to be a winner when offline costs were taken into account to calculate an effective speed-up (Table 6.4). Thus, hyper-reduction using ESCW can be seen in some sense as a balance between the *low offline-high online* costs of the *regular* apparoaches and *high offline-low online costs* of the *tensorial* approaches.

To take a step further in that direction, the following work can be recommended. If localised nonlinearities can be identified in a structure then the nonlinear tensors could be evaluated only for elements corresponding to such regions. This could drastically reduce the offline cost (since this cost is directly proportional to the number of elements on which tensors are evaluated) while maintaining the huge online speed-ups of the tensorial approaches.

The range of τ: During offline set up of the ECSW, the τ was seen as a tolerance while solving the sparse NNLS problem 4.14. It was also seen as a tool to control the sparsity of the solution vector $\boldsymbol{\xi}$. A range of 0.1-0.01 was recommended for τ in [5]. Though for $\tau = 0.01$, the results were acceptably accurate in all the tested cases, it was sometimes observed (Figure 5.16) that setting a lower value of τ resulted in much better results without any significant loss of sparsity.

QM for generation of training sets: One of the primary aim of this work was to avoid a full solution run for hyper-reduction. ECSW needs training vectors which include the essential features of nonlinear behavior. As established before, the MDs provide an effective tool to capture the second order nonlinear behaviour in the current context. The linear behavior is cheaply captured by a modal superposition run. The use of quadratic mapping in combination with a linear modal superposition run was proposed in this work (ECSW-III and IV) which turned out to be a very cheap and effective way to generate training sets. It was found to be successful in all the tested examples (Tables 5.1 to 5.5). The range of applicability of this proposal is not yet established and should be a subject for future research.

Element sampling for ECSW: It was observed that the number of elements sampled during sNNLS for ECSW-II,III were much less compared to that in ECSW-I. The reason for this is not clearly understood, however the clear difference between these approaches is the basis and training vectors.

Technique	Training vectors	Reduction basis
ECSW-I	Full solution snapshots	POD
ECSW-II	LM (All MDs) reduced solution snapshots	LM Basis(All MDs)
ECSW-III	Quadratic mapping of Linear modal solution snapshots	LM Basis(All MDs)
ECSW-IV	Quadratic mapping of Linear modal solution snapshots	LM Basis(MMI selected MDs)

While ECSW-I uses full solution snapshots as training vectors and POD basis as a reduction basis, it ends up constructing an element sampling which produces the least error results among all the reduction techniques. However, somehow it is also the slowest among them due to higher number of selected elements. ECSW-II and III use an LM basis with all MDs for reduction and produce a smaller element sampling. The ECSW-IV on the other hand use a basis of smaller size than all the other three and somehow produces an element sampling of almost half the size of that in ECSW-III and IV and leads to spectacular speed-ups with similar (or even better) accuracy. These observations indeed provide insight for further research to deliver more concrete claims about the applications of ECSW in this context.

A

Appendix

A.1. Element level implementation of ${}^2_3\mathbf{K}$ and ${}^3_4\mathbf{K}$

The tensors ${}^2_3\mathbf{K}$ and ${}^3_4\mathbf{K}$ used in Equation (3.88) can be assembled from element level contributions as follows.

The element level nonlinear internal force for element e is developed as shown in [14].

$$\begin{aligned}
\mathbf{f}_e &= A_e[\mathbf{B}_L + \mathbf{B}_{NL}(\mathbf{a}_e)]^T\mathbf{D}[\mathbf{B}_L + \frac{1}{2}\mathbf{B}_{NL}(\mathbf{a}_e)]\mathbf{a}_e && \text{(A.1)}\\
&= \underbrace{A_e\mathbf{B}_L^T\mathbf{D}\mathbf{B}_L}_{\mathbf{K}_L^e}\mathbf{a}_e + \underbrace{\frac{1}{2}A_e\mathbf{B}_L\mathbf{D}\mathbf{B}_{NL}(\mathbf{a}_e)\mathbf{a}_e + A_e\mathbf{B}_{NL}(\mathbf{a}_e)\mathbf{D}\mathbf{B}_L\mathbf{a}_e}_{\text{quadratic terms}} + \underbrace{A_e\mathbf{B}_{NL}(\mathbf{a}_e)\mathbf{D}\mathbf{B}_{NL}(\mathbf{a}_e)\mathbf{a}_e}_{\text{cubic term}} && \text{(A.2)}
\end{aligned}$$

where,

$$\mathbf{D} = \frac{Yt}{(1-\nu^2)}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & (1-\nu)/2\end{bmatrix}, \tag{A.3}$$

is the material stiffness matrix (Y being the Young's modulus, t the element thickness and ν the poisson's ratio), $\mathbf{B}_L \in \mathbb{R}^{3\times18}$ is based on the formulation in [29] and used in [14],

$$\mathbf{B}_{NL}(\mathbf{a}_e) = \begin{bmatrix}\mathbf{a}_e^T\mathbf{K}_{xx}\\ \mathbf{a}_e^T\mathbf{K}_{yy}\\ \mathbf{a}_e^T\mathbf{K}_{xy}\end{bmatrix} \in \mathbb{R}^{3\times18}, \tag{A.4}$$

($\mathbf{K}_{xx}$, $\mathbf{K}_{yy}$, $\mathbf{K}_{xy} \in \mathbb{R}^{18\times18}$ are constant matrices which the contain shape function derivatives for bending and membrane degrees of freedoms, refer to [14] for their expressions), and A_e corresponds to the surface of area of the triangular shell element e.

With these definitions, the element level contribution of the quadratic tensor ${}^2_3\mathbf{K}$ is denoted as ${}^2_3\mathbf{k}^e \in \mathbb{R}^{18\times18\times18}$ and can be deduced from Equation (A.2). Its formulation is shown in MATLAB style as follows.

$$\begin{aligned}
{}^2_3\mathbf{k}^e(i,:,:) = \frac{1}{2}(&\mathbf{C}(i,1)\mathbf{K}_{xx} + \mathbf{C}(i,2)\mathbf{K}_{yy} + \mathbf{C}(i,3)\mathbf{K}_{xy}) +\\
&(\mathbf{K}_{xx}(:,i)\mathbf{F}(1,:) + \mathbf{K}_{yy}(:,i)\mathbf{F}(2,:) + \mathbf{K}_{xy}(:,i)\mathbf{F}(3,:)) \qquad \forall i \in \{1,2,\dots,18\},
\end{aligned}$$

where,

$$\begin{aligned}
\mathbf{C} &:= A_e\mathbf{B}_L^T\mathbf{D},\\
\mathbf{F} &:= A_e\mathbf{D}\mathbf{B}_L = \mathbf{C}^T.
\end{aligned}$$

On similar lines, the element level contribution of the cubic tensor ${}^3_4\mathbf{K}$ is denoted as ${}^3_4\mathbf{k}^e \in \mathbb{R}^{18\times18\times18\times18}$ and can also be deduced and its MATLAB style formulation is as follows.

$$\begin{aligned}{}^3_4\mathbf{k}^e(i,j,:,:) = \frac{1}{2}A_e[&(\mathbf{K}_{xx}(i,j)\mathbf{D}(1,1) + \mathbf{K}_{yy}(i,j)\mathbf{D}(2,1) + \mathbf{K}_{xy}(i,j)\mathbf{D}(3,1))\mathbf{K}_{xx}+\\ &(\mathbf{K}_{xx}(i,j)\mathbf{D}(1,2) + \mathbf{K}_{yy}(i,j)\mathbf{D}(2,2) + \mathbf{K}_{xy}(i,j)\mathbf{D}(3,2))\mathbf{K}_{yy}+\\ &(\mathbf{K}_{xx}(i,j)\mathbf{D}(1,3) + \mathbf{K}_{yy}(i,j)\mathbf{D}(2,3) + \mathbf{K}_{xy}(i,j)\mathbf{D}(3,3))\mathbf{K}_{xy}]. \qquad \forall i,j \in 1,2,\ldots,18\end{aligned}$$

Rotation to global coordinates: Note that the element level quantities ${}^3_4\mathbf{k}^e$ and ${}^3_4\mathbf{k}^e$ are in local isoparametric coordinates and need to be rotated in to the global coordinates to come up with the global contributions (${}^3_4\mathbf{K}^e$ and ${}^3_4\mathbf{K}^e$ respectively) of the element level tensors. This is done as follows.

Suppose the transformation matrix $\mathbf{T}_e \in \mathbb{R}^{18\times18}$ is used to convert a displacement vector $\mathbf{a}_e \in \mathbb{R}^{18}$ in local isoparametric coordinates of element e to global coordinates such that, K2global = ttt(ttt(ttt(Q,K2,1,1),Q,2,1),Q,2,1); K3global = ttt(ttt(ttt(ttt(Q,K3,1,1),Q,2,1),Q,2,1),Q,2,1);

$$\mathbf{a}_e = \mathbf{T}_e^T\mathbf{u}_e \tag{A.5}$$

(Note that here transformation only means rotation and thus $\mathbf{T}_e$ is an orthogonal matrix), then it is easy to see that the rotated version of tensors can be obtained as

$${}^2_3\mathbf{K}^e = (((\mathbf{T}_e^T \cdot {}^2_3\mathbf{k}^e) \cdot_{21} \mathbf{T}_e) \cdot_{21} \mathbf{T}_e), \tag{A.6}$$

$${}^3_4\mathbf{K}^e = ((((\mathbf{T}_e^T \cdot {}^3_4\mathbf{k}^e) \cdot_{21} \mathbf{T}_e) \cdot_{21} \mathbf{T}_e) \cdot_{21} \mathbf{T}_e) \tag{A.7}$$

(See Section 1.3 for "dot" product notation).

Bibliography

[1] E. Ventsel and T. Krauthammer, *Thin Plates and Shells*, 0-8247-0575-0 (Marcel Dekker, Inc., 2001).

[2] M. Geradin and D. Rixen, *Mechanical Vibrations: Theory and Application to Structural Dynamics, 2nd Edition*, 0-471-97524-9 (Wiley, 1997).

[3] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation,* SIAM Journal on Scientific Computing , 2737 (2010).

[4] P. Tiso and D. J. Rixen, *Discrete empirical interpolation method for finite element structural dynamics,* in *Topics in Nonlinear Dynamics*, Vol. 1 (Springer New York, 2013).

[5] C. Farhat, P. Avery, T. Chapman, and J. Cortial, *Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency,* International Journal for Numerical Methods in Engineering **98**, 625 (2014).

[6] D. Ryckelynck, *A priori hyperreduction method: an adaptive approach,* Journal of Computational Physics **202**, 346 (2005).

[7] S. R. Idelsohn and A. Cardona, *A load-dependent basis for reduced nonlinear structural dynamics,* Computers & Structures **20**, 203 (1985).

[8] S. R. Idelsohn and A. Cardona, *A reduction method for nonlinear structural dynamic analysis,* Computer Methods in Applied Mechanics and Engineering **49**, 253 (1985).

[9] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, 1st ed. (Springer Publishing Company, Incorporated, 2007).

[10] C. Gu, *Model Order Reduction of Nonlinear Dynamical Systems*, Ph.D. thesis, EECS Department, University of California, Berkeley (2012).

[11] B. W. Bader, T. G. Kolda, *et al.*, *Matlab tensor toolbox version 2.6,* Available online (2015).

[12] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, 978-0-471-82319-3 (John Wiley & Sons, Ltd., 2000).

[13] R. de Borst, M. A. Crisfield, J. J. Remmers, and C. V. Verhoosel, *Non-Linear Finite Element Analysis of Solids and Structures*, 1st ed., 9781118375938 (John Wiley & Sons, Ltd., 2012).

[14] P. Tiso, *Finite element based reduction methods for static and dynamic analysis of thin-walled structures*, Ph.D. thesis, Delft University of Technology (2006).

[15] D. Allman, *A simple cubic displacement element for plate bending,* International Journal for Numerical Methods in Engineering **10**, 263 (1976).

[16] D. Allman, *Evaluation of the constant strain triangle with drilling rotations,* Internation Journal for Numerical Methods in Engineering **26**, 2645 (1988).

[17] V. Siddhi, *A generalized approach for calculation of the eigenvector ensitivity for various eigenvector normalizations*, Master's thesis, University of Missouri - Columbia (2005).

[18] R. B. Nelson, *Simplified calculation of eigenvector derivatives,* AIAA Journal **14**, 1201 (1976).

[19] P. Tiso, *Optimal second order reduction basis selection for nonlinear transient analysis,* in *Modal Analysis Topics, Volume 3*, Conference Proceedings of the Society for Experimental Mechanics Series, edited by T. Proulx (Springer New York, 2011) pp. 27–39.

[20] J. Barbič and D. L. James, *Real-time subspace integration for st. venant-kirchhoff deformable models,* ACM Trans. Graph. **24**, 982 (2005).

[21] C. Farhat, T. Chapman, and P. Avery, *Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models,* International Journal for Numerical Methods in Engineering **102**, 1077 (2015).

[22] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems* (Prentice-Hall, 1974).

[23] R. Peharz and F. Pernkopf, *Sparse nonnegative matrix factorization with l0-constraints,* Neurocomputing **80**, 38 (2012), special Issue on Machine Learning for Signal Processing 2010.

[24] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions,* Journal of Symbolic Computation **9**, 251 (1990), computational algebraic complexity editorial.

[25] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)* (Johns Hopkins University Press, Baltimore, MD, USA, 1996).

[26] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins studies in the mathematical sciences (Johns Hopkins University Press, 2013).

[27] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. (The MIT Press, 2009).

[28] Y. Luo and R. Duraiswami, *Efficient parallel nonnegative least squares on multicore architectures,* SIAM J. Sci. Comput. **33**, 2848 (2011).

[29] C. A. Felippa, *A study of optimal membrane triangles with drilling freedoms,* Technical Report, College of Engineering, University of Colarado (2003).