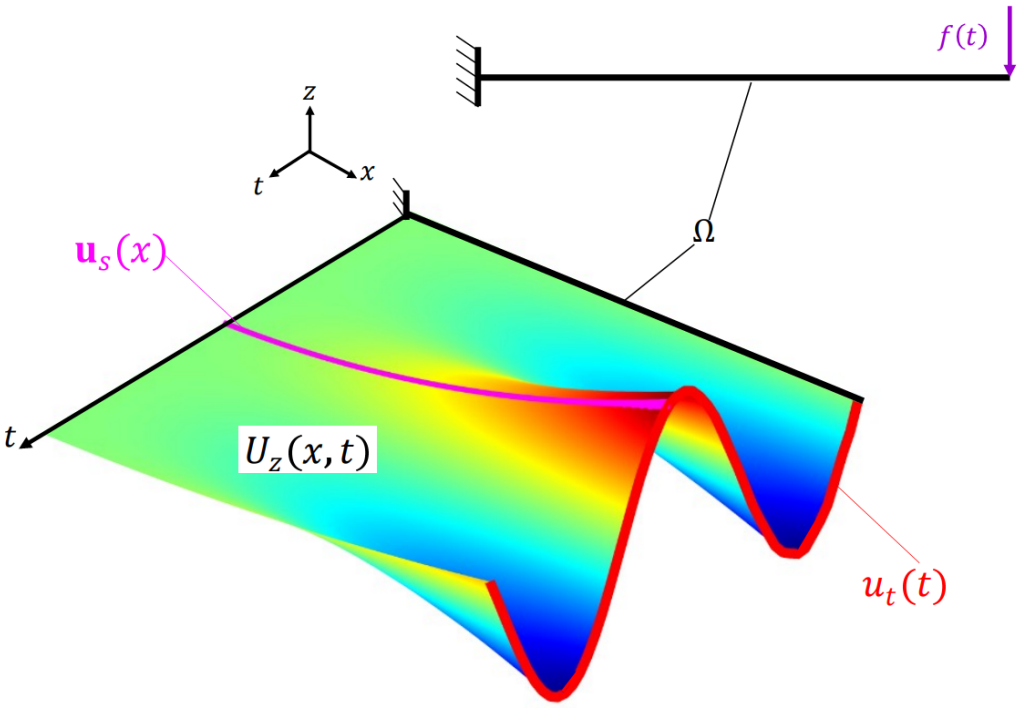
National Technical University of Athens
School of Civil Engineering
Postgraduate Program: Analysis and Design of Earthquake Resistant Structures
Institute of Structural Analysis and Antiseismic Research

Investigating Proper Generalized Decompositions with applications to Structural Dynamics



Postgraduate Thesis

George M. Pissas
Civil Engineer BSc, MEng

Supervisor: Savvas P. Triantafyllou, Assistant Professor NTUA
Athens, March 2023

Acknowledgements

First, I would like to express my most sincere gratitude to my supervisor, Assistant Professor Savvas Triantafyllou, for his continued guidance throughout the course of this thesis. His constant support and outstanding scholarship have been a motivation throughout my student years. Many thanks to my former classmates and now colleagues for the fun times after class as well as the late hour constructive talks we've had over the years. Special thanks are due to my companion Anna, for always standing by my side and supporting me. Most of all, I would like to thank my parents, Marianna and Michael and my brother Theodore. Whatever I have accomplished so far would not be possible without their selfless support, encouragement and guidance throughout my life.

Περίληψη

Οι σύγχρονοι κώδικες για την αποτίμηση των κατασκευών, παρέχουν προτάσεις και κατευθυντήριες γραμμές για τη χρήση μεθόδων επίλυσης χρονοϊστορίας στα πλαίσια των αναλύσεων. Ωστόσο, η προσομοίωση μεγάλων κατασκευών στο πεδίο του χρόνου μπορεί να καταστεί υπολογιστικά ασύμφορη. Επομένως, δημιουργείται μια ανάγκη για νέες, ταχείες και υψηλής πιστότητας μεθόδους επίλυσης. Οι μέθοδοι Μείωσης Τάξης Προσομοιώματος (MOR), οι οποίες προσεγγίζουν μεγάλα υπολογιστικά μοντέλα με υποκατάστατα χαμηλότερης τάξης, αποτελούν μια ελκυστική επιλογή προς ικανοποίηση αυτής της ανάγκης. Οι τεχνικές Μείωσης Τάξης Προσομοιώματος χωρίζονται στις κατηγορίας *εκ των υστέρων* και *εκ των προτέρων*. Η πρώτη κατηγορία, αφορά σε τεχνικές, που κατασκευάζουν το προσομοίωμα μειωμένης τάξης (ROM) μετά τη διενέργεια ορισμένων επιλύσεων του συστήματος και προσφέρονται ιδιαίτερα για την εφαρμογή σε προσομοιώματα που εξαρτώνται από λίγες παραμέτρους, αλλά έχουν διαθέσιμο μεγάλο πλήθος παρατηρήσεων. Η δεύτερη κατηγορία αφορά σε μεθόδους οι οποίες βασίζονται στη φυσική που διέπει το εκάστοτε πρόβλημα και συνθέτουν το προσομοίωμα μειωμένης τάξης χωρίς την ανάγκη προ-υπολογισμένων λύσεων του συστήματος. Αντιπροσωπευτικό παράδειγμα της δεύτερης κατηγορίας αποτελεί η μέθοδος του Ιδιο-Γενικευμένου Διαχωρισμού (Proper Generalized Decomposition - PGD), η οποία και αναλύεται στην παρούσα μεταπτυχιακή εργασία.

Η μέθοδος PGD βασίζεται πάνω στην έννοια των αναπαραστάσεων χωριζωμένων μεταβλητών. Το άγνωστο πεδίο μετατοπίσεων, αναζητείται στη μορφή ενός αθροίσματος δυαδικών/τανυστικών γινομένων χωρικών και χρονικών διανυσμάτων εμπλούτισης, επονομαζόμενα PGD-μορφές και μορφώνεται μέσω επαναλήψεων εμπλούτισης. Στα πλαίσια της δυναμικής των κατασκευών, η PGD μπορεί να χρησιμοποιηθεί ως ένας μη-επαυξητικός επιλύτης σε ολόκληρο το χωροχρονικό πεδίο στο οποίο ορίζεται το εκάστοτε πρόβλημα. Ο επαυξητικός φορμαλισμός της μεθόδου Newmark για τη χρονική ολοκλήρωση των εξισώσεων κίνησης, προβάλλεται σε μια ισοδύναμη μη-επαυξητική του μορφή. Έτσι, το δυναμικό σύστημα αντιπροσωπεύεται από μία μόνο αλγεβρική χωροχρονική εξίσωση. Κατ' αυτόν τον τρόπο οι απαιτήσεις σε μνήμη και αποθηκευτικό χώρο, καθώς και οι υπολογιστικοί χρόνοι μπορούν να μειωθούν. Η ακρίβεια, η αποδοτικότητα, καθώς και τα όρια της προσέγγισης PGD-Newmark αναδεικνύονται μέσα από αριθμητικά παραδείγματα. Τέλος, εξάγονται συμπεράσματα σχετικά με την εξέλιξη των χωρικών και χρονικών διανυσμάτων εμπλούτισης κατά τη σύγκλιση, τη συνολική απόδοση της μεθόδου, καθώς και την επιρροή διαφόρων αλγοριθμικών παραμέτρων στη συμπεριφορά σύγκλισης.

Λέξεις Κλειδιά

Προσομοίωμα Μειωμένης Τάξης, Ιδιο-Γενικευμένος Διαχωρισμός, Σεισμική Μηχανική, Δυναμική των Κατασκευών

Abstract

Modern codes for structural assessment provide recommendations and guidelines on the use of time-history solution procedures as a feasible analysis route. Unfortunately, analyzing large scale structures in the time domain is computationally taxing. Hence, a requirement is identified for rapid, yet high fidelity, solution procedures. Model Order Reduction (MOR), i.e., the approximation of large computational models with significantly smaller ones, is an attractive option towards this objective. MOR methods are divided into *a-posteriori* and *a-priori* methods. The former includes techniques that build the Reduced Order Model (ROM) after computing some solutions of the system and is especially useful when the model has a relatively small number of parameters but many observations. The latter methods rely solely on the physics that govern the problem and construct the ROM without the need of precomputing any solution. A representative example of the *a-priori* methods is the so called Proper Generalized Decomposition (PGD), which is considered in this thesis.

The PGD builds on the concept of *separated representations*. The unknown displacement field is sought in the form of a sum of dyadic products of spatial and temporal vectors, called PGD modes and is iteratively assembled by enrichments. Within the framework of structural dynamics, the PGD can be used as a non-incremental solver in the entire space-time domain. The incremental Newmark time integration scheme is cast into its space-time equivalent and the dynamic system is represented by only one algebraic space-time equation. In this manner, memory/storage requirements as well as computational runtimes could be reduced. The accuracy, efficiency as well as the limits of the PGD-Newmark space-time approach are demonstrated via numerical benchmarks. Several conclusions are drawn regarding the evolution of the spatial and temporal enrichment vectors during convergence, the overall performance of the approach as well as the influence of some algorithmic implementation parameters on the convergence behavior.

Keywords

Reduced Order Model, Proper Generalized Decomposition (PGD), Earthquake Engineering, Structural Dynamics

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Motivation and Problem Statement

Although the computer speed and memory capacity continue to double every 18 months (Moore's law), the phenomenon that the demand of computer storage and speed will always exceed existing capabilities has been consistently demonstrated in finite element analysis during the past half century, as brilliantly stated by Qu in [7]. The remarkable progress in computer capabilities has rendered the resolution of problems involving millions of unknowns an ordinary task. This progress, combined with decades of research in numerical analysis, have yielded an extensive range of computational tools that can predict the behavior of complex structures and mechanical systems.

However, many scientific and engineering problems, to this day, remain intractable either because of their numerical complexity, or because of requirements (e.g. memory, time) that make them unaffordable, even for today's advancement in computational technologies. Some examples of demanding elastodynamic problems. For example, haptic feedback applications for surgical simulators [8], micro/nano-electromechanical systems (MEMS/NEMS), dynamic data-driven applications [9] and optimization frameworks in industrial and structural design, are but a few very demanding structural dynamics problems.

A key issue in structural dynamics is the evaluation of the response time history of engineering structures subjected to transient excitations. To this end, there exist many conventional methods, including analytic strategies (for linear systems) and incremental (step-by-step) time integration schemes [1]. Models in structural dynamics, consist of a system of governing differential equations describing the system, supplemented by boundary conditions, which are usually discretized in space with the Finite Element Method (FEM), leading to an Ordinary Differential Equation (ODE) that is to be solved in the time domain. Next, implicit or explicit time integration schemes are used for the temporal discretization of the evolution problem. Employing a fine spatial mesh for accuracy and choosing a small time step to avoid instability issues (especially for explicit schemes), results in a system of a very large number of degrees of freedom, inevitably increasing the demand for memory, time and energy resources. Conventional time integration schemes (e.g. CDM, Newmark, HHT) differ in the way the temporal derivatives of the displacement field are approximated. Their common ground, however, is a resulting recursive formula that is solved via an incremental solution procedure.

An alternative, non-incremental approach, would be to cast the incremental solution procedure over the entire space-time domain. Then, the resulting problem would amount to the solution of a linear system with a dimension equal to the product of the spatial and temporal dimensions of the initial problem. The solution of this linear system, would yield the entire space-time field. In [10], finite elements have been applied simultaneously in space and time. In [11], Hughes et. al introduced the application of space-time finite elements in elastodynamics and structural dynamics. Later, space-time finite elements were employed for elastodynamics analyses on structured and unstructured meshes in [12]. The space-time FEM approach has also been applied in many different engineering problems

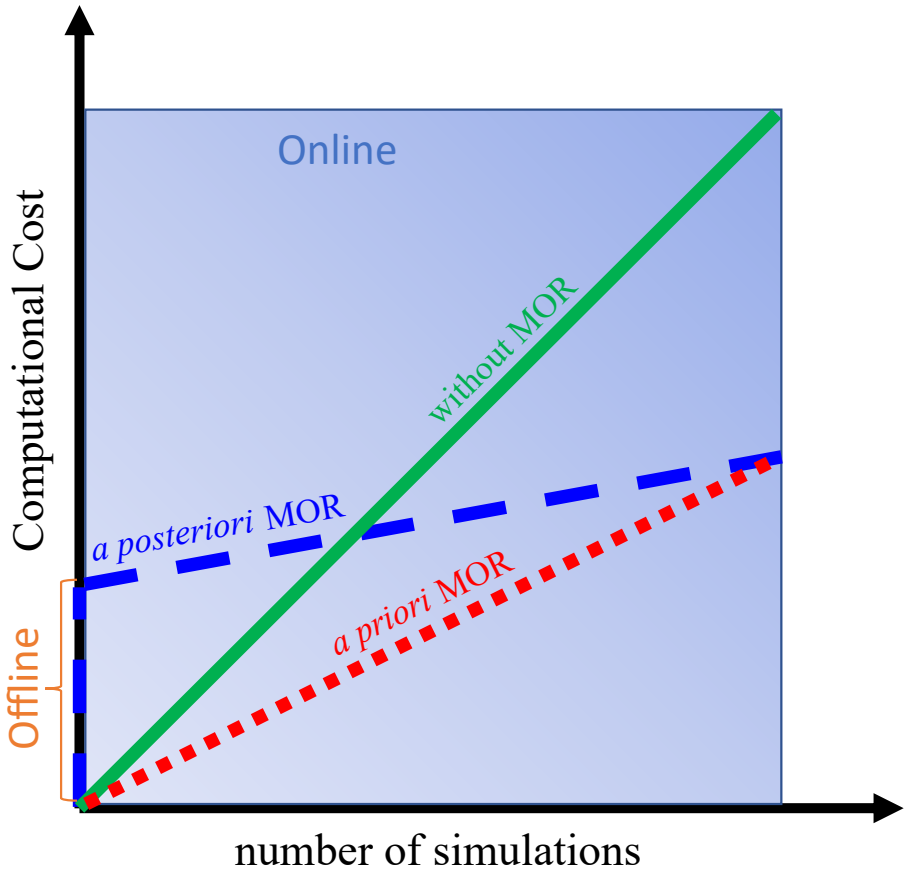


Figure 1.1: Analysis computational cost with and without MOR

such as contact mechanics [13], multiscale modelling [14] and fluid-structure interaction [15], among others.

Despite the many computational tools available, the need for novel numerical strategies, that can potentially decrease computational costs and/or memory requirements by exploiting today's computational resources in a more efficient way, is always an important task. There is an evident inclination/interest towards low dimensional descriptions/surrogates that can approximate the full order model as accurately as possible.

1.2 Model Order Reduction (MOR)

Model Order Reduction (MOR), i.e. the approximation of large computational models with significantly smaller ones, constitutes a promising tool towards this task. It is because of the above-mentioned computational bottlenecks, that structural dynamics has been one of the first fields of engineering practice where MOR has been employed. The classic method of Modal Superposition ([27]) can be viewed as a sort of MOR. In [16], system eigenmodes are used in order to reduce the number of degrees of freedom in nonlinear dynamics applications. To find sufficiently accurate solutions with significantly lower computational costs, several approximation methods and MOR strategies have been proposed in the literature. In general, they can be divided into methods that require offline computations to achieve online speedups, termed *a posteriori* and methods that not need any offline step, termed *a priori*. Fig.1.1 illustrates the online/offline cost for different MOR approaches.

The *a posteriori* methods, also referred to as *projection-based* or *data-driven*, build the Reduced

Order Model (ROM) using computed snapshots or empirical realizations of the system. In essence, projection-based MOR techniques rely on the analysis of system snapshots to define a lower dimension vector space and the projection of the governing equations onto this low dimension vector space. The original degrees of freedom of the model (e.g. FEM nodal displacements) are swapped with an as much as possible reduced number of generalized coordinates that adequately describe the eigenmodes the contain the most "energy" of the autocorrelation matrix of the snapshots. The most prominent *a posteriori* approach is the Proper Orthogonal Decomposition (POD). The POD technique has been developed and re-discovered in many branches of science under different names such as Principal Component Analysis and Karhunen-Loève Decomposition in signal processing, but also Hotelling transform, Eckart-Young theorem and Singular Value Decomposition (SVD) in linear algebra and Empirical Modal Analysis in structural dynamics (see [17] for more). In structural dynamics, the POD is used to determine the dominant motion patterns for an optimal low-order description of linear and nonlinear systems ([28], [26]). The transformation into the low order subspace depends on the derived POD modes obtained from the system snapshots. Its performance, for the case of transient excitations mainly depends on the snapshot time period [30], [29].

A priori methods build the ROM without the need of precomputing any solution of the problem. These techniques are based on knowledge of the governing equations. A representative example of *a priori* MOR is the Proper Generalized Decomposition [22]. The key point of the PGD is the concept of *separated representations*. The problem solution which is usually a multivariate function, is sought in the form of a sum of products of simpler, univariate functions.

As stated in [17], the origin of the PGD can be traced back to the *radial loading* step within the LArge Time INcrement (LATIN) method proposed by Ladevèze [31], as a space-time separated representation in non-incremental structural mechanics solvers. The term was first coined by Chinesta et al. and was developed in [32] and [33], where a novel solution method for non-Newtonian fluid models in high-dimensional phase spaces has been developed. Their work was soon identified to be a generalization of Ladevèze's work. Subsequently the PGD has been introduced as a solution/MOR method for many different scientific areas such as the solution of Helmholtz equations [34], magnetostatics [35], Boltzmann and Fokker-Planck equations [36], steady flows [43], contact problems [37], multiscale analyses [39], [40], shape optimization [38], virtual surgery simulations [41] and even real-time simulations on e-books in the context of augmented learning [42]. Lately, it has also found application in lattice structure problems [44] and brittle fracture with random field parameters [45].

In the field of structural dynamics, it has seen a wide variety of uses under different settings. In [46], the PGD is used as an efficient black-box integrator for solid dynamics, that computes a multi-parametric solution field, considering analyses parameters (e.g. initial conditions) as extra coordinates. In [47], the PGD is used for the space-time resolution of elastoplastic problems. The work of Boucinha et al. [19] was the first to propose a tensorial formulation of various time integration schemes (including the Newmark scheme) within a non-incremental solution procedure. The PGD allows one to consider the entire problem simultaneously in space and time, while using conventional spatial discretization strategies (e.g. FEM). Bamer et al. applied the Newmark tensorial PGD formulation to linear structural dynamics [21]. Shirafkan et al. implemented the PGD space-time approach to the quasistatic analysis of elastoplastic structures and physically interpreted the spatial and temporal modes, for which the coined the term PGD modes.

One key benefit of a separated representation of the solution field, is that it drastically reduces computer memory and storage requirements, especially when a few PGD modes can accurately represent the system solution (see Fig.1.2). Furthermore, in the PGD framework, the dimension and complexity of the problem scales linearly with the number of state space[1] dimensions, instead of exponentially.

Given the aforementioned advantages, one can conclude that the PGD can enact two distinct roles within a computational mechanics framework, depending on the nature of the solution field:

[1] The term state space refers to the vector space of all state variables of a system under consideration, i.e. generalized coordinates, system parameters etc.

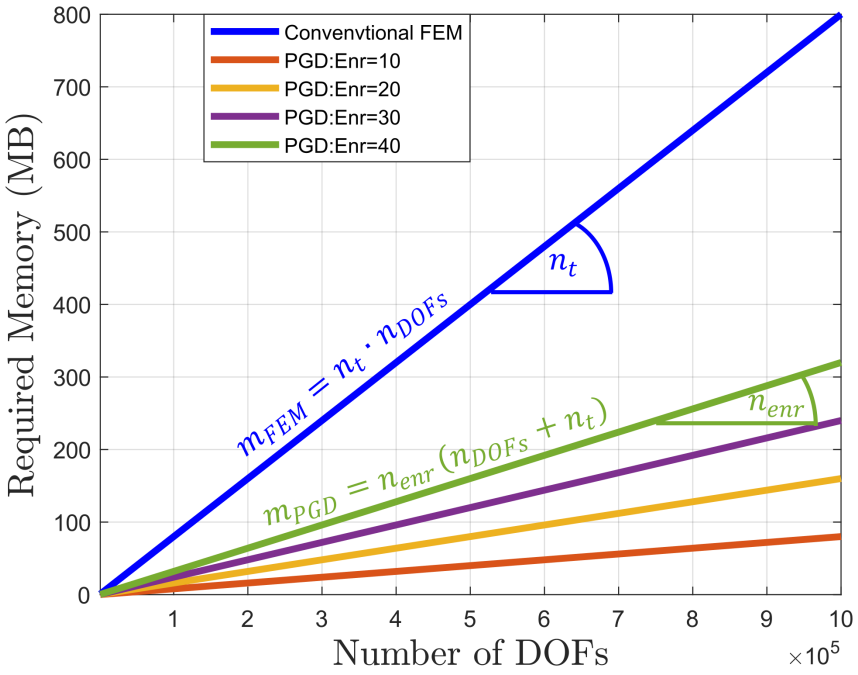
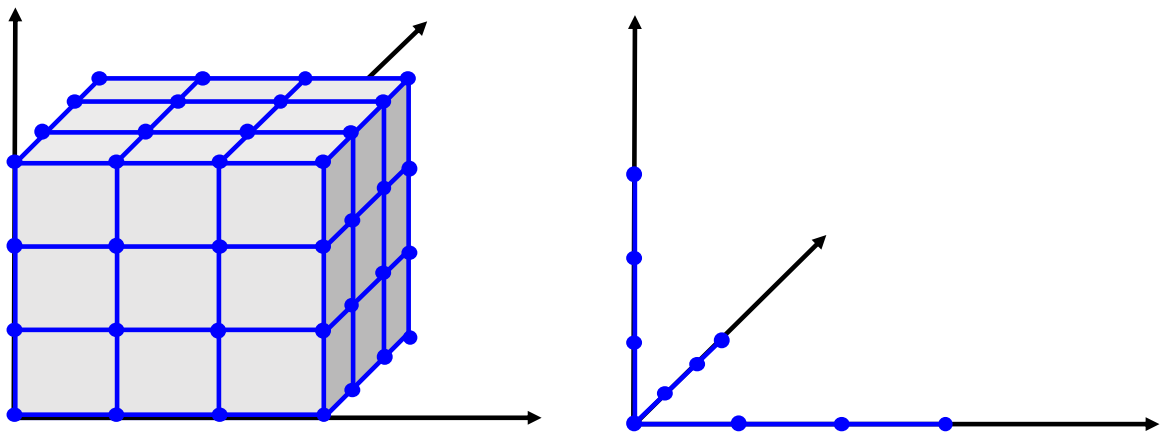


Figure 1.2: Solution field memory/storage requirements for PGD, $m_{\mathrm{PGD}} = n_{\mathrm{enr}}(n_{\mathrm{DOFs}}+n_{\mathrm{t}})$, and FEM, $m_{\mathrm{FEM}} = n_{\mathrm{DOFs}}n_{\mathrm{t}}$

1. Known Field: For a solution field that has already been computed using conventional methods, the PGD can act as a compression algorithm, achieving minimum storage requirements.

2. Unknown Field. For an unknown solution field, the PGD can act as a MOR method, mitigating the problem of high computational costs.

The concept of the PGD, as well as the space-time PGD-Newmark formulation are presented in detail in Chapter 3.



(a) FEM **(b)** PGD

Figure 1.3: Discretization of one $3D$ domain (FEM) vs three $1D$ domains (PGD)

1.3 Scope of the Thesis

The scope of this thesis is the methodological presentation of the Newmark-PGD space-time formulation and the investigation of applicability to linear structural dynamics. The method is demonstrated by means of numerical benchmarks of 3 different models. The convergence behavior, accuracy and numerical efficiency is presented, as well as the limits of the approach. The evolution of the spatial enrichment vectors is investigated during convergence and compared to the linear modes of vibration. This work aims to answer the following questions:

1. Does the PGD space-time approach yield accurate results for linear structural dynamics problems?
2. Is the approach able to accelerate the analyses?
3. Is there any memory/storage requirement reduction achieved?
4. Can the spatial PGD modes be related to the linear modes of vibration?

To this end, the following steps were carried out:

- Extensive literature research of the PGD method.
- Development of MATLAB code for the dynamic analysis of structures.
- Development of MATLAB code for the algorithmic implementation of the PGD space-time approach.

1.4 Thesis Layout

After this introductory chapter, this work is structured as follows. Chapter 2 presents the mathematical formulation of elastodynamics and the conventional Newmark method for solving the equation of motion. In Chapter 3, the Proper Generalized Decomposition (PGD) is presented in detail, the separated representation principles are given and the PGD-Newmark space-time approach is formulated. Chapter 4 contains the numerical benchmarks used for the validation of the method and Chapter 5 concludes with some remarks and proposals for future work. Some brief Tensor Algebra definitions are included in Appendix A.

Chapter 2

DYNAMIC ANALYSIS OF STRUCTURES

2.1 Introduction

Apart from static loads, engineering structures may be subjected to dynamic loads, that is, loads whose magnitude as well as direction of action and/or position may vary with time. Therefore, the fundamental objective of structural dynamics is the analysis of the response of a given structure undergoing dynamic loads. Two key facts differentiate dynamic from static analysis. The first, is the temporal variance of the loads, hence the dependence of the deformations and the stresses on time. The second one is that the position of the structural material points changes with time, so, apart from displacement, they have velocity and acceleration. Inasmuch as the structure has mass, the acceleration of the material points produces inertial forces that constitute additional loading, and therefore cannot be ignored.

In this chapter, the problem for the dynamic analysis of structures will be formulated for continuous structures, discretized as systems with a finite number of *Degrees of Freedom (DOFs)* at the nodes of the given structure. The equations of motion will be stated for a linear *Multi-Degree-of-Freedom (MDOF)* system and then the Newmark direct time integration scheme will be presented.

2.2 Problem Formulation

2.2.1 Elastodynamics

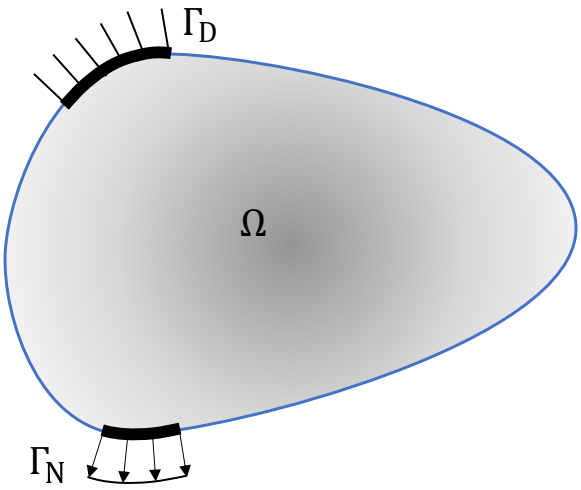


Figure 2.1: Deformable domain Ω, with volume V, subjected to boundary conditions.

Point of departure, for the initial boundary value problem (IBVP) of conventional elastodynamics, is the linear momentum balance equation at a material point of a body , in the time domain $(0, T]$.

$$\nabla \cdot \underline{\underline{\boldsymbol{\sigma}}} + \mathbf{b} = \rho \ddot{\mathbf{u}}, \qquad \text{in} \quad \Omega \times (0, T], \tag{2.1}$$

where $\underline{\underline{\boldsymbol{\sigma}}}(\mathbf{x}, t)$ is the second order Cauchy stress tensor, $\nabla \cdot \underline{\underline{\boldsymbol{\sigma}}}$ represents the divergence of the stress tensor [5], $\mathbf{b}$ is the vector field of body forces and $\mathbf{u}(\mathbf{x}, t)$ is the displacement vector field. With $\rho(\mathbf{x})$ we define the density of the material within the body Ω. Dirichlet boundary conditions are defined on the surface Γ_D and Neumann boundary conditions are defined on surface Γ_N of Ω (Fig.2.1), respectively, as:

$$\mathbf{u} = \bar{\mathbf{u}}, \qquad \text{on} \quad \Gamma_D \times (0, T] \tag{2.2}$$

$$\underline{\underline{\boldsymbol{\sigma}}} \cdot \mathbf{n} = \bar{\mathbf{t}}, \qquad \text{on} \quad \Gamma_N \times (0, T] \tag{2.3}$$

where $\bar{\mathbf{t}}$ is the external surface tractions vector and $\mathbf{n}$ is the normal to the boundary surface vector. The initial conditions are a displacement field and a velocity field at time instant $t = 0$.

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \qquad \text{in} \quad \Omega \tag{2.4}$$

$$\dot{\mathbf{u}}(\mathbf{x}, 0) = \mathbf{v}_0, \qquad \text{in} \quad \Omega \tag{2.5}$$

In a Galerkin approximation manner, multiplying Eq. (2.1) by a kinematically admissible vector field $\mathbf{w} = \delta\mathbf{u}$ (often called weight/test function or virtual displacement), and integrating over the entire body volume Ω, one gets:

$$\int_\Omega \left[\nabla \cdot \underline{\underline{\boldsymbol{\sigma}}}\right] \cdot \mathbf{w} dV + \int_\Omega \mathbf{b} \cdot \mathbf{w} dV = \int_\Omega \rho \ddot{\mathbf{u}} \cdot \mathbf{w} dV \tag{2.6}$$

Performing a partial integration of the first term and making use of the divergence theorem leads to the weak form of the IBVP (including the boundary conditions): (for the sake of simplicity, mathematic manipulations are omitted but the procedure can be found in [4], [5])

$$\int_\Omega \rho \delta\mathbf{u} \cdot \ddot{\mathbf{u}} dV + \int_\Omega \delta\underline{\underline{\boldsymbol{\epsilon}}} : \underline{\underline{\boldsymbol{\sigma}}} dV = \int_\Omega \delta\mathbf{u} \cdot \mathbf{b} dV + \int_{\Gamma_N} \delta\mathbf{u} \cdot \bar{\mathbf{t}} d\Gamma_N \tag{2.7}$$

where $\underline{\underline{\boldsymbol{\epsilon}}}$ is the strain tensor and $\bar{\mathbf{t}}$ is the boundary tractions vector field.

For the spatial discretization of Ω the Finite Element Method (FEM) is employed. Adopting shape (interpolation) functions [1] , whose coefficients are arranged in matrix $\mathbf{N}^{(\text{el})}$, the displacement field within each element $\mathbf{u}^{(\text{el})}$ and its 2nd temporal derivative, as well as the arbitrary virtual displacement are expressed, respectively, in terms of the nodal displacement vector $\mathbf{u}^{(\text{el})}_{(\text{nodal})}$ as:

$$\mathbf{u}^{(\text{el})} = \mathbf{N}^{(\text{el})} \mathbf{u}^{(\text{el})}_{(\text{nodal})} \tag{2.8}$$

[1] Lagrange polynomials are commonly adopted.

$$\ddot{\mathbf{u}}^{(\mathrm{el})} = \mathbf{N}^{(\mathrm{el})} \ddot{\mathbf{u}}^{(\mathrm{el})}_{(\mathrm{nodal})} \tag{2.9}$$

$$\delta\mathbf{u}^{(\mathrm{el})} = \mathbf{N}^{(\mathrm{el})} \delta\mathbf{u}^{(\mathrm{el})}_{(\mathrm{nodal})} \tag{2.10}$$

Equivalently, the element strain field $\underline{\underline{\boldsymbol{\epsilon}}}^{(\mathrm{el})} = \frac{1}{2}\left(\nabla\mathbf{u}^{(\mathrm{el})} + \nabla\mathbf{u}^{(\mathrm{el})^T}\right)$ and the virtual strain field $\delta\underline{\underline{\boldsymbol{\epsilon}}}^{(\mathrm{el})}$ are expressed via the element nodal displacement vector as:

$$\underline{\underline{\boldsymbol{\epsilon}}}^{(\mathrm{el})} = \mathbf{B}^{(\mathrm{el})} \mathbf{u}^{(\mathrm{el})}_{(\mathrm{nodal})} \tag{2.11}$$

$$\delta\underline{\underline{\boldsymbol{\epsilon}}}^{(\mathrm{el})} = \mathbf{B}^{(\mathrm{el})} \delta\mathbf{u}^{(\mathrm{el})}_{(\mathrm{nodal})} \tag{2.12}$$

where $\mathbf{B}^{(\mathrm{el})}$ is an element strain-displacement matrix, containing the derivatives of the shape functions $\mathbf{N}^{(\mathrm{el})}$.

Inserting Eq. (2.8) - (2.12) into the weak form Eq. (2.7) leads to a spatially discretized form of the Principle of Virtual Work for an element with volume $V^{(\mathrm{el})}$, occupying a domain $\Omega^{(\mathrm{el})}$.

$$\begin{aligned}\delta\mathbf{u}^{(\mathrm{el})} \Bigg[& \underbrace{\left(\int_{\Omega^{(\mathrm{el})}} \mathbf{N}^{(\mathrm{el})^T} \rho \mathbf{N}^{(\mathrm{el})} dV^{(\mathrm{el})}\right)}_{\mathbf{M}^{(\mathrm{el})}} \ddot{\mathbf{u}}^{(\mathrm{el})} \\ & + \underbrace{\left(\int_{\Omega^{(\mathrm{el})}} \mathbf{B}^{(\mathrm{el})^T} \underline{\underline{\boldsymbol{\sigma}}} dV^{(\mathrm{el})}\right)}_{\mathbf{f}^{(\mathrm{el})}_{\mathrm{int}}} \\ & + \underbrace{\left(\int_{\Omega^{(\mathrm{el})}} \mathbf{N}^{(\mathrm{el})^T} \mathbf{b} dV^{(\mathrm{el})}\right)}_{\mathbf{b}^{(\mathrm{el})}} \\ & + \underbrace{\left(\int_{\Gamma_N} \mathbf{N}^{(\mathrm{el})^T} \bar{\mathbf{t}} d\Gamma_N\right)}_{\mathbf{f}^{(\mathrm{el})}_{\mathrm{ext}}} \Bigg] = 0\end{aligned} \tag{2.13}$$

Next, in a general FEM fashion, an operation is defined, and performed over all elements, that casts the local element properties(matrices, coordinates, etc.) to a global structural formulation [2]. Hence, the Principle of Virtual Work is expressed in global form for the entire system:

$$\delta\mathbf{u} \left[\underbrace{\left(\int_{\Omega} \mathbf{N}^T \rho \mathbf{N} dV\right)}_{\mathbf{M}} \ddot{\mathbf{u}} + \underbrace{\left(\int_{\Omega} \mathbf{B}^T \underline{\underline{\boldsymbol{\sigma}}} dV\right)}_{\mathbf{f}_{\mathrm{int}}} + \underbrace{\left(\int_{\Omega} \mathbf{N}^T \mathbf{b} dV\right)}_{\mathbf{b}} + \underbrace{\left(\int_{\Gamma_N} \mathbf{N}^T \bar{\mathbf{t}} d\Gamma_N\right)}_{\mathbf{f}_{\mathrm{ext}}} \right] = 0 \tag{2.14}$$

[2] Details on the specifics of the global assembly operation can be found in [1].

where $\mathbf{M} \in \mathbb{R}^{n_s \times n_s}$ is the global mass matrix of the system, $\mathbf{f}_{\text{int}} \in \mathbb{R}^{n_s}$ is the global internal force vector, $\mathbf{b} \in \mathbb{R}^{n_s}$ is the global body force vector and $\mathbf{f}_{\text{ext}} \in \mathbb{R}^{n_s}$ is the global external force vector. The number of Degrees of Freedom (DOFS), which depends on the discretization, is denoted by n_s.

Under a Linear Elasticity regime, the stress-strain constitutive relation (Hooke's law) is expressed in tensorial form as:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \tag{2.15}$$

where C_{ijkl} is the 4$^{\text{th}}$ order elasticity (or stiffness) tensor, containing the elastic moduli (or material properties). Using the underline convention (see Appendix A), Eq. (2.15) can be written in matrix form as:

$$\underline{\underline{\boldsymbol{\sigma}}} = \underline{\underline{\underline{\underline{\mathbf{C}}}}} : \underline{\underline{\boldsymbol{\epsilon}}} \tag{2.16}$$

Isolating the element internal force vector from Eq. (2.13) and substituting stress and strain terms, according to Eq. (2.16) and Eq. (2.11) respectively, leads to:

$$\mathbf{f}_{\text{int}}^{(\text{el})} = \left(\int_{\Omega^{(\text{el})}} \mathbf{B}^{(\text{el})^T} \underline{\underline{\underline{\underline{\mathbf{C}}}}} \mathbf{B}^{(\text{el})} dV^{(\text{el})} \right) \mathbf{u}^{(\text{el})} = \mathbf{K}^{(\text{el})} \mathbf{u}^{(\text{el})} \tag{2.17}$$

where $\mathbf{K}^{(\text{el})}$ is the local element stiffness matrix. Assembly of all element stiffness matrices leads to the global stiffness matrix of the system $\mathbf{K} \in \mathbb{R}^{n_s \times n_s}$.

The requirement for the Principle of Virtual Work to hold for any virtual displacement, leads to the equation of motion of the system:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} + \mathbf{b} + \mathbf{f}_{\text{ext}} = \mathbf{0} \tag{2.18}$$

2.2.2 Equation of motion

The problem of determining the dynamic response of a system is mathematically described by the following initial value problem (IVP), where the right hand side is the sum of the external and the body force vectors.

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) &= \mathbf{f}(t) \\ \mathbf{u}(t_0) &= \mathbf{u}_0 \\ \dot{\mathbf{u}}(t_0) &= \dot{\mathbf{u}}_0 \end{aligned} \tag{2.19}$$

where: $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ the stiffness matrix, $\mathbf{C}$ the Rayleigh damping matrix ([1]), $\mathbf{u}$ the nodal displacement vector, $\dot{\mathbf{u}}$ the nodal velocity vector, $\ddot{\mathbf{u}}$ the nodal acceleration vector and $\mathbf{f}$ the external force vector. Eq. (2.19), can be alternatively written as:

$$\mathbf{F}_I(t) + \mathbf{F}_D(t) + \mathbf{F}_E(t) - \mathbf{f}(t) = 0, \tag{2.20}$$

where $\mathbf{F}_I(t) = \mathbf{M}\ddot{\mathbf{u}}(t)$ are the inertial forces, $\mathbf{F}_D(t) = \mathbf{C}\dot{\mathbf{u}}(t)$ the damping forces and $\mathbf{F}_E(t) = \mathbf{K}\mathbf{u}(t)$ the linear elastic restoring forces

The equation of motion of a vibrating system Eq. (2.19) written in the form of Eq. (2.20) expresses the equilibrium condition of all the forces acting upon the system, that is the external excitation forces and the internal resistance forces, at a particular time instant t.

The equilibrium condition for a dynamic, linear elastic, damped system, subjected to seismic excitation, with n_s degrees of freedom is expressed in matrix form as:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = -\mathbf{M}\mathbf{r}\ddot{u}_g(t), \tag{2.21}$$

where $\mathbf{M}$ is the $(n_s \times n_s)$ mass matrix, $\mathbf{K}$ the $(n_s \times n_s)$ stiffness matrix, $\mathbf{C}$ the $(n_s \times n_s)$ damping matrix, $\mathbf{u}$ the $(n_s \times 1)$ nodal relative displacement vector, $\ddot{u}_g$ the ground acceleration and $\mathbf{r}$ the $(n_s \times 1)$ influence vector, that expresses the spatial distribution of the external excitation, over the discretized computational domain.

Mathematically, Eq. (2.21) represents a system of linear 2nd order differential equations, the solution of which can be computed via any conventional method of solving ODEs with constant coefficients. Such conventional methods, however, can be particularly computationally taxing, especially when the order of the coefficient matrices is high. Within a Finite Element Analysis (FEA) setting, the system matrices are usually characterized by a sparse and banded matrix structure. This fact entails the use of numerical methods, tailored to engineering problems, that exploit this sparsity.

The solution techniques that are commonly adopted are the Modal Superposition Method, the Direct Numerical Integration Schemes, the Response Spectrum Analysis (RSA) and the Frequency Domain Analysis. In the current work, the linear elastic analyses are carried out via direct numerical integration of the equations of motion.

2.3 Newmark Method - Incremental Solution Approach

Point of departure for the Newmark scheme are the Taylor series expansion relations for the displacement, the velocity and the acceleration at time instant $t + \Delta t$:

$$\mathbf{u}(t+\Delta t) = \mathbf{u}(t) + \Delta t\dot{\mathbf{u}}(t) + \frac{\Delta t^2}{2!}\ddot{\mathbf{u}}(t) + \frac{\Delta t^3}{3!}\dddot{\mathbf{u}}(t) + \frac{\Delta t^4}{4!}\ddddot{\mathbf{u}}(t) + \cdots \tag{2.22}$$

$$\dot{\mathbf{u}}(t+\Delta t) = \dot{\mathbf{u}}(t) + \Delta t\ddot{\mathbf{u}}(t) + \frac{\Delta t^2}{2!}\dddot{\mathbf{u}}(t) + \frac{\Delta t^3}{3!}\ddddot{\mathbf{u}}(t) + \cdots \tag{2.23}$$

$$\ddot{\mathbf{u}}(t+\Delta t) = \ddot{\mathbf{u}}(t) + \Delta t\,\dddot{\mathbf{u}}(t) + \frac{\Delta t^2}{2!}\ddddot{\mathbf{u}}(t) + \cdots \tag{2.24}$$

Rearranging equation Eq. (2.24),

$$\Delta t\,\dddot{\mathbf{u}}(t) = \ddot{\mathbf{u}}(t+\Delta t) + -\ddot{\mathbf{u}}(t) - \frac{\Delta t^2}{2!}\ddddot{\mathbf{u}}(t) + \cdots \tag{2.25}$$

and substituting Eq. (2.25) in Eq. (2.22) and Eq. (2.23) yields the following expressions:

$$\mathbf{u}(t+\Delta t) = \mathbf{u}(t) + \Delta t\dot{\mathbf{u}}(t) + \frac{\Delta t^2}{2!}\ddot{\mathbf{u}}(t) + \frac{\Delta t^2}{3!}\left[\ddot{\mathbf{u}}(t+\Delta t) - \ddot{\mathbf{u}}(t) - \frac{\Delta t^2}{2!}\ddddot{\mathbf{u}}(t) - \cdots\right] + \frac{\Delta t^4}{4!}\ddddot{\mathbf{u}}(t) + \cdots \tag{2.26}$$

$$\dot{\mathbf{u}}(t+\Delta t) = \dot{\mathbf{u}}(t) + \Delta t\ddot{\mathbf{u}}(t) + \frac{\Delta t^2}{2!}\ddot{\mathbf{u}}(t) + \frac{\Delta t^2}{2!}\left[\ddot{\mathbf{u}}(t+\Delta t) - \ddot{\mathbf{u}}(t) - \frac{\Delta t^2}{2!}\ddddot{\mathbf{u}}(t) - \cdots\right] + \frac{\Delta t^3}{3!}\ddddot{\mathbf{u}}(t) + \cdots \tag{2.27}$$

Truncating 4th and higher order terms from the above equations, one gets the following finite difference approximations:

$$\mathbf{u}(t+\Delta t) \approx \mathbf{u}(t) + \Delta t \dot{\mathbf{u}}(t) + \frac{\Delta t^2}{6}\left[\ddot{\mathbf{u}}(t+\Delta t) + 2\ddot{\mathbf{u}}(t)\right] \tag{2.28}$$

$$\dot{\mathbf{u}}(t+\Delta t) \approx \dot{\mathbf{u}}(t) + \frac{\Delta t}{2}\left[\ddot{\mathbf{u}}(t+\Delta t) + \ddot{\mathbf{u}}(t)\right] \tag{2.29}$$

These relations are implicit, in the sense that, in order to determine $\mathbf{u}(t+\Delta t)$ and $\dot{\mathbf{u}}(t+\Delta t)$, $\ddot{\mathbf{u}}(t+\Delta t)$ is required. But, the accelerations $\ddot{\mathbf{u}}(t+\Delta t)$ cannot be computed without $\mathbf{u}(t+\Delta t)$ and $\dot{\mathbf{u}}(t+\Delta t)$. Let us note here, that the above manipulations, eliminated 3rd order temporal derivative terms and the method has thus accuracy of the order of $\Delta t^4\, \dddot{\mathbf{u}}(t)$.

The above procedure is called the linear acceleration method, since the 3rd temporal derivative of $\mathbf{u}$ is eliminated. If the rate of change of the acceleration within a time step is indeed constant, then the approximation of truncating 4th and higher order terms from the Taylor series expansions does not affect the accuracy of the solution.

According to the Newmark method [18], the displacements and the velocities within the current time step are computed by integrating the corresponding expression for the acceleration, assuming its linear variation within the time step. After this integration, the finite difference expressions of the Newmark scheme for the displacement and the velocity are the following:

$$\mathbf{u}(t+\Delta t) \approx \mathbf{u}(t) + \Delta t \dot{\mathbf{u}}(t) + \Delta t^2\left[\left(\frac{1}{2} - \beta\right)\ddot{\mathbf{u}}(t) + \beta \ddot{\mathbf{u}}(t+\Delta t)\right] \tag{2.30}$$

and

$$\dot{\mathbf{u}}(t+\Delta t) \approx \dot{\mathbf{u}}(t) + \Delta t[(1-\gamma)\ddot{\mathbf{u}}(t) + \gamma \ddot{\mathbf{u}}(t+\Delta t)] \tag{2.31}$$

where β and γ, [3] are parameters that are determined based on the desired accuracy and stability of the numerical integration.

The most commonly known cases of the Newmark scheme are the linear acceleration method and the constant - average acceleration method. First, Newmark proposed, as an unconditionally stable and implicit method, the average acceleration method, which results from setting the parameters $\gamma = 1/2$ and $\beta = 1/4$ in Eq. (2.30) and Eq. (2.31). The linear acceleration method is acquired by setting $\gamma = 1/2$ and $\beta = 1/6$, whereas choosing $\gamma = 1/2$ and $\beta = 0$ reduces the Nemwark scheme to the well known Central Difference Method (CDM).

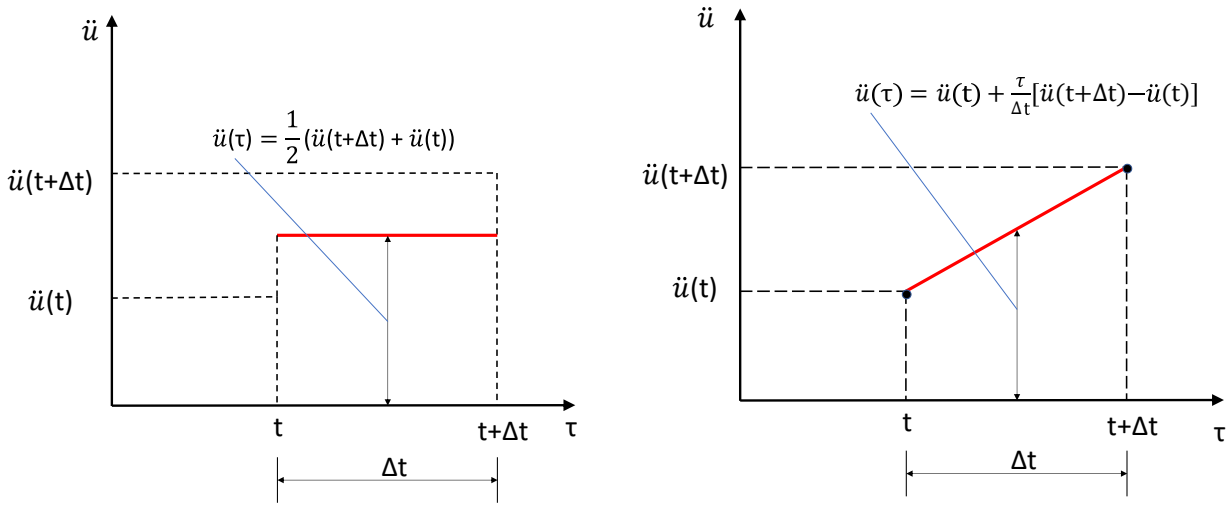
For the linear dynamic analysis case, if $2\beta \geq \gamma \geq 1/2$, then the Newmark method is unconditionally stable, i.e. regardless of the time step Δt. Also, the method is conditionally stables for values of $\gamma < 1/2$. For $\gamma = 1/2$ the method is at least 2nd order accurate [4]. Figures 2.2a and 2.2b depict graphically the constant and linear acceleration approximations respectively. The acceleration within a time step, $\ddot{\mathbf{u}}(\tau)$, is integrated twice with respect to τ to compute the velocity and displacement.

Substituting Eq. (2.30) and 2.31 in Eq. (2.21) (when expressed for time instant $t+\Delta t$) and performing some algebra, yields the following:

$$\hat{\mathbf{k}}\mathbf{u}(t+\Delta t) = \hat{\mathbf{p}}(t+\Delta t) \tag{2.32}$$

[3] In the literature, many authors, instead of β and γ [3], make use of the letters $\alpha(=\beta)$ and $\delta(=\gamma)$ [2]

[4] We note, for the approximation $\mathbf{u}_{\text{sol}}$ of the exact solution $\mathbf{u}$, that it is ν-order accurate, if the error $E := |\mathbf{u} - \mathbf{u}_{\text{sol}}|$ is proportionate to Δt^{ν} of the numerical procedure, i.e.: $E := |\mathbf{u} - \mathbf{u}_{\text{sol}}| \leq C(\Delta t)^{\nu}$, where C: a constant dependent on Δt.



(a) Average acceleration method

(b) Linear acceleration method

Figure 2.2: Acceleration variation for the Newmark method

with

$$\hat{\mathbf{k}} = \left[\frac{1}{\beta\Delta t^2} + \frac{\gamma}{\beta\Delta t}\mathbf{C} + \mathbf{K}\right] \tag{2.33}$$

and

$$\begin{aligned}\hat{\mathbf{p}}(t+\Delta t) = {} & \mathbf{f}_{\text{ext}}(t+\Delta t) + \\ & \left[\frac{1}{\beta\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\Delta t}\mathbf{C}\right]\mathbf{u}(t) + \\ & \left[\frac{1}{\beta\Delta t}\mathbf{M} + \frac{\gamma-\beta}{\beta}\mathbf{C}\right]\dot{\mathbf{u}}(t) + \\ & \left[\frac{1-2\beta}{2\beta}\mathbf{M} + \frac{\gamma-2\beta}{2\beta}\mathbf{C}\right]\ddot{\mathbf{u}}(t)\end{aligned} \tag{2.34}$$

Eq. (2.32) is solved in each time increment, given the system parameters $\mathbf{M}$, $\mathbf{K}$, $\mathbf{C}$, the algorithmic parameters β, γ and the current state of the system at time t described by $\mathbf{u}(t)$, $\dot{\mathbf{u}}(t)$, $\ddot{\mathbf{u}}(t)$.

Thus, having computed the displacement at time $t+\Delta t$:

$$\mathbf{u}(t+\Delta t) = \hat{\mathbf{p}}(t+\Delta t)/\hat{\mathbf{k}} \tag{2.35}$$

all that remains is to determine the velocity and acceleration. First, Eq. (2.30) is solved for $\ddot{\mathbf{u}}(t+\Delta t)$ and from the acquired relation, $\dot{\mathbf{u}}(t+\Delta t)$ is expressed, through Eq. (2.31), in terms[5] of $\mathbf{u}(t+\Delta t)$.

Though this procedure, the equation of motion is solved incrementally for the entire duration of the problem. Next, a pseudo-algorithm for the Newmark scheme is provided (Algorithm 1). The integration constants in line 3 of Algorithm 1 are computed as:

$$\begin{aligned} c_0 &= \frac{1}{\beta\Delta t^2} & c_1 &= \frac{\gamma}{\beta\Delta t} & c_2 &= \frac{1}{\beta\Delta t} \\ c_3 &= \frac{1}{2\beta} - 1 & c_4 &= \frac{\gamma}{\beta} - 1 & c_5 &= \frac{\Delta t}{2}\left(\frac{\gamma}{\beta} - 2\right) \\ c_6 &= \Delta t(1-\gamma) & c_7 &= \gamma\Delta t \end{aligned} \tag{2.36}$$

[5] For the analytical expressions, see algorithm 1.

and the effective force vector, in line 6 of Algorithm 1, as:

$$\hat{\mathbf{f}}(t+\Delta t) = \mathbf{f}(t+\Delta t) + \mathbf{M}[c_0\mathbf{u}(t) + c_2\dot{\mathbf{u}}(t) + c_3\ddot{\mathbf{u}}(t)] + \mathbf{C}[c_1\mathbf{u}(t) + c_4\dot{\mathbf{u}}(t) + c_5\ddot{\mathbf{u}}(t)] \tag{2.37}$$

Algorithm 1 Newmark Integration Scheme

INPUT: $\mathbf{K}, \mathbf{M}, \mathbf{C}, \mathbf{f}$

OUTPUT: $\mathbf{u}_j, \dot{\mathbf{u}}_j, \ddot{\mathbf{u}}_j$, for $j = 1 : n_t$

1: initial conditions: $\mathbf{u}(0), \dot{\mathbf{u}}(0), \ddot{\mathbf{u}}(0)$
2: select: $\Delta t, \beta, \gamma$ such that: $\gamma \geq 0.5\beta \geq 0.25(0.5+\gamma)^2$
3: compute: integration constants $\leftarrow c_1, c_2, c_3, c_4, c_5, c_6$
4: compute: effective stiffness $\leftarrow \hat{\mathbf{K}} = \mathbf{K} + c_0\mathbf{M} + c_1\mathbf{C}$
5: **for** $j = 0 : n_t - 1$ **do** ▷ $(j+1 \equiv t+\Delta t)$
6: compute: effective force vector $\leftarrow \hat{\mathbf{f}}(t+\Delta t)$
7: solve: $\hat{\mathbf{K}}\mathbf{u}(t+\Delta t) = \hat{\mathbf{f}}(t+\Delta t)$
8: compute: velocities at $t+\Delta t \leftarrow \dot{\mathbf{u}}(t+\Delta t) = \dot{\mathbf{u}}(t) + c_6\ddot{\mathbf{u}}(t) + c_7\ddot{\mathbf{u}}(t+\Delta t)$
9: compute: accelerations at $t+\Delta t \leftarrow \ddot{\mathbf{u}}(t+\Delta t) = c_0[\mathbf{u}(t+\Delta t) - \mathbf{u}(t)] - c_2\dot{\mathbf{u}}(t) - c_3\ddot{\mathbf{u}}(t)$
10: **end for**

Chapter 3

PROPER GENERALIZED DECOMPOSITION (PGD)

3.1 PGD Principles

3.1.1 Separated Representations

The Proper Generalized Decomposition (PGD) is constructed upon a very old idea, i.e. the method of separation of variables or Fourier method for partial differential equations. The novelty lies in the ability of the PGD to construct sums of separated functions *a priori*, i.e. without any prior knowledge on the solution nor the need for costly computer simulations or snapshots. Hence, it can be viewed as an *a priori* approach to reduced-order modelling.

To begin with, consider a (computational) problem defined in a D-dimensional space for the unknown field $u(x_1,\dots,x_D)$. Coordinates x_i may denote any physical coordinate (space, time, etc.) but also any problem parameters such as material parameters or boundary conditions. The key idea of the PGD, is that the solution for $(x_1,\dots,x_D) \in \Omega_1 \times \cdots \times \Omega_D$ is sought in the form of:

$$u(x_1,\dots,x_D) = \sum_{i=1}^{\infty} F_i^1(x_1)\dots F_i^D(x_D) \tag{3.1}$$

The PGD yields an approximate solution $u^{(N)}$ in separated form

$$u^{(N)}(x_1,\dots,x_D) = \sum_{i=1}^{N} F_i^1(x_1)\dots F_i^D(x_D) \tag{3.2}$$

where both the number of summands (from now on called *enrichments*) N and functions $F_i^j(x_j)$ are unknown *a priori*. Eq. (3.2) is called a *separated representation* (also known as finite sum decomposition) of the solution u. The separated representation resembles the classical separation of variables used for solving Partial Differential Equations (PDEs) analytically (only in the limited cases when this is at all possible). Furthermore, Eq. (3.2) is stated without loss of generality, in the sense that any polynomial can be expressed in that form. It is worth noting here, that, in general, the nonlinear nature of the representation Eq. (3.2) constitutes a computational burden.

To demonstrate the gain from using separated representations, we consider a $2D$ example, showcased in [6]. Assume that the sought solution of a $2D$ problem is given by $u(x,y) = x^n y^m$. Attempting to approximate the solution using standard Lagrange polynomial approximation, one must consider all monomials up to degree $n+m$.

$$u^{(n+m)}(x,y) = \sum_{i=1}^{n+m} \sum_{j=1}^{i} \alpha_{i-j,j} x^{i-j} y^{j} \tag{3.3}$$

Thus, an accurate approximation using Lagrange polynomials involves the calculation of $\frac{(n+m)(n+m+1)}{2}$ weight terms $\alpha_{i-j,j}$. Nevertheless, inspecting the solution, one can easily conclude that only one non-zero coefficient is to be obtained, i.e. $\alpha_{n,m} = 1$.

On the contrary, the separated representation

$$u^{(N)}(x,y) = \sum_{i=1}^{N} F_i^x(x) F_i^y(y) \tag{3.4}$$

would capture the solution with only one term, i.e. $N = 1$, $F_1^x(x) = x^n$, $F_2^y(y) = y^m$.

3.1.2 Successive Enrichment Procedure

From the aforementioned, one can conclude that the PGD approximation Eq. (3.2) is a sum of N functional products (*enrichments*), each involving D different functions $F_i^j(x_j)$. This approximation is constructed by successive enrichment, computing each functional product sequentially. At a particular enrichment step $n+1$, with all previous functions $F_{i\leq n}^j(x_j)$ as given, one must compute the current new product of the D unknown functions $F_{n+1}^j(x_j)$. Hence, a $1D$ problem can be defined in Ω_j for each one of the D functions $F_{n+1}^j(x_j)$.

In practice, this is achieved by invoking the weak form of the problem under consideration. In a general computational manner, to determine these functions $F_{n+1}^j(x_j)$, a discrete nonlinear algebraic system of equations is to be solved. This nonlinearity arises regardless of the character of the initial problem and stems from the fact that we seek one or more products of functions. This implies that iterations are needed within each enrichment step.

To cope with this, very simple techniques have demonstrated to provide very good results both in terms of accuracy and convergence. Throughout the PGD literature [22], greedy algorithms have been employed, allowing the computation of one finite sum at a time, and within each greedy algorithm step, naive linearization strategies (e.g. fixed point iterations) are applied. Throughout this work a Fixed Point Algorithm is employed, that will be briefly presented in Section 3.3.

Within a Finite Element setting the computational domain is descretized using a mesh over each coordinate space Ω_j. Using conventional mesh based methods, the unknowns (degrees of freedom) to be determined are in general $\propto M^D$, where M is the number of nodes used to discretize each coordinate space. But employing the PGD approach, the number of unknowns is $N \times M \times D$ (see Fig.1.2).

3.1.3 From POD to PGD-Based Model Order Reduction

Model Order Reduction (MOR) methods have found vast applications in engineering problems, especially for solving transient dynamic models in spatial domains that involve a large number of degrees of freedom and very fine temporal discretization.

Assume the response of a transient model $\mathbf{u}(\mathbf{x},t)$ defined in space ($\mathbf{x} \in \Omega$) and time ($t \in (0,T]$). Using standard mesh based techniques, the discrete solution is calculated at M nodal points and n_t time instants, provided that n_t is large enough to satisfy convergence and stability criteria.

Within a finite element setting, acquiring the solution requires in general the solution of a linear system of dimension $n_{DOF} = f(M)$, at each time instant, where n_{DOF} is the number of degrees of freedom. It is thus evident that the computational complexity of the solution procedure scales with

$M \cdot n_t$. Many computational bottlenecks, such as domain complexity or/and convergence and stability constraints can require extremely fine spatial and temporal discretization respectively. This in turn, scales the computational complexity and renders the solution unaffordable or even unattainable. Hence, Model Order Reduction techniques constitute an attractive tool to surpass such computational barriers.

3.1.3.1 Relevant Information from Solution Data

The Proper Orthogonal Decomposition (POD) is an established MOR technique that allows circumventing the above mentioned difficulties when the problem solution lives in a subspace of dimension smaller than that of the original discrete problem.

Assuming that the solution $\mathbf{u}(\mathbf{x},t)$ is known at the nodal points of the spatial mesh for n_t time instants, one can define a snapshot matrix as:

$$\mathbf{Q} = \begin{pmatrix} u_1^{(1)} & u_1^{(2)} & \dots & u_1^{(n_t)} \\ u_2^{(1)} & u_2^{(2)} & \dots & u_2^{(n_t)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_{DOF}}^{(1)} & u_{n_{DOF}}^{(2)} & \dots & u_{n_{DOF}}^{(n_t)} \end{pmatrix} \tag{3.5}$$

where each column represents a solution vector at a particular time instant. Next, what is known as the snapshot two-point correlation matrix can be defined as:

$$\mathbf{C} = \mathbf{Q}\mathbf{Q}^T \tag{3.6}$$

which is symmetric and positive definite.

The main objective of the POD method is to obtain the most characteristic structure $\boldsymbol{\phi}(\mathbf{x})$ among the solution vectors $\mathbf{u}(\mathbf{x},t_j)$, $\forall j \in [1,n_t]$. To this end, the following scalar quantity is to be maximized:

$$\kappa = \frac{\sum_{j=1}^{n_t} \left[\sum_{i=1}^{n_{DOF}} \phi_i u_{i,j}\right]^2}{\sum_{i=1}^{n_{DOF}} (\phi_i)^2} \tag{3.7}$$

where ϕ_i is the $i-$ component of vector $\boldsymbol{\phi}$ and $u_{i,j}$ is the $i-$ component of the solution vector at time instant j.

The above maximization (Eq. (3.7)), mathematically amounts to, either solving the following eigenvalue problem for the correlation matrix:

$$\mathbf{C}\boldsymbol{\phi} = \lambda\boldsymbol{\phi} \tag{3.8}$$

or performing a Singular Value Decomposition (SVD)[1] directly on the snapshot matrix $\mathbf{Q}$ [2].

$$\mathbf{Q} = \mathbf{U}\Sigma\mathbf{V}^T \tag{3.9}$$

The Reduced Order Model (ROM) is then constructed by defining a reduced basis that is spanned by a selected small number, n_r , of eigenvectors (Eq. (3.8)) (or left singular vectors (Eq. (3.9))). The

[1] The SVD of a matrix yields matrices $\mathbf{U}$ and $\mathbf{V}$ whose columns contain the left and right singular vectors and a sparse matrix Σ whose main diagonal contains the singular values.

[2] The singular values sv_i of matrix $\mathbf{Q}$ are equal to the eigenvalues λ_i of matrix $\mathbf{Q}\mathbf{Q}^T$

dimension n_r of the subspace is chosen such that most of the solution relevant information is retained. Various criteria exist for this choice, such as the selection of the n_r eigenvectors $\boldsymbol{\phi}^{(i)}$ associated with the eigenvalues that belong in an interval $[\max(\alpha_i)/Z, \max(\alpha_i))$, where Z is a large enough number (e.g. 10^8), or choosing as many first left singular vectors as are the singular values whose normalized sum is greater than 99% ($\sum_{i=1}^{n_r} \frac{sv_i}{\max(sv_i)} \geq 99\%$).

Thus, a matrix that maps the original solution to the computed subspace is defined as:

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\phi}^{(1)} & \boldsymbol{\phi}^{(2)} & \dots & \boldsymbol{\phi}^{(n_r)} \end{pmatrix} \tag{3.10}$$

For many applications n_r has been found much lower than the initial dimension of the problem.

Within an explicit incremental (time-stepping) scheme, given the solution $\mathbf{u}^{(j)}$ at time t_j, the solution at the next time instant is acquired by solving a linear algebraic system in the form of:

$$\mathbf{G}^{(j)}\mathbf{u}^{(j+1)} = \mathbf{H}^{(j)} \tag{3.11}$$

Approximating solution $\mathbf{u}^{(j+1)}$ in the subspace spanned by the basis $\boldsymbol{\Phi}$

$$\mathbf{u}^{(j+1)} \approx \boldsymbol{\Phi}\mathbf{q}^{(j+1)} \tag{3.12}$$

the reduced order model equation is obtained as:

$$\begin{aligned} \mathbf{G}^{(j)}\boldsymbol{\Phi}\mathbf{q}^{(j+1)} &= \mathbf{H}^{(j)} \\ \boldsymbol{\Phi}^T\mathbf{G}^{(j)}\boldsymbol{\Phi}\mathbf{q}^{(j+1)} &= \boldsymbol{\Phi}^T\mathbf{H}^{(j)} \end{aligned} \tag{3.13}$$

Thus, the subspace solution vector is acquired by solving an algebraic system of size n_r instead of n_{DOF}. This is generally preferred, especially when $n_r \ll n_{DOF}$, which is the case for numerous applications.

An important dispute rises, regarding the point of the entire procedure presented above. This is none other than the fact that the ROM is built *a posteriori* using already computed solutions of the evolution field. Two beneficial approaches are mainly considered to answer this. The first consists of solving the Full Order Model over a short time interval (snapshot window) allowing for the extraction of the characteristic structure of the solution field that defines the ROM. The latter is then solved over larger time intervals, leading to computing time savings. The other approach consists of solving the original model over the entire time interval and then employing the corresponding ROM to efficiently solve similar problems (in terms of slight variations in material parameters or boundary conditions).

3.1.3.2 Motivation for an *A Priori* Space-Time Separated Representation

The previous section clearly illustrates the value of MOR. However, the ideal scenario would be to be able to build a reduced order basis *a priori*, without the reliance on the precomputed solution of the initial model. The accuracy of the ROM solution could then be assessed and, if deemed necessary, the reduced basis would be enriched to improve accuracy [22]. The Proper Generalized Decomposition, as also previously mentioned, efficiently pairs with this scenario.

The above POD procedure yields one more interesting remark, that being the fact that an accurate solution approximation can often be written as a separated representation involving a few number of terms. Especially when the solution field evolution is smooth, the ordered eigenvalues (or singular values) display a rapid decay. Hence, the evolution of the field can be approximated by a low number, n_r, of modes since most of the system energy is preserved with them. Consequently, the space-time

dependent solution can be approximated by sum of a small number of functional products, with one function depending on the space coordinates and the other one on time.

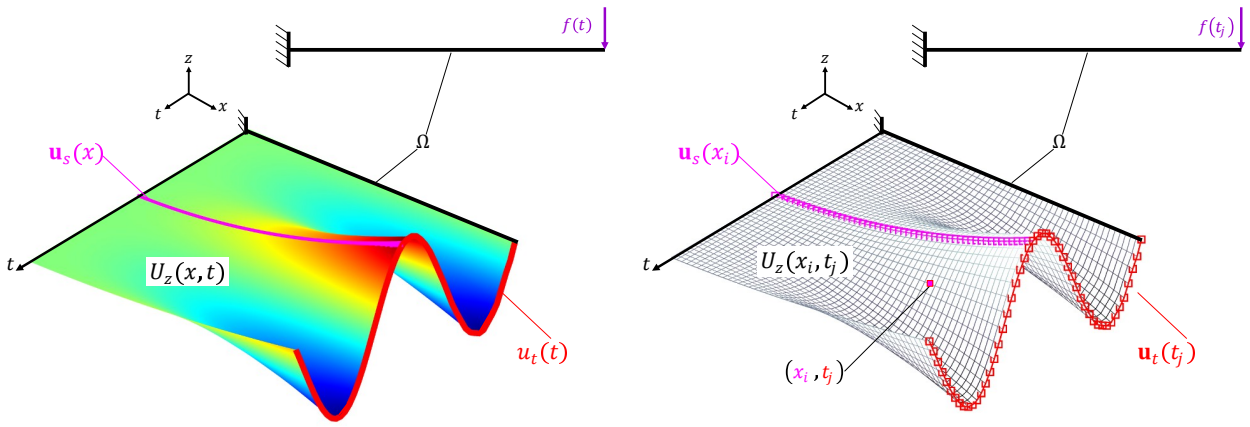
$$u(\mathbf{x},t) = \sum_{i=1}^{n_r} u_s^{(i)}(\mathbf{x}) u_t^{(i)}(t) \tag{3.14}$$

where $u_s(\mathbf{x})$ denote the spatial modes and $u_t(t)$ the temporal modes. As mentioned in Section 3.1.1, separated representations like Eq. (3.14) are at the core of the PGD.

3.2 Separating Physical Space and Time

3.2.1 Continuous Space-Time Decomposition

For the sake of intuitiveness, attention is given to Fig.3.1, which showcases the continuous (Fig.3.1a) space-time response surface for the transverse displacement $U_z(x,t)$ of a plane cantilever subjected to a sinusoidal excitation at its free end, as well as its descretized representation $U_z(x_i,t_j)$ (Fig.3.1b). The goal is to determine this space-time response surface employing a strategy that does not proceed



(a) Continuous representation **(b)** Discrete representation

Figure 3.1: Illustration of a plane cantilever beam in the space-time framework: $\Omega \times (0,T]$

in a step-by-step manner but rather determines the evolution at once.

Starting from the continuous system description, we aim to express the space-time counterparts of all the response quantities and operators of the system. Using Eq. (3.1) and Eq. (A.2), we can express the response surface as:

$$\mathbf{U}(\mathbf{x},t) = \sum_{enr=1}^{\infty} [\mathbf{u}_s(\mathbf{x}) \otimes u_t(t)]^{(enr)} \tag{3.15}$$

where $\mathbf{u}_s$ is a spatial vector function, u_t a temporal scalar function and "$\otimes$" represents the tensor product of functionals.

From Eq. (3.15), the expressions for the velocity and acceleration field become:

$$\dot{\mathbf{U}}(\mathbf{x},t) = \sum_{enr=1}^{\infty} [\mathbf{u}_s(\mathbf{x}) \otimes \dot{u}_t(t)]^{(enr)} \tag{3.16}$$

$$\ddot{\mathbf{U}}(\mathbf{x},t) = \sum_{enr=1}^{\infty} [\mathbf{u}_s(\mathbf{x}) \otimes \ddot{u}_t(t)]^{(enr)} \tag{3.17}$$

Equivalently to the response parameter fields, a space-time decomposition for the excitation vector field can also be defined:

$$\mathbf{F}(\mathbf{x},t) = \sum_{enr=1}^{\infty} [\mathbf{f}_s(\mathbf{x}) \otimes f_t(t)]^{(enr)} \tag{3.18}$$

The system stiffness represents a functional linear operator, mapping the space of displacements onto the space of force fields. Considering a purely static problem, it provides the resulting force at a position $\mathbf{x}_1$ from a displacement imposed at a position $\mathbf{x}_2$. It may be viewed as the Green's function (or kernel) $(\mathbf{x},\mathbf{x}',t) \mapsto K(\mathbf{x},\mathbf{x}',t)$ of a functional operator $u \mapsto f$, such that:

$$f(\mathbf{x}_1,t) = \int_{\Omega} K(\mathbf{x}_1,\mathbf{x}_2,t)u(\mathbf{x}_2,t) \tag{3.19}$$

Therefore, in the space-time framework, the stiffness operator can be viewed as a rank three tensor that is decomposed into space and time components as follows:

$$K(\mathbf{x}_1,\mathbf{x}_2,t) = \sum_{enr=1}^{\infty} [K_s(\mathbf{x}_1,\mathbf{x}_2) \otimes k_t(t)]^{(enr)} \tag{3.20}$$

Space-time decompositions for the mass and damping operators are expressed equivalently to the stiffness.

3.2.2 Discrete Space-Time Decomposition

In its discrete representation, the space-time response solution is sought on a structured mesh with n_s degrees of freedom and n_t time instants. It is represented by a $n_s \times n_t$ matrix $\mathbf{U}$:

$$\mathbf{U} = \left[\mathbf{u}_1,\ldots,\mathbf{u}_{n_t}\right] \in \mathbb{R}^{n_s \times n_t} \tag{3.21}$$

The discrete counterparts of the spatial and temporal functions are represented by the column vectors:

$$\mathbf{u}_s \in \mathbb{R}^{n_s}, \mathbf{u}_t \in \mathbb{R}^{n_t} \tag{3.22}$$

Thus, using the PGD approximation (Eq. (3.2)), Eq. (3.15) in its discrete form is written as the summation of N tensor (or dyadic) products:

$$\mathbf{U}^{(N)} = \sum_{enr=1}^{N} \mathbf{u}_s^{(enr)} \otimes \mathbf{u}_t^{(enr)} \tag{3.23}$$

where the discrete counterpart of the tensor product in each summand is the tensor product of vectors or dyadic product (see Eq. (A.9)). The number of summands, N, required for an exact representation of $\mathbf{U}$ by the sum of dyadic vector products can be upper-bounded by the number of spatial degrees of freedom times the number of time steps, i.e. $E \leq n_s \times n_t$. It is the ultimate goal of the PGD to yield an approximation of the space-time displacement matrix, while preserving the required level of accuracy.

Equivalently to Eq. (3.16) and 3.17, the approximations for the discrete representations of the velocity and acceleration response fields are written as:

$$\dot{\mathbf{U}}^{(N)} = \sum_{enr=1}^{N} \mathbf{u}_s^{(enr)} \otimes \dot{\mathbf{u}}_t^{(enr)} \tag{3.24}$$

$$\ddot{\mathbf{U}}^{(N)} = \sum_{enr=1}^{N} \mathbf{u}_s^{(enr)} \otimes \ddot{\mathbf{u}}_t^{(enr)} \tag{3.25}$$

As discussed in 3.2.1, the stiffness operator in the space-time framework is viewed as a rank three tensor[3], that depends on two spatial variables and one temporal variable. The discrete counterparts of functions $K_s(\mathbf{x},\mathbf{x}')$ and $k_t(t)$ are:

$$\mathbf{K}_s \in \mathbb{R}^{n_s \times n_s}, \mathbf{k}_t \in \mathbb{R}^{n_t} \tag{3.26}$$

and the discrete representation of their tensor product is defined as:

$$\mathbf{K}_s \otimes \mathbf{k}_t \in \mathbb{R}^{n_s \times n_s \times n_t} \tag{3.27}$$

The space-time stiffness is therefore given by

$$\underline{\underline{\mathbf{K}}} = \sum_{enr=1}^{N} \mathbf{K}_s^{(enr)} \otimes \mathbf{k}_t^{(enr)} \tag{3.28}$$

In the same manner, the space-time mass and space-time Rayleigh damping are defined respectively as:

$$\underline{\underline{\mathbf{M}}} = \sum_{enr=1}^{N} \mathbf{M}_s^{(enr)} \otimes \mathbf{m}_t^{(enr)} \tag{3.29}$$

$$\underline{\underline{\mathbf{C}}} = \sum_{enr=1}^{N} \mathbf{C}_s^{(enr)} \otimes \mathbf{c}_t^{(enr)} \tag{3.30}$$

Regarding linear elastodynamics, stiffness, mass and consequently damping operators are time invariant. Therefore, the 3rd order tensors are computed by only one dyadic product summand.

$$\underline{\underline{\mathbf{K}}} = \mathbf{K}_s \otimes \mathbf{k}_t \tag{3.31}$$

$$\underline{\underline{\mathbf{M}}} = \mathbf{M}_s \otimes \mathbf{m}_t \tag{3.32}$$

$$\underline{\underline{\mathbf{C}}} = \mathbf{C}_s \otimes \mathbf{c}_t \tag{3.33}$$

[3]The underline convention used for the tensors is omitted for all tensors of rank 2 and below.

For this special case, the temporal functions of the system operators are reduced to a constant vector and thus the spatial functions are equal to the well known stiffness, mass and damping matrices:

$$\mathbf{k}_t = \mathbf{m}_t = \mathbf{c}_t = \texttt{ones}(n_t, 1) \tag{3.34}$$

$$\mathbf{K}_s \equiv \mathbf{K}, \quad \mathbf{M}_s \equiv \mathbf{M}, \quad \mathbf{C}_s \equiv \mathbf{C} \tag{3.35}$$

where $\texttt{ones}(n_t, 1)$ denotes a column vector in $\mathbb{R}^{n_t}$, whose components are all equal to one.

Finally, the equation of motion Eq. (2.19) can be written for the solution approximation, under the space-time formalism, as:

$$\begin{aligned}
&(\mathbf{M}_s \otimes \mathbf{m}_t)\left(\sum_{enr=1}^{N} \mathbf{u}_s^{(enr)} \otimes \ddot{\mathbf{u}}_t^{(enr)}\right) + \\
&(\mathbf{C}_s \otimes \mathbf{c}_t)\left(\sum_{enr=1}^{N} \mathbf{u}_s^{(enr)} \otimes \dot{\mathbf{u}}_t^{(enr)}\right) + \\
&(\mathbf{K}_s \otimes \mathbf{k}_t)\left(\sum_{enr=1}^{N} \mathbf{u}_s^{(enr)} \otimes \mathbf{u}_t^{(enr)}\right) = \mathbf{F}_{ext}
\end{aligned} \tag{3.36}$$

where $\mathbf{F}_{ext} = [\mathbf{f}_{ext,1} ... \mathbf{f}_{ext,n_t}]$. Since mass, damping and stiffness terms are time invariant, they can be moved inside the respective sums. Then, employing property Eq. (A.10) leads to:

$$\begin{aligned}
&\sum_{enr=1}^{N} \left(\mathbf{M}_s \mathbf{u}_s^{(enr)}\right) \otimes \left(\mathbf{m}_t \circ \ddot{\mathbf{u}}_t^{(enr)}\right) + \\
&\sum_{enr=1}^{N} \left(\mathbf{C}_s \mathbf{u}_s^{(enr)}\right) \otimes \left(\mathbf{c}_t \circ \dot{\mathbf{u}}_t^{(enr)}\right) + \\
&\sum_{enr=1}^{N} \left(\mathbf{K}_s \mathbf{u}_s^{(enr)}\right) \otimes \left(\mathbf{k}_t \circ \mathbf{u}_t^{(enr)}\right) = \mathbf{F}_{ext}
\end{aligned} \tag{3.37}$$

Taking into account Eq. (3.34), we get:

$$\begin{aligned}
\mathbf{m}_t \circ \ddot{\mathbf{u}}_t &= \ddot{\mathbf{u}}_t \\
\mathbf{c}_t \circ \dot{\mathbf{u}}_t &= \dot{\mathbf{u}}_t \\
\mathbf{k}_t \circ \mathbf{u}_t &= \mathbf{u}_t
\end{aligned} \tag{3.38}$$

and thus, Eq. (3.37) is reduced into a simplified version that contains only rank 2 tensors:

$$\mathbf{M}_s \ddot{\mathbf{U}}^{(N)} + \mathbf{C}_s \dot{\mathbf{U}}^{(N)} + \mathbf{K}_s \mathbf{U}^{(N)} = \mathbf{F}_{ext} \tag{3.39}$$

In the next section, the $\square_s$ subscript can be omitted by employing Eq. (3.35) and the $\square^{(N)}$ superscript is omitted for simplicity.

3.3 Space-Time Newmark - Non Incremental Solution Approach

In this section, the Newmark scheme will be cast into the space-time framework via a tensorial formulation, first introduced (in a general manner) in [19] and lately implemented in [21]. Starting from the discretized equation of motion, the residual force vector[4], corresponding to a particular time step j, is defined as:

$$\mathbf{r}(\mathbf{u}_{j+1}) = \mathbf{M}\ddot{\mathbf{u}}_{j+1} + \mathbf{C}\dot{\mathbf{u}}_{j+1} + \mathbf{K}\mathbf{u}_{j+1} - \mathbf{f}_{ext,j+1} \tag{3.40}$$

where $j = 0, ..., n_t - 1$ is the current time step.

Since we are dealing with linear analyses, the system matrices are constant and independent of the configuration of the structure. We can thus express a matrix column-wise composed of the residual vectors at all time steps as:

$$\mathbf{R} = \mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} - \mathbf{F}_{ext} \tag{3.41}$$

Next, the Newmark scheme will be applied, enabling us to express the space-time velocity, $\dot{\mathbf{U}}$, and acceleration, $\ddot{\mathbf{U}}$, matrices as functions of the displacements $\mathbf{U}$ and the initial conditions, $\mathbf{u}_0, \dot{\mathbf{u}}_0$. First, for the sake of simplicity, we change the time step notation: $\mathbf{u}(t + \Delta t) \equiv \mathbf{u}_{j+1}$, and so the Newmark equations Eq. (2.31) and Eq. (2.30) are rearranged as:

$$(\dot{\mathbf{u}}_{j+1} - \dot{\mathbf{u}}_j) + \left(-\gamma\Delta t\ddot{\mathbf{u}}_{j+1} + (\gamma - 1)\Delta t\ddot{\mathbf{u}}_j\right) = \mathbf{0} \tag{3.42}$$

$$(\mathbf{u}_{j+1} - \mathbf{u}_j) + (-\Delta t\dot{\mathbf{u}}_j) + \left(-\beta\Delta t^2\ddot{\mathbf{u}}_{j+1} + \left(\beta - \frac{1}{2}\right)\Delta t^2\ddot{\mathbf{u}}_j\right) = \mathbf{0} \tag{3.43}$$

In order to formulate the space-time Newmark scheme, the following linear combination of all couples of successive displacement vectors:

$$\Delta\mathbf{u}_{j+1}(c_1, c_2) = c_1\mathbf{u}_{j+1} + c_2\mathbf{u}_j \tag{3.44}$$

with $j = 0, ..., n_t - 1$, must be expressed in matrix form as:

$$\Delta\mathbf{U}(c_1, c_2) = \left[\Delta\mathbf{u}_1(c_1, c_2) \cdots \Delta\mathbf{u}_{n_t}(c_1, c_2)\right] = \mathbf{U}\mathbf{A}^T(c_1, c_2) + \mathbf{u}_0\boldsymbol{\alpha}^T(c_2) \tag{3.45}$$

where the coefficients $c_1, c_2 \in \mathbb{R}$ depend on the Newmark constants and the time step. The auxiliary coefficient matrices are defined as:

$$\mathbf{A}(c_1, c_2) = \begin{bmatrix} c_1 & 0 & 0 & 0 & \dots & 0 \\ c_2 & c_1 & 0 & 0 & \dots & 0 \\ 0 & c_2 & c_1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ 0 & \dots & 0 & c_2 & c_1 & 0 \\ 0 & \dots & 0 & 0 & c_2 & c_1 \end{bmatrix} \in \mathbb{R}^{n_t \times n_t} \tag{3.46}$$

[4] Since the exact solution is approximated, a residual internal force naturally appears.

$$\boldsymbol{\alpha}(c_2) = \begin{bmatrix} c_2 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{n_t} \tag{3.47}$$

Space-time equivalents for Eq. (3.42), Eq. (3.43) are also expressed by employing Eq. (3.45):

$$\Delta\dot{\mathbf{U}}(1,-1) + \Delta\ddot{\mathbf{U}}(-\gamma\Delta t, (\gamma-1)\Delta t) = \mathbf{0} \tag{3.48}$$

$$\Delta\mathbf{U}(1,-1) + \Delta\dot{\mathbf{U}}(0,-\Delta t) + \Delta\ddot{\mathbf{U}}\left(-\beta\Delta t^2, (\beta-0.5)\Delta t^2\right) = \mathbf{0} \tag{3.49}$$

With the aid of Eq. (3.46) and Eq. (3.47), we define some specific auxiliary "Alpha" matrices and vectors as:

$$\begin{array}{ll} \mathbf{A}_1 = \mathbf{A}(1,-1) & \boldsymbol{\alpha}_1 = \boldsymbol{\alpha}(-1) \\ \mathbf{A}_2 = \mathbf{A}(0,-\Delta t) & \boldsymbol{\alpha}_2 = \boldsymbol{\alpha}(-\Delta t) \\ \mathbf{A}_3 = \mathbf{A}\left(-\beta\Delta t^2, (\beta-0.5)\Delta t^2\right) & \boldsymbol{\alpha}_3 = \boldsymbol{\alpha}\left((\beta-0.5)\Delta t^2\right) \\ \mathbf{A}_4 = \mathbf{A}(-\gamma\Delta t, (\gamma-1)\Delta t) & \boldsymbol{\alpha}_4 = \boldsymbol{\alpha}((\gamma-1)\Delta t) \end{array} \tag{3.50}$$

Hence, Eq. (3.48) and Eq. (3.49) take the final form:

$$\dot{\mathbf{U}}\mathbf{A}_1^T + \dot{\mathbf{u}}_0\boldsymbol{\alpha}_1^T + \ddot{\mathbf{U}}\mathbf{A}_4^T + \ddot{\mathbf{u}}_0\boldsymbol{\alpha}_4^T = \mathbf{0} \tag{3.51}$$

$$\mathbf{U}\mathbf{A}_1^T + \mathbf{u}_0\boldsymbol{\alpha}_1^T + \dot{\mathbf{U}}\mathbf{A}_2^T + \dot{\mathbf{u}}_0\boldsymbol{\alpha}_2^T + \ddot{\mathbf{U}}\mathbf{A}_3^T + \ddot{\mathbf{u}}_0\boldsymbol{\alpha}_3^T = \mathbf{0} \tag{3.52}$$

where the initial acceleration is computed from:

$$\mathbf{M}\ddot{\mathbf{u}}_0 = \mathbf{f}_{ext,0} - \mathbf{C}\dot{\mathbf{u}}_0 - \mathbf{K}\mathbf{u}_0 \tag{3.53}$$

Inserting Eq. (3.51) and Eq. (3.52) in Eq. (3.40) for the residual matrix and performing some grouping operations, leads to a set of algebraic equations that only depend on the displacements $\mathbf{U}$:

$$\mathbf{R}(\mathbf{U}) = \mathbf{M}\mathbf{U}\mathbf{Y} - \mathbf{C}\mathbf{U}\mathbf{W} + \mathbf{K}\mathbf{U} - \mathbf{L} = \mathbf{0} \tag{3.54}$$

In the above equations, matrices $\mathbf{Y}, \mathbf{W}, \mathbf{L}$ are defined as:

$$\mathbf{Y} = \mathbf{A}_1^T\mathbf{D}^{-1}\mathbf{H}, \quad \mathbf{W} = \mathbf{A}_1^T\mathbf{D}^{-1} \tag{3.55}$$

$$\begin{aligned}\mathbf{L} = \mathbf{F}_{ext} + \mathbf{M}\left[-\mathbf{u}_0\boldsymbol{\alpha}_1^T + \dot{\mathbf{u}}_0\left(\boldsymbol{\alpha}_1^T\left(\mathbf{A}_4^T\right)^{-1}\mathbf{A}_3^T - \boldsymbol{\alpha}_2^T\right) + \ddot{\mathbf{u}}_0\left(\boldsymbol{\alpha}_4^T\left(\mathbf{A}_4^T\right)^{-1}\mathbf{A}_3^T - \boldsymbol{\alpha}_3^T\right)\right]\mathbf{D}^{-1}\mathbf{H} \\ + \mathbf{M}\left[\dot{\mathbf{u}}_0\boldsymbol{\alpha}_1^T\left(\mathbf{A}_4^T\right)^{-1} + \ddot{\mathbf{u}}_0\boldsymbol{\alpha}_4^T\left(\mathbf{A}_4^T\right)^{-1}\right] \\ - \mathbf{C}\left[-\mathbf{u}_0\boldsymbol{\alpha}_1^T + \dot{\mathbf{u}}_0\left(\boldsymbol{\alpha}_1^T\left(\mathbf{A}_4^T\right)^{-1}\mathbf{A}_3^T - \boldsymbol{\alpha}_2^T\right) + \ddot{\mathbf{u}}_0\left(\boldsymbol{\alpha}_4^T\left(\mathbf{A}_4^T\right)^{-1}\mathbf{A}_3^T - \boldsymbol{\alpha}_3^T\right)\right]\mathbf{D}^{-1}\end{aligned} \quad (3.56)$$

where:

$$\mathbf{D} = \mathbf{A}_2^T - \mathbf{A}_1^T\left(\mathbf{A}_4^T\right)^{-1}\mathbf{A}_3^T, \quad \mathbf{H} = \mathbf{A}_1^T\left(\mathbf{A}_4^T\right)^{-1} \quad (3.57)$$

In the PGD framework (Sec.3.1.2), the space-time response solution at the m^{th} enrichment, $\mathbf{U}^{(m)}$ is divided into the already evaluated $m-1$ known enrichments and the current m^{th} summand to be calculated:

$$\mathbf{U}^{(m)} = \sum_{enr=1}^{m-1}\mathbf{u}_s^{(enr)} \otimes \mathbf{u}_t^{(enr)} + \mathbf{u}_s^{(m)} \otimes \mathbf{u}_t^{(m)} = \mathbf{U}^{(m-1)} + \mathbf{u}_s^{(m)} \otimes \mathbf{u}_t^{(m)} \quad (3.58)$$

Substituting Eq. (3.58) into the space-time equation of motion Eq. (3.54) yields:

$$\mathbf{M}\left[\mathbf{U}^{(m-1)} + \mathbf{u}_s^{(m)} \otimes \mathbf{u}_t^{(m)}\right]\mathbf{Y} - \mathbf{C}\left[\mathbf{U}^{(m-1)} + \mathbf{u}_s^{(m)} \otimes \mathbf{u}_t^{(m)}\right]\mathbf{W} + \mathbf{K}\left[\mathbf{U}^{(m-1)} + \mathbf{u}_s^{(m)} \otimes \mathbf{u}_t^{(m)}\right] - \mathbf{L} = \mathbf{0} \quad (3.59)$$

The above equation Eq. (3.59) has two unknowns, the spatial enrichment vector $\mathbf{u}_s^m$ and the temporal enrichment vector $\mathbf{u}_t^m$.

For the presented formulation, the Fixed-Point Algorithm with Picard iterations [24] is used to compute each spatial and temporal enrichment. The solution procedure has an inherent recursive nature, so in reality, iterations are performed until convergence is achieved and a new set of spatial and temporal modes is acquired. In Fig. 3.2 a $1D$ illustration of the fixed-point algorithm to compute a coupled pair of unknown functions is given.

Within the fixed-point algorithm solution procedure, at a particular iteration k, multiplying the left hand side of Eq. (3.59), which would be $\mathbf{R}\left(\mathbf{U}^{(m-1)} + {}^{k}\mathbf{u}_s^{(m)} \otimes {}^{k-1}\mathbf{u}_t^{(m)}\right)$, by $\left({}^{k-1}\mathbf{u}_t^{(m)}\right)$ leads to a projection into space, which produces the *Space Problem*, whereas multiplying Eq. (3.59) by $\left({}^{k}\mathbf{u}_s^{(m)}\right)$ leads to a projection into time, which produces the *Time Problem*.

The solution of the *Space Problem* (line 7 of Algorithm 2), is computed by solving the following linear system:

$$\mathbf{E}_s\left({}^{k}\mathbf{u}_s^{(m)}\right) = \mathbf{b}_s \quad (3.60)$$

with the corresponding coefficient matrix and right hand side written as:

$$\begin{aligned}\mathbf{E}_s = &\left(\left({}^{k-1}\mathbf{u}_t^{(m)}\right)^T\mathbf{Y}\left({}^{k-1}\mathbf{u}_t^{(m)}\right)\right)\mathbf{M} \\ &- \left(\left({}^{k-1}\mathbf{u}_t^{(m)}\right)^T\mathbf{W}\left({}^{k-1}\mathbf{u}_t^{(m)}\right)\right)\mathbf{C} \\ &+ \left(\left({}^{k-1}\mathbf{u}_t^{(m)}\right)^T\left({}^{k-1}\mathbf{u}_t^{(m)}\right)\right)\mathbf{K}\end{aligned} \quad (3.61)$$

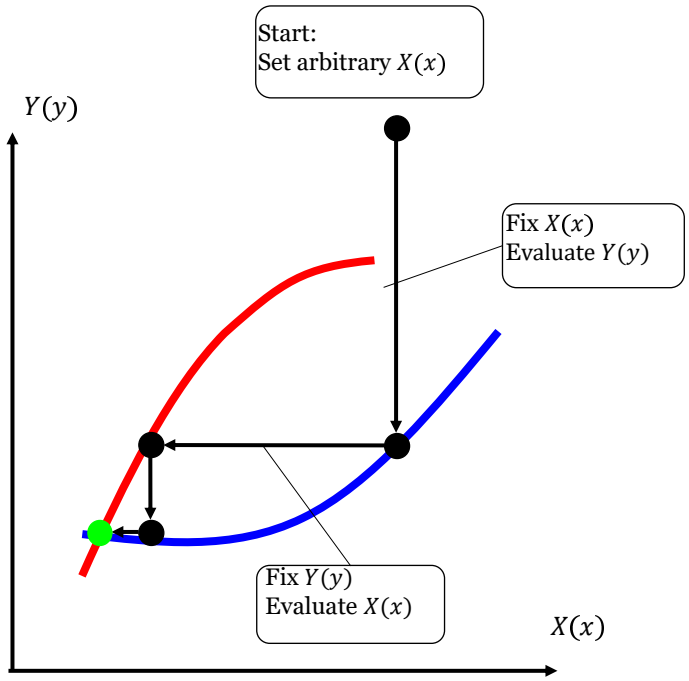


Figure 3.2: $1D$ illustration of the fixed-point algorithm

$$\mathbf{b}_s = \left(\mathbf{L} - \mathbf{M}\mathbf{U}^{(m-1)}\mathbf{Y} + \mathbf{C}\mathbf{U}^{(m-1)}\mathbf{W} - \mathbf{K}\mathbf{U}^{(m-1)}\right)\left({}^{k-1}\mathbf{u}_t^{(m)}\right) \tag{3.62}$$

In a similar manner, the solution of the *Time Problem* (line 8 of Algorithm 2), is computed by solving the following linear system:

$$\mathbf{E}_t\left({}^{k}\mathbf{u}_t^{(m)}\right) = \mathbf{b}_t \tag{3.63}$$

with the corresponding coefficient matrix and right hand side written as:

$$\begin{aligned}\mathbf{E}_t = &\left(\left({}^{k}\mathbf{u}_s^{(m)}\right)^T \mathbf{M}\left({}^{k}\mathbf{u}_s^{(m)}\right)\right)\mathbf{Y}^T \\ &-\left(\left({}^{k}\mathbf{u}_s^{(m)}\right)^T \mathbf{C}\left({}^{k}\mathbf{u}_s^{(m)}\right)\right)\mathbf{W}^T \\ &+\left(\left({}^{k}\mathbf{u}_s^{(m)}\right)^T \mathbf{K}\left({}^{k}\mathbf{u}_s^{(m)}\right)\right)\mathbf{I}_t\end{aligned} \tag{3.64}$$

$$\mathbf{b}_t = \left(\mathbf{L} - \mathbf{M}\mathbf{U}^{(m-1)}\mathbf{Y} + \mathbf{C}\mathbf{U}^{(m-1)}\mathbf{W} - \mathbf{K}\mathbf{U}^{(m-1)}\right)\left({}^{k}\mathbf{u}_s^{(m)}\right) \tag{3.65}$$

where $\mathbf{I}_t \in \mathbb{R}^{n_t \times n_t}$ is the temporal identity matrix.

Next, a pseudo-algorithm for the Newmark-PGD non-incremental solution procedure is provided (Algorithm 2).

Algorithm 2 PGD Solution Scheme

INPUT: $\mathbf{K}, \mathbf{M}, \mathbf{C}, \mathbf{f}$

OUTPUT: $\mathbf{U}$

select: solver parameters $\leftarrow m_{max}, k_{max}, \epsilon_{fp}, \epsilon_{enr}$
select: initialization case ▷ Different cases investigated in Sec.4.2
initialize: space and time modes $\leftarrow \mathbf{U}^{(0)} = \mathbf{u}_s^{(0)} \otimes \mathbf{u}_t^{(0)}$
for $m = 1 : m_{max}$ **do** ▷ ENTER enrichment loop
    initialize: ${}^0\mathbf{u}_s^{(m)} \leftarrow {}^k\mathbf{u}_s^{(m-1)}$, ${}^0\mathbf{u}_t^{(m)} \leftarrow {}^k\mathbf{u}_t^{(m-1)}$
    for $k = 1 : k_{max}$ **do** ▷ ENTER fixed point loop
        ${}^k\mathbf{u}_s^{(m)} \leftarrow$ solve: Eq. (3.60)
        ${}^k\mathbf{u}_t^{(m)} \leftarrow$ solve: Eq. (3.63)
        if $\|{}^k\mathbf{U}^{(m)} - {}^{k-1}\mathbf{U}^{(m)}\|_2 < \epsilon_{fp}$ **then** ▷ Check f.p. convergence
            $\mathbf{u}_s^{(m)} := {}^k\mathbf{u}_s^{(m)}$
            $\mathbf{u}_t^{(m)} := {}^k\mathbf{u}_t^{(m)}$
            break;
        end if
    end for ▷ EXIT fixed point loop
    if $\|\mathbf{R}(\mathbf{U}^{(m)})\|_2 < \epsilon_{enr}$ **then** ▷ Check enrichment convergence
        break;
    end if
end for ▷ EXIT enrichment loop

Chapter 4

NUMERICAL SIMULATIONS

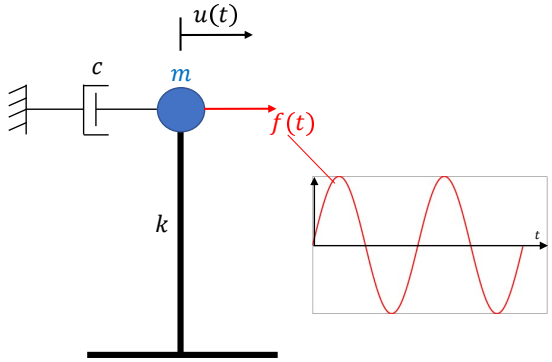
In this chapter, several numerical benchmarks are presented to validate the performance of the space-time PGD framework, in terms of robustness, accuracy and efficiency, but also to point out the limits of the presented methodology.

4.1 Single-Degree-of-Freedom System

As a first example, the response of a single-degree-of-freedom (SDOF) system is analyzed, considering three different excitation scenarios. First a harmonically excited system (Case 1) and then a system subjected to an impulsive load (Case 2). All results are compared with the solution acquired via the conventional Newmark method and, for the special cases where it exists, with an analytical solution.

4.1.1 Harmonic Excitation

First, a harmonically excited SDOF oscillator is examined. The model and excitation properties are presented in Fig. 4.1.



Model Properties	Excitation Properties
$m = 1$ [t]	$u(0) = 0$
$k = 4\pi^2$ [kN/m]	$\dot{u}(0) = 0$
$T = 1$ [sec], $\omega_n = 1$ [rad]	$f_0 = 10$ [kN/m], $\omega_p = 2\omega_n$
$\xi = 2\%$, $(c = 2m\omega_n\xi)$	$f(t) = f_0 \sin(\omega_p t)$

Figure 4.1: SDOF Case 1: Model

For this case of a damped SDOF system with ”at rest” initial conditions, there exists an analytical expression for the total solution which is given by:

$$u(t) = \underbrace{e^{\xi\omega_n t}\left[A\cos(\omega_D t) + B\sin(\omega_D t)\right]}_{u_{transient}} + \underbrace{\left[C\cos(\omega_p t) + D\sin(\omega_p t)\right]}_{u_{steady\ state}} \tag{4.1}$$

where:

$$\begin{aligned}
\omega_D &= \omega_n\sqrt{1-\xi^2} \\
C &= \frac{p_0}{k}\frac{1-(\omega_p/\omega_n)^2}{[1-(\omega_p/\omega_n)^2]^2 + [2\xi(\omega_p/\omega_n)]^2} \\
D &= \frac{p_0}{k}\frac{-2\xi(\omega_p/\omega_n)}{[1-(\omega_p/\omega_n)^2]^2 + [2\xi(\omega_p/\omega_n)]^2} \\
A &= -D \\
B &= \frac{A\xi - C(\omega_p/\omega_n)}{\sqrt{1-\xi^2}}
\end{aligned} \tag{4.2}$$

Consequently the analytical expressions for the velocity and acceleration are acquired with differentiation with respect to time as:

$$\begin{aligned}
\dot{u}(t) = &\underbrace{\xi\omega_n u_{transient} + \omega_D\left[-A\sin(\omega_D t) + B\cos(\omega_D t)\right]e^{\xi\omega_n t}}_{\dot{u}_{transient}} \\
&+ \underbrace{\omega_p\left[-C\sin(\omega_p t) + D\cos(\omega_p t)\right]}_{\dot{u}_{steady\ state}}
\end{aligned} \tag{4.3}$$

and

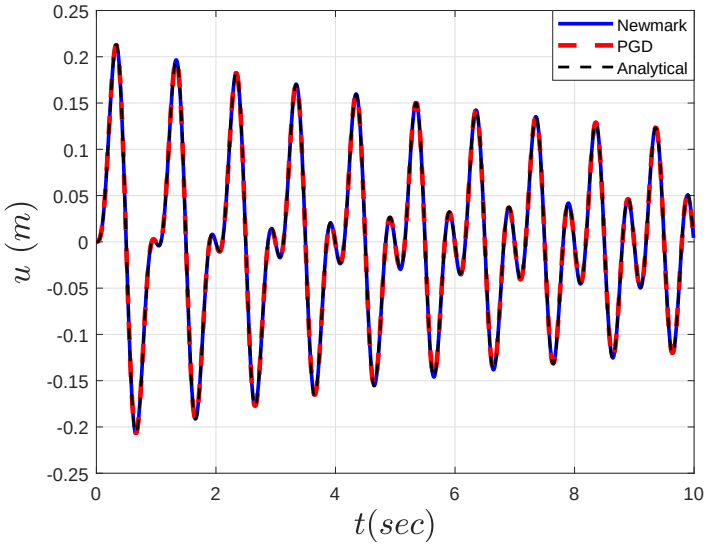
$$\ddot{u}(t) = \ddot{u}_{transient} + \ddot{u}_{steady\ state} \tag{4.4}$$

where

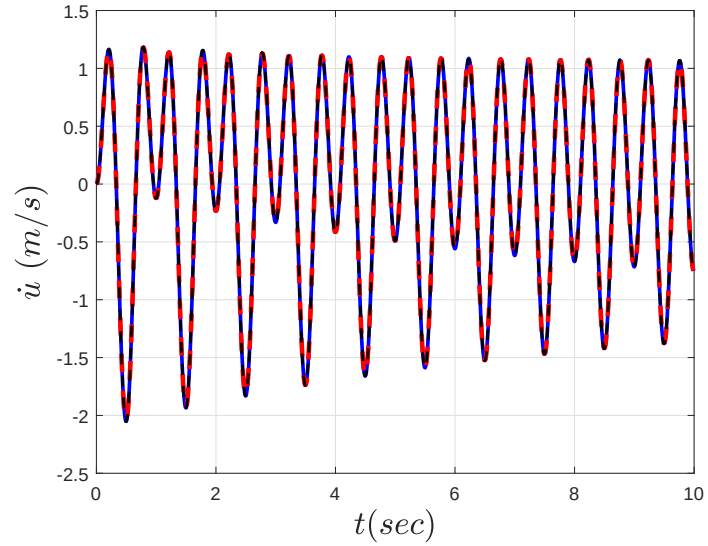
$$\begin{aligned}
\ddot{u}_{transient} = &\xi\omega_n\dot{u}_{transient} \\
&+ \xi\omega_n\omega_D\left[-A\sin(\omega_D t) + B\cos(\omega_D t)\right]e^{\xi\omega_n t} \\
&+ \omega_D^2\left[-A\cos(\omega_D t) - B\sin(\omega_D t)\right]e^{\xi\omega_n t}
\end{aligned} \tag{4.5}$$

$$\ddot{u}_{steady\ state} = \omega_p^2\left[-C\cos(\omega_p t) - D\sin(\omega_p t)\right] \tag{4.6}$$

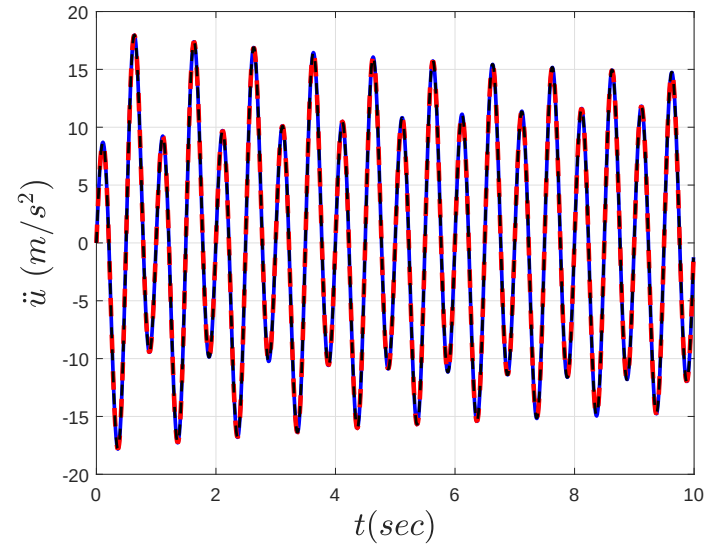
Fig. 4.2 depicts the comparison of the computed numerical response via the Newmark method and the PGD versus the analytical solution. As expected, only one enrichment with two fixed point iterations is enough to compute the response of the SDOF system with sufficient accuracy. The $L2-$norms of the displacement, velocity and acceleration errors, between the Newmark and the PGD solution, are respectively: $8.75e-09$, $5.42e-08$, $1.74e-06$.



(a) Displacement



(b) Velocity

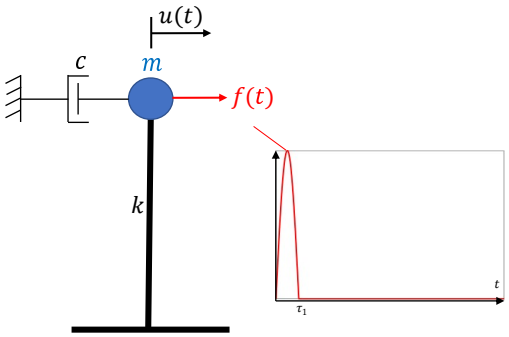


(c) Acceleration

Figure 4.2: SDOF Case 1: Response

4.1.2 Impulsive Excitation

In this subsection, the SDOF model, depicted in Fig. 4.3 is analyzed.



Model Properties	Excitation Properties
$m = 1$ [t]	$u(0) = 0$
$k = 4\pi^2 \left[\mathrm{kN/m}\right]$	$\dot{u}(0) = 0$
$T = 1$ [sec], $\omega_n = 1$ [rad/sec]	$f_0 = 10\left[\mathrm{kN/m}\right]$, $\tau_1 = T$, $\omega_p = \pi$
$\xi = 0$	$f(t) = f_0 \sin(\omega_p t), t \in [0, \tau_1]$
	$f(t) = 0 \quad , t \in (\tau_1, 10]$

Figure 4.3: SDOF Case 2: Model

For this case of an undamped SDOF system, subjected to an impulsive load (e.g. blast load), the analytical solution for the displacement assumes the following form:

$$u(t) = \begin{cases} \overbrace{u_0 \cos(\omega_n t) + \left(\frac{\dot{u}_0}{\omega_n} - \frac{p_0}{k}\frac{\rho}{1-\rho^2}\right)\sin(\omega_n t)}^{u_{transient}} + \overbrace{\frac{p_0}{k}\frac{1}{1-\rho^2}\sin(\omega_p t)}^{u_{steady\ state}} & , 0 \le t \le \tau_1 \\ u(\tau_1)\cos(\omega_n t) + \frac{\dot{u}(\tau_1)}{\omega_n}\sin(\omega_n t) & , \tau_1 \le t \end{cases} \tag{4.7}$$

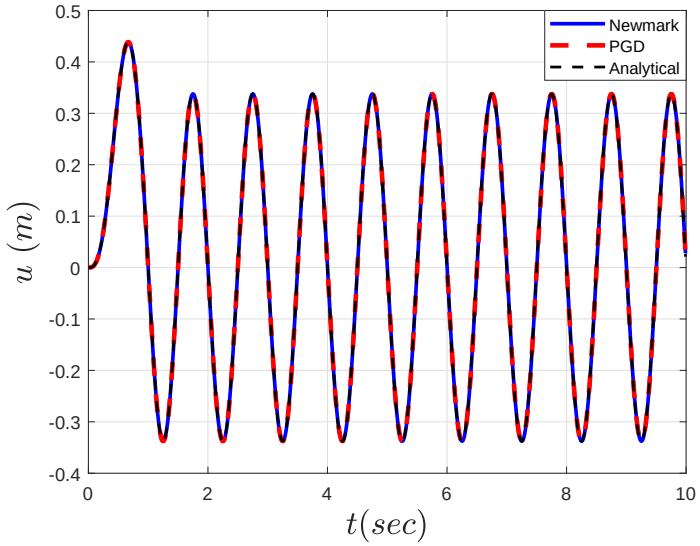
where $\rho = \omega_p/\omega_n$

Consequently, the velocity and acceleration are respectively:

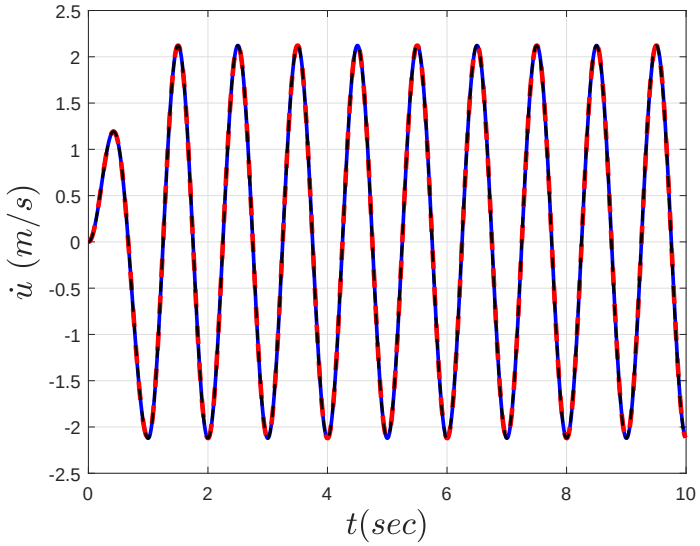
$$\dot{u}(t) = \begin{cases} \overbrace{\omega_n\left(-u_0 \sin(\omega_n t) + \left(\frac{\dot{u}_0}{\omega_n} - \frac{p_0}{k}\frac{\rho}{1-\rho^2}\right)\cos(\omega_n t)\right)}^{\dot{u}_{transient}} + \overbrace{\omega_p \frac{p_0}{k}\frac{1}{1-\rho^2}\cos(\omega_p t)}^{\dot{u}_{steady\ state}} & , 0 \le t \le \tau_1 \\ \omega_n\left(-u(\tau_1)\sin(\omega_n t) + \frac{\dot{u}(\tau_1)}{\omega_n}\cos(\omega_n t)\right) & , \tau_1 \le t \end{cases} \tag{4.8}$$

$$\ddot{u}(t) = \begin{cases} \overbrace{-\omega_n^2 u_{transient}}^{\ddot{u}_{transient}} + \overbrace{\left(-\omega_p^2 u_{steady\ state}\right)}^{\ddot{u}_{steady\ state}} & , 0 \le t \le \tau_1 \\ -\omega_n^2\left(u(\tau_1)\cos(\omega_n t) + \frac{\dot{u}(\tau_1)}{\omega_n}\sin(\omega_n t)\right) & , \tau_1 \le t \end{cases} \tag{4.9}$$

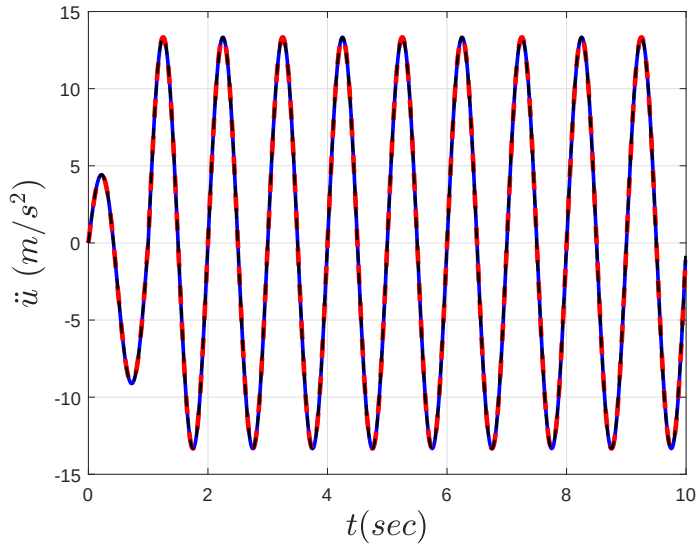
The results are shown in Fig. 4.4. Once again, only one enrichment with two Picard iterations yields an accurate response. The $L2-$norms of the displacement, velocity and acceleration errors, between the Newmark and the PGD solution, are respectively: $3.21e-08$, $2.10e-07$, $1.05e-05$.



(a) Displacement



(b) Velocity

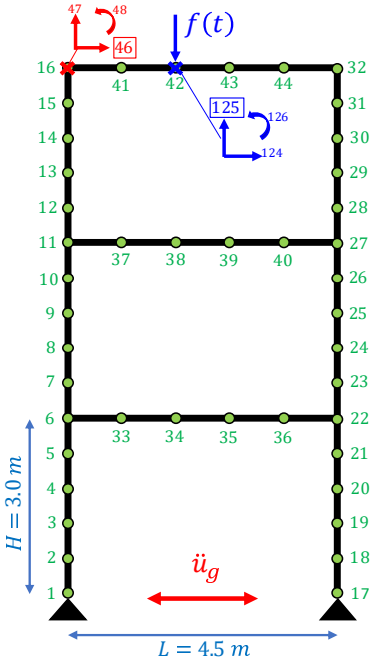


(c) Acceleration

Figure 4.4: SDOF Case 2: Response

4.2 2D Frame

In this section, the three-storey single-bay $2D$ frame, depicted in Fig. 4.5, is analyzed. The three excitation scenarios presented in the bottom table of Fig. 4.5 are considered. Both beams and columns are discretized using five Euler-Bernoulli $2D$ beam elements. The first six eigenmodes of the structure are shown in Fig. 4.6.



Model Properties
$\rho = 7850\,[\mathrm{kg/m^3}]$
$E = 210\,[GPa]$
Section: RHS 300x100/10 , $A = 76e-04\ [m^2]$

Excitation Scenarios
Case 1: $\ddot{u}_g(t) = 0.1\sin(\omega_p t), \quad \omega_p = \omega_1$
Case 2: $f(t) = 2 \cdot 10^4 \sin(10\pi t),\ t \in [0,0.1]$
Case 3: $\ddot{u}_g(t) \rightarrow$ Kobe earthquake

Figure 4.5: $2D$ Frame: Model and excitation cases

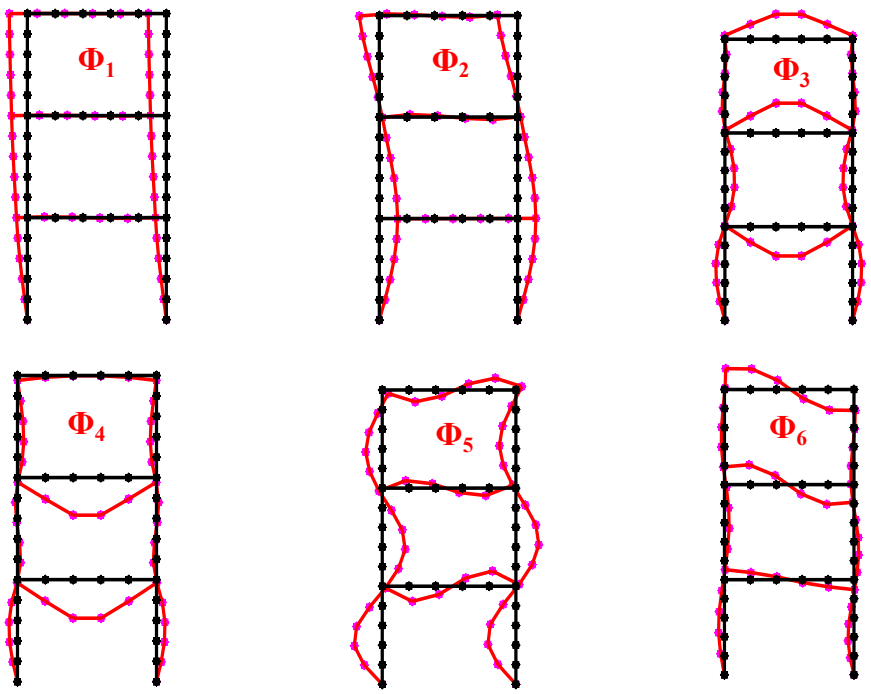


Figure 4.6: $2D$ Frame: Eigenmodes $1-6$

The first 6 corresponding eigenfrequencies are given below:

$$\begin{pmatrix} \omega_1 & = 28.988 \\ \omega_2 & = 121.055 \\ \omega_3 & = 259.626 \\ \omega_4 & = 355.481 \\ \omega_5 & = 407.517 \\ \omega_6 & = 431.006 \end{pmatrix} \tag{4.10}$$

An important aspect of the algorithmic implementation of the formulation presented in Sec.3.3 is the initial "guess" that is provided (line 2 of Algorithm 2). In this section, several different cases of spatial and temporal PGD mode initialization will be tested, in order to investigate their influence on the overall performance of the space-time framework. The only limitation for the initialization modes is that their norms must not be zero, in order for the algorithm to begin the iterations (without any singularities).

4.2.1 Harmonic Excitation-Resonance

First, the $2D$ frame of Fig. 4.5 is harmonically excited. The ground acceleration, depicted in Fig. 4.7, has a frequency equal to the first eigenfrequency of the structure. The time step is chosen equal to: $dt = 0.01$ [sec]. As expected, the response of the structure will be resonant. For the mass matrix, the consistent mass approach is adopted and for the Rayleigh damping matrix of the structure, the damping ratios for the first two eigenmodes are chosen equal to 5%. For the result comparison, the horizontal displacement of the top left node is selected as a reference degree of freedom (see Fig. 4.5).

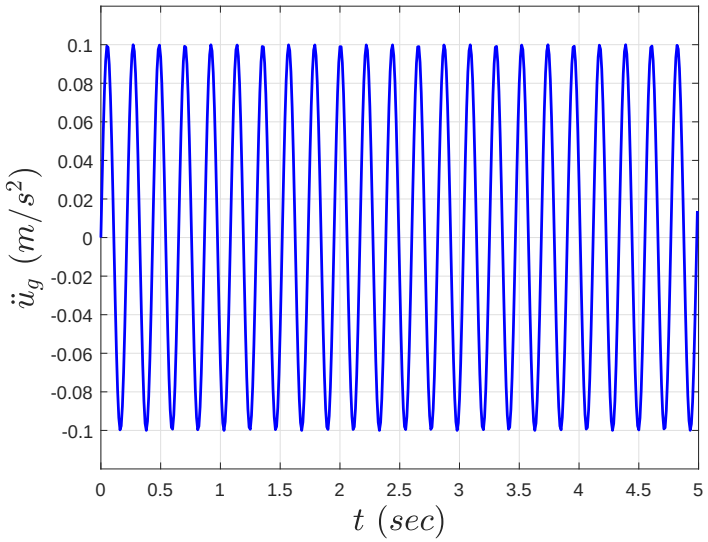


Figure 4.7: $2D$ Frame - Resonance: Ground acceleration

4.2.1.1 Case 1: *Arbitrary-1* Spatial and Temporal Initialization

For this excitation, three scenarios for the spatial and temporal initializations will be considered. The first combination, termed *Arbitrary-1*, is an arbitrary initialization for both the spatial and the temporal modes. *Arbitrary-1* initializations are considered as vectors with zero entries, apart from the first one which is taken equal to 10^{-c}, where $c = 10$.

Thus, for this particular model with 128 unconstrained degrees of freedom and the time domain discretized into 500 time steps, the initial spatial and temporal are given as:

$$\begin{aligned} \mathbf{u}_s^{(0)} &= \texttt{zeros(128,1)}, \text{ and } \mathbf{u}_s^{(0)}(1) = 10^{-10} \\ \mathbf{u}_t^{(0)} &= \texttt{zeros(500,1)}, \text{ and } \mathbf{u}_t^{(0)}(1) = 10^{-10} \end{aligned} \tag{4.11}$$

where `zeros(nL,nC)` denotes the MATLAB command for an array with `nL` lines, `nC` columns and all entries equal to zero.

Fig. 4.9 depicts the response obtained from the conventional incremental Newmark method and the approximate solution of the PGD with two enrichments. Also, for the sake of comparison, the solution obtained from a reduced basis approximation of the system, constructed using as many eigenvectors as enrichments performed, as basis vectors, is also depicted. It is evident that the first enrichment of the PGD does not yield any response, but the second enrichment shows very good accuracy. The only noticeable deviation of the PGD from the Newmark reference solution appears in the transient response part of the acceleration, which is also the case for the reduced modal basis approximation. As showcased in Fig. 4.9c, one more enrichment corrects this small error.

The convergence of the space-time residual, in terms of matrix norm, against the increasing number of enrichments is shown with the red line in Fig. 4.8. The blue line of Fig. 4.8. depicts the ratio of the time required for the PGD solution, over the time required for the incremental Newmark approach. An acceptable level accuracy is obtained by only two enrichments, accompanied by a speed-up of 2.0208 in terms of analysis runtime.

Table 4.1: 2*D* Frame Resonance-Case 1: Runtime Speed-up

Number of enrichments	$\frac{t_{\text{PGD}}}{t_{\text{Newmark}}}$	Speed-up
1	0.1911	5.2316
2	0.4949	2.0208
3	0.8103	1.2341

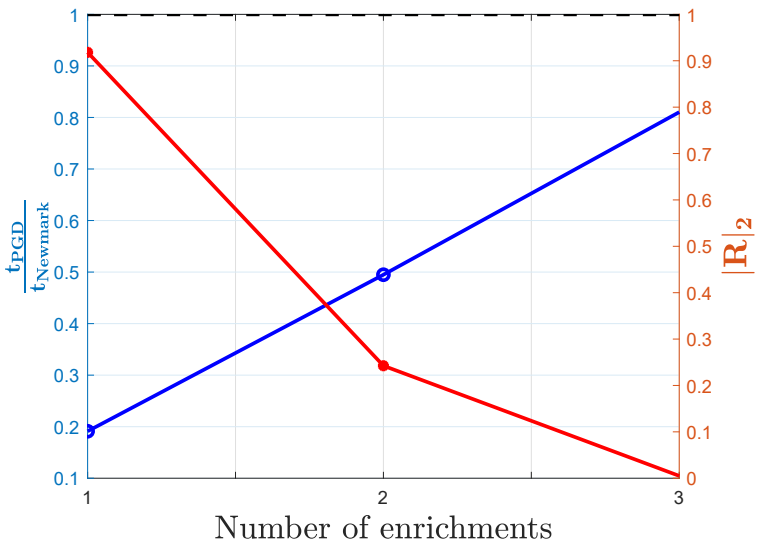
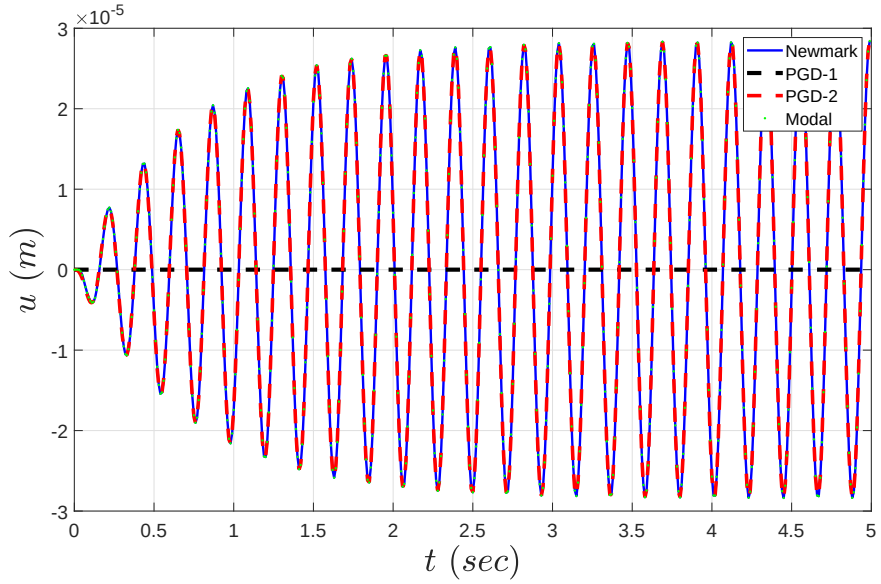
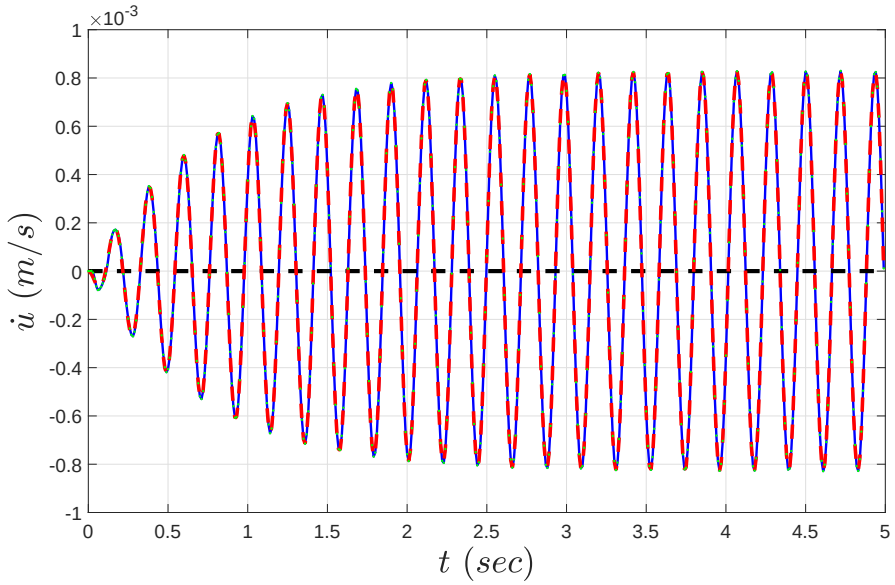


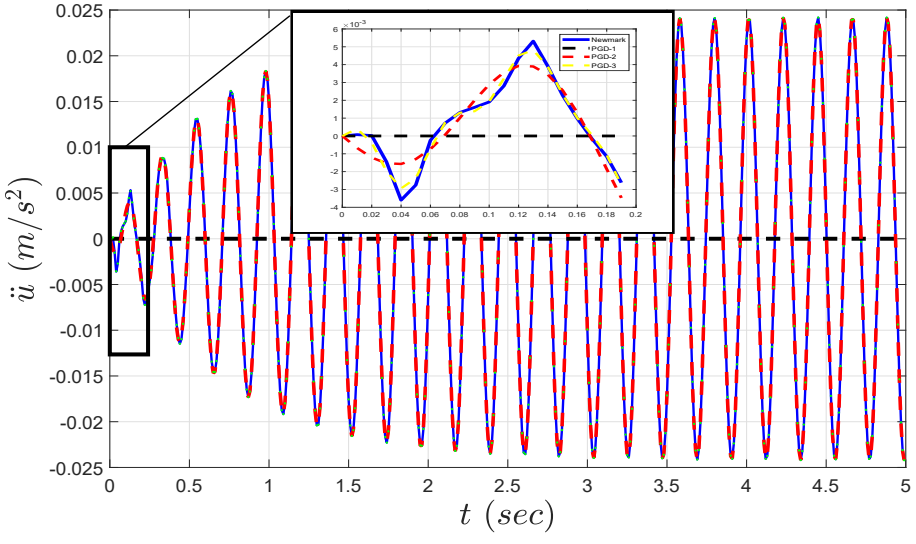
Figure 4.8: 2*D* Frame Resonance-Case 1: Convergence and relative runtimes



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.9: 2*D* Frame Resonance-Case 1: Response

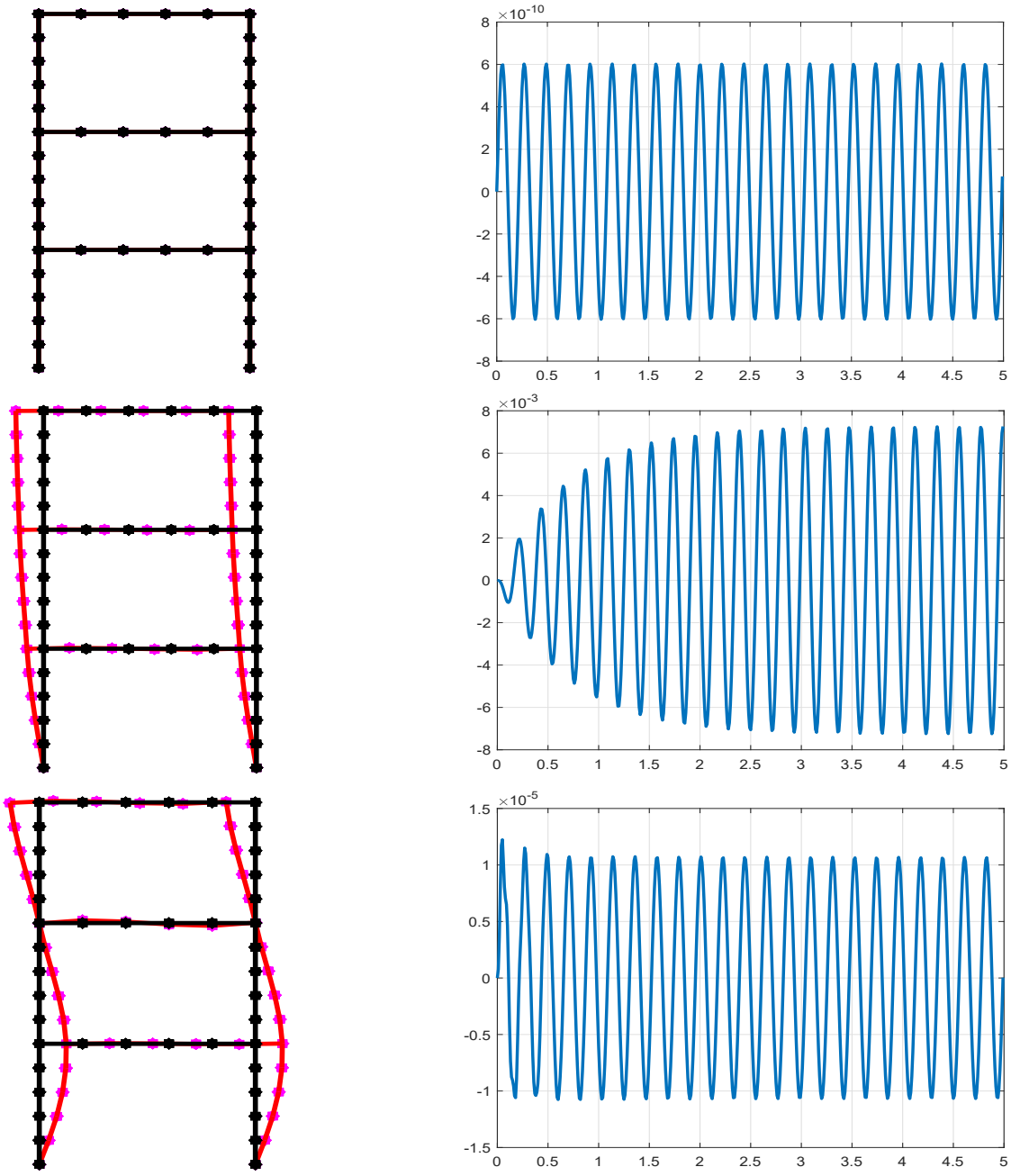


Figure 4.10: 2*D* Frame Resonance-Case 1: PGD Spatial $\left(\frac{\mathbf{u}_s^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ and Temporal $\left(\frac{\mathbf{u}_t^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ Modes

4.2.1.2 Case 2: *Arbitrary-2* Spatial and Temporal Initialization

Next, as a second arbitrary initialization, termed *Arbitrary-2*, the initial spatial and temporal modes are chosen as vectors with all entries equal to 10^{-c}, where $c = 10$ (see Eq. (4.12)).

$$\begin{aligned} \mathbf{u}_s^{(0)} &= \texttt{zeros(128,1)} + 10^{-10} \\ \mathbf{u}_t^{(0)} &= \texttt{zeros(500,1)} + 10^{-10} \end{aligned} \tag{4.12}$$

Table 4.2: 2*D* Frame Resonance-Case 2: Runtime Speed-up

Number of enrichments	$\frac{t_{\text{PGD}}}{t_{\text{Newmark}}}$	Speed-up
1	0.5011	1.9958
2	0.8420	1.1876
3	1.1718	0.8534

Fig. 4.12 depicts the response comparison for the incremental Newmark, the PGD, as well as the modal analysis. Here, the first enrichment yields very good accuracy, surpassing the algorithmic initialization barrier of the previous subsection. It also worth mentioning here, that the normalized first spatial mode, $\left(\frac{\mathbf{u}_s^{(1)}}{|\mathbf{u}_s^{(1)}|_2}\right)$, is equal to the first normalized eigenmode of the system, $\left(\frac{\phi_s^{(1)}}{|\phi_s^{(1)}|_2}\right)$

The convergence of the space-time residual, in terms of matrix norm, against the increasing number of enrichments is shown with the red line in Fig. 4.11. The blue line of Fig. 4.11. depicts the ratio of the time required for the PGD solution, over the time required for the incremental Newmark approach. An acceptable level accuracy is obtained by only one enrichment, accompanied by a speed-up of 1.9958 in terms of analysis runtime.

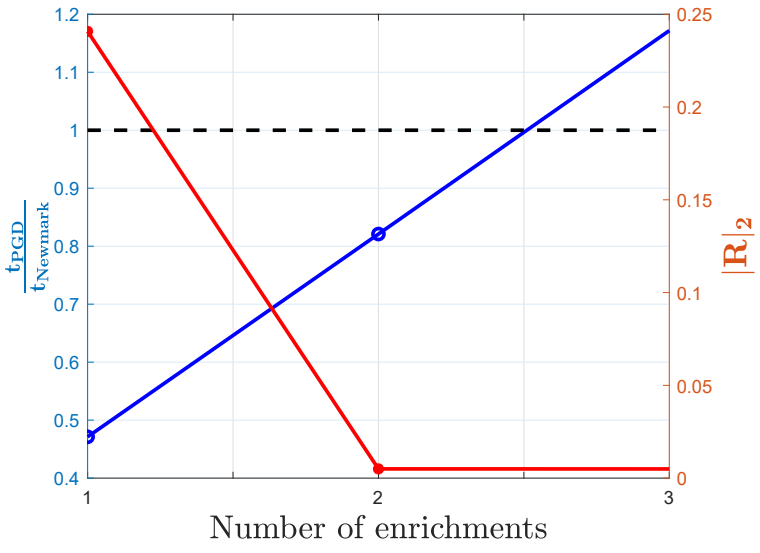
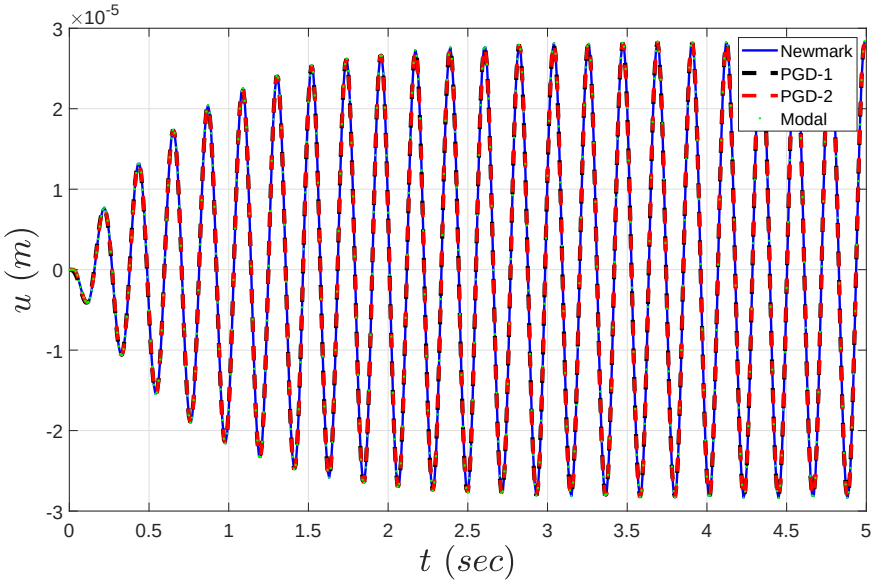
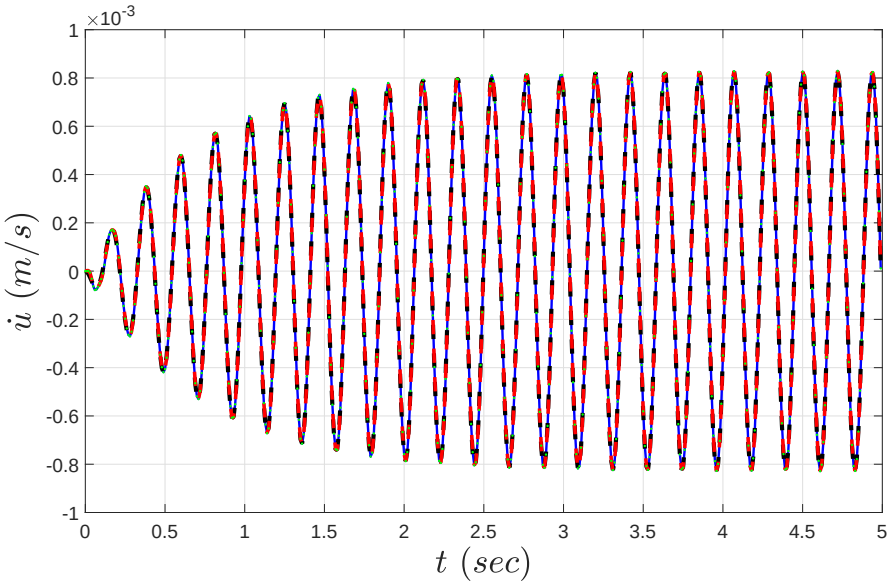


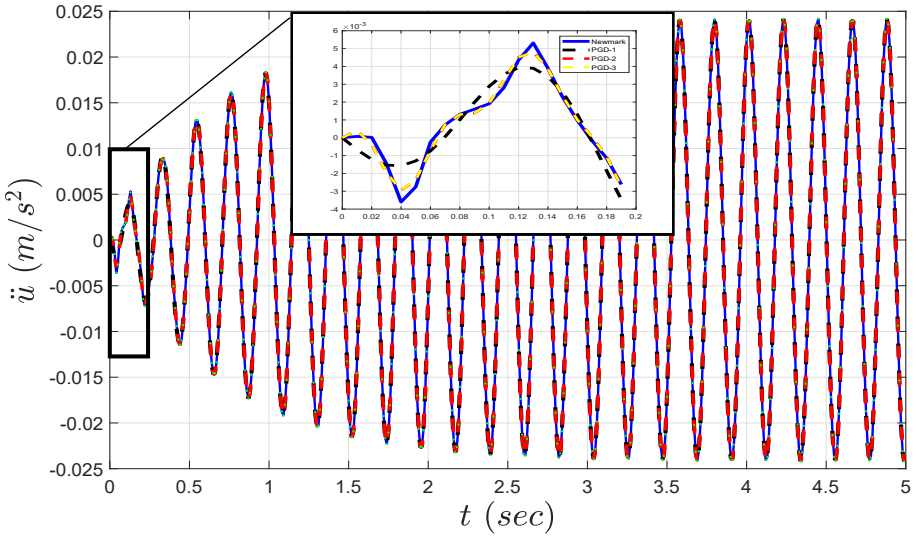
Figure 4.11: 2*D* Frame Resonance-Case 2: Convergence and relative runtimes



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.12: 2*D* Frame Resonance-Case 2: Response

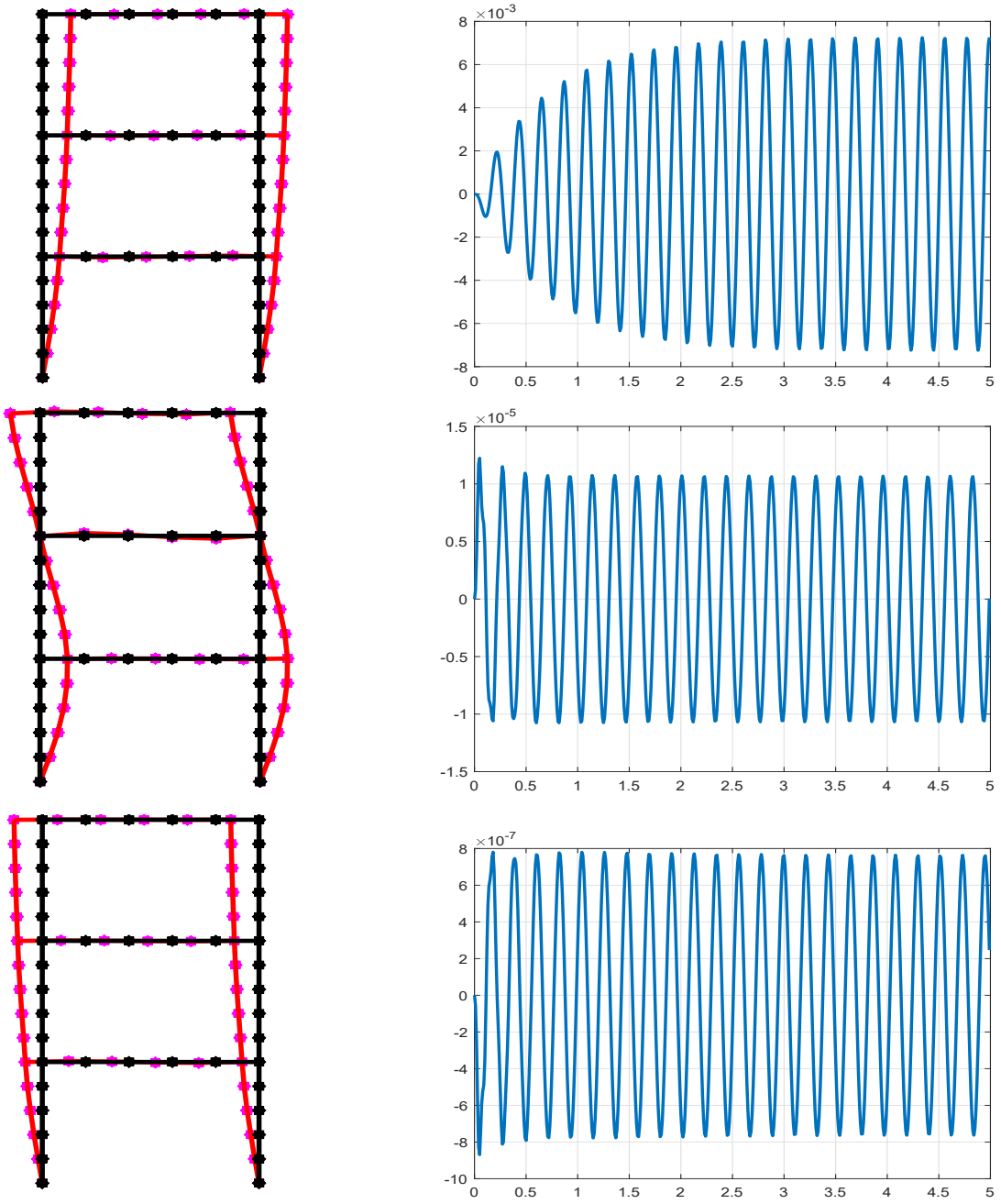


Figure 4.13: 2*D* Frame Resonance-Case 2: PGD Spatial $\left(\frac{\mathbf{u}_s^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ and Temporal $\left(\frac{\mathbf{u}_t^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ Modes

4.2.1.3 Case 3: Static Spatial and *Arbitrary-1* Temporal Initialization

Finally, the same analysis is performed, but the spatial initialization is now selected as the solution displacement vector of the system, subjected to the static load showed in Fig. 4.14.

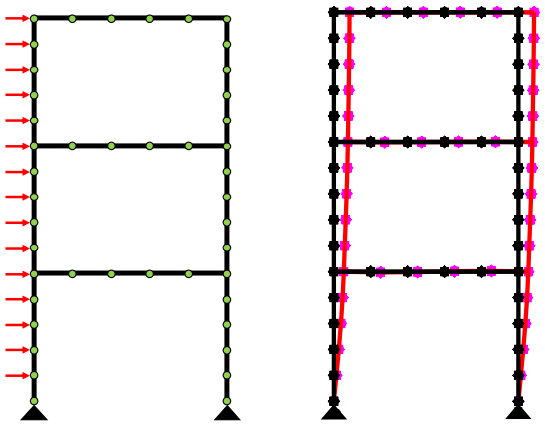


Figure 4.14: 2*D* Frame Resonance-Case 3: Reference static analysis

Convergence and runtime results are shown in Fig. 4.15 and Table 4.3. The results for two PGD enrichments, the conventional Newmark and the modal analysis using 3 eigenmodes as a reduced basis are shown in Fig. 4.16.

It is evident that initiliazing the spatial mode with a vector that roughly resembles the response of the first eigenmode of the structure, accelerates the convergence of the PGD solution, compared to Case 1, and yields accurate results with just one enrichment. The one-enrichment solution however, still can't accurately approximate the transient part of the acceleration response.

Table 4.3: 2*D* Frame Resonance-Case 3: Runtime Speed-up

Number of enrichments	$\frac{t_{\text{PGD}}}{t_{\text{Newmark}}}$	Speed-up
1	0.4749	2.1055
2	0.8246	1.2127
3	1.1801	0.8474

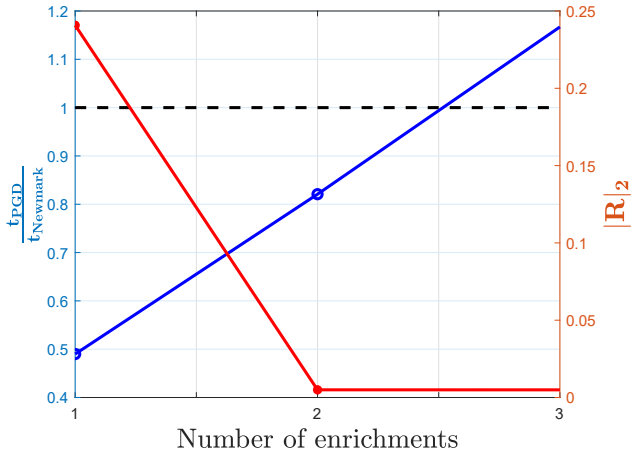
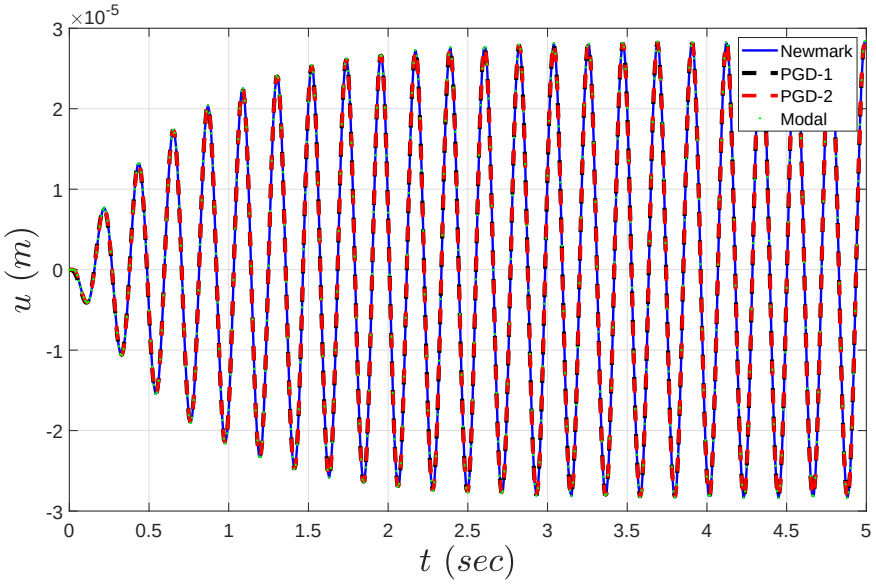
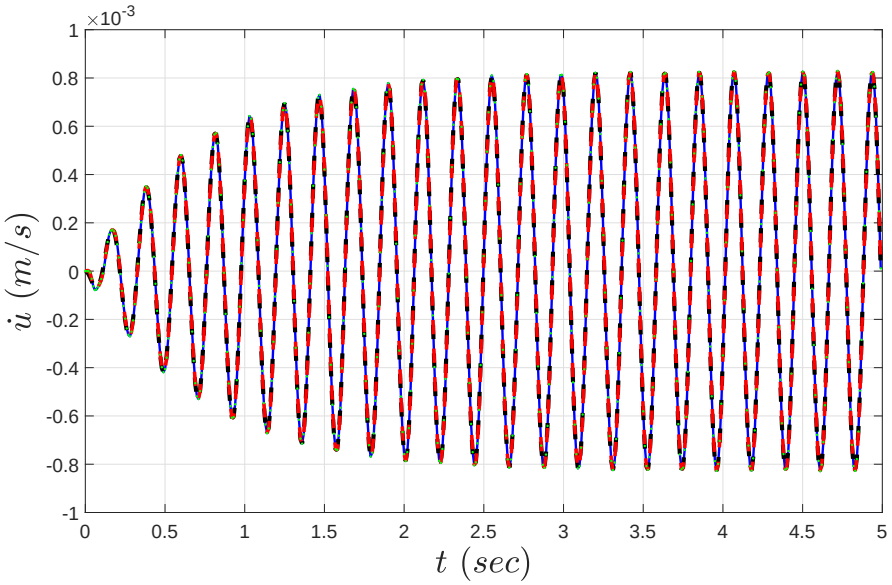


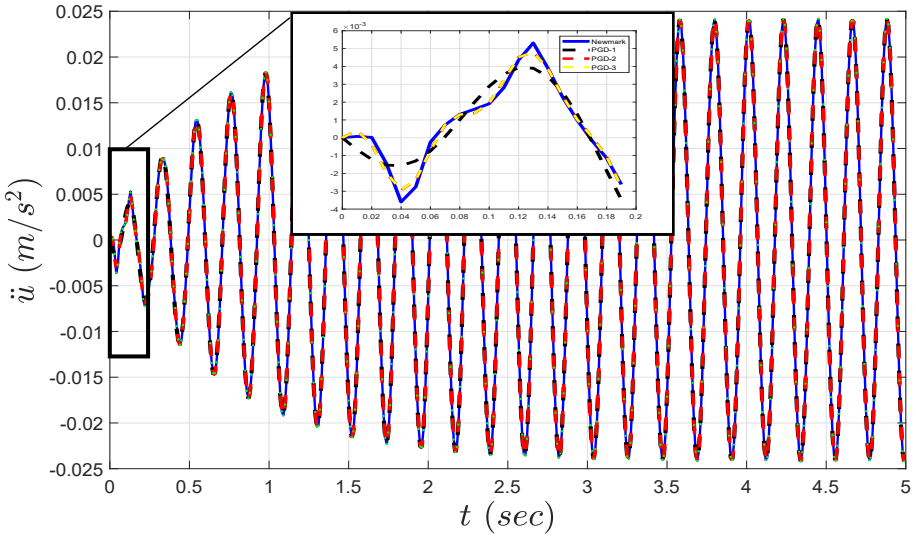
Figure 4.15: 2*D* Frame Resonance-Case 3: Convergence and relative runtimes



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.16: 2*D* Frame Resonance-Case 3: Response

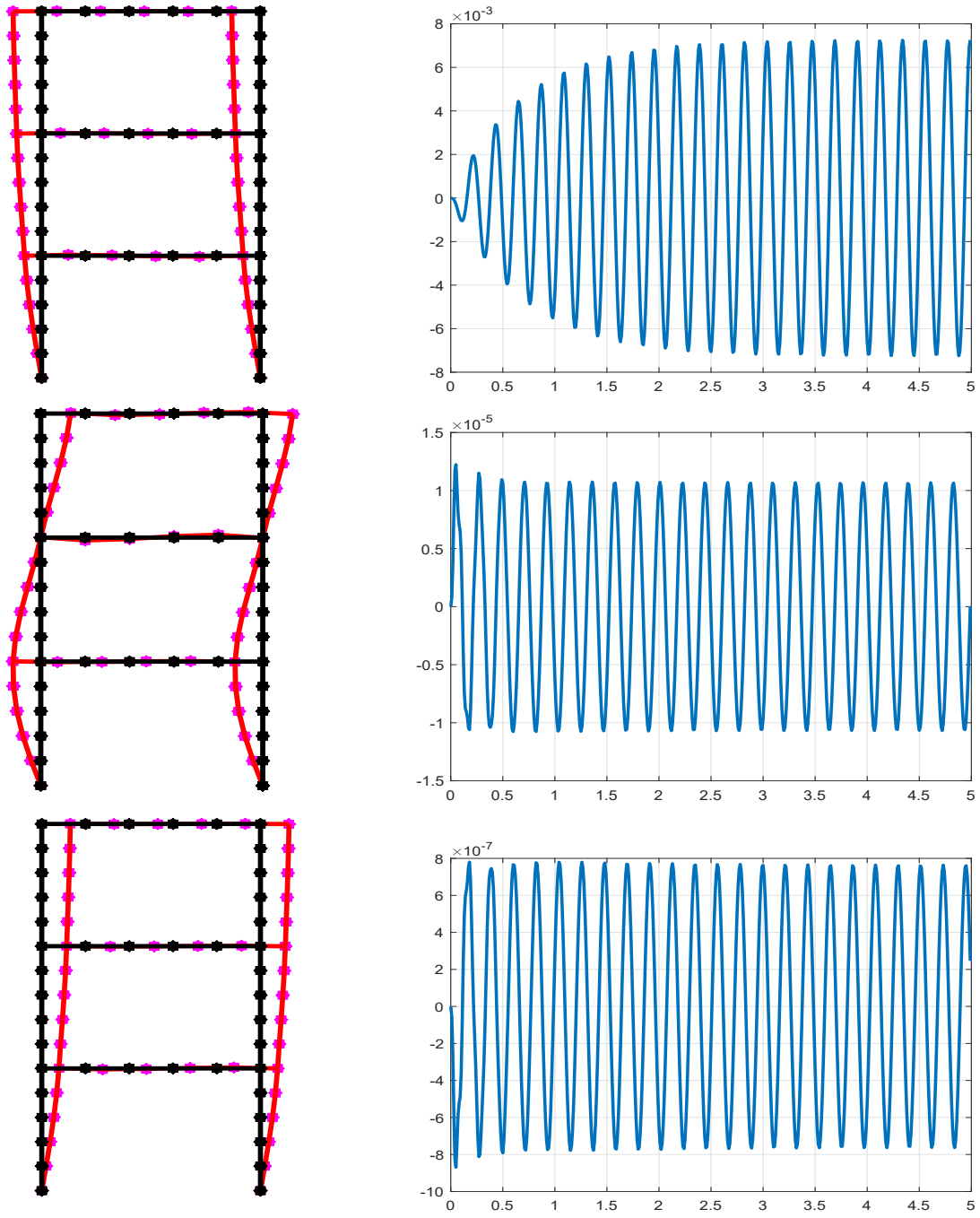
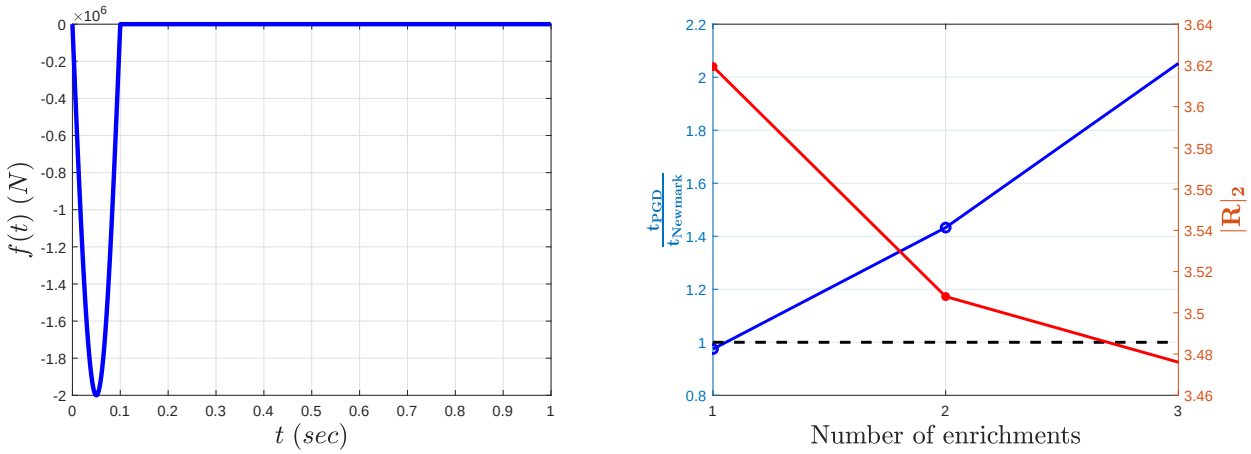


Figure 4.17: 2*D* Frame Resonance-Case 3: PGD Spatial $\left(\frac{\mathbf{u}_s^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ and Temporal $\left(\frac{\mathbf{u}_t^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ Modes

4.2.2 Impulsive Excitation

Next, the 2*D* frame of Fig. 4.5 is subjected to an impulsive external load, depicted in Fig. 4.18a, whose expression is given in the second line of the bottom table of Fig. 4.5. All other system properties remain the same. The time step is chosen equal to: $dt = 5e-04$ [sec]. For the result comparison, the vertical displacement of node #42 is selected as a reference degree of freedom (see Fig. 4.5).



(a) 2*D* Frame Impulse: Impulsive load

(b) 2*D* Frame Impulse-Case 1: Convergence and runtimes

Figure 4.18: 2*D* Frame Impulse: Excitation, Convergence and Runtimes

4.2.2.1 Case 1: *Arbitrary-2* Spatial and Temporal Initialization

First, *Arbitrary-2* initialization for both the spatial and the temporal modes is considered. For this particular excitation case, the initial modes are given similarly to Eq. (4.12), except for the length of the temporal mode, which is equal to 2000.

Response results for 2 PGD enrichments are shown in Fig. 4.20. Even though the excitation induces rather localized response patterns, the PGD approximation is very accurate with only one enrichment. Modal analysis results using 3 eigenvectors as a reduced basis are also included for comparison. Contrary to the PGD, the modal decomposition cannot yield accurate results using a small number of eigenvectors, which is expected since the external impulsive load produces higher mode effects. For an accurate approximation, the minimum number of eigenvectors for the modal reduced basis is found to be equal to 10. Convergence and runtime results are shown in Table 4.4 and Fig. 4.18b. It is noted here, that in terms of runtime, the PGD does not achieve any significant gain in this example. The PGD modes are shown in Fig. 4.19, where it is clear that the first spatial mode represents a localized response pattern.

Table 4.4: 2*D* Frame Impulse-Case 1: Runtime Speed-up

Number of enrichments	$\frac{t_{\text{PGD}}}{t_{\text{Newmark}}}$	Speed-up
1	0.9744	1.0263
2	1.4327	0.6980
3	2.0521	0.4873

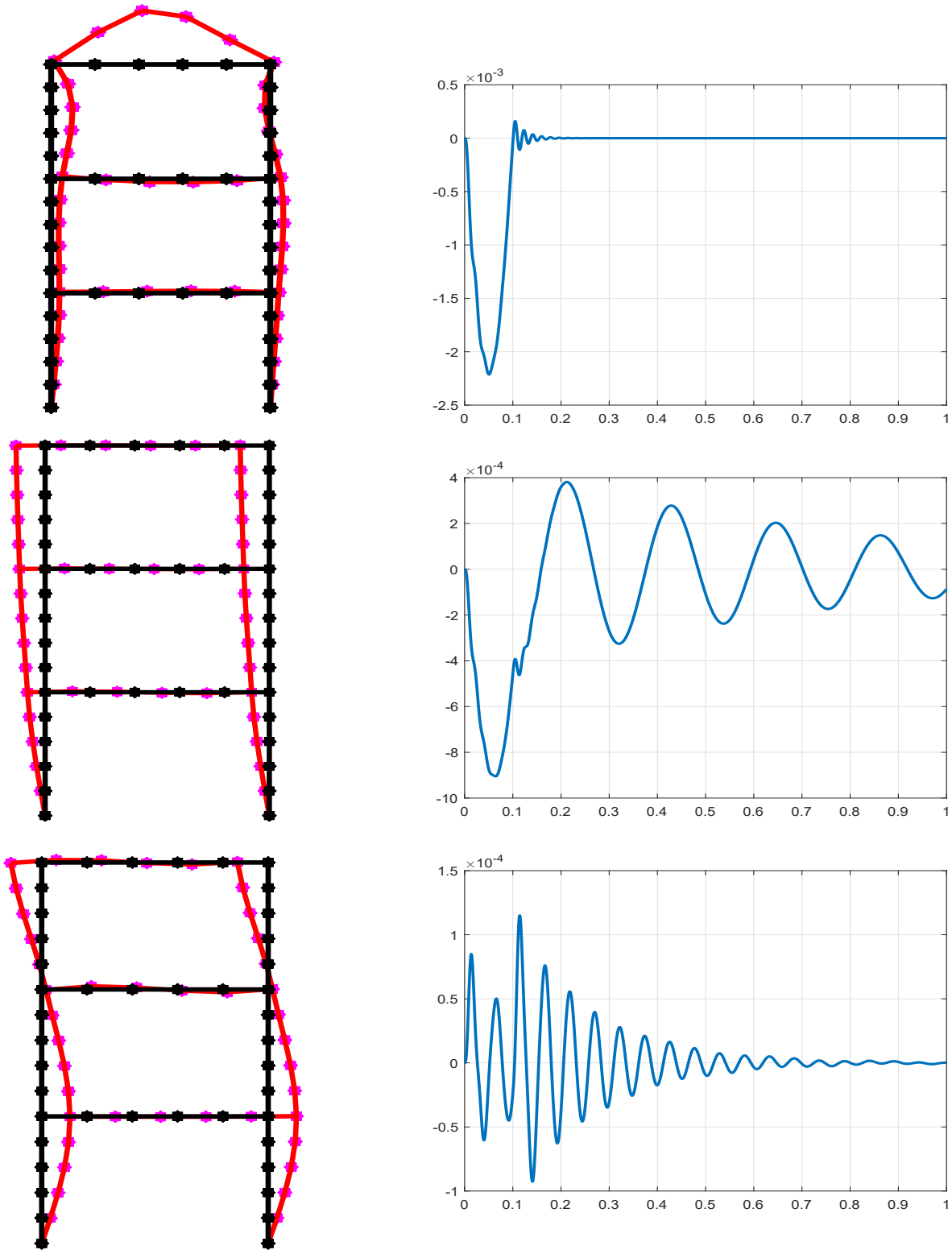
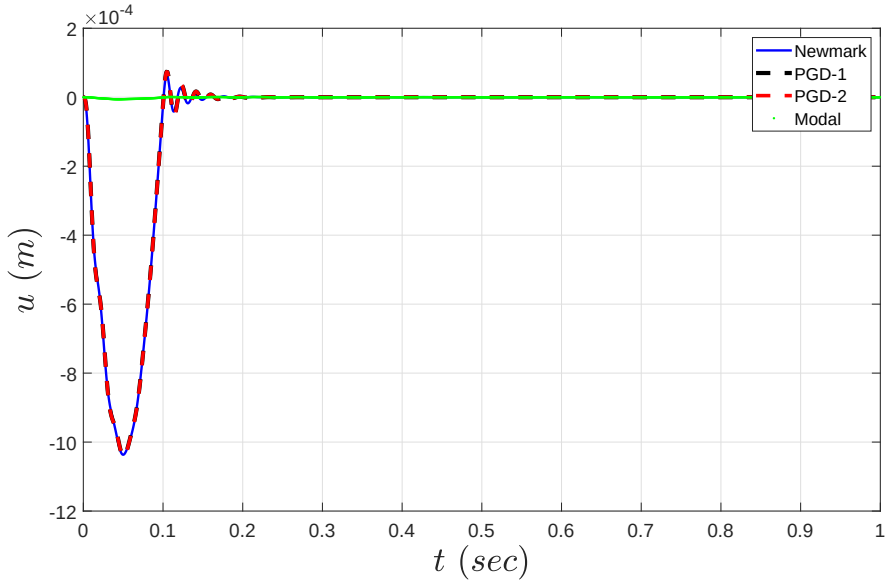
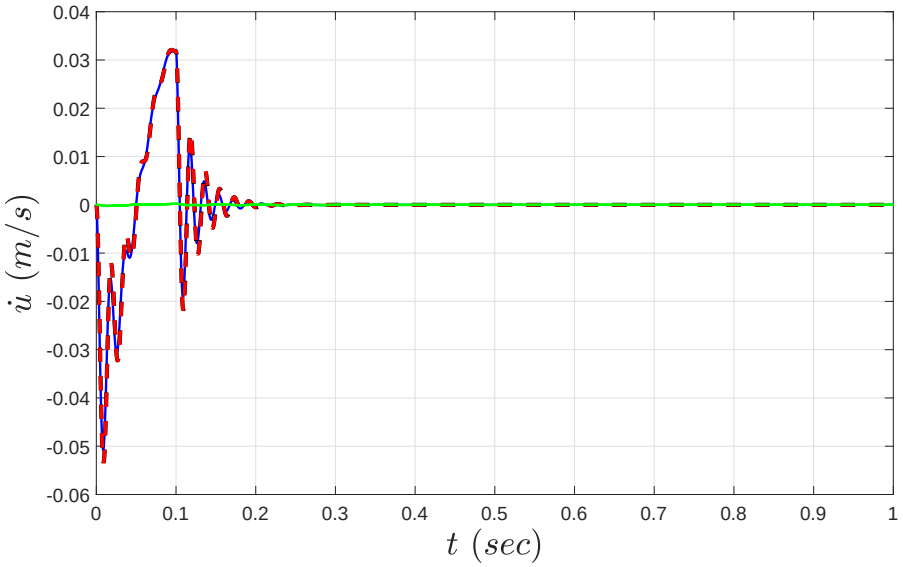


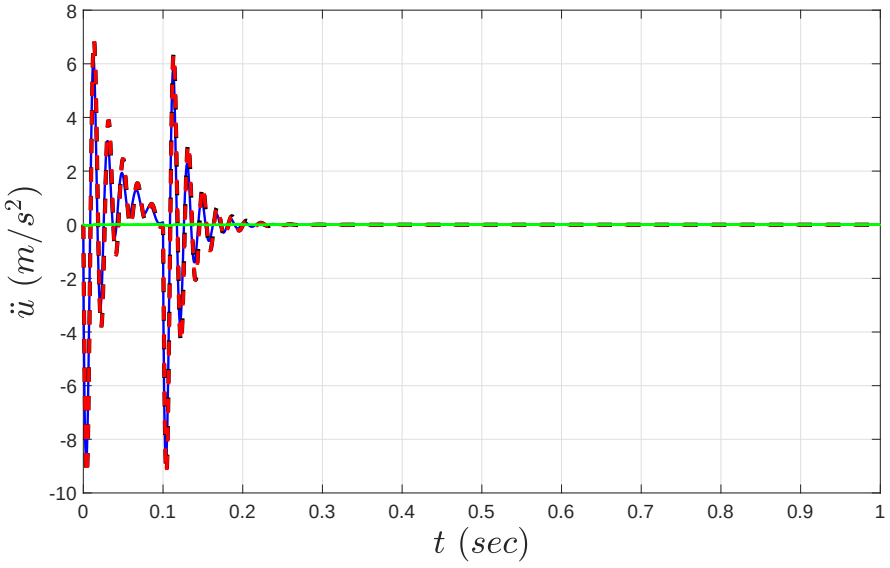
Figure 4.19: 2*D* Frame Impulse-Case 1: PGD Spatial $\left(\frac{\mathbf{u}_s^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ and Temporal $\left(\frac{\mathbf{u}_t^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ Modes



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.20: 2*D* Frame Impulse-Case 1: Response

4.2.2.2 Case 2: Static Spatial and *Arbitrary-1* Temporal Initialization

In this subsection, the 3 storey frame is again excited with the impulsive load of Fig. 4.18a, but the initial spatial mode is chosen as the solution displacement vector, acquired from the static analysis of the structure subjected to a vertical load on the reference node #42 (see Fig. 4.21).

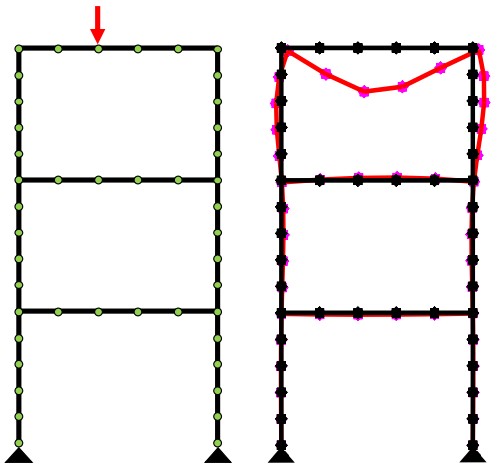


Figure 4.21: 2*D* Frame Impulse-Case 2: Reference static analysis

Same as in the previous benchmark, the PGD yields accurate results with just one enrichment. A not particularly significant runtime speed-up is also accomplished by using one enrichment, as showcased in Table 4.5 and Fig. 4.22. The corresponding spatial and temporal modes are shown in Fig. 4.23, and the response comparison in Fig. 4.24.

Table 4.5: 2*D* Frame Impulse-Case 2: Runtime Speed-up

Number of enrichments	$\frac{t_{\text{PGD}}}{t_{\text{Newmark}}}$	Speed-up
1	0.8950	1.1174
2	1.3716	0.7291
3	1.9690	0.5079

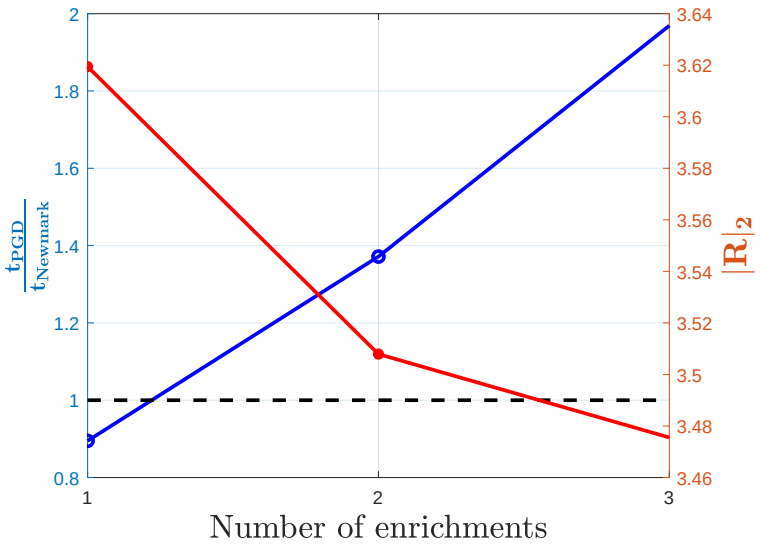


Figure 4.22: 2*D* Frame Impulse-Case 2: Convergence and relative runtimes

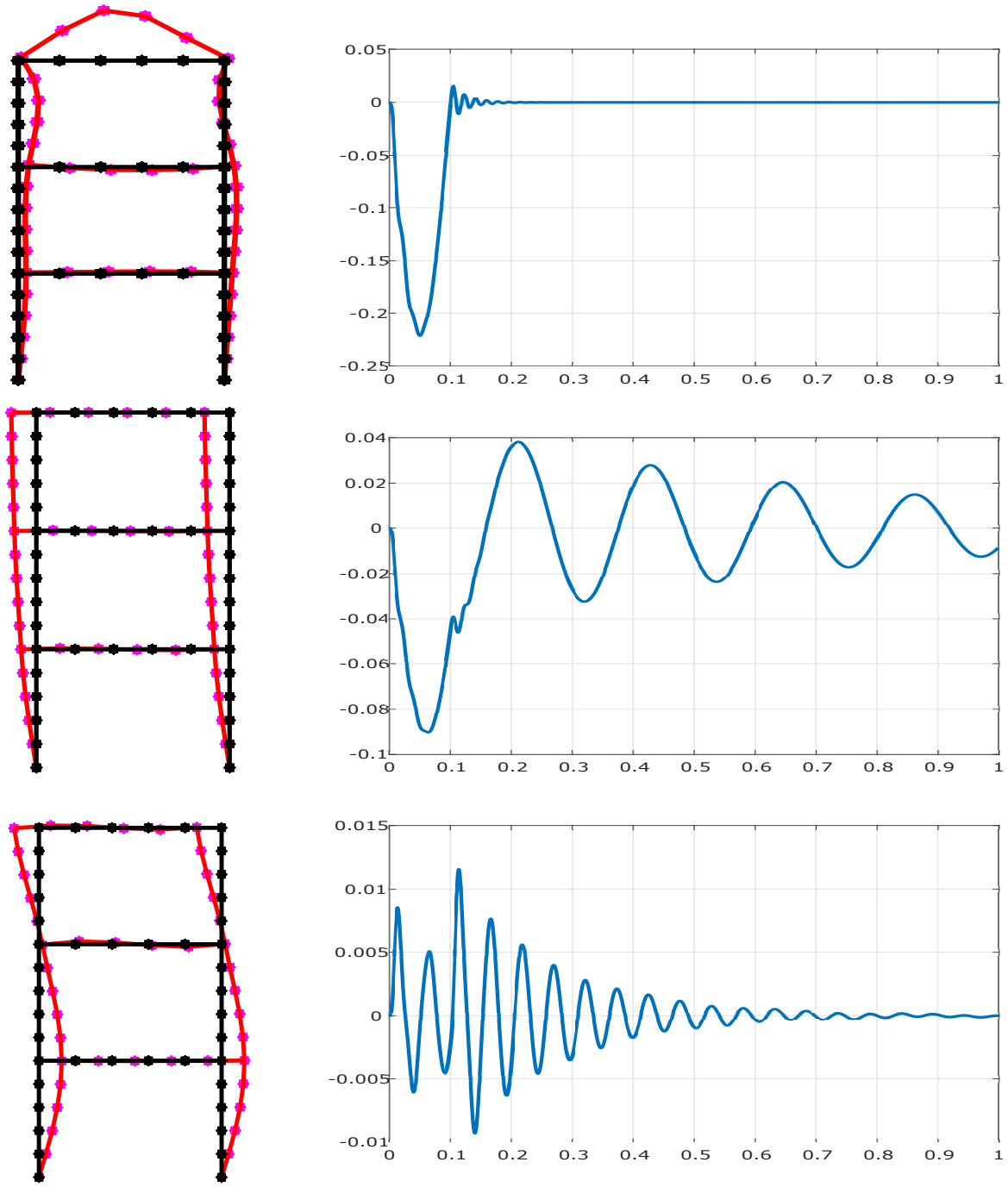
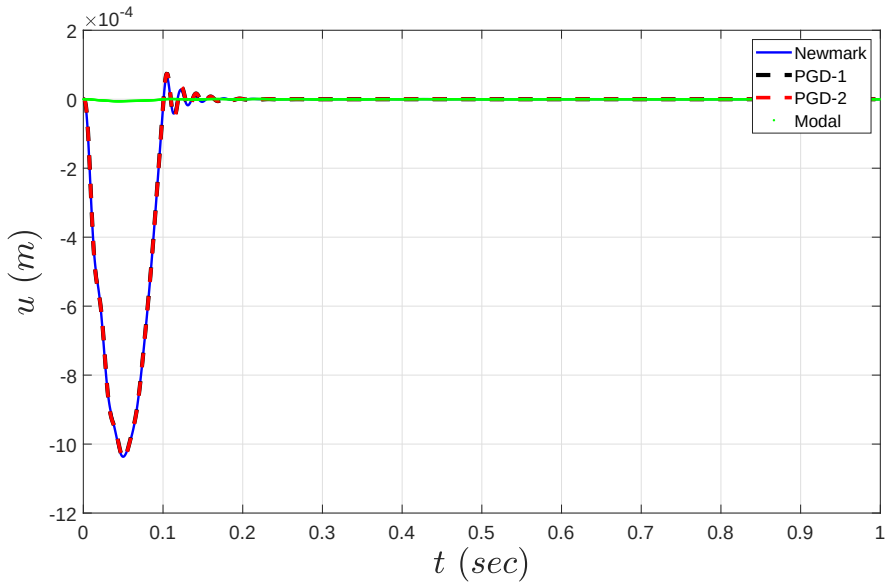
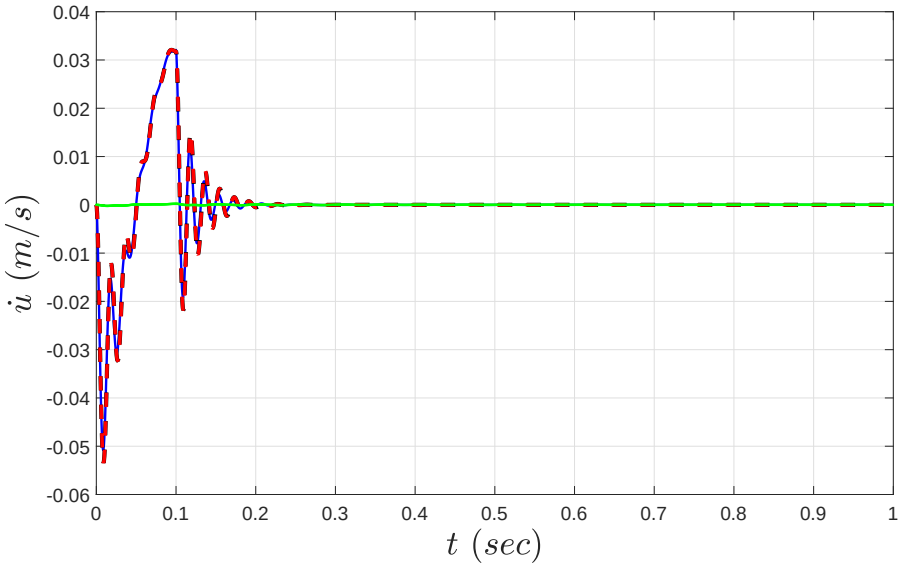


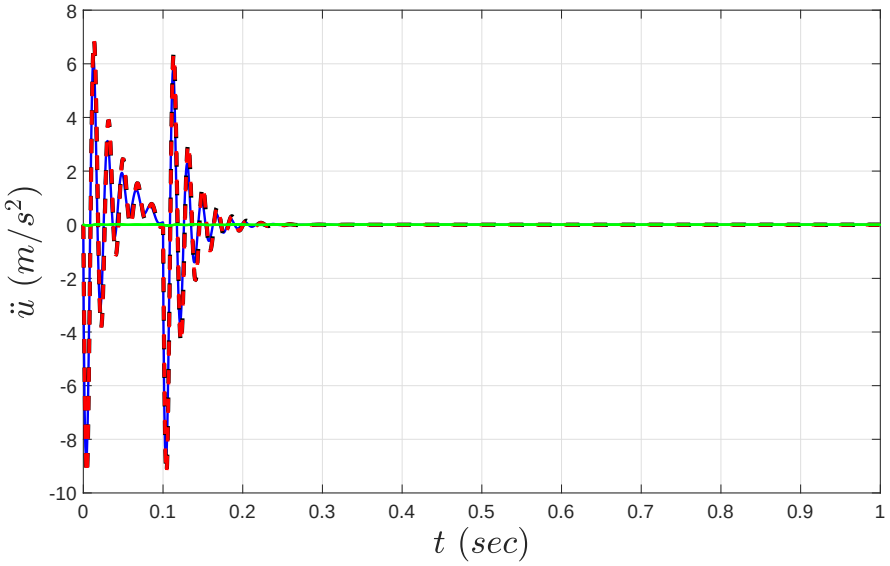
Figure 4.23: 2*D* Frame Impulse-Case 2: PGD Spatial $\left(\frac{\mathbf{u}_s^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ and Temporal $\left(\frac{\mathbf{u}_t^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ Modes



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.24: 2*D* Frame Impulse-Case 2: Response

4.2.3 Transient Excitation

Finally, the $2D$ frame is subjected to the 1995 Kobe earthquake (Fig. 4.25). *Arbitrary-2* initialization for both the spatial and the temporal modes is chosen. The time step is chosen equal to: $dt = 0.01$ [sec] and the record duration is 20 [sec]. Thus, the initial mode vectors are the given by Eq. (4.12).

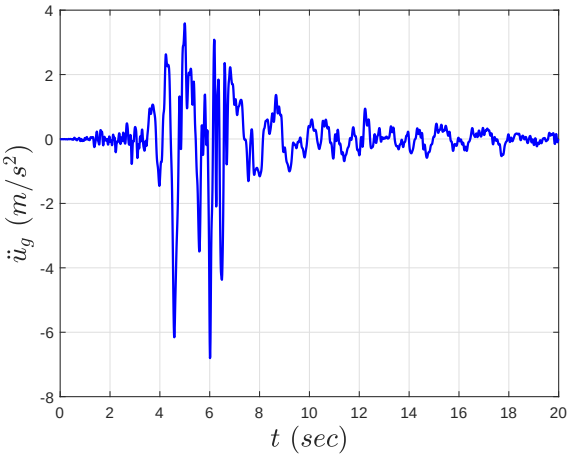


Figure 4.25: $2D$ Frame Transient: Kobe ground acceleration [23]

For this excitation, the PGD once again yields accurate response with one enrichment (Fig. 4.28). A speed-up is achieved by using up to 2 enrichments(Fig 4.26, Table 4.6). The corresponding spatial modes, depicted in Fig. 4.27, resemble the low frequency eigenmodes of the structure.

Table 4.6: $2D$ Frame Transient: Runtime Speed-up

Number of enrichments	$\frac{t_{\text{PGD}}}{t_{\text{Newmark}}}$	Speed-up
1	0.5449	1.8350
2	0.9272	1.0785
3	1.1465	0.8722

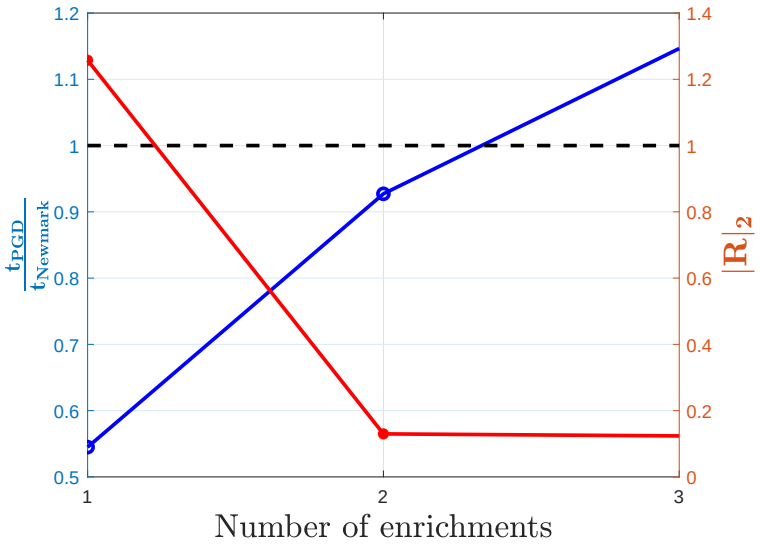


Figure 4.26: $2D$ Frame Transient: Convergence and relative runtimes

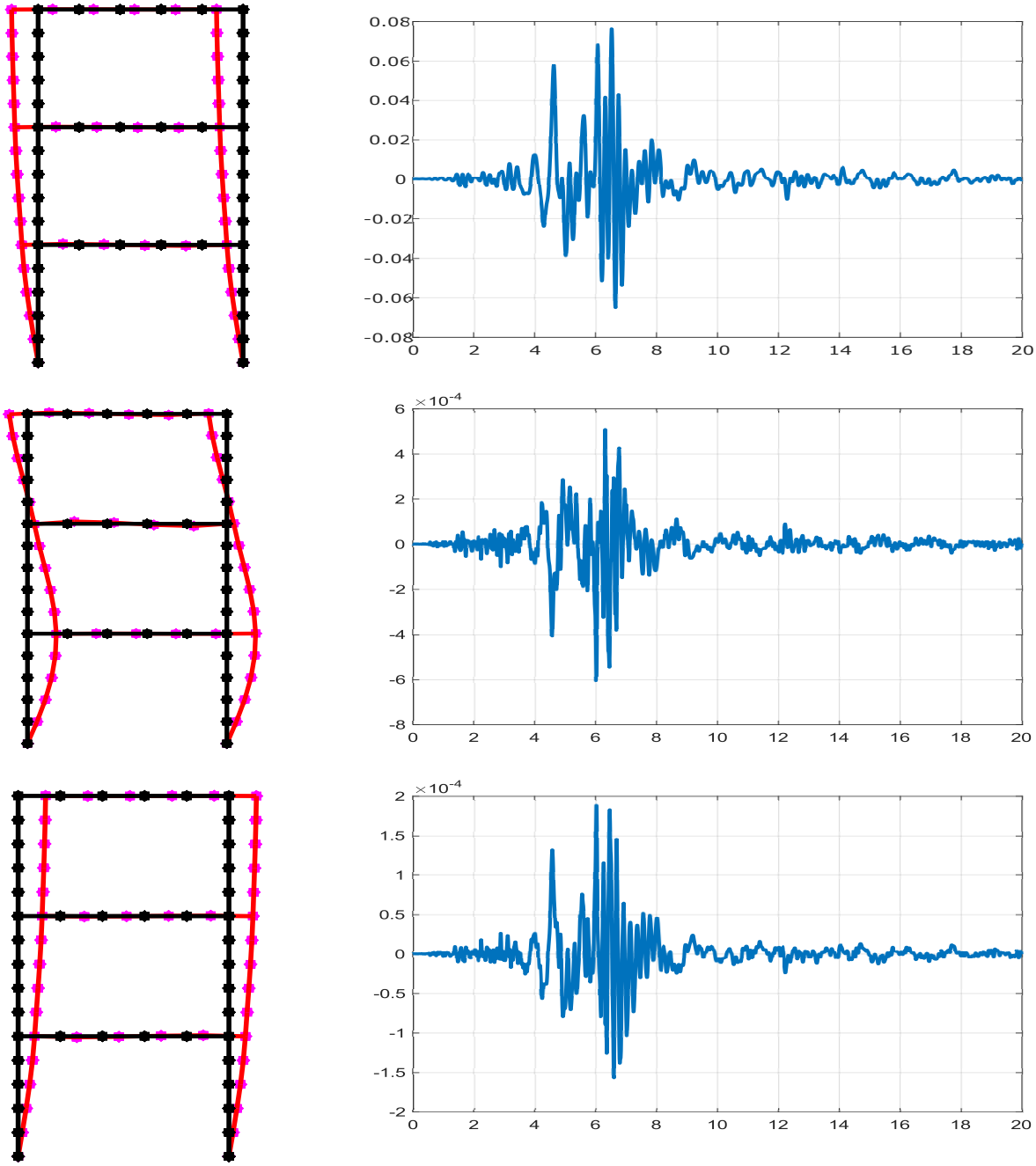
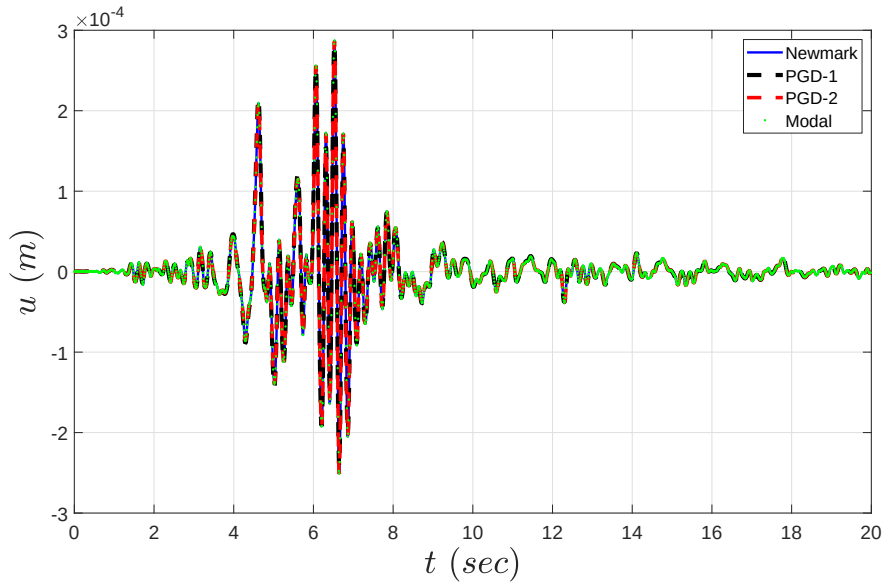
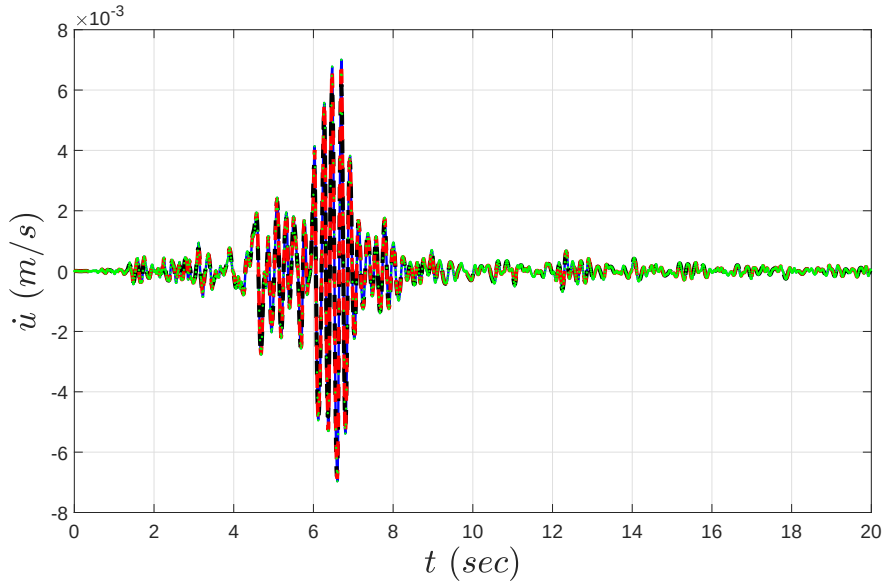


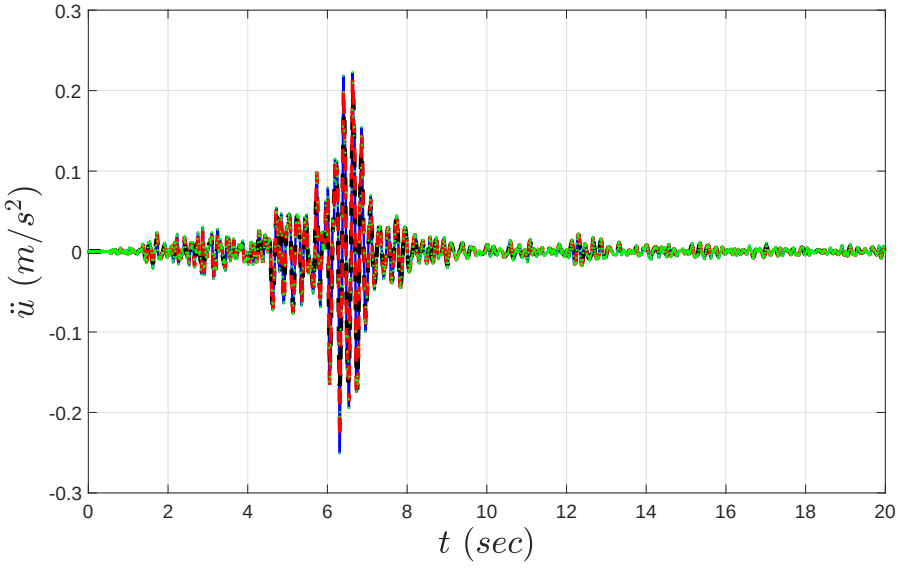
Figure 4.27: 2*D* Frame Transient: PGD Spatial $\left(\frac{\mathbf{u}_s^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ and Temporal $\left(\frac{\mathbf{u}_t^{(m)}}{|\mathbf{u}_s^{(m)}|_2}\right)$ Modes



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.28: 2*D* Frame Transient: Response

For this example, as an error metric, as well as a convergence indicator, the Frobenius norm of the global displacement error matrix, is investigated. To that end, the algorithm is set to run for 20 enrichments. To define the global displacement error matrix (which can also be seen as the trace of the covariance matrix of the space-time displacement matrix), we first define the two point correlation matrix of the difference between the displacement matrix via the Newmark method and the PGD space-time displacement matrix.

$$\mathbf{C_U} = (\mathbf{U}_{\text{Newmark}} - \mathbf{U}_{\text{PGD}})(\mathbf{U}_{\text{Newmark}} - \mathbf{U}_{\text{PGD}})^T \tag{4.13}$$

The temporal average of the square root of $\mathbf{C_U}$ yields the global displacement error matrix:

$$\epsilon_{\mathbf{U}} = \frac{1}{n_t}\mathbf{C_U}^{\frac{1}{2}} \tag{4.14}$$

Fig. 4.29 showcases the convergence behavior of the global error for two values of the fixed point algorithm tolerance.

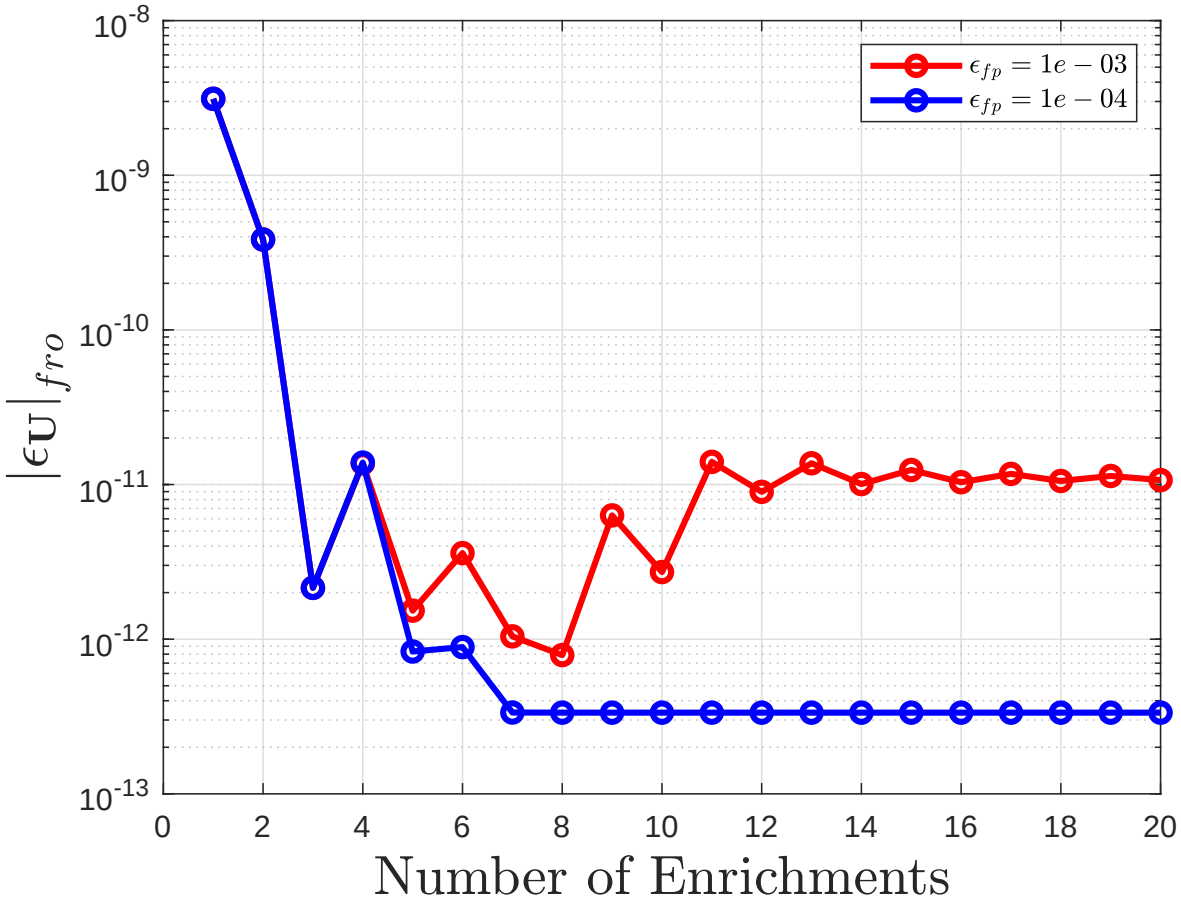
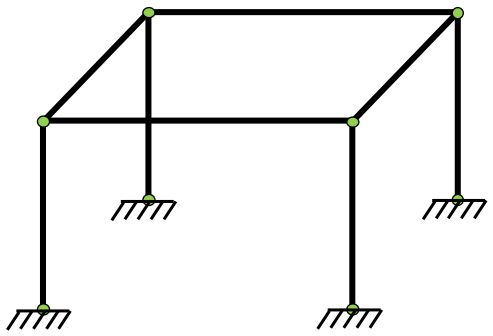


Figure 4.29: Global error convergence

4.3 3D Frame

As a final benchmark, the $3D$ frame structure of Fig. 4.30 is analyzed, using the Kobe earthquake excitation of Fig. 4.25.



Model Properties		
$\rho = 2500\ [\mathrm{kg/m^3}]$		
$E = 0.1\,[GPa]$		
Columns:	$I_x = 10^{-5}$	$[m^4]$
	$I_y = 10^{-3}$	$[m^4]$
	$I_z = 5 \cdot 10^{-4}$	$[m^4]$
	$A = 0.5$	$[m^2]$
Beams:	$I_x = 2.1 \cdot 10^{-5}$	$[m^4]$
	$I_y = 2.1 \cdot 10^{-3}$	$[m^4]$
	$I_z = 2.5 \cdot 10^{-4}$	$[m^4]$
	$A = 0.25$	$[m^2]$

Figure 4.30: 3*D* Frame: Model

Fig. 4.31 showcases the space-time residual convergence and the response comparison for the floor displacement along the $x-$axis is given in Fig. 4.32. The PGD with *Arbitrary-2* initialization again yields results with just one enrichment. The analysis is carried out using both the lumped mass and the consistent mass approach. The relative results (between Newmark and PGD) are similar but it is worth noting that the PGD runtime using the lumped mass approach is 51% of the PGD runtime considering a consistent mass matrix, whereas the lumped mass Newmark solution runtime is 81% of the consistent mass Newmark runtime. This fact states that the efficiency of the PGD can be influenced by the ”structure” characteristics of the system matrices (e.g. sparsity, bandwidth, etc).

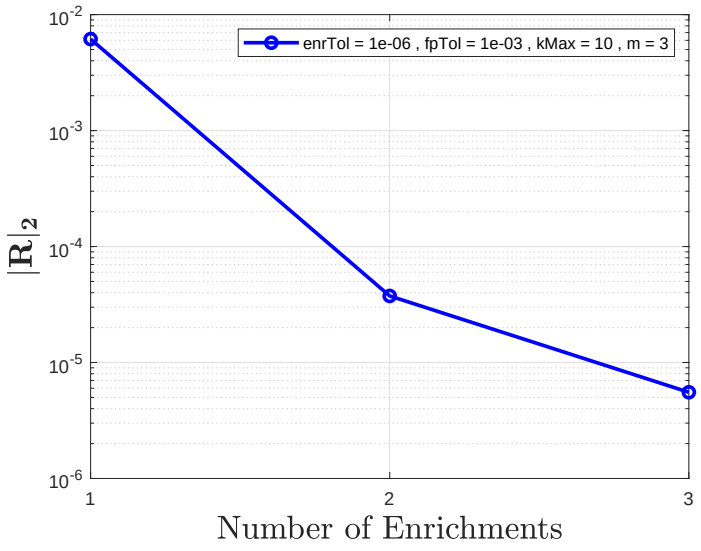
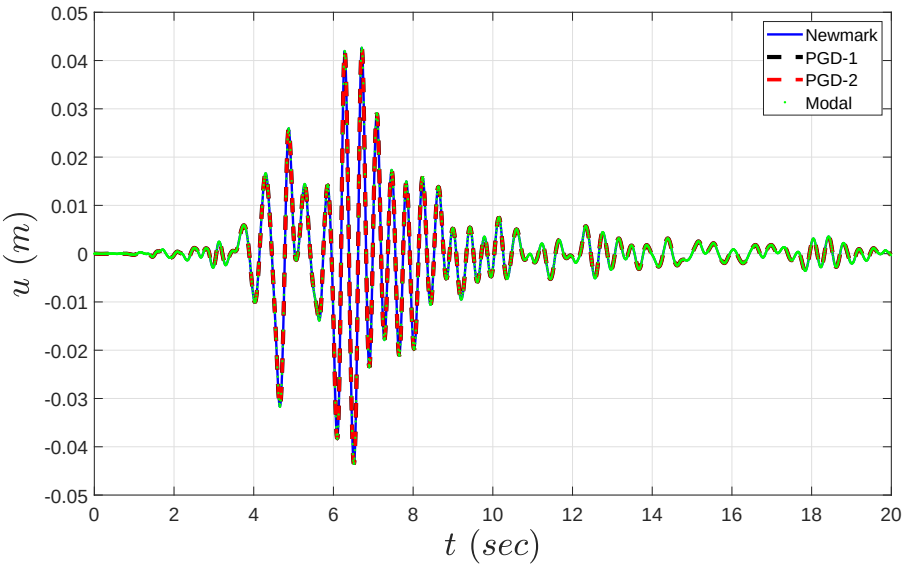
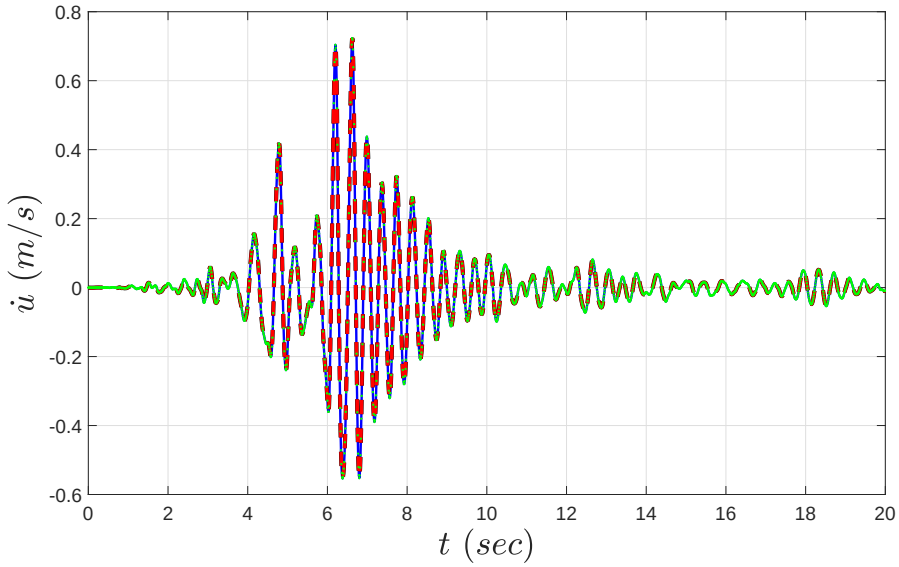


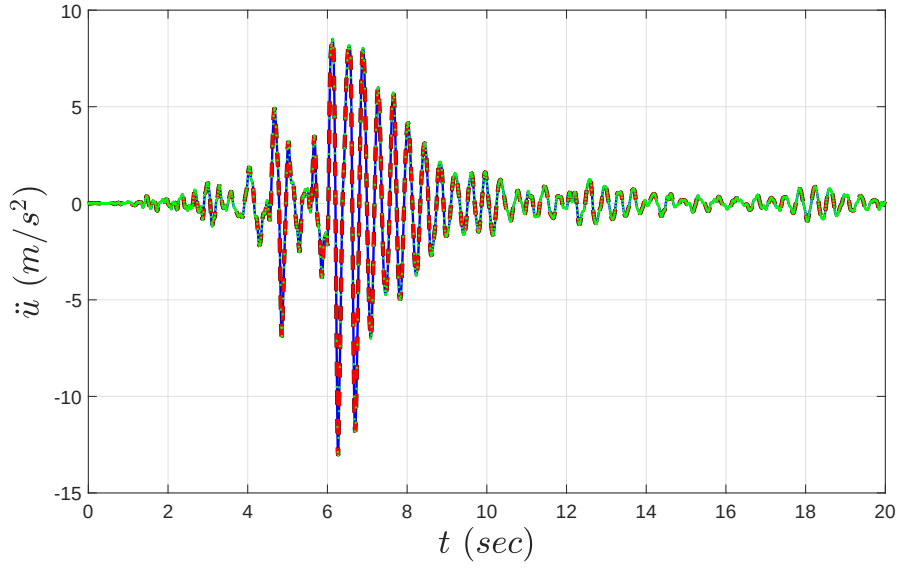
Figure 4.31: 3*D* Frame Transient: Convergence



(a) Displacement



(b) Velocity



(c) Acceleration

Figure 4.32: 3*D* Frame Transient: Response

Chapter 5

CONCLUSIONS AND FUTURE WORK

5.1 Summary - Conclusions

In this thesis, a Proper Generalized Decomposition(PGD)-based space-time formulation of the Newmark time integration scheme for linear structural dynamics is presented. The PGD approach is used to cast the incremental solution scheme into its non-incremental space-time equivalent. The latter produces a system of two coupled space and time equations, which is solved using a greedy algorithm, namely a fixed point algorithm with Picard iterations. The displacement field is sought in a *separated representation* form, i.e. a product of spatial and temporal modes. Thus, the solution of the initial boundary value problem in the entire space-time domain is iteratively constructed via enrichments until convergence is achieved.

The presented formulation is demonstrated and verified via several numerical benchmarks. During convergence, the occurring spatial modes (spatial response patterns of the analyzed structure) and temporal modes (time series of specific frequencies) are investigated. First, two single-degree-of-freedom (SDOF) systems, whose analytical solution is available, are considered. The PGD solver yields very accurate results when compared to the conventional Newmark scheme as well as the analytical solution.

As the main benchmark model, a 3-storey $2D$ frame structure is chosen and analyzed using different excitations. Harmonically exciting the structure with a ground acceleration whose frequency matches the first eigenfrequency of the structure produces a resonant response. The PGD yields accurate results with only one enrichment. In this particular case, the computed spatial PGD mode normalized with its $L2$ norm is equal the structures $L2$ normalized first eigenmode. Frequency analysis of the corresponding first temporal PGD mode yields a dominant frequency equal to the first eigenfrequency of the structure.

Next, an impulsive (half-sine) point load is applied. The PGD is able to capture the localized deformation pattern within the first enrichment. The spatial PGD modes contain both local and global response behaviors induced by the impulse. This fact states an advantage of the PGD over the traditional approach of modal truncation, as its computed modes adapt to the structural response patterns. Furthermore, data-driven approaches like the Proper Orthogonal Decomposition (POD) are also outperformed, since the localized deformation patterns need to be captured in the data used to compute the reduced order basis in order to be predicted by the reduced order model.

Finally, the structure is excited with a seismic ground acceleration. Once again, one enrichment is enough to adequately approximate the system response. The first few spatial PGD modes reveal a high correlation with the structure first few, low frequency, linear modes of vibration.

The benchmarks conclude with the analysis of the 1-story $3D$ frame. The system mass matrix is assembled both using the lumped mass approach and the consistent mass approach. Comparison of the PGD runtimes for this example showcases an important acceleration of the PGD algorithm when lumped masses are considered, which is not the case for the conventional Newmark method.

Several remarks regarding the algorithmic implementation of the formulation are summarized below.

- Accuracy is achieved with a small number of enrichments for different excitation scenarios.
- The PGD approach can accelerate the analysis runtimes.
- Memory requirements (for processing) and storage requirements (for storing) the solution are reduced, thanks to the separated representation of the solution field.
- The PGD spatial modes can be correlated with the structure eigenmodes.
- Good convergence behavior of the solution procedure is observed.
- The algorithmic parameters, mainly the fixed point convergence tolerance has an important influence on the algorithm's accuracy and convergence behavior.

Concluding, the presented approach follows a solution procedure that differs from the conventional step-by-step methods. Its computational efficiency mainly depends on the number of enrichments performed, but also on the algorithmic parameters and the internal numerical strategies involved. In some cases, a speed-up factor of two was achieved. However, the PGD's possibilities, as well as its limits, within space-time structural dynamics problems, are yet to be discovered.

5.2 Future Research

The following are research directions that could greatly extend/improve the current work:

1. Extension of the space-time formulation to problems involving material and/or geometrical nonlinearities (e.g. history-dependent material behavior, large deformations, nonlinear boundary conditions).
2. Application of the space-time formulation to a wide variety of structural dynamics problems and investigation of its applicability/performance.
3. Integration of numerical analysis procedures, other than the fixed point algorithm, for solving the coupled space-time equations (e.g. family of Newton methods).
4. Following in the footsteps of [19], establishment of space-time formulations for other implicit/explicit time integration schemes (e.g. CDM, Bathe, HHT-α) and investigation of their efficiency.
5. Integration of new strategies within the current formulation, that could reduce the computational cost while preserving good convergence behavior. One goal could be to provide new techniques for assembling/representing the space-time matrices of the algorithm, whose dimension can become very high when dealing with fine spatial and temporal meshes. Furthermore, alternatives to the serial nature of the successive enrichment procedure should be investigated, making its deployment on parallel computing platforms possible.

Appendix A

Tensor Algebra Definitions

In this section, some basic operations from tensor and multilinear algebra are presented ([25]). The ultimate goal is applying a tensor formulation, onto the equations governing structural dynamics, to construct (or identify) space-time structural system operators. The notation incorporated in this work for the tensors is the ”abstract index notation” as well as the ”underline convention”. Thus, $\underline{\mathbf{A}} \equiv \mathbf{A}_i$ is a 1st order [1] tensor, $\underline{\underline{\mathbf{A}}} \equiv \mathbf{A}_{ij}$ is a 2nd order tensor etc..

First, the tensor product $\otimes$ is defined as:

$$\underline{\underline{\mathbf{A}}} = \underline{\mathbf{Q}} \otimes \underline{\mathbf{W}} \Leftrightarrow \mathbf{A}_{ij} = \mathbf{Q}_i \otimes \mathbf{W}_j \tag{A.1}$$

or using matrix and vector algebra notation (similarly to the definition of the Kronecker product)

$$\underline{\mathbf{Q}} \otimes \underline{\mathbf{W}} = \begin{bmatrix} Q_1 \underline{\mathbf{W}} \\ \vdots \\ Q_{n_Q} \underline{\mathbf{W}} \end{bmatrix} \tag{A.2}$$

Next, the single tensor contraction ”$\cdot$” between a 2nd order and a 1st order tensor is defined as:

$$\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{Q}} = \underline{\mathbf{W}} \tag{A.3}$$

and the double contraction ”:” between a 4th order and a 2nd order tensor as:

$$\underline{\underline{\underline{\underline{\mathbf{A}}}}} : \underline{\underline{\mathbf{Q}}} = \underline{\underline{\mathbf{W}}} \tag{A.4}$$

The double contraction between two 2nd order tensors can be viewed as the inner product between them and using Einstein’s summation convention, we write:

$$\underline{\underline{\mathbf{Q}}} : \underline{\underline{\mathbf{W}}} = Q_{ij} W_{ij} \tag{A.5}$$

From Eq. (A.1) and Eq. (A.4), we can consider a 4th order tensor as a linear (or better yet bilinear) mapping between two 2nd order tensors:

$$\left(\underline{\underline{\mathbf{Q}}} \otimes \underline{\underline{\mathbf{W}}} \right) : \underline{\underline{\mathbf{Z}}} = \underline{\underline{\mathbf{X}}} \Leftrightarrow Q_{ij} W_{kl} Z_{jl} = X_{ik} \tag{A.6}$$

[1] Terms *order* and *rank* are used without distinction.

Additionally, the definitions for the dyadic, the outer (for column vectors) and the tensor product of two vectors are given respectively as:

$$\mathbf{A} = \mathbf{ab} \tag{A.7}$$

$$\mathbf{A} = \mathbf{ab}^T \tag{A.8}$$

$$\mathbf{A} = \mathbf{a} \otimes \mathbf{b} \tag{A.9}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^n$. It is noted that definitions Eq. (A.7), Eq. (A.8) and Eq. (A.9) are equivalent and the only the notation differs.

Next, a proof is provided, for the following property of the product between a 3rd order tensor and a 2nd order tensor.

$$\underline{\underline{\underline{\mathbf{KU}}}} = \left(\underline{\underline{\underline{\mathbf{K}}}}_s \otimes \underline{\mathbf{k}}_t\right)\left(\underline{\mathbf{u}}_s \otimes \underline{\mathbf{u}}_t\right) = \left(\underline{\underline{\underline{\mathbf{K}}}}_s \underline{\mathbf{u}}_s\right) \otimes \left(\underline{\mathbf{k}}_t \circ \underline{\mathbf{u}}_t\right) \tag{A.10}$$

where $\circ$ denotes the component-wise product. The aim is to employ this property for the product between a decomposed 3rd order tensor that represents a space-time system operator and the corresponding decomposed 2nd order tensor space-time response quantity.

Using the abstract index notation, the components of a 3rd order system operator tensor (e.g. stiffness) are written as:

$$K_{ijk} = K_{ij}(t_k) = (K_s)_{ij}(k_t)_k \tag{A.11}$$

and the components of a 2nd order system response tensor (e.g. displacement) are written as:

$$U_{jk} = u_j(t_k) = (u_s)_j(u_t)_k \tag{A.12}$$

The product between the stiffness tensor and the displacement tensor yield the components:

$$\begin{aligned}
&\sum_{j=1}^{n_s} K_{ij}(t_k)u_j(t_k) = \sum_{j=1}^{n_s} (K_s)_{ij}(k_t)_k(u_s)_j(u_t)_k \Rightarrow \\
&\sum_{j=1}^{n_s} K_{ij}(t_k)u_j(t_k) = \sum_{j=1}^{n_s} (K_s)_{ij}(u_s)_j(k_t)_k(u_t)_k \Rightarrow \\
&\sum_{j=1}^{n_s} K_{ij}(t_k)u_j(t_k) = (K_s u_s)_i(k_t u_t)_k
\end{aligned} \tag{A.13}$$

The left hand sides of Eq. (A.13) are the components of the product $\underline{\underline{\underline{\mathbf{KU}}}}$ whereas the right hand side of the third line of Eq. (A.13) represents the components of the term: $\left(\underline{\underline{\underline{\mathbf{K}}}}_s \underline{\mathbf{u}}_s\right) \otimes \left(\underline{\mathbf{k}}_t \circ \underline{\mathbf{u}}_t\right)$. Combining the first part of Eq. (A.10) and the last line of Eq. (A.13) yields the proof of the property.

Bibliography

[1] Katsikadelis, J.T.: *Dynamic Analysis of Structures.* 1st edn,Academic Press, (2020).

[2] Bathe, K.J.: *Finite element procedures.* 2nd edn, Prentice Hall, (2014).

[3] Chopra, A.K.: *Dynamics of Structures, Theory and Applications to Earthquake Engineering.* 5th edn, Pearson Education Limited, (2019).

[4] Wriggers, P.: *Computational Contact Mechanics.* John Wiley & Sons LTD, (2002).

[5] Gurtin, M., Fried, E., Anand, L.: *The Mechanics and Thermodynamics of Continua* Cambridge University Press, (2013).

[6] Chinesta, F., Keunings, R., Leygue, A.: *The Proper Generalized Decomposition for Advanced Numerical Simulations*. Springer (2014).

[7] Qu, Z.Q.: *Model Order Reduction Techniques with Application in Finite Element Analysis*. Springer-Verlag London Limited, London,(2004).

[8] Quesada, C., Gonzalez, D., Alfaro, I., Cueto, E., Chinesta, F.: *Computational vademecums for real-time simulation of surgical cutting in haptic environments.* International Journal for Numerical Methods in Engineering, (2015).

[9] Ghnatios, Ch., Masson, F., Huerta, A., Leygue, A., Cueto, E., Chinesta, F.: *Proper generalized decomposition based dynamic data-driven control of thermal processes.* Computer Methods in Applied Mechanics and Engineering; 213–216:29–41, (2012).

[10] Sanz-Serna, J.M.: *On finite elements simultaneously in space and time.* International Journal for Numererical Methods in Engineering; 19:623–624, (1983).

[11] Hughes, T.JR., Hulbert, G.M.: *Space-time finite element methods for elastodynamics: formulation and error estimates.* Computer Methods for Applied Mechanics in Engineering; 66:339–363, (1988).

[12] Idesman, A.V.: *Solution of linear elastodynamics problems with space-time finite elements on structured and unstructured meshes.* Computer Methods for Applied Mechanics and Engineering; 196:1787–1815, (2007).

[13] Blum, H., Jansen, T., Rademacher, A., Wienert, K. *A contact algorithm for the Signorini problem using space-time finite elements.* International Journal for Numerical Methods in Engineering; 76:1632–1644, (2008).

[14] Yang, Y., Chirputkar, S., Alpert, D.N., Eason, T., Spottswood, S., Qian, D.: *Enriched space-time finite element method: a new paradigm for multiscaling from elastodynamics to molecular dynamics.* International Journal for Numerical Methods in Engineering; 92:115–140, (2012).

[15] Takizawa, K., Moormann, C., Wright, S., Christopher, J., Tezduyar, T.E.: *Wall shear stress calculations in space-time finite element computation of arterial fluid–structure interactions.* Computational Mechanics; 46:31–41, (2010).

[16] Nickell, R.E.: *Nonlinear dynamics by mode superposition.* Computer Methods in Applied Mechanics and Engineering; 7(1):107–129, (1976).

[17] Cueto, E., González, D., Alfaro, I.: *Proper Generalized Decompositions: An Introduction to Computer Implementation with Matlab.* Springer, ISBN: 978-3-319-29994-5, (2016).

[18] Newmark, N.M.: *A Method of Computation for Structural Dynamics*. ASCE Journal of Engineering Mechanics Division, Vol 85. No EM3, (1959).

[19] Boucinha, L., Gravouil, A., Ammar, A.: *Space–time proper generalized decompositions for the resolution of transient elastodynamic models.* Computer Methods in Applied Mechanics and Engineering; 255:67-88, (2013).

[20] Arjoune, T., Markert, B., Bamer, F.,: *Non-incremental response evaluation in geometrically nonlinear structural dynamics using a space-time stiffness operator.* Computational Mechanics; 70:309-333, (2022).

[21] Bamer, F., *et al.*: *A Newmark space-time formulation in structural dynamics.* Computational Mechanics, 67:1331-1348, (2021).

[22] Chinesta, F., Ladeveze, P., Cueto, E.: *A short review on model order reduction based on proper generalized decomposition.* Archive of Computational Methods in Engineering; 18:395–404, (2011).

[23] https://strongmotioncenter.org/vdc/waveform_data/PEER/ath/KOBE/TAZ000.AT2

[24] Ammar, A. et al: *Non-Incremental strategies based on separated representations: Applications in computational rheology.* Communications in Mathematical Sciences; 8(3):671–695, (2010).

[25] Itskov, M.: *Tensor Algebra and Tensor Analysis for Engineers, with Applications to Continuum Mechanics.* Springer, (2007)

[26] Bamer F., Amiri, A.K.,Bucher C.: *A new model order reduction strategy adapted to nonlinear problems in earthquake engineering.* Eartquake Engineering & Structural Dynamics; 46(8):537-559, (2016).

[27] B. J.W. S. Lord Rayleigh: *Theory of Sound*, Vol.1 2nd. London and New York: Macmillan and Co., (1894).

[28] Kerschen G., Golivani J.C.: *Physical interpretation of the proper orthogonal modes using the singular value decomposition*, Journal of Sound and Vibration; 249:849–865, (2002).

[29] Bamer F., Bucher C.: *Application of the proper orthogonal decomposition for linear and nonlinear structures under transient excitation. Acta Mechanica*, **223**:2549-2563, (2012).

[30] Pissas, G.: *Nonlinear dynamic analysis of structures using model order reduction.* Diploma Thesis NTUA, (2021).

[31] Ladeveze, P.: *Nonlinear computational structural mechanics: New Approaches and Non-Incremental Methods of Calculation.* Springer, New York, (1999).

[32] Ammar, A. et al.: *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids.* Journal of Non-Newtonian Fluid Mechanics; 139:153–176, (2006).

[33] Ammar, A. et al.: *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II. Transient simulation using space-time separated representations.* Journal of Non-Newtonian Fluid Mechanics; 144:98–121, (2007).

[34] Modesto, D., Zlotnik, S., Huerta, A.: *Proper generalized decomposition for parameterized helmholtz problems in heterogeneous and unbounded domains: application to harbor agitation.* Computer Methods in Applied Mechanics and Engineering; 295:127–149, (2015).

[35] Henneron, T., Clenet, S.: *Proper generalized decomposition method applied to solve 3-d magneto quasi-static field problems coupling with external electric circuits.* IEEE Transactions on Magnetics; 51(6), (2015).

[36] Chinesta, F. et al.: *Efficient stabilization of advection terms involved in separated representations of boltzmann and fokker-planck equations.* Computer Physics Communications; 17(4):975–1006, (2015).

[37] Gonzalez, D. et al.: *Computational vademecums for the real-time simulation of haptic collision between nonlinear solids.* Computer Methods in Applied Mechanics in Engineering; 283:210–223, (2015).

[38] Ammar, A. et al.: *Parametric solutions involving geometry: a step towards efficient shape optimization.* Computer Methods in Applied Mechanics and Engineering; 268:178–193, (2014).

[39] Chinesta, F., Ammar, A., Cueto, E.: *Proper generalized decomposition of multiscale models.* International Journal for Numerical Methods in Engineering; 83(8–9):1114–1132, (2010).

[40] Cremonesi, M. et al.: *A PGD-based homogenization technique for the resolution of nonlinear multiscale problems.* Computer Methods in Applied Mechanics and Engineering; 267:275–292, (2013).

[41] Niroomandi, S. et al.: *Real-time simulation of biological soft tissues: a PGD approach.* International Journal for Numerical Methods in Biomedical Engineering; 29(5):586–600, (2013).

[42] Quesada, C. et al.: *Real-time simulation techniques for augmented learning in science and engineering.* The Visual Computer; 32:1465-1479, (2015).

[43] Papageorgiou, V.: Programming of the Proper Generalized Decomposition Method for the Prediction and/o
Diploma Thesis NTUA, (2017).

[44] Sibileau, A. et al.: *Explicit parametric solutions of lattice structures with proper generalized decomposition (PGD).* Computational Mechanics; 62(4):871–891, (2018).

[45] Garikapati, H. et al.: *A proper generalized decomposition (PGD) approach to crack propagation in brittle materials: with application to random field material properties.* Computational Mechanics; 65(2):451–473, (2020).

[46] González, D., Cueto, E., Chinesta, F.: *Real time direct integrationof reduced solid dynamics equations.* International Journal for Numerical Methods in Engineering; 99:633–653, (2014).

[47] Bergheau, M. et al.: *The proper generalized decomposition as a space-time integrator for elastoplastic problems.* Comptes Rendus Mécanique; 344:759–768, (2016).