# Isogeometric analysis with local adaptivity based on a posteriori error estimation for elastodynamics

**Peng Yu**

Supervisors: Dr. Pierre Kerfriden
Professor Stéphane Bordas

*A thesis submitted to the graduate school in fulfilment of the requirement for the degree of Doctor of Philosophy*

February, 21, 2019


Advanced Materials and Computational Mechanics Group
The School of Engineering, Cardiff University
Cardiff, Wales, UK

To the people loving me, and the people beloved.

# Summary

IsoGeometric Analysis (IGA) was invented to integrate the Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) into a unified process. According to the recent research, IGA performs a super convergence in case of vibration, and especially, it perfectly addresses the Gibbs phenomenon (fluctuation) occurring in discrete spectra when using standard Finite Element Method (FEM). However, due to the tensor-product structure of Non-Uniform Rational B-Splines (NURBS), it fails to achieve the local refinement, which restricts its application to engineering fields performing local characteristics that require local refinement, such as sharp geometrical feature and/or varying material properties.

In this context, the first goal of thesis is to extend the recently proposed paradigm, called Geometry Independent Field approximaTion (GIFT), to be applied in the scheme of dynamics. The GIFT methodology allows geometry of structure to be described within the NURBS provided directly by the existing CAD software, and solution field to be approximated by the Polynomial splines over Hierarchical T-meshes (PHT) with the feature of local refinement meanwhile.

Subsequently, in the framework of GIFT, an adaptivity technique based on hierarchical a posteriori error estimation on the modal vector is established for the free vibration of thick plate. The proposed adaptive mesh achieves a faster convergence than uniform refinement. Especially, the employment of Modal Assurance Criterion (MAC)-style strategy is able to better determine the modal correspondence between coarse and fine discretizations than Frequency Error Criterion (FEC) method.

Furthermore, based on hierarchical a *posteriori* error estimation strategy, three types of adaptivity algorithms are constructed to deal with the space-time refinement. Specially, unidirectional multi-level space-time adaptive GIFT/Newmark (UM-STAGN) well catches stress wave propagation but fails in error information transfer. Energy-based space-time adaptive GIFT/Newmark (E-STAGN) can reassess the error but

cannot uncover the source of error. Dual weighted residual adaptive GIFT/Newmark (DWR-STAGN) methods are error-sensitive so that it leads to the best convergence among these three approaches.

# Declaration

DECLARATION

This work has not been submitted in substance for any other degree or award at this or any other university or place of learning, nor is being submitted concurrently in candidature for any degree or other award.

Signed..... (Candidate) Date.....

STATEMENT 1

This thesis is being submitted in partial fulfillment of the requirements for the degree of (insert MCh, MD, MPhil, PhD etc, as appropriate)

Signed..... (Candidate) Date.....

STATEMENT 2

This thesis is the result of my own independent work/investigation, except where otherwise stated, and the thesis has not been edited by a third party beyond what is permitted by Cardiff Universitys Policy on the Use of Third Party Editors by Research Degree Students. Other sources are acknowledged by explicit references. The views expressed are my own.

Signed..... (Candidate) Date.....

STATEMENT 3

I hereby give consent for my thesis, if accepted, to be available online in the Universitys Open Access repository and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed..... (Candidate) Date.....

STATEMENT 4

I hereby give consent for my thesis, if accepted, to be available online in the Universitys Open Access repository and for inter-library loans **after expiry of a bar on access previously approved by the Academic Standards & Quality Committee.**

Signed..... (Candidate) Date.....

# Acknowledgments

First and foremost, I would like to present my great appreciation to my supervisor Dr.Pierre Kerfriden. To be honest, working with him is a big challenge. He is brilliant but tough so that I could not follow his mind in most cases. However, thanks to his intelligence, it opens my eye to see the principle of computational mechanics and where its beauty arises from. Also, he strictness makes me learn the critical thinking and what the good quality of research should be like. Regardless of the stress and pain, I would rather take it as the very best time during my PhD period.

The sincere acknowledgment gives to my another supervisor Professor Stéphane Bordas. I have been impressed by his talent, passion, and humor since the first time we met in Cardiff. I personally quite appreciated that he invited me twice to visit in University of Luxembourg. There I witnessed the group full of smart researches and advanced scientific topics they were working on. Meanwhile, I was surprised by the excellent management skill shown by Stéphane to make the team rum smoothly, which is really inspired to my following academia career.

The gratitude is given to Dr.Cosmin Anitescu and Dr.Satyendra Tomar. It was my pleasure to work with them on a high-quality paper and it certainly got published in a high-quality journal ultimately. Thanks for their valuable scientific contributions, and the patience to the thousands of mistakes I made.

I would like to thank the financial support from China Scholarship Council (CSC).

The special thanks are delivered to Yue Du, for the selfless support, and more importantly, for letting me realize what the person I really would like to be.

In addition, thanks to my colleagues and friends in my office. The list of name is too long to be presented here. But the great time we shared together will be the precious memory in my life.

Last but never the least, I am grateful to my family. Without you, the happiness would be meaningless and the sadness would be nowhere to settle. Our mutual love is and will always be my motivation leading to a good life.

# List of publications

**Journal papers**:

1. **Yu, P.**, Anitescu, C., Tomar, S., Bordas, S. P. A., and Kerfriden, P. (2018). Adaptive Isogeometric analysis for plate vibrations: An efficient approach of local refinement based on hierarchical a posteriori error estimation. Computer Methods in Applied Mechanics and Engineering, 342, 251-286.

2. **Yu, P.**, Bordas, S. P. A., and Kerfriden, P. Adaptive Isogeometric analysis for transient dynamics: three types of space-time refinement methods based on hierarchical a posteriori error estimations. Computer Methods in Applied Mechanics and Engineering. (In preparation)

**Conference papers**:

1. **Yu, P.**, Claus, S., Bordas, S. P. A., and Kerfriden, P. Error estimation and space-time adaptivity for the isogeometric analysis of transient structural dynamics. 24th Conference on Computational Mechanics, Cardiff, United Kingdom, 31th, March-1st, April, 2016.

2. **Yu, P.**, Anitescu, C., Bordas, S. P. A., and Kerfriden, P. Isogeometric analysis with local adaptivity for vibration of Reissner-Mindlin plate. 25th Anniversary of the European Community on Computational Methods in Applied Sciences (ECCOMAS), Glasgow, United Kingdom, 11th-15th, June, 2018.

# Acronyms

**CAD** Computer-Aided Design. iii, 2

**CAE** Computer-Aided Engineering. iii

**DWR** Dual Weighted Residual. 4

**DWR-STAGN** Dual-Weighted Residual Space-Time Adaptive GIFT/Newmark. 7

**E-STAGN** Energy-based Space-Time Adaptive GIFT/Newmark. 7

**FEA** Finite Element Analysis. 2

**FEC** Frequency Error Criterion. iii

**FEM** Finite Element Method. iii, 2

**GIFT** Geometry Independent Field approximaTion. 5

**IGA** IsoGeometric Analysis. iii, 2

**NURBS** Non-Uniform Rational B-Splines. iii, 2

**ODEs** Ordinary Differential Equations. 1

**PDEs** Partial Differential Equations. 1

**PHT** Polynomial splines over Hierarchical T-meshes. iii, 4

**QoI** Quantity of Interest. 4

**RHT** Rational splines over Hierarchical T-meshes. 4

**ST** space-time. 4

**UM-STAGN** Unidirectional Multi-level Space-Time Adaptive GIFT/Newmark. 6

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

The study of structural dynamics is an issue of specific research fields including: civil engineering, aeronautics, automotive, robotics, and so on. With respect to the simulation on structural dynamics, it is often required to solve Ordinary Differential Equations (ODEs) or Partial Differential Equations (PDEs).

The ODEs solvers are commonly regarded as time integration numerical methods. They are most comprehensively classified as explicit, implicit and hybrid approaches. The characteristics of these strategies are that selected time step should neither too small, which makes the time to convergence never reaches, nor too large, which leads to the instability. The explicit algorithms, primarily involving forward/explicit Euler method [1], explicit Runge-Kutta methods [2], leapfrog/velocity Verlet methods [3], predict the solution at forward time based on the solution at current step using the differential equation. Consequently, explicit methods are stable in the case of suitable adopted time step size. Implicit Methods, i.e., backward/implicit Euler method [4], Crank-Nicholson method [5], Gear's/backward Differentiation formula Method [6], aim to meet equilibrium at current time step as long as the solution at forward time step is satisfied. In this scheme, one has to solve a group of linear systems but it ensures a good or even unconditional stability.

The numerical method to address PDEs is kind of matter integration. The-

oretically, the number of degrees should be infinite to represent the dynamics of matter. But actually, it is essential to control number of degrees to be finite, such as discretization, in order to solve the problem numerically. That is the basic idea of the mesh-based methods. FEM [7] is constructed based on the configuration of elements. The flexibility of the shape and size of the element provides a very rich set of powerful and efficient tools. However, it would get stuck when the mesh is heavily distorted. The situation can be improved using the finite difference method (FDM) [8] where the shape and volume of elements keep consistent during the computation. Nevertheless, note that, for a time domain problem, if the uniform discretization can not be promised over the whole time domain, FDM would be less steady and reliable compared with FEM, especially for the structure with irregular geometry. For sake of getting rid of mesh limitation, meshless method [9] was proposed to deliver robust calculation for PDEs with various possible boundary conditions using a set of particles instead of any mesh. Specially, as one of meshfree method, smoothed particle hydrodynamics (SPH) [10], where the system state is described by a set of particles holding independent material characters and varies according to the governing conservation equations, is quite suitable to hydrodynamics.

### 1.1.1 Why IGA

In various engineering applications, the engineers generally employ CAD to design the geometry of structures, and exploit Finite Element Analysis (FEA) to proceed the simulations. Nevertheless, the analysis can be conducted only when the representation of CAD is converted into FEA-fit pattern. This conversion usually encompasses dull and cumbersome work, and more crucially, it will lead to the geometric information loss, since the standard Lagrange basis functions in FEA are not able to describe curves exactly. This was the motivation for the establishment of IGA by Hughes et al.[11]. The core concept of IGA is to apply the same smooth and high-order spline basis functions, such as NURBS, most widely used in CAD software, to represent both the geometry and the solution field in FEA. Then the purpose of IGA becomes very intuitive that it aims to assemble design-through-analysis into a

single, unified process.

IGA holds a lot of attractive benefits for industry applications. For instance, for the issues that only need mesh at boundary, Isogeometric boundary element methods [12], where grids used for solution field can be obtained from CAD straightforward, performs well. In addition, the research fields on shell are where IGA presents more competitive merits in contrast of conventional methods [13]. The high continuity of basis functions used in IGA endows a direct establishment of plate/shell elements. Especially for thin shells, rotation-free equation is able to be built readily [14]. What is more, isogeometric plate/shell elements is less likely to experience shear-locking behavior than typical finite plate/shell elements.

The smoothness of IGA basis functions are also productive to study fluids [15] and fluid-structure interaction problems [16]. Furthermore, owing to the relief of creating high order basis functions, IGA has been successfully employed to solve fourth(or higher)-order PDEs, (i.e., the Hill-Cahnard equation [17], explicit gradient damage models [18] and gradient elasticity [19]). It also owns potential utilizations in the Kohn-Sham equation normally existing in modelling electronic structure for semiconducting materials [20].

Structural dynamics (hyperbolic problem in sense of mathematics), e.g., vibration and wave propagation, plays a significant role in mechanical and civil engineering, including aircrafts, vehicles, bridges, buildings, etc.. The expectation to achieve a high-order precision by using explicit FEM have been lost for ages. This is because the high-oder finite elements fail to perform a robustness in representation of high modes, which was recognized in recent researches (cf. [21, 22, 23, 24]). Moreover, through their investigations, it was found that different from FEM, the high modes approximated by IGA are convergent with the increase of degrees of freedom (dofs). It means the shortcoming of using conventional high-order FEM to compute high-frequency problem is overcome by IGA method.

### 1.1.2 Why adaptivity

Note that numerical method is conducted by discretizing computational fields. As a result, a sufficient fine mesh is usually the guarantee for a convinced solution. However, for structural dynamics, especially transient dynamics, it is a space-time (ST) problem, where the discretizations include spatial and temporal discretizations. In this sense, if the fine uniform meshes are carried out for both space and time domain simultaneously, it would soon lead to the limit of computer resources. Consequently, it is required to find the balance between cost and accuracy. That is the inspiration for adaptivity technique.

A reliable adaptive approach is heavily dependent on a locally effective and quantitative a *posteriori* error estimation. According to the measured objects, the error estimations can be categorized as two types: (1) *norm-based* error estimation [25] and (2) *goal-oriented* error estimation [26]. The previous one aims to assess a norm of the error (i.e., for displacement, energy) on the whole computational field so that it is viewed as a *global* estimator. Whilst the latter one is designed to evaluate the error of a representative scalar, which can be extracted by the integral of a function over a local area, and hence it is considered as a *local* estimator. That scalar variable is the so-called Quantity of Interest (QoI), and the methodology to assess the error of QoI is the goal-oriented error estimation. In order to obtain the error of QoI, it is required to solve an auxiliary problem, namely, the *adjoint* or *dual* problem. The relevant technique to tackle this problem is referred as the Dual Weighted Residual (DWR) method [27].

In addition, adaptivity requires a local refinement technique. It is easy to be realized in FEM, as the Lagrange basis functions are linearly independent. Yet for IGA, since the NUBRS is constructed by tensor product, the refinement is restricted to be the global grids (see Fig.5.1(a)). This limitation triggered the development of alternative spline-basis functions with local refinement characters, i.e., T-splines [28], PHT and Rational splines over Hierarchical T-meshes (RHT) [29, 30], hierarchical B-splines (HB-splines) [31], truncated hierarchical B-splines (THB-splines) [32], locally refined B-splines (LR B-splines) [33], Powell-Sabin splines [34], Patchwork B-spline

[35], etc.

## 1.2 Aims and thesis layout

In this thesis, it is intended to propose a methodology of adaptive Isogeometric analysis to investigate linear elastodynamics, including vibration and wave propagation. This adaptive IGA method is based on hierarchical a *posteriori* error estimation, and with an efficient local refinement approach. The specific contributions of thesis are outlined by following:

- Although local-refinement splines can replace NUBRS, these alternatives are not perfect. Because, firstly, the aforementioned local-refinement splines, except T-splines and RHT-splines, are not rational. It means they are not able to exactly represent the basic geometrical features, e.g. circles, ellipses, that are ubiquitous in engineering design and analysis. Second, even though this problem may be circumvented by applying T-splines or RHT-splines, unlike NURBS, T-splines and RHT-splines, cannot be seamlessly extracted from existing CAD softwares. These difficulties prompt us to look at a recently developed method, Geometry Independent Field approximaTion (GIFT). The GIFT scheme was firstly proposed by Marussig et al. [36], and then was improved to solve static issues [37, 38, 39]. Briefly, the virtue of GIFT is that we can use NURBS to precisely describe the geometry of structure, and simultaneously employ arbitrary local-refinement splines (we choose PHT-splines in this work) to approximate the solution domains. In this thesis, the GIFT method is attempted to solve the elastodynamic problems.

- For vibration, we propose a novel frequency-domain adaptation strategy based on a *posteriori* error estimation on modal shapes and mode sweeping. Closely related to the proposed adaptivity scheme is the method described in [40], whereby RHT splines are used to obtain a higher convergence rate of free vibration frequencies when compared to that observed when using tensor-product-based NURBS. However, the local refinement of the aforementioned

study is driven by a priori error estimation, which does not provide any quantitative measure of accuracy (i.e. it only provides a spatial map of error sources). It is aimed to develop a comprehensive refinement strategy, which includes a reliable stopping criterion as provided by a *posteriori* error estimation (see for instance [41, 42, 43, 44]). More precisely, we will define a hierarchical a *posteriori* error estimator that makes the best of the PHT-spline local refinement capabilities. More precisely, the accuracy of GIFT solutions will be estimated by computing refined solutions using a finer mesh generated by systematically subdividing every GIFT element. The mesh adaptivity will be performed for every free vibration mode (i.e. frequency and associated mode shape) within a frequency band, sweeping from lower to higher frequencies. The algorithm requires for coarse and fine GIFT estimations of the modes to be put in correspondence in order to be compared. This is not a trivial task. A new algorithm inspired by the Modal Assurance Criterion (MAC) strategy is proposed, which is widely used in experimental structural dynamics [45, 46]. It will be shown that the proposed algorithm is robust with respect to the order of multiplicity of the fine and coarse modes.

- In the context of transient dynamics, the spatial domain is discretized by GIFT method, and the time discretization is processed by Newmark method. Inspired by the work [47], where a space-time unidirectional-multilevel adaptive approach in the framework of FEM/Newmark was proposed, it is extended to be the Unidirectional Multi-level Space-Time Adaptive GIFT/Newmark (UM-STAGN). In the framework of this UM-STAGN scheme, at every adaptation step, the domains where error estimators satisfy the tolerance are not required to be computed any more. Only the remaining domains, denoted as the sub-domains (or sub-level systems), are computed next adaptive step. Repeat this process until the error of all domains is underneath the threshold. Thus the solution of the whole system can be considered as the composition by solutions of all sub-level systems. The merit of UM-STAGN method is to save computational resources by degrading the system to be a smaller size

gradually.

- The UM-STAGN method may be efficient, though, since this multilevel algorithm is unidirectional (always goes oriented to the finer mesh), and only information at interface between two level meshes can be transfered, it will lead to the accuracy limitation of the solution, especially in case of wave propagation. Hence, Energy-based Space-Time Adaptive GIFT/Newmark (E-STAGN) method is proposed in this work. In contrast of UM-STAGN method, E-STAGN technique is able to achieve a better accuracy. This is because at each step in E-STAGN scheme, the computation is gone through the whole system, and the error of each space-time element will be updated, according to energy-based a *posteriori* error estimation. In such a way, the error propagation can be taken into account.

- Even though E-STAGN method is able to trace the error transmission, it is not efficient enough. As we know, in wave propagation problem, the error is accumulated along with time, leading to the phenomenon that the refinement gets finer and finer with time increase. It means the error identified by E-STAGN method actually includes a lot of pollution. To avoid the wastage of refinement in theses polluted areas, and discover the source of the error instead, the Dual-Weighted Residual Space-Time Adaptive GIFT/Newmark (DWR-STAGN) method is created. Using DWR-STAGN method, besides solving an original (primal) problem, it is required to solve another dual problem, which is a wave equation running backward in time. Then, exploiting dual weighted residual consisting of primal and dual solutions as the error estimator, it can identify the sensitivity of the error for QoI and then drive an efficient adaptive refinement.

The thesis is organized as follows:

- In Chapter 2, the basic idea of IGA is reviewed.

- In Chapter 3, the classic theory of a *posteriori* error estimation is presented, and several types of specific error estimation methods are discussed.

- In Chapter 4, the conceptual difference of IGA and GIFT is compared, and PHT splines and RHT splines are introduced.
- In Chapter 5, a novel methodology of local adaptivity for the frequency-domain analysis of the vibration of Reissner-Mindlin plate is proposed.
- In Chapter 6, three space-time adaptive strategies (UM-STAGN, E-STAGN and DWR-STAGN) are developed, and comparisons are made within the numerical examples.
- In Chapter 7, the thesis is closed with the drawn conclusions and outlooks of the future work.

# Chapter 2

# A brief introduction to IGA

During the last decades, IGA has been widely used in diverse problems, including shell [13] and plate issues [48, 49, 50], fracture and damage mechanics [51, 52] which involves the delamination of composite materials [53], fluid-structure interaction [16, 54], contact problem [55, 56]. Due to the tight interaction with CAD, IGA has been successfully utilized in the structural shape and topology optimization [57, 58] as well. Despite the fact that direct generation of volumetric parameterizations from CAD programme still requires more efforts [59, 60], the description for geometric boundary can be obtained straightforward, which delivers the IGA coupling with Boundary Element Method (BEM)[12, 61, 62].

In this Chapter, the basic idea of IGA is introduced, including the most widely used splines, i.e., B-splines and NURBS, the mapping among different spaces, the multi-patch coupling methods, locally-refined splines and some alternative techniques for geometric representation and analysis.

## 2.1 B-splines and NURBS

### 2.1.1 B-splines

The univariate B-spline functions with $p$ order are defined in a recursive form over a knot vector $\Xi = \{\xi_1, \ldots, \xi_i, \ldots, \xi_n\}$ using the following formula [63]

for $p = 0$

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{2.1.1}$$

and $p \geqslant 1$

$$B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi). \tag{2.1.2}$$

A representative example of a set of quadratic B-spline shape functions defined in an open, non-uniform knot vector is presented in Fig.2.1. Due to marvelous features, the B-splines are kept being used since the CAD technology was initiated. The characteristics are such as [63]: 1) partition of unity; 2) non-negativity; 3) linear independence; 4) local support that $B_{i,p}$ is non-zero in the given $p+1$ knot span $(\xi_i, \xi_{i+p+1})$; 5) $C^{p-m_i}$ continuity at knot $\xi_i$ (see Fig.2.1) where $m_i$ is the multiplicity of knot $\xi_i$; 6) generally non-interpolation which allows more flexibility for geometric description. To be specific, partition of unity denotes that the sum of all B-spline basis functions at arbitrary knot point is equal to 1, which reads: $\sum_{i=1}^{n} B_{i,p}(\xi) = 1$. It is especially obvious at $\xi = 0$ and $\xi = 1$. For example, at $\xi = 0$, there exists $B_{1,2}(0) = 1$, $B_{2,2}(0) = 0$ and $B_{1,2}(0) = 0$, and thus it yields $B_{1,2}(0) + B_{2,2}(0) + B_{1,2}(0) = 1$. In addition, the local support property indicates the shape function is only non-zero in a specific open knot span. For instance, $B_{3,2}$ is positive over the interval $(\xi_3, \xi_6)$, namely, $(0, 0.6)$. Especially, as to the basis function such as $B_{5,2}$, the relevant knot span is $(\xi_5, \xi_8)$, that is $(0.4, 0.6, 0.8, 0.8)$. Because $\xi = 0.8$ is a double knot point, the actual resulting non-zero knot interval for $B_{5,2}$ is $(0.4, 0.8)$. $B_{1,2}$, $B_{2,2}$, $B_{6,2}$, $B_{7,2}$, and $B_{8,2}$ are in the similar situations. In Fig.2.1, three notations of continuity $C^{-1}$, $C^0$, $C^1$ are marked based on the principle $C^{p-m_i}$. Then, it can be learnt that, by taking $B_{5,2}$ as an example, at knot point $\xi = 0.4$ and $\xi = 0.6$, the continuity is $C^1$, while at $\xi = 0.8$, the continuity is $C^0$. Except that, $B_{5,2}$ over the knot span $(0.4, 0.6)$ and $(0.6, 0.8)$ are with $C^2$ continuity.

Furthermore, the first derivative of B-spline is also defined in a recursive means

$$\frac{d}{d\xi} B_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi). \tag{2.1.3}$$

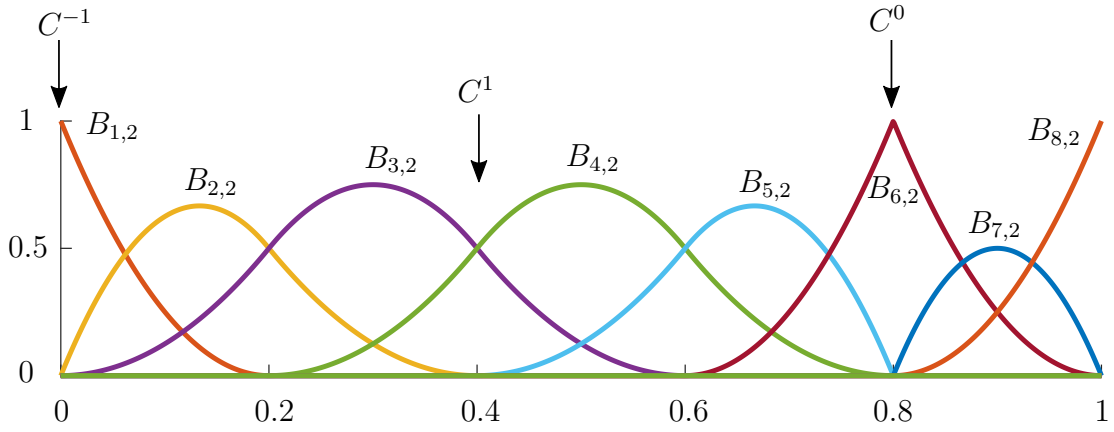



Figure 2.1: Quadratic B-spline shape functions defined in an open, non-uniform knot vector $\Xi = \{0, 0, 0, 0.2, 0.4, 0.6, 0.8, 0.8, 1, 1, 1\}$

.

The B-spline curves can be defined as

$$\boldsymbol{C}(\xi) = \sum_{\boldsymbol{k} \in \boldsymbol{I}_n} B_{\boldsymbol{k},p}(\xi) \boldsymbol{P}_{\boldsymbol{k}}, \tag{2.1.4}$$

where $\boldsymbol{P}_{\boldsymbol{k}} \in \mathbb{R}^d$ are control points, $\xi \in \mathcal{P}^1$ is coordinate belonging to parametric space $\mathcal{P}$. A case of quadratic B-spline curve is presented in Fig.2.2. As it can be seen, as referred previously, except the first and last points, the other control points are non-interpolatory.

Furthermore, given a net of knot vectors $\Xi_\xi \times \Xi_\eta$, a tensor-product B-spline surface can be expressed as

$$\boldsymbol{S}(\boldsymbol{\xi}) = \sum_{\boldsymbol{k} \in \boldsymbol{I}_{n \times m}} \boldsymbol{B}_{\boldsymbol{k}}^{p,q}(\boldsymbol{\xi}) \boldsymbol{P}_{\boldsymbol{k}}, \tag{2.1.5}$$

where $\boldsymbol{B}_{\boldsymbol{k}}^{p,q} = B_{i,p}(\xi) B_{j,q}(\eta)$ and $\boldsymbol{\xi} = (\xi, \eta) \in \mathcal{P}^2$. Thus, the B-spline volume is readily to be obtained through

$$\boldsymbol{V}(\boldsymbol{\xi}) = \sum_{\boldsymbol{k} \in \boldsymbol{I}_{n \times m \times l}} \boldsymbol{B}_{\boldsymbol{k}}^{p,q,r}(\boldsymbol{\xi}) \boldsymbol{P}_{\boldsymbol{k}}, \tag{2.1.6}$$

with $\boldsymbol{B}_{\boldsymbol{k}}^{p,q,r} = B_{i,p}(\xi) B_{j,q}(\eta) B_{k,r}(\zeta)$ and $\boldsymbol{\xi} = (\xi, \eta, \zeta) \in \mathcal{P}^3$.

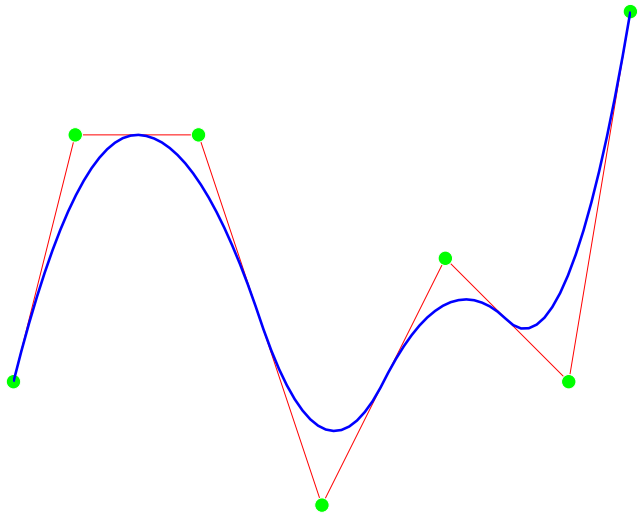


Figure 2.2: An example of a quadratic B-spline curve (blue line). The green solid circles denote control points.

### 2.1.2 Refinement

There are 3 sorts in terms of refinement strategies for B-splines in the framework of IGA, that is, knot insertion, degree elevation, and $k-$refinement. The knot insertion and degree elevation are analogous to $h-$refinement and $p$-refinement in conventional FEM. Whereas, the $k-$ refinement is distinctive to IGA. It is realized through degree elevation firstly and knot insertion afterwards, which contributes to save degrees of freedom compared to the conventional $p-$refinement [11]. For example, given a knot vector $\Xi = \{0, 0, 1, 1\}$, within the classic $p-$refinement method, the knot insertion is followed by order raise, that is, the knot vector turns to be $\Xi = \{0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1\}$ and extended to be $\Xi = \{0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1\}$ later on. While using $k-$refinement, the order elevation allows the knot vector to be $\Xi = \{0, 0, 0, 1, 1, 1\}$ firstly and become $\Xi = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$ by knot insertion after that. Obviously, within the $k-$refinement, it costs less dofs than traditional $p-$refinement and meanwhile holds the higher interior continuity. This advantage will become more apparent supposed that more elements are desired to be created during the discretization.

### 2.1.3 NURBS basis functions

Although B-splines are equipped with various merits mentioned above, they are not able to precisely depict the geometry such as circle, ellipse, etc. Therefore, nowadays, the state-of-the-art CAD technology applies NURBS for geometric generation. NURBS inherits all the beneficial properties of B-splines, and meanwhile owns the capability to accurately represent shapes, i.e., spheres, ellipsoids.

The univariate NURBS shape functions are defined by [53]

$$N_{i,p}(\xi) = \frac{B_{i,p}(\xi)w_i}{W(\xi)} = \frac{B_{i,p}(\xi)w_i}{\sum_{\hat{i}=1}^{n} B_{\hat{i},p}w_{\hat{i}}}, \tag{2.1.7}$$

where $w_i$ are a set of weights. Intuitively, NURBS are constructed by taking weights into account based on the B-splines. Then, we can also derive the first derivative of NURBS as

$$\frac{d}{d(\xi)}N_{i,p}(\xi) = w_i\frac{B'_{i,p}(\xi)W(\xi) - B_{i,p}(\xi)W'(\xi)}{W^2(\xi)}, \tag{2.1.8}$$

where

$$B'_{i,p}(\xi) = \frac{d}{d\xi}B_{i,p}(\xi),\ W'(\xi) = \sum_{\hat{i}=1}^{n} B'_{\hat{i},p}w_{\hat{i}}. \tag{2.1.9}$$

Similar to the B-splines, the curve $\boldsymbol{C}(\xi)$, surface $\boldsymbol{S}(\boldsymbol{\xi})$ and volume $\boldsymbol{V}(\boldsymbol{\xi})$ of NURBS are built respectively as below

$$\boldsymbol{C}(\xi) = \sum_{\boldsymbol{k}\in\boldsymbol{I}_n} N_{\boldsymbol{k},p}(\xi)\boldsymbol{P}_{\boldsymbol{k}}, \tag{2.1.10}$$

$$\boldsymbol{S}(\boldsymbol{\xi}) = \sum_{\boldsymbol{k}\in\boldsymbol{I}_{n\times m}} \boldsymbol{N}_{\boldsymbol{k}}^{p,q}(\boldsymbol{\xi})\boldsymbol{P}_{\boldsymbol{k}}, \tag{2.1.11}$$

$$\boldsymbol{V}(\boldsymbol{\xi}) = \sum_{\boldsymbol{k}\in\boldsymbol{I}_{n\times m\times l}} \boldsymbol{N}_{\boldsymbol{k}}^{p,q,r}(\boldsymbol{\xi})\boldsymbol{P}_{\boldsymbol{k}}, \tag{2.1.12}$$

where

$$\boldsymbol{N}_{\boldsymbol{k}}^{p,q}(\boldsymbol{\xi}) = \frac{B_{i,p}(\xi)B_{j,q}(\eta)}{\sum_{\hat{i}=1}^{\hat{n}}\sum_{\hat{j}=1}^{\hat{m}} B_{\hat{i},p}(\xi)B_{\hat{j},q}(\eta)w_{\hat{i},\hat{j}}}, \tag{2.1.13}$$

$$\boldsymbol{N}_{\boldsymbol{k}}^{p,q,r}(\boldsymbol{\xi}) = \frac{B_{i,p}(\xi)B_{j,q}(\eta)B_{k,r}(\zeta)}{\sum_{\hat{i}=1}^{\hat{n}}\sum_{\hat{j}=1}^{\hat{m}}\sum_{\hat{k}=1}^{\hat{l}} B_{\hat{i},p}(\xi)B_{\hat{j},q}(\eta)B_{\hat{k},r}(\zeta)w_{\hat{i},\hat{j},\hat{k}}}. \tag{2.1.14}$$

## 2.2 IGA finite element space

When using the NURBS-IGA, the computation is conducted across several discrete domains. There are reference (parent) domain, parametric domain, and physical domain respectively, exhibited in Fig.2.3. The reference space used for the allocation of Gaussian quadrature points, and the physical domain employed to represent the structure shape has the definitions in accordance with that in FEM. While the parametric space is unique for IGA, which is used to define the spline basis functions.

Suppose that we would like to compute an integration in the physical domain, the following formulation is used, such as

$$\int_{\Omega} f(\boldsymbol{x}) d\Omega = \int_{\tilde{\mathcal{P}}} f(\tilde{\boldsymbol{\xi}}) |\boldsymbol{J}(\boldsymbol{\xi})| |\boldsymbol{J}(\tilde{\boldsymbol{\xi}})| d\tilde{\mathcal{P}}, \tag{2.2.15}$$

where $|\cdot|$ denotes a determinant. Jacobian matrix of transformation $\boldsymbol{J}(\tilde{\boldsymbol{\xi}})$ indicates the mapping from reference space to parametric space, and $\boldsymbol{J}(\boldsymbol{\xi})$ denotes the projection from parametric domain to physical domain. In case of 2D domain, the expressions of $\boldsymbol{J}(\tilde{\boldsymbol{\xi}})$ and $\boldsymbol{J}(\boldsymbol{\xi})$ are given by

$$\boldsymbol{J}(\tilde{\boldsymbol{\xi}}) = \begin{bmatrix} \dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial \xi}{\partial \tilde{\eta}} \\ \dfrac{\partial \eta}{\partial \tilde{\xi}} & \dfrac{\partial \eta}{\partial \tilde{\eta}} \end{bmatrix}, \ \boldsymbol{J}(\boldsymbol{\xi}) = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}. \tag{2.2.16}$$

## 2.3 Coupling of multiple patches

Although IGA offers a high-continuity approximation to solution field, it is still a challenge to realize a global high continuity ($\geqslant C^1$) when the geometry is constructed through multiple patches.

Some efforts have been contributed to enhance the smoothness across the multi-patch boundary surface. Among them, the most well-developed and robust method is supposed to be the Nitsches method [64, 65, 66]. In the framework of Nitsches method Lagrange multipliers are alternated, through incorporating an mesh-relied penalty-like term to recover the bilinear-type coercivity to accomplish the coupling

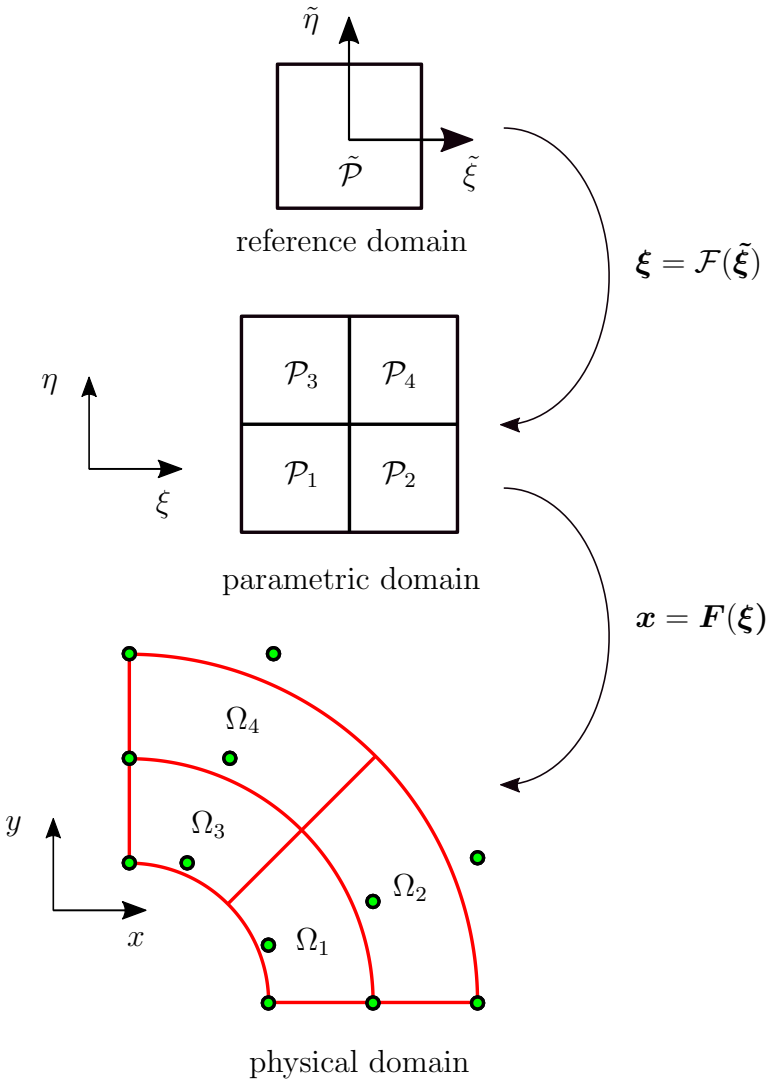



Figure 2.3: The illustration of mappings across over different computational domains.

.

of non-conforming patches. Other approaches involve bending strip method invented by Kiendl et al. [14] and then adopted to glue trimmed NURBS patches [67]. The mortar method couples the neighboring patches through a master-slave connection, and then utilizes Lagrange multipliers to improve smoothness in terms of variation [68].

An alternative means to guarantee the continuity request is to create a space with higher order defined basis functions to approximate the original $C^0$ or $C^1$ space at interface. Kapl et al.[69, 70] derived the $C^1$ and $C^2$ continuity piecewise polynomial functions by symbolic algebra to combine bilinear planar patches. However, Collin et al. [71] pointed out that over controlling of the piecewise polynomial basis in sense of a prescribed degree of freedom would give rise to $C^1$ locking problem. This triggered the establishment of analysis-feasible parameterizations applicable to bilinear-similar planar mappings [70].

## 2.4 Splines with local refinement

In the context of analysis, the shortcoming of NURBS is its tensor-product-based construction, being responsible for the inefficiency in adaptivity strategy driven by error estimation. This problem evoked the development of local-refinement splines.

T-splines were firstly proposed [28] in order to fix the imperfection of NURBS in watertight geometric generation, and on the other hand, it is free to be locally refined or coarsened due to the character of linear independence. Upon on this basis, the Locally Refined (LR) and hierarchical T-splines using the Bézier extraction algorithm were proposed [72, 73, 74]. Also, Polynomial splines over Hierarchical T-meshes (called PHT-splines) and its rational form, namely, Rational splines over Hierarchical T-meshes (RHT-splines) were introduced [29, 30]. Note that PHT-splines are a generalization of B-splines over hierarchical T-meshes, in essence. PHT-splines and RHT-splines will be discussed concretely in Chapter 4.

Different from T-spline family, where the local refinement is attributed to inserting knot repeatedly into the current B-spline cell to ensure linear independence of

basis functions inside, hierarchical B-splines [75, 76, 77] achieve the local refinement using a refined tensor-product domain to cover a coarse tensor-product domain, instead of building hierarchical refinement based on the coarse mesh like PHT-splines. Yet, overlapping-level spline surfaces may lead to the sacrifice of the degree of freedom. To solve this problem, Giannelli et al. [78] constructed truncated hierarchical B-spline (THB-spline), which diminishes repeated support for shape functions located in the common areas shared by different hierarchical levels, and facilities the numerical features of the related basis function at the same time. Nevertheless, the hierarchical mesh gives rise to the problem that refinement from coarse level to fine level strictly follows a single nested sequence. It results in the bound to generate some desired discretization (cf. [79]) during the refinement since the gird is heavily dependent to the initial mesh. In this circumstance, Engleitner et al. [79] introduced the Patchwork B-splines defined in a multivariate space, wherein splines with arbitrary continuity and smoothness can be combined. Moreover, through choosing the form of tensor-product B-splines suitably and constructing an updated truncation principle, the flexibility improvement of hierarchical refinement is achieved. Another locally refined splines are Powell-Sabin splines described over a macro-structure triangulation [80]. This triangulated construction is advantageous since it instinctively integrates IGA and FEM (where triangle elements are widely used).

## 2.5 Alternatives to IGA

Subdivision surface paradigm is one of the existing technologies that aim to assemble CAD and analysis for thin-shell modeling. In this method [81], the undeformed status of the shell shape is characterized, and the displacement is obtained within the smooth interpolation holding a finite energy in the scheme of Kirchoff-Love theory. Another technique is immersed finite element method designed to address the geometrically and topologically sophisticated problem [82]. This method carries out the geometric representation and finite element solution for an enclosed district of interest within a block structured logically Cartesian mesh, and exploits improved

B-spline shape functions to approximate the Dirichlet boundary conditions by local interpolation. In the context of discontinuous issues, like fracture, IGA generally performs bad accuracy, whilst extended finite element method (XFEM) is capable to deal with problem, by locally enriching a approximated space using partition-of-unity to characterize embedded singularity and discontinuity [83]. Another option for discontinuous problem is the CutFEM method, where the inserted discontinuity is modeled with fictitious filed strategy and robustness is promised via a known "ghost-penalty" mechanism [84].

## 2.6 Conclusion

The basic concept of IGA was presented in this Chapter. Firstly, the B-splines were introduced, and then their rational forms, namely, NURBS, mostly ubiquitously used in CAD softwares, were discussed. The algorithm of $h-$refinement and $p-$refinement is consistent to that of FEM, whereas the $k$-refinement is particular to IGA. The computation in IGA is required to go through two mappings. One is from reference space to parameter space, and the other is from parameter domain to physical domain. Several approaches were refereed for the multi-patch coupling, such as Nitsche's method, bending strip strategy, mortar paradigm, and high-order space approximation methodology. Two kinds of local-refinement splines were described. The family of T-splines are constructed inserting additional knots to enable the linear independence of functions. Whilst, the series of (truncated) hierarchical B-splines are formulated within the overlapping sets of finer level space on the coarse mesh.

Here, it is should emphasized that although T-splines can achieve adaptive refinement, the algorithm of knot insertion and weight update is complicated during refinement. Therefore, it is intended to employ PHT-splines for local refinement in this work. PHT-splines are the special forms of T-splines, which are constructed based on classical Hermite element. They inherit the advantages of T-splines and the relevant refinement algorithm is simple to be implemented. More details will

be discussed in Chapter 4. Finally, some alternative techniques to IGA were introduced to let readers realize the limits of IGA and be able to find the solutions. These techniques involve subdivision surface approach, immersed finite element method, XFEM and CutFEM.

# Chapter 3

# A concise review of a *posteriori* error estimation

## 3.1 Introduction

Regarding any finite-element-based method, the approximation leads to two types of error, that is *modeling error* and *discretization error*. The previous one aims to measure the distance between the constructed *theoretical model* and the *truth*. Whereas the latter one is dedicated to assess the difference between the *exact* solution of the proposed theoretical model and *computational* solution obtained by numerical model. In our work, we focus on the discretization error so that the accurate solution of mathematical formulation is regarded as the *reality*.

For instance, suppose that we consider a general linear equation, such that

$$\boldsymbol{L}\boldsymbol{u} + \boldsymbol{p} = 0, \tag{3.1.1}$$

where $\boldsymbol{L}$ is a self-adjoint operator. Then the error of displacement is defined by

$$\boldsymbol{e} := \boldsymbol{u} - \boldsymbol{u}^h, \tag{3.1.2}$$

where $\boldsymbol{u}$ is the exact solution, and $\boldsymbol{u}^h$ is the numerical solution within the element size $h$. Then the L2 norm of the error can be defined as

$$\|\boldsymbol{e}\| := \left[\int_\Omega (\boldsymbol{u} - \boldsymbol{u}^h)^T (\boldsymbol{u} - \boldsymbol{u}^h) d\Omega\right]^{\frac{1}{2}}. \tag{3.1.3}$$

Furthermore, for linear elasticity, we have the following relationship, which yields

$$\boldsymbol{\varepsilon} = \mathbb{L}\boldsymbol{u},\ \boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}, \tag{3.1.4}$$

where $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ are strain and stress. $\mathbb{L}$ is the gradient operator and $\mathbf{D}$ is the constant matrix. Thus, the energy norm of error is defined as

$$\|\boldsymbol{e}\|_E = \left[\int_\Omega (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h)^T \mathbf{D} (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h) d\Omega\right]^{\frac{1}{2}} = \left[\int_\Omega (\boldsymbol{\sigma} - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\boldsymbol{\sigma} - \boldsymbol{\sigma}^h) d\Omega\right]^{\frac{1}{2}}. \tag{3.1.5}$$

As we are aware, the simulation relies on the mesh of computational domain so the finer discretization often delivers a better accuracy. Because of the finite computational resources in practice, it is essential to establish a criterion to tell us when to stop the refinement, especially with regard to the complex problem for which the exact solution remains unknown. For this reason, the concept of error estimation was proposed [85].

It is well known that there are *a priori* and *a posteriori* error estimation. A priori error estimate relies on the exact solution so that if the exact solution is unknown in the case, then it is rather an experience-based assessment, which is limited in practical usage. In contrast, a *posteriori* error estimate gives the error information over the whole computation domain contributed by every element. What is more, it can provide the information of accuracy for a specific quantity, which allows users to choose whether to stop refining or not. In the framework of error estimation, we replace the theoretical solution $\boldsymbol{u}$ (or $\nabla\boldsymbol{u}$) by an estimated solution $\tilde{\boldsymbol{u}}$ (or $\nabla\tilde{\boldsymbol{u}}$), and correspondingly a true error is replaced with an error estimation.

In this chapter, recovery-based methodology is introduced. Then the discussion moves to the residual-based error estimate. Afterwards, the goal-oriented error estimation, and the dual weighted residual method to compute this estimator is referred. Finally, these methods of error assessment applied in the framework of IGA are outlined.

## 3.2 Recovery-based methodology

The recovery based error estimates was initiated by Zienkiewicz and Zhu [86]. It derives from a post-processing technique [87], where a smooth higher-oder interpolation based on the "discrete" computational solution is considered to be an improved solution. Then, the error estimation $\tilde{\boldsymbol{e}}$ can be regarded as the difference measurement between this improved solution $\tilde{\boldsymbol{u}}$ and the calculated solution $\boldsymbol{u}^h$, which reads

$$\tilde{\boldsymbol{e}} = \tilde{\boldsymbol{u}} - \boldsymbol{u}^h. \tag{3.2.6}$$

Specially, in sense of FEM, the recovery method often focuses on measuring the gradient approximation (i.e., stress $\tilde{\boldsymbol{\sigma}}$).

In the early work of FEM [88], the recovery process was through the averaging the nodal stress, and then restoring the stress by interpolation within the same basis functions for displacement, i.e.,

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{N}\tilde{\boldsymbol{\sigma}}^* \tag{3.2.7}$$

Even though this approach was improved by considering computing the nodal stress via the extrapolation of Gaussian points [89], or supposing that it could denote the real computational stress in a least square manner [90], it is unable to be proved to be always robust or precise. Since in order to acquire the results, one has to invert a mass-like matrix. If this mass-type matrix is singular or with ill condition number, the results may be unstable or incorrect. As a consequence, it triggered the establishment of an advanced technique, *superconvergent patch recovery* (SPR).

It was found that [86] the stresses converge with a super rate (the same rate as displacement) and own the error order $O(h^{p+1})$, when they are located at some optimal sampling points (often considered as the Gauss-Legendre quadrature points according to the Herrmann theorem [89]) in the element (or patch). Utilizing the sampled stresses ($\bar{\boldsymbol{\sigma}}$) as example data, we establish a surface function $\boldsymbol{p\alpha}$ with respect to the geometric coordinates $\boldsymbol{p} = [1, x, y, \ldots, y^p]$, where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$ is a vector of parameters to be solved and $p$ denotes the order of basis functions.

Fitting the $\boldsymbol{p\alpha}$ with sample data $\bar{\boldsymbol{\sigma}}_i$ in the least-square manner yields

$$f(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \left[\bar{\boldsymbol{\sigma}}_i(x_i, y_i) - \boldsymbol{p}_i\boldsymbol{\alpha}\right]^2 . \tag{3.2.8}$$

Solving the equation $\dfrac{\partial f(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = 0$ can obtain the value of $\boldsymbol{\alpha}$. Then, the nodal superconvergent stress $\tilde{\boldsymbol{\sigma}}^*_{\text{sup}}$ is considered as $\tilde{\boldsymbol{\sigma}}^*_{\text{sup}} = \boldsymbol{p\alpha}$. Furthermore, the recovered superconvergent stress is computed in Eq.(3.2.7), yielding $\tilde{\boldsymbol{\sigma}}_{\text{sup}} = \boldsymbol{N}\tilde{\boldsymbol{\sigma}}^*_{\text{sup}}$.

Thus, the recovery-based error estimation of energy norm can be obtained based on the Eq.(3.1.5) by following

$$\|\tilde{\boldsymbol{e}}\|_E = \left[\int_\Omega (\tilde{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h)^T \mathbf{D}^{-1} (\tilde{\boldsymbol{\sigma}} - \boldsymbol{\sigma}^h) d\Omega\right]^{\frac{1}{2}} . \tag{3.2.9}$$

If we rewrite error norm $\|\tilde{\boldsymbol{e}}\|$ as

$$\|\tilde{\boldsymbol{e}}\| = \left\|(\boldsymbol{u} - \boldsymbol{u}^h) - (\boldsymbol{u} - \tilde{\boldsymbol{u}})\right\| , \tag{3.2.10}$$

then we can acquire the triangle inequality as below [91]

$$\|\boldsymbol{e}\| - \|\hat{\boldsymbol{e}}\| \leqslant \|\tilde{\boldsymbol{e}}\| \leqslant \|\boldsymbol{e}\| + \|\hat{\boldsymbol{e}}\| , \tag{3.2.11}$$

where $\hat{\boldsymbol{e}} := \boldsymbol{u} - \tilde{\boldsymbol{u}}$. Moreover Eq.(3.2.11) arrives at

$$1 - \frac{\|\hat{\boldsymbol{e}}\|}{\|\boldsymbol{e}\|} \leqslant \frac{\|\tilde{\boldsymbol{e}}\|}{\|\boldsymbol{e}\|} \leqslant 1 + \frac{\|\hat{\boldsymbol{e}}\|}{\|\boldsymbol{e}\|}. \tag{3.2.12}$$

It is acknowledged that, using classical FEM, the error norm is gained with $\|\boldsymbol{e}\| = O(h^p)$. Assume that the recovery error offers a higher order, such as, $\|\boldsymbol{e}\| = O(h^{p+\beta})$, then it leads to

$$1 - O(h^\beta) \leqslant \frac{\|\tilde{\boldsymbol{e}}\|}{\|\boldsymbol{e}\|} \leqslant 1 + O(h^\beta). \tag{3.2.13}$$

The Eq.(3.2.13) implies the conclusion [91] that if the recovered error estimator gets a higher convergent rate than true error, namely, $\beta > 0$, the error estimate will converge ultimately to the true error, as $h \to 0$ results in $\dfrac{\|\tilde{\boldsymbol{e}}\|}{\|\boldsymbol{e}\|} \to 1$. This property is unique to recovered-based error estimation.

## 3.3 Residual based error estimation

The method exploiting residuals of the finite element approximation, either explicitly or implicitly, is known as the residual based error estimation. The explicit error estimate makes use of the present residual approximation straightforward, whilst the implicit one employs residual indirectly where it is often obliged to tackle with a linear equation. In this section, the explicit residual error estimation is introduced, as it is convinced that explicit algorithm is normally more computationally economical.

Suppose that an elliptic boundary value problem is expressed by following

$$\begin{aligned} -\triangle u + cu &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega_D, \\ \frac{\partial u}{\partial n} &= g \text{ on } \partial\Omega_N, \end{aligned} \tag{3.3.14}$$

where $u$ is the solution of problem, $c$ is a nonnegative constant, and $f$ is the source term. $\Omega_D$ and $\Omega_N$ are Dirichlet and Neumann boundary respectively with $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$. $n$ is the unit outward normal vector. Then the weak form of the problem can be written as

$$B(u, v) = L(v), \ \forall v \in \mathscr{V}, \tag{3.3.15}$$

where the space $\mathscr{V}$ is defined as

$$\mathscr{V} = \left\{ v \in H^1(\Omega) : v = 0, \text{ on } \partial\Omega_D \right\}, \tag{3.3.16}$$

and

$$B(u, v) = \int_\Omega (\nabla u \cdot \nabla v + cuv) dx, \tag{3.3.17}$$

and

$$L(v) = \int_\Omega f v dx + \int_{\partial\Omega_N} g v ds. \tag{3.3.18}$$

Assumed that $u^h \in \mathscr{V}^h$, $\mathscr{V}^h \subset \mathscr{V}$ is the finite element solution, define the error such as $e := u - u^h$, and then the residual $R(v)$ can be acquired by following

$$R(v) = B(e, v) = B(u, v) - B(u^h, v) = L(v) - B(u^h, v) \tag{3.3.19}$$

The residual integrated over the whole computational domain can be proceeded as the sum of the integration over each element, i.e.,

$$R(v) = \sum_{\Omega_e \in \Omega} \left\{ \int_{\Omega_e} f v dx + \int_{\partial\Omega_e \cap \partial\Omega_N} g v dx - \int_{\Omega_e} (\nabla u^h \cdot \nabla v + c u^h v) dx \right\}. \quad (3.3.20)$$

Adopting integral by parts of each term gives

$$R(v) = \sum_{\Omega_e \in \Omega} \left\{ \int_{\Omega_e} r_i v dx + \int_{\partial\Omega_e \cap \partial\Omega_N} r_b v ds - \int_{\partial\Omega_e \setminus \partial\Omega_N} \frac{\partial u^h}{\partial n_{\Omega_e}} v ds \right\}, \quad (3.3.21)$$

where $r_i$ is the internal residual given by

$$r_i = f + \Delta u^h - c u^h \text{ in } \Omega_e, \quad (3.3.22)$$

and $r_b$ is the boundary residual denoted as

$$r_b = g - \frac{\partial u^h}{\partial n_{\Omega_e}} \text{ on } \Omega_e \cap \partial\Omega_N. \quad (3.3.23)$$

Given $v \in \mathscr{V}$, define $\theta v$ as the approximation to $v$ from the subspace. According to the theorem derive from [85], it yields

$$R(v) = \sum_{\Omega_e \in \Omega} \int_{\Omega_e} r_i (v - \theta v) dx + \sum_{\Gamma \in \partial\Omega} \int_{\Gamma} r_b (v - \theta v) ds, \quad (3.3.24)$$

The Eq.(3.3.24) is crucial as it can be applied to deduce an explicit a *posteriori* error estimate by following. Adopting the Cauchy-Schwarz inequality leads to

$$R(v) \leqslant \sum_{\Omega_e \in \Omega} \|r_i\|_{L2(\Omega_e)} \|v - \theta v\|_{L2(\Omega_e)} + \sum_{\Gamma \in \partial\Omega} \|r_b\|_{L2(\Gamma)} \|v - \theta v\|_{L2(\Gamma)}. \quad (3.3.25)$$

Define $\bar{\Omega}_e$ as the subdomain making up of elements sharing the common boundary with element $\Omega_e$, e.g.,

$$\bar{\Omega}_e := \text{int} \left\{ \bigcup \tilde{\Omega}_e \in \Omega : \tilde{\Omega}_e^* \cap \Omega_e^* \neq \emptyset \right\}. \quad (3.3.26)$$

Based on the theorem proposed by [85], there exists the constant $C$ and parameter $h_{\Omega_e}$ which is the diameter of the element $\Omega_e$, delivering the following inequalities

$$\begin{aligned} \|v - \theta v\|_{L2(\Omega_e)} &\leqslant C h_{\Omega_e} \|v\|_{H^1(\bar{\Omega}_e)}, \\ \|v - \theta v\|_{L2(\partial\Omega_e)} &\leqslant C h_{\Omega_e}^{1/2} \|v\|_{H^1(\bar{\Omega}_e)}. \end{aligned} \quad (3.3.27)$$

Substituting Eq.(3.3.27) into Eq.(3.3.25) and employing the Cauchy-Schwarz inequality provides

$$R(v) \leqslant C \left\| v \right\|_{H^1(\Omega)} \left\{ \sum_{\Omega_e \in \Omega} h_{\Omega_e}^2 \left\| r_i \right\|_{L2(\Omega_e)}^2 + \sum_{\Gamma \in \partial\Omega} h_{\Omega_e} \left\| r_b \right\|_{L2(\Gamma)}^2 \right\}^{1/2}. \tag{3.3.28}$$

Therefore, replacing $v$ by $e$, and reorganizing the terms on the right hand side yields a *posteriori* error estimator $\|e\|^2$, such that

$$\|e\|^2 \leqslant C \sum_{\Omega_e \in \Omega} \left\{ h_{\Omega_e}^2 \left\| r_i \right\|_{L2(\Omega_e)}^2 + h_{\Omega_e} \left\| r_b \right\|_{L2(\partial\Omega_e)}^2 \right\}. \tag{3.3.29}$$

It is an explicit error estimator since all the quantities on the right hand side are able to be computed directly.

## 3.4 Goal-oriented error estimation

The aforementioned error estimation in Eq.(3.2.9) is in form of the energy norm. It is available and quite straightforward to statics, since it is directly associated with the residual (related work can be found in [92, 93]). In context of residual-based methods, they are explicit or implicit. The difference is that an explicit error assessment exploits approximated residuals [94], whereas an implicit error estimation applies residuals obtained solving another small linear algebraic system [95]. The energy-like or residual-type error evaluations generally indicate the field of the error in energy-norm, though, in practice, engineers often concentrate on a certain variable over a specific domain, which is known as the *quantity of interest* (QoI). The technique to measure the error of QoI is referred as the goal oriented error estimate.

The goal-oriented error estimation was initiated to investigate the elliptic problem [96, 97, 98], and then developed to advection-diffusion-reaction equation [99], heat conduction problems [100, 101], coupling problems [102, 103], and the wave propagation topics [43, 104, 105, 106]. For sake of evaluating the error of QoI, one requires to solve a dual (adjoint) problem.

Suppose that we deal with a problem, for $\forall v \in \mathscr{V}$, find $u \in \mathscr{U}$, such that

$$\boldsymbol{\mathcal{A}}(u, v) = \boldsymbol{\mathcal{B}}(v), \tag{3.4.30}$$

where $\boldsymbol{\mathcal{A}}(\cdot,\cdot)$ and $\boldsymbol{\mathcal{B}}(\cdot)$ are linear functions. Given another bounded linear function $\boldsymbol{\mathcal{Q}}(\cdot)$ denoting the QoI, and according to the Riesz representation theorem [107], there exists an unique solution for the dual problem by following

$$z \in \mathscr{U} : \ \boldsymbol{\mathcal{A}}(v,z) = \boldsymbol{\mathcal{Q}}(v), \ v \in \mathscr{V}. \tag{3.4.31}$$

Solving this adjoint problem yields the dual solution z. Assume that $\mathscr{V}^h$ and $\mathscr{U}^h$ represent a finite element space, the error of QoI is defined by

$$u^h \in \mathscr{U}^h : \boldsymbol{\mathcal{Q}}(e) = \boldsymbol{\mathcal{Q}}(u) - \boldsymbol{\mathcal{Q}}(u^h) = \boldsymbol{\mathcal{B}}(v) - \boldsymbol{\mathcal{A}}(u^h, v). \tag{3.4.32}$$

According to the Galerkin orthogonality [27], we obtain

$$\boldsymbol{\mathcal{A}}(u, v^h) - \boldsymbol{\mathcal{A}}(u^h, v^h) = 0, v^h \in \mathscr{V}^h. \tag{3.4.33}$$

Hence, the QoI error in Eq.(3.4.32) can be rewritten as

$$\boldsymbol{\mathcal{Q}}(e) = \boldsymbol{\mathcal{B}}(e, e_z), \tag{3.4.34}$$

where $e_z := z - z^h$. Following the Cauchy-Schwarz inequality, the error bound is determined as

$$\left|\boldsymbol{\mathcal{Q}}(u) - \boldsymbol{\mathcal{Q}}(u^h)\right| \leqslant \|e\| \, \|e_z\| \, . \tag{3.4.35}$$

However this bound is often such disappointed (it is not sharp enough to predict the error) that people make use of the parallelogram identity to derive

$$\boldsymbol{\mathcal{Q}}(e) = \frac{1}{4}\left(\left\|a\boldsymbol{e} + \frac{1}{a}\boldsymbol{e}_z\right\| - \left\|a\boldsymbol{e} - \frac{1}{a}\boldsymbol{e}_z\right\|\right), a \neq 0. \tag{3.4.36}$$

Thus, the subsequent deduction of the error bounds will be simplified to error assessment in form of energy norm. The details can be found in [85].

## 3.5 Error estimate used in IGA framework

The three types of methodology of error estimation referred above has been widely used in IGA scheme. Based on the presence of derivative superconvergent points in

the framework of FEM, Kumar et al. [108] enhanced the gradient of solution in IGA framework using SPR method, and meanwhile allowed the development to satisfy the prescribed consistent criteria. For the sake of driving an adaptive refinement based on PHT/RHT splines, Anitescu et al. [39] exploited an efficient recovery-based error estimator that offers a "recovered solution" with better precision than the calculated solution. Similar work was done by Nguyen-Thanh et al. [40], where they applied recovery-based error estimator for RHT-splines to study elasto-statics and elasto-dynamics. Some residual-based error estimators, which allow a lower and also upper bounds for the Galerkin error to be solved, were taken into account by Feischl et al. [109], to conduct an BEM adaptation within IGA-BEM scheme. Atroshchenko et al. [37] proposed an IGA-extended method, called GIFT, and carried out an adaptive mesh using the residual-type a *posteriori* error estimation. Goal-oriented error estimation was employed to investigate IGA free-boundary issues by Van der Zee and Verhoosel [110]. They pointed out that IGA contributes to continuously differential spline-basis functions , which allows the adjoint problem involving a curvature-dependent boundary condition to be readily tackled with.

## 3.6 Conclusion

In this Chapter, three types of a *posteriori* error estimations, e.g., recovery-based error estimate , residual-type error estimate and goal-oriented error estimate were briefly reviewed. It was proved that the recovery-based error estimation would converge to true error, in the case that, the error measurement between selected recovered solution and exact solution achieves a higher convergent rate than true error. Especially, using the shape function and nodal recovered stress that obtained by fitting sampled superconvergent points to represent superconvergent stress, the SPR method achieves remarkable speed of convergence. A remarkable advantage of the explicit residual-based error estimate is that it can be computed quite efficiently. Because the terms on the right-hand side of error estimator can be directly determined through the numerical solutions. Goal-oriented error estimate is proposed to

evaluate the error of certain scalar variable in a specific area, so that it is generally interpreted as rather a local error estimator than the global type of the energy-norm error estimator. We introduced the approach to compute the error of QoI, which is to solve a dual problem. This is the reason that the strategy is also viewed as dual weighted residual method. According to the discussion above, these error estimate strategies are all being well implemented within the IGA scheme. Particularly, goal-oriented error estimate methodology will be discussed in great detail, as seen in Chapter 6, and furthermore its space-time form will be derived as well.

# Chapter 4

# IGA vs GIFT and PHT/RHT splines

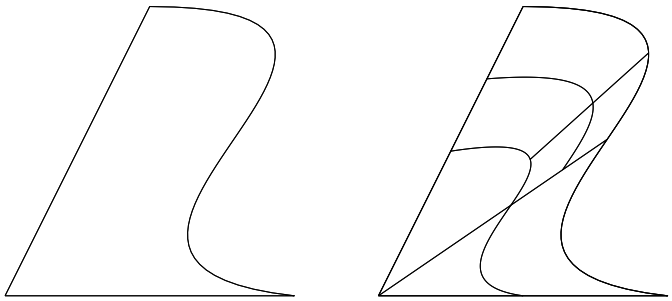
In this chapter, the distinction for framework between IGA and GIFT on the parameterization for geometry and solution space is presented. Then the concept of PHT-splines, including the corresponding local refinement approach, is introduced in detail. Furthermore, the rational type of PHT-splines, namely, RHT-splines are discussed. Also, the process of creation for new control points and weights during $h-$refinement is derived.

## 4.1 The framework of IGA vs GIFT

Suppose that an arbitrary surface and target mesh are presented in Fig.4.1. Let us recall the basic concept of IGA introduced in Chapter 2, which is to represent the geometry and solution domain by the same parametric space and basis functions. However, that is also where the obstacle arises from. For instance, look at the example in Fig.4.1, where we intend to achieve a local refinement on the northeast domain for surface. The core concept of NURBS-based IGA is given by

$$\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}) = \sum N_{\boldsymbol{k}}^{\mathrm{NURBS}}(\boldsymbol{\xi})\boldsymbol{P}_{\boldsymbol{k}},\ \boldsymbol{u}(\boldsymbol{x}) = \sum \boldsymbol{U}_{\boldsymbol{k}} N_{\boldsymbol{k}}^{\mathrm{NURBS}} \circ \boldsymbol{F}^{-1}(\boldsymbol{x}), \tag{4.1.1}$$

where $\boldsymbol{x}$ and $\boldsymbol{\xi}$ indicate the coordinates in physical domain and parametric domain respectively, and $\boldsymbol{F}(\cdot)$ denotes the mapping between these two spaces. $N_{\boldsymbol{k}}^{\mathrm{NURBS}}$ are



(a) Geometry (b) Desired local mesh

Figure 4.1: A surface with arbitrary geometry and targeted local mesh.

the NURBS basis functions and $\boldsymbol{P_k}$ are control points. $\boldsymbol{u}(\boldsymbol{x})$ refers the physical displacement and $\boldsymbol{U_k}$ represents control variables. The notation $\circ$ means the mapping projection. Due to tensor-product-based NURBS, the refinement inevitably extends to the southwest and southeast elements, as shown in Fig.4.2. Although the local refinement barrier can be overcome by PHT-splines, that is,

$$\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}) = \sum N_{\boldsymbol{k}}^{\mathrm{PHT}}(\boldsymbol{\xi})\boldsymbol{P_k},\ \boldsymbol{u}(\boldsymbol{x}) = \sum \boldsymbol{U_k} N_{\boldsymbol{k}}^{\mathrm{PHT}} \circ \boldsymbol{F}^{-1}(\boldsymbol{x}), \tag{4.1.2}$$

PHT-based IGA fails to describe the geometry exactly (see Fig.4.3).

The GIFT method can well address both of problems in IGA-NURBS and IGA-PHT methods. The main idea of GIFT is to use different parametric spaces and basis functions to approximate the geometry and solution respectively. Specifically in this case, we adopt NURBS to catch the surface and PHT-splines to deliver the local refinement for solution field, such as

$$\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}) = \sum N_{\boldsymbol{k}}^{\mathrm{NURBS}}(\boldsymbol{\xi})\boldsymbol{P_k},\ \boldsymbol{u}(\boldsymbol{x}) = \sum \boldsymbol{U_k} N_{\boldsymbol{k}}^{\mathrm{PHT}} \circ \boldsymbol{F}^{-1}(\boldsymbol{x}). \tag{4.1.3}$$

It is clear in Fig.4.4 that the objective local mesh is actualized perfectly by GIFT strategy. Note that since the NURBS element at initial stage is sufficient to depict the surface geometry accurately, owing to the IGA character that both $h-$refinement and $p-$refinement do not change geometry description, it is not required to refine the NURBS parametric domain anymore in subsequent computations, namely, $\boldsymbol{F} = \boldsymbol{F}_0$, which can save computational cost.

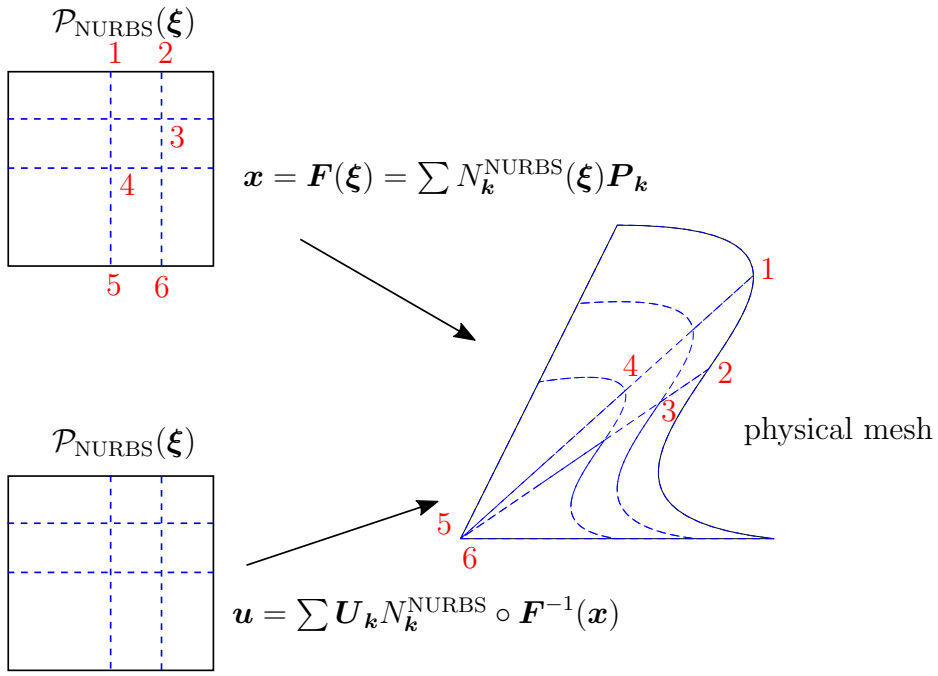



Figure 4.2: The core principle of IGA: the same parameterization and basis functions (NURBS) are applied for geometry representation and solution approximation. The corresponding relationship of representative control points between parametric domain and physical space are noted with red color. Particularly, point 5 and point 6 in parametric space are projected to a overlapping point in physical domain.

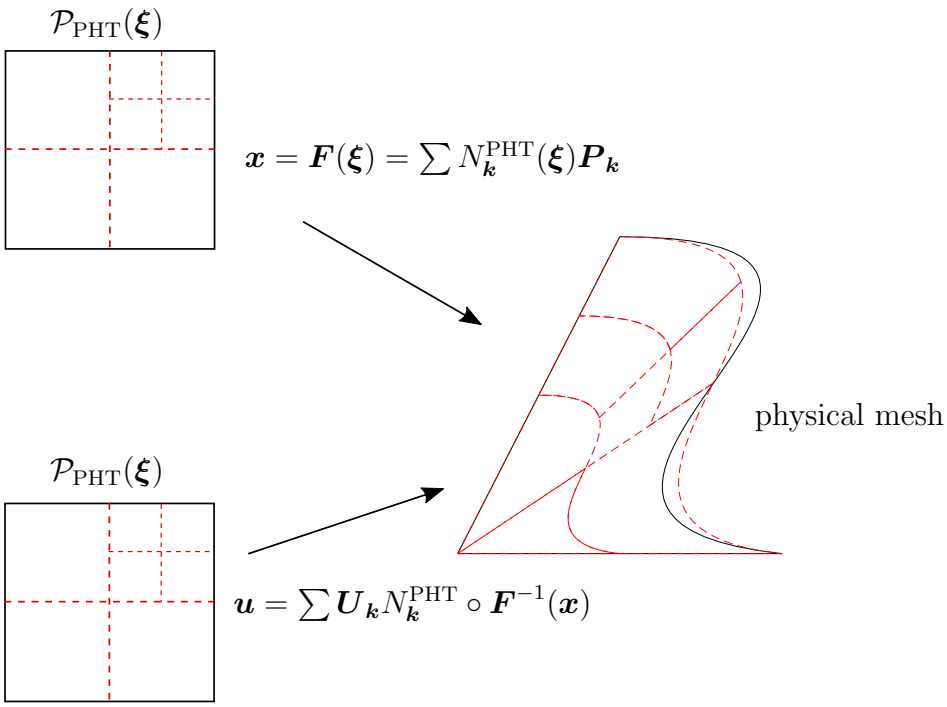



Figure 4.3: The core principle of IGA: the same parameterization and basis functions (PHT) are applied for geometry representation and solution approximation.

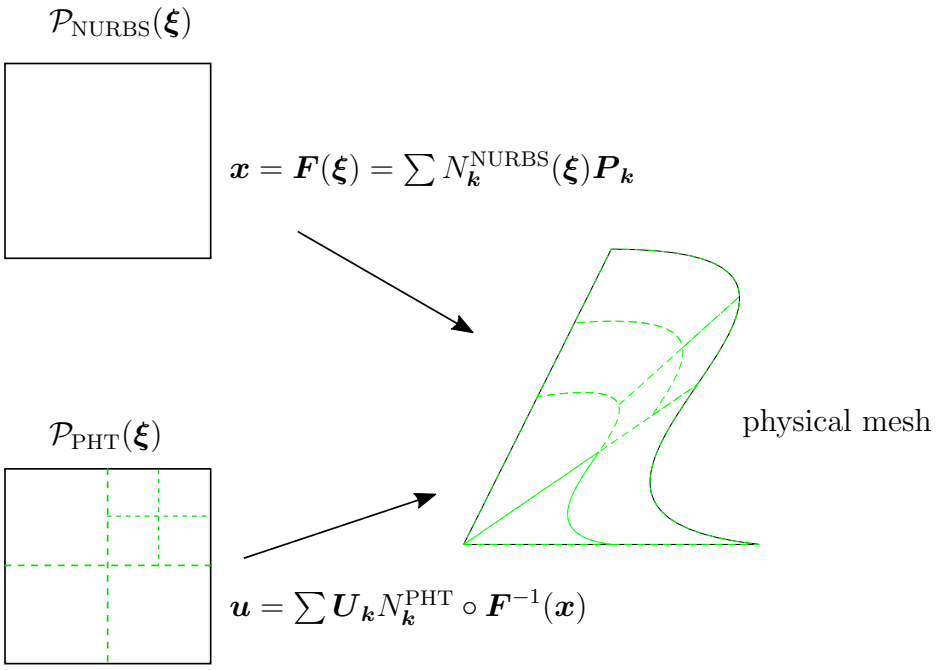



Figure 4.4: The main concept of GIFT: different parameterizations and basis functions are applied for geometry representation (NURBS) and solution approximation (PHT).

In conclusion, the virtues of GIFT can be summarized as two points: 1) it provides the flexibility to select parametric domain and basis functions differently to characterize geometric and solution space, which enable geometric exactness preserved and meanwhile the feature of local refinement is possessed for analysis. 2) Once the precise parameterization of geometry is determined at initial stage, the geometric refinement can be avoidable in the following computations, which is computationally competitive.

The performance of GIFT methodology in dynamics are introduced explicitly in Chapter 5 and Chapter 6. In the following sections, the PHT-splines and the RHT-splines are discussed. Here, it is acknowledged that Section 4.2 and Section 4.3 are based on Appendix B.1 and Appendix B.2 in the following published paper:

**Yu, P.**, Anitescu, C., Tomar, S., Bordas, S. P. A., and Kerfriden, P. (2018). Adaptive Isogeometric analysis for plate vibrations: An efficient approach of local refinement based on hierarchical a posteriori error estimation. Computer Methods in Applied Mechanics and Engineering, 342, 251-286.

## 4.2 PHT-spline basis functions

The PHT-spline with bi-cubic orders was proposed by Deng et al. in [29], and is developed recently to arbitrary degree by Anitescu et al. in [39]. For brevity, only main properties of the bi-cubic PHT-spline basis functions applied in this paper and the refinement process are introduced. The reasons to choose the bi-cubic orders involve as follows. Firstly, the use of PHT-spline basis functions requires the consistent orders of polynomials on each direction of coordinate. Furthermore, the continuity of PHT-splines is computed by $[(p-1)/2]$, where $[\cdot]$ is the integer-valued function and $p$ is the order of PHT-splines. Hence, it can be seen that if the selected order is lower than third, only $C^0$ can be achieved, which is no better than FEM. Although $C^2$ would be reached if fifth-order polynomial is adopted, it may cost too much computational effort. Therefore, the option of bi-cubic PHT-splines is a good

balance.

### 4.2.1 Construction of the PHT-spline basis function

Assume that all cells $\mathcal{T} = \bigcup \mathcal{T}_e$ are described on a hierarchical T-mesh $\mathbb{T}$ on a parameterized space $\mathcal{P}$. Then a linear space for PHT-splines can be defined [29]

$$\mathscr{S}(p, q, \alpha, \beta, \mathbb{T}) = \left\{ T(\xi, \eta) \in C^{\alpha,\beta}(\mathcal{P}) | T(\xi, \eta) \in \mathbb{P}_{p,q}, \forall \mathcal{T}_e \in \mathcal{T} \right\}, \tag{4.2.4}$$

where the space $\mathbb{P}_{p,q}$ makes up of all the bivariate polynomials with order $p, q$, and $C^{\alpha,\beta}(\Omega)$ is the space including all continuous bivariate spline functions with $C^{\alpha}$ on the $\xi$-direction and $C^{\beta}$ on the $\eta$-direction. The dimension equation of spline space $\mathscr{S}(p, q, \alpha, \beta, \mathbb{T})$ with $p \geqslant 2\alpha + 1$ and $q \geqslant 2\beta + 1$ is described in [111]. Especially, the bi-cubic PHT-spline space can be indicated that

$$\dim \mathscr{S}(3, 3, 1, 1, \mathbb{T}) = 4(V^b + V^+). \tag{4.2.5}$$

Here $V^b$ and $V^+$ denotes the quantity of boundary vertices and interior crossing vertices separately. Assumed that the parametric domain $\mathcal{P} = [0, 1] \times [0, 1]$ is given, the tensor-product PHT surface can be defined as

$$\boldsymbol{S}_{\text{PHT}} = \sum_{i=1}^{n} \sum_{j=1}^{m} T_{i,j}^{p,q}(\xi, \eta) \boldsymbol{P}_{i,j}, \quad (\xi, \eta) \in \mathcal{P}, \tag{4.2.6}$$

where $T_{i,j}^{p,q}$ are constructed $C^{\alpha,\beta}$ continuous PHT-spline functions and $\boldsymbol{P}_{i,j}$ are control points. According to the literature [39, 29], the $T_{i,j}^{p,q}$ is generally computed through Bézier representation, which will be introduced subsequently. Let that $\hat{\boldsymbol{F}}$ represents linear mapping from a reference domain $\hat{\mathcal{P}}$ to parametric domain $\mathcal{P}$ that

$$\hat{F} : \hat{\mathcal{P}} \to \mathcal{P}, \quad \hat{\boldsymbol{F}}(\hat{\xi}, \hat{\eta}) = (\xi, \eta), \quad \hat{\mathcal{P}} = [-1, 1] \times [-1, 1], \tag{4.2.7}$$

the basis function $T_{i,j}^{p,q}$ can be rewritten in the form of a linear combination of Bernstein polynomials that

$$T_{i,j}^{p,q}(\xi, \eta) = \sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} b_{\hat{i},\hat{j}} \hat{B}_{\hat{i},\hat{j}} \circ \hat{\boldsymbol{F}}^{-1}(\xi, \eta), \tag{4.2.8}$$

where $\hat{B}_{\hat{i},\hat{j}}(\hat{\xi},\hat{\eta}) = \hat{B}_{\hat{i}}(\hat{\xi})\hat{B}_{\hat{j}}(\hat{\eta})$ are the tensor product of univariate Bernstein functions, which are defined on $\hat{\mathcal{P}}$ as follows

$$\hat{B}_{\hat{i}}(\hat{\xi}) = \frac{1}{2^p}\begin{pmatrix} p \\ i-1 \end{pmatrix}(1-\xi)^{p-i+1}(1+\xi)^{i-1}, \quad i = 1,\ 2, \ldots,\ p+1. \tag{4.2.9}$$

The $b_{\hat{i},\hat{j}}$ are Bézier ordinates obtained by a recursive method called De Casteljaus algorithm. De Casteljaus algorithm is known to calculate the Bernstein polynomials or locate a point on Bézier geometry for a specific parametric value. It is also employed to subdivide a single Bézier curve into two Bézier curves for an arbitrary parameter value. More detail about this algorithm can be found in [39, 30].

### 4.2.2 Tree structure and local refinement

The data structure of 2D hierarchical T-mesh $\mathbb{T}$ for PHT-splines is saved within the quadtree scheme, as described in Fig.4.5. Each leaf or node of the tree denotes one element $\mathcal{T}_e$ at different mesh levels, which stores all the essential information regarding PHT-spline basis functions, i.e., Bézier ordinates, numbering method of nodes and elements, refinement level, etc.. What is more, every leaf or node also reserves hierarchical connections employed to track parent and children elements during the refinement process, and adjacent connections combining neighboring elements with pointers. The procedure of classic vertices insertions during refinements is performed in Fig.4.6. More details with respect to the concept of the methodology and implementation can be found in [112].

Before this tree system is exploited for adaptive refinement, we recall the issue of refinement procedure on PHT mesh affected by the level between the target element and its adjoint elements. As stated in [29], assumed that the level of the element $\mathcal{T}_e$ is $\mathcal{K}$ and maximum level of the neighboring elements is $\mathcal{K}'$. If $\mathcal{K} \geqslant \mathcal{K}' - 1$, the refinement would be straightforward as illustrated in Fig.4.7. Whilst when $\mathcal{K} < \mathcal{K}' - 1$, the situation would be a bit more complex but still in a good control with the application of the quadtree configuration, exhibited in Fig.4.8.

The following simple example is to help readers to better interpret the developed refinement strategy. To be specific, a 2D initial mesh is presented in Fig.4.9(a),

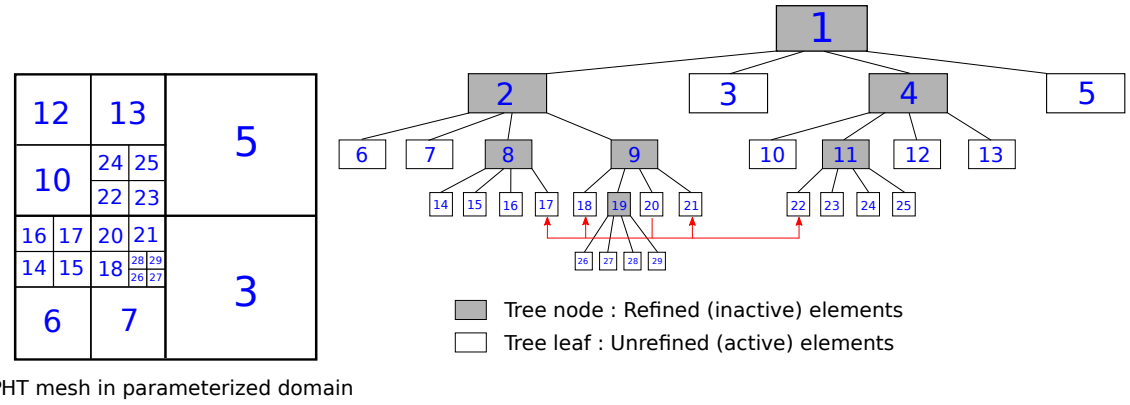
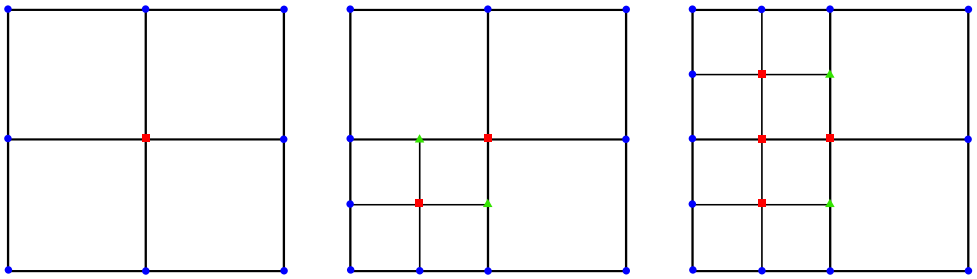



Figure 4.5: A typical quadtree system to represent data structure of PHT adaptive mesh. The numbering of elements is executed from coarse to fine level. The numbers in parametric space (on the left) just denote the elements without any children. With the help of quadtree structure, the relationship of all elements is readily observed. For instance, element 1 is the parent of cell 4, and element 4 is inherited by children cells 10, 11, 12, 13. The adjoint cells of element 20 are elements 17,18,21,22.



(a) (b) (c)

Figure 4.6: An example to present vertices insertion at local refinement on a PHT mesh. Particularly, the blue dots denote boundary vertices, and the green triangles express T-junctions, which are generated when refinements are created between adjacent elements with different levels. It should be noted that T-junctions do not change basis functions until they are transfered to crossing vertices, which are indicated by red squares. A crossing vertex will lead to the truncation over $(\alpha+1)(\beta+1)$ Bézier ordinates around, which are set as zeros firstly, and then reset by new values on the updated spline space. (a) The initial mesh, (b) the local refinement in one cell, (c) the local refinement in the adjoint cell.

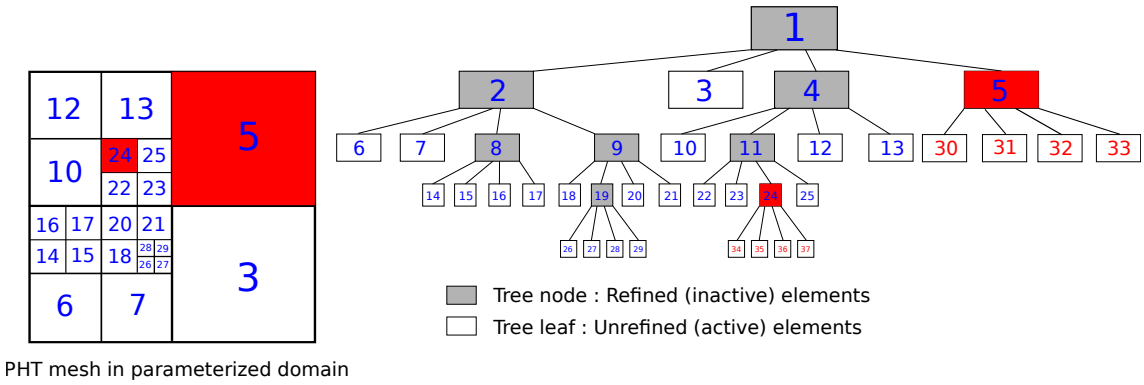



Figure 4.7: The direct refinement for PHT mesh in case of level $\mathcal{K} \geqslant \mathcal{K}'-1$. The cells 5 and 24 are marked and then refined. Note that refinements are always proceeded from coarse level to fine level in implementation regardless of the order of marking. So the numbering of children elements of cell 5 is smaller than those of cell 24.

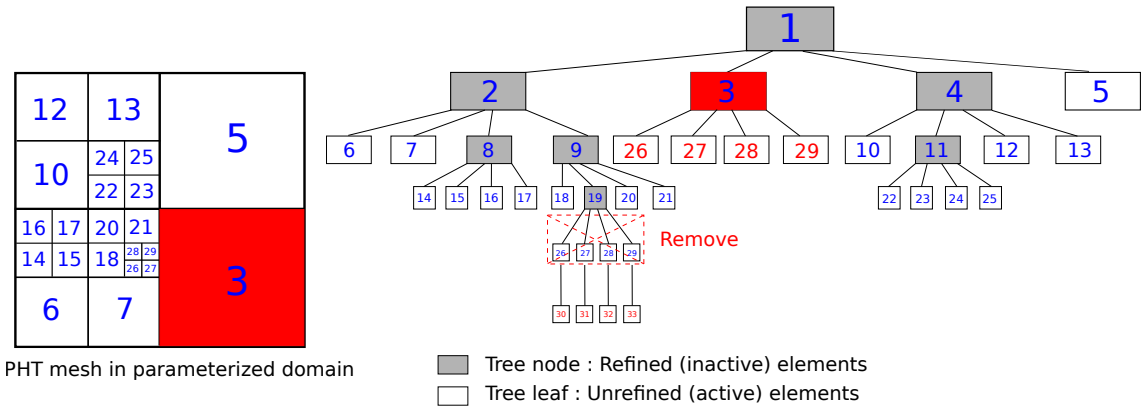



Figure 4.8: The refinement rule for PHT mesh in terms of level $\mathcal{K} < \mathcal{K}'-1$. Assuming that cell 3 is considered to be refined, we have to remove the adjacent elements (26, 27, 28, 29) with level $\mathcal{K}'$ temporarily to ensure the updated maximum level $\mathcal{K}''$ of neighboring elements satisfies that $\mathcal{K} \geqslant \mathcal{K}'' - 1$. Afterwards, refine the cell 3 to obtain the 4 children elements with numbering (26,27,28,29). Ultimately, refine cell 19 again to acquire new children elements (30,31,32,33) as the replacement of removal elements (26,27,28,29) before.

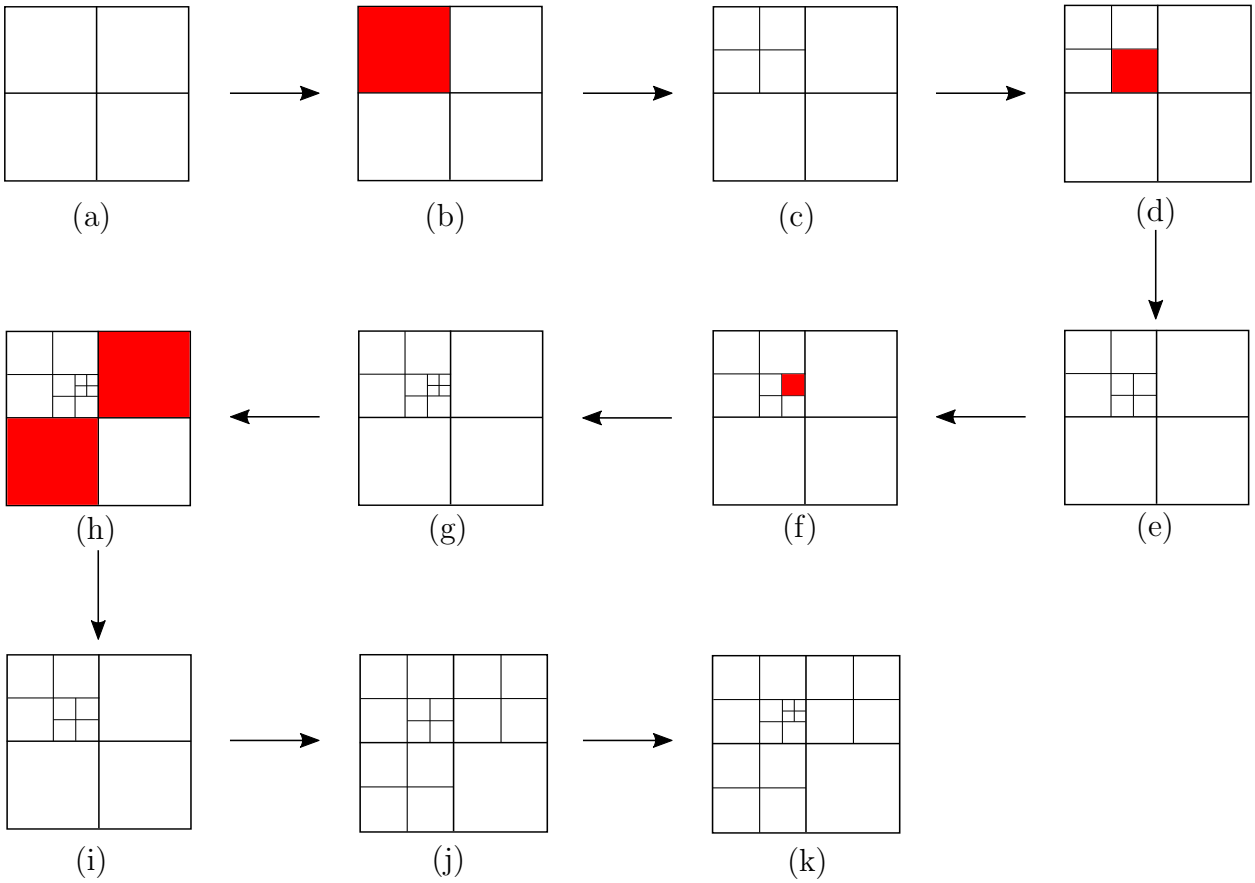



Figure 4.9: An simple 2D example to illustrate the proposed refinement strategy of PHT-splines.

with 4 elements. The element desired to be refined is marked with red colour. It is observed that the discretization is straightforward from Fig.4.9(a) to Fig.4.9(g), which belongs to the case such that $\mathcal{K} \geqslant \mathcal{K}' - 1$. While, when the refinement of elements marked in Fig.4.9(h) is desired to be achieved, which is considered as the case $\mathcal{K} < \mathcal{K}' - 1$, one has to remove the finest level firstly (as shown in Fig.4.9(i)), and afterwards refine the marked elements (as seen in Fig.4.9(j)). Finally, complement the finest refinement removed before, as presented in Fig.4.9(k).

## 4.3 RHT-spline basis functions

The RHT-spline basis functions defined over 2D hierarchical T-mesh $\mathbb{T}$ can be expressed as [30]

$$R_{i,j}^{p,q} = \frac{T_{i,j}^{p,q}(\xi,\eta)\hat{w}_{i,j}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m} T_{\hat{i},\hat{j}}^{p,q}(\xi,\eta)\hat{w}_{\hat{i},\hat{j}}}, \quad (\xi,\eta) \in \mathcal{P} = [0,1]\times[0,1], \tag{4.3.10}$$

where $T_{i,j}^{p,q}(\xi,\eta)$ are PHT-spline basis functions defined as in Section 4.2, and $w_{i,j}$ are weights. Correspondingly, a RHT-spline surface at level $k$ mesh reads

$$\boldsymbol{S}_{\mathrm{RHT}}^{k} = \sum_{\boldsymbol{I}} \boldsymbol{R}_{I}^{k}(\xi,\eta)\boldsymbol{P}_{I}^{k}, \tag{4.3.11}$$

where $\boldsymbol{I}$ is the multi-index, $\boldsymbol{R}_{I}^{k}(\xi,\eta)$ and $\boldsymbol{P}_{I}^{k}$ are RHT-spline basis functions and control points at level $k$ mesh. As discussed above, PHT-splines are adopted within the GIFT method where it is not necessary to calculate control points because the geometry is identified by NURBS. The discretization of geometry will remain at the initial phase in the refinement process. However, when RHT-splines are applied in IGA framework, it is obliged to update the control points, as well as weights, at each refinement step.

### 4.3.1 Update of control points and weights with h-refinement

Following the approach proposed by Deng et al.[29] to get the new control points after refinement for the cubic PHT-spline surface,it aims to develop it for a cubic RHT-spline surface. Assuming that the control points $\boldsymbol{P}_{\boldsymbol{I}}^{k} = (x_{\boldsymbol{I}}, y_{\boldsymbol{I}}, z_{\boldsymbol{I}})$ are depicted over a 3D Euclidean space, we introduce the *homogeneous coordinates* [63] to denote $\boldsymbol{P}_{\boldsymbol{I}}^{k}$ in a four-dimensional space as follows

$$\boldsymbol{P}_{\boldsymbol{I}}^{k} = \mathbb{H}(\boldsymbol{P}_{\boldsymbol{I}}^{k(w)}) = \mathbb{H}[(\hat{w}_{\boldsymbol{I}}x_{\boldsymbol{I}}, \hat{w}_{\boldsymbol{I}}y_{\boldsymbol{I}}, \hat{w}_{\boldsymbol{I}}z_{\boldsymbol{I}}, \hat{w}_{\boldsymbol{I}})] = \begin{cases} (\dfrac{\hat{w}_{\boldsymbol{I}}x_{\boldsymbol{I}}}{\hat{w}_{\boldsymbol{I}}}, \dfrac{\hat{w}_{\boldsymbol{I}}y_{\boldsymbol{I}}}{\hat{w}_{\boldsymbol{I}}}, \dfrac{\hat{w}_{\boldsymbol{I}}z_{\boldsymbol{I}}}{\hat{w}_{\boldsymbol{I}}}), & \text{with } \hat{w}_{\boldsymbol{I}} \neq 0, \\ x_{\boldsymbol{I}}, y_{\boldsymbol{I}}, z_{\boldsymbol{I}}, & \text{with } \hat{w}_{\boldsymbol{I}} = 0, \end{cases} \tag{4.3.12}$$

where $\boldsymbol{P}_{\boldsymbol{I}}^{k(w)}$ are the weighted control points on a 4D space and $\mathbb{H}$ is the mapping function. Also, we can create this analogous relationship for the surface such that

$\boldsymbol{S}^k_{\mathrm{RHT}} = \mathbb{H}(\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}) = \mathbb{H}(W\boldsymbol{S}^k_{\mathrm{PHT}}, W)$ with

$$\boldsymbol{S}^{k(w)}_{\mathrm{PHT}} = \sum_{\boldsymbol{I}} \boldsymbol{T}^k_I(\xi,\eta)\boldsymbol{P}^{k(w)}_I,\ W = \sum_{\boldsymbol{I}} \boldsymbol{T}^k_I(\xi,\eta)\hat{w}^k_I, \tag{4.3.13}$$

where $\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}$ is the weighted PHT-spline surface at level $k$, and $W$ are weights for the surface. Note that the basis functions to represent $\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}$ are PHT-spline basis functions so that we can directly call the algorithm [29] to generate new control points and weights. To be specific, for instance, when a new vertex is inserted into an element belonging to cells $\mathcal{T}_k$ at level $k$, there will be $(\alpha+1)(\beta+1)$ new added basis functions $\boldsymbol{T}^{k+1}_{\boldsymbol{J}}$, weighted control points $\Delta\boldsymbol{P}^{k+1(w)}_{\boldsymbol{J}}$ created at $\mathcal{T}_{k+1}$ such that

$$\boldsymbol{S}^{k+1(w)}_{\mathrm{PHT}}(\xi,\eta) = \sum_{\boldsymbol{I}}^{N} \tilde{\boldsymbol{T}}^{k+1}_{\boldsymbol{I}}(\xi,\eta)\underline{\underline{\boldsymbol{P}}}^{k+1(w)}_{\boldsymbol{I}} + \sum_{\boldsymbol{J}=N+1}^{(\alpha+1)(\beta+1)} \boldsymbol{T}^{k+1}_{\boldsymbol{J}}(\xi,\eta)\Delta\boldsymbol{P}^{k+1(w)}_{\boldsymbol{J}}, \tag{4.3.14}$$

where $\tilde{\boldsymbol{T}}^{k+1}_I$ are the basis functions $\boldsymbol{T}^k_I$ defined on the $\mathcal{T}_{k+1}$ and consequently the relevant control points are remained as $\underline{\underline{\boldsymbol{P}}}^{k+1(w)}_{\boldsymbol{I}} = \boldsymbol{P}^{k(w)}_I$. Now only the parametric space (by setting it as $(\xi^*,\eta^*)$), dominated by the new basis, is considered. Due to the truncation character [29, 39], the basis functions $\tilde{\boldsymbol{T}}^{k+1}_I$ and the derivatives will eliminate in this domain. Meanwhile, Owing to the geometry preservation during the refinement, it yields

$$\boldsymbol{S}^{k+1(w)}_{\mathrm{PHT}}(\xi^*,\eta^*) = \sum_{\boldsymbol{J}=N+1}^{(\alpha+1)(\beta+1)} \boldsymbol{T}^{k+1}_{\boldsymbol{J}}(\xi^*,\eta^*)\Delta\boldsymbol{P}^{k+1(w)}_{\boldsymbol{J}} = \boldsymbol{S}^{k(w)}_{\mathrm{PHT}}(\xi^*,\eta^*). \tag{4.3.15}$$

For the sake of computation $\Delta\boldsymbol{P}^{k+1(w)}_{\boldsymbol{J}}$, a linear operator including the *geometric information* for the surface $\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}(\xi^*,\eta^*)$ is defined by

$$\mathscr{G}\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}(\xi^*,\eta^*) = \left(\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}(\xi^*,\eta^*),\ \frac{\partial\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}}{\partial\xi^*},\ \frac{\partial\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}}{\partial\eta^*},\ \frac{\partial^2\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}}{\partial\xi^*\partial\eta^*}\right). \tag{4.3.16}$$

After that rewriting Eq.(4.3.15) leads to

$$\mathscr{G}\boldsymbol{S}^{k(w)}_{\mathrm{PHT}}(\xi^*,\eta^*) = \sum_{\boldsymbol{J}} \mathscr{G}\boldsymbol{T}^{k+1}_{\boldsymbol{J}}(\xi^*,\eta^*)\Delta\boldsymbol{P}^{k+1(w)}_{\boldsymbol{J}} = \underline{\underline{\mathbb{T}}}\cdot\Delta\boldsymbol{P}^{k+1(w)}. \tag{4.3.17}$$

Here, in terms of cubic basis functions, the matrix $\underline{\underline{\mathbb{T}}}$ can be simply identified through the distance measured between the inserted vertex and the adjacent elements by

following

$$\underline{\underline{\mathbb{T}}}(\Delta u_1, \Delta u_2, \Delta v_1, \Delta v_2) = \begin{pmatrix} (1-\bar{\lambda})(1-\mu) & \bar{\lambda}(1-\mu) & (1-\bar{\lambda})\mu & \bar{\lambda}\mu \\ -\alpha(1-\mu) & \alpha(1-\mu) & -\alpha\mu & \alpha\mu \\ -\beta(1-\bar{\lambda}) & -\beta\bar{\lambda} & \beta(1-\bar{\lambda}) & \beta\bar{\lambda} \\ \alpha\beta & -\alpha\beta & -\alpha\beta & \alpha\beta \end{pmatrix}, \tag{4.3.18}$$

where $\alpha = \dfrac{1}{\Delta u_1 + \Delta u_2},\ \beta = \dfrac{1}{\Delta v_1 + \Delta v_2},\ \bar{\lambda} = \alpha\Delta u_1,\ \mu = \beta\Delta v_1$ can be seen in [29]. Hence, $\Delta \boldsymbol{P}_{\boldsymbol{J}}^{k+1(w)}$ are acquired by

$$\Delta \boldsymbol{P}^{k+1(w)} = (\underline{\underline{\mathbb{T}}})^{-1} \cdot \mathscr{G} \boldsymbol{S}_{\mathrm{PHT}}^{k(w)}, \tag{4.3.19}$$

and furthermore the new weighted control points are computed by

$$\boldsymbol{P}^{k+1(w)} = \underline{\underline{\boldsymbol{P}}}_{\boldsymbol{I}}^{k+1(w)} + \Delta \boldsymbol{P}^{k+1(w)}. \tag{4.3.20}$$

As a consequence, the new control points at level $k+1$ mesh are renewed as

$$\boldsymbol{P}^{k+1(w)} = \mathbb{H}(\boldsymbol{P}^{k+1(w)}). \tag{4.3.21}$$

It is worth noting that since PHT-splines are generated by B-splines on the hierarchical T-mesh, RHT-splines at level 0 mesh are certainly NURBS. As a result, without any refinement at initial phase, a RHT surface is consistent to a NURBS surface, which is, $\boldsymbol{S}_{\mathrm{RHT}}^0 = \boldsymbol{S}_{\mathrm{NURBS}}^0$, with $\boldsymbol{R}_{\boldsymbol{I}}^0 = \boldsymbol{N}_{\boldsymbol{I}}, \boldsymbol{P}_{\boldsymbol{I}}^0 = \boldsymbol{P}_{\boldsymbol{I}}, \hat{w}_{\boldsymbol{I}}^0 = w_{\boldsymbol{I}}$. After that, following the derivation from Eq.(4.3.12) to Eq.(4.3.21), the new control points and weights for RHT-spline basis functions can be gained during the hierarchical refinement.

## 4.4 Conclusion

In this chapter, the frameworks of IGA and GIFT were introduced. Compared with IGA scheme, GIFT method owns an better flexibility that different basis functions can be exploited to parameterize the geometric and solution domains respectively. This methodology is prospective since it provides the availability that the geometric representation can be seamlessly extracted from existing state-of-the-art CAD

technology, which is the core concept of the IGA, and simultaneously it allows the solution field to be approximated within basis functions more suitable to analysis. Hence, by overcoming the weakness of IGA, GIFT is intuitively a superior technique to bridge CAD and CAE.

In addition, the main concept of PHT-splines were outlined. The local hierarchical refinement feature is proceeded within a quadtree structure, where the parent and children elements can be readily traced. Based on this structure, the refinement strategy was developed in the case that level difference between adjacent elements is more than 1, and was implemented successfully which will demonstrated by numerical examples in Section 5.6. This improvement makes the local refinement approach of PHT-splines more robust.

Moreover, RHT-splines were introduced. The algorithm for the generation of new control points and weights during the $h-$refinement were deduced. Although RHT-splines are rational so that it is able to replace PHT-splines to precisely depict geometry for some instances (e.g. circle), it should be pointed out that the capability of geometric description of RHT-splines is weaker than NURBS, owing to the lower order continuity. For example, cubic RHT-splines only hold $C^1$ continuity, while cubic NURBS can reach $C^2$ continuity.

Therefore, it is preferred to harness the newly developed GIFT method, wherein NURBS is used for geometry design and PHT-splines are applied for analysis, to study the structural dynamics subsequently. The detail contents can be found in Chapter 5 (for vibration) and Chapter 6 (for wave propagation).

# Chapter 5

# Adaptivity for vibration

**Statement**: It is acknowledged that this chapter is based on the following published paper. Hence there are some similarities. However, the chapter has been rewritten by the author of thesis. The theoretical and technical contents in the thesis are more explicit and clear than those in the paper.

**Yu, P.**, Anitescu, C., Tomar, S., Bordas, S. P. A., and Kerfriden, P. (2018). Adaptive Isogeometric analysis for plate vibrations: An efficient approach of local refinement based on hierarchical a posteriori error estimation. Computer Methods in Applied Mechanics and Engineering, 342, 251-286.

## 5.1 Introduction

Based on the discussion of IGA in Chapter 4, due to the high continuity order of NURBS basic functions, NURBS-based IGA has been successfully used to investigate plate vibrations, including Kirchoff plate [24, 113] and ReissnerMindlin plate [114, 115]). The results obtained when using IGA are often more accurate than those obtained using the traditional finite element method (FEM) [21]. The previously mentioned studies of plate vibrations with IGA are mostly dedicated to homogeneous structures, whereby the vibrations occur globally so that the uniform refinement of NURBS is an adequate method to control the discretization error. However, when

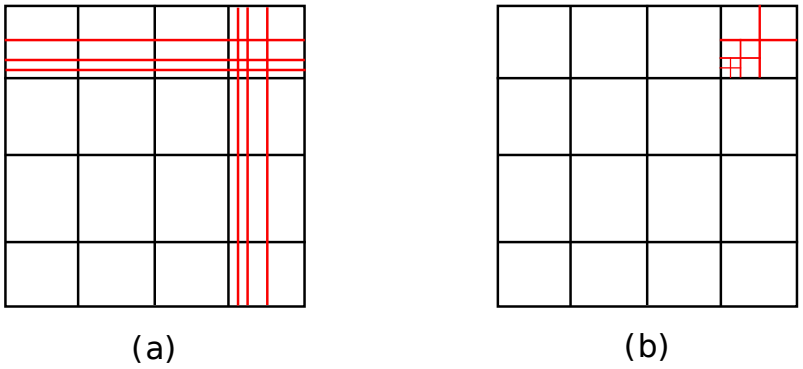



Figure 5.1: (a) NURBS global refinement and (b) PHT splines local refinement

the dynamic solution exhibits local features, due to e.g. sharp geometrical feature and/or varying material properties, the uniform NURBS-based refinement may become inefficient. This is because NURBS basis functions are defined by a tensor product form, which leads to globally structured grid (see Fig.5.1(a)), which in turns result in computational wastage when trying to capture the local features of interest.

To overcome these limitations, splines with local refinement properties such as (truncated) hierarchical B-splines [116, 117], hierarchical NURBS [112], locally refined (LR) B-splines [33], T-splines [118, 119], and polynomial/rational splines over hierarchical T-meshes (PHT/RHT)-splines [29, 30] were developed. In this study, PHT-splines are applied, as they inherit the main merits of both B-splines and T-splines: basis functions can be represented by Bézier-Bernstein polynomials over a set of Hermite finite elements, and mesh refinement is local and simple (as seen in Fig.5.1(b)). In the recent past, PHT-splines have been successfully used to solve static elastic solid problems. The numerical results of [30] showed that the adaptive PHT refinement delivers a higher convergence rate than uniform NURBS refinement. However, since PHT-splines are polynomial splines and not rational splines, they are not able to exactly represent the basic geometrical features, e.g. circles, ellipses, that typically arise in engineering design and analysis. This problem may be circumvented by making use of RHT-splines, as proposed in [30]. However, RHT-splines, unlike NURBS and T-splines, cannot be seamlessly extracted from existing

CAD softwares. Besides, in the context of adaptivity, the updating of the weights during the refinement process require dedicated numerical developments. These difficulties prompted the author to look into another direction.

Inspired by the work proposed in [36, 38, 39, 37], the Geometry-Independent Field approximaTion (GIFT) will be employed to deal with the aforementioned issue, by allowing the geometry and solution fields to be described using different functional spaces. The GIFT framework was first developed within the context of the boundary element method [36]. Later, Toshniwal et al. [38] established a scheme for unstructured quadrilateral meshes, where the space of geometric design $\mathcal{S}_{\mathbb{D}}$ and the space of solution analysis $\mathcal{S}_{\mathbb{A}}$ were different. The GIFT model, wherein NURBS basis functions are used to describe the geometry without approximation and PHT splines are utilized for analysis, will be used in this chapter. This combination is compatible with state-of-the-art CAD technology (the net of control points is inherited from CAD directly), whilst allowing local mesh refinement to take place in a non-degenerate manner. It is worth noticing that the GIFT scheme may not satisfy the isogeometric compatibility condition [38], which requires the solution space to be adequately rich compared to the functional space used to represent the geometry. This is because PHT splines are polynomials while NURBS are rational functions. However, the NURBS/PHT combo has been successfully used to develop (adaptive) GIFT schemes and achieve a higher convergent rates in the context of linear elasticity, compared to other methods in the investigations [37, 39]. It should also be noted that with increasing refinement level of the solution space, the space of PHT-splines will get closer to encompassing the NURBS-based functional space used to represent the geometry.

The main contributions of this chapter are twofold. Firstly, develop a novel methodology of local adaptivity based on GIFT for structural vibration problems. Secondly, propose a novel frequency-domain adaptation strategy based on a posteriori error estimation and mode sweeping. Closely related to the proposed adaptivity scheme is that described in [40], whereby RHT splines are used to obtain a higher convergence rate of free vibration frequencies when compared to that ob-

served when using tensor-product-based NURBS. However, the local refinement of the aforementioned study is driven by a priori error estimation, which does not provide any quantitative measure of accuracy (i.e. it only provides a spatial map of error sources). It is aimed to develop a comprehensive refinement strategy, which includes a reliable stopping criterion as provided by a posteriori error estimation (see for instance [41, 42, 43, 44]). More precisely, a hierarchical a posteriori error estimator that makes the best of the PHT-spline local refinement capabilities will be defined. More precisely, the accuracy of GIFT solutions will be estimated by computing refined solutions using a finer mesh generated by systematically subdividing every GIFT element. The mesh adaptivity will be performed for every free vibration mode (i.e. frequency and associated mode shape) within a frequency band, sweeping from lower to higher frequencies. The algorithm requires for coarse and fine GIFT estimations of the modes to be put in correspondence in order to be compared. This is not a trivial task. It is intended to propose a new algorithm inspired by the Modal Assurance Criterion (MAC) strategy, which is widely used in experimental structural dynamics [45, 46]. It will be shown that the proposed algorithm is robust with respect to the order of multiplicity of the fine and coarse modes.

The organization of this chapter is as follows. In Section 5.2 and Section 5.3, formulate the variational form of the free vibrations of Reissner-Mindlin plates is formulated, in the frequency domain. Then describe how to discretise such problems using IGA and GIFT. In Section 5.4, the proposed error estimation strategy is described, in the context of the adaptation of one, clearly isolated, free vibration mode. The strategy is then extended to the accuracy control of multiple modes in Section 5.5, which includes a technical discussion regarding the control of discretisation errors in the context of free vibration modes of order of multiplicity larger than one. In Section 5.6, several numerical examples are presented to evaluate the efficiency of the proposed methodology, and conclusions are drawn in Section 5.7.

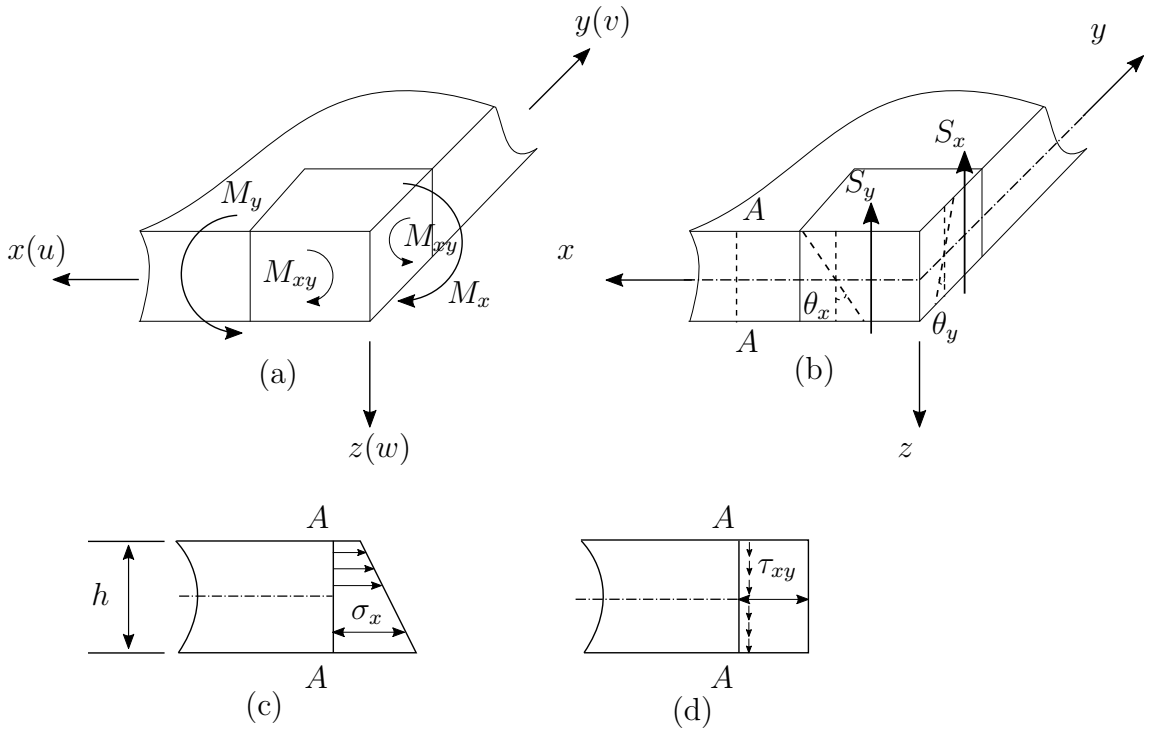



Figure 5.2: Illustration of variables for general plate theory. (a) Displacement and moment, (b) resultant force and rotation, (c) axial stress (d) shear stress

## 5.2 Problem Statement

### 5.2.1 General pate theory

The plate problem is actually a special-type three-dimensional solid problem. The difference to the conventional 3D elasticity is that the thickness is quite smaller than other dimensions so that it may result in the serious numerical ill-conditioning problems. To deal with this occasion, the thin plate theory was firstly formulated by Kirchhoff [120]. After that, a weaker constrain of the postulate for thin plate was established by Reissner [121] and then modified slightly by Mindlin [122]. These modifications are the foundations for thick plate theory. The plate theory is built on two hypotheses. The first one is that sections normal to the middle plane remain plane during deformation. The second postulate is that stress and strain at $z$ direction can be ignored. Based on these two assumptions and as observed in Fig.5.2, the

displacements $u, v$ and deflection $w$ at each direction can be written by following

$$u = -z\theta_x(x,y),\ v = -z\theta_y(x,y),\ w = w(x,y), \tag{5.2.1}$$

where $\theta_x$ and $\theta_y$ are rotations. Subsequently, the strain tensor $\boldsymbol{\varepsilon}$ is obtained such that

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = -z \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix} = -z\mathbb{L}\boldsymbol{\theta}, \tag{5.2.2}$$

where $\mathbb{L}$ is an operator. In terms of isotropic elasticity, all the moment components can be written as the expression as follows

$$\boldsymbol{M} = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \mathbf{D}\mathbb{L}\boldsymbol{\theta}, \tag{5.2.3}$$

where $M_x, M_y$ are bending moments and $M_{xy}$ is twisting moment (as shown in Fig.5.2(a)), with expressions as below

$$M_x = -\int_{-h/2}^{h/2} \sigma_x z dz,\ M_y = -\int_{-h/2}^{h/2} \sigma_y z dz,\ M_{xy} = -\int_{-h/2}^{h/2} \tau_{xy} dz, \tag{5.2.4}$$

where $\sigma_x$ and $\sigma_y$ are axial stress, and $\tau_{xy}$ are shear stress. $h$ is the thickness of the plate. $\mathbf{D}$ is a elastic matrix and defined by the assumption of plane stress, i.e.,

$$\mathbf{D} = D \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{(1-\nu)}{2} \end{bmatrix},$$

where the parameter $D = \dfrac{Eh^3}{12(1-\nu^2)}$ denotes the bending stiffness of the plate, and $E, \nu$ express the Young's modulus, the Poisson's ratio, respectively. In what follows, the shear force resultants $\boldsymbol{S}$ are obtained by

$$\boldsymbol{S} = \begin{bmatrix} S_x \\ S_y \end{bmatrix} = \kappa G h(\boldsymbol{\nabla} w - \boldsymbol{\theta}), \tag{5.2.5}$$

where

$$S_x = \int_{-h/2}^{h/2} \tau_{xz} dz,\ S_y = \int_{-h/2}^{h/2} \tau_{yz} dz. \tag{5.2.6}$$

where the shear elastic modulus is $G$, $\boldsymbol{\nabla}$ is the gradient operator. $\kappa$ is the shear correction factor that derives from the shear force-shear strain constitutive model. It is introduced here, since it leads to the accuracy loss by assuming that the shear stress is distributed uniformly across the thickness of the plate. The value of $\kappa$ (varying from 0.76 to 0.91) is approximately linearly related to the Poisson's ratio $\nu$ (ranging from 0 to 0.5). More details of shear correction factor $\kappa$ can be found in [122], where it also provides $\kappa = 5/6$ for the Mindlin plate. Thus, the "out-plane" governing equation of plate is obtained based on the equilibrium equation, e.g.

$$\begin{aligned} \boldsymbol{\nabla}^T \boldsymbol{S} + q = 0, \\ \mathbb{L}^T \boldsymbol{M} + \boldsymbol{S} = 0 \end{aligned} \tag{5.2.7}$$

Where $q$ is the transverse loading.

With respect to Kirchhoff plate (thin plate), it gives the relationship below

$$\boldsymbol{\nabla} w - \boldsymbol{\theta} = 0. \tag{5.2.8}$$

Furthermore, the Eq.(5.2.2) can be rewritten as

$$\boldsymbol{\varepsilon} = -z \mathbb{L} \boldsymbol{\nabla} w \tag{5.2.9}$$

where the operator denotes $\mathbb{L}\boldsymbol{\nabla} = \left[ \frac{\partial^2}{\partial x^2},\ \frac{\partial^2}{\partial y^2},\ 2\frac{\partial^2}{\partial x \partial y} \right]^T$. Then for simplicity, one can vanish $\boldsymbol{S}, \boldsymbol{M}$ and $\boldsymbol{\theta}$ and only leave $w$ as the variable, based on the Eq.(5.2.3), and Eq.(5.2.7), and then it yields a scalar equation, such that

$$(\mathbb{L}\boldsymbol{\nabla})^T \mathbf{D} \mathbb{L} \boldsymbol{\nabla} w - q = 0. \tag{5.2.10}$$

For isotropy where the bending stiffness $D$ is constant, it turns to well-known bi-harmonic equation

$$D \left( \frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} \right) - q = 0. \tag{5.2.11}$$

Unlike the thin plate, the variables $w, \theta_x, \theta_y$ are independent in case of the Mindlin plate (thick plate). Hence, based on the discussion above, equilibrium of thick plate reads

$$\begin{aligned} \mathbb{L}^T \mathbf{D} \mathbb{L} \boldsymbol{\theta} + \boldsymbol{S} &= 0, \\ \boldsymbol{\nabla}^T \boldsymbol{S} + q &= 0, \\ \boldsymbol{S} + \kappa G h(\boldsymbol{\theta} - \boldsymbol{\nabla} w). & \end{aligned} \tag{5.2.12}$$

It is clear that, for Kirchhoff plate theory, the mechanical behaviour can be characterized within only one variable $w$, which leads to an irreducible formulation and saving of computational cost. However, the $C_1$ continuity of element is mandatory, and otherwise it will leads to an "kink". In contrast, in thick plate theory, more degree of freedom is demanded since the quantities $w, \theta_x, \theta_y$ are independent. Nevertheless, $C_0$ continuity is sufficient for computation, which is reason that thick plate is widely used in FEM. In this thesis, since the implemented algorithm for multi-patch coupling is only available for $C^0$ continuity, the theory of Reissner-Mindlin plate is adopted.

### 5.2.2 Weak form for vibration of Reissner-Mindlin plate

Based on the static equilibrium obtained in Eq.(5.2.12) for thick plate. In this section, the PDEs for elastodynamics is introduced and weak form of vibration is derived.

Let $\Omega \subset \mathbb{R}^2$ represent $x - y$ domain of the middle plane of a typical Mindlin plate, as shown in Fig.5.3. The boundary $\partial\Omega$ involves $\partial\Omega_u$, $\partial\Omega_s$ and $\partial\Omega_m$ such that: $\partial\Omega = \overline{\partial\Omega_u \cup \partial\Omega_s \cup \partial\Omega_m}$, $\partial\Omega_u \bigcap \partial\Omega_s = \emptyset$, $\partial\Omega_u \bigcap \partial\Omega_m = \emptyset$. Labuschagne et al. [123] studied a plate-beam model where the ReissnerMindlin plate and Timoshenko beam are coupled. The solutions of eigenvalue and eigenvector were compared with KirchhoffEuler Bernoulli system. Furthermore, different strategies used to tackle with situation at contact interface between plate and beams were introduced. Based the that research, the formal statement of governing equation of motion for Mindlin

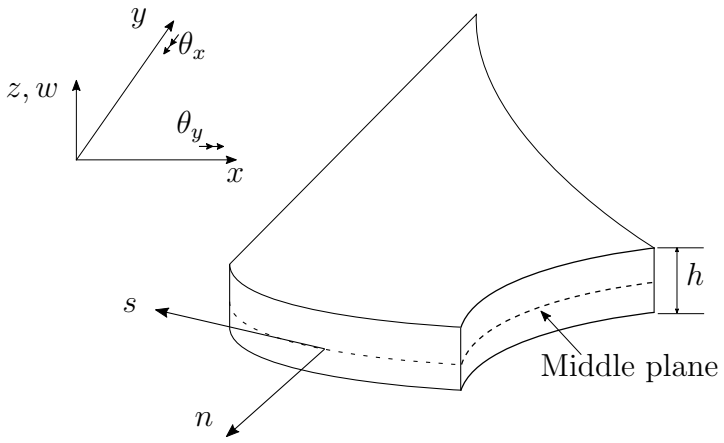



Figure 5.3: Geometry and coordinate system of a classical Reissner-Mindlin plate.

plate can be expressed as [7]

$$\left.\begin{aligned} -\frac{\rho h^3}{12}\ddot{\boldsymbol{\theta}} + \mathbb{L}^T\boldsymbol{M} + \boldsymbol{S} = 0 \\ -\rho\ddot{w}h + \boldsymbol{\nabla}^T\boldsymbol{S} + q = 0 \end{aligned}\right\} \text{ in } \Omega, \tag{5.2.13}$$

$$\left.\begin{aligned} w = \bar{w} \\ \boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \end{aligned}\right\} \text{ on } \partial\Omega_u, \tag{5.2.14}$$

$$\boldsymbol{S} = \bar{\boldsymbol{S}} \text{ on } \partial\Omega_s, \tag{5.2.15}$$

$$\boldsymbol{M} = \bar{\boldsymbol{M}} \text{ on } \partial\Omega_m. \tag{5.2.16}$$

The $\rho$ is density and $h$ is thickness. The directions of deflection $w$ and rotation $\boldsymbol{\theta} = (\theta_x, \theta_y)^T$ are presented in Fig.5.3. The $q$ is transverse loading, and the operator $\mathbb{L}$ is defined as

$$\mathbb{L} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}.$$

According to static equilibrium obtained in Eq.(5.2.12), Eq.(5.2.13) can be rewritten

as

$$\begin{aligned} -\frac{\rho h^3}{12}\ddot{\boldsymbol{\theta}} + \mathbb{L}^T\mathbf{D}\mathbb{L}\boldsymbol{\theta} + \kappa G h(\boldsymbol{\nabla} w - \boldsymbol{\theta}) = 0, \\ -\rho\ddot{w}h + \boldsymbol{\nabla}^T[\kappa G h(\boldsymbol{\nabla} w - \boldsymbol{\theta})] + q = 0. \end{aligned} \tag{5.2.17}$$

The $\ddot{()} := \dfrac{d^2}{dt^2}()$ denotes the second derivative with respect to time $t$. $\ddot{w}$ is a scalar, followed by $\boldsymbol{\nabla}^T\boldsymbol{\nabla} w$ and $\boldsymbol{\nabla}^T\boldsymbol{\theta}$ as scalars as well. Now introduce the trial solution space $\mathscr{U}$ and test function space $\mathscr{V}$

$$\mathscr{U} = \{u \in H^2(\Omega) : u = \bar{u} \text{ on } \partial\Omega_u\}, \tag{5.2.18}$$

$$\mathscr{V} = \{v \in H^2(\Omega) : v = 0 \text{ on } \partial\Omega_u\}, \tag{5.2.19}$$

and for all $[\delta w, \delta\boldsymbol{\theta}]^T \in \mathscr{V}$, $[w, \boldsymbol{\theta}]^T \in \mathscr{U}$, we can have the variational form of Eq.(5.2.17)

$$\begin{aligned} -\int_\Omega \delta\boldsymbol{\theta}^T\frac{\rho h^3}{12}\ddot{\boldsymbol{\theta}}d\Omega + \int_\Omega \delta\boldsymbol{\theta}^T\mathbb{L}^T\mathbf{D}\mathbb{L}\boldsymbol{\theta}d\Omega + \int_\Omega \delta\boldsymbol{\theta}^T\kappa G h(\boldsymbol{\nabla} w - \boldsymbol{\theta})d\Omega = 0, \\ -\int_\Omega \delta w^T\rho\ddot{w}h d\Omega + \int_\Omega \delta w^T\boldsymbol{\nabla}^T[\kappa G h(\boldsymbol{\nabla} w - \boldsymbol{\theta})]d\Omega + \int_\Omega \delta w^T q d\Omega = 0. \end{aligned} \tag{5.2.20}$$

We integrate by parts two terms in Eq.(5.2.20) such that

$$\begin{aligned} \int_\Omega \delta\boldsymbol{\theta}^T\mathbb{L}^T\mathbf{D}\mathbb{L}\boldsymbol{\theta}d\Omega = -\int_\Omega (\mathbb{L}\delta\boldsymbol{\theta})^T\mathbf{D}\mathbb{L}\boldsymbol{\theta}d\Omega + \int_{\partial\Omega_m} \delta\boldsymbol{\theta}^T\bar{\boldsymbol{M}}d\Omega, \\ \int_\Omega \delta w^T\boldsymbol{\nabla}^T[\kappa G h(\boldsymbol{\nabla} w - \boldsymbol{\theta})]d\Omega = -\int_\Omega (\boldsymbol{\nabla}\delta w)^T\kappa G h\boldsymbol{\nabla} w d\Omega + \int_\Omega (\boldsymbol{\nabla}\delta w)^T\kappa G h\boldsymbol{\theta}d\Omega + \int_{\partial\Omega_s} \delta w^T\bar{\boldsymbol{S}}d\Omega, \end{aligned} \tag{5.2.21}$$

where $\bar{\boldsymbol{M}}$ and $\bar{\boldsymbol{S}}$ denote the prescribed moment and shear force respectively. Since only the free vibration analysis is considered in this chapter, $\bar{\boldsymbol{M}}, \bar{\boldsymbol{S}}, q$ are all set to be zero. Substitution of Eq.(5.2.21) into Eq.(5.2.20) gives the weak form of the governing equation as

$$\begin{aligned} \int_\Omega \delta\boldsymbol{\theta}^T\frac{\rho h^3}{12}\ddot{\boldsymbol{\theta}}d\Omega + \int_\Omega (\mathbb{L}\delta\boldsymbol{\theta})^T\mathbf{D}\mathbb{L}\boldsymbol{\theta}d\Omega + \int_\Omega \delta\boldsymbol{\theta}^T\kappa G h\boldsymbol{\theta}d\Omega - \int_\Omega \delta\boldsymbol{\theta}^T\kappa G h\boldsymbol{\nabla} w d\Omega = 0, \\ \int_\Omega \delta w^T\rho\ddot{w}h d\Omega + \int_\Omega (\boldsymbol{\nabla}\delta w)^T\kappa G h\boldsymbol{\nabla} w d\Omega - \int_\Omega (\boldsymbol{\nabla}\delta w)^T\kappa G h\boldsymbol{\theta}d\Omega = 0. \end{aligned} \tag{5.2.22}$$

The general time-dependent solution of the free vibration equation can be constructed by assuming

$$\boldsymbol{u}(x,t) = \boldsymbol{\phi}(x)\exp(i\lambda t) = \begin{bmatrix} w \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_w \\ \boldsymbol{\phi}_{\boldsymbol{\theta}} \end{bmatrix} \exp(i\lambda t), \tag{5.2.23}$$

where $\lambda$ is the frequency and $\boldsymbol{\phi}$ is the eigenvector. Substituting Eq.(5.2.23) in Eq.(5.2.22), the weak form becomes an eigenvalue problem as follows

$$\begin{aligned} -\lambda^2 \int_\Omega \delta\boldsymbol{\phi}_{\boldsymbol{\theta}}^T \frac{\rho h^3}{12} \boldsymbol{\phi}_{\boldsymbol{\theta}} d\Omega + \int_\Omega (\mathbb{L}\delta\boldsymbol{\phi}_{\boldsymbol{\theta}})^T \mathbf{D} \mathbb{L} \boldsymbol{\phi}_{\boldsymbol{\theta}} d\Omega + \int_\Omega \delta\boldsymbol{\phi}_{\boldsymbol{\theta}}^T \kappa G h \boldsymbol{\phi}_{\boldsymbol{\theta}} d\Omega - \int_\Omega \delta\boldsymbol{\phi}_{\boldsymbol{\theta}}^T \kappa G h \boldsymbol{\nabla}\boldsymbol{\phi}_w d\Omega = 0, \\ -\lambda^2 \int_\Omega \delta\boldsymbol{\phi}_w^T \rho \boldsymbol{\phi}_w h d\Omega + \int_\Omega (\boldsymbol{\nabla}\delta\boldsymbol{\phi}_w)^T \kappa G h \boldsymbol{\nabla}\boldsymbol{\phi}_w d\Omega - \int_\Omega (\boldsymbol{\nabla}\delta\boldsymbol{\phi}_w)^T \kappa G h \boldsymbol{\phi}_{\boldsymbol{\theta}} d\Omega = 0. \end{aligned} \tag{5.2.24}$$

## 5.3 Discretization of the eigenvalue problem using IGA and GIFT

Let $\mathcal{P}$ be the parametric domain, and the physical domain $\Omega$ is parametrized on $\mathcal{P}$ by a geometrical mapping $\boldsymbol{F}$

$$\boldsymbol{F} : \mathcal{P} \to \Omega, \quad \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}). \tag{5.3.25}$$

We assume that the domain $\Omega$ consists of sub-domains $\Omega_i$, such that $\Omega = \bigcup_{i=1}^N \Omega_i$. The geometrical map $\boldsymbol{F}$ is given by a set of basis functions $N_{\boldsymbol{k}}(\boldsymbol{\xi})$ and a set of control points $\boldsymbol{P}_{\boldsymbol{k}}$ as

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{\boldsymbol{k}\in\boldsymbol{I}} N_{\boldsymbol{k}}(\boldsymbol{\xi}) \boldsymbol{P}_{\boldsymbol{k}}, \tag{5.3.26}$$

where $N_{\boldsymbol{k}}(\boldsymbol{\xi})$ is a bivariate tensor product of the univariate basis functions $\boldsymbol{N}_i^p(\xi), \boldsymbol{N}_j^q(\eta)$ with the orders $p, q$, such that $N_{\boldsymbol{k}}(\boldsymbol{\xi}) = \boldsymbol{N}_i^p(\xi)\boldsymbol{N}_j^q(\eta), (\xi,\eta) \in \boldsymbol{\xi}$. Moreover, $\boldsymbol{I}$ is introduced as 2-dimensional multi-index $(i,j)$, and $\boldsymbol{k}$ is interchangeably regarded as the collapsed notation for $\boldsymbol{I}$. For example, $\boldsymbol{I} = 1, 2, 3, \ldots$ denotes $(i,j) = (1,1), (1,2), (1,3), \ldots$. In what follows, we will refer to the set $\{N_{\boldsymbol{k}}(\boldsymbol{\xi})\}_{\boldsymbol{k}\in\boldsymbol{I}}$ as the *geometry basis*, and introduce the discretization of the eigenvalue problem via the scheme of IGA and GIFT respectively.

### 5.3.1 The framework of IGA

In IGA, the solution field $\boldsymbol{\phi} = [\boldsymbol{\phi}_w, \boldsymbol{\phi}_{\boldsymbol{\theta}}]^T$ is represented through the same spline functions which are used for the geometry, i.e.,

$$\boldsymbol{\phi} = \sum_{\boldsymbol{k} \in \boldsymbol{I}} N_{\boldsymbol{k}}(\boldsymbol{\xi}) \bar{\boldsymbol{\phi}}_{\boldsymbol{k}}, \tag{5.3.27}$$

where $\bar{\boldsymbol{\phi}}_{\boldsymbol{k}}$ are unknown control variables. Then the deflection and rotations which serve as components of the solution variable can be denoted in the matrix form

$$\begin{bmatrix} \boldsymbol{\phi}_w \\ \boldsymbol{\phi}_{\boldsymbol{\theta}_x} \\ \boldsymbol{\phi}_{\boldsymbol{\theta}_y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{N}_w & 0 & 0 \\ 0 & \boldsymbol{N}_{\theta_x} & 0 \\ 0 & 0 & \boldsymbol{N}_{\theta_y} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\phi}}_w \\ \bar{\boldsymbol{\phi}}_{\theta_x} \\ \bar{\boldsymbol{\phi}}_{\theta_y} \end{bmatrix}, \tag{5.3.28}$$

where $[\boldsymbol{\phi}_w, \boldsymbol{\phi}_{\boldsymbol{\theta}_x}, \boldsymbol{\phi}_{\boldsymbol{\theta}_y}]^T$ is a $3 \times 1$ vector and $[\bar{\boldsymbol{\phi}}_w, \bar{\boldsymbol{\phi}}_{\theta_x}, \bar{\boldsymbol{\phi}}_{\theta_y}]^T$ is a $3n \times 1$ vector, where $n$ is the number of control points. Supposed that the test functions $\delta\boldsymbol{\phi}_w$ and $\delta\boldsymbol{\phi}_{\boldsymbol{\theta}}$ are discretized using Eq.(5.3.28), the discrete form of Eq.(5.2.24) will become

$$\begin{aligned} \delta\bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}}^T \left( -\lambda^2 \int_\Omega \boldsymbol{N}_{\boldsymbol{\theta}}^T \frac{\rho h^3}{12} \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega \right) \bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}} + \delta\bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}}^T \left( \int_\Omega (\mathbb{L}\boldsymbol{N}_{\boldsymbol{\theta}})^T \mathbf{D} \mathbb{L} \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega + \int_\Omega \boldsymbol{N}_{\boldsymbol{\theta}}^T \kappa G h \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega \right) \bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}} \\ - \delta\bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}}^T \left( \int_\Omega \boldsymbol{N}_{\boldsymbol{\theta}}^T \kappa G h \boldsymbol{\nabla} \boldsymbol{N}_w d\Omega \right) \bar{\boldsymbol{\phi}}_w = 0, \\ \delta\bar{\boldsymbol{\phi}}_w^T \left( -\lambda^2 \int_\Omega \boldsymbol{N}_w^T \rho h \boldsymbol{N}_w d\Omega \right) \bar{\boldsymbol{\phi}}_w - \delta\bar{\boldsymbol{\phi}}_w^T \left( \int_\Omega (\boldsymbol{\nabla} \boldsymbol{N}_w)^T \kappa G h \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega \right) \bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}} \\ + \delta\bar{\boldsymbol{\phi}}_w^T \left( \int_\Omega (\boldsymbol{\nabla} \boldsymbol{N}_w)^T \kappa G h \boldsymbol{\nabla} \boldsymbol{N}_w d\Omega \right) \bar{\boldsymbol{\phi}}_w = 0. \end{aligned} \tag{5.3.29}$$

Suppose that the number of control points is $n$, the dimension of both $\bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}_{\boldsymbol{\theta}}$ is $2n$ . While the dimension of $\bar{\boldsymbol{\phi}}_w$ and $\delta\bar{\boldsymbol{\phi}}_w$ is $n$. The terms $\int_\Omega \boldsymbol{N}_{\boldsymbol{\theta}}^T \frac{\rho h^3}{12} \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega$ (unit: $kg$) with $2n \times 2n$ dimension and $\int_\Omega \boldsymbol{N}_w^T \rho h \boldsymbol{N}_w d\Omega$ (unit: $kg$) with $n \times n$ dimension are the inner product of shape functions of deflection $w$ and rotation $\boldsymbol{\theta}$ respectively, which contribute to the establishment of mass matrix. The matrix -$\int_\Omega \boldsymbol{N}_{\boldsymbol{\theta}}^T \kappa G h \boldsymbol{\nabla} \boldsymbol{N}_w d\Omega$ (dimension: $2n \times n$, unit: $N/m$), and -$\int_\Omega (\boldsymbol{\nabla} \boldsymbol{N}_w)^T \kappa G h \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega$ (dimension: $n \times 2n$, unit: $N/m$) are the crossing products of basis functions and their gradients. Combining with $\int_\Omega \boldsymbol{N}_{\boldsymbol{\theta}}^T \kappa G h \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega$ (dimension: $2n \times 2n$, unit: $N/m$), they construct shear stiffness matrix. The integrals $\int_\Omega (\mathbb{L}\boldsymbol{N}_{\boldsymbol{\theta}})^T \mathbf{D} \mathbb{L} \boldsymbol{N}_{\boldsymbol{\theta}} d\Omega$ (size: $2n \times$

$2n$, unit: $N/m$) and $\int_\Omega (\boldsymbol{\nabla} \boldsymbol{N}_w)^T \kappa G h \boldsymbol{\nabla} \boldsymbol{N}_w d\Omega$ (size: $n \times n$, unit: $N/m$) belong to ingredients of bending stiffness matrix. The shear stiffness matrix and bending stiffness matrix constitute stiffness matrix of structure.

The Jacobian matrix $J(\boldsymbol{\xi})$ of the mapping $\boldsymbol{F}$ (defined in Eq.(4.1.1)) denotes the projection from parametric space to physical domain, is introduced as

$$\mathbf{J}(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} = \sum_{\boldsymbol{k} \in \boldsymbol{I}} \boldsymbol{P}_{\boldsymbol{k}} \frac{\partial N_{\boldsymbol{k}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}. \tag{5.3.30}$$

We can rewrite Eq.(5.3.29) by defining the stiffness $\mathbf{K}$ and mass matrix $\mathbf{M}$ integrated in the parametric space $\mathcal{P}$ as follows:

$$\begin{aligned}
&\mathbf{K} = \mathbf{K}_b + \mathbf{K}_s, \\
&\mathbf{K}_b = \int_{\mathcal{P}} (\mathbb{B}_b \boldsymbol{N})^T \mathbf{D} \mathbb{B}_b \boldsymbol{N} \left|\mathbf{J}(\boldsymbol{\xi})\right| d\mathcal{P}, \quad \text{bending stiffness} \\
&\mathbf{K}_s = \int_{\mathcal{P}} (\mathbb{B}_s \boldsymbol{N})^T \mathbf{D} \mathbb{B}_s \boldsymbol{N} \left|\mathbf{J}(\boldsymbol{\xi})\right| d\mathcal{P}, \quad \text{shear stiffness} \\
&\mathbf{M} = \int_{\mathcal{P}} \rho \boldsymbol{N}^T \mathbf{m} \boldsymbol{N} \left|\mathbf{J}(\boldsymbol{\xi})\right| d\mathcal{P},
\end{aligned} \tag{5.3.31}$$

with

$$\mathbb{B}_b = \begin{bmatrix} 0 & \dfrac{\partial}{\partial x} & 0 \\ 0 & 0 & \dfrac{\partial}{\partial y} \\ 0 & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}, \mathbb{B}_s = \begin{bmatrix} \dfrac{\partial}{\partial x} & -1 & 0 \\ \dfrac{\partial}{\partial y} & 0 & -1 \end{bmatrix}, \mathbf{m} = \begin{bmatrix} h & 0 & 0 \\ 0 & \dfrac{h^3}{12} & 0 \\ 0 & 0 & \dfrac{h^3}{12} \end{bmatrix}, \tag{5.3.32}$$

where the operator $\mathbb{B}_b$ is related to the bending terms such as $\partial \boldsymbol{N}_{\theta_x}/\partial x$, $\partial \boldsymbol{N}_{\theta_y}/\partial y$, $\partial \boldsymbol{N}_{\theta_x}/\partial y$, $\partial \boldsymbol{N}_{\theta_y}/\partial x$, while operator $\mathbb{B}_s$ is associated to shear terms, i.e., $\partial \boldsymbol{N}_w/\partial x$, $\partial \boldsymbol{N}_w/\partial y$, $-\boldsymbol{N}_{\theta_x}$, $-\boldsymbol{N}_{\theta_y}$. Matrix $\mathbf{m}$ denotes the integral of geometric coefficients correlated to moment and shear force. To be specific, recall the Eq.(5.2.4) and Eq.(5.2.6), the terms $h$ and $\dfrac{h^3}{12}$ in matrix $m$ are obtained through

$$h = \int_{-h/2}^{h/2} dz, \ \frac{h^3}{12} = \int_{-h/2}^{h/2} z^2 dz. \tag{5.3.33}$$

Then Eq.(5.3.29) can be compactly written into the final matrix form of eigenvalue problem by IGA scheme

$$\delta\bar{\boldsymbol{\phi}}^T(\mathbf{K}-\lambda^2\mathbf{M})\bar{\boldsymbol{\phi}}=0. \tag{5.3.34}$$

### 5.3.2 The framework of GIFT

A detailed exposition of GIFT is presented in [37]. GIFT allows to choose a *solution basis* $\{\Psi_{\boldsymbol{k}}(\boldsymbol{\xi})\}_{\boldsymbol{k}\in\boldsymbol{J}}$ that can be different from the geometry basis $\{N_{\boldsymbol{k}}(\boldsymbol{\xi})\}_{\boldsymbol{k}\in\boldsymbol{I}}$, but is defined on the physical domain with the help of the same mapping $\boldsymbol{F}^{-1}$ as in Eq.(5.3.25), i.e.,

$$\Psi_{\boldsymbol{k}}(\boldsymbol{\xi})=\Psi_{\boldsymbol{k}}\circ\boldsymbol{F}^{-1}(\boldsymbol{x}). \tag{5.3.35}$$

Hence, the solution variables $\boldsymbol{\phi}=[\boldsymbol{\phi}_w,\boldsymbol{\phi}_{\boldsymbol{\theta}}]^T$ are described accordingly by

$$\boldsymbol{\phi}=\sum_{\boldsymbol{k}\in\boldsymbol{J}}\Psi_{\boldsymbol{k}}(\boldsymbol{\xi})\bar{\boldsymbol{\phi}}_{\boldsymbol{k}}, \tag{5.3.36}$$

Following the steps of derivation to obtain the statement of matrix form Eq.(5.3.34) from the weak form Eq.(5.2.24) using IGA method, the discrete eigenvalue equation can be similarly acquired through GIFT as in Eq.(5.3.34) with notations

$$\begin{aligned}
\mathbf{K}&=\mathbf{K}_b+\mathbf{K}_s,\\
\mathbf{K}_b&=\int_{\mathcal{P}}(\mathbb{B}_b\boldsymbol{\Psi})^T\mathbf{D}\mathbb{B}_b\boldsymbol{\Psi}\,|\mathbf{J}(\boldsymbol{\xi})|\,d\mathcal{P},\\
\mathbf{K}_s&=\int_{\mathcal{P}}(\mathbb{B}_s\boldsymbol{\Psi})^T\mathbf{D}\mathbb{B}_s\boldsymbol{\Psi}\,|\mathbf{J}(\boldsymbol{\xi})|\,d\mathcal{P},\\
\mathbf{M}&=\int_{\mathcal{P}}\rho\boldsymbol{\Psi}^T\mathbf{m}\boldsymbol{\Psi}\,|\mathbf{J}(\boldsymbol{\xi})|\,d\mathcal{P}.
\end{aligned} \tag{5.3.37}$$

**Remark 1** *No matter what shape functions are employed to assemble the matrix for IGA (Eq.*(5.3.31)*) and GIFT (Eq.*(5.3.37)*), there is no difference for mapping function* $\boldsymbol{F}$ *in terms of these two methods. As a consequence, assumed that the initial geometric shape functions* $N_{\boldsymbol{k}}(\boldsymbol{\xi})$ *are good enough to precisely represent the geometry mapping* $\boldsymbol{F}_0$*, there is no need to refine the geometric space. It denotes that the computation for new geometric basis functions and control are not mandatory any more, namely,* $\boldsymbol{F}=\boldsymbol{F}_0$*, which is computationally advantageous. This property*

*is based on the principle that both h−refinement and p−refinement in IGA are with the geometry preservation.*

In this work, three kinds of spline basis functions, NURBS, PHT and RHT are applied for IGA and GIFT methods.

### 5.3.3 Boundary conditions and multiple patch coupling

Two types of Dirichlet boundary conditions for free vibration are applied in this study

$$\begin{aligned} \bar{w} = 0,\ \bar{\theta}_s = 0,\ \bar{\theta}_n = 0 \quad & \text{clampled}, \\ \bar{w} = 0 \quad & \text{simply supported}, \end{aligned} \tag{5.3.38}$$

where the subscripts $n$ and $s$ denote normal and tangent direction, respectively, as shown in Fig.5.3. Spline basis functions are generally not with interpolatory nature. This does not allow the imposition of Dirichlet boundary condition as directly as in conventional FEM. Some strategies proposed for the mesh free method, e.g., [124, 125] can be extended to the isogeometric framework, but we desire to seek a more simple method. As presented in Fig.5.4, at knot values corresponding to the boundaries of the B-splines, that is $\xi = 0$ and $\xi = 1$, only one B-spline basis function is equal to 1, while others are zero. So it is just required to find the control variables related to the non-zeros basis functions, and remove the relevant degrees of freedom. Thus, the boundary conditions in Eq.(5.3.38) can be imposed. As the NURBS, PHT-splines and RHT-splines are all based on the B-splines, and refinements do not change the situations that only one B-spline basis function is equal to 1, while others are zero on the boundaries, imposing the boundary conditions will be direct. Regarding the coupling of multiple patches, we utilize a convenient and robust approach that patches are conforming at interfaces, which allows the $C^0$ continuity. We choose the discretization of each pair of patches such that the resulting spaces are compatible at the interfaces. To be specific, if the elements of patch A at interface are with the higher refinement level, as shown in Fig.5.5(a), we will refine the corresponding element in Patch B at interface to achieve the same refinement

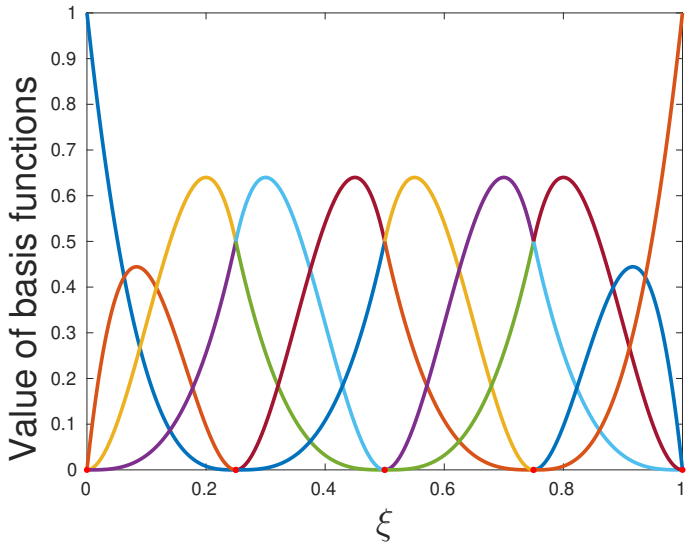



Figure 5.4: 1D cubic B-spline basis functions defined in knot vector $\Xi = [0, 0.25, 0.5, 0.75, 1]$. Compared to the Fig.2.1, where the 1D quadratic B-spline basis functions are presented, there are 1 more shape function in each knot vector interval. In addition, this set of cubic basis functions has been modified to be linear independent. The evidence is that all the shape functions are non-zero in only one knot interval.

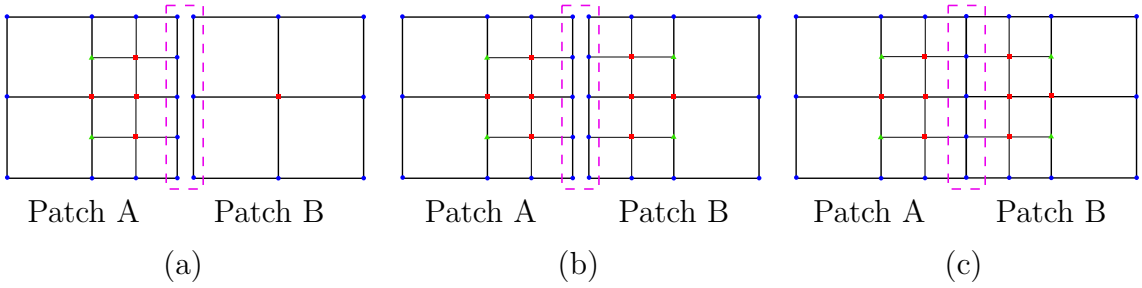



Figure 5.5: The illustration of the approach to couple two patches.

level (see Fig.5.5(b)). Subsequently, we couple the patches by eliminating duplicated degrees of freedom, as presented in Fig.5.5(c). More advanced coupling approaches, for instead using and strong multipatch $C^1$-coupling, can be found in [66] and [126]. Owing to the property of local refinement possessed by PHT, it is simple to realize that this process is with minimal additional refinements and computing resources. Specifically, when we need to refine the elements due to the patches coupling, as illustrated in Fig.5.5, the PHT-splines are helpful since the refinement will only be carried out at the boundary interface, without affecting other elements in the interior.

## 5.4 Adaptivity for one mode

As mentioned in Remark 1, in the GIFT scheme, the refinement is only required for solution space. Therefore, in this section, it is going to present the procedure of PHT mesh adaptivity when a particular mode is targeted.

### 5.4.1 Error estimator

In Chapter 3, the several error estimation methods introduced. Based on these approaches, a *posteriori* hierarchical error estimation is employed in this chapter. The details are as follows. Supposed that $i$ is the mode of interest on *current (coarse) mesh* $\mathbb{T}$ ($\mathbb{T}$ is a hierarchical T-mesh), we define a corresponding mode $\tilde{i}$ on *refined mesh* $\tilde{\mathbb{T}}$, wherein the elements are created by dividing each element in $\mathbb{T}$ into $2^{d \cdot L_e}$ elements, where $d$ is the dimension of the problem, and $L_e$ is the level of refinement.

Assuming that $\boldsymbol{\phi}_i^h$ and $\lambda_i^h$ denote the eigenvector and frequency obtained using $\mathbb{T}$ for mode $\tilde{i}$, and $\tilde{\boldsymbol{\phi}}_{\tilde{i}}$ and $\tilde{\lambda}_{\tilde{i}}$ indicate solutions acquired on $\tilde{\mathbb{T}}$ for mode $\tilde{i}$, then the error estimators for frequency and mode shape can be defined as

$$\left|e_i^{\lambda}\right| = \left|\log\tilde{\lambda}_{\tilde{i}} - \log\lambda_i\right|, \;\; \delta_i^{\phi} = \frac{\left\|e_i^{\phi}\right\|_E}{\left\|\tilde{\boldsymbol{\phi}}_{\tilde{i}}\right\|_E} = \frac{\left\|\tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h\right\|_E}{\left\|\tilde{\boldsymbol{\phi}}_{\tilde{i}}\right\|_E}, \tag{5.4.39}$$

where the computation of energy norm denotes as

$$\|\cdot\|_E := \left[\int_{\Omega}(\cdot)^T\mathbb{B}_b^T\mathbf{D}\mathbb{B}_b(\cdot)d\Omega + \int_{\Omega}(\cdot)^T\mathbb{B}_s^T(\cdot)\mathbf{D}\mathbb{B}_s(\cdot)d\Omega\right]^{\frac{1}{2}}. \tag{5.4.40}$$

Since $\boldsymbol{\phi}_i^h$ and $\tilde{\boldsymbol{\phi}}_{\tilde{i}}$ are discretized by

$$\boldsymbol{\phi}_i^h = \boldsymbol{T}\bar{\boldsymbol{\phi}}_i^h, \;\; \tilde{\boldsymbol{\phi}}_{\tilde{i}} = \tilde{\boldsymbol{T}}\bar{\tilde{\boldsymbol{\phi}}}_{\tilde{i}},$$

where $\boldsymbol{T}$ and $\tilde{\boldsymbol{T}}$ are PHT-spline basis functions defined over spaces $\mathbb{T}$ and $\tilde{\mathbb{T}}$, as defined in Section 4.2. In order to compute $\left\|\tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h\right\|_E$, the control variables $\bar{\boldsymbol{\phi}}_i^h$ should be prolongated onto the $\tilde{\mathbb{T}}$

$$\bar{\boldsymbol{\phi}}_i^h \longrightarrow \mathbb{P}\bar{\boldsymbol{\phi}}_i^h \tag{5.4.41}$$

where $\mathbb{P}$ is the prolongation operator. Two strategies are introduced to compute this prolongation in Appendix A. One is based on the insertion of control points in PHT refinement, and the other one is based upon the projection from coarse mesh to refined mesh. Owing to the features of the isogeometric system that refinement does not change the field, the solution $\boldsymbol{\phi}_i^h$ is preserved exactly after the prolongation, which reads

$$\boldsymbol{\phi}_i^h = \boldsymbol{T}\bar{\boldsymbol{\phi}}_i^h = \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}_i^h. \tag{5.4.42}$$

Hence, define the error $e_i^{\phi} := \tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h$, and according to Eq.(5.4.40), the energy norm of error yields

$$\begin{aligned}\left\|e_i^{\phi}\right\|_E &= \left[\int_{\Omega}(\tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h)^T\mathbb{B}_b^T\mathbf{D}\mathbb{B}_b(\tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h)d\Omega + \int_{\Omega}(\tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h)^T\mathbb{B}_s^T\mathbf{D}\mathbb{B}_s(\tilde{\boldsymbol{\phi}}_{\tilde{i}} - \boldsymbol{\phi}_i^h)d\Omega\right]^{\frac{1}{2}} \\ &= \left[\int_{\Omega}(\bar{\tilde{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P}\bar{\boldsymbol{\phi}}_i^h)^T\tilde{\boldsymbol{T}}^T\mathbb{B}_b^T\mathbf{D}\mathbb{B}_b\tilde{\boldsymbol{T}}(\bar{\tilde{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P}\bar{\boldsymbol{\phi}}_i^h)d\Omega + \int_{\Omega}(\bar{\tilde{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P}\bar{\boldsymbol{\phi}}_i^h)^T\tilde{\boldsymbol{T}}^T\mathbb{B}_s^T\mathbf{D}\mathbb{B}_s\tilde{\boldsymbol{T}}(\bar{\tilde{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P}\bar{\boldsymbol{\phi}}_i^h)d\Omega\right]^{\frac{1}{2}}.\end{aligned} \tag{5.4.43}$$

If the stiffness matrix $\tilde{\mathbf{K}}$ is defined as

$$\tilde{\mathbf{K}} := \int_{\Omega} \tilde{\boldsymbol{T}}^T \mathbb{B}_b^T \mathbf{D} \mathbb{B}_b \tilde{\boldsymbol{T}} d\Omega + \int_{\Omega} \tilde{\boldsymbol{T}}^T \mathbb{B}_s^T \mathbf{D} \mathbb{B}_s \tilde{\boldsymbol{T}} d\Omega, \tag{5.4.44}$$

then the Eq.(5.4.43) can be rewritten as

$$\left\| e_i^{\phi} \right\|_E = \left[ (\tilde{\bar{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P} \bar{\boldsymbol{\phi}}_i^h)^T \tilde{\mathbf{K}} (\tilde{\bar{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P} \bar{\boldsymbol{\phi}}_i^h) \right]^{\frac{1}{2}}. \tag{5.4.45}$$

For the reason that the adaptive mesh requires a local criterion, by referring to $\Omega_e$ as an element-wise physical domain, the local error estimator of eigenvector is posed, i.e.

$$\left\| e_i^{\boldsymbol{\phi}}(\Omega_e) \right\|_E = \left[ (\tilde{\bar{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P} \bar{\boldsymbol{\phi}}_i^h)_{\Omega_e}^T \tilde{\mathbf{K}}_{\Omega_e} (\tilde{\bar{\boldsymbol{\phi}}}_{\tilde{i}} - \mathbb{P} \bar{\boldsymbol{\phi}}_i^h)_{\Omega_e} \right]^{\frac{1}{2}}. \tag{5.4.46}$$

Letting $N$ denote the total number of elements, it yields

$$\left\| e_i^{\boldsymbol{\phi}} \right\|_E^2 = \sum_{j=1}^{N} \left\| e_i^{\boldsymbol{\phi}}(\Omega_e^j) \right\|_E^2. \tag{5.4.47}$$

### 5.4.2 Marking method

In order to take the error contribution by each cell into consideration, the marking strategy is proposed based on the approach [127], which provides the idea that the cells with largest errors can be marked, leading to a efficient adaptive refinement. To be specific, we sort the values of $\left\| e_i^{\boldsymbol{\phi}}(\Omega_e^j) \right\|_E^2 (j = 1, 2, \ldots, N)$ from large to small. Then mark the set of $N^*$ elements to be refined, if the following criterion is satisfied

$$\sum_{j=1}^{N^*} \left\| e_i^{\boldsymbol{\phi}}(\Omega_e^j) \right\|_E^2 \geqslant \tau \left\| e_i^{\boldsymbol{\phi}} \right\|_E^2, \tag{5.4.48}$$

where $\tau \in (0, 1]$ is the percentage of the norm of the error over the whole domain, $N^*$ is the minimum number of elements to satisfy Eq.(5.4.48). Each marked element will be subdivided into 4 elements in case of $d = 2, L_e = 1$. The interpretation for the adaptive PHT refinement process is presented in Fig.5.7, and more details on PHT refinement can be found in Section 4.2. The adaptivity for mode $i$ will proceed until both of the following criteria are fulfilled

$$\left| e_i^{\lambda} \right| \leqslant \tau_{\lambda}, \ \delta_i^{\boldsymbol{\phi}} \leqslant \tau_{\boldsymbol{\phi}}, \tag{5.4.49}$$

where $\tau_\lambda$ and $\tau_\phi$ are error tolerances for the frequency and the eigenvector respectively.

In terms of the option for $\tau$ by a given accuracy, there is always a compromise between refinement steps and number of elements to be refined at each step. When $\tau \ll 1$, it may achieve an optimal final mesh, however, it would sacrifice the computational cost due to too many refinement steps. Whilst if $\tau$ is too large, the effect of adaptivity would be diminished, since $\tau = 1$ leads to the uniform refinement. In this study, the following numerical test is carried out to perform the methodology to determine the coefficient $\tau$. The structure is presented in Fig.5.27 and the details of the simulation can be found in Section 5.6.3. Here, the computational time is considered as the key factor to identify the $\tau$, which means the adopted $\tau$ is expected to minimize time consuming. Look at the Fig.5.6, 9 numerical tests are simulated with $\tau = [0.1, 0.5]$ and related computational time is recorded. Then assume that the relationship between CPU time $t$ and $\tau$ is a parabolic function, which reads

$$t = a(\tau - \tau_0)^2 + t_0. \tag{5.4.50}$$

Then fitting the points of numerical experiments through least square method, the constants $a$, $\tau_0$ and $t_0$ in Eq.(5.4.50) can be obtained as $a = 6.67 \times 10^4$, $\tau_0 \approx 0.3$ and $t_0 = 7132$. Therefore, it is obvious that $\tau = 0.3$ offers an optimal balance in the context of this work.

## 5.5 Adaptivity for a range of frequencies

Following the section above, the method to deal with the adaptivity when modes are inside a band of frequencies of interest will be discussed.

### 5.5.1 Algorithm of adaptivity by sweeping modes

Assume that frequencies of interest are inside a band $\lambda_i^h \in [\lambda_{min}, \lambda_{max}]$ (marked with red dash line in Fig.5.8), including four modes $\lambda_3^h, \lambda_4^h, \lambda_5^h, \lambda_6^h$. Begin with the mode 3 (the lowest mode). After multiplicity identification, it is a unique mode. Then call

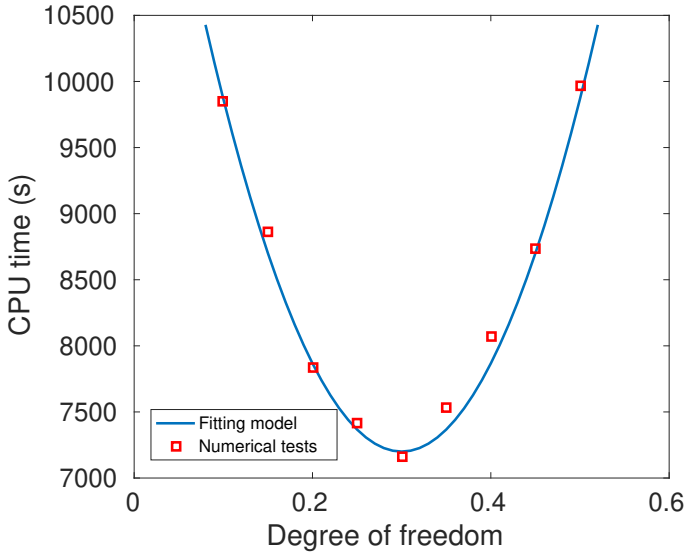



Figure 5.6: The numerical tests and the fitting model to recognize the parameter $\tau$.

**Algorithm 1:** Adaptivity process for the single mode $i$

**Input:** $\left|e_i^{\lambda}\right|$ and $\delta_i^{\phi}$ on coarse mesh $\mathbb{T}$.

**Output:** Updated $\mathbb{T}$ after refinement.

**while** $\left|e_i^{\lambda}\right| \leqslant \tau_{\lambda},\ \delta_i^{\phi} \leqslant \tau_{\phi}$ **do**

  **for** $j \leftarrow 1$ **to** $N$ **do**

    Compute $\left\|e_i^{\phi}(\Omega_e^j)\right\|_E^2$ by Eq.(5.4.46).

  **end**

  Sort values of $\left\|e_i^{\phi}(\Omega_e^j)\right\|_E^2$ from large to small.

  **for** $j \leftarrow 1$ **to** $N$ **do**

    **if** $\sum_{j^*=1}^{j} \left\|e_i^{\phi}(\Omega_e^{j^*})\right\|_E^2 \geqslant \tau \left\|e_i^{\phi}\right\|_E^2$ **then**

      Mark $N^* = j$

      **break**

    **end**

  **end**

  **for** $j \leftarrow 1$ **to** $N^*$ **do**

    Refine element $j$ to update $\mathbb{T}$.

  **end**

  Renew $\left|e_i^{\lambda}\right|$ and $\delta_i^{\phi}$ by Eq.(5.4.39).

**end**

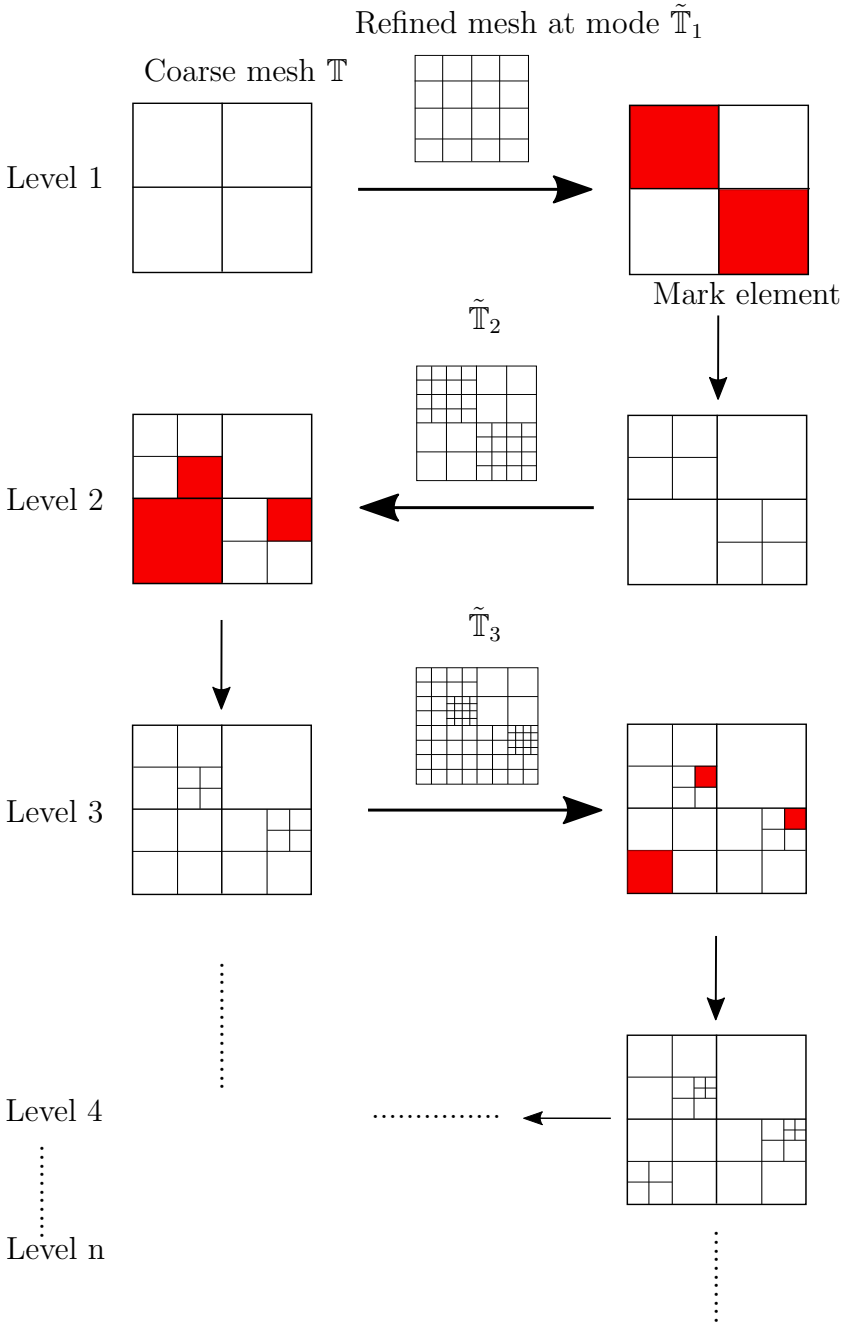



Figure 5.7: The schematic illustration for adaptive PHT refinement procedure in parametric domain at mode $i$ in case of $d = 2, L_e = 1$.

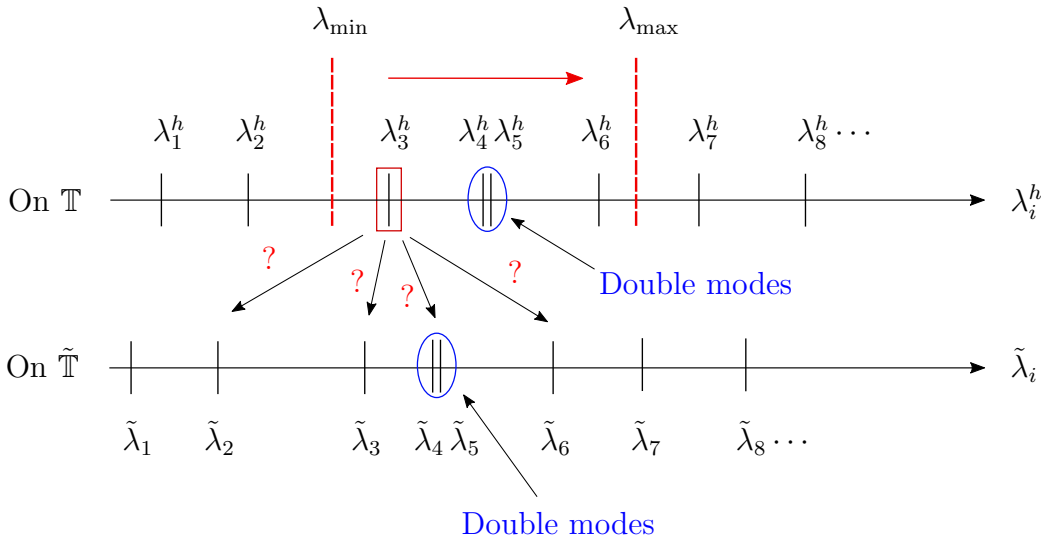



Figure 5.8: The schematic of adaptivity algorithm for an interval of frequencies of interest by sweeping modes from low to high.

Algorithm 1 directly to finish the adaptivity. Next, move to mode 4. By checking multiplicity, mode 4 and mode 5 are double modes. It is noticed that the method to recognize the modal multiplicity is numerical since there no real multiple modes actually. For instance, if $|\lambda_4^h - \lambda_5^h| \leqslant \tau_\lambda^{\text{mul}}$ is satisfied, where $\tau_\lambda^{\text{mul}}$ is a tolerance, mode 4 and mode 5 are regarded as a double mode. The adaptivity for double mode $(4, 5)$ can be achieved through applying Algorithm 5. Hence, the adaptivity for the frequencies of interest can be conducted by sweeping the modes from mode 4 to mode 6. This process is summarized to be the algorithm presented in Algorithm 2.

### 5.5.2 Location of modal correspondence

As the focus is put on the error estimation and adaptivity in Section 5.4, mode $i$ and mode $\tilde{i}$ are assumed to be related. But in fact, for example, as illustrated in Fig.5.8, $\lambda_3^h$ could be related to any mode on $\tilde{\mathbb{T}}$, such as $\tilde{\lambda}_2$, $\tilde{\lambda}_3$, $\tilde{\lambda}_4$, $\tilde{\lambda}_5$ or $\tilde{\lambda}_6$. Therefore, it is required to find the method to recognize this modal resemblance. Two approaches are introduced in the following sections.

**Algorithm 2:** Adaptivity for a range of frequencies of interest $[\lambda_{\min}, \lambda_{\max}]$

**while** $\lambda_{min} \leqslant \lambda_i \leqslant \lambda_{max}$ **do**

**Step 1.** Call Algorithm 3 or Algorithm 4 to find the multiplicity $n$ of mode $i$ on $\mathbb{T}$, and the related modes on $\tilde{\mathbb{T}}$.

**Step 2.**

**if** $n = 1$ **then**

Call Algorithm 1 to perform adaptivity for mode $i$.

**else**

Call Algorithm 5 to process adaptivity for $n$ multiple modes $\{i, \ldots, i+n-1\}$.

**end**

**Step 3.** $\begin{cases} i = i + 1, \text{ for single mode.} \\ i = i + n, \text{ for } n \text{ multiple modes.} \end{cases}$

**end**

#### 5.5.2.1 Frequency Error Criterion (FEC)

The FEC strategy is to regard the modes, with the closest frequencies, as the related modes. The algorithm is summarized in Algorithm 3. Specifically, when using FEC strategy, first of all, we check the multiplicity of mode $i$ with $\lambda_i^h \in [\lambda_{\min},\ \lambda_{\max}]$. If $|\lambda_{i+1}^h - \lambda_i^h| > \tau_\lambda^{\text{mul}}$, then mode $i$ is considered as a single mode. While, if the modes satisfy the following conditions

$$|\lambda_{i+1} - \lambda_i| \leqslant \tau_\lambda^{\text{mul}}, |\lambda_{i+2} - \lambda_{i+1}| \leqslant \tau_\lambda^{\text{mul}}, \cdots, |\lambda_{i+n-1} - \lambda_{i+n-2}| \leqslant \tau_\lambda^{\text{mul}},$$

the multiplicity of mode $i$ is $n$.

Then, we have the set of multiple modes $\{i, i+1, \ldots, i+n-1\}$.

Now, we need to find the corresponding modes on $\tilde{\mathbb{T}}$. If mode $i$ is a single mode, we compute the set of absolute values of errors

$$\{|e_{i,\tilde{i}}|\} = \{|\lambda_i - \tilde{\lambda}_{\tilde{i}}|\},\ \forall \tilde{\lambda}_{\tilde{i}} \in [\lambda_{\min} - a,\ \lambda_{\max} + a], \tag{5.5.51}$$

where the constant $a$ is to make sure the interval $[\lambda_{\min} - a,\ \lambda_{\max} + a]$ is wide enough

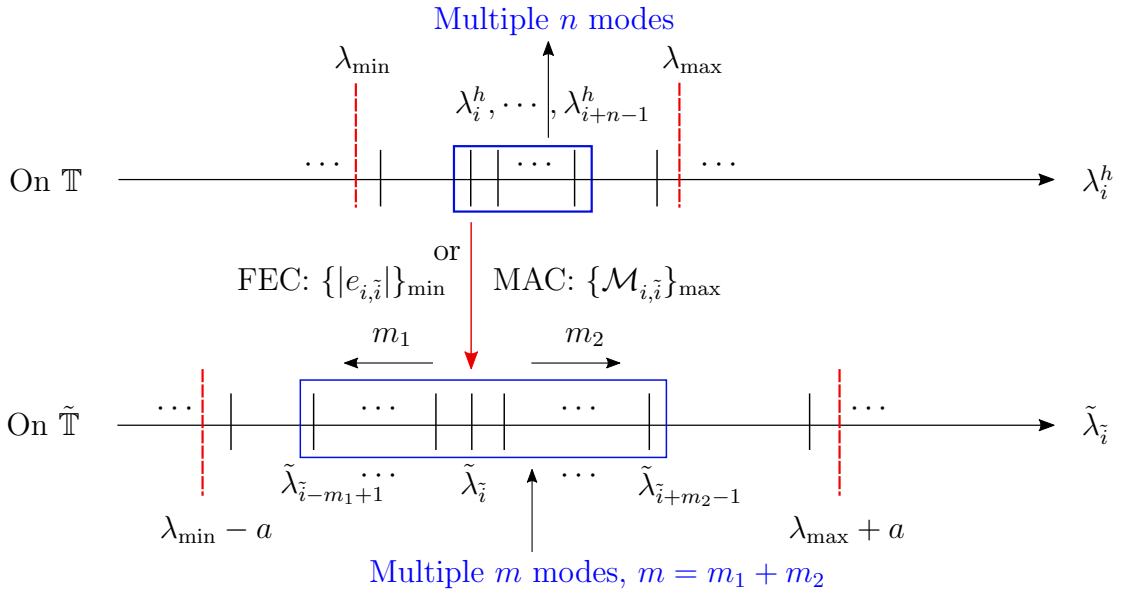



Figure 5.9: The interpretation of the modal resemblance location for multiple modes by FEC and MAC scheme.

to include the corresponding mode inside. Select the minimum of $\{|e_{i,\tilde{i}}|\}$, and then the relevant $\tilde{i}$ is the corresponding mode number.

When modes $\{i, i+1, \ldots, i+n-1\}$ are $n$ multiple modes, the approach is presented in the Fig.5.9. We firstly still find the mode $\tilde{i}$ by obtaining the minimum of $\{|e_{i,\tilde{i}}|\}$ by Eq.(5.5.51). Next, we check the multiplicity of mode $\tilde{i}$. It is required to check the modes lower than mode $\tilde{i}$, as well the modes higher than mode $\tilde{i}$ (look at Fig.5.9). By this way, we can have that

If

$$|\tilde{\lambda}_{\tilde{i}-m_1+2} - \tilde{\lambda}_{\tilde{i}-m_1+1}| \leqslant \tau_\lambda^{\text{mul}}, \cdots, |\tilde{\lambda}_{\tilde{i}} - \tilde{\lambda}_{\tilde{i}-1}| \leqslant \tau_\lambda^{\text{mul}}, \ |\tilde{\lambda}_{\tilde{i}+1} - \tilde{\lambda}_{\tilde{i}}| \leqslant \tau_\lambda^{\text{mul}},$$
$$\cdots, |\tilde{\lambda}_{\tilde{i}+m_2-1} - \tilde{\lambda}_{\tilde{i}+m_2-2}| \leqslant \tau_\lambda^{\text{mul}},$$

then, the multiplicity of mode $\tilde{i}$ is $m = m_1 + m_2$.

Reset $\tilde{i} = \tilde{i} - m_1 + 1$, and thus we obtain the related multiple modes $\{\tilde{i}, \tilde{i}+1, \ldots, \tilde{i}+m-1\}$.

**Algorithm 3:** Identification of mode multiplicity and location of mode correspondence by FEC

**Input:** $\lambda_i$ and mode $i$ on $\mathbb{T}$.

**Output:** Multiplicity $n$ of mode $i$ and the related set of modes $\{\tilde{i}, \ldots, \tilde{i}+m-1\}$ on $\tilde{\mathbb{T}}$.

$n = 1,\ j = i + 1.$

**while** $\lambda_j^h \leqslant \lambda_{max}$ ; `/* Find multiplicity` $n$ `of mode` $i$ `*/`

**do**

  **if** $\left|\lambda_j^h - \lambda_{j-1}^h\right| \leqslant \tau_\lambda^{mul}$ **then**

    $j = j + 1.$

    $n = n + 1.$

  **else**

    **break**

  **end**

**end**

The multiplicity of mode $i$ is $n$.

**while** $\lambda_{min} - a \leqslant \tilde{\lambda}_{\tilde{i}} \leqslant \lambda_{max} + a$ **do**

  Compute $\{|e_{i,\tilde{i}}|\} = |\lambda_i^h - \tilde{\lambda}_{\tilde{i}}|$.

  $\tilde{i} = \tilde{i} + 1.$

**end**

Select the minimum of $\{|e_{i,\tilde{i}}|\}$ and mark the relevant $\tilde{i}$.

**Algorithm 3:** Identification of mode multiplicity and location of mode correspondence by FEC (continued...)

**if** $n = 1$ **then**

The mode $i$ is related to mode $\tilde{i}$.

**else**

$m_1 = 1,\ j = \tilde{i} - 1.$ ; `/* Find multiplicity` $m$ `of mode` $\tilde{i}$ `*/`

**while** $\lambda_{min} - a \leqslant \tilde{\lambda}_j$ **do**

**if** $\left|\tilde{\lambda}_j - \tilde{\lambda}_{j+1}\right| \leqslant \tau_\lambda^{mul}$ **then**

$j = j - 1.$

$m_1 = m_1 + 1.$

**else**

**break**

**end**

**end**

$m_2 = 1,\ j = \tilde{i} + 1.$

**while** $\tilde{\lambda}_j \leqslant \lambda_{max} + a$ **do**

**if** $\left|\tilde{\lambda}_j - \tilde{\lambda}_{j-1}\right| \leqslant \tau_\lambda^{mul}$ **then**

$j = j + 1.$

$m_2 = m_2 + 1.$

**else**

**break**

**end**

**end**

The multiplicity of mode $\tilde{i}$ is $m = m_1 + m_2$.

Reset $\tilde{i} = \tilde{i} - m_1 + 1$. Then, the multiple modes $\{i, \ldots, i + n - 1\}$ are related to multiple modes $\{\tilde{i}, \ldots, \tilde{i} + m - 1\}$.

**end**

#### 5.5.2.2 Modal Assurance Criterion (MAC)

Modal Assurance Criterion (MAC) is a statistical indicator originally proposed for orthogonality check [45] and has been developed as one of the most well-known method to compare modal vectors quantitatively [63]. It is widely used to build correlation between analytical and experimental modal vectors for validation of experiment. In this chapter, we utilize it to locate the correspondence between two computational modal shapes. The values of MAC are computed by computational eigenvectors, and then they are assembled into MAC matrix $\mathcal{M}$ by using the formula

$$\mathcal{M}_{i,j}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \frac{\int_\Omega \boldsymbol{\phi}_i^T \mathbf{m} \boldsymbol{\phi}_j d\Omega}{\|\boldsymbol{\phi}_i\|_m^2 \, \|\boldsymbol{\phi}_j\|_m^2}, \tag{5.5.52}$$

where $\|\cdot\|_m$ is the mass norm and defined by

$$\|\cdot\|_m := \left[\int_\Omega (\cdot)^T \mathbf{m} (\cdot) d\Omega\right]^{\frac{1}{2}}, \tag{5.5.53}$$

and $\mathbf{m}$ is defined in Eq.(5.3.32). It is worth noticing that the matrix $\mathbf{m}$ is used in mass norm. As seen in the expression of Eq.(5.3.32), it means when identifying correspondence between two modal shapes, deflection $w$ and rotations $\theta_x, \theta_y$ are all taken into the consideration, which is because in terms of Mindlin plate, these quantities are independent. Note that, if the eigenvectors $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ are obtained by the same mesh, the term $\int_\Omega \boldsymbol{\phi}_i^T \mathbf{m} \boldsymbol{\phi}_j d\Omega$ would be computed directly. If not, we should use projection to ensure integral is processed in the same domain. The details of projection can be found in Appendix A. The values of the MAC are located in the interval $[0, 1]$, where 0 means no consistent resemblance whereas 1 means a consistent correspondence. Generally, it is accepted that large values denote relatively consistent correlation whilst small value represents poor association.

In the MAC method, for mode $i$ with $\lambda_i^h \in [\lambda_{\min},\ \lambda_{\max}]$, if the MAC value $\mathcal{M}_{i,i+1} < \tau_{\text{MAC}}$ , where $\tau_{\text{MAC}}$ is the tolerance, mode $i$ is interpreted as a single mode.

If the modes are multiple, such that

$$\mathcal{M}_{i,i+1} \geqslant \tau_{\text{MAC}},\ \mathcal{M}_{i+1,i+2} \geqslant \tau_{\text{MAC}}, \cdots,\ \mathcal{M}_{i+n-2,i+n-1} \geqslant \tau_{\text{MAC}},$$

the multiplicity of mode $i$ is $n$, and the set of multiple modes are $\{i, i+1, \ldots i-n+1\}$.

The strategy to deal with the multiple modes is similar to the FEC scheme, as shown in Fig.5.9. The slight difference is that, in MAC method, we find the mode $\tilde{i}$ by the maximum of $\{\mathcal{M}_{i,\tilde{i}}\},\ \forall\tilde{\lambda}_{\tilde{i}} \in [\lambda_{\min} - a,\ \lambda_{\max} + a]$. Also, we use MAC values to recognize the multiplicity of mode $\tilde{i}$ by following:

If

$$\mathcal{M}_{\tilde{i}-m_1+2,\ \tilde{i}-m_1+1} \geqslant \tau_{\text{MAC}}, \cdots, \mathcal{M}_{\tilde{i},\tilde{i}-1} \geqslant \tau_{\text{MAC}},\ \mathcal{M}_{\tilde{i}+1,\tilde{i}} \geqslant \tau_{\text{MAC}}, \cdots, \mathcal{M}_{\tilde{i}+m_2-1,\ \tilde{i}+m_2-2} \geqslant \tau_{\text{MAC}},$$

the multiplicity of mode $\tilde{i}$ is $m = m_1 + m_2$.

Reset $\tilde{i} = \tilde{i} - m_1 + 1$, and then we obtain the related multiple modes $\{\tilde{i}, \tilde{i}+1, \ldots, \tilde{i}+m-1\}$.

For instance, an example of MAC matrix $\mathcal{M}$ is illustrated with 3D view in Fig.5.10. It is obvious that, the single modes 1,2 and 3 on coarse mesh are correlated to the single modes 1,2 and 3 on refined mesh, respectively. In addition, double modes 4 and 5 on $\mathbb{T}$ are associated to the double modes 4 and 5 on $\tilde{\mathbb{T}}$.

These two schemes, that is, FEC and MAC, will be compared in the numerical examples Section 5.6.2.

### 5.5.3 Adaptivity for multiple modes

In Section 5.4, the adaptivity strategy for a single mode was established. In this section, it is intended to propose a methodology to deal with the adaptivity for multiple modes.

Suppose that the $n$ multiple modes $\{i, i+1, \ldots, i+n-1\}$ on coarse mesh $\mathbb{T}$ are related to $m$ multiple modes $\{\tilde{i}, \tilde{i}+1, \ldots, \tilde{i}+m-1\}$ on refined mesh $\tilde{\mathbb{T}}$. Then all the vectors for modes $\{i, \ldots, i+n-1\}$ are actually defined by the linear combinations

**Algorithm 4:** Identification of mode multiplicity and location of mode correspondence by MAC

**Input:** $\lambda_i$ and mode $i$ on $\mathbb{T}$.

**Output:** Multiplicity $n$ of mode $i$ and the related set of modes $\left\{\tilde{i}, \ldots, \tilde{i}+m-1\right\}$ on $\tilde{\mathbb{T}}$.

$n = 1,\ j = i+1$

**while** $\lambda_{min} \leqslant \lambda_j \leqslant \lambda_{max}$ ; `/* Find multiplicity` $n$ `of mode` $i$ `*/`

**do**

  Compute $\mathcal{M}(\boldsymbol{\phi}_j^h, \boldsymbol{\phi}_{j-1}^h)$ by Eq.(5.5.52).

  **if** $\mathcal{M}(\boldsymbol{\phi}_j^h, \boldsymbol{\phi}_{j-1}^h) \geqslant \tau_{MAC}$ **then**

    $j = j+1.$

    $n = n+1.$

  **else**

    **break**

  **end**

**end**

The multiplicity of mode $i$ is $n$.

**while** $\lambda_{min} - a \leqslant \tilde{\lambda}_{\tilde{i}} \leqslant \lambda_{max} + a$ **do**

  Compute $\left\{\mathcal{M}_{i,\tilde{i}}(\boldsymbol{\phi}_i^h, \tilde{\boldsymbol{\phi}}_{\tilde{i}})\right\}$ by Eq.(5.5.52).

  $\tilde{i} = \tilde{i}+1.$

**end**

Select the maximum of $\left\{\mathcal{M}_{i,\tilde{i}}\right\}$ and mark the relevant $\tilde{i}$.

**Algorithm 4:** Identification of mode multiplicity and location of mode correspondence by MAC (continued...)

**if** $n = 1$ **then**
  The mode $i$ is related to mode $\tilde{i}$.
**else**
  $m_1 = 1,\ j = \tilde{i} - 1.$
  **while** $\lambda_{min} - a \leqslant \tilde{\lambda}_j$ ; `/* Find multiplicity` $m$ `of mode` $\tilde{i}$ `*/`
  **do**
    Compute $\mathcal{M}(\tilde{\boldsymbol{\phi}}_j, \tilde{\boldsymbol{\phi}}_{j+1})$ by Eq.(5.5.52).
    **if** $\mathcal{M}(\tilde{\boldsymbol{\phi}}_j, \tilde{\boldsymbol{\phi}}_{j+1}) \geqslant \tau_{MAC}$ **then**
      $j = j - 1.$
      $m_1 = m_1 + 1.$
    **else**
      **break**
    **end**
  **end**
  $m_2 = 1,\ j = \tilde{i} + 1.$
  **while** $\tilde{\lambda}_j \leqslant \lambda_{max} + a$ **do**
    **if** $\mathcal{M}(\tilde{\boldsymbol{\phi}}_j, \tilde{\boldsymbol{\phi}}_{j-1}) \geqslant \tau_{\lambda}^{mul}$ **then**
      $j = j + 1.$
      $m_2 = m_2 + 1.$
    **else**
      **break**
    **end**
  **end**
  The multiplicity of mode $\tilde{i}$ is $m = m_1 + m_2$.
  Reset $\tilde{i} = \tilde{i} - m_1 + 1$. Then, The multiple modes $\{i, \ldots, i + n - 1\}$ are related to multiple modes $\{\tilde{i}, \ldots, \tilde{i} + m - 1\}$.
**end**

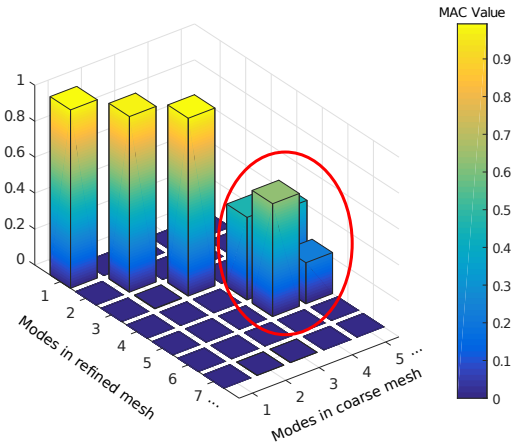



Figure 5.10: An example of 3D view for a MAC matrix obtained between coarse and refined meshes. Specially, the blocks of MAC values in red circle mean that double modes arise.

of basis eigenvectors $\boldsymbol{\Phi}$ in a eigenspace $\mathscr{P}$

$$\mathscr{P} = \{\boldsymbol{\varphi} : \boldsymbol{\varphi} = \boldsymbol{\Phi}\boldsymbol{\alpha}, \quad \boldsymbol{\alpha} \in \mathbb{R}^n\}, \tag{5.5.54}$$

where $\boldsymbol{\alpha} = (\alpha_i \ \cdots \ \alpha_{i+n-1})^T$ are arbitrary coefficients. The basis eigenvectors $\boldsymbol{\Phi}$ are defined by

$$\boldsymbol{\Phi} = (\boldsymbol{\phi}_i^h \quad \cdots \quad \boldsymbol{\phi}_{i+n-1}^h) \in \mathbb{R}^{3\times n},$$

where $\boldsymbol{\phi}_i^h = [\boldsymbol{\phi}_{i,w}^h \quad \boldsymbol{\phi}_{i,\theta_x}^h \quad \boldsymbol{\phi}_{i,\theta_y}^h]^T$ is the solution field for mode $i$. Likewise, the eigenspace for multiple modes $\{\tilde{i}, \ldots, \tilde{i}+m-1\}$ is defined by

$$\tilde{\mathscr{P}} = \{\tilde{\boldsymbol{\varphi}} : \tilde{\boldsymbol{\varphi}} = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\alpha}}, \quad \tilde{\boldsymbol{\alpha}} \in \mathbb{R}^m\}, \tag{5.5.55}$$

where

$$\tilde{\boldsymbol{\Phi}} = (\tilde{\boldsymbol{\phi}}_{\tilde{i}} \quad \cdots \quad \tilde{\boldsymbol{\phi}}_{\tilde{i}+m-1}) \in \mathbb{R}^{3\times m}, \ \tilde{\boldsymbol{\phi}}_{\tilde{i}} = [\tilde{\boldsymbol{\phi}}_{\tilde{i},w} \quad \tilde{\boldsymbol{\phi}}_{\tilde{i},\theta_x} \quad \tilde{\boldsymbol{\phi}}_{\tilde{i},\theta_y}]^T.$$

Now we aim to define error estimator of eigenvectors $e_{\boldsymbol{\phi}}$, between $\mathscr{P}$ and $\tilde{\mathscr{P}}$. The strategy, briefly summarized, is to project all the vectors in $\mathscr{P}$ onto $\tilde{\mathscr{P}}$, and then compute all the errors between the projected vectors and the vectors in $\tilde{\mathscr{P}}$.

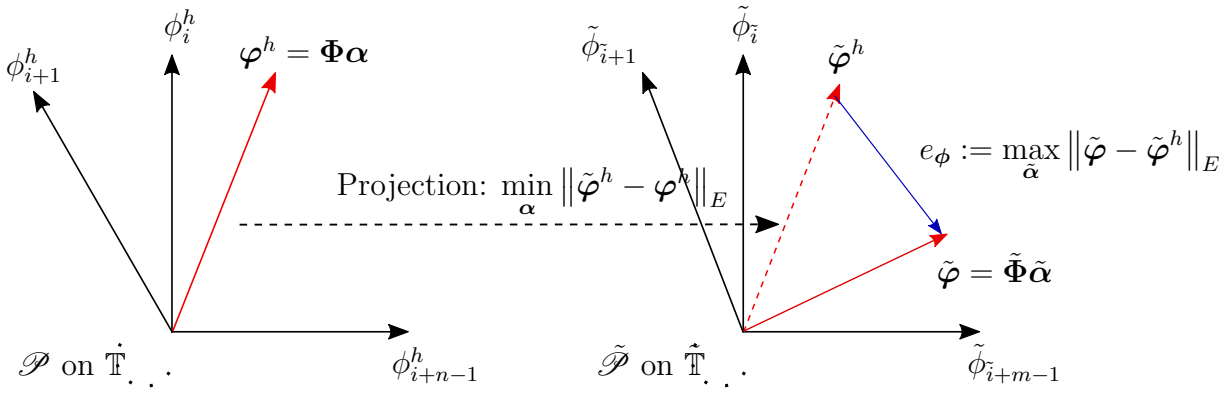



Figure 5.11: The schematic of the measurement of the error estimator of eigenvectors for multiple modes between $\mathscr{P}$ and $\tilde{\mathscr{P}}$.

Among these errors, the maximum will be regarded as the $e_{\boldsymbol{\phi}}$. This process is presented in Fig.5.11. As shown in Fig.5.11, the minimization of the projection and the maximization of the errors can be defined as the following problem

Find $\boldsymbol{\alpha}$ and $\tilde{\boldsymbol{\alpha}}$ such that

$$e_{\boldsymbol{\phi}} := \max_{\tilde{\boldsymbol{\alpha}}}, \min_{\boldsymbol{\alpha}} \left\|\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\alpha}} - \boldsymbol{\Phi}\boldsymbol{\alpha}\right\|_E, \tag{5.5.56}$$

with $\|\tilde{\boldsymbol{\alpha}}\|_2 = \sqrt{\tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\alpha}}} = 1$.

Aiming to solve the problem in Eq.(5.5.56), we build a Lagrange function $\mathscr{L}(\cdot)$ such that

$$\mathscr{L}(\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}, \mu) = \left\|\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\alpha}} - \boldsymbol{\Phi}\boldsymbol{\alpha}\right\|_E^2 + \mu\left(\|\tilde{\boldsymbol{\alpha}}\|_2^2 - 1\right), \tag{5.5.57}$$

where $\|\tilde{\boldsymbol{\alpha}}\|_2 = 1$ specifies the range of vectors $\|\tilde{\boldsymbol{\varphi}}\|$ and $\left\|\boldsymbol{\varphi}^h\right\|$. Then the problem defined in Eq.(5.5.56) can also be mathematically understood as follows

Find $(\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}, \mu)$, such that,

$$\delta_{\boldsymbol{\alpha}}\mathscr{L} = 0, \Rightarrow \delta_{\boldsymbol{\alpha}}\left\|\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\alpha}} - \boldsymbol{\Phi}\boldsymbol{\alpha}\right\|_E^2 = 0, \tag{5.5.58}$$

$$\delta_{\tilde{\boldsymbol{\alpha}}}\mathscr{L} = 0, \Rightarrow \delta_{\tilde{\boldsymbol{\alpha}}}\left\|\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\alpha}} - \boldsymbol{\Phi}\boldsymbol{\alpha}\right\|_E^2 + \mu\cdot\delta_{\tilde{\boldsymbol{\alpha}}}\left(\|\tilde{\boldsymbol{\alpha}}\|_2^2 - 1\right) = 0, \tag{5.5.59}$$

$$\delta_{\mu}\mathscr{L} = 0, \Rightarrow \|\tilde{\boldsymbol{\alpha}}\|_2 = 1. \tag{5.5.60}$$

The basis eigenvectors $\boldsymbol{\Phi}$ and $\tilde{\boldsymbol{\Phi}}$ are discretized by PHT-spline basis functions $\boldsymbol{T}$ and

$\tilde{\boldsymbol{T}}$, that is,

$$\boldsymbol{\Phi} = \boldsymbol{T}\bar{\boldsymbol{\Phi}},\ \tilde{\boldsymbol{\Phi}} = \tilde{\boldsymbol{T}}\bar{\tilde{\boldsymbol{\Phi}}}, \tag{5.5.61}$$

where $\bar{\boldsymbol{\Phi}} \in \mathbb{R}^{3k\times n}$ and $\bar{\tilde{\boldsymbol{\Phi}}} \in \mathbb{R}^{3\tilde{k}\times m}$ are control variables of eigenvectors, and $k$ and $\tilde{k}$ are the numbers of control variables over $\mathbb{T}$ and $\tilde{\mathbb{T}}$, respectively. As discussed in Section 5.4.1, since refined mesh $\tilde{\mathbb{T}}$ is obtained by the hierarchical refinement of coarse mesh $\mathbb{T}$, we can have the prolongation of $\bar{\boldsymbol{\Phi}}$ in the $\tilde{\mathbb{T}}$, i.e.,

$$\boldsymbol{\Phi} = \boldsymbol{T}\bar{\boldsymbol{\Phi}} = \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\Phi}}. \tag{5.5.62}$$

Substitution of Eq.(5.5.61) and Eq.(5.5.62) into Eq.(5.5.58) and solving the equation, yields

$$\bar{\boldsymbol{\Phi}}\mathbf{K}\bar{\boldsymbol{\Phi}}\boldsymbol{\alpha} = (\mathbb{P}\bar{\boldsymbol{\Phi}})^T\tilde{\mathbf{K}}\bar{\tilde{\boldsymbol{\Phi}}}\tilde{\boldsymbol{\alpha}}, \tag{5.5.63}$$

where $\mathbf{K}$ and $\tilde{\mathbf{K}}$ are the stiffness matrices on $\mathbb{T}$ and $\tilde{\mathbb{T}}$, defined in Eq.(5.3.37). Defining a $n \times m$ matrix $\mathbf{A}$, that is,

$$\mathbf{A} := \left((\mathbb{P}\bar{\boldsymbol{\Phi}})^T\tilde{\mathbf{K}}\bar{\tilde{\boldsymbol{\Phi}}}\right)^{-1}\bar{\boldsymbol{\Phi}}\mathbf{K}\bar{\boldsymbol{\Phi}},$$

we can rewrite Eq.(5.5.63), which reads

$$\boldsymbol{\alpha} = \mathbf{A}\tilde{\boldsymbol{\alpha}}. \tag{5.5.64}$$

Similarly, substituting Eq.(5.5.64) into Eq.(5.5.59) and solving the equation, we have an eigenvalue problem for $\tilde{\boldsymbol{\alpha}}$

$$\mathbf{K}^*\tilde{\boldsymbol{\alpha}} = \mu\tilde{\boldsymbol{\alpha}}, \tag{5.5.65}$$

where

$$\mathbf{K}^* = \bar{\tilde{\boldsymbol{\Phi}}}^T\tilde{\mathbf{K}}\bar{\tilde{\boldsymbol{\Phi}}} + \mathbf{A}^T\bar{\boldsymbol{\Phi}}\mathbf{K}\bar{\boldsymbol{\Phi}}\mathbf{A} - \mathbf{A}(\mathbb{P}\bar{\boldsymbol{\Phi}})^T\tilde{\mathbf{K}}\bar{\tilde{\boldsymbol{\Phi}}}.$$

Solving the eigenvalue problem in Eq.(5.5.65), $\tilde{\boldsymbol{\alpha}}$ can be obtained. Then, substituting $\tilde{\boldsymbol{\alpha}}$ into Eq.(5.5.64), $\boldsymbol{\alpha}$ will be obtained. Furthermore, the error estimator $e_{\boldsymbol{\phi}}$ in Eq.(5.5.56) can be determined. Additionally, the local error estimator $e_{\boldsymbol{\phi}}(\Omega_e)$ can be given by

$$e_{\boldsymbol{\phi}}(\Omega_e) = \left[\int_{\Omega_e}(\mathbb{B}_b\boldsymbol{e}_{\boldsymbol{\varphi}})^T\mathbf{D}\mathbb{B}_b\boldsymbol{e}_{\boldsymbol{\varphi}}d\Omega_e + \int_{\Omega_e}(\mathbb{B}_s\boldsymbol{e}_{\boldsymbol{\varphi}})^T\mathbf{D}\mathbb{B}_s\boldsymbol{e}_{\boldsymbol{\varphi}}d\Omega_e\right]^{\frac{1}{2}}, \tag{5.5.66}$$

where $\boldsymbol{e}_{\boldsymbol{\varphi}} = \tilde{\boldsymbol{\varphi}} - \boldsymbol{\varphi}$. The marking strategy has been discussed in Section 5.4.2. Define the error estimator of frequency and relative error estimator of eigenvector

$$|e_\lambda| = \left| \frac{1}{n} \sum_{j=i}^{i+n-1} \lambda_j - \frac{1}{m} \sum_{j=\tilde{i}}^{\tilde{i}+m-1} \tilde{\lambda}_j \right|, \ \delta_{\boldsymbol{\phi}} = \frac{e_\phi}{\|\tilde{\boldsymbol{\varphi}}\|_E}, \tag{5.5.67}$$

and subsequently, the adaptivity can be conducted until they satisfy the given thresholds

$$|e_\lambda| \leqslant \tau_\lambda, \ \delta_{\boldsymbol{\phi}} \leqslant \tau_\phi. \tag{5.5.68}$$

Note that the values of $\tau_\lambda$ and $\tau_\phi$ are generally set according to the practical engineering applications. Here, for all numerical tests in this chapter, they are allocated as $\tau_\lambda = 10^{-5}$ and $\tau_\phi = 0.5\%$. The methodology proposed above is summarized in Algorithm 5.

## 5.6 Numerical examples

In this section, three numerical examples are carried out for the following purposes. The example in Section 5.6.1 aims to show that the GIFT method (NURBS for design and PHT splines for analysis) delivers the results with good accuracy, compared to those obtained in IGA framework and using other existing methods . Afterwards, we use GIFT method for examples in both Section 5.6.2 and Section 5.6.3. The example in Section 5.6.2 presents a comparison between two strategies that is, MAC and FEC proposed in Section 5.5.2, for modal resemblance determinations. The results illustrate that the shorter the width of frequencies of interest is, the better the MAC method is to be used. Finally, in Section 5.6.3, we apply MAC method within GIFT framework to study the local adaptivity for structural vibration by sweeping the modes from low to high frequencies. All the algorithms are implemented and numerical tests are conducted via programming in Matlab software. Also, the post-processing including plot, contours, etc.. are proceeded within Matlab.

**Algorithm 5:** Adaptivity process for the $n$ multiple modes $\{i, \ldots, i+n-1\}$

**Input:** Multiple modes $\{i, \ldots i-n+1\}$ on $\mathbb{T}$ and $\{\tilde{i}, \ldots \tilde{i}-m+1\}$ on $\tilde{\mathbb{T}}$.

**Output:** Updated $\mathbb{T}$ after refinement.

**Step 1.** Define eigenspaces $\mathscr{P}$ and $\tilde{\mathscr{P}}$, and vectors $\boldsymbol{\varphi} = \boldsymbol{\Phi}\boldsymbol{\alpha} \in \mathscr{P}$ and $\tilde{\boldsymbol{\varphi}} = \tilde{\boldsymbol{\Phi}}\tilde{\alpha} \in \tilde{\mathscr{P}}$.

**Step 2.** Define the error estimator of eigenvector $e_{\boldsymbol{\phi}}$ in Eq.(5.5.56).

**Step 3.** Build a Lagrange function $\mathscr{L}(\cdot)$ in Eq.(5.5.57), and solve it by Eq.(5.5.58)-(5.5.60) to obtain vectors $\boldsymbol{\varphi}$, $\tilde{\boldsymbol{\varphi}}$ and $e_{\boldsymbol{\phi}}$.

**Step 4.** Compute $|e_\lambda|$ and $\delta_{\boldsymbol{\phi}}$ by Eq.(5.5.67).

**while** $|e_\lambda| \leqslant \tau_\lambda,\ \delta_{\boldsymbol{\phi}} \leqslant \tau_{\boldsymbol{\phi}}$ **do**

  **for** $j \leftarrow 1$ **to** $N$ **do**

    Compute $\|e_{\boldsymbol{\phi}}(\Omega_e^j)\|_E^2$ by Eq.(5.5.66).

  **end**

  Sort values of $\|e_{\boldsymbol{\phi}}(\Omega_e^j)\|_E^2$ from large to small.

  **for** $j \leftarrow 1$ **to** $N$ **do**

    **if** $\sum_{j^*=1}^{j} \|e_{\boldsymbol{\phi}}(\Omega_e^{j^*})\|_E^2 \geqslant \tau e_{\boldsymbol{\phi}}^2$ **then**

      Mark $N^* = j$

      **break**

    **end**

  **end**

  **for** $j \leftarrow 1$ **to** $N^*$ **do**

    Refine element $j$ to update $\mathbb{T}$.

  **end**

  Repeat **Step 1** – **Step 4**.

**end**

### 5.6.1 Homogeneous circular plate

In this example, the comparison of relative error for natural frequency $e^{\lambda}$ obtained by IGA(NURBS), IGA(RHT) and GIFT(NURBS+PHT) and other methods in literatures [128, 129, 130, 131] in case of the vibration of the disk are made. The $e^{\lambda}$ is defined by

$$e^{\lambda:=\log\lambda^h-\log\lambda_{\text{ext}}}, \tag{5.6.69}$$

where $\lambda^h$ and $\lambda_{\text{ext}}$ [132] are computational and exact dimensionless frequency respectively, which are defined by

$$\lambda^h = \frac{\omega^h r^2}{\pi^2}\sqrt{\frac{\rho h}{D}},\ \lambda_{\text{ext}} = \frac{\omega_{\text{ext}} r^2}{\pi^2}\sqrt{\frac{\rho h}{D}}, \tag{5.6.70}$$

where $\omega^h$ and $\omega_{\text{ext}}$ are computational and theoretical frequency respectively, and $D$ denotes $D = \dfrac{Eh^3}{12(1-\nu^2)}$. The material parameters are as follows: Young's modulus $E = 1\ Pa$, density $\rho = 1\ kg/m^3$, Poisson's ratio $\nu = 0.3$, thickness-span ratios $h/r$ ($h$ is the thickness and $r$ is the radius). As mentioned in Section 4.3, without any refinement, the initial RHT splines are NURBS so that bi-cubic IGA(RHT) and IGA(NURBS) share the same control points, as shown in Fig.5.12(b). Whilst in GIFT method, the quadratic NURBS is adopted for geometry as it is precise enough to generate the circular shape, as seen in Fig.5.12(a), and the cubic PHT splines are exploited to represent solution fields. In terms of the uniform refinement, as it can be seen in Fig.5.14, the PHT and RHT mesh are exactly the same.

The comparison of computation time cost using IGA(NURBS), IGA(RHT) and GIFT is carried out in case of clamped boundary condition with ratio $h/r = 0.2$, as shown in Fig.5.20. It can be observed that the CPU time rises almost linearly with the increase of degree of freedom for all the three methods. Besides, obviously, using GIFT method is cheapest. In terms of the comparison between GIFT and IGA (NURBS) as it is mentioned before, in the framework of GIFT, once the initial mesh for geometry is established, it is not needed to be refined any more during the subsequent refinement for solution field. However, in IGA(NURBS), every time refining the solution field, the geometry has to be refined as well, which leads to

extra cost in the generation of new control points and weights. While regarding the contrast between IGA(RHT) and GIFT, since the weights related to the RHT basis functions have to be updated at each refinement step, it requires more computational efforts within IGA(RHT) algorithm than GIFT approach.

Furthermore, it is illustrated from the results in Tab.5.1, Tab.5.2 and Fig.5.15 that, the results obtained by GIFT(NURBS+PHT) method perform a good convergence has an excellent agreement with those computed by IGA method, for the first 6 modes with both simply supported and clamped boundary conditions, in case of $h/r = 0.1$ and $h/r = 0.2$. The vibration of disk was also investigated by other strategies. For instance, in the literature [128], the circular plate was discretized through NURBS shape functions with different mesh levels and patches. As it can be seen in Tab.5.1 and Tab.5.2, NURBS2P-L1 means the structure was refined by 1-level NURBS basis functions in 2 patches. The similar expressions are given to the NURBS2P-L2, NURBS2P-L3 and NURBS5P as well. There are also existing methods, i.e., discrete shear gap triangle element (DSG3) [129], edge-based smoothed discrete shear gap triangular element (ES-DSG3) [129], node-based smoothed stabilized discrete shear gap method using threenode triangular elements (NS-DSG3) [130], and four node assumed natural strain (ANS4) [131]. The proposed GIFT method is compared with those techniques discussed above, as shown in Tab.5.1, Tab.5.2 and from Fig.5.16 to Fig.5.19. It is clear that the GIFT method achieves at least as excellent exactness as other approaches, and in some cases it performs better.

### 5.6.2 Heterogeneous eye shape with a hole

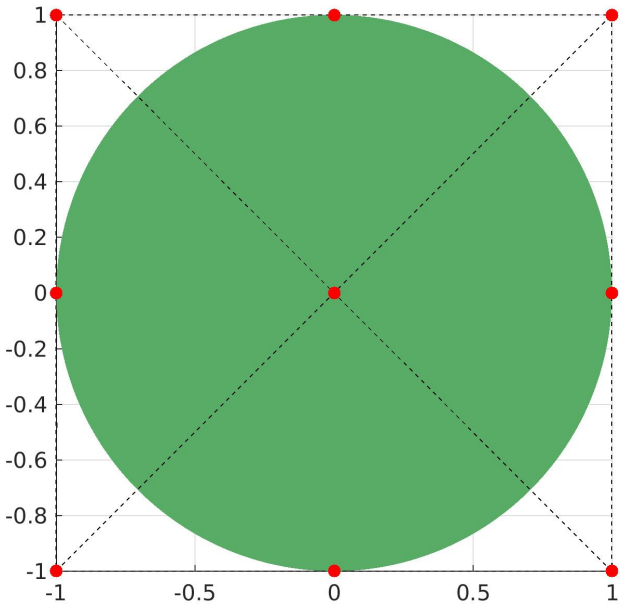
The geometry of eye shape with a hole and its material property are presented in Fig.5.21. The boundary condition of the outer edge is simply supported, and the edge of the hole is free. This structure was built by 8 patches, where patch 1 with $E_1 = 0.03, \rho_1 = 0.7$ is softer and lighter than other patches with $E_i = 1, \rho_i = 1$ ($i = 2 \dots 8$). For all patches, the Poisson's rate are $\nu = 0.3$ and thickness-span ratio is $h/a = 0.1$. The frequencies of interest are in the range of $\lambda_i^h \in [\lambda_{min}, \lambda_{max}]$ (see

Table 5.1: Comparison of dimensionless frequency $\lambda$ for simply supported circular plates.

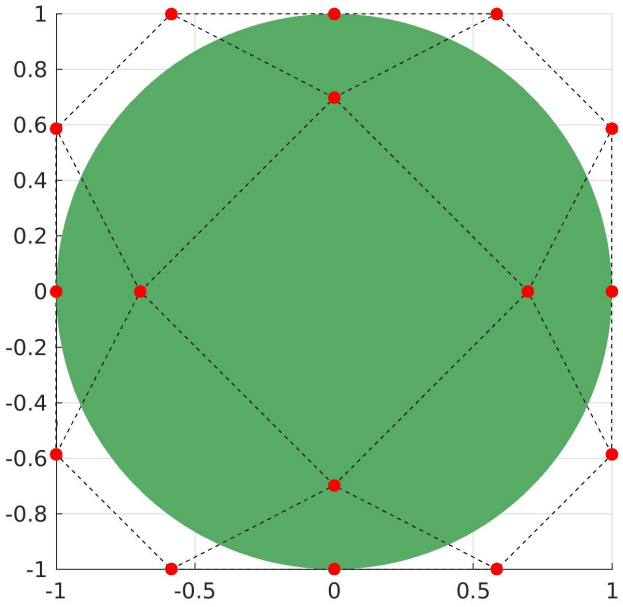
| $h/r$ | Method | Dof | Mode number | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 |
| 0.1 | NURBS | | 4.9997 | 14.592 | 14.592 | 27.102 | 43.7423 | 48.6731 |
| | RHT | 108 | 4.9037 | 15.6303 | 15.6303 | 29.4834 | 34.815 | 39.6105 |
| | GIFT | | 4.9191 | 16.2468 | 16.2468 | 30.1915 | 38.4002 | 43.1109 |
| | NURBS | | 4.8952 | 13.5422 | 13.5422 | 24.4403 | 24.671 | 28.6776 |
| | RHT | 300 | 4.8972 | 13.5873 | 13.5873 | 24.4011 | 24.5953 | 28.325 |
| | GIFT | | 4.9001 | 13.614 | 13.614 | 24.4127 | 24.6605 | 28.3452 |
| | NURBS | | 4.8942 | 13.5139 | 13.5139 | 24.3249 | 24.3254 | 28.255 |
| | RHT | 972 | 4.8943 | 13.5157 | 13.5157 | 24.3316 | 24.3361 | 28.2664 |
| | GIFT | | 4.8944 | 13.5162 | 13.5162 | 24.3323 | 24.338 | 28.2685 |
| | NURBS2P-L1 [128] | | 4.8808 | 13.4335 | 13.9479 | 24.4537 | 24.6549 | 29.292 |
| | NURBS2P-L2 [128] | | 4.8783 | 13.4215 | 13.5176 | 24.1726 | 24.3233 | 28.1771 |
| | NURBS2P-L3 [128] | | 4.8783 | 13.4206 | 13.516 | 24.1638 | 24.3198 | 28.1629 |
| | NURBS5P [128] | | 4.8941 | 13.5138 | 13.514 | 24.326 | 24.3263 | 28.2541 |
| | $\lambda_{\text{ext}}$ [132] | – | 4.894 | 13.51 | 13.51 | 24.313 | 24.313 | 28.24 |
| 0.2 | NURBS | | 4.8432 | 13.1315 | 13.1315 | 23.0597 | 32.3415 | 36.3684 |
| | RHT | 108 | 4.7847 | 13.3414 | 13.3414 | 23.4721 | 27.6379 | 31.5313 |
| | GIFT | | 4.7944 | 13.5496 | 13.5496 | 23.7437 | 29.5505 | 33.4231 |
| | NURBS | | 4.7789 | 12.638 | 12.638 | 21.7494 | 21.8244 | 25.1699 |
| | RHT | 300 | 4.7797 | 12.6534 | 12.6534 | 21.7642 | 21.7981 | 25.0825 |
| | GIFT | | 4.7806 | 12.6612 | 12.6612 | 21.7721 | 21.8162 | 25.0934 |
| | NURBS | | 4.7787 | 12.6303 | 12.6303 | 21.7177 | 21.7179 | 25.0316 |
| | RHT | 972 | 4.7787 | 12.6308 | 12.6308 | 21.7193 | 21.7216 | 25.0357 |
| | GIFT | | 4.7787 | 12.631 | 12.631 | 21.7196 | 21.7222 | 25.0364 |
| | NURBS2P-L1 [128] | | 4.7633 | 12.5514 | 12.7197 | 21.7338 | 21.7397 | 25.2435 |
| | NURBS2P-L2 [128] | | 4.7628 | 12.5474 | 12.6319 | 21.5869 | 21.7124 | 24.9586 |
| | NURBS2P-L3 [128] | | 4.7628 | 12.5472 | 12.6317 | 21.5856 | 21.7119 | 24.9564 |
| | NURBS5P [128] | | 4.7775 | 12.6279 | 12.628 | 21.7137 | 21.7166 | 25.0301 |
| | $\lambda_{\text{ext}}$ [132] | – | 4.777 | 12.62 | 12.62 | 21.687 | 21.687 | 24.994 |

Table 5.2: Comparisons of normalized frequency $\lambda_N$ for fully clamped circular plates.

| $h/r$ | Method | Dof | Mode number | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 |
| 0.1 | NURBS | | 10.9187 | 23.6745 | 23.6745 | 38.7526 | 86.9575 | 89.5851 |
| | RHT | 108 | 10.1924 | 25.0979 | 25.0979 | 40.1403 | 75.0977 | 77.4168 |
| | GIFT | | 10.4267 | 25.622 | 25.622 | 40.9466 | 80.0458 | 82.4535 |
| | NURBS | | 9.9535 | 20.3047 | 20.3047 | 32.5611 | 33.2277 | 37.5821 |
| | RHT | 300 | 9.9611 | 20.4369 | 20.4369 | 32.4949 | 32.8485 | 36.8225 |
| | GIFT | | 9.9671 | 20.5024 | 20.5024 | 32.5965 | 32.9663 | 36.9359 |
| | NURBS | | 9.9439 | 20.1884 | 20.1884 | 32.2325 | 32.2346 | 36.5146 |
| | RHT | 972 | 9.9447 | 20.193 | 20.193 | 32.2422 | 32.257 | 36.537 |
| | GIFT | | 9.945 | 20.1942 | 20.1942 | 32.244 | 32.2602 | 36.5401 |
| | NURBS2P-L1 [128] | | 9.9314 | 20.084 | 22.1907 | 32.5233 | 33.3679 | 41.0644 |
| | NURBS2P-L2 [128] | | 9.9105 | 20.0594 | 20.1901 | 32.0676 | 32.204 | 36.4285 |
| | NURBS2P-L3 [128] | | 9.9103 | 20.0591 | 20.183 | 32.0476 | 32.202 | 36.3988 |
| | NURBS5P [128] | | 9.9438 | 20.1876 | 20.1878 | 32.232 | 32.2329 | 36.5122 |
| | $\lambda_{\text{ext}}$ [132] | – | 9.941 | 20.232 | 20.232 | 32.406 | 32.406 | 36.479 |
| 0.2 | NURBS | | 9.7537 | 19.324 | 19.324 | 29.807 | 45.7146 | 48.6745 |
| | RHT | 108 | 9.3809 | 19.2785 | 19.2785 | 29.5878 | 40.3199 | 43.1369 |
| | GIFT | | 9.5038 | 19.57 | 19.57 | 29.9821 | 42.4817 | 45.3109 |
| | NURBS | | 9.2523 | 17.8187 | 17.8187 | 27.1355 | 27.3255 | 30.5883 |
| | RHT | 300 | 9.2554 | 17.8557 | 17.8557 | 27.1676 | 27.2129 | 30.4077 |
| | GIFT | | 9.2574 | 17.8724 | 17.8724 | 27.196 | 27.2432 | 30.4398 |
| | NURBS | | 9.25 | 17.7891 | 17.7891 | 27.0546 | 27.0551 | 30.2838 |
| | RHT | | 9.2502 | 17.7905 | 17.7905 | 27.0575 | 27.062 | 30.291 |
| | GIFT | | 9.2503 | 17.7908 | 17.7908 | 27.058 | 27.063 | 30.2919 |
| | DSG3 [129] | – | 9.3012 | 18.0038 | 18.0098 | 27.601 | 27.6082 | 30.9865 |
| | ES–DSG3 [129] | – | 9.2527 | 17.8372 | 17.8428 | 27.2344 | 27.2391 | 30.5173 |
| | NS-DSG3 [130] | – | 9.2789 | 17.9195 | 17.9366 | 27.4301 | 27.4531 | 30.7906 |
| | ANS4 [131] | – | 9.2605 | 17.9469 | 17.9469 | 27.0345 | 27.6566 | 30.3221 |
| | $\lambda_{\text{ext}}$ [132] | – | 9.24 | 17.834 | 17.834 | 27.214 | 27.214 | 30.211 |

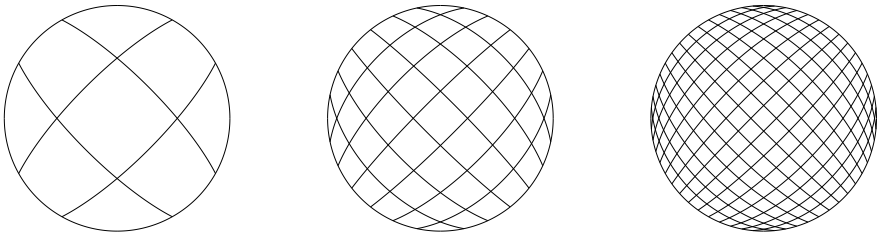



(a) Quadratic NURBS



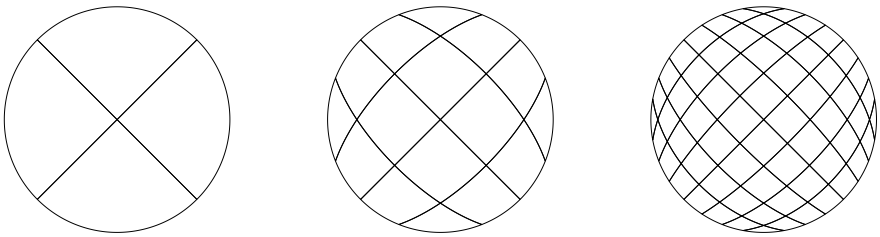

(b) Cubic NURBS and RHT

Figure 5.12: Geometry and intial control points for disk generated by GIFT(a) and IGA(b).



(a) 108 dofs (b) 300 dofs (c) 972 dofs

Figure 5.13: Mesh of solution field with NURBS ($p = 3, q = 3$).



(a) 108 dofs (b) 300 dofs (c) 972 dofs

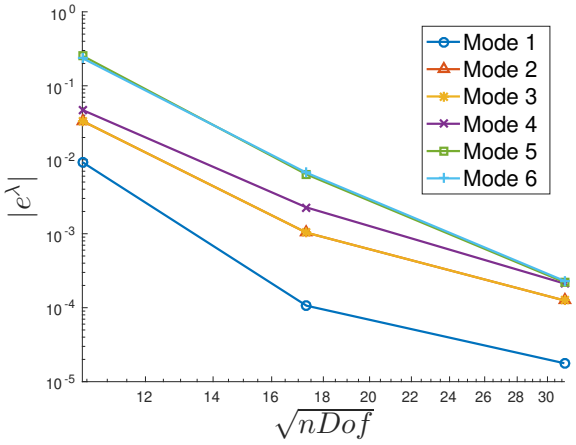
Figure 5.14: Mesh of solution field with PHT ($p = 3, q = 3$) and RHT ($p = 3, q = 3$).

Fig.5.22, $[\lambda_{min}, \lambda_{max}] = [0.15, 0.20]$, where the range is set by a window marked by red dash lines). The reason to choose the range $[0.15, 0.20]$ is that the frequencies in this range are relatively high, so that the resemble modal shapes (not corresponding modal shapes) occur more frequently, which is able to demonstrate the advantage the MAC method. Define the $\boldsymbol{\mathcal{I}}$, i.e.,

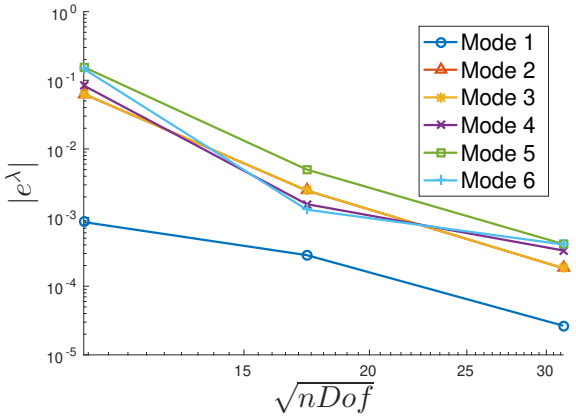
$$\boldsymbol{\mathcal{I}}(n) = \begin{cases} i, & n = 1 \\ \{i, \ldots, i + n - 1\}, & n > 1, \end{cases} \tag{5.6.71}$$

where $n$ is the multiplicity of mode $i$. The adaptive process is based on the Algorithm 2, but there is a slight difference. In this example, the adaptivity is not conducted by sweeping modes from low to high. Instead, at each step of adaptivity, we pick up the mode(s) $\boldsymbol{\mathcal{I}}$ with the maximum of $\delta^{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}}$ to deliver the adaptivity, where $\delta_{\boldsymbol{\mathcal{I}}}$ reads

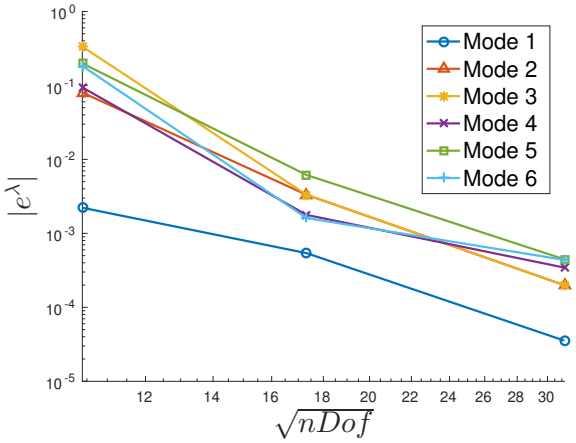
$$\delta^{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}(n)} = \begin{cases} \delta^{\boldsymbol{\phi}}_i & \text{in Eq.(5.4.39)}, \quad n = 1 \\ \delta_{\boldsymbol{\phi}} & \text{in Eq.(5.5.67)}, \quad n > 1. \end{cases} \tag{5.6.72}$$




(a) NURBS




(b) RHT




(c) GIFT

Figure 5.15: Convergence of absolute value of relative error for natural frequency $|e^\lambda|$ computed by NURBS, RHT and GIFT for vibration of the simply supported circular plate with $h/r = 0.1$. The nDof is short for number of degree of freedom.

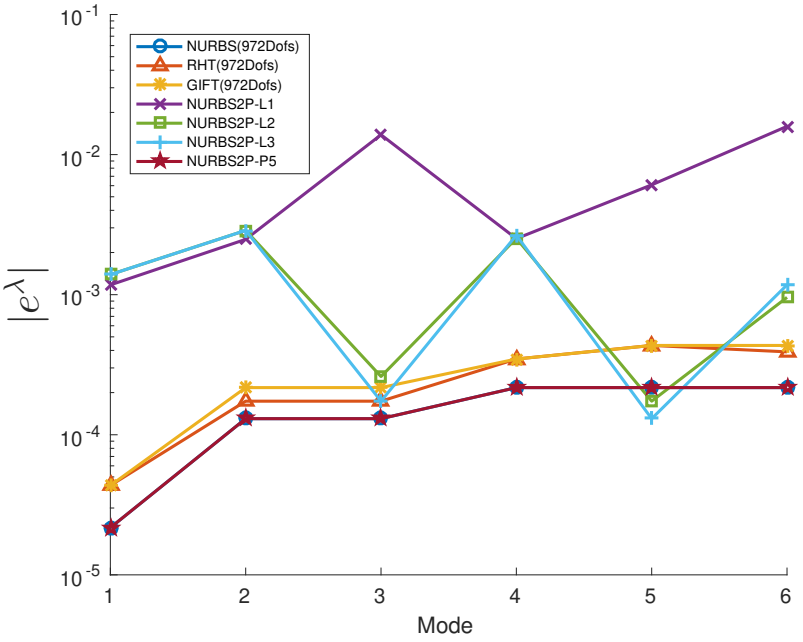



Figure 5.16: The comparison of absolute value of relative error for natural frequency $|e^{\lambda}|$ for modes 1 to mode 6 obtained by different methods listed Tab.5.1 for vibration of the simply supported circular plate with $h/r = 0.1$.

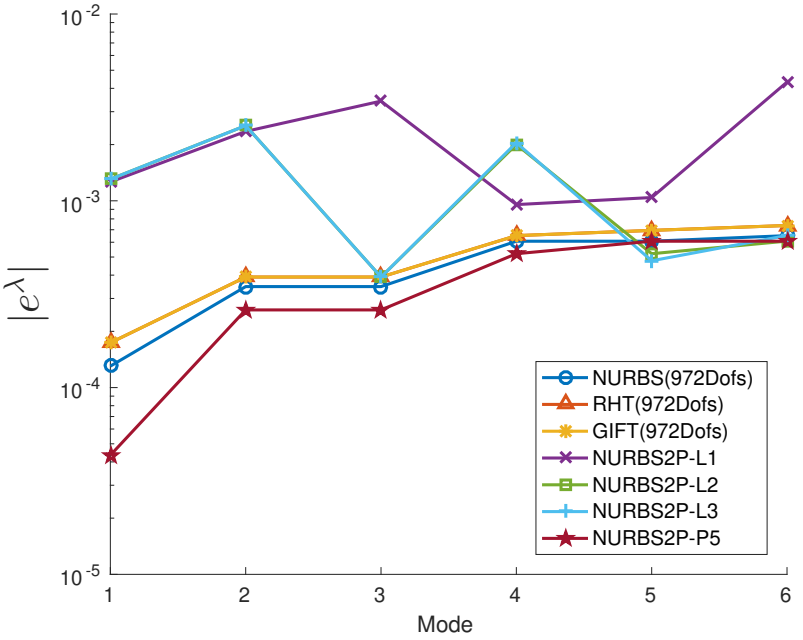



Figure 5.17: The comparison of absolute value of relative error for natural frequency $|e^{\lambda}|$ for modes 1 to mode 6 obtained by different methods listed Tab.5.1 for vibration of the simply supported circular plate with $h/r = 0.2$.

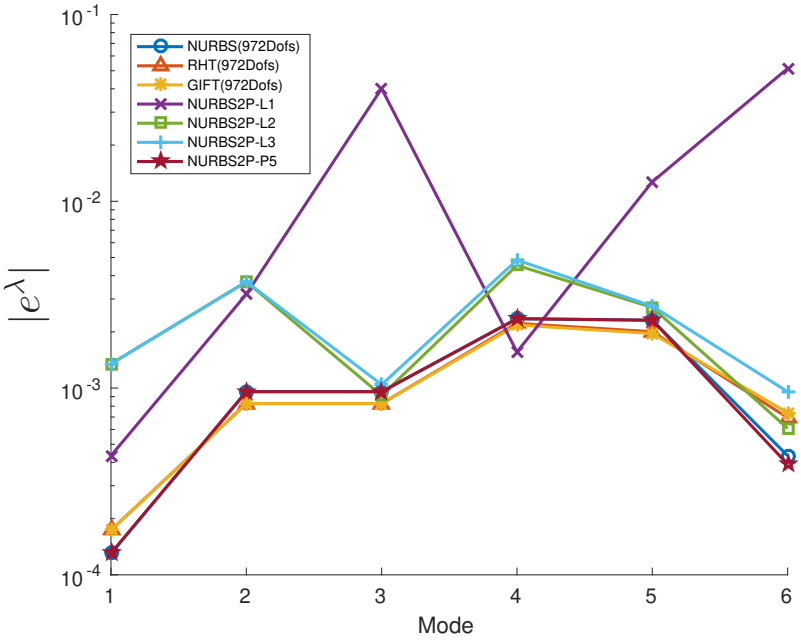



Figure 5.18: The comparison of absolute value of relative error for natural frequency $|e^{\lambda}|$ for modes 1 to mode 6 obtained by different methods listed Tab.5.2 for vibration of the clamped circular plate with $h/r = 0.1$.

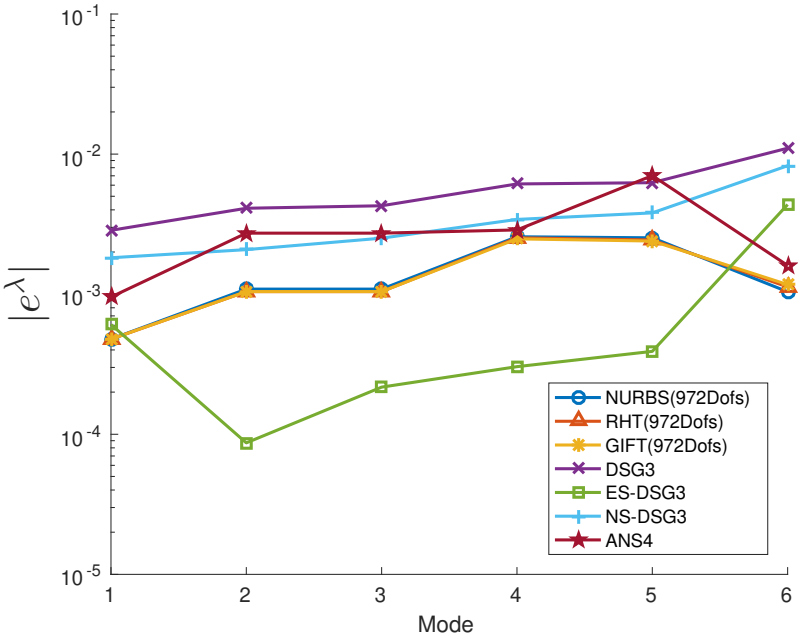



Figure 5.19: The comparison of absolute value of relative error for natural frequency $|e^{\lambda}|$ for modes 1 to mode 6 obtained by different methods listed Tab.5.2 for vibration of the clamped circular plate with $h/r = 0.2$.

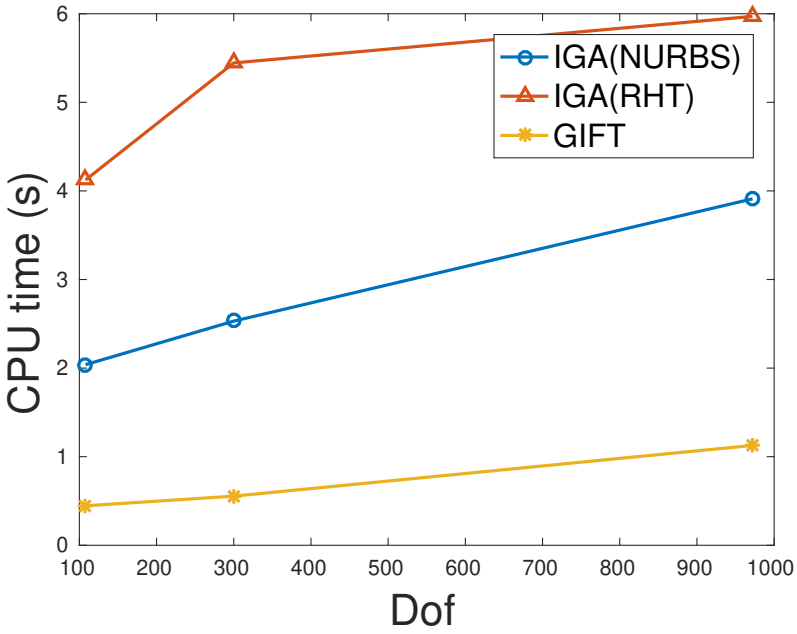



Figure 5.20: The comparison of CPU time by IGA(NUBRS), IGA(RHT) and GIFT method for vibration of the clamped circular plate with $h/r = 0.2$.

Accordingly, the error estimator of frequencies is referred as

$$|e^{\lambda}_{\boldsymbol{\mathcal{I}}(n)}| = \begin{cases} |e^{\lambda}_i| & \text{in Eq.(5.4.39)}, \quad n = 1 \\ |e_{\lambda}| & \text{in Eq.(5.5.67)}, \quad n > 1. \end{cases} \tag{5.6.73}$$

In this case, two schemes, FEC and MAC, proposed in Section 5.5.2 are compared in order to investigate theirs effects on the adaptivity. As it can be seen in Fig.5.23(a),(b), it is clear that MAC method has a better convergence than FEC method. The reason can be obtained by tracing the results in Tab.5.3 and Tab.5.4. To be specific, the local adaptivity is driven by the error estimation of mode shapes, FEC can not always guarantee to locate the right corresponding modal vector on refined mesh (see Tab.5.3 that $\boldsymbol{\phi}^h_{\boldsymbol{\mathcal{I}}}$ and $\tilde{\boldsymbol{\phi}}_{\tilde{\boldsymbol{\mathcal{I}}}}$ are not consistent until adaptive step 13). Here, $\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}}$ is expressed as

$$\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}(n)} = \begin{cases} \boldsymbol{\phi}_i & \text{eigenvector}, \quad n = 1 \\ \boldsymbol{\varphi} & \text{vector in Eq.(5.5.54)}, \quad n > 1. \end{cases} \tag{5.6.74}$$

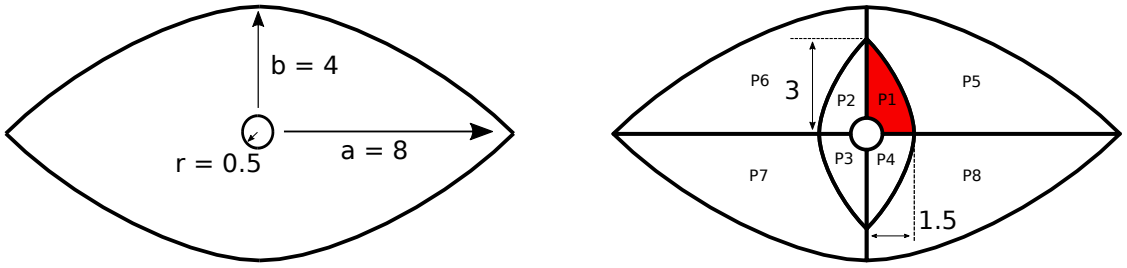
Therefore, it leads to the inefficient adaptive mesh, and furthermore results in the divergence of error estimator $\delta^{\phi}_{\boldsymbol{\mathcal{I}}}$, as shown in Fig.5.23(b). Only when FEC scheme

is able to identify the related mode correctly (from the step 13 and forwards in Tab.5.3), the error estimators are just convergent accordingly. In contrast, MAC method can find the associated mode accurately at very early stage of adaptivity (at around 6th step displayed in Tab.5.4 and Fig.5.23(c)). Therefore, for the given accuracy such as $\left|e^{\lambda}_{\boldsymbol{\mathcal{I}}}\right| \leqslant 10^{-4}$ and $\delta^{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}} \leqslant 10^{-2}$, MAC method is more efficient than FEC method. As the implements of both MAC method and FEC method are very cheap, the comparison of computation time is not presented here.

As the window expands to cover $\lambda^h_i \in [0.1, 0.2]$ in Fig.5.24, the advantage of using MAC is not so noticeable that good convergence is achieved by both MAC and FEC, as presented in Fig.5.25(a),(b). This is because that the modal shapes at low frequency modes are often distinct. If the adaptivity starts from low frequency modes, it is easy to locate the corresponding mode even by a FEM scheme. Thus, the precisely adaptive refinement in low modes will help to accurately locate modal correspondence for high frequency modes. Regardless of that, the MAC will be the preference to the remaining computations as it will not make mistakes on modal resemblance recognition in any case.

Note that the final meshes in both Tab.5.3 (at step 28) and Tab.5.4 (at step 24) are close to uniform meshes. This is because the modes of interest are with high frequencies (see Fig.5.23), the structural vibrations are normally global. If modes of interest are low (as in Fig.5.26(a)), the refinement will localize around the patch with soft material and the hole (see Fig.5.26(d)). While as the modes of interest become higher (as shown in Fig.5.26(b),(c)), with the same number of elements, the adaptive refinements will get closer to uniform refinements, as in Fig.5.26(e),(f), and the error estimators $\left|e^{\lambda}_{\boldsymbol{\mathcal{I}}}\right|$ and $\delta^{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}}$ are larger.

In Tab 5.3 and Tab 5.4, the representative modal shapes $\boldsymbol{\phi}^h_{\boldsymbol{\mathcal{I}}}$ and $\tilde{\boldsymbol{\phi}}^h_{\boldsymbol{\mathcal{I}}}$ are presented. It is worth noticing that the targeted modal vector $\boldsymbol{\phi}^h_{\boldsymbol{\mathcal{I}}}$ at each step is not obliged to be same, since the targeted modal vector only means this modal vector produces largest relative error $\delta^{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}}$ compared to other mode shapes. It is not essential to be the same mode at each step. For example, mode shapes $\boldsymbol{\phi}^h_{\boldsymbol{\mathcal{I}}}$ at step 13 and step 28 are different, as shown in Tab 5.3. Also, mode shapes $\boldsymbol{\phi}^h_{\boldsymbol{\mathcal{I}}}$ at step 13 and




(a) Geometry of structure (b) Initial discretization of patches

Figure 5.21: The simply supported eye shape with a hole is discretized by 8 patches, with the material parameters that $E_1 = 0.03,\ \rho_1 = 0.7.\ E_i = 1, \rho_i = 1, (i = 2 \ldots 8)$. The geometry is represented by NURBS ($p = 2,\ q = 2$), and the solution field is approximated by PHT splines ($p = 3,\ q = 3$).

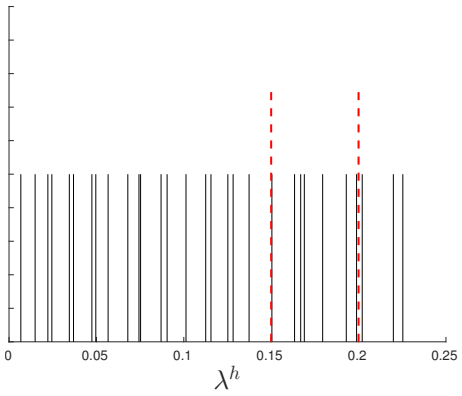
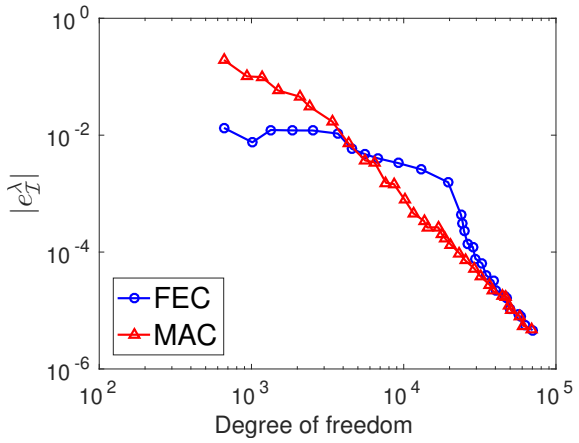



Figure 5.22: Frequencies of interest in the window with interval $[0.15, 0.2]$.

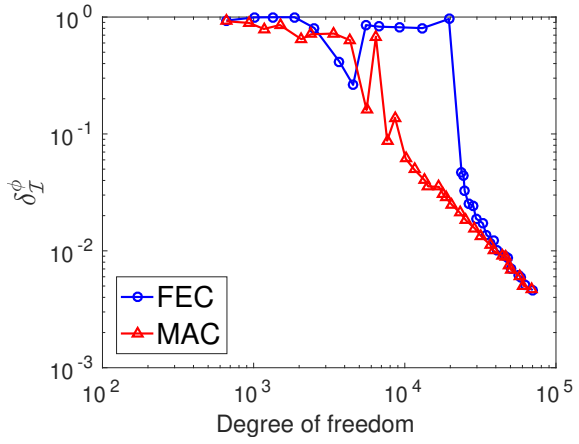
step 24 are different, as presented in Tab 5.4. As long as the modal vectors obtained between the coarse mesh $\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}}^h$ and refined mesh $\tilde{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}}^h$ are resemble, the algorithm can be validated to be stable. As it can be seen in Fig.5.23(c), the close correspondence between $\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}}^h$ and $\tilde{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}}^h$ starts to be established (MAC value is approximate to 1) since step 11 for MAC method and step 13 for FEC method. It is also demonstrated in Tab 5.3 and Tab 5.4, $\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}}^h$ and $\tilde{\boldsymbol{\phi}}_{\boldsymbol{\mathcal{I}}}^h$ are not closely resemble until step 13.

### 5.6.3 Heterogeneous square plate with holes

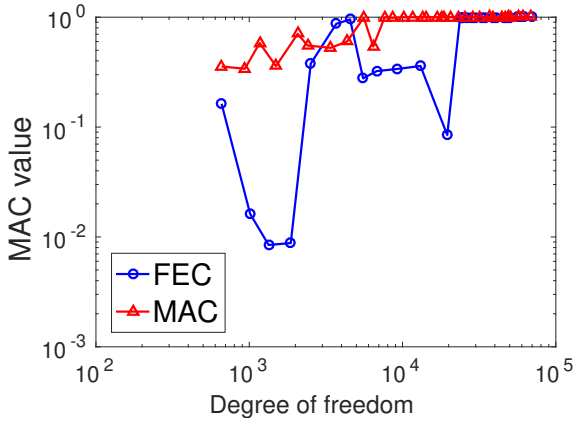
As shown in Fig.5.27, the geometry and discretization of a plate with 9 holes are illustrated. The softer patches are marked with red color. The density is allocated




(a)




(b)




(c)

Figure 5.23: Comparisons between MAC and FEC methods in a frequency range [0.15,0.2] in respect of (a) $|e_{\boldsymbol{\mathcal{I}}}^{\lambda}|$, (b) $\delta_{\boldsymbol{\mathcal{I}}}^{\phi}$, (c) MAC value. Refinement level $L_e$ is chosen to be 1.

Table 5.3: Targeted mode shapes $\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}}^{h}$ at coarse mesh and related mode shapes $\tilde{\boldsymbol{\phi}}_{\tilde{\boldsymbol{\mathcal{I}}}}$ over refined mesh, and the adaptive refinement at different steps obtained through FEC.

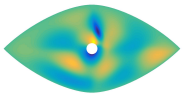
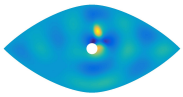
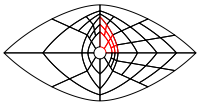
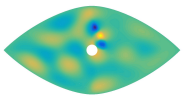
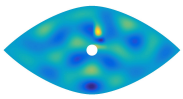
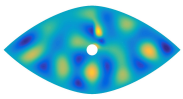
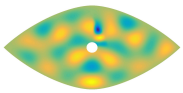
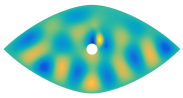
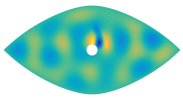
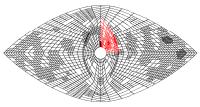
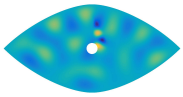
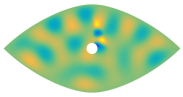
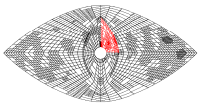
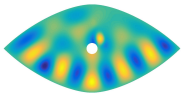
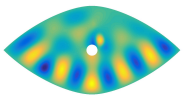
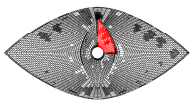
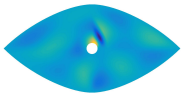
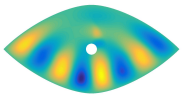
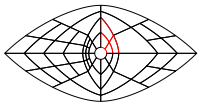
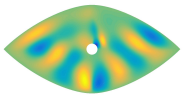
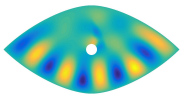
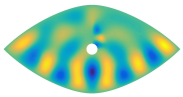
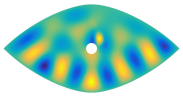
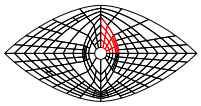
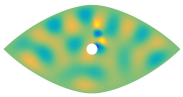
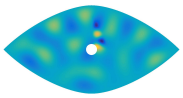
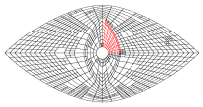
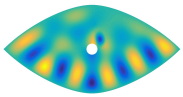
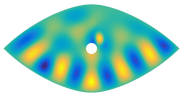
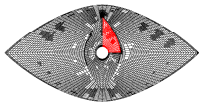
| **Step** | $\boldsymbol{\phi}_{\boldsymbol{\mathcal{I}}}^{h}$ | $\tilde{\boldsymbol{\phi}}_{\tilde{\boldsymbol{\mathcal{I}}}}$ | **Adaptive mesh** |
|---|---|---|---|
| 2 |  |  |  |
| 4 |  |  |  |
| 6 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 28 |  |  |  |

Table 5.4: Targeted mode shapes $\boldsymbol{\phi}_{\mathcal{I}}^{h}$ at coarse mesh and related mode shapes $\tilde{\boldsymbol{\phi}}_{\tilde{\mathcal{I}}}$ over refined mesh, and the adaptive refinement at different steps acquired through MAC.

| **Step** | $\boldsymbol{\phi}_{\mathcal{I}}^{h}$ | $\tilde{\boldsymbol{\phi}}_{\tilde{\mathcal{I}}}$ | **Adaptive mesh** |
|---|---|---|---|
| 2 |  |  |  |
| 3 |  |  |  |
| 6 |  |  |  |
| 13 |  |  |  |
| 24 |  |  |  |

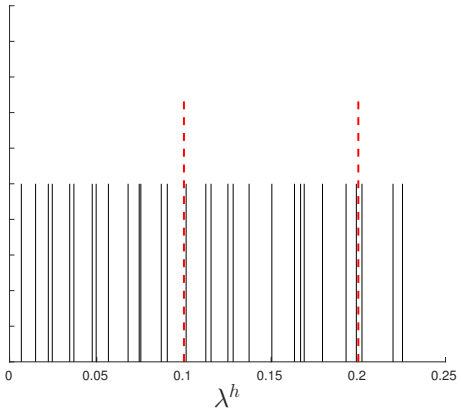
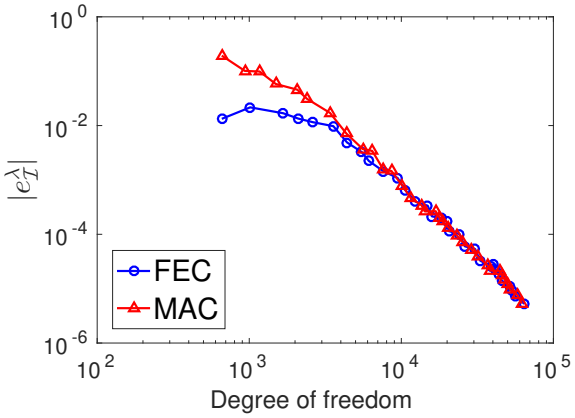
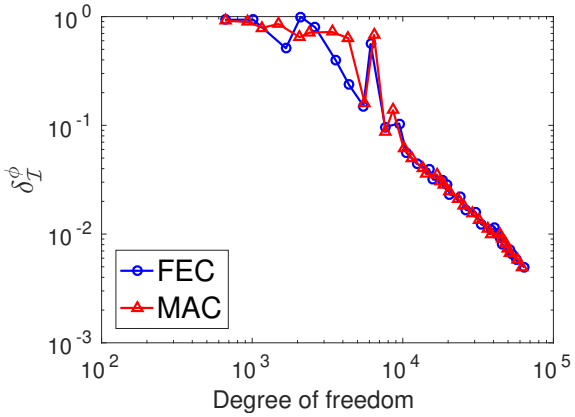


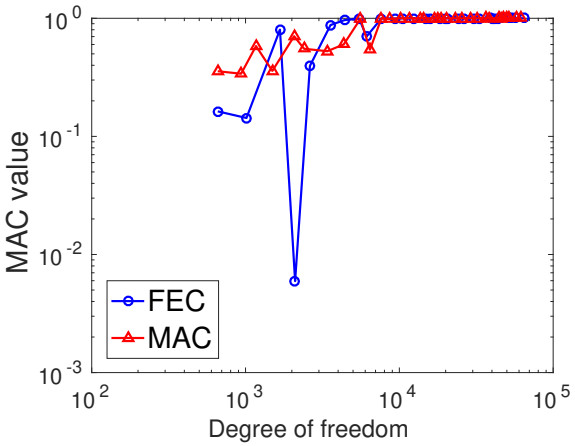

Figure 5.24: Frequencies of interest in the window with interval $[0.1, 0.2]$.




(a)

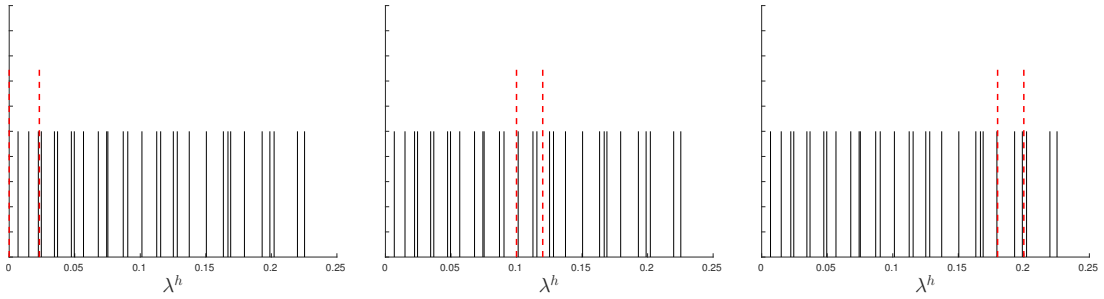



(b)

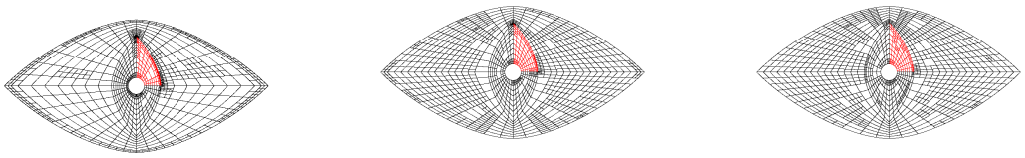



(c)

Figure 5.25: Comparisons between MAC and FEC methods in a frequency band [0.1,0.2] in terms of (a) $|e^{\lambda}_{\mathcal{I}}|$, (b) $\delta^{\phi}_{\mathcal{I}}$, (c) MAC value.




(a) [0, 0.023] (b) [0.1, 0.12] (c) [0.18 0.2]



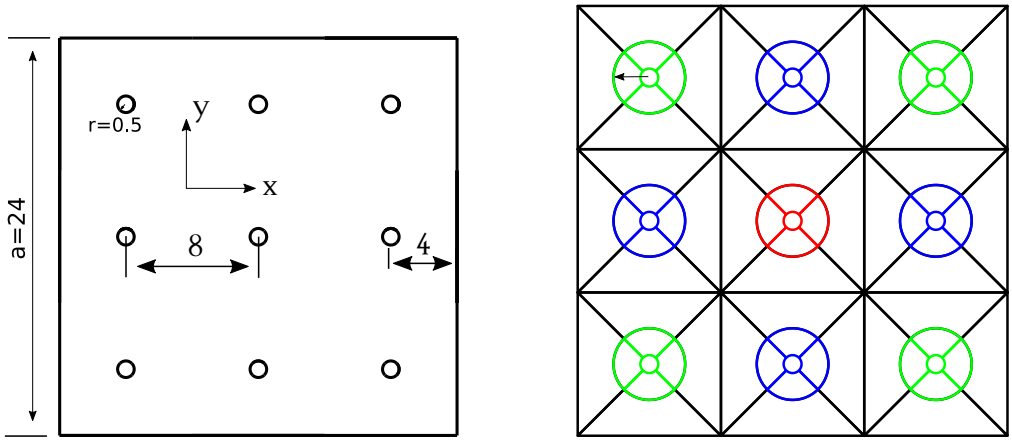
(d) Refinement at $[0, 0.02]$, with $\left|e^{\lambda}_{\mathcal{I}}\right| = 2 \times 10^{-5}$, $\delta^{\phi}_{\mathcal{I}} = 0.0096$

(e) Refinement at [0.1, 0.12], with $\left|e^{\lambda}_{\mathcal{I}}\right| = 1.2 \times 10^{-4}$, $\delta^{\phi}_{\mathcal{I}} = 0.024$

(f) Refinement at [0.18 0.20], with $\left|e^{\lambda}_{\mathcal{I}}\right| = 3.2 \times 10^{-4}$, $\delta^{\phi}_{\mathcal{I}} = 0.042$

Figure 5.26: The Comparison of adaptive refinements among different intervals of frequencies of interest (a)-(c) based on the MAC method. The numbers of elements for the meshes in (d),(e) and (f) are all 1150.

as $\rho = 1$ and thickness-span ratio is set as $h/a = 0.1$. The edges of square plate are imposed by simply supported boundary conditions, the edges of holes are kept free. As discussed in Section 5.5, using Algorithm 2 by sweeping modes, the adaptivity for the frequencies of interest marked in Fig.5.28 is guided applying the GIFT method combined with MAC. It denotes that the initial mesh of adaptivity for mode(s) $(\boldsymbol{\mathcal{I}} + 1)$, or $\boldsymbol{\mathcal{I}} + n$ for $n$ multiple modes, follows the mesh ending with the phase of which adaptivity for mode $\boldsymbol{\mathcal{I}}$ (defined in Eq.(5.6.71)) is completed. Due to the symmetric geometry and material characteristics, it is not difficult to find in Fig.5.29 that double modes arise at modes (2,3), modes (7,8) and modes (10,11). Note that although mode 5 and mode 6 are very close, they are not double modes. Besides, it is observed that global modal shapes dominate from 1st to 3rd mode which leads to the nearly global refinement in adaptivity, as shown from Fig.5.30(a) to Fig.5.30(c). Afterwards, as it can be seen in Fig.5.29(d)-(i), local vibration gradually appears, which results in local refinements at areas around the holes. The vibrations at 10th and 11th modes distribute like an orthogonal crossing on the plate (see Fig.5.29(j,k)), and then adaptive refinement follows horizontally and vertically at conjunctions of multiple patches in Fig.5.30(f). This denotes that the accuracy of these $C^0$ coupling fields is not good enough and expected to be enhanced. According to the results shown in Fig.5.31, error estimators $|e^{\lambda}_{\boldsymbol{\mathcal{I}}}|$ (defined in Eq.(5.6.72)) and $\delta^{\phi}_{\boldsymbol{\mathcal{I}}}$ (defined in Eq.(5.6.73)) both perform the excellent convergence from low to high modes. It is worth noting that the plot of convergence rate for some modes (such as 4th, 6th and 9th modes) is a point respectively instead of lines. That is because the $|e^{\lambda}_{\boldsymbol{\mathcal{I}}}|$ and $\delta^{\phi}_{\boldsymbol{\mathcal{I}}}$ at these modes have met the threshold defined in Eq.(5.4.49) since the starting of the adaptivity process. As a result, not any further discretization is needed for these modes. A numerical test is carried out to validate the proper option on the level of $h-$refinements for the refined solution. As presented in Fig.5.32, the convergence obtained by different level of $h-$refinements is nearly consistent, implying the chosen level of refinement $L_e = 1$ is sufficient to for the compute of error estimators. Finally, an improved convergent rate achieved by local adaptive refinement is depicted in Fig.5.33, compared to the global uniform PHT refinement obtained within GIFT




(a) Geometry of structure (b) Discretization of patches

Figure 5.27: The simply supported square plate with holes is discretized by 72 patches with the Young's modulus $E_{\text{red}} = 0.05$, $E_{\text{blue}} = 0.08$, $E_{\text{green}} = 0.12$, $E_{\text{black}} = 1$. The geometry is represented by NURBS ($p = 2, q = 2$), and the solution field is approximated by PHT splines ($p = 3, q = 3$).

scheme.

## 5.7 Conclusion

In this chapter, a strategy of local adaptivity for the structural vibrations of Reissner-Mindlin plate was presented. Within the context of the GIFT framework, hlone may make use of the geometrical descriptors given by CAD directly, and independently apply PHT splines for analysis, which allows for local refinement to be performed

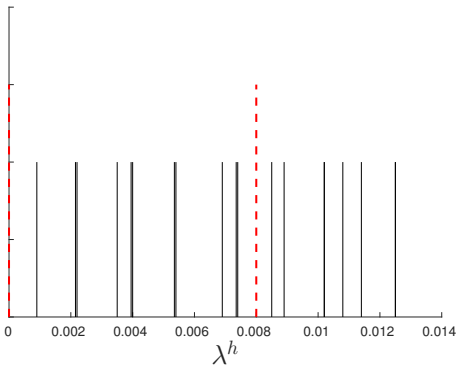
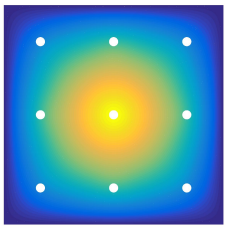
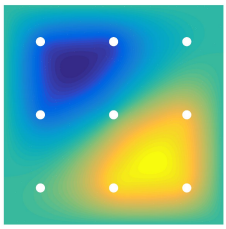


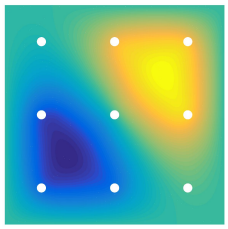

Figure 5.28: Frequencies of interest in the window with interval $[0, 0.008]$.



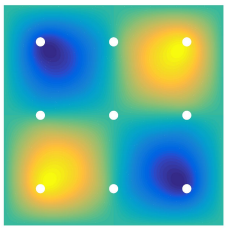
(a) 1st mode, $\lambda_N^1 = 1$



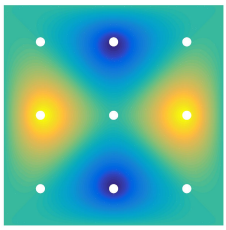
(b) 2nd mode, $\lambda_N^2 = 2.498$



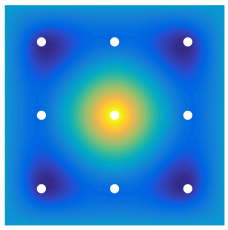
(c) 3rd mode, $\lambda_N^3 = 2.498$



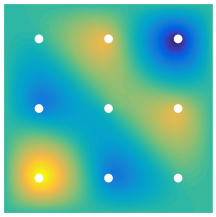
(d) 4th mode, $\lambda_N^4 = 4.013$



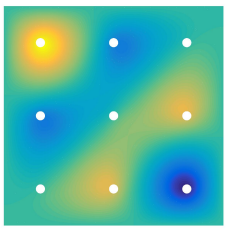
(e) 5th mode, $\lambda_N^5 = 4.616$



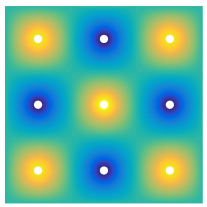
(f) 6th mode, $\lambda_N^6 = 4.618$



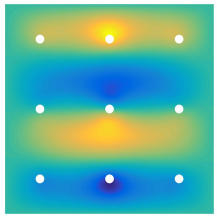
(g) 7th mode, $\lambda_N^7 = 6.166$

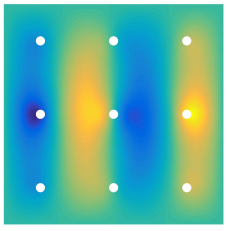


(h) 8th mode, $\lambda_N^8 = 6.166$



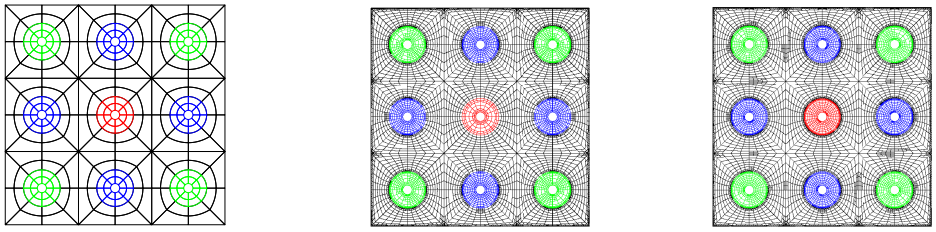
(i) 9th mode, $\lambda_N^9 = 7.874$



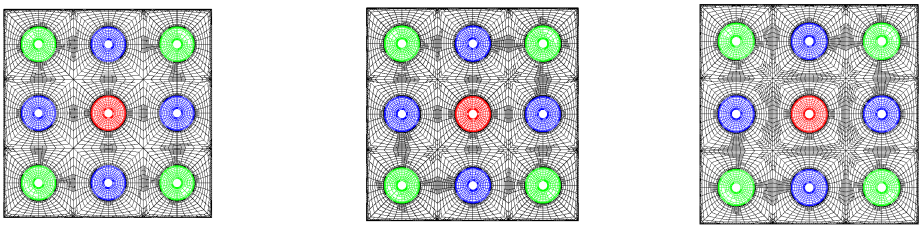


(j) 10th mode, $\lambda_N^{10} = 8.383$ (k) 11th mode, $\lambda_N^{11} = 8.383$

Figure 5.29: Vibration of mode shapes for structure of plate with 9 holes. The bright color denotes the vibration towards the reader and the dark color indicates the vibration backwards to the reader.



(a) Initial (b) 1th mode (c) 2-3, 4th modes



(d) 5, 6th modes (e) 7-8, 9th modes (f) 10-11th modes

Figure 5.30: Adaptive refinement process of 1-11th modes for vibration of the plate with holes, with refinement level $L_e = 1$.

without the limitation occuring when using tensor-product-based shape functions. The adaptivity algorithm is fully automatised, and relies on a hierarchical posteriori error estimation strategy that makes the best of the PHT-spline element subdivision capabilities. In the frequency domain, adaptivity is performed in a mode-by-mode manner, sweeping from lower to higher frequencies, and identifying the correspondence between coarse and fine mesh solutions using a MAC-type approach. As shown in the numerical section, super convergent solutions are obtained (i.e. faster convergence than that observed when using a uniform $h-$refinement), in particular when the problem exhibits local features that require a local refinement. In the next chapter, it aims to extend the adaptivity strategy to the application of time-domain dynamics.

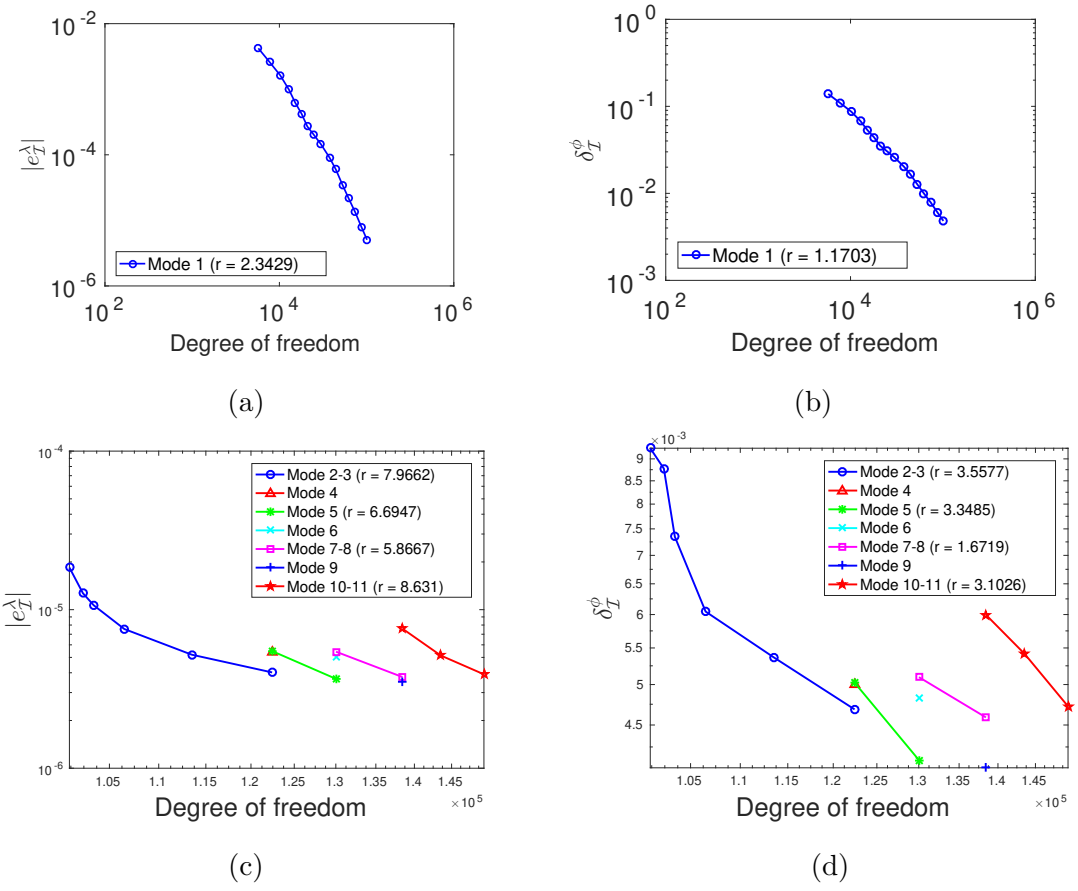
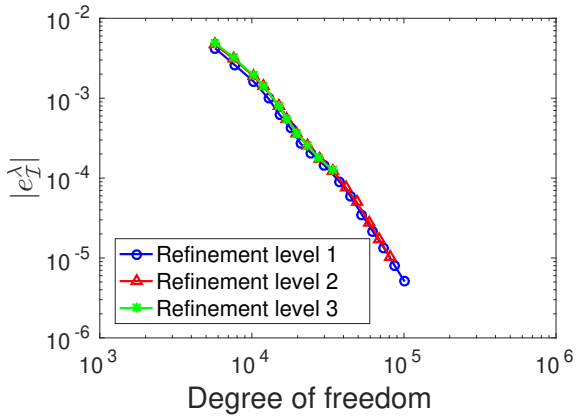
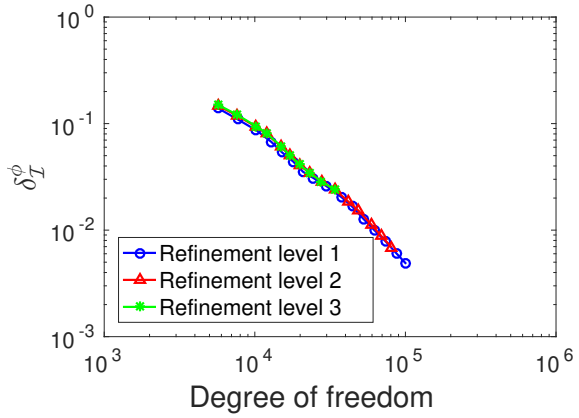



Figure 5.31: The study of convergence of error estimators (a) $|e^{\lambda}_{\mathcal{I}}|$ mode 1, (b) $\delta^{\phi}_{\mathcal{I}}$ mode 1, (c) $|e^{\lambda}_{\mathcal{I}}|$ modes 2 to 11, (d) $\delta^{\phi}_{\mathcal{I}}$ modes 2 to 11 of the plate with holes.

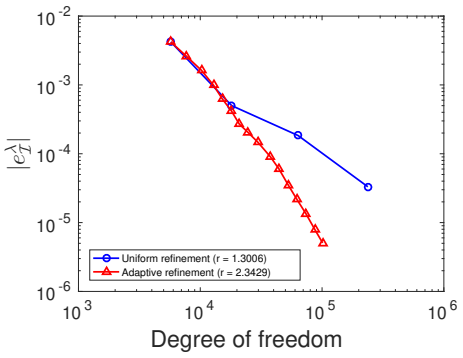



(a) eigenvalue

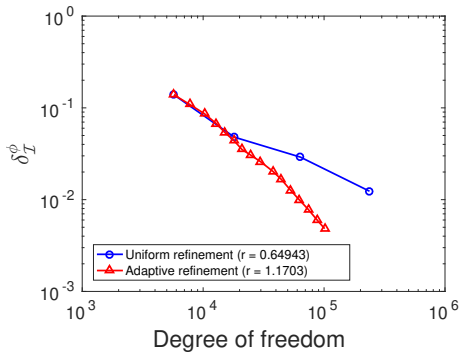



(b) eigenvector

Figure 5.32: The comparison of error estimators (a) $|e_{\boldsymbol{\mathcal{I}}}^{\lambda}|$ and (b) $\delta_{\boldsymbol{\mathcal{I}}}^{\phi}$ at 1st mode obtained by different refinement levels: $L_e = 1, L_e = 2, L_e = 3$.




(a)




(b)

Figure 5.33: The Comparison of error estimators (a) $|e_{\boldsymbol{\mathcal{I}}}^{\lambda}|$ and (b) $\delta_{\boldsymbol{\mathcal{I}}}^{\phi}$ at 1st mode obtained by adaptive refinement and uniform refinement.

# Chapter 6

# Adaptivity for transient dynamics

**Statement**: It is acknowledged that this chapter will be organized as a paper to submit to a journal in the future. However, since the algorithm is still under improvement, the paper is in preparation presently. Therefore, it is sure that the contents of the coming paper will have a lot of difference with those in this chapter. In spite of that, the chapter has been still written by the author.

## 6.1 Introduction

In the previous chapter, an adaptivity algorithm in the framework of GIFT was proposed to study vibration (frequency-domain problem). In this chapter, it is intended to be extended to address transient dynamics (time-domain problem). Transient dynamics, also understood as the second-order hyperbolic partial differential equation (PDE), is a time-domain problem. In terms of this problem, the discretization is generally divided into two portions, i.e., spatial and temporal discretizations. Regarding time discretization, the popular methods primarily contains integration schemes (such as continuous Galerkin method [133, 134] and Discontinuous Galerkin (DG) method [135, 136]), and difference schemes (including the family of Newmark methods [137, 138, 139], the forward (explicit) Euler method [140], backward (implicit) Euler method [3] and the Crank-Nicolson method [141]. Since IGA is a very good tool to replace the FEM for spacial discretization, it can be seamlessly com-

bined with the temporal discretization approaches mentioned above, to develop a full *space-time* algorithm. Langer et al. [142] proposed a novel stabilized space-time IGA methodology to solve the parabolic PDEs (e.g. heat conduction and diffusion in engineering fields) with fixed and moving spacial domains. This method was then extended to a parallel time-multipatch Discontinuous Galerkin IGA-multigrid algorithm, based on the parallel time integration approaches (see [143]). For an implicit time-stepping method to solve the wave propagation problem, at each time step, it is required to invert a large sparse mass matrix to get the solution, which is a huge computational consuming. In order to deal with this difficulty, Chan et al. [144] introduced an efficient approach in the framework of multi-patch DG Isogeometric analysis, by using a weighted adaptive tensor product structure to represent the matrix inversion, which was proved to be accurate and stable. Similarly, for reducing storage and cost of computation resulting from the excessive number of quadrature points, Auricchio et al. [145] took advantage of the smoothness and high continuity of IGA basis functions, to develop a steady, high-order and effective collocation method with only a minimal number of quadrature points, to analyze the explicit dynamic problem. In order to decrease the dispersion errors of finite element upshots in 1D and 2D transient wave propagation problem, Wen et al. [146] introduced variable parameters in mass and stiffness matrices assembled by the uniform quadratic B-spline basis functions. Yavari et al. [147] made use of NURBS to exactly describe the geometry of a heterogeneous bar with an non-uniform cross section, and studied the 1D wave propagation inside subjected to a strike load.

Presently, the error estimation and adaptivity strategies have been well developed for linear statics (i.e. [148, 149]) and heat transfer problem (e.g. [150, 151, 152]), nevertheless, for transient dynamics, the techniques are still under construction. It is a challenge since the error transfers with the wave propagation, which makes it less recognizable and predictable, and consequently it often leads to the poor quality of the estimated error bounds [153]. Specifically, whether explicit method or implicit method is employed, the information at forward step derives from that at previous step. Hence it is impossible to predict the solution at termination

from the beginning, and as result, the estimation of error bounds is not so reliable. The energy-based error estimation is quite direct in the static problems, as the evaluation of potential energy error is closely associated with the residual. As to transient dynamics, besides the potential energy, the kinetic energy and the dissipated energy (for the system with damping [104]), are also computed for error measurements. These energy error estimators are usually calculated, based upon a post-processing approach [154, 87] or a Taylor expansion [155], in the framework of Newmark method. Particularly, Romero et al. [154] built both a priori error bounds and a posteriori error bounds for their proposed error estimation that is able to catch the order of magnitude for error in non-smooth problem, which was difficult to achieve. It should be referred the work by Cavin et al.[47], who initiated a unidirectional multi-level space-time adaptive method for 1D transient dynamics. The multi-level process was posed by a hierarchical refinement based on energy-like a posteriori error estimation, where only the elements in which the errors did not reach the prescribed accuracy were required to be refined and computed at next step, while the rest elements were not. This method is efficient to some extent because the number of elements requested to be calculated keep decreasing roughly. Then, Biboulet et al. [156, 157, 158] extended this strategy to be applicable in the 2D and 3D elastoplastic dynamics in the framework of multigrid.

Although the computation for global energy norm of the error is cheap, it is not adequate in practice. Instead, engineers are normally interested in a more physically meaningful scalar variable, such as an average value of the solution over a certain local domain (Verdugo et al. [159] proposed a timeline-dependent quantity recently). It is the so-called *quantity of interest* (QoI). For the sake of obtaining the QoI, it is required to solve an auxiliary problem, namely, the *adjoint* or *dual* problem. Due to the *Galerkin orthogonality*, when using Galerkin method to compute the dual weighted residual for time-dependent problems, one can get rid of the oscillation effect caused by the dual solution obtained by coarse mesh. That is the reason why Galerkin method, including the continuous Galerkin (CG) and discontinuous Galerkin (DG) methods are widely used for a posteriori goal-oriented

error estimation in analysis of linear transient dynamics [160, 153, 161, 43, 105], and also has been attempted to apply in nonlinear dynamics of structures with large deformation [106, 162]. In particular, Bangerth et al. [43] proposed a dual weight residual (DWR) method based on the FEM Galerkin underlying space-time discretization, and pointed out that the robust time finite difference algorithms (like Crank-Nicolson and Newmark family methods) are also amendable to this scheme, since they can be equivalently understood as the specific lower-order CG or DG methods. According to goal-oriented adaptivity for analysis of frequency-domain problem, established by using the error assessment obtained through bilinear-type dual problems [163], Muñoz-Matute et al. [161] extended it to address the 1D time-domain wave propagation problem. Although, DWR-driven adaptivity is able to chase the source of error, and thus delivering an efficient refinement adjustable to QoI, as pointed out by Francesc et al. [153], it is expensive, because adjoint solutions have to be calculated and conserved at every time step. It may lead to unaffordable memory sources in large complex problems. Therefore, they employed modal analysis to acquire an appropriate approximation to substitute the solution of dual problem [159, 160]. Meanwhile, this approximation is simple to be improved via the post-processing approach to handle each vibration mode at only one spatial space, and afterwards recuperate the time-dependent form based upon the time-harmonic assumption. As the approximated adjoint solution can be obtained before the computation of primal problem, only primal solution at one time step is required to be stored, which is computationally advantageous.

The main contributions of this chapter can be summarized as follows. Firstly, construct a full-discrete form for linear transient dynamic problem in the framework of GIFT (for spatial discretization) and Newmark method (for time discretization). Then, derive the space-time first-order discrete form based on the second-order Newmark method. This first-order form is extremely helpful to develop the goal-oriented space-time adaptivity technique afterwards. Inspired by the aforementioned work [47] where a unidirectional multilevel space-time adaptive FEM/Newmark strategy was built, it is extended to be in the GIFT/Newmark platform and name it as

the unidirectional multi-level space-time adaptive GIFT/Newmark (UM-STAGN) method. However, according to the study, even though one can control the maximum error of the element (following the work [47]), and error estimation of QoI as well, the true error of QoI is kept at a certain magnitude and cannot be reduced any further. That is owing to the fact that only the information at the boundary of domains between two mesh levels is transfered while the information in the interior of lower level mesh gets stuck, leading to consequence that the error estimation fails to catch the error propagation. To overcome this limit, it is intended to propose a energy-based space-time adaptive GIFT/Newmark (E-STAGN) methodology. In this scheme, the space-time adaptation is driven by energy-based (involving potential and kinetic energy) a posteriori error estimation. More crucially, in order to avoid the error locking phenomenon occurring in UM-STAGN algorithm, the *whole* system is solved at each adaptive step. It allows the solution of the whole system is updated and the error is re-assessed, and thus the error conduction can be taken into consideration. In addition, different from common space-time refinement strategy [105, 43], where people decomposed space-time error into spatial and temporal error in sense of experience. Within this approach, the temporal discretization is determined firstly and the spatial refinement is identified afterwards. Producing the Rothe method space-time meshes [43], a refinement method wherein both spatial element and time step are discretized at the same time is established. This is for the reason that, in our opinion, it is actually difficult to separately quantify the exact error contributions from space and time discretization. Although, the E-STAGN approach is able to well settle the error locking problem, it is not efficient enough. As it is acknowledged, the error is accumulated with time so that the energy-based error estimation will result in an increasing intensive refinement as time step gets closed to the terminal, which is inefficient. Therefore, a goal-oriented adaptive technique is developed, that is, the dual-weighted residual adaptive GIFT/Newmark (DWR-STAGN) method. This technique derives from the aforementioned first-order discrete form, and is able to uncover the source of error and thus deliver an error-sensitive adaptive refinement. Particularly, when computing the local error

estimator of DWR, the dual solution is replaced by the dual error calculated within the difference between the dual solution on refined mesh and the projection of dual solution from refined mesh to coarse mesh. By this way, the oscillation of dual solution can be diminished.

The context of this chapter is structured as follows. In Section 6.2, a time-domain full-discrete form by GIFT/Newmark method for 2D linear elastodynamics is constructed, and furthermore the first-order matrix form based on the second-order Newmark method is deduced. Then the principles of three proposed space-time adaptive technologies, i.e., UM-STAGN, E-STAGN and DWR-STAGN methods are concretely introduced in Section 6.3, and subsequently make comparisons through the several examples in Section 6.4. The conclusions are drawn in Section 6.5.

## 6.2 2D Linear elastic dynamics

Let us consider a PDE of motion equation based on the Newton's second law

$$\begin{aligned}
&\rho\ddot{u} = \boldsymbol{\nabla}\boldsymbol{\sigma} + \boldsymbol{f} \text{ in } \Omega \times I,\\
&u(\boldsymbol{x}, t) = 0, \text{on } \partial\Omega_D \times I,\\
&\partial_n u(\boldsymbol{x}, t) = \bar{\boldsymbol{f}}(t), \text{ on } \partial\Omega_N \times I,\\
&u(\boldsymbol{x}, 0) = u_0, \text{ at } \Omega \times I_0,\\
&\dot{u}(\boldsymbol{x}, 0) = \dot{u}_0, \text{ at } \Omega \times I_0,
\end{aligned} \tag{6.2.1}$$

where $\dot{(\bullet)} = \dfrac{d}{dt}(\bullet)$ indicates the derivation with respect to time. $\boldsymbol{f}(\boldsymbol{x}, t)$ is the source term, also understood as the body force. $\rho$ is density, and $\boldsymbol{\sigma}$ is considered as Cauchy stress tensor. $\Omega \subset \mathbb{R}^2$ is defined as a two-dimensional spatial open bounded domain, with boundary conditions, i.e., $\partial\Omega_D$ (Dirichlet boundary) and $\partial\Omega_N$ (Neumann boundary), such that $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. The time domain $I := (0, T]$ is bounded with the initial condition $I_0 := \{0\}$. The space-time Dirichlet boundary ($\partial\Omega_D \times I$) condition is zero and $\bar{\boldsymbol{f}}(t)$ is regarded as the space-time Neumann boundary ($\partial\Omega_N \times I$) condition. $\dot{u}_0$ and $u_0$ are expressed as initial velocity and displacement.

Before inducing the weak form for the problem, we intend to introduce the

functional spaces to be used as follows. The spatial functional space is defined by a standard Sobolev space, such as

$$\mathscr{V} := \left\{ v \in \left[ H^1(\Omega) \right]^2 : v = 0 \text{ on } \partial\Omega_D \right\}. \tag{6.2.2}$$

Then we can define a space-time functional space by Bochner space $L^2(I;\mathscr{V})$. Bochner space is designed to investigate the time-domain PDEs within the functional analysis method. Suppose that a measurable space $(T, \sum, \mu)$, a Banach space $(X, \|\cdot\|_X)$ and a constant $p$ are provided, the Bochner space $L^p(T;X)$ is defined to be the Kolmogorov quotient of the space including all Bochner measurable functions $u : T \to X$ such that the relevant norm is finite, which reads:

$$\|u\|_{L^p(T;X)} := \left( \int_T \|u(t)\|_X^p \, d\mu(t) \right)^{1/p} < +\infty, \text{ for } 1 \leqslant p < +\infty. \tag{6.2.3}$$

Thus, the solution of Eq.(6.2.1) belongs to the space defined by

$$\mathscr{U} := \left\{ u \in L^2(I;\mathscr{V}),\ \dot{u} \in L^2(I;[L^2(\Omega)]^2),\ \ddot{u} \in L^2(I;V^*) \right\}, \tag{6.2.4}$$

where $V^*$ indicates the dual space of $\mathscr{V}$. It should be noted that, in the space $\mathscr{U}$, there is a particular so-called "weak" (or "variational") solution, such that $u \in L^2(I;\mathscr{V}) \cap C(\bar{I};\mathscr{V})$, $\dot{u} \in L^2(I;[L^2(\Omega)]^2) \cap C(\bar{I};[L^2(\Omega)]^2)$, and $\ddot{u} \in L^2(I;V^*)$, see [164]. It denotes the solution $u \in \mathscr{U}$ and its corresponding time derivations are continuous in time domain.

### 6.2.1 Spatial discretization by GIFT method

We start with the weak form of Eq.(6.2.1) that $\forall \delta u \in \mathscr{V}$, find $(u, \dot{u}, \ddot{u}) \in \mathscr{U}$ such that,

$$\begin{aligned} \int_\Omega \rho \delta u^T \ddot{u} d\Omega + \int_\Omega \boldsymbol{\varepsilon}^T(\delta u) \mathbf{D} \boldsymbol{\varepsilon}(u) d\Omega - \int_\Omega \delta u^T \boldsymbol{f} d\Omega - \int_{\partial\Omega_N} \delta u^T \bar{\boldsymbol{f}} d\Gamma = 0, \\ \int_\Omega \rho \delta u^T u(\boldsymbol{x}, 0) d\Omega = \int_\Omega \rho \delta u^T u_0 d\Omega, \\ \int_\Omega \rho \delta \dot{u}^T \dot{u}(\boldsymbol{x}, 0) d\Omega = \int_\Omega \rho \delta \dot{u}^T \dot{u}_0 d\Omega. \end{aligned} \tag{6.2.5}$$

All the integral terms in Eq.(6.2.5) are scalars. As $\delta u$ is arbitrary, the integrals $\int_\Omega \rho \delta u^T \ddot{u} d\Omega$ (unit: $J$) and $\int_\Omega \boldsymbol{\varepsilon}^T(\delta u) \mathbf{D} \boldsymbol{\varepsilon}(u) d\Omega$ (unit: $J$) can be understood as virtual

kinetic and potential energy of structure. Accordingly, terms $\int_\Omega \delta u^T \boldsymbol{f} d\Omega$ (unit: $J$) and $\int_{\partial\Omega_N} \delta u^T \bar{\boldsymbol{f}} d\Gamma$ (unit: $J$) are interpreted as the virtual work carried out by internal (body) force $\boldsymbol{f}$ and external (boundary) force $\bar{\boldsymbol{f}}$. As a result, the first equation in Eq.(6.2.5) denotes the principle of virtual energy balance. The second and third equation implies integrals of field of displacement $u$ and velocity $\dot{u}$ respectively at initial time step, which are employed to quantify the initial conditions.

In linear elasticity, we define stress tensor $\boldsymbol{\sigma} := \mathbf{D}\boldsymbol{\varepsilon}(u)$, where strain tensor $\boldsymbol{\varepsilon}(u)$ and elastic matrix $\mathbf{D}$ for plane stress are defined by

$$\boldsymbol{\varepsilon}(u) = \mathbb{L}u = \begin{Bmatrix} \dfrac{\partial}{\partial x}, & 0 \\ 0, & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y}, & \dfrac{\partial}{\partial x} \end{Bmatrix} u, \ \mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & \nu & 0 \\ 0 & 0 & \dfrac{(1-\nu)}{2} \end{bmatrix}, \tag{6.2.6}$$

where $E$ is Young's modulus and $\nu$ is Poission rate. Let us define some bilinear forms, which reads

$$\begin{aligned} m(v,u) = \int_\Omega \rho v^T u d\Omega, \ a(v,u) = \int_\Omega \boldsymbol{\varepsilon}^T(v)\mathbf{D}\boldsymbol{\varepsilon}(u) d\Omega, \\ (v,u) = \int_\Omega v^T u d\Omega, \ (v,u)_{\partial\Omega_N} = \int_{\partial\Omega_N} v^T u d\Gamma. \end{aligned} \tag{6.2.7}$$

Then the Eq.(6.2.5) can be rewritten as

$$\begin{aligned} m(\delta u, \ddot{u}) + a(\delta u, u) - (\delta u, \boldsymbol{f}) - (\delta u, \bar{\boldsymbol{f}})_{\partial\Omega_N} = 0 \\ m(\delta u, u(\boldsymbol{x}, 0)) = m(\delta u, u_0), \\ m(\delta \dot{u}, \dot{u}(\boldsymbol{x}, 0)) = m(\delta \dot{u}, \dot{u}_0). \end{aligned} \tag{6.2.8}$$

Here, the spatial domain is discretized using Galerkin GIFT scheme. This framework has been introduced in Chapter 4 and employed to address the adaptivity for vibration in Chapter 5. Analogously, PHT-splines basis functions are applied to represent the solution field, and NURBS shape functions are used to characterize the geometric mapping $\mathcal{F}$, which leads to,

$$u^h(\boldsymbol{x},t) = \sum_{\boldsymbol{k}\in\boldsymbol{\mathcal{I}}} \boldsymbol{T}_k(\boldsymbol{\xi})\mathbf{U}_{\boldsymbol{k}}(t), \ \dot{u}^h(\boldsymbol{x},t) = \sum_{\boldsymbol{k}\in\boldsymbol{\mathcal{I}}} \boldsymbol{T}_k(\boldsymbol{\xi})\dot{\mathbf{U}}_{\boldsymbol{k}}(t), \ \ddot{u}^h(\boldsymbol{x},t) = \sum_{\boldsymbol{k}\in\boldsymbol{\mathcal{I}}} \boldsymbol{T}_k(\boldsymbol{\xi})\ddot{\mathbf{U}}_{\boldsymbol{k}}(t), \tag{6.2.9}$$

$$\boldsymbol{x} = \mathcal{F}(\boldsymbol{\xi}) = \sum_{\boldsymbol{k}\in\boldsymbol{\mathcal{I}}} \boldsymbol{N_k}(\boldsymbol{\xi})\boldsymbol{P_k},\ \mathcal{F}:\mathcal{P}\to\Omega \tag{6.2.10}$$

where $\mathbf{U}_{\boldsymbol{k}}(t)$, $\dot{\mathbf{U}}_{\boldsymbol{k}}(t)$ and $\ddot{\mathbf{U}}_{\boldsymbol{k}}(t)$ are time-dependent unknown control variables. $T_{\boldsymbol{k}}(\boldsymbol{\xi})$ are the PHT-splines shape functions defined in Section 4.2. $\mathcal{F}$ denotes the mapping from parametric domain $\mathcal{P}$ to physical space $\Omega$. $\boldsymbol{N_k}(\boldsymbol{\xi})$ are the NURBS basis functions and $\boldsymbol{P_k}$ are control points. $\boldsymbol{\mathcal{I}}$ is introduced as 2-dimensional multi-index $(i,j)$, and $\boldsymbol{k}$ is interchangeably regarded as the collapsed notation for $\boldsymbol{\mathcal{I}}$. For example, $\boldsymbol{I} = 1,2,3,\ldots$ denotes $(i,j) = (1,1),(1,2),(1,3),\ldots$. Then the semi-discrete equations can be written, such as, $\forall\,\{\delta u, \delta\dot{u}\} \in \mathscr{V}^h \times \mathscr{V}^h$,

$$\begin{aligned} m(\delta u, \ddot{u}^h) + a(\delta u, u^h) - (\delta u, \boldsymbol{f}) - (\delta u, \bar{\boldsymbol{f}})_{\partial\Omega_N} &= 0 \\ m(\delta u, u^h(\boldsymbol{x},0)) &= m(\delta u, u_0), \\ m(\delta\dot{u}, \dot{u}^h(\boldsymbol{x},0)) &= m(\delta\dot{u}, \dot{u}_0). \end{aligned} \tag{6.2.11}$$

Here $\mathscr{V}^h$ is a standard finite element space with $\mathscr{V}^h \subset \mathscr{V}$, defined by

$$\mathscr{V}^h = \left\{ v^h \in C(\Omega)^2 : v^h|_{\mathcal{K}} \in Q_s(\mathcal{K})^2,\ \forall \mathcal{K} \in \mathbb{K},\ v^h|_{\partial\Omega_D=0} \right\}, \tag{6.2.12}$$

where $\mathbb{K}$ is the division of physical domain $\Omega$ into 2D non-overlapping quadrilaterals elements. The mesh size $h$ is defined as

$$h := \max_{\mathcal{K}\in\mathbb{K}} h_{\mathcal{K}},\ h_{\mathcal{K}} = \mathrm{diam}(\mathcal{K}) \tag{6.2.13}$$

Moreover, $Q_s(\mathcal{K})^2$ is two-dimensional full $s$-degree polynomial spaces.

Based on the semi-discrete form in Eq.(6.2.11), we can acquire linear ordinary differential equations (ODEs) with respect to time as follows

$$\begin{aligned} \delta\mathbf{U}^T \left[ \mathbf{M}\ddot{\mathbf{U}}(t) + \mathbf{K}\mathbf{U}(t) - \mathbf{F}(t) \right] &= 0, \\ \delta\mathbf{U}^T\mathbf{M}\mathbf{U}(0) &= \delta\mathbf{U}^T\boldsymbol{T}^T u_0, \\ \delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}(0) &= \delta\dot{\mathbf{U}}^T\boldsymbol{T}^T\dot{u}_0. \end{aligned} \tag{6.2.14}$$

where the mass matrix $\mathbf{M}$, stiffness matrix $\mathbf{K}$, and force term $\mathbf{F}$ are represented respectively by

$$\begin{aligned} \mathbf{K} &= \int_{\mathcal{P}} (\mathbb{L}\boldsymbol{T})^T \mathbf{D}\mathbb{L}\boldsymbol{T}|\mathbf{J}(\boldsymbol{\xi})|d\mathcal{P}, \\ \mathbf{M} &= \int_{\mathcal{P}} \boldsymbol{T}^T\mathbf{D}\boldsymbol{T}|\mathbf{J}(\boldsymbol{\xi})|d\mathcal{P}, \\ \mathbf{F} &= \int_{\mathcal{P}} \boldsymbol{T}^T\boldsymbol{f}|\mathbf{J}(\boldsymbol{\xi})|d\mathcal{P} + \int_{\partial\mathcal{P}_N} \boldsymbol{T}^T\bar{\boldsymbol{f}}|\mathbf{J}(\boldsymbol{\xi})|d\Gamma_{\mathcal{P}}. \end{aligned} \tag{6.2.15}$$

where $\mathbf{J}(\boldsymbol{\xi})$ is the Jacobian matrix with the expression in Eq.(5.3.30). Since $\delta\mathbf{U}$ can be arbitrary, we obtain

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{U}}(t) + \mathbf{K}\mathbf{U}(t) &= \mathbf{F}(t), \\ \mathbf{M}\mathbf{U}(0) &= \boldsymbol{T}^T u_0, \\ \mathbf{M}\dot{\mathbf{U}}(0) &= \boldsymbol{T}^T \dot{u}_0. \end{aligned} \tag{6.2.16}$$

### 6.2.2 Time discretization by Newmark scheme

At the initial time step $t_0 = 0$, the initial acceleration, velocity and displacement are calculated directly from Eq.(6.2.16):

$$\begin{aligned} \ddot{\mathbf{U}}(0) &= \mathbf{M}^{-1}\left[\mathbf{F}(0) - \mathbf{K}\mathbf{U}(0)\right], \\ \mathbf{U}(0) &= \mathbf{M}^{-1}\boldsymbol{T}^T u_0, \\ \dot{\mathbf{U}}(0) &= \mathbf{M}^{-1}\boldsymbol{T}^T \dot{u}_0. \end{aligned} \tag{6.2.17}$$

Next, in the time domain $I = (0, T]$, the sequence of discrete time levels can be described as

$$0 = t_0 < t_1 < \ldots < t_m < \ldots < t_M = T, \tag{6.2.18}$$

then the length of time step between $t_m$ and $t_{m-1} (m > 1)$ is denoted as $\Delta t = t_m - t_{m-1}$. According to the Newmark method [165], we can have following formulations:

$$\begin{aligned} \dot{\mathbf{U}}_m &= \dot{\mathbf{U}}_{m-1} + (1-\gamma)\Delta t\ddot{\mathbf{U}}_{m-1} + \gamma\Delta t\ddot{\mathbf{U}}_m, \\ \mathbf{U}_m &= \mathbf{U}_{m-1} + \Delta t\dot{\mathbf{U}}_{m-1} + \frac{\Delta t^2}{2}(1-2\beta)\ddot{\mathbf{U}}_{m-1} + \beta\Delta t^2\ddot{\mathbf{U}}_m, \end{aligned} \tag{6.2.19}$$

where the parameters $\beta, \gamma$ are set as $\beta = \dfrac{1}{4}, \gamma = \dfrac{1}{2}$ in this work to make sure the stability of Newmark algorithm is unconditional (see [137]). The first equation in Eq.(6.2.16) at time step $t_m$ is written as

$$\mathbf{M}\ddot{\mathbf{U}}_m = \mathbf{F}_m - \mathbf{K}\mathbf{U}_m. \tag{6.2.20}$$

Thus, the combination of Eq.(6.2.17), Eq.(6.2.19) and Eq.(6.2.20) arrives at full-discrete system of the problem.

Generally, when solving this full-discrete system over time domain $I = (0, T]$, we firstly substitute Eq.(6.2.19) into Eq.(6.2.20), to compute the acceleration $\ddot{\mathbf{U}}_m$ by

$$\ddot{\mathbf{U}}_m = \left(\mathbf{M} + \beta\Delta t^2\mathbf{K}\right)^{-1}\left[\mathbf{F}_m - \mathbf{K}\left(\mathbf{U}_{m-1} + \Delta t\dot{\mathbf{U}}_{m-1} + \frac{\Delta t^2}{2}(1-2\beta)\ddot{\mathbf{U}}_{m-1}\right)\right]. \tag{6.2.21}$$

Note that the unit of stiffness matrix $\mathbf{K}$ is $N/m$, $\beta$ is dimensionless constant, so that the unit of $\beta\Delta t^2\mathbf{K}$ is $N \cdot s^2/m$ (also understood as $kg$) which is the same as that of mass matrix $\mathbf{M}$. Thus $\mathbf{M}+\beta\Delta t^2\mathbf{K}$ can be regarded as an equivalent mass-like matrix updated at every time step, used to compute the acceleration $\ddot{\mathbf{U}}_m$. Then putting Eq.(6.2.21) back to the Eq.(6.2.19), one can figure out $\dot{\mathbf{U}}_m$ and $\mathbf{U}_m$ accordingly.

### 6.2.3 First-order form based on Newmark method

As we know, Newmark method mentioned above is second-order. In this section, it is intended to deduce the first-order (also the so-called one-step) formulation and its full-discrete form for the problem in Eq.(6.2.1). It is for DWR error estimation in Section 6.3.3.

#### 6.2.3.1 First-order weak form

We begin with the following weak-form equation, i.e.,

$$\mathscr{A} + \mathscr{A}_0 = \mathscr{B} + \mathscr{B}_0 + \mathscr{F}, \tag{6.2.22}$$

where notations $\mathscr{A}$, $\mathscr{A}_0$, $\mathscr{B}$, $\mathscr{B}_0$ and $\mathscr{F}$ are defined by, $(\delta\dot{u}, \delta u) \in \mathscr{V} \times \mathscr{V}$,

$$
\begin{aligned}
&\mathscr{A}\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}, \begin{bmatrix}\dot{u}_m\\ u_m\end{bmatrix}\right) := m(\delta\dot{u}, \dot{u}_m) + \gamma\Delta t a(\delta\dot{u}, u_m) + m(\delta u, u_m) + \beta\Delta t^2 a(\delta u, u_m),\\
&\mathscr{A}_0\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}, \begin{bmatrix}\dot{u}(\boldsymbol{x},0)\\ u(\boldsymbol{x},0)\end{bmatrix}\right) := m(\delta\dot{u}, \dot{u}(\boldsymbol{x},0)) + m(\delta u, u(\boldsymbol{x},0)),\\
&\mathscr{B}\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}; \begin{bmatrix}\dot{u}_{m-1}\\ u_{m-1}\end{bmatrix}\right) := m(\delta\dot{u}, \dot{u}_{m-1}) - (1-\gamma)\Delta t a(\delta\dot{u}, u_{m-1}) + \Delta t m(\delta u, \dot{u}_{m-1})\\
&+ m(\delta u, u_{m-1}) - \frac{\Delta t^2}{2}(1-2\beta)a(\delta u, u_{m-1}),\\
&\mathscr{B}_0\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}; \begin{bmatrix}\dot{u}_0\\ u_0\end{bmatrix}\right) := m(\delta\dot{u}, \dot{u}_0) + m(\delta u, u_0),\\
&\mathscr{F}\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}; \left[\boldsymbol{f}_{m-1}, \boldsymbol{f}_m, \bar{\boldsymbol{f}}_{m-1}, \bar{\boldsymbol{f}}_m\right]\right) := (1-\gamma)\Delta t\left[(\delta\dot{u}, \boldsymbol{f}_{m-1}) + (\delta\dot{u}, \bar{\boldsymbol{f}}_{m-1})_{\partial\Omega_N}\right]\\
&+ \gamma\Delta t\left[(\delta\dot{u}, \boldsymbol{f}_m) + (\delta\dot{u}, \bar{\boldsymbol{f}}_m)_{\partial\Omega_N}\right] + \frac{\Delta t^2}{2}(1-2\beta)\left[(\delta u, \boldsymbol{f}_{m-1}) + (\delta u, \bar{\boldsymbol{f}}_{m-1})_{\partial\Omega_N}\right]\\
&+ \beta\Delta t^2\left[(\delta u, \boldsymbol{f}_m) + (\delta u, \bar{\boldsymbol{f}}_m)_{\partial\Omega_N}\right].
\end{aligned}
\tag{6.2.23}
$$

The detailed process to obtain Eq.(6.2.22) and Eq.(6.2.23) can be found in Appendix B. Define $\boldsymbol{u} := [\dot{u}, u]^T \in \mathscr{U} \times \mathscr{U}$, $\boldsymbol{v} := [\delta\dot{u}, \delta u]^T \in \mathscr{V} \times \mathscr{V}$, $\boldsymbol{p}_{m-1,m} = \left[\boldsymbol{f}_{m-1}, \boldsymbol{f}_m, \bar{\boldsymbol{f}}_{m-1}, \bar{\boldsymbol{f}}_m\right]$, then the dynamic problem can be understood as solving the following problem step by step:

For $m = 0$, given $\boldsymbol{u}_0$, find $\boldsymbol{u}(\boldsymbol{x}, 0)$ such as

$$\mathscr{A}_0(\boldsymbol{u}(\boldsymbol{x},0), \boldsymbol{v}) = \mathscr{B}_0(\boldsymbol{v}; \boldsymbol{u}_0). \tag{6.2.24}$$

For $m > 0$, given $\boldsymbol{u}_{m-1}$, find $\boldsymbol{u}_m$ such that

$$\mathscr{A}(\boldsymbol{u}_m, \boldsymbol{v}) = \mathscr{B}(\boldsymbol{v}; \boldsymbol{u}_{m-1}) + \mathscr{F}(\boldsymbol{v}; \boldsymbol{p}_{m-1,m}). \tag{6.2.25}$$

Then the time accumulation can be indicated by

$$\mathbb{A}(\boldsymbol{u}, \boldsymbol{v}) := \sum_{m=1}^{M} \mathscr{A}(\boldsymbol{u}_m, \boldsymbol{v})\Delta t_m + \mathscr{A}_0(\boldsymbol{u}(\boldsymbol{x},0), \boldsymbol{v}), \tag{6.2.26}$$

$$\mathbb{B}(\boldsymbol{v}) := \sum_{m=1}^{M} \left[\mathscr{B}(\boldsymbol{v}; \boldsymbol{u}_{m-1}) + \mathscr{F}(\boldsymbol{v}; \boldsymbol{p}_{m-1,m})\right] \Delta t_m + \mathscr{B}_0(\boldsymbol{v}; \boldsymbol{u}_0), \tag{6.2.27}$$

$$\Delta t_m := t_m - t_{m-1}. \tag{6.2.28}$$

Thus, with these notations, the one-step space-time problem turns out to be a very simply expression as

For all $\boldsymbol{v} \in \mathscr{V} \times \mathscr{V}$, find $\boldsymbol{u} \in \mathscr{U} \times \mathscr{U}$ such that

$$\mathbb{A}(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{B}(\boldsymbol{v}). \tag{6.2.29}$$

#### 6.2.3.2 First-order discrete form

Now, it aims to derive the one-step discretized equations in this section. To begin with, we substitute discrete variables in Eq.(6.2.9) into Eq.(6.2.24) and Eq.(6.2.25) respectively and then arrive at, $\boldsymbol{v} \in \mathscr{V}^h \times \mathscr{V}^h$,

$$\mathscr{A}_0(\boldsymbol{u}^h(\boldsymbol{x}, 0), \boldsymbol{v}) = \mathscr{B}_0(\boldsymbol{v}; \boldsymbol{u}_0), \tag{6.2.30}$$

$$\mathscr{A}(\boldsymbol{u}^h_m, \boldsymbol{v}) = \mathscr{B}(\boldsymbol{v}; \boldsymbol{u}^h_{m-1}) + \mathscr{F}(\boldsymbol{v}; \boldsymbol{p}_{m-1,m}). \tag{6.2.31}$$

The discrete form of Eq.(6.2.30) at initial time step can be found in Eq.(6.2.14). Next, it can be easily recast into the following matrix form

$$\boldsymbol{\mathcal{A}}_0 = \boldsymbol{\mathcal{B}}_0, \tag{6.2.32}$$

where

$$\boldsymbol{\mathcal{A}}_0 \left( \begin{bmatrix} \delta\dot{\mathbf{U}} \\ \delta\mathbf{U} \end{bmatrix}, \begin{bmatrix} \dot{\mathbf{U}}(\boldsymbol{x}, 0) \\ \mathbf{U}(\boldsymbol{x}, 0) \end{bmatrix} \right) = \begin{bmatrix} \delta\dot{\mathbf{U}} \\ \delta\mathbf{U} \end{bmatrix}^T \begin{bmatrix} \mathbf{M} & 0 \\ 0 & \mathbf{M} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{U}}(\boldsymbol{x}, 0) \\ \mathbf{U}(\boldsymbol{x}, 0) \end{bmatrix}, \tag{6.2.33}$$

$$\boldsymbol{\mathcal{B}}_0 \left( \begin{bmatrix} \delta\dot{\mathbf{U}} \\ \delta\mathbf{U} \end{bmatrix}; \begin{bmatrix} \mathbf{T}^T \dot{u}_0 \\ \mathbf{T}^T u_0 \end{bmatrix} \right) = \begin{bmatrix} \delta\dot{\mathbf{U}} \\ \delta\mathbf{U} \end{bmatrix}^T \begin{bmatrix} \mathbf{T}^T \dot{u}_0 \\ \mathbf{T}^T u_0 \end{bmatrix}. \tag{6.2.34}$$

As to the Eq.(6.2.31), according to Eq.(6.2.23), if we choose $\delta u = 0$, it yields

$$\begin{aligned} m(\delta\dot{u}, \dot{u}^h_m) + \gamma\Delta t a(\delta\dot{u}, u^h_m) =& m(\delta\dot{u}, \dot{u}^h_{m-1}) - (1-\gamma)\Delta t a(\delta\dot{u}, u^h_{m-1}) \\ &+ (1-\gamma)\Delta t \left[(\delta\dot{u}, \boldsymbol{f}_{m-1}) + (\delta\dot{u}, \bar{\boldsymbol{f}}_{m-1})_{\partial\Omega_N}\right] \\ &+ \gamma\Delta t \left[(\delta\dot{u}, \boldsymbol{f}_m) + (\delta\dot{u}, \bar{\boldsymbol{f}}_m)_{\partial\Omega_N}\right], \end{aligned} \tag{6.2.35}$$

and furthermore we can have the matrix form

$$\begin{aligned}\delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}_m + \gamma\Delta t\delta\dot{\mathbf{U}}^T\mathbf{K}\mathbf{U}_m =& \delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}_{m-1} - (1-\gamma)\Delta t\delta\dot{\mathbf{U}}^T\mathbf{K}\mathbf{U}_{m-1} \\ &+ (1-\gamma)\Delta t\delta\dot{\mathbf{U}}^T\mathbf{F}_{m-1} + \gamma\Delta t\delta\dot{\mathbf{U}}^T\mathbf{F}_m.\end{aligned} \tag{6.2.36}$$

While $\delta\dot{u} = 0$ is selected, it leads to

$$\begin{aligned}m(\delta u, u_m^h) + \beta\Delta t^2 a(\delta u, u_m^h) =& \Delta t m(\delta u, \dot{u}_{m-1}^h) + m(\delta u, u_{m-1}^h) - \frac{\Delta t^2}{2}(1-2\beta)a(\delta u, u_{m-1}^h) \\ &+ \frac{\Delta t^2}{2}(1-2\beta)\left[(\delta u, \boldsymbol{f}_{m-1}) + (\delta u, \bar{\boldsymbol{f}}_{m-1})_{\partial\Omega_N}\right] \\ &+ \beta\Delta t^2\left[(\delta u, \boldsymbol{f}_m) + (\delta u, \bar{\boldsymbol{f}}_m)_{\partial\Omega_N}\right],\end{aligned} \tag{6.2.37}$$

and the related matrix formulation

$$\begin{aligned}\delta\mathbf{U}^T(\mathbf{M} + \beta\Delta t^2\mathbf{K})\mathbf{U}_m =& \Delta t\delta\mathbf{U}^T\mathbf{M}\dot{\mathbf{U}}_{m-1} + \delta\mathbf{U}^T\left[\mathbf{M} - \frac{\Delta t^2}{2}(1-2\beta)\mathbf{K}\right]\mathbf{U}_{m-1} + \\ &+ \frac{\Delta t^2}{2}(1-2\beta)\delta\mathbf{U}^T\mathbf{F}_{m-1} + \beta\Delta t^2\delta\mathbf{U}^T\mathbf{F}_m.\end{aligned} \tag{6.2.38}$$

Merging Eq.(6.2.36) and Eq.(6.2.38) results in the following expression

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{B}} + \boldsymbol{\mathcal{F}}, \tag{6.2.39}$$

where

$$\boldsymbol{\mathcal{A}}\left(\begin{bmatrix}\delta\dot{\mathbf{U}}\\ \delta\mathbf{U}\end{bmatrix}, \begin{bmatrix}\dot{\mathbf{U}}_m\\ \mathbf{U}_m\end{bmatrix}\right) = \begin{bmatrix}\delta\dot{\mathbf{U}}\\ \delta\mathbf{U}\end{bmatrix}^T \begin{bmatrix}\mathbf{M} & \gamma\Delta t\mathbf{K}\\ 0 & \mathbf{M} + \beta\Delta t^2\mathbf{K}\end{bmatrix} \begin{bmatrix}\dot{\mathbf{U}}_m\\ \mathbf{U}_m\end{bmatrix}, \tag{6.2.40}$$

$$\boldsymbol{\mathcal{B}}\left(\begin{bmatrix}\delta\dot{\mathbf{U}}\\ \delta\mathbf{U}\end{bmatrix}, \begin{bmatrix}\dot{\mathbf{U}}_{m-1}\\ \mathbf{U}_{m-1}\end{bmatrix}\right) = \begin{bmatrix}\delta\dot{\mathbf{U}}\\ \delta\mathbf{U}\end{bmatrix}^T \begin{bmatrix}\mathbf{M} & -(1-\gamma)\Delta t\mathbf{K}\\ \Delta t\mathbf{M} & \mathbf{M} - \frac{\Delta t^2}{2}(1-2\beta)\mathbf{K}\end{bmatrix} \begin{bmatrix}\dot{\mathbf{U}}_{m-1}\\ \mathbf{U}_{m-1}\end{bmatrix}, \tag{6.2.41}$$

$$\boldsymbol{\mathcal{F}}\left(\begin{bmatrix}\delta\dot{\mathbf{U}}\\ \delta\mathbf{U}\end{bmatrix}; [\mathbf{F}_{m-1}, \mathbf{F}_m]\right) = \begin{bmatrix}\delta\dot{\mathbf{U}}\\ \delta\mathbf{U}\end{bmatrix}^T \begin{bmatrix}(1-\gamma)\Delta t\mathbf{F}_{m-1} + \gamma\Delta t\mathbf{F}_m\\ \frac{\Delta t^2}{2}(1-2\beta)\mathbf{F}_{m-1} + \beta\Delta t^2\mathbf{F}_m\end{bmatrix} \tag{6.2.42}$$

Like Eq.(6.2.24) and Eq.(6.2.25), define $\boldsymbol{\mathcal{U}} := [\dot{\mathbf{U}}, \mathbf{U}]^T$, $\mathbf{V} := [\delta\dot{\mathbf{U}}, \delta\mathbf{U}]^T$, $\mathbf{P}_{m-1,m} := [\mathbf{F}_{m-1}, \mathbf{F}_m]$, and the solution process can be understood as a step-by-step procedure, which is summarized by following

For $m = 0$, given $\boldsymbol{\mathcal{U}}_0$, find $\boldsymbol{\mathcal{U}}(\boldsymbol{x}, 0)$ such as

$$\boldsymbol{\mathcal{A}}_0(\boldsymbol{\mathcal{U}}(\boldsymbol{x}, 0), \mathbf{V}) = \boldsymbol{\mathcal{B}}_0(\mathbf{V}; \boldsymbol{\mathcal{U}}_0). \tag{6.2.43}$$

For $m > 0$, given $\boldsymbol{\mathcal{U}}_{m-1}$, find $\boldsymbol{\mathcal{U}}_m$ such that

$$\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{U}}_m, \mathbf{V}) = \boldsymbol{\mathcal{B}}(\mathbf{V}; \boldsymbol{\mathcal{U}}_{m-1}) + \boldsymbol{\mathcal{F}}(\mathbf{V}; \mathbf{P}_{m-1,m}). \tag{6.2.44}$$

Then the space-time problem can be interpreted by

$$\bar{\mathcal{A}}(\boldsymbol{\mathcal{U}}, \mathbf{V}) := \sum_{m=1}^{M} \boldsymbol{\mathcal{A}}(\mathbf{U}_m, \mathbf{V})\Delta t_m + \boldsymbol{\mathcal{A}}_0(\mathbf{U}(\boldsymbol{x}, 0), \mathbf{V}), \tag{6.2.45}$$

$$\bar{\mathcal{B}}(\mathbf{V}) := \sum_{m=1}^{M} \left[\boldsymbol{\mathcal{B}}(\mathbf{V}; \mathbf{U}_{m-1}) + \boldsymbol{\mathcal{F}}(\mathbf{V}; \mathbf{P}_{m-1,m})\right] \Delta t_m + \boldsymbol{\mathcal{B}}_0(\mathbf{V}; \mathbf{U}_0), \tag{6.2.46}$$

$$\Delta t_m := t_m - t_{m-1}. \tag{6.2.47}$$

Thus, the first-order discrete form of problem can be also written in a brief form as follows

For all $\mathbf{V}$, find $\boldsymbol{\mathcal{U}}$ such that

$$\bar{\mathcal{A}}(\boldsymbol{\mathcal{U}}, \mathbf{V}) = \bar{\mathcal{B}}(\mathbf{V}). \tag{6.2.48}$$

## 6.3 Error estimation and adaptivity

In this section, it aims to introduce three types of space-time adaptive methods with local refinement based on a-*posteriori* error estimations. In particular, they are UM-STAGN, E-STAGN, DWR-STAGN techniques.

### 6.3.1 The UM-STAGN method

Motivated by the work [47, 166], we proposed UM-STAGN method. The core concept of this method is that during adaptivity process, only the elements wherein the

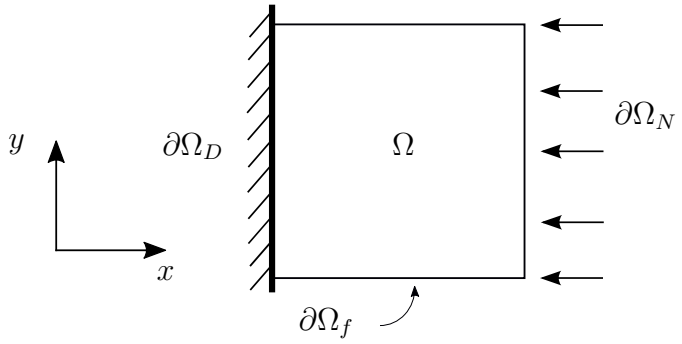



Figure 6.1: An example for 2D spatial domain. $\partial\Omega_D, \partial\Omega_N, \partial\Omega_f$ are Dirichlet, Neumann, and free boundary, respectively.

errors do not reach the prescribed accuracy are refined and go into the computation at next level.

#### 6.3.1.1 Error estimator and space-time refinement

Consider a case in 2D domain, as illustrated in Fig.6.1. A *posteriori* error evaluation is posed based on a hierarchical mesh. To be more specific, as shown in Fig.6.2(a), a space-time (ST) element in coarse mesh ($\Omega \times I$) is defined by

$$\Omega^{e,\mathcal{I}}_{(m-1,m)} = [\Omega^{e,i}_{m-1}, \Omega^{e,i}_{m}] \times [t_{m-1}, t_m],\ \Omega^{e,\mathcal{I}}_{(m-1,m)} \in \Omega \times I, \tag{6.3.49}$$

where $\mathcal{I} = (1, 2, \ldots, \mathcal{N})$ express as the number index of space-time cells. As it can be seen in Fig.6.2(b), one ST element is subdivided into eight ST elements, by cutting 1 space element into 4 (for 2D case) and separate one time step to 2 time intervals. Thus, the corresponding refined ST element in mesh ($\tilde{\Omega} \times \tilde{I}$) is generated. This process provides the following relationship of element domains between coarse mesh and refined mesh, such that,

$$\begin{gathered}\Omega^{e,\mathcal{I}}_{(m-1,m)} = \sum_{\mathcal{J}=1}^{4} \tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m}-1,\tilde{m})} + \sum_{\mathcal{J}=1}^{4} \tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m},\tilde{m}+1)},\ \Delta t_m = 2\Delta t_{\tilde{m}+1} = 2\Delta t_{\tilde{m}}, \\ \Omega^{e,i}_{m-1} = \sum_{j=1}^{4} \tilde{\Omega}^{e,j}_{\tilde{m}-1},\ \Omega^{e,i}_{m} = \sum_{j=1}^{4} \tilde{\Omega}^{e,j}_{\tilde{m}+1}.\end{gathered} \tag{6.3.50}$$

According to [158], the local error estimator $e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}}$ for the space-time element

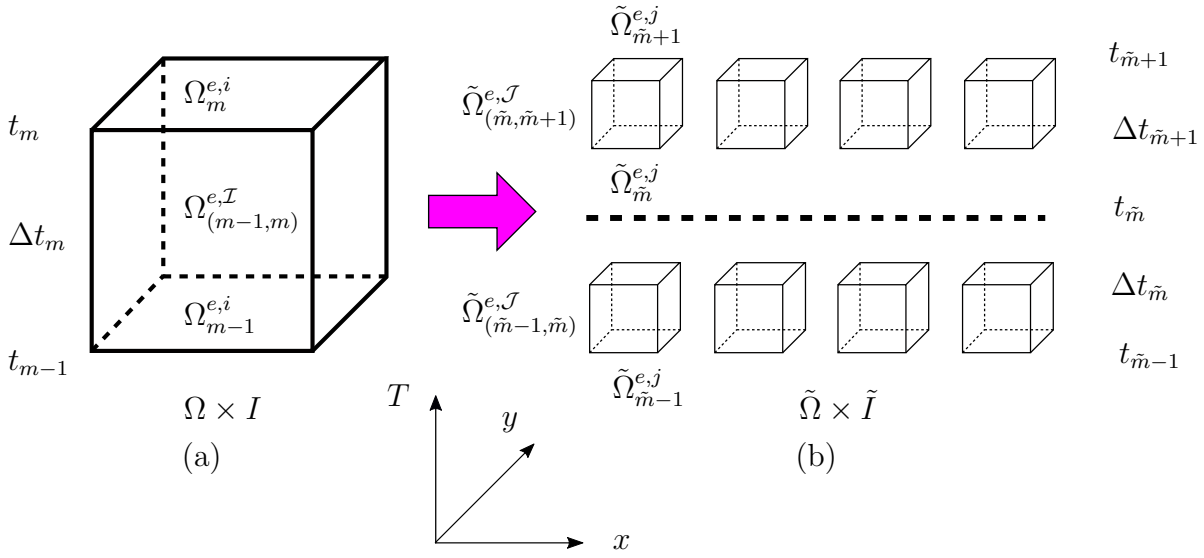



Figure 6.2: For 2D spatial domain, a space-time element in coarse mesh ($\Omega \times I$) (a) is discretized as eight space-time elements in refined mesh ($\tilde{\Omega} \times \tilde{I}$) (b).

$\Omega^{e,\mathcal{I}}_{(m-1,m)}$ is computed as follows

$$e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}} = \sqrt{\frac{\max\left\{\Delta E(\Omega^{e,i}_{m-1}),\ \Delta E(\Omega^{e,i}_m)\right\}}{E^0_{\max}}}. \tag{6.3.51}$$

Here $\Delta E(\cdot)$ is the energy increment defined by

$$\Delta E(\Omega^{e,i}_{m-1}) = \sum_{j=1}^{4} E(\tilde{\boldsymbol{u}}_{\tilde{m}-1}, \tilde{\Omega}^{e,j}_{\tilde{m}-1}) - E(\boldsymbol{u}^h_{m-1}, \Omega^{e,i}_{m-1}), \tag{6.3.52}$$

$$\Delta E(\Omega^{e,i}_m) = \sum_{j=1}^{4} E(\tilde{\boldsymbol{u}}_{\tilde{m}+1}, \tilde{\Omega}^{e,j}_{\tilde{m}+1}) - E(\boldsymbol{u}^h_m, \Omega^{e,i}_m), \tag{6.3.53}$$

where $E(\boldsymbol{u}, \Omega^e)$ is regarded as the energy indicator for spatial area $\Omega^e$, denoted by

$$E(\boldsymbol{u}, \Omega^e) = \underbrace{\frac{1}{2}\int_{\Omega^e} \rho \dot{u}^T \dot{u} d\Omega}_{\text{Kinetic energy}} + \underbrace{\frac{1}{2}\int_{\Omega^e} \boldsymbol{\varepsilon}^T(u)\mathbf{D}\boldsymbol{\varepsilon}(u) d\Omega}_{\text{Potential energy}}. \tag{6.3.54}$$

The $E^0_{\max} = \max\left\{E^0(\boldsymbol{u}^h_0, \Omega_0), \ldots, E^0(\boldsymbol{u}^h_m, \Omega_m), \ldots, E^0(\boldsymbol{u}^h_M, \Omega_M)\right\}$ ($t_M = T$) is the maximum energy of all time steps at the initial mesh. If $e\big|_{\Omega^{e,\mathcal{I}}_{(\text{m-1,m})}} > \tau$, where $\tau$ is the requested tolerance, the relevant elements will be recorded with red color and refined and fall into the computation at next level. Whilst the element fulfilling desired precision will not be touched anymore. It leads to the trend that the

computational domains turn to be smaller and smaller (see Fig.6.3), until the error estimator satisfies the threshold, namely, $e\big|^{\max}_{\Omega^n \times I^n} \leqslant \tau$ (see Algorithm 6).

It is worth noting that the boundary $\partial\Omega_f$ in Fig.6.1 is projected onto a space-time domain, for better interpretation of the space-time adaptive procedure. As a consequence, the space-time adaptation process can be illustrated from a 2D aspect, as performed in Fig.6.3. The vision is simple to be expanded to 3D space-time coordinate (2D spatial domain with 1D time interval), just by restoring $\partial\Omega_f$ to the whole space $\Omega$.

#### 6.3.1.2 Initial and boundary conditions for new subdomains

It should be pointed out that when a new subdomain is created for computation next step during the space-time adaptivity process, one should be careful to the imposition of initial and boundary conditions. The principle is set and explained via an example of 2D spatial domain here. Assume that the spatial domain is at time step $t_m (m > 0)$ (as shown in Fig.6.4), it is observed that the variables on the free boundary ($\partial\Omega_f$) and Neumann boundary ($\partial\Omega_N$), including those in the domain $\Omega^{n+1}$ (marked with solid circles), are all demanded to be computed. Except that , the variables on the Dirichlet boundary $\partial\Omega_D$ (marked with hollow circles) are fixed. With respect to the interface $\Gamma_{\Omega^n \cap \Omega^{n+1}}$ between $\Omega^n$ and $\Omega^{n+1}$, which can be also interpreted as the updated Dirichlet boundary for $\Omega^{n+1}$, the variables onside are dependent on solutions obtained by previous level discretization $\Omega^n$. To be more specific, supposed that, the solution at the boundary $\Gamma_{\Omega^n \cap \Omega^{n+1}}$ obtained by $\Omega^n$ is $\boldsymbol{u}_{\Omega^n}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_m)$, we impose the continuous condition, that is,

$$\boldsymbol{u}_{\Omega^n}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_m) = \boldsymbol{u}_{\Omega^{n+1}}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_m) \tag{6.3.55}$$

where $\boldsymbol{u}$ indicates $\boldsymbol{u} = (u, \dot{u}, \ddot{u})$. The interpolation method is used to identify the control variables at $\Gamma_{\Omega^n \cap \Omega^{n+1}}$ (denoted by blue triangles). Since the solution field is represented with PHT-splines shape functions, which are independent in each cell, and $\Omega^n$ and $\Omega^{n+1}$ are nested meanwhile, the interpolation algorithm is easy to be implemented using the approach referred in Appendix A.1.

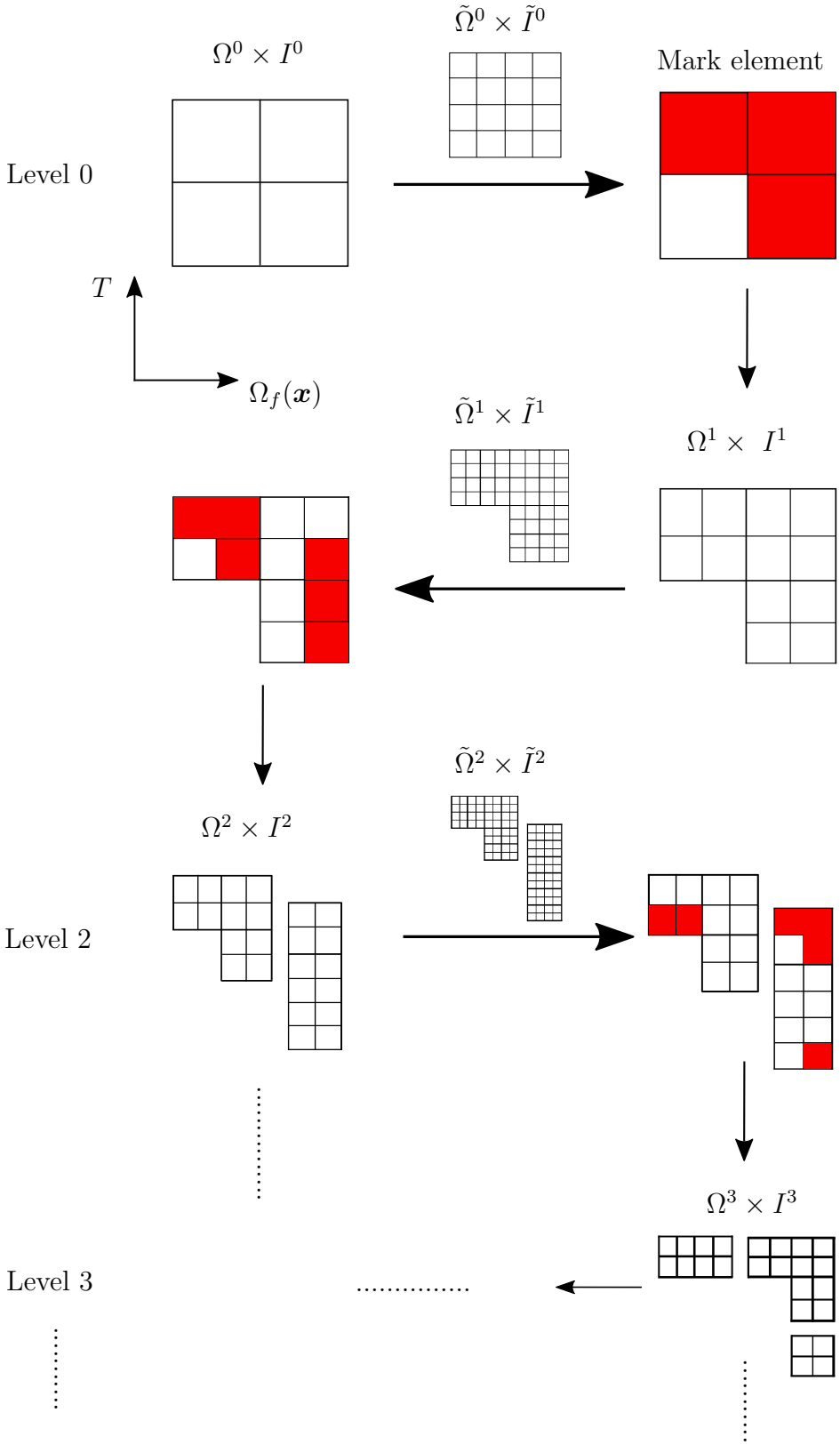



Figure 6.3: The 2D schematic illustration for adaptive space-time refinement procedure in the framework of UM-STAGN method. The spatial domain is the projection of boundary $\partial\Omega_f$ from Fig.6.1.

If the space field is at initial time step ($t_0 = 0$) (see Fig.6.4), similar to the case at $t_m (m > 0)$, we enforce the continuous condition, i.e.,

$$\boldsymbol{u}_{\Omega^n}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_0) = \boldsymbol{u}_{\Omega^{n+1}}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_0). \tag{6.3.56}$$

Then after refinement, the control variables (hollow squares) at $n+1$ level mesh that coincide with those on $n$ level mesh keep invariant. While new generated control variables can be also obtained through the efficient interpolation in Section A.1 in Appendix A.

**Remark 2** *In order to provide a better vision and interpretation, as seen in Fig.6.4 and Fig.6.5, the control variables are presented in a form of linear representation. Hence, the control variables belong to boundary locate at geometric edge, coinciding with nodes generated through linear finite element method. Nevertheless, it should be noticed that if spline basis functions adopted in IGA or GIFT are with higher-order (like quadratic, cubic,...), control variables (or points) will generally suspend around the bound.*

Furthermore, the internal interface boundary at the mid-time step (given as $t_m$) is built due to the temporal discretization, the variable $\boldsymbol{u}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_m)$ are determined employing linear interpolation method between time step $t_{m+1}$ and $t_{m-1}$, which reads

$$\boldsymbol{u}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_m) = \frac{1}{2}\left[\boldsymbol{u}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_{m-1}) + \boldsymbol{u}(\Gamma_{\Omega^n \cap \Omega^{n+1}}, t_{m+1})\right], \tag{6.3.57}$$

#### 6.3.1.3 The composition of the final mesh and solution

A space-time domain sometimes are subdivided into several independent ST subdomains, as seen in Fig.6.3. Suppose that a projected boundary $\partial\Omega_f$ is recovered to a entire 2D spatial area $\Omega$, which arrives at

$$\Omega^n \times I^n = (\Omega^{n,(1)} \times I^{n,(1)}) \cup \ldots \cup (\Omega^{n,(k)} \times I^{n,(k)}) \cup \ldots \cup (\Omega^{n,(K)} \times I^{n,(K)}), \tag{6.3.58}$$

where $k$ is the index and $K$ is the total number of of subregions. For example, $\Omega^2 \times I^2$ is split into 2 subdomains, that is, $\Omega^2 \times I^2 = (\Omega^{2,(1)} \times I^{2,(1)}) \cup (\Omega^{2,(2)} \times I^{2,(2)})$. Thus,

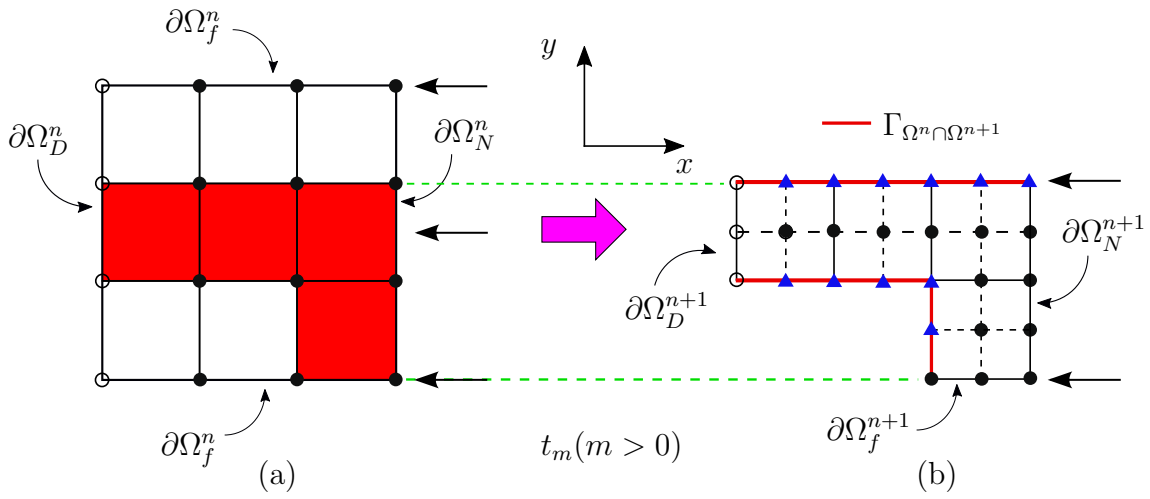



Figure 6.4: A two-dimension example of boundary imposition from $n$ level mesh to $n+1$ level mesh for spacial field. $\partial\Omega_D$, $\partial\Omega_N, \partial\Omega_f$ are Dirichlet, Neumann, free boundary conditions, respectively. $\Gamma_{\Omega^n/\Omega^{n+1}}$ is the internal interface between the $\Omega^n$ and $\Omega^{n+1}$. Hollow circles mean invariants on the Dirichlet boundary condition, the solid circles denote the control variables to be computed, the triangles indicate the discrete variables determined by the interpolation based on the enforced continuous condition in Eq.(6.3.55) at the interface between two level mesh.

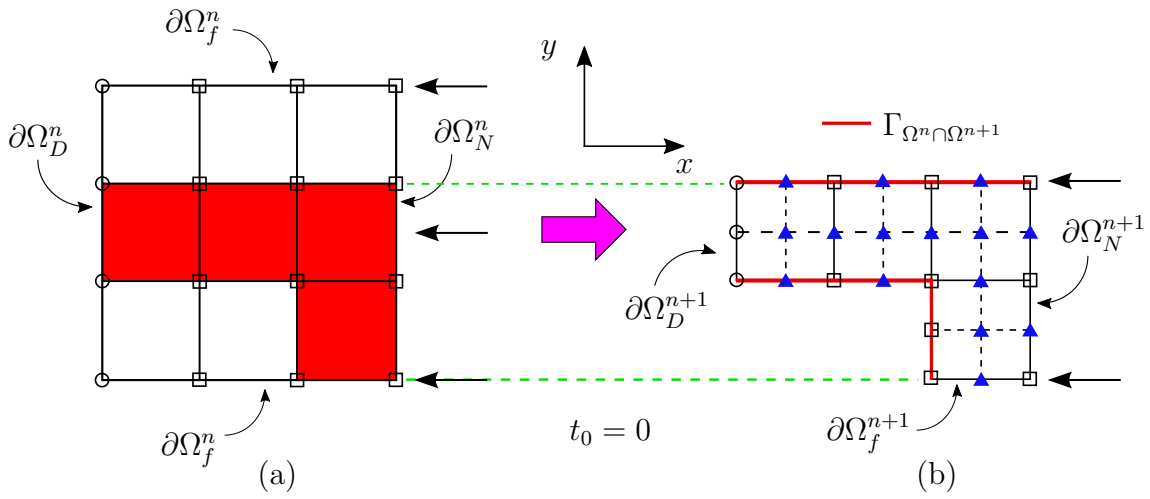



Figure 6.5: A two-dimension example of boundary imposition from $n$ level mesh to $n+1$ level mesh for spacial field at initial time step $t_0$. The hollow squares denote the initial control variables. Other information can be found in the caption of Fig.6.4.

it is intuitive that the final mesh (or solution) (assumed to be obtained after $N$ level adaptation) is composed by all meshes (or solutions) from the previous adaptive steps, which reads:

$$\Omega^{\text{final}} \times I^{\text{final}} = \Omega_*^0 \times I_*^0 \cup \Omega_*^1 \times I_*^1 \cup \ldots \cup \Omega_*^n \times I_*^n \cup \ldots \cup \Omega^N \times I^N, \tag{6.3.59}$$

$$\boldsymbol{u}(\Omega^{\text{final}}, I^{\text{final}}) = \boldsymbol{u}(\Omega_*^0, I_*^0) \cup \boldsymbol{u}(\Omega_*^1, I_*^1) \cup \ldots \cup \boldsymbol{u}(\Omega_*^n, I_*^n) \cup \ldots \boldsymbol{u}(\Omega^N, I^N), \tag{6.3.60}$$

where $\Omega_*^n \times I_*^n$ is defined as

$$\Omega_*^n \times I_*^n = (\Omega^n \times I^n) \backslash (\Omega^{n+1} \times I^{n+1}),\ n = 0, 1, \ldots, N-1. \tag{6.3.61}$$

Overall, the UM-STAGN method is summarized in Algorithm 6.

### 6.3.2 The E-STAGN method

The scheme of E-STAGN method is different from UM-STAGN technology. Within E-STAGN scheme, at each adaptive step, the error of every ST cell is essential to be evaluated.

#### 6.3.2.1 Error estimator

Here a ST element in Fig.6.2 is taken as an example. The local space-time error estimator of energy is defined by following

$$e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}} = \frac{\Delta t_m}{2}\left[E(\tilde{\boldsymbol{e}}_{m-1}, \Omega^{e,i}_{m-1}) + E(\tilde{\boldsymbol{e}}_m, \Omega^{e,i}_m)\right], \tag{6.3.62}$$

where $\mathcal{I} = 1, 2 \ldots \mathcal{N}$ denotes the index number of space-time element. The error assessment field is defined by $\tilde{\boldsymbol{e}} := \tilde{\boldsymbol{u}} - \boldsymbol{u}^h$, and $\tilde{\boldsymbol{u}}$ serves as the solution computed by a hierarchically refined ST mesh $\tilde{\Omega} \times \tilde{I}$. As a matter of fact, when computing $e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}}$, it is processed in the refined space $\tilde{\Omega} \times \tilde{I}$, which reads

$$e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}} = e\big|_{\sum_{\mathcal{J}=1}^{4} \tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m}-1,\tilde{m})}} + e\big|_{\sum_{\mathcal{J}=1}^{4} \tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m},\tilde{m}+1)}}, \tag{6.3.63}$$

$$e\big|_{\sum_{\mathcal{J}=1}^{4} \tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m}-1,\tilde{m})}} = \sum_{j=1}^{4} \frac{\Delta t_{\tilde{m}}}{2}\left[E(\tilde{\boldsymbol{e}}_{\tilde{m}-1}, \tilde{\Omega}^{e,j}_{\tilde{m}-1}) + E(\tilde{\boldsymbol{e}}_{\tilde{m}}, \tilde{\Omega}^{e,j}_{\tilde{m}})\right], \tag{6.3.64}$$

$$e\big|_{\sum_{\mathcal{J}=1}^{4} \tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m},\tilde{m}+1)}} = \sum_{j=1}^{4} \frac{\Delta t_{\tilde{m}+1}}{2}\left[E(\tilde{\boldsymbol{e}}_{\tilde{m}}, \tilde{\Omega}^{e,j}_{\tilde{m}}) + E(\tilde{\boldsymbol{e}}_{\tilde{m}+1}, \tilde{\Omega}^{e,j}_{\tilde{m}+1})\right]. \tag{6.3.65}$$

**Algorithm 6:** Adaptivity process for UM-STAGN method.

**Input:** The initial ST mesh $\Omega^0 \times I^0, e\big|^{\max}_{\Omega^0\times I^0}$.

**Output:** The final ST mesh $\Omega^{\text{final}} \times I^{\text{final}}$ and the solutions $\boldsymbol{u}^h(\Omega^{\text{final}}, I^{\text{final}})$.

$n = 0$.

**while** $e\big|^{max}_{\Omega^n\times I^n} > \tau$ **do**

Refine the $n$-level ST domain to $K$ subdomains with $n+1$ level.

$n = n + 1$.

**for** $k \leftarrow 1$ **to** $K$ **do**

Identify initial conditions $\boldsymbol{u}^h(\Omega_0^{n,(k)}, I_0^{n,(k)})$, $\tilde{\boldsymbol{u}}(\tilde{\Omega}_0^{n,(k)}, \tilde{I}_0^{n,(k)})$.

Determine boundary conditions $\boldsymbol{u}^h(\partial\Omega_D^{n,(k)}, I^{n,(k)})$, $\boldsymbol{u}^h(\Gamma^{n,(k)}_{\Omega^n\cap\Omega^{n-1}}, I^{n,(k)})$, $\bar{\boldsymbol{f}}_(\partial\Omega_N^{n,(k)}, I^{n,(k)})$ and $\tilde{\boldsymbol{u}}(\partial\tilde{\Omega}_D^{n,(k)}, \tilde{I}^{n,(k)})$, $\tilde{\boldsymbol{u}}(\tilde{\Gamma}^{n,(k)}_{\Omega^n\cap\Omega^{n-1}}, \tilde{I}^{n,(k)})$, $\bar{\boldsymbol{f}}(\partial\tilde{\Omega}_N^{n,(k)}, \tilde{I}^{n,(k)})$.

Solve the coarse system in $\Omega^{n,(k)} \times I^{n,(k)}$ to obtain solution $\boldsymbol{u}^h(\Omega^{n,(k)}, I^{n,(k)})$.

Solve the refined system in $\tilde{\Omega}^{n,(k)} \times \tilde{I}^{n,(k)}$ to obtain solution $\tilde{\boldsymbol{u}}(\tilde{\Omega}^{n,(k)}, \tilde{I}^{n,(k)})$.

**for** $\mathcal{I} \leftarrow 1$ **to** $\mathcal{N}_k$ **do**

Compute the $e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}}$ in Eq.(6.3.51).

**end**

Define $e\big|^{\max}_{\Omega^{n,(k)}\times I^{n,(k)}} = \max\left\{e\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}}\right\}^{\mathcal{N}_k}_{\mathcal{I}}$.

**end**

Define $e\big|^{\max}_{\Omega^n\times I^n} = \max\left\{e\big|^{\max}_{\Omega^{n,(k)}\times I^{n,(k)}}\right\}^{K}_{k}$

**end**

Acquire $\Omega^{\text{final}} \times I^{\text{final}}$ by Eq.(6.3.59).

Obtain $\boldsymbol{u}^h(\Omega^{\text{final}}, I^{\text{final}})$ by Eq.(6.3.60).

Here the $E(\tilde{\boldsymbol{e}}, \Omega^e)$, computed by Eq.(6.3.54), is the element-wise estimated energy error . Especially, when we compute the $\tilde{\boldsymbol{e}}_{\tilde{m}} = \tilde{\boldsymbol{u}}_{\tilde{m}} - \boldsymbol{u}^h_{\tilde{m}}$ at time step $t_{\tilde{m}}$, the $\boldsymbol{u}^h_{\tilde{m}}$ is obtained by the linear interpolation

$$\boldsymbol{u}^h_{\tilde{m}} = \frac{1}{2}(\boldsymbol{u}^h_{m-1} + \boldsymbol{u}^h_m). \tag{6.3.66}$$

#### 6.3.2.2 Marking strategy

We employ the approach discussed in Section 5.4.2 to mark the element to be refined. In that method, we seek the group of ST elements with the largest errors in the whole time domain $[0, T]$, such as

$$\forall e\big|_{\Omega^{e,\mathcal{I}^*}_{[0,T]}} \geqslant \forall e\big|_{\Omega^{e,\bar{\mathcal{I}}}_{[0,T]}}, \tag{6.3.67}$$

where $\mathcal{I}^* = 1, 2, \ldots, \mathcal{N}^*$ and $\bar{\mathcal{I}} = \mathcal{N}^* + 1, \ldots, \mathcal{N}$, and meantime it satisfies

$$\sum_{\mathcal{I}^*=1}^{\mathcal{N}^*} e\big|_{\Omega^{e,\mathcal{I}^*}_{[0,T]}} \geqslant \eta \cdot e\big|_{\Omega \times I} \tag{6.3.68}$$

where $\eta \in (0, 1]$ is the percentage. Then the series of ST elements ($\mathcal{I}^* = 1, 2, \ldots, \mathcal{N}^*$) is identified to be refined.

With respect to the determination of parameter $\eta$, readers can recall the strategy discussed in Section 5.4.2. The simulation is posed with the example in Section 6.4.2 and more details can be followed in Section 6.4.2. It can be seen in Fig.6.6 that, 9 numerical tests are proceeded in the range of $\eta \in [0.05, 0.5]$ and CPU time is marked. According to the observation for the numerical experiments, a parabolic targeted function with the similar expression in Eq.(5.4.50) is proposed to fit the sample points, such that

$$t = a(\eta - \eta_0)^2 + t_0. \tag{6.3.69}$$

Based on the least-square method, the coefficients $a$, $\eta_0$ and $t_0$ in Eq.(5.4.50) can be acquired as $a = 2.25 \times 10^5$, $\eta_0 \approx 0.28$ and $t_0 = 36500$. Thereby, the selected $\eta = 0.3$ leads to the best compromise between the time consumption and adaptive property.

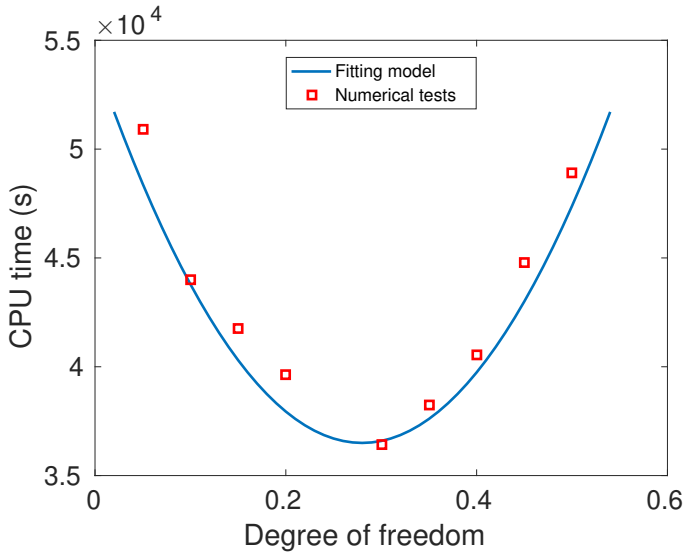



Figure 6.6: The numerical tests and the fitting model to identify the parameter $\eta$.

#### 6.3.2.3 Space-time refinement

The space-time adaptive procedure is illustrated in the Fig.6.7, still, the spatial domain is represented through the projection of $\Omega_f$ in Fig.6.1. When the ST element is refined (e.g., see $\Omega^1 \times I^1$), the black solid line denotes both the spatial domain and temporal domain in that ST element are discretized at the same time. When the temporal domain is refined, it is unavoidable to refine all the ST elements at the same time interval. The refinement denoted by the blue dash line is only for the time step. If we recover the $\Omega_f$ to $\Omega$ for a ST element, the illustration of only temporal refinement (blue dash line) can be seen in Fig.6.8.

Using E-STAGN refinement methodology, one should pay attention to the mesh generation at mid-time step. Let us take the example of a space-time domain $\Omega^2 \times I^2$ in Fig.6.7. Given that the mid-time step is $t_m$, and the spatial mesh at $t_{m+1}$ and $t_{m-1}$ is exhibited within $\Omega_{t_{m+1}}$ and $\Omega_{t_{m-1}}$, as been displayed in Fig.6.9. Our measurement is to find the least finest mesh that is able to cover both $\Omega_{t_{m+1}}$ and $\Omega_{t_{m-1}}$, and then determine it as the mid-time discretization $\Omega_{t_m}$. This measure can guarantee the wave propagates smoothly through the middle time step as all the information is preserved by the fine mesh.

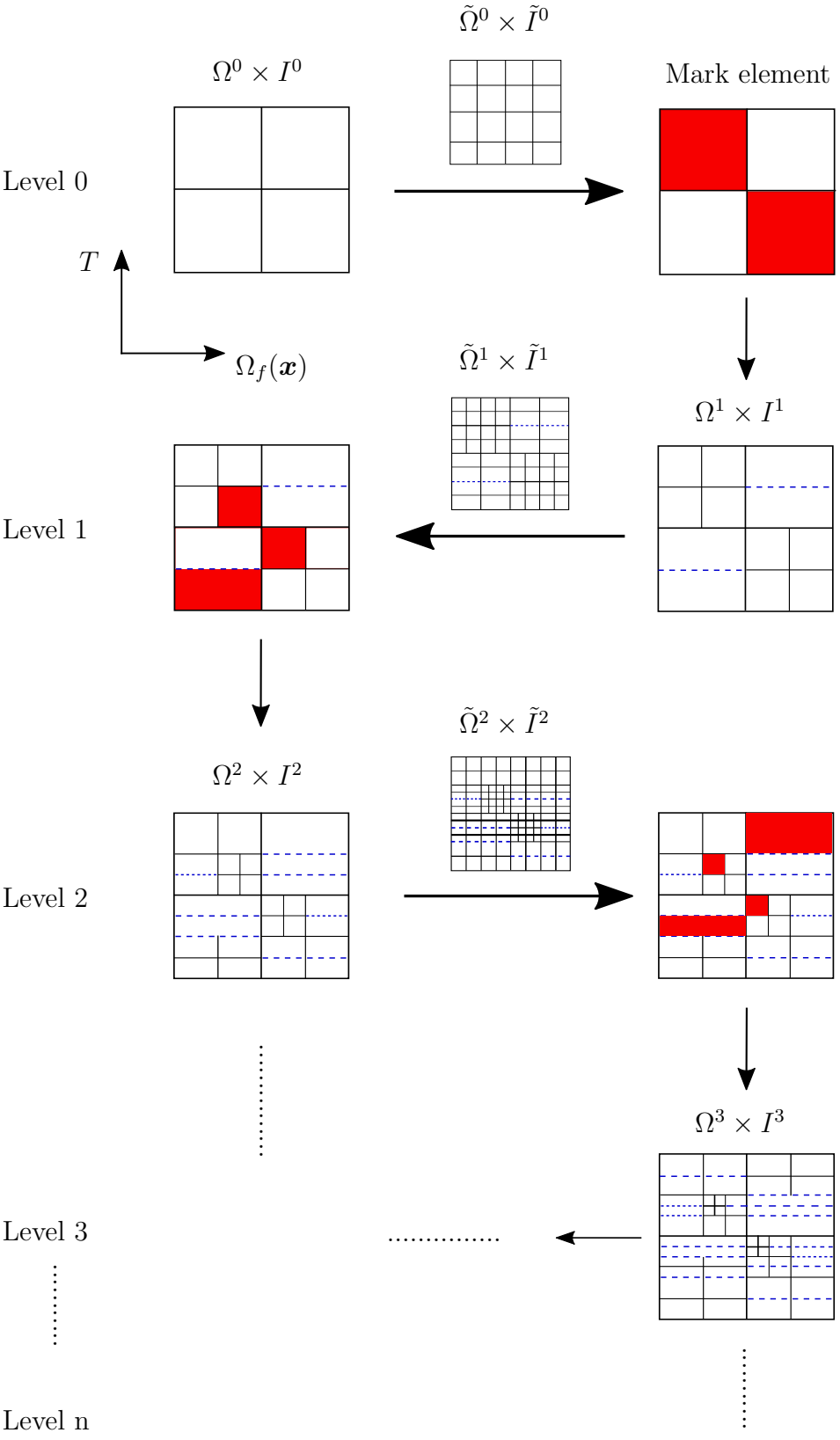



Figure 6.7: The principle illustration for adaptive space-time refinement process in the scheme of E-STAGN (or DWR-STAGN) technology. The spatial domain is the projection of boundary $\partial\Omega_f$ from Fig.6.1.

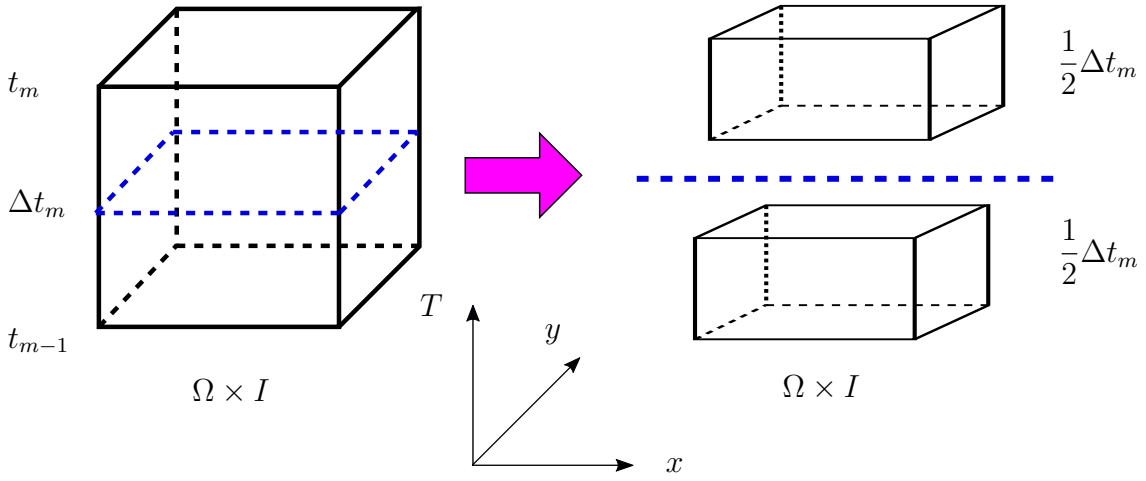



Figure 6.8: For 2D spatial domain, a space-time element is discretized to two space-time elements by only temporal refinement (blue dash line).

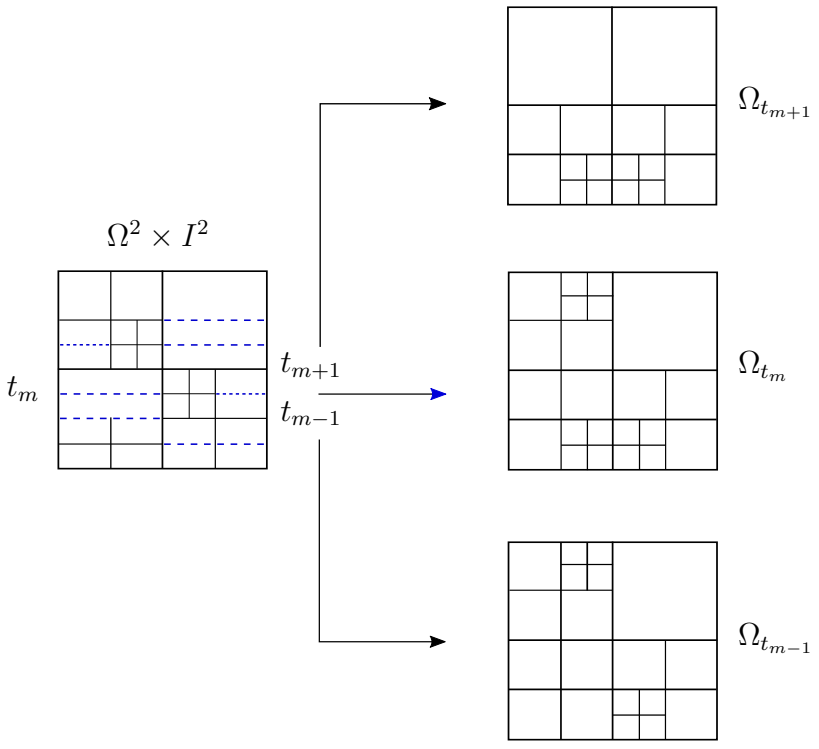



Figure 6.9: The measurement for spatial refinement at middle time step.

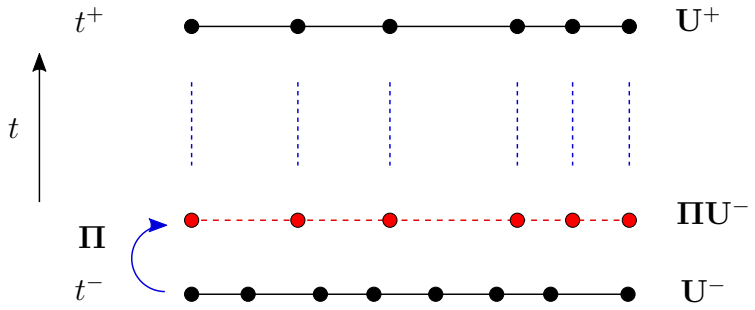



Figure 6.10: The schematic exhibition for the projection between two time step where the meshes are not nested. $U^+$ is the control displacement at the forward time step $t^+$, and $U^-$ denotes the control displacement at the backward time step $t^-$. $\mathbf{\Pi}$ is the projection operator.

#### 6.3.2.4 Projection method between adjacent time steps

Unlike the UM-STAGN method, where the space-time domains to be computed are in the form of "Method of Lines" (cf. [43]), which means the spatial and temporal discretization are both structured. As a comparison, E-STAGN method refinement strategy leads to a space-time mesh called "Rothe Method" mesh (e.g. [43]), where spatial mesh is unstructured but the time discretization is structured (as seen in Fig.6.7). Hence, it implies that when solving the problem within Newmark approach, it is required to apply the algorithm of interpolation or projection to keep the compatibility for variables between adjacent two time steps with different degrees of freedom. To be specific, If the discretization at backward time step is in the nest of that at forward time, that is $\Omega_{t^-} \subseteq \Omega_{t^+}$, as demonstrated in Fig.6.9, it will be fine since the interpolation store all information at backward time step in this sense. However, if the case is $\Omega^h_{t^-} \nsubseteq \Omega^h_{t^+}$ as presented in Fig.6.10, imposition of some constraints for the projection to remain stability is mandatory. In this situation, it aims to determine the projection $\mathbf{\Pi}\mathbf{U}^-$ for the control displacement $\mathbf{U}^-$ at backward time step $t^-$, and let it be consistent to the control displacement $\mathbf{U}^+$ at forward time step $t^+$. In this thesis, a methodology wherein the L2 norm of the projection is minimized and meanwhile the energy is preserved is developed.

To be specific, let us define a Lagrange function $\mathscr{L}(\cdot\,,\cdot)$ by following

$$\mathscr{L}(\mathbf{\Pi U}^-,\lambda)=\frac{1}{2}\left\|\mathbf{\Pi U}^- - \mathbb{P}\mathbf{U}^-\right\|_{L2}^2+\lambda\left[\frac{1}{2}(\mathbf{\Pi U}^-)^T\mathbf{K}^+\mathbf{\Pi U}^- - E_p^-\right], \tag{6.3.70}$$

where $\frac{1}{2}(\mathbf{\Pi U}^-)^T\mathbf{K}^+\mathbf{\Pi U}^-$ is the potential energy after projection and $E_p^-=\frac{1}{2}\mathbf{U}^{-^T}\mathbf{K}^-\mathbf{U}^-$ represents the potential energy before the projection. $\lambda$ is the Lagrange multiplier, and $\lambda\left[\frac{1}{2}(\mathbf{\Pi U}^-)^T\mathbf{K}^+\mathbf{\Pi U}^- - E_p^-\right]$ plays the role as the constraint in energy conservation. $\mathbb{P}\mathbf{U}^-$ is the projection obtained by the method introduced in Section A.2, which can be also regarded as the initial value for the iterative projection process in what follows. Thus, the projection problem can be mathematically interpreted as

Find $(\mathbf{\Pi U}^-,\lambda)$, such that,

$$\delta_{\mathbf{\Pi U}^-}\mathscr{L}=0,\ \Rightarrow \delta_{\mathbf{\Pi U}^-}\left\|\mathbf{\Pi U}^- - \mathbb{P}\mathbf{U}^-\right\|_{L2}^2+2\lambda\cdot\delta\mathbf{\Pi U}^{-^T}\mathbf{K}^+\mathbf{\Pi U}^-=0, \tag{6.3.71}$$

$$\delta_\lambda\mathscr{L}=0,\ \Rightarrow\ (\mathbf{\Pi U}^-)^T\mathbf{K}^+\mathbf{\Pi U}^- - 2E_p^-=0. \tag{6.3.72}$$

By solving Eq.(6.3.71), we have

$$\delta\mathbf{\Pi U}^{-^T}(\mathbf{\Pi U}-\mathbb{P}\mathbf{U}^-+\lambda\cdot\mathbf{K}^+\mathbf{\Pi U}^-)=0, \tag{6.3.73}$$

Since $\delta\mathbf{\Pi U}^{-^T}$ is arbitrary, we obtain

$$\mathbf{\Pi U}^- - \mathbb{P}\mathbf{U}^-+\lambda\cdot\mathbf{K}^+\mathbf{\Pi U}^-=0 \tag{6.3.74}$$

Combining Eq.(6.3.72), we can write the equation in the matrix form as follows

$$\begin{bmatrix}\mathbf{I} & \mathbf{K}^+\mathbf{\Pi U}^-\\ \mathbf{\Pi U}^{-^T}\mathbf{K}^+ & 0\end{bmatrix}\begin{bmatrix}\mathbf{\Pi U}^-\\ \lambda\end{bmatrix}=\begin{bmatrix}\mathbb{P}\mathbf{U}^-\\ 2E_p^-\end{bmatrix}. \tag{6.3.75}$$

It is clear that the Eq.(6.3.75) is the nonlinear equation with respect to $\mathbf{\Pi U}^-$. It is intended to apply the fixed-point iteration method [167] to address this problem. In this scheme, firstly the Eq.(6.3.75) is rewritten as

$$\begin{bmatrix}\mathbf{I} & \mathbf{K}^+\mathbf{\Pi U}_i^-\\ \mathbf{\Pi U}_i^{-^T}\mathbf{K}^+ & 0\end{bmatrix}\begin{bmatrix}\mathbf{\Pi U}_{i+1}^-\\ \lambda_{i+1}\end{bmatrix}=\begin{bmatrix}\mathbb{P}\mathbf{U}^-\\ 2E_p^-\end{bmatrix}, \tag{6.3.76}$$

where $i=0,1,2\ldots$ is the index. Solving the Eq.(6.3.76) directly by inverting the matrix $\begin{bmatrix}\mathbf{I} & \mathbf{\Pi U}_i^-\\ \mathbf{\Pi U}_i^{-^T}\mathbf{K}^+ & 0\end{bmatrix}$ is heavily computational consuming, and to avoid this,

we firstly acquire the $\lambda_{i+1}$ by following

$$\lambda_{i+1} = \frac{2E_p^- - \boldsymbol{\mathcal{A}}_i^T \mathbb{P}\mathbf{U}^-}{\boldsymbol{\mathcal{A}}_i^T \boldsymbol{\mathcal{A}}_i}, \tag{6.3.77}$$

where $\boldsymbol{\mathcal{A}}_{\boldsymbol{i}} = \mathbf{K}^+ \boldsymbol{\Pi}\mathbf{U}_i^-$. As $\boldsymbol{\mathcal{A}}_i^T \mathbb{P}\mathbf{U}^-$ and $\boldsymbol{\mathcal{A}}_i^T \boldsymbol{\mathcal{A}}_i$ are both a scalar, it is cheap to compute $\lambda_{i+1}$. Subsequently, it is also easy to obtain $\boldsymbol{\Pi}\mathbf{U}_{i+1}^-$ by

$$\boldsymbol{\Pi}\mathbf{U}_{i+1}^- = \mathbb{P}\mathbf{U}^- - \boldsymbol{\mathcal{A}}_i^T \lambda_{i+1}. \tag{6.3.78}$$

The initial value is chosen as $\boldsymbol{\Pi}\mathbf{U}_0^- = \mathbb{P}\mathbf{U}^-$. Repeat the process referred above until the the following criteria are satisfied:

$$\left| \frac{\lambda_{i+1} - \lambda_i}{\lambda_{i+1}} \right| \leqslant \tau_\lambda, \ \frac{\left\| \boldsymbol{\Pi}\mathbf{U}_{i+1}^- - \boldsymbol{\Pi}\mathbf{U}_i^- \right\|_{L2}}{\left\| \boldsymbol{\Pi}\mathbf{U}_{i+1}^- \right\|_{L2}} \leqslant \tau_{\boldsymbol{\Pi}\mathbf{U}^-}. \tag{6.3.79}$$

As in Newmark scheme, the velocity is independent to displacement, then the process to obtain the projection of control velocity $\boldsymbol{\Pi}\dot{\mathbf{U}}^-$ can follow the derivation for projected displacement above, by starting to construct the Lagrange function such that

$$\mathscr{L}(\boldsymbol{\Pi}\dot{\mathbf{U}}^-, \lambda) = \frac{1}{2} \left\| \boldsymbol{\Pi}\dot{\mathbf{U}}^- - \mathbb{P}\dot{\mathbf{U}}^- \right\|_{L2}^2 + \lambda \left[ \frac{1}{2} (\boldsymbol{\Pi}\dot{\mathbf{U}}^-)^T \mathbf{M}^+ \boldsymbol{\Pi}\dot{\mathbf{U}}^- - E_k^- \right], \tag{6.3.80}$$

where $\mathbf{M}^+$ is the mass matrix and $E_k^- = \frac{1}{2}(\dot{\mathbf{U}}^-)^T \mathbf{M}^- \dot{\mathbf{U}}^-$ is the kinetic energy. This fixed-point iterative projection methodology is summarized in the Algorithm 7. Furthermore, the E-STAGN technique is summarized in Algorithm 8. The adaptive refinement will not terminate until the relative error $e_{\text{rel}}$ fulfills

$$e_{\text{rel}} = \frac{\sum_{\tilde{m}=1}^{\tilde{M}} \Delta t_{\tilde{m}} E(\tilde{\boldsymbol{e}}, \tilde{\Omega}_{\tilde{m}}) + \frac{\Delta t_1}{2} E(\tilde{\boldsymbol{e}}, \tilde{\Omega}_0) + \frac{\Delta t_{\tilde{M}}}{2} E(\tilde{\boldsymbol{e}}, \tilde{\Omega}_{\tilde{M}})}{\sum_{\tilde{m}=1}^{\tilde{M}} \Delta t_{\tilde{m}} E(\tilde{\boldsymbol{u}}, \tilde{\Omega}_{\tilde{m}}) + \frac{\Delta t_1}{2} E(\tilde{\boldsymbol{u}}, \tilde{\Omega}_0) + \frac{\Delta t_{\tilde{M}}}{2} E(\tilde{\boldsymbol{u}}, \tilde{\Omega}_{\tilde{M}})} \leqslant \tau, \tag{6.3.81}$$

### 6.3.3 The DWR-STAGN method

The DWR-STAGN devised in this thesis, is a goal-oriented methodology that is designed to improve the refinement efficiency over a specific spatial area and (or) a certain temporal interval, and furthermore to enhance precision of QoI inside. In order to use this method, the local error estimator based on the DWR method should be derived.

**Algorithm 7:** The fixed-point iterative projection methodology between two non-nested meshes.

**Input:** $\mathbb{P}\mathbf{U}^-, \mathbf{K}^+, E_p^-$ ; /* $\mathbb{P}\dot{\mathbf{U}}^-, \mathbf{M}^+, E_k^-$ for velocity */

**Output:** $\mathbf{\Pi U}^-$ ; /* $\mathbf{\Pi}\dot{\mathbf{U}}^-$ for velocity */

Set initial value $\mathbf{\Pi U}_0^- = \mathbb{P}\mathbf{U}^-$.

Compute $\lambda_1$ and $\mathbf{\Pi U}_1^-$ by Eq.(6.3.77) and Eq.(6.3.78).

$i = 0$.

**while** $\left|\dfrac{\lambda_{i+1} - \lambda_i}{\lambda_{i+1}}\right| > \tau_\lambda$ *or* $\dfrac{\left\|\mathbf{\Pi U}_{i+1}^- - \mathbf{\Pi U}_i^-\right\|_{L2}}{\left\|\mathbf{\Pi U}_{i+1}^-\right\|_{L2}} > \tau_{\mathbf{\Pi U}^-}$ **do**

$i = i + 1$.

Compute $\lambda_{i+1}$ and $\mathbf{\Pi U}_{i+1}^-$ by Eq.(6.3.77) and Eq.(6.3.78).

**end**

**Algorithm 8:** Adaptivity process for E-STAGN method.

**Input:** The initial ST mesh $\Omega^0 \times I^0, e_{\text{rel}}$.

**Output:** The final ST mesh $\Omega^n \times I^n$ and the solutions $\boldsymbol{u}^h(\Omega^n, I^n)$.

$n = 0$.

**while** $e_{rel} > \tau$ **do**

**for** $\mathcal{I} \leftarrow 1$ **to** $\mathcal{N}$ **do**

Compute the $e\big|_{\Omega_{(m-1,m)}^{(n)e,\mathcal{I}}}$ in Eq.(6.3.62).

**end**

Mark the elements based on Eq.(6.3.67) and Eq.(6.3.68) to be refined.

Refine $\Omega^n \times I^n$ mesh to be $\Omega^{n+1} \times I^{n+1}$ mesh.

$n = n + 1$.

Solve the coarse system in $\Omega^n \times I^n$ to obtain solution $\boldsymbol{u}^h(\Omega^n, I^n)$.

Solve the refined system in $\tilde{\Omega}^n \times \tilde{I}^n$ to obtain solution $\tilde{\boldsymbol{u}}(\tilde{\Omega}^n, \tilde{I}^n)$.

Obtain $e_{\text{rel}}$ in Eq.(6.3.81).

**end**

#### 6.3.3.1 The dual problem

We recall the first-order weak form in Eq.(6.2.29), such that

$$\mathbb{A}(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{B}(\boldsymbol{v}), \ \boldsymbol{v} \in \mathscr{V} \times \mathscr{V} \tag{6.3.82}$$

In order to establish DWR error estimation, it is necessary to tackle with a dual problem, which reads,

$$\mathbb{A}(\boldsymbol{v}, \boldsymbol{z}) = \mathbb{Q}(\boldsymbol{v}), \ \boldsymbol{v} \in \mathscr{V} \times \mathscr{V} \tag{6.3.83}$$

Where $\boldsymbol{z} = (\dot{z}, z) \in \mathscr{U} \times \mathscr{U}$ is the dual solution. $\mathbb{Q}(\boldsymbol{v})$ is the notation characterized by the expression as below

$$\mathbb{Q}(\boldsymbol{v}) := \sum_{m=1}^{M} \left[\mathscr{B}(\boldsymbol{v}; \boldsymbol{z}_{m-1}) + \mathscr{Q}(\boldsymbol{v}; \boldsymbol{p}^{\mathcal{Q}}_{M-m,M-m+1})\right] \Delta t_{M-m+1} + \mathscr{B}(\boldsymbol{v}; \boldsymbol{z}_0) \tag{6.3.84}$$

where $\mathscr{Q}$ is the function of QoI, and $\boldsymbol{p}^{\mathcal{Q}}_{M-m,M-m+1} = \left[\boldsymbol{f}^{\mathcal{Q}}_{M-m}, \boldsymbol{f}^{\mathcal{Q}}_{M-m+1}, \bar{\boldsymbol{f}}^{\mathcal{Q}}_{M-m}, \bar{\boldsymbol{f}}^{\mathcal{Q}}_{M-m+1}\right]$. $\boldsymbol{f}^{\mathcal{Q}}$ is the weight function of QoI and also can be understood as the source term. $\bar{\boldsymbol{f}}^{\mathcal{Q}}$ is the Neumann boundary condition for adjoint problem. The time step sequence is presented as $0 = t_0 < t_1 < \ldots < t_m < \ldots < t_M = T$.

According to Eq.(6.2.26), we can obtain the expression for left-hand side of the adjoint problem, such that

$$\mathbb{A}(\boldsymbol{v}, \boldsymbol{z}) := \sum_{m=1}^{M} \mathscr{A}(\boldsymbol{z}_m, \boldsymbol{v}) \Delta t_{M-m+1} + \mathscr{A}_0(\boldsymbol{z}(\boldsymbol{x}, T), \boldsymbol{v}). \tag{6.3.85}$$

Thus, the step-by-step formulation to solve the dual problem can be presented as

For $m = 0$, given $\boldsymbol{z}_0$, find $\boldsymbol{z}(\boldsymbol{x}, T)$ such that

$$\mathscr{A}_0(\boldsymbol{z}(\boldsymbol{x}, T), \boldsymbol{v}) = \mathscr{B}(\boldsymbol{v}; \boldsymbol{z}_0). \tag{6.3.86}$$

Given $\boldsymbol{z}_{m-1}$, find $\boldsymbol{z}_m$ such that

$$\mathscr{A}(\boldsymbol{z}_m, \boldsymbol{v}) = \mathscr{B}(\boldsymbol{v}; \boldsymbol{z}_{m-1}) + \mathscr{Q}(\boldsymbol{v}; \boldsymbol{p}^{\mathcal{Q}}_{M-m,M-m+1}). \tag{6.3.87}$$

Then following the derivation in Section 6.2, it is not difficult to find this weak form

of dual system is equivalent to the PDE of a wave propagation problem (backward in time), given as

$$\begin{aligned}
&\rho\ddot{z}(\boldsymbol{x},t) = \boldsymbol{\nabla}\boldsymbol{\sigma} + \boldsymbol{f}^{\mathcal{Q}}(\boldsymbol{x},t) \text{ in } \Omega \times I_z,\\
&z(\boldsymbol{x},t) = 0, \text{ on } \partial\Omega_D \times I_z,\\
&\partial_n z(\boldsymbol{x},t) = \bar{\boldsymbol{f}}^{\mathcal{Q}}(t), \text{ on } \partial\Omega_N \times I_z,\\
&z(\boldsymbol{x},T) = z_0, \text{ at } \Omega \times I_{z_0}.\\
&\dot{z}(\boldsymbol{x},T) = \dot{z}_0, \text{ at } \Omega \times I_{z_0}.
\end{aligned} \tag{6.3.88}$$

It is noted that $I_z = [0, T)$ is the time domain for dual system. It is slightly different from the time domain $I = (0, T]$ for primal system. Additionally, the condition at time termination for the primal problem is the dual initial condition $I_{z_0} = \{T\}$.

#### 6.3.3.2 Dual weighted residual and local ST error estimator

Suppose that $Q(\boldsymbol{u})$ is denoted as the function of QoI, where $\boldsymbol{u}$ serves as the primal solution. Assumed that the function $\mathbb{Q}(\boldsymbol{u})$ involves $Q(\boldsymbol{u})$, then the error representation $\mathbb{Q}(\boldsymbol{e})$ can be defined as below

$$\mathbb{Q}(\boldsymbol{e}) := \mathbb{Q}(\boldsymbol{u}) - \mathbb{Q}(\boldsymbol{u}^h) = \mathbb{A}(\boldsymbol{e}, \boldsymbol{z}) = \mathbf{R}(\boldsymbol{z}), \tag{6.3.89}$$

where $\mathbf{R}(\boldsymbol{z})$ is the dual weighted residual. $\boldsymbol{u}$ and $\boldsymbol{z}$ are the exact solutions for forward and dual problem respectively. Next, we define the error estimation such that

$$\mathbb{Q}(\tilde{\boldsymbol{e}}) = \mathbb{Q}(\tilde{\boldsymbol{u}}) - \mathbb{Q}(\boldsymbol{u}^h) = \mathbb{A}(\tilde{\boldsymbol{e}}, \tilde{\boldsymbol{z}}) = \mathbf{R}(\tilde{\boldsymbol{z}}), \tag{6.3.90}$$

where $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{z}}$ are solutions computed by a refined mesh $\tilde{\Omega} \times \tilde{I}$ (see Fig.6.2), and the estimated error is defined as $\tilde{\boldsymbol{e}} := \tilde{\boldsymbol{u}} - \boldsymbol{u}^h$. However, because the dual solution $\tilde{\boldsymbol{z}}$ may present a large oscillation amplitude, it is unfortunate that estimated DWR $\mathbf{R}(\tilde{\boldsymbol{z}})$ is not appropriate to be regarded as an error estimator. For the sake of alleviating this oscillation, an improved residual indicator $\mathbf{R}(\tilde{\boldsymbol{e}}_z)$ is introduced as below

$$\mathbf{R}(\tilde{\boldsymbol{e}}_z) := \mathbb{A}(\tilde{\boldsymbol{e}}, \tilde{\boldsymbol{e}}_z) = \mathbb{A}(\tilde{\boldsymbol{e}}, \tilde{\boldsymbol{z}} - \Pi\tilde{\boldsymbol{z}}), \tag{6.3.91}$$

where the operator $\Pi$ denotes the projection of $\tilde{\boldsymbol{z}}$ from a refined mesh $\tilde{\Omega}$ into a coarse mesh $\Omega$, based on the Algorithm 7.

Thus, the local ST error indicator can be defined such as

$$\hat{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\Omega^{e,\mathcal{I}}_{(m-1,m)}} := \hat{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\sum_{\mathcal{J}=1}^{4}\tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m}-1,\tilde{m})}} + \hat{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\sum_{\mathcal{J}=1}^{4}\tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m},\tilde{m}+1)}}, \tag{6.3.92}$$

where the information of space-time domains $\Omega^{e,\mathcal{I}}_{(m-1,m)}$ and $\tilde{\Omega}^{e,\mathcal{J}}_{(\cdot,\cdot)}$ can be found in the Fig.6.2. Furthermore, we define

$$\hat{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\sum_{\mathcal{J}=1}^{4}\tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m}-1,\tilde{m})}} := \sum_{j=1}^{4}\frac{\Delta t_{\tilde{m}}}{2}\left[\left|\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}-1},\tilde{\boldsymbol{z}}_{\tilde{m}-1}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}-1})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}-1}}\right| + \left|\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}},\tilde{\boldsymbol{z}}_{\tilde{m}}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}}}\right|\right], \tag{6.3.93}$$

$$\hat{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\sum_{\mathcal{J}=1}^{4}\tilde{\Omega}^{e,\mathcal{J}}_{(\tilde{m},\tilde{m}+1)}} := \sum_{j=1}^{4}\frac{\Delta t_{\tilde{m}+1}}{2}\left[\left|\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}},\tilde{\boldsymbol{z}}_{\tilde{m}}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}}}\right| + \left|\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}+1},\tilde{\boldsymbol{z}}_{\tilde{m}+1}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}+1})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}+1}}\right|\right], \tag{6.3.94}$$

where $|\cdot|$ means the absolute value, and

$$\begin{aligned}\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}-1},\tilde{\boldsymbol{z}}_{\tilde{m}-1}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}-1})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}-1}} = \big[&m(\dot{\tilde{u}}_{\tilde{m}-1}-\dot{u}^h_{m-1},\dot{\tilde{z}}_{\tilde{m}-1}-\Pi\dot{\tilde{z}}_{\tilde{m}-1})\\ &+\gamma\Delta t_{\tilde{m}-1}a(\dot{\tilde{u}}_{\tilde{m}-1}-\dot{u}^h_{m-1},\tilde{z}_{\tilde{m}-1}-\Pi\tilde{z}_{\tilde{m}-1})\\ &+m(\tilde{u}_{\tilde{m}-1}-u^h_{m-1},\tilde{z}_{\tilde{m}-1}-\Pi\tilde{z}_{\tilde{m}-1})\\ &+\beta\Delta t^2_{\tilde{m}-1}a(\tilde{u}_{\tilde{m}-1}-u^h_{m-1},\tilde{z}_{\tilde{m}-1}-\Pi\tilde{z}_{\tilde{m}-1})\big]\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}-1}},\end{aligned} \tag{6.3.95}$$

and

$$\begin{aligned}\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}},\tilde{\boldsymbol{z}}_{\tilde{m}}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}}} = \big[&m(\dot{\tilde{u}}_{\tilde{m}}-\dot{u}^h_{\tilde{m}},\dot{\tilde{z}}_{\tilde{m}}-\Pi\dot{\tilde{z}}_{\tilde{m}})\\ &+\gamma\Delta t_{\tilde{m}}a(\dot{\tilde{u}}_{\tilde{m}}-\dot{u}^h_{\tilde{m}},\tilde{z}_{\tilde{m}}-\Pi\tilde{z}_{\tilde{m}})\\ &+m(\tilde{u}_{\tilde{m}}-u^h_{\tilde{m}},\tilde{z}_{\tilde{m}}-\Pi\tilde{z}_{\tilde{m}})\\ &+\beta\Delta t^2_{\tilde{m}}a(\tilde{u}_{\tilde{m}}-u^h_{\tilde{m}},\tilde{z}_{\tilde{m}}-\Pi\tilde{z}_{\tilde{m}})\big]\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}}},\end{aligned} \tag{6.3.96}$$

and

$$\begin{aligned}\mathbb{A}(\tilde{\boldsymbol{e}}_{\tilde{m}+1},\tilde{\boldsymbol{z}}_{\tilde{m}+1}-\Pi\tilde{\boldsymbol{z}}_{\tilde{m}+1})\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}+1}} = \big[&m(\dot{\tilde{u}}_{\tilde{m}+1}-\dot{u}^h_{m},\dot{\tilde{z}}_{\tilde{m}+1}-\Pi\dot{\tilde{z}}_{\tilde{m}+1})\\ &+\gamma\Delta t_{\tilde{m}+1}a(\dot{\tilde{u}}_{\tilde{m}+1}-\dot{u}^h_{m},\tilde{z}_{\tilde{m}+1}-\Pi\tilde{z}_{\tilde{m}+1})\\ &+m(\tilde{u}_{\tilde{m}+1}-u^h_{m},\tilde{z}_{\tilde{m}+1}-\Pi\tilde{z}_{\tilde{m}+1})\\ &+\beta\Delta t^2_{\tilde{m}+1}a(\tilde{u}_{\tilde{m}+1}-u^h_{m},\tilde{z}_{\tilde{m}+1}-\Pi\tilde{z}_{\tilde{m}+1})\big]\big|_{\tilde{\Omega}^{e,j}_{\tilde{m}+1}}.\end{aligned} \tag{6.3.97}$$

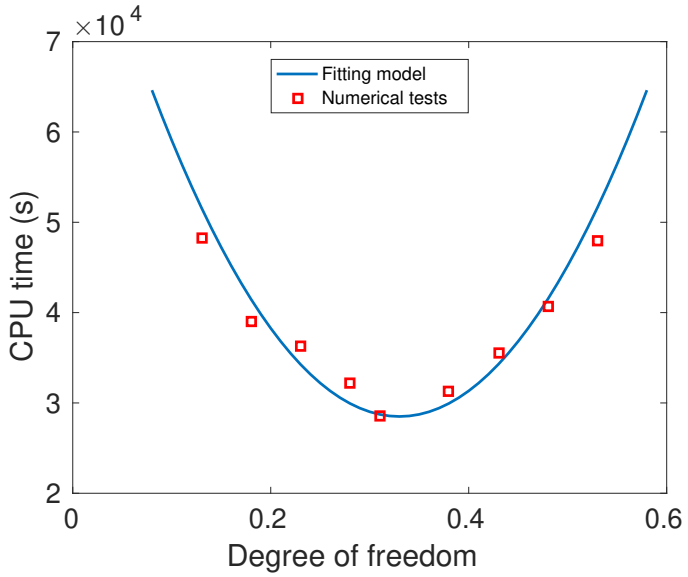



Figure 6.11: The numerical tests and the fitting model to identify the parameter $\eta$.

The $\boldsymbol{u}^h_{\tilde{m}}$ at mid-time step $t_{\tilde{m}}$ is obtained by the linear interpolation in Eq.(6.3.66).

Employ the strategy referred in Section 6.3.2.2 to mark elements to be discretized. Specifically, pick up the space-time elements ($\mathcal{I}^* = 1, 2, \ldots, \mathcal{N}^*$) fulfilling with

$$\forall \tilde{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\Omega^{e,\mathcal{I}^*}_{[0,T]}} \geqslant \forall \tilde{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\Omega^{e,\bar{\mathcal{I}}}_{[0,T]}} \text{ and } \sum_{\mathcal{I}^*=1}^{\mathcal{N}^*} \tilde{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\Omega^{e,\mathcal{I}^*}_{[0,T]}} \geqslant \eta \cdot \tilde{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\Omega\times I}, \tag{6.3.98}$$

and then refine.

The methodology to characterize the value of $\eta$ has been referred in Section 5.4.2 and Section 6.3.2.2. The method is analogous. Firstly, based on the distribution of numerical samples located in the range of $\eta = [0.1, 0.6]$ , a parabolic objective model described in Eq.(6.3.69) is established to predict the proper value of $\eta$ to minimize computational cost. Subsequently, fitting the data through the least square method quantifies the coefficients in Eq.(6.3.69) as presented in Fig.6.11, such as $a = 5.78 \times 10^5$, $\eta_0 \approx 0.33$ and $t_0 = 28500$. Consequently, $\eta = 0.33$ is chosen in this study.

The diagram of space-time adaptive mesh for DWR-STAGN technique is the same as that of E-STAGN method, presented in Fig.6.7. Therefore, it will also lead to the "Rothe Method" mesh style. Thus, when dealing with the primal or dual

problem, if the spatial meshes between two the successive time steps are not nested (see Fig.6.10), we will apply the projection approach in Algorithm 7. The refinement will terminate if the relative error of QoI meets the threshold, such as

$$e_{\mathrm{rel}}(Q) = \left| \frac{Q(\tilde{\boldsymbol{u}}) - Q(\boldsymbol{u}^h)}{Q(\tilde{\boldsymbol{u}})} \right| \leqslant \tau. \tag{6.3.99}$$

Overall, the DWR-STAGN technology can be summarized in Algorithm 9.

As mentioned before, it is worth noting that the determination for the values of $\tau$ in Eq.(6.3.81) and $\tau$ in Eq.(6.3.99) respectively is according to the requirement of engineering filed. As to the numerical experiments in this study, they are both assigned to be 0.5%.

**Algorithm 9:** Adaptivity process for DWR-STAGN method.

**Input:** The initial ST mesh $\Omega^0 \times I^0, e_{\mathrm{rel}}(Q)$.

**Output:** The final ST mesh $\Omega^n \times I^n$ and the solutions $\boldsymbol{u}^h(\Omega^n, I^n)$.

$n = 0$.

**while** $e_{rel}(Q) > \tau$ **do**

Solve the dual refined system in $\tilde{\Omega}^n \times \tilde{I}^n$ to obtain solution $\tilde{\boldsymbol{z}}(\tilde{\Omega}^n, \tilde{I}^n)$.

**for** $\mathcal{I} \leftarrow 1$ **to** $\mathcal{N}$ **do**

Compute the $\tilde{\mathbf{R}}(\tilde{\boldsymbol{e}}_z)\big|_{\Omega^{(n)e,\mathcal{I}}_{(m-1,m)}}$ in Eq.(6.3.92).

**end**

Mark the elements based on Eq.(6.3.98) to be refined.

Refine $\Omega^n \times I^n$ mesh to be $\Omega^{n+1} \times I^{n+1}$ mesh.

$n = n + 1$.

Solve the primal coarse system in $\Omega^n \times I^n$ to obtain solution $\boldsymbol{u}^h(\Omega^n, I^n)$.

Solve the primal refined system in $\tilde{\Omega}^n \times \tilde{I}^n$ to obtain solution $\tilde{\boldsymbol{u}}(\tilde{\Omega}^n, \tilde{I}^n)$.

Compute $e_{\mathrm{rel}}(Q)$ in Eq.(6.3.99).

**end**

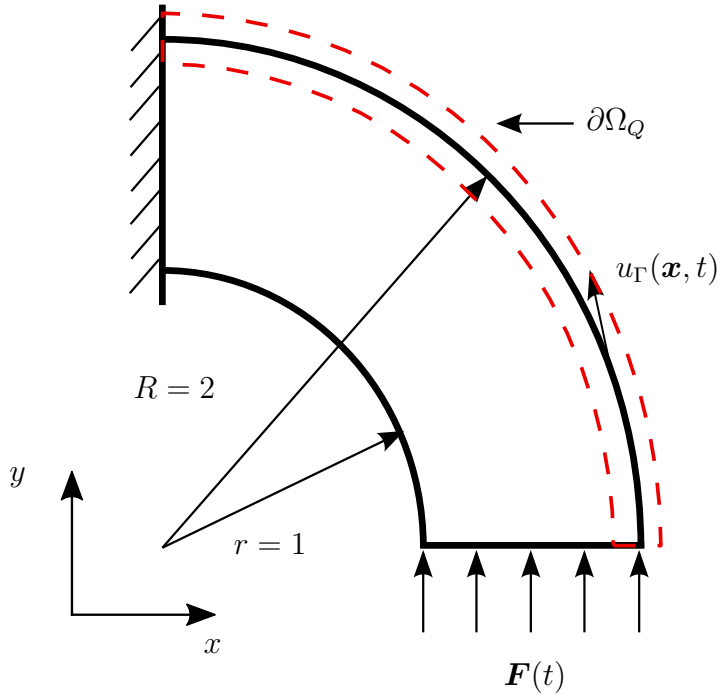



Figure 6.12: The illustration of 2D quarter annulus. The geometry is described by NURBS ($p=1, q=2$), and the analysis is processed within the bi-cubic PHT splines.

## 6.4 Numerical examples

In this section, two examples are presented. The first example is designed to compare the three proposed methodologies, such that, UM-STAGN, E-STAGN and DWR-STAGN techniques in a single-patch structure. In the second example, by comparing E-STAGN method against DWR-STAGN approach used in a multiple-patch structure, we examine the convergence of QoI.

### 6.4.1 2D quarter annulus

We begin with an example of a 2D quarter annulus illustrated in Fig.6.12. The annulus is clamped on one side, and is loaded by an uniform pressure on the other side. The Young's modulus is $E=1$, density is $\rho=1$ and Poisson's rate is $\nu=0.3$. The force $\boldsymbol{F}(t)$ is the function of time $t$ expressed as

$$\boldsymbol{F}(t) = f_0 \sin\omega_0 \pi t, (0 \leqslant t \leqslant \frac{T_f}{2}), \tag{6.4.100}$$

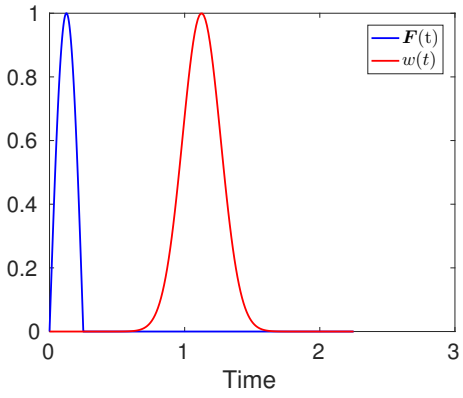



Figure 6.13: The function of force $\boldsymbol{F}(t)$ and weight $w(t)$.

where $T_f = 2\pi/\omega_0\pi$ is the period, and $f_0 = 1$ is the force constant.

The selection of value $\omega_0$ is crucial. If it is too small, then we would deal with a eigenvalue problem (vibration), in stead of transient dynamics. In order to select a suitable $\omega_0$, we firstly compute the natural frequency $\lambda_0$ of the first mode for free vibration of this structure by solving the following eigenproblem

$$(\mathbf{K} - \lambda^2\mathbf{M})\boldsymbol{\phi} = 0. \tag{6.4.101}$$

The obtained frequency is $\lambda_0 = 0.17$. Then we set $\omega_0 = 4$ and it yields

$$\lambda_f = \frac{\omega_0\pi}{2\pi} = 2 \approx 12\lambda_0. \tag{6.4.102}$$

Thus, we can make sure the frequency of load is high enough to trigger a wave propagation problem. The plot of force $\boldsymbol{F}(t)$ is presented in Fig.6.13.

To be simple, it is intended to create a quantity of interest where the term to be integrated can be directly converted into body force or Neumann condition. As a consequence, the following QoI is adopted, such that

$$Q(u) = \int_0^T \int_{\partial\Omega_Q} ue_x w(t) d\Gamma dt, \tag{6.4.103}$$

where $\partial\Omega_Q$ is the boundary domain of QoI, $u = (u_x, u_y)$ is the displacement, $e_x = (1, 0)^T$ is a vector, and $w(t)$ is a weighted factor with Gaussian function of time, defined by

$$w(t) = \exp\left[-\left(\frac{t - \bar{t}}{\bar{T}}\right)^2\right], \tag{6.4.104}$$

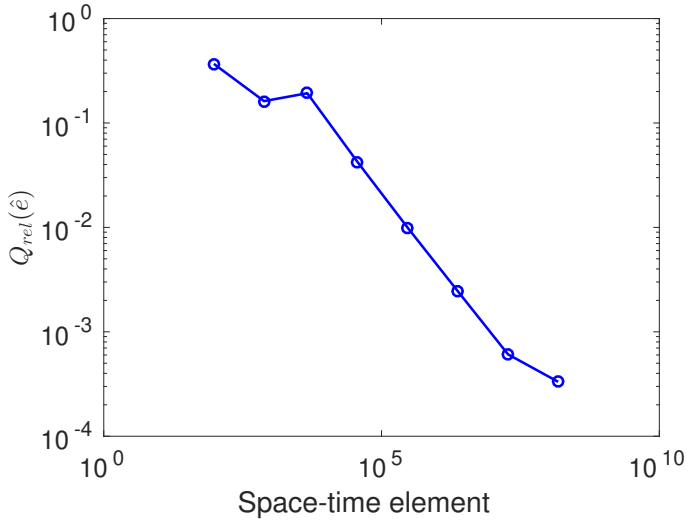



Figure 6.14: The convergence plot of relative error $Q_{\text{rel}}(\tilde{e})$ obtained by uniform space-time refinement for annular structure presented in Fig.6.12.

where $\bar{t}$ is the expected value and $\bar{T}$ is the square root of variance. In this case, we set $\bar{t} = 0.125$ and $\bar{T} = 0.05$, we can obtain the plot of weight function $w(t)$ in Fig.6.13. Compared with force function $\boldsymbol{F}(t)$, weight function $w(t)$ is somewhat flatter. We define the error estimation of QoI as $Q(\tilde{e}) = Q(\tilde{u}) - Q(u^h)$, and the "true" error of QoI as $Q(\hat{e}) = Q(\hat{u}) - Q(u^h)$. Here, $\tilde{u}$ is the solution deriving from the refined gird, while $\hat{u}$ is the solution obtained by a very fine uniform mesh, which is considered as the approximation of exact solution, i.e, $u \approx \hat{u}$. Then, we can acquire the relative form of $Q(\tilde{e})$ and $Q(\tilde{e})$, such that

$$Q_{\text{rel}}(\tilde{e}) = \left| \frac{Q(\tilde{e})}{Q(\tilde{u})} \right|, \quad Q_{\text{rel}}(\hat{e}) = \left| \frac{Q(\hat{e})}{Q(\hat{u})} \right|. \tag{6.4.105}$$

Here comes the issue that it is demanded to determine a reference QoI, $Q(\hat{u})$. In this thesis, the proposed method is to keep refining both space and time uniformly level by level and compute the solution within GIFT/Newmark scheme, until the relative error $Q_{\text{rel}}(\tilde{e})$ between two adjacent mesh levels satisfies the required tolerance (set by 0.05% in this example). Look at the convergence plot of $Q_{\text{rel}}(\tilde{e})$ in Fig.6.14. It can be observed that when $Q_{\text{rel}}(\tilde{e})$ reaches the threshold, the number of relevant ST cells arrives at 150994944 (262144 spatial elements and 576 time intervals).

The initial condition ($t_0 = 0$) is set to be $u_0 = 0$, $\dot{u}_0 = 0$ for primal problem so that the initial condition for dual problem ($t_M = T$) serves as $z_0 = 0$, $\dot{z}_0 = 0$

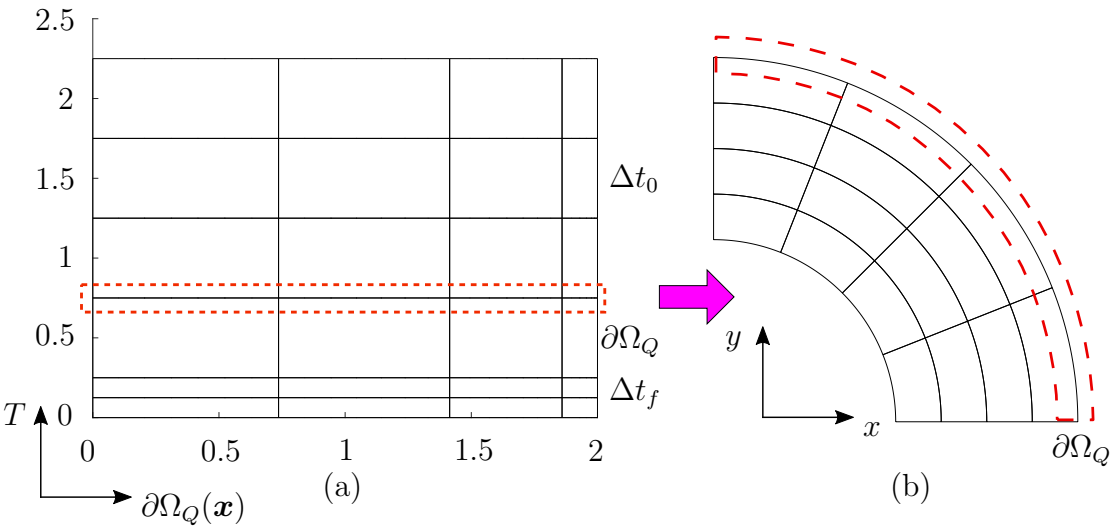



Figure 6.15: The initial space-time grid (a) and spatial mesh (b) of annulus. The horizontal axis in (a) denotes the edge $\partial\Omega_Q$.

relevantly according to the theory in Section 6.3.3. The initial space-time mesh is illustrated in Fig 6.15(a), and related spatial discretization is exhibited in Fig 6.15(b). Then the number of initial ST elements is 96, counted by 16 spatial elements by 6 time intervals. The initial time discretization is determined based on two principles : 1) the length of $\Delta t_f$ (see Fig.6.15(a)) is set as $2\Delta t_f = T_f/2$, aiming to guarantee the loading of $\boldsymbol{F}(t)$ is completed during the time period $[0, 2\Delta t_f]$; 2) the subsequent time gap $\Delta t_0$ is identified by $\Delta t_0 = h/v_0$, where $h$ is the mesh size defined in Eq.(6.2.13), and $v_0 = \sqrt{E/\rho}$ is the acoustic speed in the structure.

The primal and dual solutions are illustrated in Fig.6.16. As can be seen, the primal tangent displacement $u_\Gamma(\boldsymbol{x}, t)$ (with direction shown in Fig.6.12) appears at beginning of time, and arrives at the left side of the annulus at $t = 2.38$. While adjoint horizontal displacement $z_x(\boldsymbol{x}, t)$ rises around $t = 1.44$, when the weight function $w(t)$ starts to work, and then travels backward in time, reaching the side of the inner arc at $t = 0.75$.

The resulting representative spatial grids at different time steps produced by UM–STAGN, E–STAGN, DWR–STAGN methods respectively, are presented in Fig.6.17. The tangent primal stress $\sigma_\Gamma(\boldsymbol{x}, t)$ is presented in Fig.6.17(a1)-(a4). It is clear that the refinement obtained by UM–STAGN method almost perfectly tracks

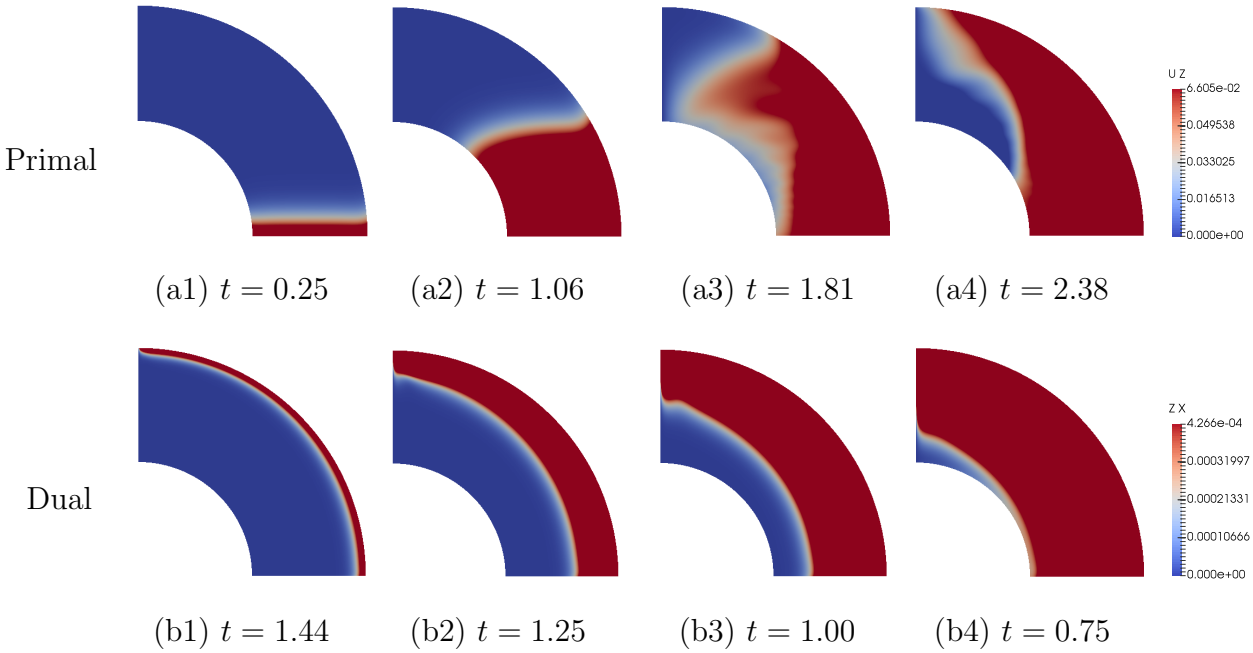



Figure 6.16: The primal solution (tangent displacement $u_\Gamma(\boldsymbol{x}, t)$) at different time steps (a1)-(a4), and dual results (horizontal displacement $z_x(\boldsymbol{x}, t)$) at different moments (b1-b4).

the propagating path of the stress (also see space-time mesh in Fig.6.18(a)). The Fig.6.18 should be interpreted combing with Fig.6.17. For example, since the refinement of UM–STAGN method is unidirectional, the element which is not refined at current level, will not be refined at following level any more. As a result, spatial mesh at south-west part of Fig.6.18(a) is very coarse since the stress wave does not reach the boundary of the annulus, as shown in Fig.6.17(b1)-(b2). In addition, compared with the finest mesh part (diagonal of Fig.6.18(a)), north-east portion is relatively coarse as well. That is because the width of stress wave is generally short in transient dynamics, as seen in Fig.6.17(a1)-(a4). As we discuss before, in UM–STAGN method, the elements wherein the errors satisfy the prescribed accuracy will not be computed anymore in the following adaptivity process, which is demonstrated in the Fig.6.18(a) that the initial coarse ST elements in time interval [0.25, 0.75] are never refined. Thus, it results in the block of the error information delivery over the whole domain. Consequently, although the relative error estimation $Q_{\text{rel}}(\tilde{e})$ can be convergent to good accuracy (about 0.5%), the "true" error $Q_{\text{rel}}(\hat{e})$ just stays at

roughly 10% can not reach a better accuracy.

In contrast, the E–STAGN overcomes this weakness via re-valuating the error of all the cells at every adaptive cycle, so that the error propagation over the whole space-time field can be taken into the account, which enables both $Q_{\mathrm{rel}}(\tilde{e})$ and $Q_{\mathrm{rel}}(\hat{e})$ get convergence to adequate value (1%), as seen in Fig.6.19. Nevertheless, if we take a further look at the plot, the convergence rate of E–STAGN approach finally get close to that of uniform refinement. This is because the error is accumulated with time in transient dynamics, the error detected by energy-based error estimation increases along with time history as well, leading to the space-time discretization performed in Fig.6.18(b). It results in the waste of refinement somehow as there are many pollutions of the error.

The goal-oriented adaptive method DWR–STAGN can well tackle this problem. The mesh not only catches the primal wave propagation, but also well trace the gradient of dual solution (horizontal dual stress $\sigma_x^z$) that is tightly associated with error estimation of QoI. Particularly, as seen in Fig.6.17(d3-d4), nearly not any refinement is made upon the basis of initial mesh for $t \geqslant 1.75$ (it is more obvious to look at the ST grids in Fig.6.18(c)), since the dual solution there is almost zero, in consistence with the weight function $w(t)$ related to QoI plotted in Fig.6.13, which indicates the fact that the quantity after $t \geqslant 1.75$ is out of our interest. If look at backward the time coordinate, that is from Fig.6.17(d4) to Fig.6.17(d1), the stress wave of dual solution starts from the out arc of annulus and travel to the bound of the inner arc. The refinement within DWR–STAGN well traces it. Furthermore, compared to the space-time refinement shown in Fig.6.18(a) and Fig.6.18(b), the ST refinement illustrated in Fig.6.18(c) at time period $t \in [0, 1]$ is more intensive. This implies that the error at time interval $t \in [0, 1]$ contributes to the error of QoI (occurring around $t \in [0.75, 1.5]$), and it is detected by the error estimation of DWR–STAGN. Then the problem of error accumulating along with time increase is addressed. Therefore, it is not surprised that the DWR-STAGN method delivers an optimal convergent rate, especially for $Q_{\mathrm{rel}}(\hat{e})$ in Fig.6.19(b).

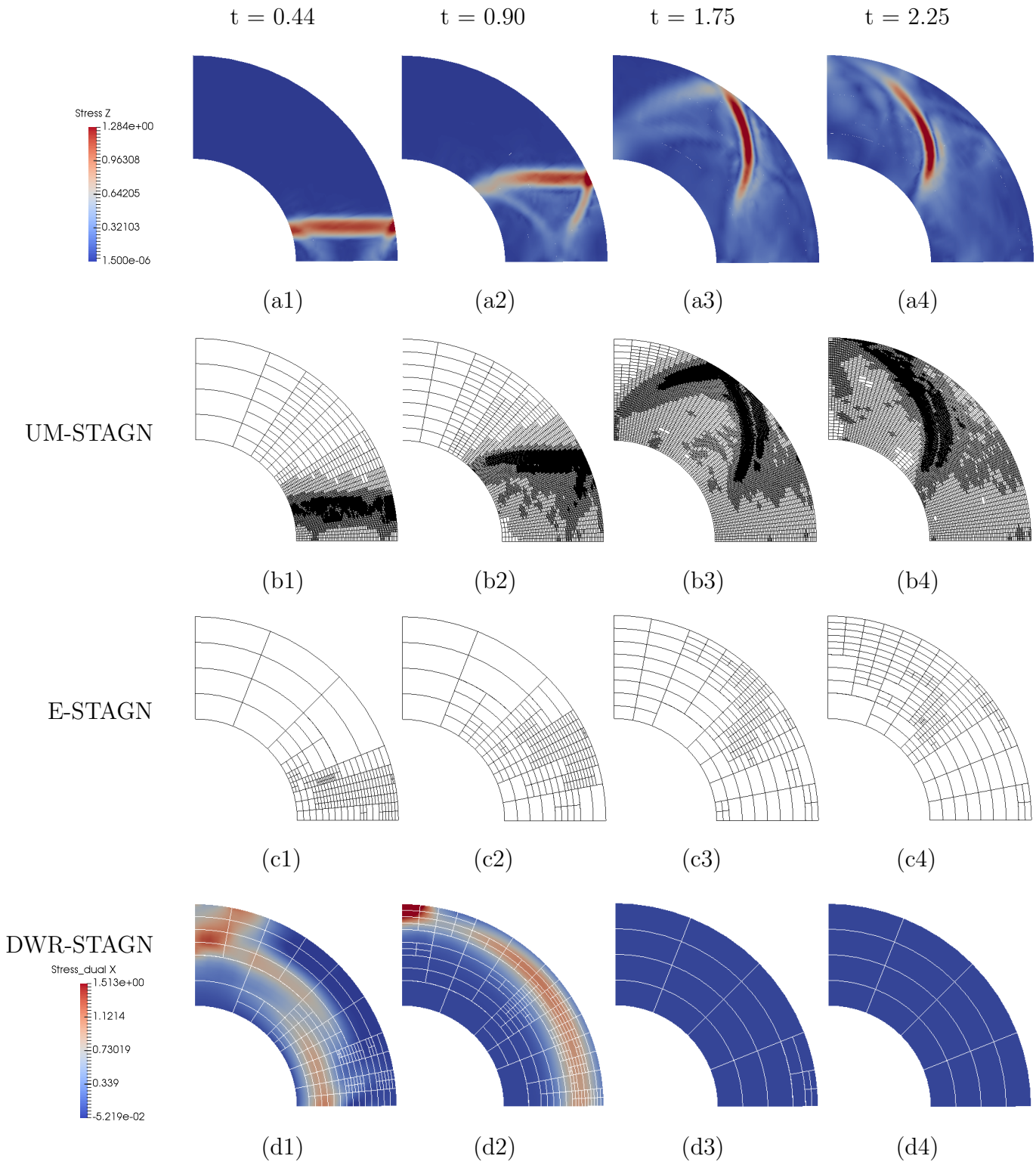



Figure 6.17: The propagation of tangent stress $\sigma_\Gamma$ in annulus structure a(1)–a(4). The spatial refinement at different time moments ($t = 0.44,\ 0.90,\ 1.75,\ 2.25$) obtained by the methods of UM-STAGN (b1-b4), E-STAGN (c1-c4) and DWR-STAGN (d1-d4) respectively. Especially, the travel of horizontal dual stress $\sigma_x^z$ is also presented in the structure (d1-d4).

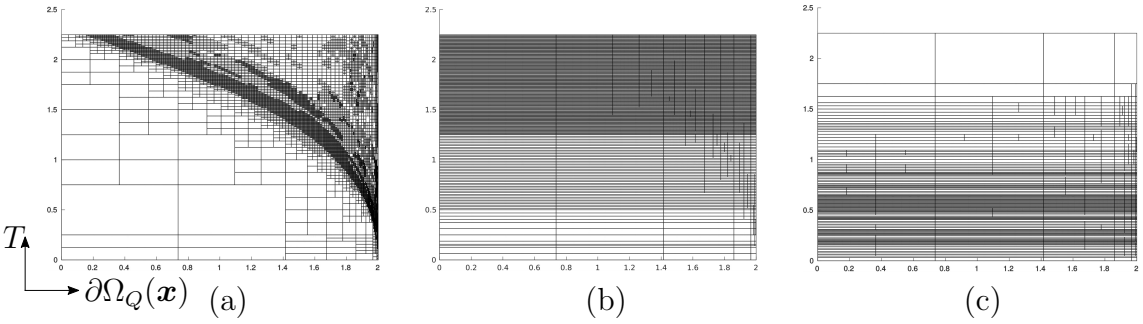



Figure 6.18: The space-time grids obtained within UM–STAGN (a), E–STAGN (b), DWR–STAGN (c), respectively. Specially, The horizontal axis denotes the edge $\partial\Omega_Q$, as also seen in Fig.6.15.

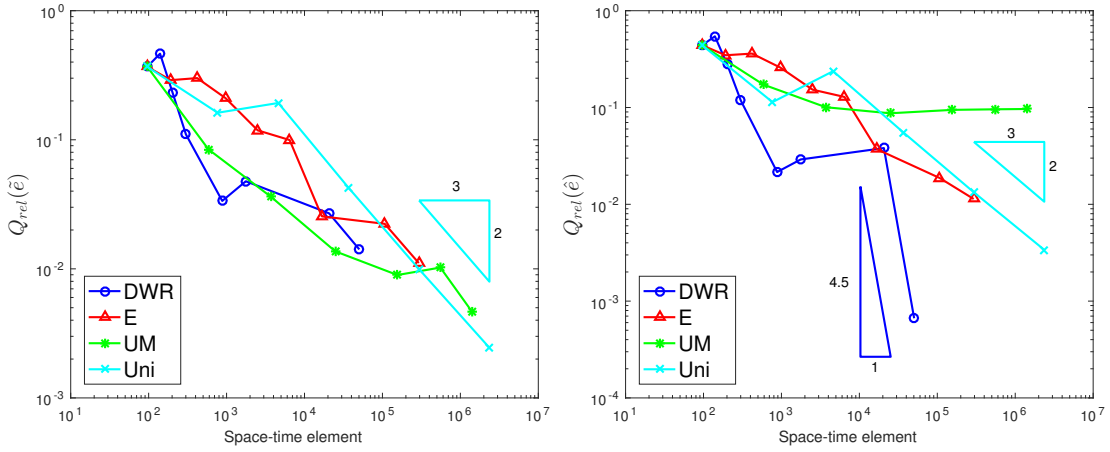



Figure 6.19: The convergence of relative error estimation $Q_{\text{rel}}(\tilde{e})$ (a) and "real" error $Q_{\text{rel}}(\hat{e})$ (b) obtained by UM–STAGN (UM), E–STAGN (E), DWR–STAGN(DWR) and uniform refinement (Uni).

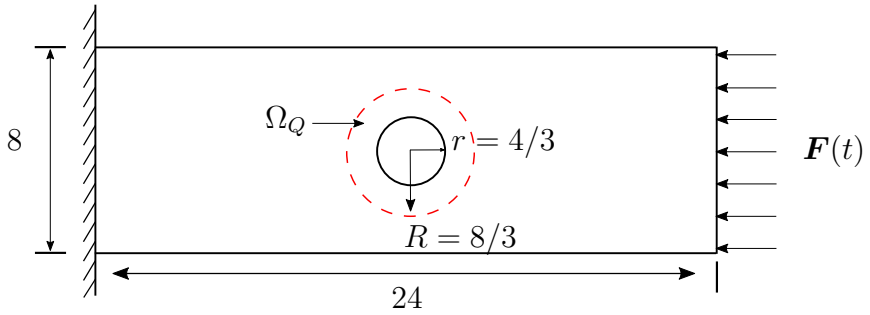



Figure 6.20: A 2D rectangle with a hole. The geometry is characterized by the bi-quadratic NURBS, and the solution filed is represented with PHT splines $p = 3, q = 3$.

### 6.4.2 Rectangle with a hole

Now, let us concern another goal function over a sub-domain in a 2D rectangle structure with a hole (see Fig.6.20), such that

$$Q(u) = \int_0^T \int_{\Omega_Q} u\Theta w(t) d\Omega dt, \ \Omega_Q \subset \Omega, \tag{6.4.106}$$

where $\Omega_Q$ is the area of QoI and $u = (u_x, u_y)$ is the displacement. Weight function $w(t)$ is expressed within Eq.(6.4.104), with assigned $\bar{T} = 0.2$ and $\bar{t} = 1.125$. The material constants are Young's modulus $E = 1$, density $\rho = 1$ and Poisson's rate $\nu = 0.3$. The uniform pressure load $\boldsymbol{F}(t)$ is in the form defined by Eq.(6.4.100). Like what we do in Section 6.4.1, the first natural frequency for the structure computed by Eq.(6.4.101) is $\lambda_0 = 0.013$. Then set $\omega_0 = 0.25$ and it yields an high-frequency impulsive load with $\lambda_f = \dfrac{\omega_0 \pi}{2\pi} = 0.125 \approx 10\lambda_0$. The plots of $w(t)$ and $\boldsymbol{F}(t)$ are displayed in Fig.6.21. The vector $\Theta$ is defined by following

$$\Theta = \begin{cases} (1,0)^T, \ \theta \in [-\dfrac{\pi}{4}, \dfrac{\pi}{4}], \\ (0,1)^T, \ \theta \in (\dfrac{\pi}{4}, \dfrac{3\pi}{4}), \\ (-1,0)^T, \ \theta \in [\dfrac{3\pi}{4}, \pi] \bigcup [-\pi, -\dfrac{3\pi}{4}], \\ (0,-1)^T, \theta \in (-\dfrac{3\pi}{4}, -\dfrac{\pi}{4}), \end{cases} \tag{6.4.107}$$

where the angle $\theta$ is presented in the Fig.6.22. As seen in Fig.6.22, the physical meaning of $Q(u)$ can be understood as the interest in the displacement field around

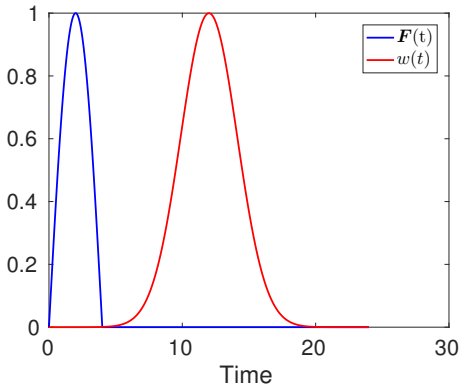



Figure 6.21: The function of force $\boldsymbol{F}(t)$ and weight $w(t)$ with respect to time.

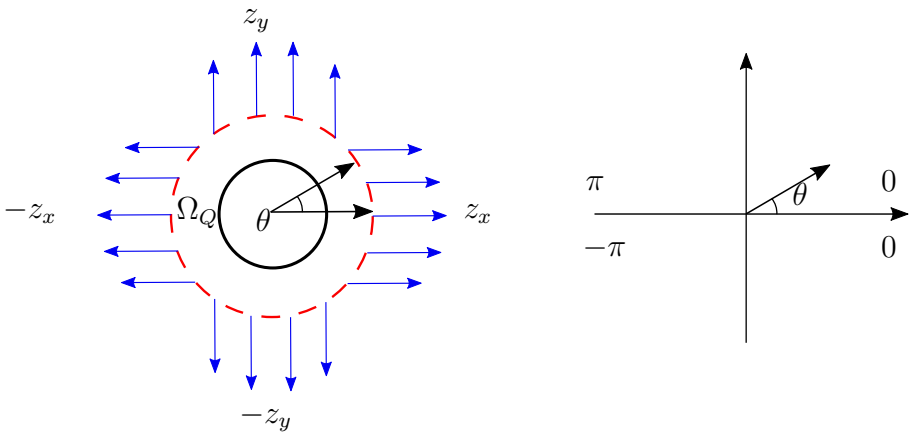



Figure 6.22: The illustration of the angle $\theta$ in Eq.(6.4.107), and the physical interpretation of QoI.

the vicinity of the hole.

The zero initial condition $u_0 = 0$, $\dot{u}_0 = 0$ is allocated so that the zero dual initial condition $z_0 = 0$, $\dot{z}_0 = 0$ at $t = T$ is generated. The initial space-time grid and the associated space discretization are preformed in Fig.6.23. There are 280 initial ST elements (40 space cells $\times$ 7 time intervals). Similar to the case described in Section 6.4.1, we allocate time intervals, like $\Delta t_f = T_f/4$, and $\Delta t_0 = h/v_0$.

As referred above, the UM-STAGN methods fails in getting convergence in an goal-oriented analysis. Therefore, in this example, we just compare the E-STAGN and DWR-STAGN methods. The simulation data are presented in Fig.6.24. We can observe the path of forward stress ($\sigma_x$) in the structure from Fig.6.24(a1)-(a4). It is shown that when the wave travels across the hole (at $t = 11$), it is split into

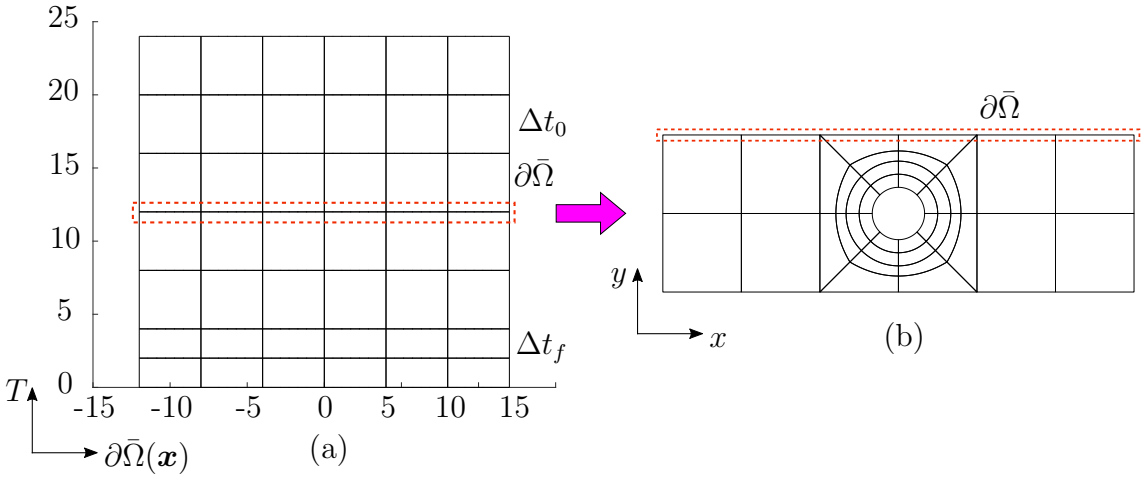



Figure 6.23: The initial space-time grid (a) and spatial mesh (b) of rectangle with a hole. The horizontal axis in (a) refers the edge $\partial\bar{\Omega}$.

two portions: a part of wave moves around the hole and keeps going forward, and the other part is reflected and then move backward (see the moment at $t = 15.3$). Afterwards, the wave propagation is actually composed by forward and backward waves and the forward wave reaches the left side of structure at $t = 24$. The hole barrier results in the wave localization around, which is also the motivation to create the goal function in Eq.(6.4.106). As discussed before, E-STAGN approach excellent traces the gradient of primal displacement, though, still unavoidably leads to the wasting of computational source. To be specific, E-STAGN is not able to identify the weight of the error so that it just tells refinement to discretize where the error is large. This scheme works well for static problem, though, it is not suitable to time-domain problem where the error accumulates along with time history. It indicates that the error generally increases with time period. Thus, within E-STAGN method, the refinement grids gets increasing intensive with the time movement, as seen in Fig.6.25(a). In this sense, if the quantity of interest locates at the end of the time, it is fine. However, if the QoI is at a specific time interval in the middle of time period (as presented in this example), then the adaptive refinement is not efficient enough. As a consequence, the resulting convergent speed of $Q_{\text{rel}}(\tilde{e})$ and $Q_{\text{rel}}(\hat{e})$ is not so fast as that obtained by DWR-STAGN method, as performed in Fig.6.26.

In contrast, the heuristic DWR-STAGN technique has a much better perfor-

mance. Firstly, for $t \geqslant 20$, none of refinement is made at all (see both Fig.6.24(b4) and Fig.6.25(b)), which is consistent to fact that the weight function $w(t)$ is almost zero after $t = 20$. More significantly, DWR-STAGN method precisely seizes the track of dual stress ($\sigma_x^z$) wave (see Fig.6.24(b1)-(b3)) to allow them to affect the QoI function $Q(\cdot)$. Hence, it is obvious to establish a more economic refinement strategy. The demonstration can be seen in Fig.6.26 that DWR-STAGN methodology delivers highest convergent rates in terms of both $Q_{\text{rel}}(\tilde{e})$ and $Q_{\text{rel}}(\hat{e})$ defined in Eq.(6.4.105), compared with E-STAGN and uniform refinement. Here, the approach to obtain "real" QoI $Q(\hat{u})$ expressed in Eq.(6.4.105) has been discussed in the Section 6.4.1. It is computed through an uniform GIFT/Newmark algorithm refined by 503316480 space time cells (655360 spacial elements by 768 time intervals), when it is less than the tolerance 0.05% set in this case, as shown in Fig.6.27.

Note that in context of DWR-STAGN method, the values of $Q_{\text{rel}}(\tilde{e})$ and $Q_{\text{rel}}(\hat{e})$ experience a slight raise (for the range of $[10^3, 10^5]$ of space-time element, as shown in Fig.6.19, and for the range of $[10^3, 10^4]$ of space-time element, as shown in Fig.6.26). It is acceptable and reasonable in author's opinion. Since the adaptivity within DWR-STAGN method leads to incompatible meshes between two neighboring time steps "Rothe Method" mesh type as mentioned before), it produces the error during the computation as the projection from a fine mesh into a coarse mesh results in the loss of information. Fortunately, the subsequent DWR-STAGN adaptivity strategy will detect the error and conduct the corresponding refinement automatically so that it can be seen that both $Q_{\text{rel}}(\tilde{e})$ and $Q_{\text{rel}}(\hat{e})$ achieve better accuracy after $10^4$ space-time elements for convergent plots in Fig.6.19 and Fig.6.26.

The comparison of CPU time obtained by E-STAGN, DWR-STAGN and uniform refinement are presented in Fig.6.28. As it can be seen, if the system is with certain degrees of freedom, the computation within DWR-STAGN is more expensive than E-STAGN and uniform refinement. It is not surprised because when using DWR-STAGN method, it is essential to solve a dual problem additionally. It is not surprised that using E-STAGN costs more CPU time than uniform refinement, either. That is because E-STAGN has to deal with the error estimation while uniform

refinement does not have to. Nevertheless, recall Fig.6.26 and it is clear that both $Q_{\text{rel}}(\tilde{e})$ and $Q_{\text{rel}}(\hat{e})$ achieve a better accuracy with the certain dofs, compared to E-STAGN and uniform refinement. Particularly, the last points in Fig.6.26 are related to the last points in Fig.6.28. Then it is obvious that it requires more dofs within E-STAGN and uniform refinement (much more dofs) than using DWR-STAGN to reach the same level of precision, and consequently it demands more computational efforts (much more CPU time by uniform refinement). Therefore, in this sense, the conclusion can be drawn that applying DWR-STAGN method is more efficient than E-STAGN and uniform refinement.

## 6.5 Conclusion

In this chapter, the linear elastic transient dynamics in the framework of GIFT/Newmark method was studied. To be precise, the GIFT scheme is employed for the discretization of spatial domain, while the time integration is tackled within the Newmark method. Based on the second-order type of Newmark algorithm, the first-order (or the so-called one-step) form formulations was derived, which is extremely productive to the DWR error estimation, as in this scheme, all the unknown variables locate at the left side of the equation.

Furthermore, three kinds of adaptivity techniques based upon hierarchical a posteriori error estimations were proposed, that is, UM–STAGN, E–STAGN, and DWR–STAGN. The error indicators of UM–STAGN and E–STAGN are energy-based style, while the error estimator for DWR–STAGN is built on the dual weighted residual.

The UM–STAGN method is a multi-level algorithm that gets rids of the element where the error is underneath the tolerance, and localizes the refined domain to be a new unsolved subsystem at each adaptation step. Although this method is able to grasp the propagation of solution gradient and keep reducing the computational domain efficiently, it fails to pass the error information over the whole field and thus it gets stuck in the convergence for the goal function. This problem is addressed

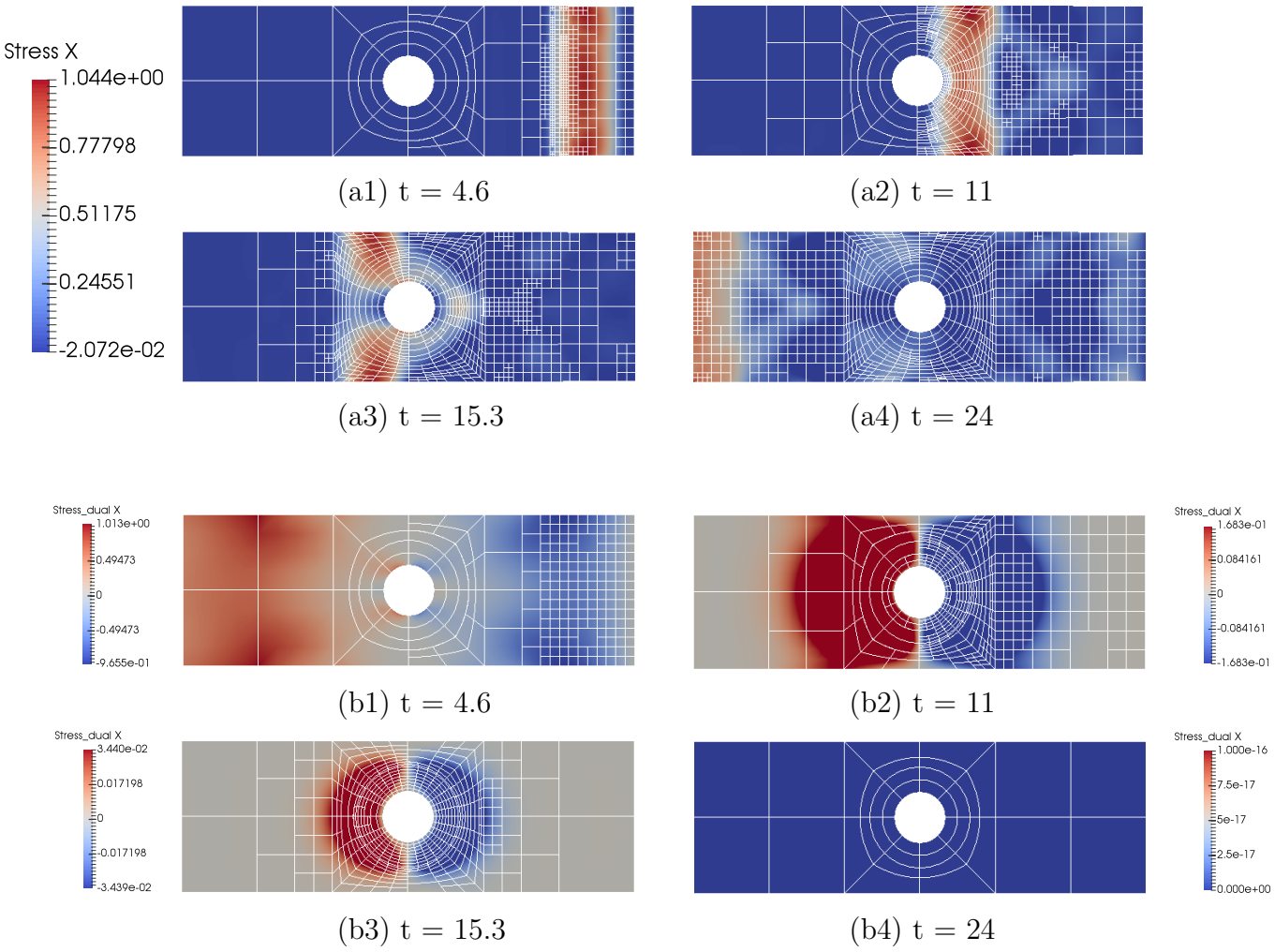



Figure 6.24: The wave propagation of $x$-direction stress $\sigma_x$ (a1-a4) and $x$-direction dual stress $\sigma_x^z$ (b1-b4) in the structure. The spatial refinement at different time steps ($t = 4.6,\ 11,\ 15.3,\ 24$) obtained by the methodology of E-STAGN (a1-a4) and DWR-STAGN (b1-b4) respectively.

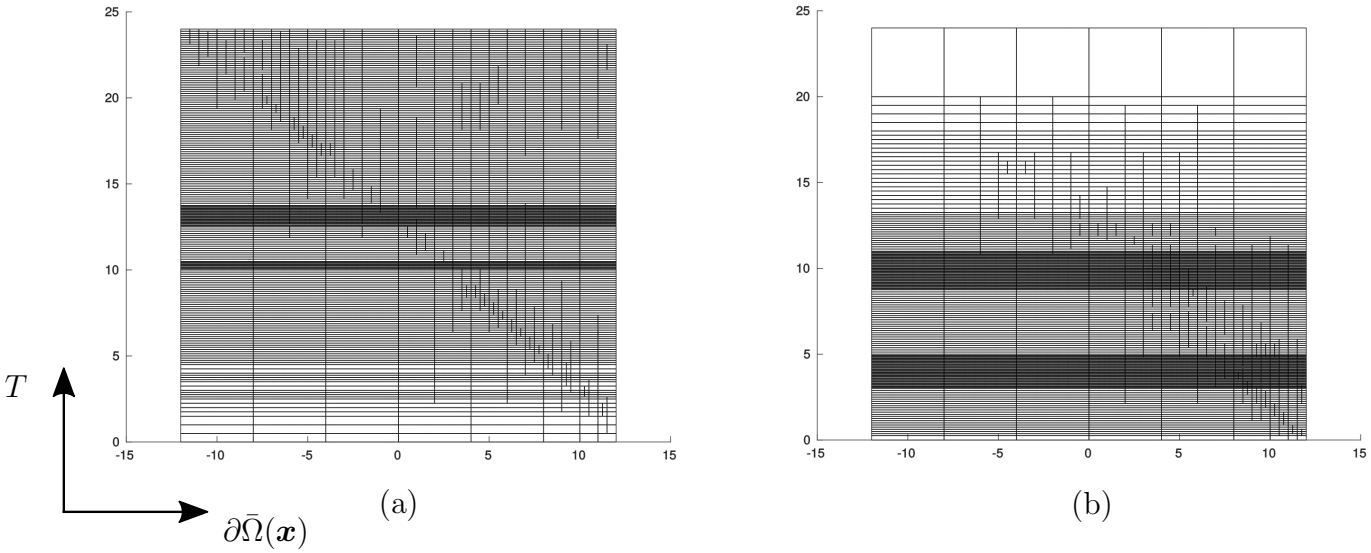



Figure 6.25: The space-time grids acquired by E–STAGN (a), DWR–STAGN (b), respectively. Specially, The horizontal axis in (a) refers the edge $\partial\bar{\Omega}$, as also seen in Fig.6.23.

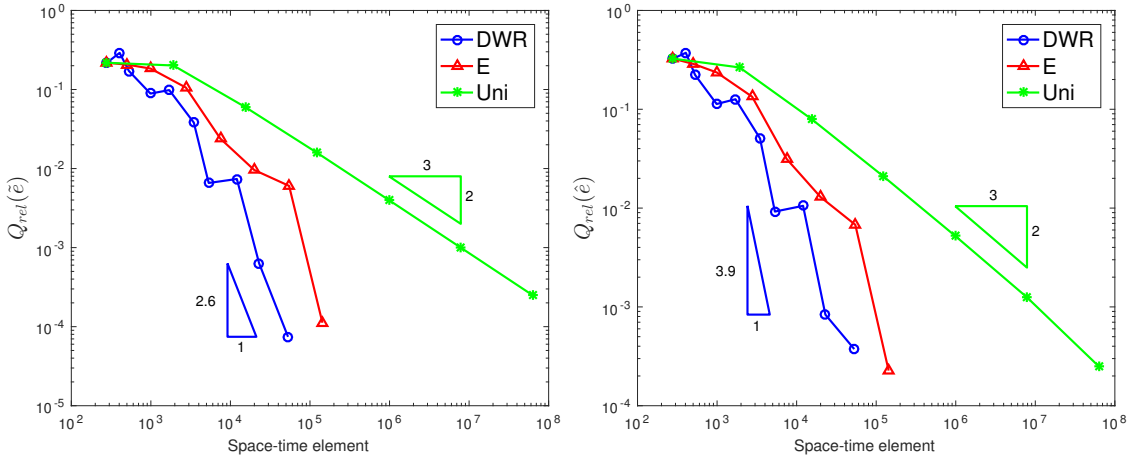



Figure 6.26: The Convergence of relative error estimation $Q_{\text{rel}}(\tilde{e})$ (a) and "real" error $Q_{\text{rel}}(\hat{e})$ (a) obtained by, E–STAGN (E), DWR–STAGN (DWR) and uniform refinement (Uni).

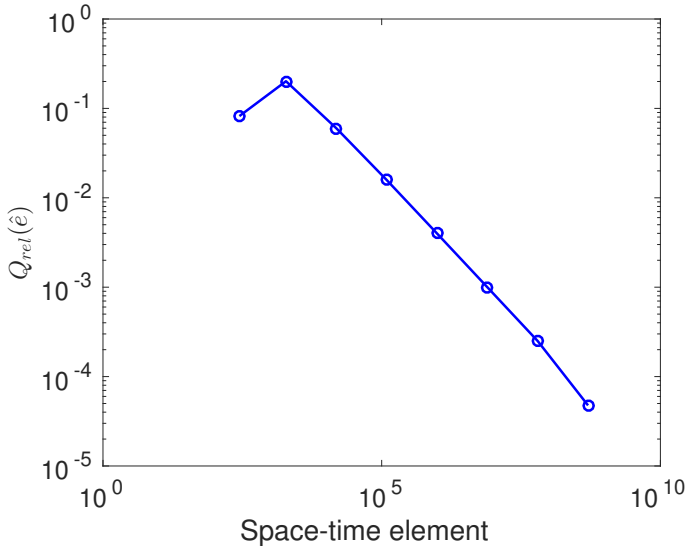



Figure 6.27: The convergence plot of relative error $Q_{\text{rel}}(\tilde{e})$ obtained by uniform space-time refinement for the structure presented in Fig.6.20.

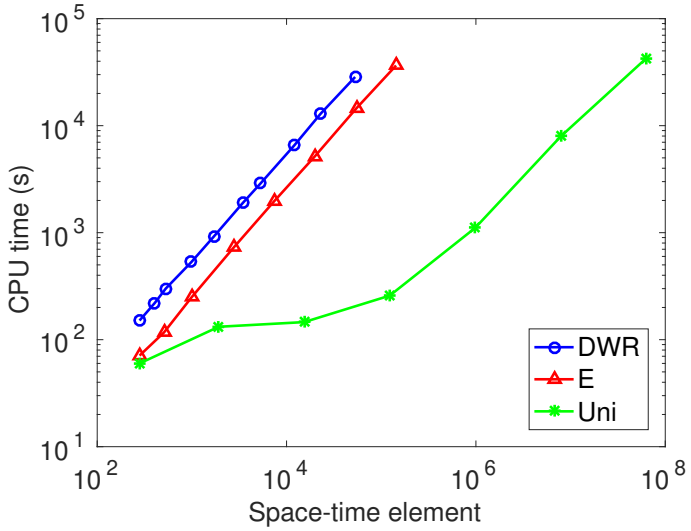



Figure 6.28: The comparison of CPU time obtained by E–STAGN (E), DWR–STAGN (DWR) and uniform refinement (Uni), respectively.

within the proposed E–STAGN methodology, wherein the error of each element will be reassessed by computing the whole system again in each adaptivity cycle. The E–STAGN also succeeds in tracking to path of gradient wave and moreover it achieves the convergence. Nevertheless, without considering the sensitivity of the error source, it is not economic sufficiently. The DWR-STAGN is designed to deal with this issue. Making the best of the first-order form and the theory of dual weighted residual, the established DWR-STAGN algorithm perfectly catches the travel laws of dual solution, and thus it allows to effectively control the error of QoI and certainly delivers the super convergent results. It is demonstrated in the numerical examples that convergence of DWR-STAGN is much faster than obtained by E–STAGN and uniform $h-$refinement GIFT/Newmark methods.

# Chapter 7

# Conclusion and outlook

In this thesis, a robust fully-automatic adaptivity technology was proposed to investigate the linear elastodynamics using isogeometric analysis. The contributions we made are summarized by following:

- The traditional generalized NURBS-based IGA softwares are limited to the generation of local refinement, due to the tensor-product construction of NURBS shape functions. Therefore, in Chapter 4, a GIFT methodology developed recently was introduced to fill in the gap. In GIFT, the geometric representation is described with NURBS given straightforward by the existing CAD software, and meanwhile the analysis is proceed by applying PHT-splines, which allows the local refinement.

- Exploiting this GIFT method, firstly the effort was made to study the vibration of Reissner-Mindlin plate in Chapter 5. An adaptivity algorithm that is dependent on hierarchical a *posteriori* error estimation regarding the modal shapes was initiated. The adaptation is proceeded within a mode-by-mode fashion in frequency domain, sweeping from lower to higher frequencies. The identification of modal correspondence between coarse and refined mesh is processed using MAC-manner method. The numerical examples demonstrate that the obtained convergent rate is higher than that by a uniform $h-$discretization, especially when the structure performs the local mechanical behavior.

- Afterwards, the study was moved to transient dynamics in Chapter 6. Three sorts of adaptive techniques relied on hierarchical a *posteriori* error estimation were established for the time domain problem, in the framework of GIFT/Newmark. They are UM–STAGN, E–STAGN and DWR–STAGN methods respectively.

  - The development of UM-STAGN method was inspired by the work [47, 157]. In UM-STAGN approach, the whole space-time system is split into numbers of multi-level subsystems during adaptivity procedure. Then these local systems are solved independently until the stopping criterion is satisfied. Due to the lack of capability of transferring error information across the separate subdomains, the method fails to achieve a suitable accuracy for the target function, although it is able to well record the trace of solution gradient.

  - The divergence problem above was addressed through the introduce of E–STAGN method, wherein the whole system is computed at every adaptation stage so that the error of each element is updated accordingly. Unfortunately, even though the energy-type error estimation is feasible to statics, it is unable to uncover the source of error in dynamics problem. As a consequence, the efficiency of refinement is not guaranteed.

  - Therefore, the heuristic DWR–STAGN technique was set up for the sake. Taking advantage of the our derived one-step Newmark formulation (where the unsolved variables all situate at the left side of equilibrium), and the dual weighted residual principle, the DWR–STAGN method succeeds in tracking the wave propagation route of dual solution gradient, and furthermore control the error of goal function powerfully through the detection of error sensitivity. Hence, DWR–STAGN, as a type of goal-oriented adaptivity, is able to generate the economical refinement . It is validated in the numerical tests that the DWR–STAGN method delivers an better observed convergence compared to that obtain by UM-STAGN, DWR–STAGN and uniform space-time $h-$refinement approaches.

As far as we are aware, the following potential improvements can be made in the near future:

- Regardless of the convergence problem exciting in UM-STAGN method, it is prospective since it provides the possibility to generate the incompatible (unstructured) time discretizations. It is critical since it can lower computational cost by avoiding the case that when refining a ST element, the whole time step has be to discretized, which inevitably creates extra degree of freedom in other ST elements not required to be refined (e.g., UM-STAGN, DWR–STAGN methods). More importantly, in UM-STAGN scheme, the refinement does not change the ratio of spatial element size and temporal element size, $h/\Delta t$. Thus if this ratio at initial mesh is properly chose (fulfilling the Courant–Friedrichs–Lewy (CFL) condition), then the stability would be always guaranteed during the adaptive computation. In addition, the subsystems generated during the refinement are independent so that the resolving process can be posed in the platform of parallel computation, which can remarkably enhance the efficiency.

  The main shortcoming of UM-STAGN method is the failure of message transfer across the individual subdomains. To overcome this flaw, it is intended to introduce a Lagrangian multiplier to hold the continuous condition between two neighboring multi-level subfields, and simultaneously adjust the information transformation at the boundary interface. The independent computations for solving two adjacent multi-level systems are iterative until the Lagrangian multiplier reaches convergence. The concept is analogous to multigrid algorithm somewhat, and moreover we will incorporate dual problem analysis inside. This work is currently being investigated and will be presented in a following paper.

- What is more, the patch-coupling method used in this work is a strong conforming imposition that is limited to be $C^0$ continuity at boundary between two patches. In order to improve this situation, in what follows, it aims to employ Nitsche method to raise the order of continuity at the interface of

patches. Thus, our proposed methodology can be applied for analysis of the thin plate and shell with multiple patches.

- In the thesis, $h-$ refinement was used. It is desired to develop $p-$refinement with for PHT-splines. As we know, the $k-$refinement , obtained by $p-$refinement firstly and $h-$refinement afterwards, is an unique optimal refinement strategy to achieve an higher order and continuity with less increase of degree of freedom for NURBS in IGA. It can be also duplicated to PHT-splines to facilitate the power of local refinement.

- Finally, we aim to extend our algorithm to the 3D complex structure more related to practical engineering applications. It is also desired to develop the methodology to be applicable to the nonlinear problem (like large deformation), which is more associated with real physical fields, such as civil engineering, biomechanics, and so on.

# Appendix A

# Prolongation of control variables from a coarse mesh to a refined mesh

Let $\boldsymbol{\phi}^h$ be the solution on a *coarse mesh* $\mathbb{T}$, which is discretized by PHT-spline basis functions such that $\boldsymbol{\phi}^h = \boldsymbol{T}\bar{\boldsymbol{\phi}}^h$, where $\bar{\boldsymbol{\phi}}^h$ are control variables. Now we aim to find the prolongation of $\bar{\boldsymbol{\phi}}^h$, namely, $\mathbb{P}\bar{\boldsymbol{\phi}}^h$, on a *refined mesh* $\tilde{\mathbb{T}}$. Two methods are presented as follows.

## A.1 Update control variables in hierarchical refinement

since the coarse mesh $\mathbb{T}$ is the subclass mesh of refined mesh $\tilde{\mathbb{T}}$, that is, $\mathbb{T} \subset \tilde{\mathbb{T}}$, by recalling the method to create new control points referred in Section 4.3, the construction of projection is able to be understood as the generation of updated control points similarly. Specifically, assume that $\Delta\bar{\boldsymbol{\phi}}^h$ denotes as the added control variables due to the increase of the degrees of freedom caused by refinement. Based on the method discussed in Section 4.3, it reads

$$\Delta\bar{\boldsymbol{\phi}}^h = (\underline{\underline{\mathbb{T}}})^{-1}\mathscr{G}\boldsymbol{\phi}^h. \tag{A.1.1}$$

then the prolongation yields

$$\mathbb{P}\bar{\boldsymbol{\phi}}^h = \bar{\boldsymbol{\phi}}^h + \Delta\bar{\boldsymbol{\phi}}^h. \tag{A.1.2}$$

## A.2 Projection

Supposed that $\tilde{\boldsymbol{\phi}}^h = \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h$ is the projection of $\boldsymbol{\phi}^h$ from $\mathbb{T}$ onto $\tilde{\mathbb{T}}$, then the prolongation problem can also be understood that

Find $\mathbb{P}\bar{\boldsymbol{\phi}}^h$ such that

$$\max_{\mathbb{P}\bar{\boldsymbol{\phi}}^h} \left\| \boldsymbol{\phi}^h - \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h \right\|_m, \tag{A.2.1}$$

namely,

$$\delta_{\mathbb{P}\bar{\boldsymbol{\phi}}^h} \left\| \boldsymbol{\phi}^h - \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h \right\|_m^2 = 0, \tag{A.2.2}$$

where the mass norm $\|\cdot\|_m$ is defined in Eq.(5.5.53). The term $\left\| \boldsymbol{\phi}^h - \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h \right\|_m^2$ can be extended to

$$\left\| \boldsymbol{\phi}^h - \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h \right\|_m^2 = \int_\Omega (\boldsymbol{T}\bar{\boldsymbol{\phi}}^h)^T \mathbf{m}\boldsymbol{T}\bar{\boldsymbol{\phi}}^h d\Omega - 2\int_\Omega (\tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h)^T \mathbf{m}\boldsymbol{T}\bar{\boldsymbol{\phi}}^h d\Omega + \int_\Omega (\tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h)^T \mathbf{m}\tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h d\Omega. \tag{A.2.3}$$

Then Eq.(A.2.2) is written by

$$\delta_{\mathbb{P}\bar{\boldsymbol{\phi}}^h} \left\| \boldsymbol{\phi}^h - \tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h \right\|_m^2 = \int_\Omega \tilde{\boldsymbol{T}}^T \mathbf{m}\tilde{\boldsymbol{T}}\mathbb{P}\bar{\boldsymbol{\phi}}^h d\Omega - \int_\Omega \tilde{\boldsymbol{T}}^T \mathbf{m}\boldsymbol{T}\bar{\boldsymbol{\phi}}^h d\Omega = 0. \tag{A.2.4}$$

Define mass matrices $\mathbf{M}_{\tilde{\boldsymbol{T}},\boldsymbol{T}} = \int_\Omega \tilde{\boldsymbol{T}}^T \mathbf{m}\boldsymbol{T}d\Omega$, $\mathbf{M}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{T}}} = \int_\Omega \tilde{\boldsymbol{T}}^T \mathbf{m}\tilde{\boldsymbol{T}}d\Omega$, and we obtain the prolongation as

$$\mathbb{P}\bar{\boldsymbol{\phi}}^h = \mathbf{M}^{-1}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{T}}}\mathbf{M}_{\tilde{\boldsymbol{T}},\boldsymbol{T}}\bar{\boldsymbol{\phi}}^h. \tag{A.2.5}$$

**Remark A.2.1** *When* $\mathbf{M}_{\tilde{\boldsymbol{T}},\boldsymbol{T}} = \int_\Omega \tilde{\boldsymbol{T}}^T(\boldsymbol{\xi})\mathbf{m}\boldsymbol{T}(\boldsymbol{\xi})d\Omega$ *is calculated, assume that the integral is conducted over* $\tilde{\mathbb{T}}$. *As the mappings* $\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi})$ *and* $\boldsymbol{x} = \tilde{\boldsymbol{F}}(\boldsymbol{\xi})$ *represent for* $\mathbb{T}$ *and* $\tilde{\mathbb{T}}$ *respectively, the term* $\boldsymbol{T}(\boldsymbol{\xi})$ *is obliged to be computed actually by*

$$\boldsymbol{T}(\xi) = \boldsymbol{T} \circ \tilde{\boldsymbol{F}}^{-1}[\boldsymbol{F}(\boldsymbol{\xi})]. \tag{A.2.6}$$

*Because of the geometry is preserved by isogeometric technique during the refinement, it is obtained, such as* $\tilde{\boldsymbol{F}}^{-1}[\boldsymbol{F}(\boldsymbol{\xi})] = \boldsymbol{\xi}$*, implying that* $\boldsymbol{T}(\xi)$ *can be obtained directly.*

**Remark A.2.2** *Noted that the prolongation algorithm illustrated in Appendix A.1 is limited to cubic PHT-spline basis functions, nevertheless, it is computationally economic. In comparison, because the computation for the inverse matrix* $\mathbf{M}^{-1}_{\tilde{\boldsymbol{T}},\tilde{\boldsymbol{T}}}$ *is inevitable, the method referred in Appendix A.2 requires more computational efforts. Regardless of that, it can be applied for any arbitrary degree of PHT-splines or other kinds of basis functions.*

# Appendix B

# Derivation of first-order form based on second-order Newmark method

In this appendix, the induction of process to acquire the first-order form in Eq.(6.2.22) and Eq.(6.2.23) is presented.

The $\mathscr{A}_0$ and $\mathscr{B}_0$ can be computed directly according to the initial condition (at time step $t_0 = 0 \in I_0 = \{0\}$) in Eq.(6.2.11), with $\mathscr{A}_0 = \mathscr{B}_0$. With respect to the remaining terms $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{F}$ over time domain $I = (0, T]$ (with time steps $0 < t_1 < \ldots < t_m < \ldots < t_M = T$) , it can be started with Eq.(6.2.19) at time steps $t_{m-1}$ and $t_m$ as follows

$$\begin{aligned} \dot{\mathbf{U}}_m &= \dot{\mathbf{U}}_{m-1} + (1-\gamma)\Delta t\ddot{\mathbf{U}}_{m-1} + \gamma\Delta t\ddot{\mathbf{U}}_m, \\ \mathbf{U}_m &= \mathbf{U}_{m-1} + \Delta t\dot{\mathbf{U}}_{m-1} + \frac{\Delta t^2}{2}(1-2\beta)\ddot{\mathbf{U}}_{m-1} + \beta\Delta t^2\ddot{\mathbf{U}}_m. \end{aligned} \tag{B.0.1}$$

Multiplying the Eq.(B.0.1) by mass matrix $\mathbf{M}$, it leads to

$$\begin{aligned} \mathbf{M}\dot{\mathbf{U}}_m &= \mathbf{M}\dot{\mathbf{U}}_{m-1} + (1-\gamma)\Delta t\mathbf{M}\ddot{\mathbf{U}}_{m-1} + \gamma\Delta t\mathbf{M}\ddot{\mathbf{U}}_m, \\ \mathbf{M}\mathbf{U}_m &= \mathbf{M}\mathbf{U}_{m-1} + \Delta t\mathbf{M}\dot{\mathbf{U}}_{m-1} + \frac{\Delta t^2}{2}(1-2\beta)\mathbf{M}\ddot{\mathbf{U}}_{m-1} + \beta\Delta t^2\mathbf{M}\ddot{\mathbf{U}}_m. \end{aligned} \tag{B.0.2}$$

using the equilibrium equation at time steps $t_{m-1}$ and $t_m$, it delivers

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{U}}_{m-1} &= \mathbf{F}_{m-1} - \mathbf{K}\mathbf{U}_{m-1}, \\ \mathbf{M}\ddot{\mathbf{U}}_m &= \mathbf{F}_m - \mathbf{K}\mathbf{U}_m. \end{aligned} \tag{B.0.3}$$

The occurrence of $\ddot{\mathbf{U}}_{m-1}$ and $\ddot{\mathbf{U}}_m$ will be vanished substituting Eq.(B.0.3) into Eq.(B.0.2), which yields

$$\begin{aligned}
\mathbf{M}\dot{\mathbf{U}}_m =& \mathbf{M}\dot{\mathbf{U}}_{m-1} + (1-\gamma)\Delta t(\mathbf{F}_{m-1} - \mathbf{K}\mathbf{U}_{m-1}) + \gamma\Delta t(\mathbf{F}_m - \mathbf{K}\mathbf{U}_m),\\
\mathbf{M}\mathbf{U}_m =& \mathbf{M}\mathbf{U}_{m-1} + \Delta t\mathbf{M}\dot{\mathbf{U}}_{m-1} + \frac{\Delta t^2}{2}(1-2\beta)(\mathbf{F}_{m-1} - \mathbf{K}\mathbf{U}_{m-1})\\
&+ \beta\Delta t^2(\mathbf{F}_m - \mathbf{K}\mathbf{U}_m).
\end{aligned} \tag{B.0.4}$$

Multiply the first equation with $\delta\dot{\mathbf{U}}$ and the second equation with $\delta\mathbf{U}$ in Eq.(B.0.4), and make a simple transformation to obtain

$$\begin{aligned}
\underbrace{\delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}_m + \gamma\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{K}\mathbf{U}_m}_{\mathscr{A}_1} =& \underbrace{\delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}_{m-1} - (1-\gamma)\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{K}\mathbf{U}_{m-1}}_{\mathscr{B}_1}\\
&+ \underbrace{(1-\gamma)\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{F}_{m-1} + \gamma\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{F}_m}_{\mathscr{F}_1},\\
\underbrace{\delta\mathbf{U}^T\mathbf{M}\mathbf{U}_m + \beta\Delta t^2\cdot\delta\mathbf{U}^T\mathbf{K}\mathbf{U}_m}_{\mathscr{A}_2} =& \underbrace{\Delta t\cdot\delta\mathbf{U}^T\mathbf{M}\dot{\mathbf{U}}_{m-1} + \delta\mathbf{U}^T\mathbf{M}\mathbf{U}_{m-1} - \frac{\Delta t^2}{2}(1-2\beta)\cdot\delta\mathbf{U}^T\mathbf{K}\mathbf{U}_{m-1}}_{\mathscr{B}_2}\\
&+ \underbrace{\frac{\Delta t^2}{2}(1-2\beta)\cdot\delta\mathbf{U}^T\mathbf{F}_{m-1} + \beta\Delta t^2\cdot\delta\mathbf{U}^T\mathbf{F}_m}_{\mathscr{F}_2}.
\end{aligned} \tag{B.0.5}$$

Combine with two equations in Eq.(B.0.5) and recall Eq.(6.2.7), and it leads to

$$
\begin{aligned}
\mathscr{A}_1 + \mathscr{A}_2 &= \delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}_m + \gamma\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{K}\mathbf{U}_m + \delta\mathbf{U}^T\mathbf{M}\mathbf{U}_m + \beta\Delta t^2\cdot\delta\mathbf{U}^T\mathbf{K}\mathbf{U}_m \\
&= m(\delta\dot{u}, u_{m-1}) + \gamma\Delta t a(\delta\dot{u}, u_m) + m(\delta u, u_m) + \beta\Delta t^2 a(\delta u, u_m) \\
&= \mathscr{A}\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}, \begin{bmatrix}\dot{u}_m\\ u_m\end{bmatrix}\right), \\
\mathscr{B}_1 + \mathscr{B}_2 &= \delta\dot{\mathbf{U}}^T\mathbf{M}\dot{\mathbf{U}}_{m-1} - (1-\gamma)\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{K}\mathbf{U}_{m-1} + \Delta t\cdot\delta\mathbf{U}^T\mathbf{M}\dot{\mathbf{U}}_{m-1} + \delta\mathbf{U}^T\mathbf{M}\mathbf{U}_{m-1} \\
&\quad - \frac{\Delta t^2}{2}(1-2\beta)\cdot\delta\mathbf{U}^T\mathbf{K}\mathbf{U}_{m-1} = m(\delta\dot{u}, \dot{u}_{m-1}) - (1-\gamma)\Delta t a(\delta\dot{u}, u_{m-1}) \\
&\quad + \Delta t m(\delta u, \dot{u}_{m-1}) + m(\delta u, u_{m-1}) - \frac{\Delta t^2}{2}(1-2\beta)a(\delta u, u_{m-1}) \\
&= \mathscr{B}\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}; \begin{bmatrix}\dot{u}_{m-1}\\ u_{m-1}\end{bmatrix}\right), \\
\mathscr{F}_1 + \mathscr{F}_2 &= (1-\gamma)\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{F}_{m-1} + \gamma\Delta t\cdot\delta\dot{\mathbf{U}}^T\mathbf{F}_m + \frac{\Delta t^2}{2}(1-2\beta)\cdot\delta\mathbf{U}^T\mathbf{F}_{m-1} + \beta\Delta t^2\cdot\delta\mathbf{U}^T\mathbf{F}_m \\
&= (1-\gamma)\Delta t\left[(\delta\dot{u}, \boldsymbol{f}_{m-1}) + (\delta\dot{u}, \bar{\boldsymbol{f}}_{m-1})_{\partial\Omega_N}\right] + \gamma\Delta t\left[(\delta\dot{u}, \boldsymbol{f}_m) + (\delta\dot{u}, \bar{\boldsymbol{f}}_m)_{\partial\Omega_N}\right] \\
&\quad + \frac{\Delta t^2}{2}(1-2\beta)\left[(\delta u, \boldsymbol{f}_{m-1}) + (\delta u, \bar{\boldsymbol{f}}_{m-1})_{\partial\Omega_N}\right] + \beta\Delta t^2\left[(\delta u, \boldsymbol{f}_m) + (\delta u, \bar{\boldsymbol{f}}_m)_{\partial\Omega_N}\right] \\
&= \mathscr{F}\left(\begin{bmatrix}\delta\dot{u}\\ \delta u\end{bmatrix}; \left[\boldsymbol{f}_{m-1}, \boldsymbol{f}_m, \bar{\boldsymbol{f}}_{m-1}, \bar{\boldsymbol{f}}_m\right]\right).
\end{aligned}
\tag{B.0.6}
$$

Thereby, expressions of operators $\mathscr{A}$, $\mathscr{B}$, $\mathscr{F}$, $\mathscr{A}_0$ and $\mathscr{B}_0$ in Eq.(6.2.23) are obtained with the formulation as follows

$$
\mathscr{A} = \mathscr{B} + \mathscr{F}. \tag{B.0.7}
$$

Finally, combine with the initial condition $\mathscr{A}_0 = \mathscr{B}_0$ mentioned above, the objective single-step first-order formulation in Eq.(6.2.23) can be readily recast, such as

$$
\mathscr{A} + \mathscr{A}_0 = \mathscr{B} + \mathscr{B}_0 + \mathscr{F}. \tag{B.0.8}
$$

# Bibliography

[1] T. Belytschko, A survey of numerical methods and computer programs for dynamic structural analysis, Nuclear Engineering and Design 37 (1) (1976) 23–34.

[2] J. G. Verwer, Explicit runge-kutta methods for parabolic partial differential equations, Applied Numerical Mathematics 22 (1-3) (1996) 359–379.

[3] J. C. Butcher, Numerical methods for ordinary differential equations, John Wiley & Sons, 2016.

[4] V. Acary, B. Brogliato, Implicit euler numerical scheme and chattering-free implementation of sliding mode systems, Systems & Control Letters 59 (5) (2010) 284–293.

[5] S. A. Teukolsky, Stability of the iterated crank-nicholson method in numerical relativity, Physical Review D 61 (8) (2000) 087501.

[6] C. W. Gear, L. R. Petzold, ODE methods for the solution of differential/algebraic systems, SIAM Journal on Numerical analysis 21 (4) (1984) 716–728.

[7] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method: Solid Mechanics, Vol. 2, Butterworth-heinemann, 2000.

[8] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Computers and structures 11 (1-2) (1980) 83–95.

[9] J. Wang, G. Liu, A point interpolation meshless method based on radial basis functions, International Journal for Numerical Methods in Engineering 54 (11) (2002) 1623–1648.

[10] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, Monthly notices of the royal astronomical society 181 (3) (1977) 375–389.

[11] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39) (2005) 4135–4195.

[12] Z. Liu, M. Majeed, F. Cirak, R. N. Simpson, Isogeometric FEM-BEM coupled structural-acoustic analysis of shells using subdivision surfaces, International Journal for Numerical Methods in Engineering 113 (9) (2018) 1507–1530.

[13] D. Benson, Y. Bazilevs, M. C. Hsu, T. Hughes, Isogeometric shell analysis: the Reissner–Mindlin shell, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 276–289.

[14] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, K.-U. Bletzinger, The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches, Computer Methods in Applied Mechanics and Engineering 199 (37) (2010) 2403–2416.

[15] H. Gomez, T. J. Hughes, X. Nogueira, V. M. Calo, Isogeometric analysis of the isothermal navier–stokes–korteweg equations, Computer Methods in Applied Mechanics and Engineering 199 (25-28) (2010) 1828–1840.

[16] Y. Bazilevs, V. M. Calo, Y. Zhang, T. J. Hughes, Isogeometric fluid–structure interaction analysis with applications to arterial blood flow, Computational Mechanics 38 (4-5) (2006) 310–322.

[17] H. Gómez, V. M. Calo, Y. Bazilevs, T. J. Hughes, Isogeometric analysis of the cahn–hilliard phase-field model, Computer methods in applied mechanics and engineering 197 (49-50) (2008) 4333–4352.

[18] C. V. Verhoosel, M. A. Scott, T. J. Hughes, R. De Borst, An isogeometric analysis approach to gradient damage models, International Journal for Numerical Methods in Engineering 86 (1) (2011) 115–134.

[19] P. Fischer, M. Klassen, J. Mergheim, P. Steinmann, R. Müller, Isogeometric analysis of 2d gradient elasticity, Computational Mechanics 47 (3) (2011) 325–334.

[20] A. Masud, R. Kannan, B-splines and NURBS based finite element methods for kohn–sham equations, Computer Methods in Applied Mechanics and Engineering 241 (2012) 112–127.

[21] J. A. Cottrell, T. J. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.

[22] T. J. Hughes, A. Reali, G. Sangalli, Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS, Computer Methods in Applied Mechanics and Engineering 197 (49-50) (2008) 4104–4124.

[23] A. Reali, An isogeometric analysis approach for the study of structural vibrations, Journal of Earthquake Engineering 10 (spec01) (2006) 1–30.

[24] J. Cottrell, A. Reali, Y. Bazilevs, T. Hughes, Isogeometric analysis of structural vibrations, Computer Methods in Applied Mechanics and Engineering 195 (44–43) (2006) 5257 – 5296.

[25] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method Volume 1 : The Basis, 2000.

[26] J. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, Computers & Mathematics with Applications 41 (5) (2001) 735 – 756.

[27] R. Becker, R. Rannacher, An optimal control approach to a *posteriori* error estimation in finite element methods, Acta Numerica 10 (2001) 1–102.

[28] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM transactions on graphics (TOG) 22 (3) (2003) 477–484.

[29] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, Y. Feng, Polynomial splines over hierarchical T-meshes, Graphical Models 70 (4) (2008) 76–86.

[30] N. Nguyen-Thanh, H. Nguyen-Xuan, S. Bordas, T. Rabczuk, Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids, Computer Methods in Applied Mechanics and Engineering 200 (21-22) (2011) 1892–1908.

[31] D. R. Forsey, R. H. Bartels, Hierarchical B-spline refinement, ACM Siggraph Computer Graphics 22 (4) (1988) 205–212.

[32] C. Giannelli, B. JüTtler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (7) (2012) 485–498.

[33] K. A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines, Computer Methods in Applied Mechanics and Engineering 269 (2014) 471–514.

[34] P. Dierckx, On calculating normalized Powell-Sabin B–splines, Computer Aided Geometric Design 15 (1) (1997) 61–78.

[35] N. Engleitner, B. Juettler, Patchwork B–spline refinement, Computer-Aided Design 90 (2017) 168–179.

[36] B. Marussig, J. Zechner, G. Beer, T.-P. Fries, Fast isogeometric boundary element method based on independent field approximation, Computer Methods in Applied Mechanics and Engineering 284 (2015) 458–488.

[37] E. Atroshchenko, S. Tomar, G. Xu, S. P. Bordas, Weakening the tight coupling between geometry and simulation in isogeometric analysis: From sub- and super-geometric analysis to geometry-independent field approximation (GIFT), International Journal for Numerical Methods in Engineering 114 (10) 1131–1159.

[38] D. Toshniwal, H. Speleers, T. J. Hughes, Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations, Computer Methods in Applied Mechanics and Engineering 327 (2017) 411–458.

[39] C. Anitescu, M. N. Hossain, T. Rabczuk, Recovery-based error estimation and adaptivity using high-order splines over hierarchical T-meshes, Computer Methods in Applied Mechanics and Engineering 328 (2018) 638–662.

[40] N. Nguyen-Thanh, J. Muthu, X. Zhuang, T. Rabczuk, An adaptive three-dimensional RHT-splines formulation in linear elasto-statics and elasto-dynamics, Computational Mechanics 53 (2) (2014) 369–385.

[41] E. Stein, M. Olavi Rüter, Finite element methods for elasticity with error-controlled discretization and model adaptivity, Encyclopedia of Computational Mechanics Second Edition (2018) 1–96.

[42] P. Ladevèze, F. Pled, L. Chamoin, New bounding techniques for goal-oriented error estimation applied to linear problems, International Journal for Numerical Methods in Engineering 93 (13) (2013) 1345–1380.

[43] W. Bangerth, M. Geiger, R. Rannacher, Adaptive Galerkin finite element methods for the wave equation, Computational Methods in Applied Mathematics 10 (1) (2010) 3–48.

[44] O. A. González-Estrada, E. Nadal, J. Ródenas, P. Kerfriden, S. P. A. Bordas, F. Fuenmayor, Mesh adaptivity driven by goal-oriented locally equilibrated superconvergent patch recovery, Computational Mechanics 53 (5) (2014) 957–976.

[45] R. J. Allemang, The Modal Assurance Criterion–twenty years of use and abuse, Journal of Sound and Vibration 37 (8) (2003) 14–23.

[46] M. Pastor, M. Binda, T. Harik, Modal Assurance Criterion, Procedia Engineering 48 (2012) 543 – 548.

[47] P. Cavin, A. Gravouil, A. Lubrecht, A. Combescure, Automatic energy conserving space–time refinement for linear dynamic structural problems, International Journal for Numerical Methods in Engineering 64 (3) (2005) 304–321.

[48] L. B. Da Veiga, A. Buffa, C. Lovadina, M. Martinelli, G. Sangalli, An isogeometric method for the Reissner–Mindlin plate bending problem, Computer Methods in Applied Mechanics and Engineering 209 (2012) 45–53.

[49] R. Deokar, D. Maxam, K. Tamma, A novel and simple a *posteriori* error estimator for lms methods under the umbrella of GSSSS framework: Adaptive time stepping in second-order dynamical systems, Computer Methods in Applied Mechanics and Engineering 334 (2018) 414 – 439.

[50] P. Yu, C. Anitescu, S. Tomar, S. P. A. Bordas, P. Kerfriden, Adaptive isogeometric analysis for plate vibrations: An efficient approach of local refinement based on hierarchical a *posteriori* error estimation, Computer Methods in Applied Mechanics and Engineering 342 (2018) 251 – 286.

[51] L. Chen, E. J. Lingen, R. de Borst, Adaptive hierarchical refinement of NURBS in cohesive fracture analysis, International Journal for Numerical Methods in Engineering 112 (13) (2017) 2151–2173.

[52] R. de Borst, T. J. Hughes, M. A. Scott, C. V. Verhoosel, Isogeometric failure analysis, in: Multiscale and Multiphysics Processes in Geomechanics, Springer, 2011, pp. 113–116.

[53] V. P. Nguyen, P. Kerfriden, S. P. Bordas, Two- and three-dimensional isogeometric cohesive elements for composite delamination analysis, Composites Part B: Engineering 60 (2014) 193 – 212.

[54] D. Kamensky, M. C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, T. J. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, Computer Methods in Applied Mechanics and Engineering 284 (2015) 1005–1053.

[55] Q. Hu, F. Chouly, P. Hu, G. Cheng, S. P. Bordas, Skew-symmetric Nitsche's formulation in isogeometric analysis: Dirichlet and symmetry conditions, patch coupling and frictionless contact, Computer Methods in Applied Mechanics and Engineering 341 (2018) 188 – 220.

[56] I. Temizer, P. Wriggers, T. Hughes, Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS, Computer Methods in Applied Mechanics and Engineering 209-212 (2012) 115 – 128.

[57] H. Lian, P. Kerfriden, S. Bordas, Shape optimization directly from CAD: An isogeometric boundary element approach using T-splines, Computer Methods in Applied Mechanics and Engineering 317 (2017) 1 – 41.

[58] H. Ghasemi, H. S. Park, T. Rabczuk, A level-set based IGA formulation for topology optimization of flexoelectric materials, Computer Methods in Applied Mechanics and Engineering 313 (2017) 239 – 258.

[59] G. Xu, T. H. Kwok, C. C. Wang, Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization, Computer-Aided Design 91 (2017) 1 – 13.

[60] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: Methods and comparison, Computer Methods in Applied Mechanics and Engineering 200 (23) (2011) 2021 – 2031.

[61] R. Simpson, Z. Liu, R. Vzquez, J. Evans, An isogeometric boundary element method for electromagnetic scattering with compatible B-spline discretizations, Journal of Computational Physics 362 (2018) 264 – 289.

[62] X. Peng, E. Atroshchenko, P. Kerfriden, S. Bordas, Isogeometric boundary element methods for three dimensional static fracture and fatigue crack growth, Computer Methods in Applied Mechanics and Engineering 316 (2017) 151 – 185.

[63] L. Piegl, W. Tiller, The NURBS book, Springer Science & Business Media, 2012.

[64] Y. Guo, M. Ruess, Nitsches method for a coupling of isogeometric thin shells and blended shell structures, Computer Methods in Applied Mechanics and Engineering 284 (2015) 881 – 905.

[65] Y. Guo, M. Ruess, D. Schillinger, A parameter-free variational coupling approach for trimmed isogeometric thin shells, Computational Mechanics 59 (4) (2017) 693–715.

[66] V. P. Nguyen, P. Kerfriden, M. Brino, S. P. Bordas, E. Bonisoli, Nitsches method for two and three dimensional NURBS patch coupling, Computational Mechanics 53 (6) (2014) 1163–1182.

[67] R. Schmidt, R. Wchner, K.-U. Bletzinger, Isogeometric analysis of trimmed NURBS geometries, Computer Methods in Applied Mechanics and Engineering 241-244 (2012) 93 – 111.

[68] E. Brivadis, A. Buffa, B. Wohlmuth, L. Wunderlich, Isogeometric mortar methods, Computer Methods in Applied Mechanics and Engineering 284 (2015) 292 – 319.

[69] M. Kapl, F. Buchegger, M. Bercovier, B. Jttler, Isogeometric analysis with geometrically continuous functions on planar multi-patch geometries, Computer Methods in Applied Mechanics and Engineering 316 (2017) 209 – 234.

[70] M. Kapl, V. Vitrih, Space of C2-smooth geometrically continuous isogeometric functions on two-patch geometries, Computers & Mathematics with Applications 73 (1) (2017) 37 – 59.

[71] A. Collin, G. Sangalli, T. Takacs, Analysis-suitable G1 multi-patch parametrizations for C1 isogeometric spaces, Computer Aided Geometric Design 47 (2016) 93 – 113.

[72] R. de Borst, L. Chen, The role of Bézier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement, International Journal for Numerical Methods in Engineering 113 (6) (2018) 999–1019.

[73] L. Chen, R. de Borst, Locally refined T–splines, International Journal for Numerical Methods in Engineering 114 (6) (2018) 637–659.

[74] L. Chen, R. de Borst, Adaptive refinement of hierarchical T–splines, Computer Methods in Applied Mechanics and Engineering 337 (2018) 220 – 245.

[75] A. V. Vuong, C. Giannelli, B. Jttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 200 (49) (2011) 3554 – 3567.

[76] D. Schillinger, L. Ded, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. Hughes, An isogeometric design–through–analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Computer Methods in Applied Mechanics and Engineering 249-252 (2012) 116 – 150.

[77] P. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical B-spline finite element method, Computer Methods in Applied Mechanics and Engineering 253 (2013) 584 – 598.

[78] C. Giannelli, B. Jttler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (7) (2012) 485 – 498.

[79] N. Engleitner, B. Jttler, Patchwork B-spline refinement, Computer-Aided Design 90 (2017) 168 – 179.

[80] H. Speleers, C. Manni, F. Pelosi, M. L. Sampoli, Isogeometric analysis with Powel–Sabin splines for advection–diffusion–reaction problems, Computer Methods in Applied Mechanics and Engineering 221-222 (2012) 132 – 148.

[81] F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis, International Journal for Numerical Methods in Engineering 47 (12) (2000) 2039–2072.

[82] R. Sanches, P. Bornemann, F. Cirak, Immersed B-spline (I-spline) finite element method for geometrically complex domains, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1432 – 1445.

[83] D. Sutula, P. Kerfriden, T. van Dam, S. P. Bordas, Minimum energy multiple crack propagation. Part I: Theory and state of the art review, Engineering Fracture Mechanics 191 (2018) 205 – 224.

[84] S. Claus, P. Kerfriden, A stable and optimally convergent LaTIn–CutFEM algorithm for multiple unilateral contact problems, International Journal for Numerical Methods in Engineering 113 (6) (2018) 938–966.

[85] M. Ainsworth, J. T. Oden, A *posteriori* error estimation in finite element analysis, Vol. 37, John Wiley & Sons, 2011.

[86] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a *posteriori* error estimates. Part 1: The recovery technique, International Journal for Numerical Methods in Engineering 33 (7) (1992) 1331–1364.

[87] L. F. Zeng, N. E. Wiberg, X. Li, Y. Xie, A *posteriori* local error estimation and adaptive time-stepping for newmark integration in dynamic analysis, Earthquake engineering & structural dynamics 21 (7) (1992) 555–571.

[88] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineerng analysis, International Journal for Numerical Methods in Engineering 24 (2) (1987) 337–357.

[89] E. Hinton, J. Campbell, Local and global smoothing of discontinuous finite element functions using a least squares method, International Journal for Numerical Methods in Engineering 8 (3) (1974) 461–480.

[90] J. Oden, H. Brauchli, On the calculation of consistent stress distributions in finite element approximations, International Journal for Numerical Methods in Engineering 3 (3) (1971) 317–325.

[91] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a *posteriori* error estimates. Part 2: Error estimates and adaptivity, International Journal for Numerical Methods in Engineering 33 (7) (1992) 1365–1382.

[92] I. Babuvška, W. C. Rheinboldt, Error estimates for adaptive finite element computations, SIAM Journal on Numerical Analysis 15 (4) (1978) 736–754.

[93] P. Ladeveze, D. Leguillon, Error estimate procedure in the finite element method and applications, SIAM Journal on Numerical Analysis 20 (3) (1983) 485–509.

[94] I. Babuška, W. C. Rheinboldt, A-*posteriori* error estimates for the finite element method, International Journal for Numerical Methods in Engineering 12 (10) (1978) 1597–1615.

[95] P. L. Baehmann, M. S. Shephard, J. E. Flaherty, A *posteriori* error estimation for triangular and tetrahedral quadratic elements using interior residuals, International Journal for Numerical Methods in Engineering 34 (3) (1992) 979–996.

[96] M. Paraschivoiu, J. Peraire, A. T. Patera, A *posteriori* finite element bounds for linear-functional outputs of elliptic partial differential equations, Computer Methods in Applied Mechanics and Engineering 150 (1-4) (1997) 289–312.

[97] F. Cirak, E. Ramm, A *posteriori* error estimation and adaptivity for linear elasticity using the reciprocal theorem, Computer Methods in Applied Mechanics and Engineering 156 (1-4) (1998) 351–362.

[98] S. Prudhomme, J. T. Oden, On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors, Computer Methods in Applied Mechanics and Engineering 176 (1-4) (1999) 313–331.

[99] N. Parés, P. Díez, A. Huerta, Exact bounds for linear outputs of the advection-diffusion-reaction equation using flux-free error estimates, SIAM journal on scientific computing 31 (4) (2009) 3064–3089.

[100] N. Parés, P. Díez, A. Huerta, Bounds of functional outputs for parabolic problems. Part I: Exact bounds of the discontinuous Galerkin time discretization, Computer Methods in Applied Mechanics and Engineering 197 (19-20) (2008) 1641–1660.

[101] N. Parés, P. Díez, A. Huerta, Bounds of functional outputs for parabolic problems. Part II: Bounds of the exact solution, Computer Methods in Applied Mechanics and Engineering 197 (19-20) (2008) 1661–1679.

[102] M. G. Larson, F. Bengzon, Adaptive finite element approximation of multiphysics problems, Communications in Numerical Methods in Engineering 24 (6) (2008) 505–521.

[103] M. G. Larson, R. Söderlund, F. Bengzon, Adaptive finite element approximation of coupled flow and transport problems with applications in heat transfer, International Journal for Numerical Methods in Fluids 57 (9) (2008) 1397–1420.

[104] J. Waeytens, L. Chamoin, P. Ladevéze, Guaranteed error bounds on pointwise quantities of interest for transient viscodynamics problems, Computational Mechanics 49 (3) (2012) 291–307.

[105] A. Schleupen, E. Ramm, Local and global error estimations in linear structural dynamics, Computers & Structures 76 (6) (2000) 741–756.

[106] D. Fuentes, D. Littlefield, J. T. Oden, S. Prudhomme, Extensions of goal-oriented error estimation methods to simulations of highly-nonlinear response

of shock-loaded elastomer-reinforced structures, Computer Methods in Applied Mechanics and Engineering 195 (37) (2006) 4659 – 4680.

[107] A. E. Taylor, D. C. Lay, Introduction to functional analysis, Vol. 2, Wiley New York, 1958.

[108] M. Kumar, T. Kvamsdal, K. A. Johannessen, Superconvergent patch recovery and a *posteriori* error estimation technique in adaptive isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 316 (2017) 1086–1156.

[109] M. Feischl, G. Gantner, D. Praetorius, Reliable and efficient a *posteriori* error estimation for adaptive IGA boundary element methods for weakly-singular integral equations, Computer Methods in Applied Mechanics and Engineering 290 (2015) 362–386.

[110] K. Van der Zee, C. Verhoosel, Isogeometric analysis-based goal-oriented error estimation for free-boundary problems, Finite Elements in Analysis and Design 47 (6) (2011) 600–609.

[111] J. Deng, F. Chen, Y. Feng, Dimensions of spline spaces over T-meshes, Journal of Computational and Applied Mathematics 194 (2) (2006) 267–283.

[112] D. Schillinger, L. Dede, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Computer Methods in Applied Mechanics and Engineering 249 (2012) 116–150.

[113] S. Shojaee, E. Izadpanah, N. Valizadeh, J. Kiendl, Free vibration analysis of thin plates by using a NURBS-based isogeometric approach, Finite Elements in Analysis and Design 61 (2012) 23–34.

[114] P. Sobota, W. Dornisch, R. Müller, S. Klinkel, Implicit dynamic analysis using an isogeometric Reissner–Mindlin shell formulation, International Journal for Numerical Methods in Engineering 110 (9) (2017) 803–825.

[115] C. H. Thai, H. Nguyen-Xuan, N. Nguyen-Thanh, T.-H. Le, T. Nguyen-Thoi, T. Rabczuk, Static, free vibration, and buckling analysis of laminated composite reissnermindlin plates using NURBS-based isogeometric approach, International Journal for Numerical Methods in Engineering 91 (6) (2012) 571–603.

[116] C. Giannelli, B. Jüttler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (7) (2012) 485 – 498.

[117] C. Giannelli, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon, J. Špeh, THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 299 (2016) 337–365.

[118] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, ACM Transactions on Graphics 23 (3) (2004) 276.

[119] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 229–263.

[120] G. Kirchoff, Uber das gleichgewicht und die bewegung einer elastischen scheibe, Journal fur die reine und angewandte Mathematik (Crelle's Journal) 40 (1850) 51–88.

[121] E. Reissner, The effect of transverse shear deformation on the bending of elastic plates, J. appl. Mech. (1945) A69–A77.

[122] R. D. Mindlin, Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates, J. appl. Mech. 18 (1951) 31–38.

[123] A. Labuschagne, N. J. van Rensburg, A. Van Der Merwe, Vibration of a Reissner–Mindlin–Timoshenko plate–beam system, Mathematical and Computer Modelling 50 (7-8) (2009) 1033–1044.

[124] Y. Liu, Y. Hon, K. Liew, A meshfree hermite-type radial point interpolation method for Kirchhoff plate problems, International Journal for Numerical Methods in Engineering 66 (7) (2006) 1153–1178.

[125] S. Fernández-Méndez, A. Huerta, Imposing essential boundary conditions in mesh-free methods, Computer Methods in Applied Mechanics and Engineering 193 (12) (2004) 1257–1275.

[126] C. Chan, C. Anitescu, T. Rabczuk, Isogeometric analysis with strong multipatch c1-coupling, Computer Aided Geometric Design 62 (2018) 294–310.

[127] W. Dörfler, A convergent adaptive algorithm for poissons equation, SIAM Journal on Numerical Analysis 33 (3) (1996) 1106–1124.

[128] G. Zhao, X. Du, W. Wang, B. Liu, H. Fang, Application of isogeometric method to free vibration of reissnermindlin plates with non-conforming multipatch, Computer-Aided Design 82 (2017) 127 – 139.

[129] H. Nguyen-Xuan, G. Liu, C. a. Thai-Hoang, T. Nguyen-Thoi, An edge-based smoothed finite element method (ES-FEM) with stabilized discrete shear gap technique for analysis of Reissner–Mindlin plates, Computer Methods in Applied Mechanics and Engineering 199 (9-12) (2010) 471–489.

[130] H. Nguyen-Xuan, T. Rabczuk, N. Nguyen-Thanh, T. Nguyen-Thoi, S. Bordas, A node-based smoothed finite element method with stabilized discrete shear gap technique for analysis of Reissner–Mindlin plates, Computational Mechanics 46 (5) (2010) 679–701.

[131] S. Lee, Free vibration analysis of plates by using a four-node finite element formulated with assumed natural transverse shear strain, Journal of sound and vibration 278 (3) (2004) 657–684.

[132] K. Liew, Y. Xiang, S. Kitipornchai, Transverse vibration of thick rectangular plates–I. comprehensive sets of boundary conditions, Computers & Structures 49 (1) (1993) 1–29.

[133] L. Bales, I. Lasiecka, Continuous finite elements in space and time for the nonhomogeneous wave equation, Computers & Mathematics with Applications 27 (3) (1994) 91 – 102.

[134] D. French, T. Peterson, A continuous space-time finite element method for the wave equation, Mathematics of Computation of the American Mathematical Society 65 (214) (1996) 491–506.

[135] M. J. Grote, A. Schneebeli, D. Schötzau, Discontinuous Galerkin finite element method for the wave equation, SIAM Journal on Numerical Analysis 44 (6) (2006) 2408–2431.

[136] C. Johnson, Discontinuous Galerkin finite element methods for second order hyperbolic problems, Computer Methods in Applied Mechanics and Engineering 107 (1-2) (1993) 117–129.

[137] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.

[138] O. Zienkiewicz, W. Wood, N. Hine, R. Taylor, A unified set of single step algorithms. Part 1: General formulation and applications, International Journal for Numerical Methods in Engineering 20 (8) (1984) 1529–1552.

[139] W. Wood, Practical time-stepping schemes, Oxford University Press, USA, 1990.

[140] K. E. Atkinson, An introduction to numerical analysis, John Wiley & Sons, 2008.

[141] T. Cebeci, Convective heat transfer, Springer, 2002.

[142] U. Langer, S. E. Moore, M. Neumller, Spacetime isogeometric analysis of parabolic evolution problems, Computer Methods in Applied Mechanics and Engineering 306 (2016) 342 – 363.

[143] M. J. Gander, 50 years of time parallel time integration, in: Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 69–113.

[144] J. Chan, J. A. Evans, Multi-patch discontinuous Galerkin isogeometric analysis for wave propagation: Explicit time-stepping and efficient mass matrix inversion, Computer Methods in Applied Mechanics and Engineering 333 (2018) 22 – 54.

[145] F. Auricchio, L. B. Da Veiga, T. J. Hughes, A. Reali, G. Sangalli, Isogeometric collocation for elastostatics and explicit dynamics, Computer Methods in Applied Mechanics and Engineering 249 (2012) 2–14.

[146] W. Yavari, S. Duan, K. Wei, Y. Ma, D. Fang, A quadratic B-spline based isogeometric analysis of transient wave propagation problems with implicit time integration method, Applied Mathematical Modelling 59 (2018) 115 – 131.

[147] A. Yavari, M. H. Abolbashari, Elastic wave propagation in non-uniform rational B–spline rods under mechanical impact loading using an isogeometrical approach, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science.

[148] D. Kelly, J. De SR Gago, O. Zienkiewicz, I. Babuska, A *posteriori* error analysis and adaptive processes in the finite element method: Part I–error analysis, International Journal for Numerical Methods in Engineering 19 (11) (1983) 1593–1619.

[149] J. De SR Gago, D. Kelly, O. Zienkiewicz, I. Babuska, A *posteriori* error analysis and adaptive processes in the finite element method: Part II–adaptive mesh refinement, International Journal for Numerical Methods in Engineering 19 (11) (1983) 1621–1656.

[150] U. Langer, S. Matculevich, S. Repin, A *posteriori* error estimates for space-time IGA approximations to parabolic initial boundary value problems, arXiv preprint arXiv:1612.08998.

[151] U. Langer, S. Matculevich, S. Repin, Guaranteed error control bounds for the stabilised space-time iga approximations to parabolic problems, arXiv preprint arXiv:1712.06017.

[152] U. Langer, S. Matculevich, S. Repin, Adaptive space-time isogeometric analysis for parabolic evolution problems, arXiv preprint arXiv:1807.05950.

[153] F. Verdugo, N. Parés, P. Díez, Error assessment in structural transient dynamics, Archives of Computational Methods in Engineering 21 (1) (2014) 59–90.

[154] I. Romero, L. M. Lacoma, A methodology for the formulation of error estimators for time integration in linear solid and structural dynamics, International Journal for Numerical Methods in Engineering 66 (4) (2006) 635–660.

[155] X. Li, L. F. Zeng, N. E. Wiberg, A simple local error estimator and an adaptive time-stepping procedure for direct integration method in dynamic analysis, Communications in Numerical Methods in Engineering 9 (4) (1993) 273–292.

[156] N. Biboulet, A. Gravouil, D. Dureisseix, A. Lubrecht, A. Combescure, An efficient linear elastic FEM solver using automatic local grid refinement and accuracy control, Finite Elements in Analysis and Design 68 (2013) 28–38.

[157] E. Biotteau, A. Gravouil, A. Lubrecht, A. Combescure, Multigrid solver with automatic mesh refinement for transient elastoplastic dynamic problems, International Journal for Numerical Methods in Engineering 84 (8) (2010) 947–971.

[158] E. Biotteau, A. Gravouil, A. A. Lubrecht, A. Combescure, Three dimensional automatic refinement method for transient small strain elastoplastic finite element computations, Computational Mechanics 49 (1) (2012) 123–136.

[159] F. Verdugo, N. Parés, P. Díez, Modal-based goal-oriented error assessment for timeline-dependent quantities in transient dynamics, International Journal for Numerical Methods in Engineering 95 (8) (2013) 685–720.

[160] F. Verdugo, N. Parés, P. Díez, Goal-oriented space-time adaptivity for transient dynamics using a modal description of the adjoint solution, Computational Mechanics 54 (2) (2014) 331–352.

[161] J. Muñoz-Matute, E. Alberdi, D. Pardo, V. M. Calo, Time-domain goal-oriented adaptivity using pseudo-dual error representations, Computer Methods in Applied Mechanics and Engineering 325 (2017) 395–415.

[162] T. Erhart, W. A. Wall, E. Ramm, Robust adaptive remeshing strategy for large deformation, transient impact simulations, International Journal for Numerical Methods in Engineering 65 (13) (2006) 2139–2166.

[163] V. Darrigrand, D. Pardo, I. Muga, Goal-oriented adaptivity using unconventional error representations for the 1D Helmholtz equation, Computers & Mathematics with Applications 69 (9) (2015) 964 – 979.

[164] J. Wloka, Partial differential equations. 1987, Cambridge University.

[165] M. Géradin, D. J. Rixen, Mechanical vibrations: theory and application to structural dynamics, John Wiley & Sons, 2014.

[166] A. Chemin, T. Elguedj, A. Gravouil, Isogeometric local h-refinement strategy based on multigrids, Finite Elements in Analysis and Design 100 (2015) 77–90.

[167] S. Ishikawa, Fixed points by a new iteration method, Proceedings of the American Mathematical Society 44 (1) (1974) 147–150.