Doctoral Dissertation

博士論文

STOCHASTIC ISOGEOMETRIC ANALYSIS (SIGA) METHOD FOR EVALUATION OF UNCERTAINTY IN SHAPE IN STRUCTURE

構造物の形状不確定性評価のための確率アイソジオメトリック解析方法

By

HONGGUAN ZHANG

Graduate School of Environment and Information Science

Yokohama National University

YNU

2019

SHIBUTANI.LAB

ABSTRACT

In reality, various uncertainties exist in structural systems, because various the physical characteristics are not deterministic in the actual engineering application. Such characteristics include material properties, boundary conditions, and structural shape, etc. These uncertain factors can be investigated by a numerical analysis method, that is commonly known as the uncertainty analysis method. Generally, the *stochastic finite element method* (SFEM) and *Monte Carlo simulation method* (MCS) are employed in order to analyse uncertain problems in the structural systems. However, in recent years, with the rapid development of structure design methods and improvement of risk-based rules, traditional deterministic numerical analysis methods have been unable to meet the need of analysing the uncertainty and randomness in practical engineering projects, especially for some complex geometries and geometric models that are sensitive to analytical precision, for instance corroded surfaces and smooth surface structures. Therefore, the development of uncertainty analysis methods is currently attracting an increasing amount of attention.

In this study, we mainly research the probability problem for uncertainty in shape that exists in the engineering field. We broke the classic "*FEA-based*" analysis point of view and proposed a "*physically-based*" analysis viewpoint based on the isogeometric analysis method, in order to address the numerical solving in regard to

uncertainty in shape. In addition, in this thesis, a new probability numerical analysis method is proposed that extend the classical deterministic *isogeometric analysis method* (IGA) into a probabilistic analytical framework in order to evaluate the uncertainty in shape, and aim to investigate a possible extension of isogeometric analysis method in the field of computational stochastic mechanics. This probability numerical analysis is called *stochastic isogeometric analysis* (SIGA). The SIGA method for uncertainty in shape is developed by employing the geometric characteristics of the *non-uniform rational basis spline* (NURBS) and the probability characteristics of polynomial chaos expansions (PCE). The NURBS is a very important and commonly used geometric drawing instrument in *Computer Aided Geometric Design* (CAGD), which has very good geometric representation and control ability.

In this study, we used NURBS as the basis functions, thus the proposed method fully inherits its excellent geometric characteristics. Based on these characteristics, the proposed method overcomes some of the shortcomings of the classic stochastic finite element method during dealing with uncertainty in shape. Despite classical SFEM has excellent analytical performance and a sound analytical system, there are still some inevitable limitations. Especially, in the uncertainty analysis in shape, because of what its use of a geometry approximated by a finite element mesh (FE-mesh), some of its innate disadvantages have been exposed. In many situations, this geometry approximated can cause errors in the analytical results during the performing uncertainty analysis on some complex and sensitive geometric structures. In order to overcome the shortcomings in the SFEM mentioned above, based on the natural characteristics of NURBS, we proposed the stochastic isogeometric analysis method to deal with the problem of uncertainty in shape. Firstly, throughout the probabilistic analysis, an exact geometric entity was used to represent uncertainty in shape, which effectively reduced errors in terms of geometry. From the algebraic aspect, the NURBS basis function used for discretization is a smoother, highly continuous basis function, thus that greatly improved the accuracy and reliability of the analysis. Furthermore, unlike typically FE-mesh in the classical SFEM, usually,

the NURBS uses the control points to manipulate geometric shape, and consequently, the problem of mesh repartitioning does not need to be considered, the costs and difficulty of the analysis were diminished, significantly. Secondly, based on excellent geometric properties of NURBS, the boundaries of the analytical region containing random field can be identified exactly, easily and flexibly.

Finally, we use the intrusive formulation approach to incorporate PCE into the IGA framework and gave the corresponding formalization method. this analysis procedure is implemented by using the C++ programming language. In addition, in order to verify the validity and applicability of the proposed method, three numerical examples are presented. The validity and accuracy of the results are assessed by comparing them to the results obtained by Monte Carlo simulation (MCS) based on the IGA algorithm.

ACKNOWLEDGEMENTS

Contents

1. INTRODUCTION

1.1. Background

With the rapid development of science and technology and the computer industry, numerical analysis technology has been paid more and more attention. Especially for numerical simulation technology has become an important analytical tool for product reliability evaluation. This technology can greatly shorten the cycle of product research and development and quickly and easily simulate some high-risk extreme experiments. For large-scale buildings and machinery that have been put into use, such as bridges, ships and so on, we can make real-time and effective risk assessment according to their actual use. For example, in the field of structural engineering, finite element method is often used in numerical simulation technology, which can quickly and effectively evaluate various physical and engineering problems encountered in real life, such as car crash problems are shown in Figure 1.1; and vibration presented in Figure 1.2 etc. These are very important for building a safe and secure production and living environment.

In the vast majority of cases, since the deterministic boundary conditions are easy to set up and the mathematical configuration is simple, engineering and technical personnel are accustomed to using deterministic parameters to approximate numerically simulate engineering and physical problems encountered, and that achieved very good results and has been widely used. However, in reality, various uncertainties exist in structural systems, because the various physical characteristics are not deterministic in the actual engineering application. Such physical characteristics include material properties, boundary conditions, and structural shape, etc. Furthermore, in recent years, with the rapid development of risk-based rules and structure design methods, traditional deterministic numerical analysis methods have been unable to meet the need of analyzing the uncertainty and randomness in practical engineering projects, especially for some high precision machinery. Therefore, it is currently attracting an increasing amount of attention to the development and research of uncertainty analysis methods. Usually, uncertainties exist in the actual situation have large random statistical properties. These uncertain factors can be investigated by a numerical analysis method, that is commonly known as the *uncertainty numerical analysis method* and also it is often referred to as *probabilistic numerical analysis* or *stochastic numerical analysis*. The probabilistic numerical analytical technology is widely used in reliability analysis which can evaluate the reliability of simple or complex systems as realistically as possible. It is similar to the deterministic numerical analysis method, but the uncertainty of the variables is considered in the analysis. For example, suppose we want to assess the extent to which a typhoon damages a building during its use and the resulting economic loss. First, we need to predict the maximum typhoon wind speed that it may encounter during the period of use, and then analyze the wind-induced dynamic response of the house under the typhoon wind speed. In this process, it can be found that many parameters are not a deterministic value. For example, the maximum typhoon wind speed is between V-max and V-min, and the wind speed field spatial correlation factor is between 7 and 21. Obviously, it would be unrealistic to consider these parameters as deterministic and use them to numerically simulate buildings. But if we set these parameters as the uncertain variables, we can quickly implement

the analytical process and get more realistic results. Of course, it depends on whether the corresponding deterministic algorithm has sufficient analytical accuracy and perfect analytical framework, and also on the relevant information we have.
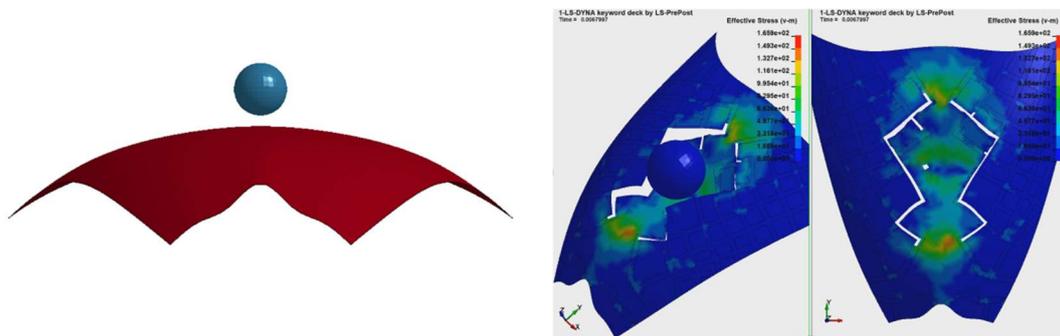
Figure 1.1 The crash problems for the ball hit the composite material car hood.
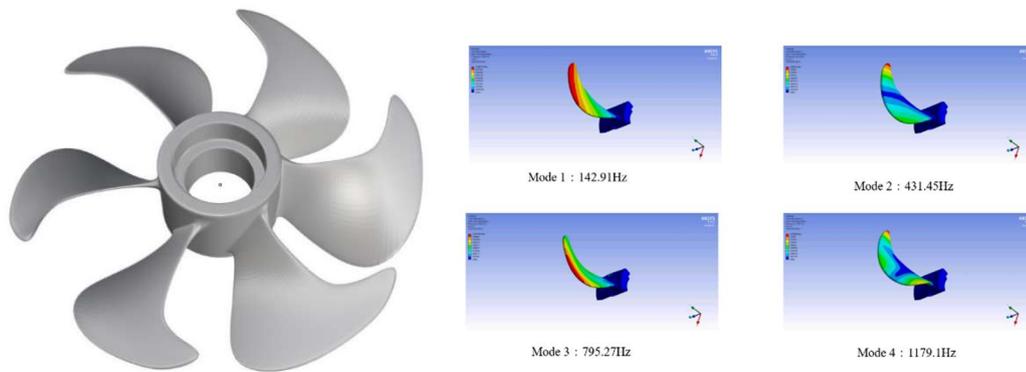
Figure 1.2 The vibration problem of the propeller blades.

In this study, we mainly discuss the problem of shape uncertainty in the field of structural engineering, and based on this, develop a new uncertainty numerical method suitable for uncertainty in shape. Although uncertain numerical analysis

technology has been applied to engineering construction system maturely, and many uncertainty analytical methods for different problems have been developed, there are few discussions on the application of uncertainty analytical technology to shape uncertainties and the development of related technologies. However, for the construction system, there are uncertainties in shape. For example, the structure in Figure 1.3 is part of the frame of a motorcycle, where the shape of the welded part of a structure is uncertain; These uncertainties in shape caused by such welding or processing will cause great potential safety hazards in the operation of machinery and equipment. Actually, there are many more examples like this one with regard to generating shape uncertainty in the process of product development and design such as shape uncertainty caused by corrosion, and shape errors that occur during processing, etc. Therefore, effective evaluation of these uncertainties in advance can detect potential safety hazards in time and avoid accidents.
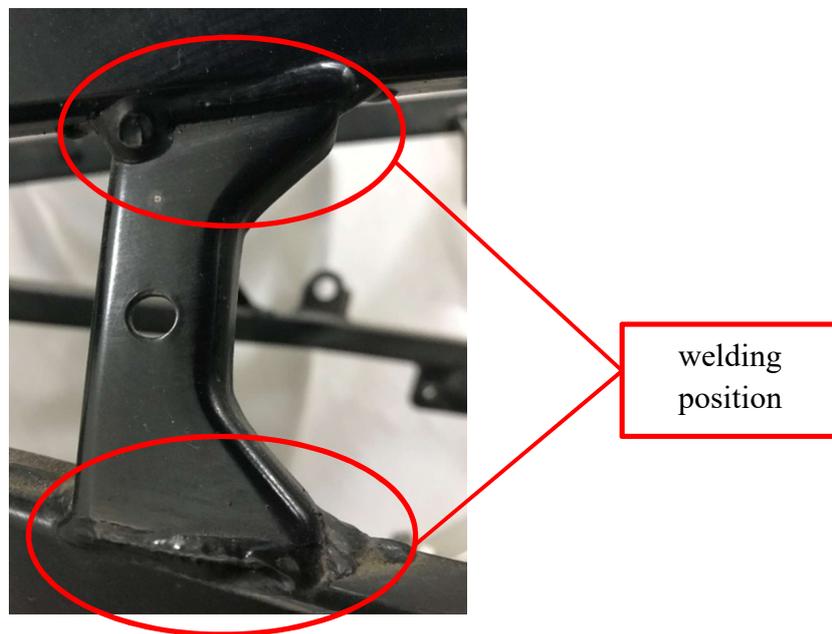
Figure 1.3 A motorcycle part; the uncertainty in shape consist in welding position, that is marked with closed red lines.

1.2. Uncertainty and Its Analysis

1.2.1. stochastic finite element method (SFEM)

At present, in the field of engineering technology, uncertain analysis of various physical phenomena is usually based on *stochastic finite element method* (SFEM), which is also often referred to as *probabilistic finite element method*. The SFEM is an extension of the classical finite element method that is from a deterministic numerical analysis framework to a stochastic analysis framework. The stochastic finite element center idea is a more elaborate description of the performance function, and this description is based on the deterministic complex structural finite element algorithm. Therefore, it can theoretically contain the characteristics of all deterministic finite element methods. For example, considering nonlinear effects and dynamic response calculations, in addition to solving the deterministic algorithm results, we can simultaneously solve the mean, variance and reliability indicators of the response. In addition, the stochastic finite element method is only one of many reliability algorithms. It is good at solving the checking point of implicit performance function in complex structure system. It incorporates sensitivity analysis technology and improves second-order moment method on the basis of deterministic FEM. Its basic solution can be expressed as a differential equation, $\partial g / \partial \varsigma$, where g is the performance function, and ς basic random variable. Note that every time the differential equation is solved, it is needed to be run for a structural finite element analysis process used as a sub-function in the probabilistic finite element method. Therefore, for an SFEM-based construction of uncertainty analysis process, there are the following several calculation modules.

- Modeling with deterministic finite element method.
- Choosing the appropriate performance function
- Determining the probability description of basic random variables and equivalent normalization or non-positive mode

- Performing a perturbation analysis of basic random variables by perturbation or difference method, etc.
- Checkpoint iterative algorithm
- Checkpoint iteration convergence criterion

As a reliability analysis algorithm, stochastic finite element method aims at mainly predicting the probability of an engineering event or physical phenomenon and one has been widely applied in science and engineering as an important uncertainty analysis method. Initially, Astill and Shinozuka[1] have presented the *Monte Carlo simulation method* (MCS) basis on the framework of the finite element method that is a combination of the FEM and MCS. This method performs a numerical solution for each sample collected using the finite element method. Therefore, the deterministic finite element analysis program is run the same number of times as the number of samples, and finally, it is observed whether or not a certain probability distribution is presented based on the obtained results. The latter is the most general and simplest approach for dealing with response variability in the structural system[2]–[8]. However, the MCS requires excessive computational power in comparison to other stochastic FEMs, and especially for handling complex models involving several stochastic variables. The perturbation method[9]–[17] overcomes this drawback, when the perturbation of response variability is in the first and second-order. The perturbation method was developed by applying the Taylor series expansion of the response vector into the physical system, and the results of the analysis are the distribution-free[18]. This method is limited within a minute perturbation range that is usually less than 20 or 30 percent of a variable's mean value. In recent years, another important branch of the SFEM has been presented by Ghanem and Spanos [19], namely, the spectral stochastic finite element method (SSFEM)[20]–[25]. In general, this method makes use of the Karhunen-Loève (K-L) expansion of the Gaussian random field in order to represent the uncertain parameters of a problem (such as material properties, nodal displacement, etc.). For the representation of nodal displacement in the SSFEM, an alternative approach has been provided and consists of using polynomial chaos expansions (PCE) [26]. In

recent years, some progress has been made in shape uncertainty analysis. For instance, Nakagiri et al. [27] published a paper on considering shape uncertainty, material uncertainty, and boundary uncertainty, and SFEM based on perturbation method is proposed to calculate the mean value and the variance of stress. However, it only can deal with the small deformation problem because perturbation method has limitations. Honda [28] proposed spectral stochastic boundary element method (SSBEM) based on PCE and the K-L expansion in order to analyse the problem of uncertainty in shape in the boundary. However, this paper only describes the case where the uncertainty of the boundary shape follows the positive distribution, because the K-L expansion method is based on normal distribution. Htun et al.[29] studied a problem for the ultimate tensile strength of the plates with a random field of corrosion. In the study, the corroded plate is represented by K-L expansion method, and the stochastic properties of their strength are estimated by PCE method. In addition to studying the random ultimate tensile strength problem of the plates with a random field of corrosion, the capability of the random field model to represent the randomness of corroded surfaces is also investigated. And the random characteristics of the real corroded surface were investigated based on the measurement data of the real corroded plates. Chen et al. [30] presented a new method of structural analysis for the solution of response uncertainty problems in the cases involving uncertainty in shape. The proposed method includes a mathematical formulation, which is a natural extension of the deterministic finite element concept to the space of random functions by the Hermite polynomial chaos expansion, in order to represent the uncertainty of shapes and the response surface. And the problem of shape in uncertainty following the no-normal distribution has also been studied. Developments have been described in some key articles [31]–[33].

1.2.2. Stochastic isogeometric analysis (SIGA)

Recently, isogeometric analysis (IGA) method was proposed by Hughes et al. [33] as an important alternative technology in computational mechanics. Its core idea was

the use of the smooth geometric basis in CAD as the basis functions for numerical analysis [3]. This method successfully integrated computer-aided design (CAD) and computer-aided engineering (CAE) into a unified process, i.e. parameterizing the CAD objects to obtain an effective computation domain and generate a mesh, which is applied to the FEM analysis framework [36]. Xu et al. have done a lot of important work to improve the quality of the parameterization of the computational domain of CAD objects [37], [38]. A detailed schematic diagram of the IGA method is shown in Figure 1.4.
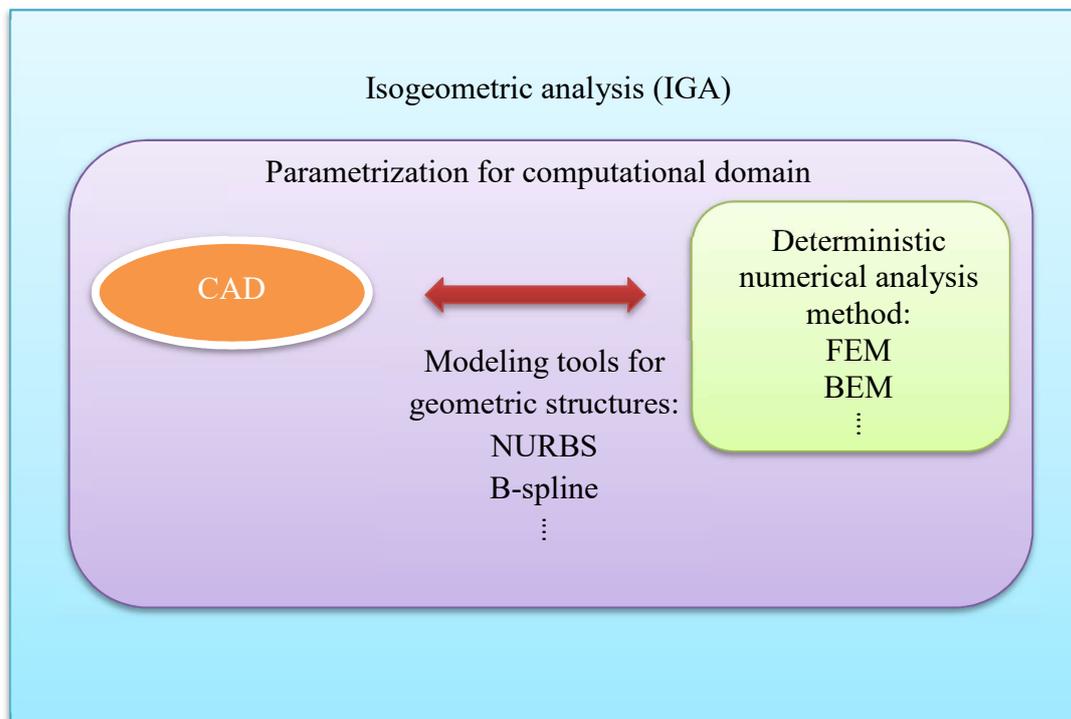
Figure 1.4 The schematic diagram for IGA method.

At present, IGA extended rapidly to other fields of numerical analysis, including uncertainty analysis. Researchers in the numerical analysis have noticed some advantages of IGA in numerical analysis, so they try to combine some stochastic analytical methods with the isogeometric analytical framework. Rossana et al. [39] presented an innovative numerical method for computing the stress concentration

factors in an isotropic plate with discontinuities by using IGA and SFEM. This work confirms the potentials and accuracy of the proposed methodologies to capture the stress concentrations in fracture mechanics, also for coarse mesh discretizations. Hien and Noh [40] developed a perturbation technique in conjugation with IGA for the stochastic eigenvalue problem of free vibration of functionally graded material (FGM) plates with two random parameters for the elastic modulus and mass density, respectively. The governing equation of stochastic isogeometric analysis for free vibration of functionally graded plates is derived in conjunction with perturbation expansions to predict the first and second moments of eigenvalues. Hien and Lam [41] use IGA and MCS to address the bending of a plate under random load. The governing equations are derived to use higher-order plate theory. The fluctuations in the random load are modeled as a two-dimensional random field. The random process of random loading is simulated by using the spectral representation method. Li et al. [42] have proposed spectral stochastic isogeometric analysis (SSIGA) for stochastic linear elasticity. Thereinto, NURBS- and T-splines-based Karhunen–Loève expansion is proposed for random field decomposition. Furthermore, Strength and serviceability limit state designs can be incorporated within the proposed SSIGA analysis framework. By utilizing the nonparametric statistical analysis, both probability density functions (PDFs) and cumulative distribution functions (CDFs) of concerned structural displacements and stresses can be effectively established. In addition, Li et al. [43] have also investigated the nondeterministic structural responses of functionally graded material (FGM) plates under static loads with uncertain material property. The considered spatially dependent uncertainties are modeled as random fields with Gaussian distribution. A novel spectral stochastic isogeometric analysis (SSIGA) framework is proposed for such uncertainty quantification through the first-order shear deformation theory. Within the SSIGA framework, the non-uniform rational B-spline (NURBS) is adopted for both the geometry modeling of the random fields of the uncertain material properties and random field discretization through the Karhunen-Loève (K-L) expansion. Such new feature provides an effective and practically applicable random field modeling technique, especially for uncertain parameters over complex physical domains.

1.3. Motivation and Challenges

Actually, with the rapid development of industry and the improvement of structural design technology, the geometric shape of structures has become increasingly complex. Simultaneously, it has brought more new challenges to the field of uncertainty analysis techniques. From the above discussion, we can see that at present, among all the proposed SIGA methods, there is no one to discuss the problem for uncertainty in shape. Therefore, in the field of stochastic isogeometric analysis, the research on shape uncertainty is a blank space. Despite the remarkable advantages and benefits of what stochastic FEM has brought to modern engineering applications, but with regard to research uncertainty in shape, there are still some inevitable limitations. It is because of what stochastic FEA is an extension based on the analytical framework of the classical FEA, i.e. integrating probability analysis into FEA. Therefore, it inherits all the characteristics of FEA. Note that these inheritances include advantages and disadvantages of FEA [44]. And, some of the inherited disadvantages have severely astricted uncertainty representation of the structure. Here, from the perspective of uncertainty analysis in shape, we discuss the effects of these shortcomings on the uncertainty represent of geometry.

Throughout the stochastic FEM analysis process, a geometry approximated by finite element mesh is used to represent the uncertainty in shape. In many situations, this approximation can cause errors in the analytical results, especially for engineering applications that are extremely sensitive to geometric imperfections. For instance, the random field acting on a physical object with conic geometry is already difficult enough to be accurately modeled within stochastic FEA, not to mention the stochastic system with complicated, yet realistic, geometries which are often designed from modern industrial applications.

In addition, within stochastic FEA, since uncertainty in shape is represented by introducing the uncertain parameter into the nodes, the finite element mesh needs to be divided repeatedly. The example of a 1/4 model regarding plate with a circular hole at the center shown in Figure 1.5 from Reference [32], where the geometry

model is described by the approximated finite element mesh is divided by 8 elements and 15 nodes. We assume that the uncertainty takes place in the central hole, so the uncertain parameters are introduced into the nodes 1, 2, 3, 4 and 5 denoted by blue dote in Figure 1.5. When uncertainty in shape occurs, the nodes being moved are denoted by red dots and changes of FE mesh are denoted by dotted lines.
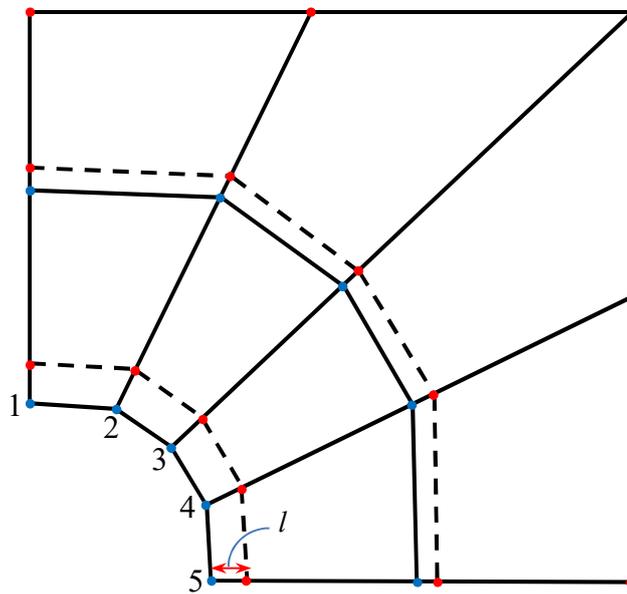
Figure 1.5 A 1/4 model regarding plate with a circular hole at the center which is approximated from FE mesh.

From Figure 1.5, we can see that in order to represent the uncertainty in shape and calculate response surfaces, the FE mesh must be divided repeatedly and this remeshing is not a single node location change but will affect elements and nodes in the entire analytical area. Especially, in actual application, the FE mesh structure in the analytical position is usually complex and fine, which adds more difficulty and workload to analysis. Note that the geometrical shape of the hole in Figure 1.5 cannot be accurately represented by the FE mesh.

Another major disadvantage of SFEM in current uncertainty analysis in shape is that the classical SFEM cannot define the analytical boundaries flexibly, exactly and easily. For example, if we assume that we want to change the shape of the circular hole in Figure 1.5 into other geometric shapes. Commonly, in the FEA framework through geometric visualization tools, we can visualize the geometric model first and then intuitively artificially move the nodes at the position of the hole to change its shape into the desired shapes, and then repartition the FE mesh. Unfortunately, in the actual analysis and formulation of probability system, due to the fact that we can't artificially physically manipulate the nodes in the analytical process, instead, use algebraic manipulation to deal with this process. For the SFEA, the degree of change of each node at the position of the circular hole need to be calculated in order to represent uncertainty in shape and FE mesh needs to be repartitioned, tautologically [32]. At present, as the example in Figure 1.5, within the SFEM analytical framework, the uncertainty of the hole is represented only by changing the radius of the hole, i.e. the uncertainty in shape is only considered as a change in the size of the radius at the hole. Because, the coefficients of the degree of change at each node are very difficult to be calculated, that leads to difficulties in the representation of uncertainty in shape and re-division of the mesh. In Reference [32], Chen et al. gave an algorithm with respect to determining the coefficients of the degree of change at each node, but that cannot perform well for geometric structures with curves and surfaces due to the drawbacks inherited from the FEM mentioned above. For example, As shown in Figure 1.5, in order to represent the uncertainty at the position of the hole, the all nodes at the hole are moved in same the distance l. For preventing bad mesh, the corresponding the surrounding nodes are also moved based on the coefficients of the degree of change at each node. Note that these coefficients of the degree of change also there is an impact on the analytical results to a certain degree. In Reference [32], Chen et al. gave a discussion with respect to the influence of the determining of the coefficients on the analysis result. It is expounded that if the bad coefficients are determined, a bad mesh will be generated, resulting in an inaccuracy result that may be obtained.

1.4. Contributions

With the development of the times, structural safety problems are being concerned more and more. Therefore, it is expected that the research and development in regard to the more efficient and more convenient stochastic analysis method are applied to the fields of structural reliability analysis and structural risk assessment etc. In addition, in order to overcome the shortcomings in the SFEM mentioned above. Therefore, in this study, we proposed the stochastic isogeometric analysis method to deal with the problem of uncertainty in shape. moreover, we explored one possible extension of IGA in the field of computational stochastic mechanics, that is incorporating of classical response surface methodology into the IGA analytical framework to evaluate the uncertainty problem in shape in stochastic analysis. The main contributions of this study can be summarized as follows:

➢ The stochastic isogeometric analysis method is freshly proposed for uncertainty in shape. From the above discussion, we can see that at present, among all the proposed SIGA methods, there is no one to discuss the problem for uncertainty in shape. Therefore, in the field of stochastic isogeometric analysis, the research on shape uncertainty is a blank space. This study makes up for the gap in the field of shape uncertainty analysis of SIGA, and further improves the processing method of shape uncertainty in the field of engineering computing.

➢ The PCE is introduced in IGA analysis framework.
  In this study, we proposed to introduce PCE into the IGA analysis framework to address the uncertainty problems in shape and given a formalization method. According to the authors' best knowledge, the presented work herein is the first reported work to integrate the PCE computational scheme into the isogeometric analysis.

➢ Comparing with SFEM.

This approach breaks some of the limitations of SFEM in uncertain analysis and effectively improves the analysis environment.

- In the SFEM, the uncertainty in shape is represented by setting the coefficients of the degree of change at each node, and these coefficients have an impact on the analytical results. In this study, we can achieve numerical solving without using these coefficients. Therefore, our work effectively improves computational efficiency and reduces the complexity of the analysis.

- From the perspective of probabilistic analysis, the random field is directly applied to an exact geometric entity through the control point, rather than an approximate mesh structure. Therefore, it is sensitive to geometric shape changes, which effectively reduced geometrical errors of numerical solving. This is beneficial to uncertainty analysis in shape, especially for engineering applications that are extremely sensitive to geometric imperfections.

- Moreover, by inheriting the advantage of IGA, the proposed SIGA approach with several unique superiorities. Firstly, this approach is applicable to situations where the physical domains of the random fields are possessing complex geometries. In particular, the proposed SIGA can exactly represent commonly encountered shapes such as ellipse, circles, spheres, and cylinders. That is, exact geometries of the physical domains can be promised even at a relatively coarse level of discretization and consequently, the corresponding geometrical errors can be significantly diminished. Secondly, this approach does not require meshing, and that use the smoother, highly continuous basis function, thus the calculation time is reduced and the analysis accuracy is improved. Finally, the CAD model can be directly used to analyze uncertainty in shape, which makes probability analysis and CAD integration. In this study, we use the geometry data output directly from the CAD tool to implement the interaction between the probabilistic system and CAD.

➢ Comparing with MCS based on IGA

As a typical representative of the non-intrusive method, Monte Carlo Simulation Method (MCS) is the most widely used technique which is used to evaluate response uncertainty. Its main advantages are easy implementation, but require a lot of samples and repetitive calculations in the analytical process. As a result, the computational cost is increased, especially for complex and large structures the analytical workload of the MCS is enormous. In this study, the analysis framework was built by using the intrusive method and is therefore very scalable, and the construction of stochastic response surface does not require multiple simulations, thus it can deliver very fast results at a minimum computational cost.

1.5. Organization of This Thesis

In the light of achieving a self-contained thesis with enhanced readability, this paper has been meticulously structured as follows.

First of all, in Chapter 2, the theory of IGA is briefly presented. Some important notions which are a key to understanding IGA are described and illustrated in detail, including the physical space, parameter space, and knot vector etc. In addition, some main paraphernalia employed in isogeometric analysis be given, for example, NURBS, B-spline, and basis function etc. Finally, we also focused on some of the properties of B-spline, which played a crucial role in the development of later algorithms.

The purpose of Chapter 3 is to review the mathematical basis of probability theory. In this chapter, we mainly describe the relevant knowledge of stochastic analysis, including the concept of stochastic analysis, the types of stochastic analysis, and the knowledge of the current mainstream polynomial expansion method. In addition, We can learn from this chapter that stochastic analysis is mainly divided into two categories: the intrusive method and non-intrusive method.

In Chapter 4, we describe in detail the formalization process for stochastic isogeometric analysis method for uncertainty in shape. The formalization constructed is an intrusive formulation procedure. The deterministic isogeometric analysis framework was rewritten as an uncertainty form based on PCE, and the orthogonal properties of PCE were fully utilized in order to solve the stiffness matrix. In addition, we propose a new analytical perspective to deal with the shape uncertainty problem in engineering analysis, namely, in this study, a novel method is proposed in the aspect of reliability analysis for uncertainty in shape, and it is carried out from a physically-based point of view. Because the uncertainty in the shape of structure model was represented by directly introducing stochastic parameters into the control points in the physical space, the new analytical viewpoint overcomes many shortcomings of the traditional SFEM method.

In order to illustrate the effectiveness and efficiency of the proposed method, four distinctive numerical examples are thoroughly explored in Chapter 5. For the first numerical example which is a cantilever beam model, the freshly proposed generalized stochastic isogeometric analysis based polynomial chaos expansion is rigorously verified against the well-established theoretical results. Subsequently, the proposed stochastic isogeometric analysis framework is further implemented for the stochastic static analysis of localized corrosion in the second example. In order to apply to more complex analytical models, we present examples of two multi-patch NURBS geometry models i.e. infinite plate with circular hole and butt joint, these ones fully demonstrate the ability of SIGA to handle multiple patch geometries.

Finally, conclusions and further work are drawn in Chapter 6.

1.6. Conclusion

In this part, we present the research and development background of uncertainty analysis. It also briefly introduced some uncertain analytical methods that are widely used in the field of current computational stochastic mechanics, for example, Monte Carlo simulation method, stochastic finite element method, and spectral stochastic analysis. In addition, we point out some classical SFEM methods in dealing with the problem of shape uncertainty. Especially, because of what its use of a geometry approximated by a finite element mesh (FE-mesh), some of its innate disadvantages have been exposed. In many situations, this geometry approximated can cause errors in the analytical results during the performing uncertainty analysis on some complex and sensitive geometric structures. in order to overcome these shortcomings, we propose stochastic isogeometric analysis method, that extend the classical deterministic isogeometric analysis (IGA) into a probabilistic analytical framework in order to evaluate the uncertainty in shape, and aim to investigate a possible extension of IGA in the field of computational stochastic mechanics. Stochastic isogeometric analysis (SIGA) method for uncertainty in shape is developed by employing the geometric characteristics of the non-uniform rational basis spline (NURBS) and the probability characteristics of polynomial chaos expansions (PCE). The proposed method can accurately and freely evaluate problems of uncertainty in shape caused by the deformation of the structural model. Finally, we list in detail some of the main contributions of this study.

References

[1]   C. J. ASTILL, S. B. IMOSSEIR, and M. SHINOZUKA, 'Impact Loading on Structures with Random Properties', J. Struct. Mech., vol. 1, no. 1, pp. 63–77, Jan. 1972.

[2]   C. J. ASTILL, S. B. IMOSSEIR, and M. SHINOZUKA, 'Impact Loading on Structures with Random Properties', J. Struct. Mech., vol. 1, no. 1, pp. 63–77, Jan. 1972.

[3]   J. E. Hurtado and A. H. Barbat, 'Monte Carlo techniques in computational stochastic mechanics', Arch. Comput. Methods Eng., vol. 5, no. 1, p. 3, Mar. 1998.

[4] J. Argyris, M. Papadrakakis, and G. Stefanou, 'Stochastic finite element analysis of shells', Comput. Methods Appl. Mech. Eng., vol. 191, no. 41, pp. 4781–4804, Sep. 2002.

[5]   C. A. Schenk and G. I. Schuëller, 'Buckling analysis of cylindrical shells with cutouts including random boundary and geometric imperfections', Comput. Methods Appl. Mech. Eng., vol. 196, no. 35, pp. 3424–3434, Jul. 2007.

[6]   F. Schoefs, M. Chevreuil, O. Pasqualini, and M. Cazuguel, 'Partial safety factor calibration from stochastic finite element computation of welded joint with random geometries', Reliab. Eng. Syst. Saf., vol. 155, pp. 44–54, Nov. 2016.

[7]   B. Van den Nieuwenhof and J.-P. Coyette, 'Modal approaches for the stochastic finite element analysis of structures with material and geometric uncertainties', Comput. Methods Appl. Mech. Eng., vol. 192, no. 33, pp. 3705–3729, Aug. 2003.

[8]   G. Stefanou and M. Papadrakakis, 'Stochastic finite element analysis of shells with combined random material and geometric properties', Comput. Methods Appl. Mech. Eng., vol. 193, no. 1, pp. 139–160, Jan. 2004.

[9]   W. K. Liu, T. Belytschko, and A. Mani, 'Random field finite elements', Int. J. Numer. Methods Eng., vol. 23, no. 10, pp. 1831–1845, Oct. 1986.

[10]   W. K. Liu, T. Belytschko, and A. Mani, 'Probabilistic finite elements for nonlinear structural dynamics', Comput. Methods Appl. Mech. Eng., vol. 56, no. 1, pp. 61–81, May 1986.

[11]   O. Cavdar, A. Bayraktar, A. Çavdar, and S. Adanur, 'Perturbation Based Stochastic Finite Element Analysis of the Structural Systems with Composite Sections under Earthquake Forces', Steel Compos. Struct., vol. 8, Apr. 2008.

[13]   M. Kamiński and P. Świta, 'Structural stability and reliability of the underground steel tanks with the Stochastic Finite Element Method', Arch. Civ. Mech. Eng., vol. 15, no. 2, pp. 593–602, Feb. 2015.

[14]   F. Wu, Q. Gao, X.-M. Xu, and W.-X. Zhong, 'A Modified Computational Scheme for the Stochastic Perturbation Finite Element Method', Lat. Am. J. Solids Struct., vol. 12, no. 13, pp. 2480–2505, 2015.

[15]   Y. Xu, Y. Qian, and G. Song, 'Stochastic finite element method for free vibration characteristics of random FGM beams', Appl. Math. Model., vol. 40, no. 23, pp. 10238–10253, Dec. 2016.

[16]   G. A. da Silva and E. L. Cardoso, 'Stress-based topology optimization of continuum structures under uncertainties', Comput. Methods Appl. Mech. Eng., vol. 313, pp. 647–672, Jan. 2017.

[17]   'The stochastic finite element method (basic perturbation technique and computer implementation), M. Kleiber and T. D. Hien, Wiley, Chichester, U.K., 1992. ISBN–0-471-93626–X No. of pages: xiv + 322. Price: £49.95', Appl. Stoch. Models Data Anal., vol. 10, no. 4, pp. 297–297, 1994.

[18]   B. Sudret, 'Stochastic Finite Element Methods and Reliability A State-of-the-Art Report', p. 190.

[19]   R. G. Ghanem and P. D. Spanos, Stochastic Finite Elements: A Spectral Approach. Courier Corporation, 2003.

[20]   R. G. Ghanem and R. M. Kruger, 'Numerical solution of spectral stochastic finite element systems', Comput. Methods Appl. Mech. Eng., vol. 129, no. 3, pp. 289–303, Jan. 1996.

[21]   A. Nouy, 'Recent Developments in Spectral Stochastic Methods for the Numerical Solution of Stochastic Partial Differential Equations', Arch. Comput. Methods Eng., vol. 16, no. 3, pp. 251–285, Sep. 2009.

[22]   R. Ghanem, 'Ingredients for a general purpose stochastic finite elements implementation', Comput. Methods Appl. Mech. Eng., vol. 168, no. 1, pp. 19–34, Jan. 1999.

[23]   S.-K. Choi, R. V. Grandhi, and R. A. Canfield, Reliability-based structural design. London: Springer, 2007.

[24]   K. Sepahvand, 'Stochastic collocation-based finite element of structural nonlinear dynamics with application in composite structures', MATEC Web Conf., vol. 83, p. 01009, 2016.

[25]   D. M. Do, W. Gao, and C. Song, 'Stochastic finite element analysis of structures in the presence of multiple imprecise random field parameters', Comput. Methods Appl. Mech. Eng., vol. 300, pp. 657–688, Mar. 2016.

[26]   G. Blatman and B. Sudret, 'Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach', Comptes Rendus Mécanique, vol. 336, no. 6, pp. 518–523, Jun. 2008.

[27]   Nakagiri S, Hisada T, and Itotani Y, 'Analysis of structure with uncertainty in shape by stochastic finite element method', Trans. Jpn. Soc. Mech. Eng. Ser. A, pp. 339–348, 1982.

[28]   R. Honda, 'SPECTRAL STOCHASTIC BOUNDARY ELEMENT METHOD FOR ELASTIC PROBLEMS WITH GEOMETRICAL

UNCERTAINTY', Doboku Gakkai Ronbunshu, vol. 2004, no. 759, pp. 111–120, Apr. 2004.

[29]   M. M. Htun, Y. Kawamura, and M. Ajiki, 'A Study on Random Field Model for Representation of Geometry of Corroded Plates and Estimation of Stochastic Properties of Their Strength', J. Jpn. Soc. Nav. Archit. Ocean Eng., vol. 18, pp. 91–99, 2013.

[30]   X. Chen, Y. Kawamura, and T. Okada, 'A Study on the Method of Structural Analysis with Uncertainty in Shape by Stochastic Finite Element Method', J. Jpn. Soc. Nav. Archit. Ocean Eng., vol. 22, pp. 187–195, Jan. 2015.

[31]   X. Chen, Y. Kawamura, and T. Okada, 'Development of Structural Analysis Method with Uncertainty in Shape to Follow Non-normal Distribution by Stochastic Finite Element Method', Trans. Jpn. Soc. Comput. Eng. Sci., pp. 20160019–20160019, Aug. 2016.

[32]   Chen X, Kawamura Y, Okada T., 'A Study on the Method of Structural Analysis with Uncertainty in Shape by Stochastic Finite Element Method (In Japanese)', 13th Int. Symp. Pract. Des. Ship Float. Struct. 2016.

[33]   Chen X, Kawamura Y, Okada T., 'Development of Stochastic Finite Element Method for Problems with Uncertainty in Shape Following Non-normal Distribution', Conf. JASNAOE, pp. 437–442, 2017.

[34]   T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 39–41, pp. 4135–4195, Oct. 2005.

[35]   J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.

[36] Xu G, Mourrain B, Duvigneau R, Galligo A. Parameterization of computational domain in isogeometric analysis: Methods and comparison. Computer Methods in Applied Mechanics and Engineering, 2011, 200(23-24): 2021-2031

[37] Xu G, Li M, Mourrain B, Rabczuk T, Xu JL, Stephane P.A. Bordas. Constructing IGA-suitable planar parameterization from complex CAD boundary by domain partition and global/local optimization. Computer Methods in Applied Mechanics and Engineering, 2018, 328, 175-200.

[38] Xu G, Kwok Tsz-Ho, Wang Charlie C.L. Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization. Computer-Aided Design, 2017, 91, 1-13.

[39] Dimitri R, Fantuzzi N, Tornabene F, Zavarise G. Innovative numerical methods based on SFEM and IGA for computing stress concentrations in isotropic plates with discontinuities, International Journal of Mechanical Sciences 2016; 118 (2016) 166–187.

[40] Hien TD, Noh HC. Stochastic isogeometric analysis of free vibration of functionally graded plates considering material randomness. Comput. Methods Appl. Mech. Engrg 2017; 318 (2017): 845–863.

[41] T. D. Hien TD, Lam NN. Investigation into the effect of random load on the variability of response of plate by using Monte Carlo simulation. International Journal of Civil Engineering and Technology (IJCIET) 2016; 7(5):169–176.

[42]   K. Li, W. Gao, D. Wu, C. Song, and T. Chen, "Spectral stochastic isogeometric analysis of linear elasticity," Comput. Methods Appl. Mech. Eng., vol. 332, pp. 157–190, Apr. 2018.

[43]   K. Li, D. Wu, and W. Gao, "Spectral stochastic isogeometric analysis for the static response of FGM plate with material uncertainty," Thin-Walled Struct., vol. 132, pp. 504–521, Nov. 2018.

[44] G. Stefanou, The stochastic finite element method: past, present and future, Comput. Methods Appl. Mech. Engrg. 2009;198 (9):1031–1051.

[45]   K. Li, W. Gao, D. Wu, C. Song, and T. Chen, "Spectral stochastic isogeometric analysis of linear elasticity," *Comput. Methods Appl. Mech. Eng.*, vol. 332, pp. 157–190, Apr. 2018.

2. ISOGEOMETRIC ANALYSIS (IGA) AND PRE-ANALYSIS TOOL NURBS

2.1. Overview

In the practically development and manufacturing processes of products and structures, the geometrical shape design and physical analysis methods for the model are completely different engineering fields. *Computer Aided Geometric Design* (CAGD) as a widely used geometric design technique, that usually attaches more importance to the construction and description of the geometric model of the product other than the physical characteristics of the product in the model analysis process. In addition, classical *CAE (Computer Aided Engineering)* tools focus only on the analysis and simulation of the physical characteristics of the product, for instance, structural strength, safety etc. and cannot be carried out the optimization of geometry

design for the product. The great divergence between geometric modeling and structural analysis is that the corresponding design and analysis systems of objects differ in their concerns at the implementation level [1]. From the conceptual perspective, this difference is manifested in the functionality of the model, which is based on purely geometric operations, while the analytical model is an abstract representation of physical phenomena. From the technical perspective, this difference is manifested in the description and calculation method of the model, that is, the geometric design model contains complex and fine geometric information, and its calculation method is based on the computational geometry method. The geometric model in the analysis is a simplified geometric representation, and its calculation method is mainly based on finite element methods.

In order to unify CAD and CAE, and achieve seamless integration between the two, T.J.R. Hughes et al. proposed the Isogeometric analysis (IGA) method [2]. This method mainly has the following characteristics.

- The IGA method offers the possibility of integrating classical CAE method and geometry design tool CAD for numerical analysis by using the same basis functions. That is, the spline basis functions used to structure geometrical models in the CAD are also used as shape functions in the analysis process, for example, NURBS, T-splines, etc.

- Analyzing object is the exact geometrical structure. In the traditional FEM method, the unknown solution field during the analysis is approximately described by dividing the mesh. In the IGA, the geometry of the analytical object can be represented exactly, especially for the description of common engineering geometries, such as circles, ellipses, etc.

- In addition, the IGA provides some very perfect refinement methods such as order elevation, knot insertion, and *k*-refinement etc. These refinement types greatly improve the accuracy of the analysis results, especially for some complex and highly sophisticated components. these refinement methods are far superior to the traditional FEM, which maintains high analytical accuracy

while greatly reducing the complexity in the subdivision process. For instance, the order elevation of IGA is similar to *p-* refinement in FEM. In finite element analysis, adding nodes within an element means increasing the polynomial order of the element. However, the order elevation increased in the FEM does not improve the continuity of the shape function. In the IGA, along with the increase in the number of order elevation, the continuity of the basis function is also increased accordingly.

- It is possible to change the shape without repartitioning the mesh. The IGA avoids the complex process of meshing in FEM by dividing the parameter domain of the analytical model and the geometric mapping of the parameter domain to the physical domain. That greatly reduces the parsing time.

Besides these, there are some good characteristics unique to Isogeometric analysis. For example, since it is also possible to pass the analysis result data back to the CAD, it can be applied to the optimized design. Furthermore, the basis functions typically used in IGA contain high smoothness and continuity, thus, from the perspective of algebra, the analytical results of IGA have higher precision.

In Table 2.1, the comparison of the FEM method and the IGA method is shown, from which we can see that the IGA analysis method is better than FEM in many aspects.

According to the above, we can see that the IGA analysis method overcomes the shortcomings of the traditional numerical analysis method FEM, but since IGA is a recently developed numerical analysis method, it still has many shortcomings in practical applications, so it cannot be widely applied to various engineering fields like the FEM. For example, parameterization problem of the computational domain [3]; Finding a numerical integration method that suitable for IGA method etc.

Table 2.1 Comparison of the IGA and the FEM.

	FEM	IGA
Shape representation	Approximate mesh	Surface, solid
Mesh	polygon mesh	Control mesh, physical mesh
Vector for the DOFs[1]	Nodal points	Control points
Shape function	Polynomial basis function	NURBS basis function
Control Variables	Nodal variables	Control variables
Continuity	C^0-continuity	High continuity
Shape precision	Approximate geometry	Exact geometry
Refinement space	*hp*- refinement space	*hpk*- refinement space
Basis	Basis not necessarily positive	Pointwise positive basis
Convex hull property	NO	YES
In the presence of discontinuous data	Oscillatory	Variation diminishing
Mesh generation	Mesh generation required	No need to generate a grid.

1) DOFs: The Degree of freedom.

2.2 The concepts of space and knot in the IGA

First of all, some basic concepts need to be introduced in order to understand Isogeometric Analysis method. There are four concepts of space defined in the IGA, which are *physical space*, *parameter space*, *parent space*, *index space*, and these spaces are related to each other. In these spaces, various elements and meshes are built, for ones, *knots* play a key role. in this section, from the notion of the knot, the concept of these spaces will be briefly introduced.

2.2.1 Knot vector

The *knot* is the basic elements of building parameter spaces. The knots are the points, curves, and surface in the one, two, and three dimensional topologies, respectively. By arranging knots in ascending order, the *knot vector* can be constructed. As an example, a knot vector is given as

$$\Xi = \left\{ \xi_0, \xi_1 \cdots, \xi_m \right\}, \tag{2.1}$$

where

$$\xi_i \leq \xi_{i+1},$$

ξ_i is i^{th}-knot in the knot vector; The half-open interval, $[\xi_i, \xi_{i+1})$, is called the ith-*knot span* where i denotes the knots index. From Equation (2.1) we can see the knot spans are bounded by knots; The length of the knot vector $m=n+p+1$, where n is the number of control points (or basis functions), and the p is the polynomial degree.

The knot vector can be *uniform* or *non-uniform*[4]. If the intervals of all the knots are equal, the knot vector is considered to be uniform, otherwise, it can be considered to be non-uniform. This non-uniform meaning that the knot value can also take on the same value for more than one knot, except taking a different length of knot span. non-uniform of knot vector has the peculiar implications for building the basis function. By this property, an *open knot vector* can be defined, that is, first and last

knot values in the knot vector are repeated $p+1$ (order of basis function + 1) times, defined as follows

$$\Xi = \left\{ \underbrace{\xi_0, \xi_1, \cdots, \xi_{p+1}}_{p+1}, \cdots, \xi_i, \cdots, \underbrace{\xi_n, \xi_{n+1}, \cdots, \xi_{p+n+1}}_{p+1} \right\} \tag{2.2}$$

where

$$\xi_0 = \xi_1 = \cdots = \xi_{p+1} \tag{2.3}$$

$$\xi_n = \xi_{n+1} = \cdots = \xi_{p+n+1} \tag{2.4}$$

The open vector knot has important implications for IGA. It makes the basis function have the property of interpolation. We will discuss this feature in more detail in a later section.

2.2.2 Parameter space

The parameter space is a special space, this space does not exist in the FEM method. The parameter space is constructed from non-zero intervals between knot values and partitioned into elements by the knot vector in the different parameter direction. These knot values are considered as the coordinates of the parameter space. Based on the parameter space constructed, corresponding basis functions can be acquired. In Figure 2.1, an example of two-dimensional parameter space is shown, and corresponding basis functions are given. The degree of the basis functions in each direction is second-degree. This parameter space is created by the knot vectors $\Xi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$ and $\Pi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$ in different directions, respectively. Note that, both of these knot vectors are open. Through Equations (2.2), (2.3) and (2.4) we can see that the knot values at both ends of the knot vector are repeated 3 times, i.e. the knot value for knot repeated at both ends of the knot vector equal first and last knot value of knot vector.
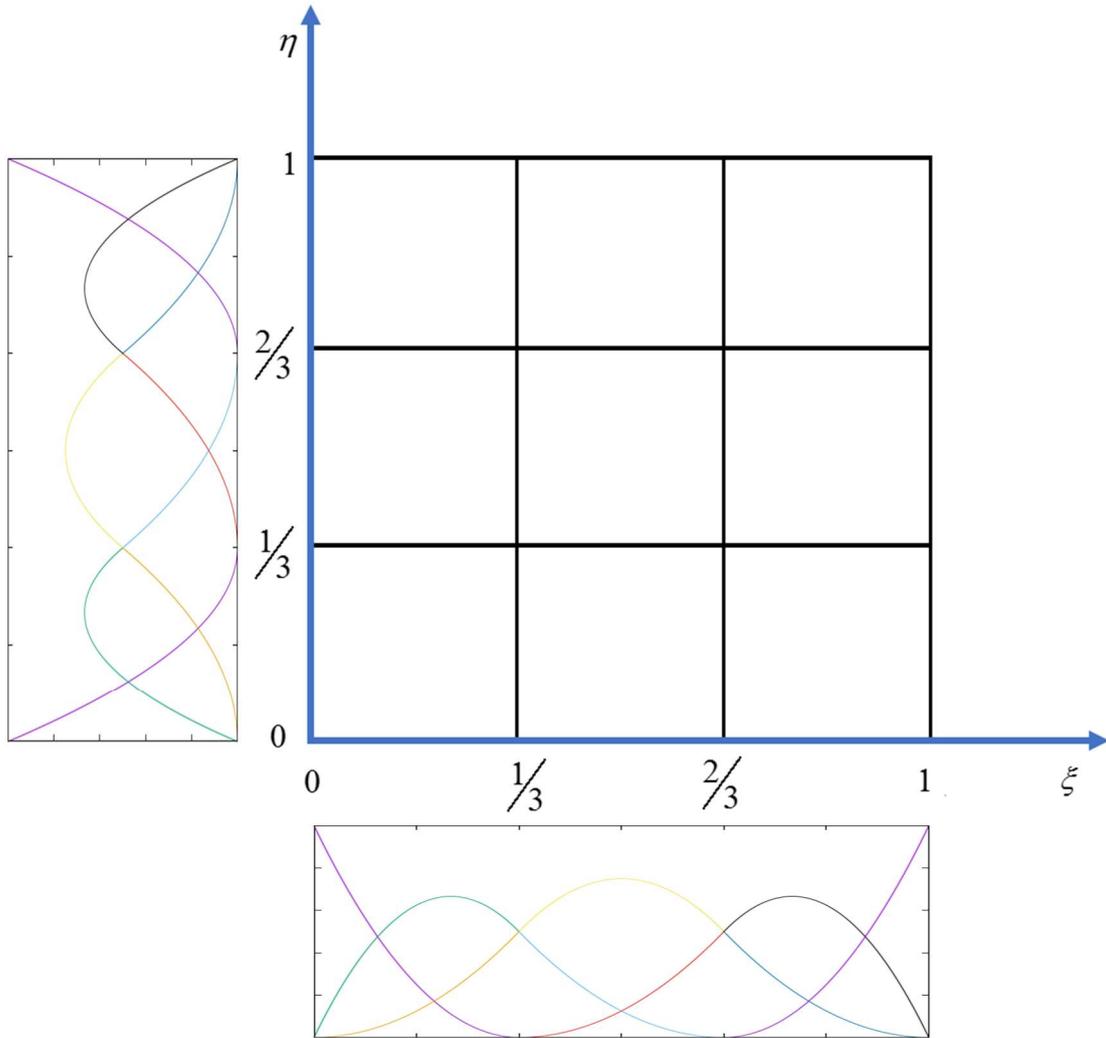
Figure 2.1 A two-dimensional parameter space and corresponding basis function in the ξ- and η- directions, respectively. They are built by $\Xi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$ and $\Pi = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$. Note that the values of the knots being repeated are only marked once.

2.2.3 Index space

Index space is also built based on the knot vector. What differs in the parameter space is that it recognizes every knot, even if it is multiple repeating knot values. The import of index spaces is very important for isogeometric analysis. Based on this space, many connectivity matrices can be created in the numerical analysis process, which is very important for the development of analytical algorithms. An example of an index space is shown in Figure 2.2, which is constructed also using the same knot vectors as ones used for creating parameter space in Figure 2.1.
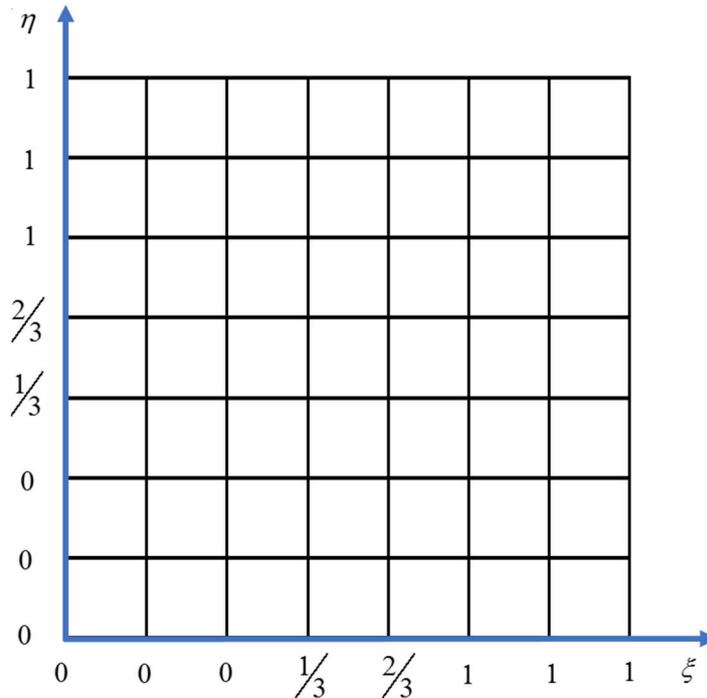

Figure 2.2 The two-dimensional index space, that is constructed using the same knot vectors as ones in Figure 2.1. Note that even knots with the same knot value are marked separately.

2.2.4 Physical space

The physical space is the exact representation of the actual shape for the geometry, that is the actual geometry exists in this space. The physical space mainly includes *physical mesh*, *control point*, and *control mesh*.

The physical mesh is the partition for the actual geometry. The geometrical structure is decomposed into *elements*. Note that these elements are different from the elements in the traditional finite element method, in the IGA, there are two notions for elements which are called the *knot span* and the *patch*, respectively. The knots span is the smallest element in the physical space, so that it may be thought of as the basic element. As defined in Section 2.2.1, a knot span is defined by two adjacent knots with the different knot values. Thus, the basis on the knot in the parameter space, the physical mesh can be built on the geometry, they are points and lines in one- and two-dimensional topologies, respectively. The patch is a subdomain composed of multiple knot span, so it may be thought of as a macro element. For simple geometric models, we can use a single patch to represent them, but for complex models, we need to use multiple patches. The concept of multiple patches will be described in detail in later chapters. Note that, in this study, when we speak of "element" without further description, we usually mean knot spans.

The images of the physical mesh in the case of one and two dimensions are shown in Figure 2.2. Firstly, in Figure 2.3 (a), the examples of one dimension is a NURBS curve, that was built from the open knot vector $\Xi = \{0,\ 0,\ 0,\ 1/6,\ 2/6,\ 3/6,\ 4/6,\ 5/6,\ 1,\ 1,\ 1\}$, the knots denoted with a red cross. The curve is divided into elements (denoted by different colors), whose geometrical shape are the curve segments that are identical to the original shape. Second, the image in Figure 2.3 (b) is a quadratic surface built by the knot vectors $\Xi = \{0,0,0,1/3,2/3,1,1,1\}$ and $\Pi = \{0,0,0,1/3,2/3,1,1,1\}$. In the case of two dimensions, the knots are the lines, and they partition the surface into quadrilateral elements. As you can see, it is the knots, mapped into the physical space, that

partitioned the surface (or curve) into elements, thus forming the physical mesh. So the physical mesh is exact decomposition of the actual geometry.
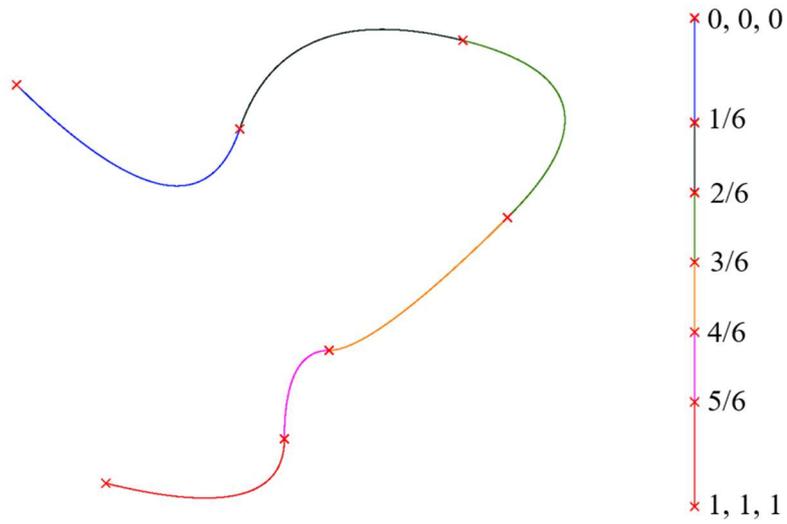
(a) Curve and physical mesh denoted by knot locations
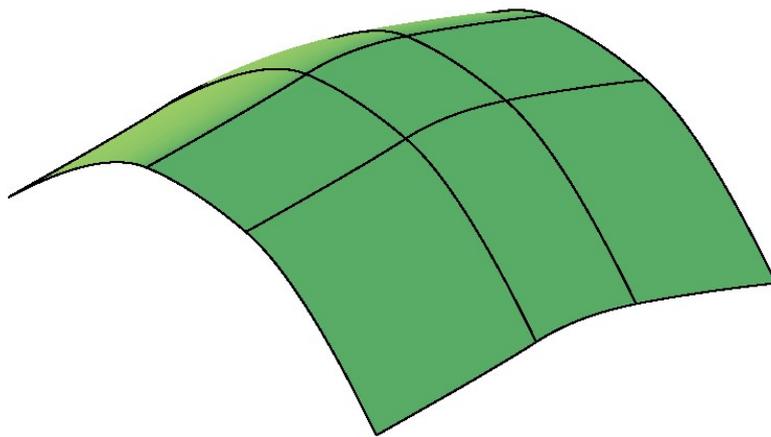
(b) Surface and physical mesh built by the line.

Figure 2.3 The physical mesh for cure and surface.

In the IGA, geometric models are constructed by taking a linear combination of NURBS basis functions, where the vector-valued coefficients of the basis functions are referred to as *control points*. Sometimes it also called *weighted control points*, because coordinates of the control points are *homogeneous coordinates*. Piecewise linear interpolation of the control points gives the so-called control mesh (that is also known as the "control net"). The control mesh consists of multilinear elements, in one dimension they are line segments, in two dimensions they are bilinear quadrilateral elements. The control mesh is different from the traditional finite element mesh. In the finite element method, the basis functions are interpolatory and are therefore often referred to as shape functions or interpolation functions, the corresponding nodes are inserted into the geometric entities. In the isogeometric analysis, because the basis-functions are usually no interpolatory, most control points are not inserted into geometric entities, and control net does not conform to the practical geometric shape. note that, the *degrees-of-freedoms* (DOFs) called as control variables locate at the control points. In Figure 2.4, an example of a one-dimensional and two-dimensional control mesh is given, respectively. In the one-dimensional case, the geometric model is the curve that is the same as the one in Figure 2.3 (a), the control mesh is a polyline segment constructed by the control points, and that is usually called as *control polygon*. The geometry in Figure 2.4 (b) is the surface in Figure 2.3 (b), the constructed control mesh looks like a finite element mesh, but it is not attached to the surface. Note that the difference between the control mesh shown in Figure 2.4 and the images of the physical mesh shown in Figure 2.3. The control mesh (indicated by the dashed line) was constructed from piecewise linear interpolation of control points denoted by red dots in Figure 2.4. In addition, because of the use of the open knot vector, the control points at both ends of the mesh are inserted into the geometric entity. Through the above discussion, we may clearly see that, in the isogeometric analysis, there are two notions of mesh i.e. control mesh and physical mesh, and two notions of elements i.e. patch and knot spans. In finite element, there is one notion of a mesh and one notion of an element. Elements are usually defined by their nodal coordinates and the degrees-of-freedoms. The finite element mesh is structured from elements connected by nodes.

Note that, geometry in Figure 2.3 and Figure 2.4 are described by the open knot vectors and basis functions with the 2nd-degree, thus the knot value at the first and last knot of knot vector is repeated degree of basis function, p plus 1 time, namely 3 times. In addition, since the knot vector is open, in one dimension, the control points at both ends of the control polygon are inserted at both ends of the curve. For a surface in case of two dimensions, the control points at the boundary of the control mesh are also interpolatory.
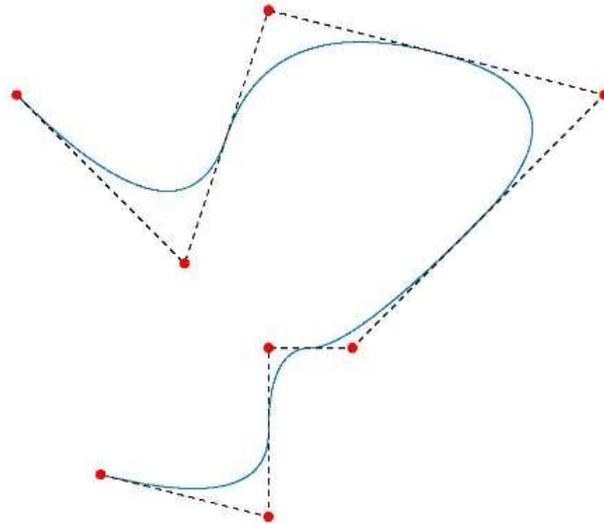
(a) Curve, control points and control mesh
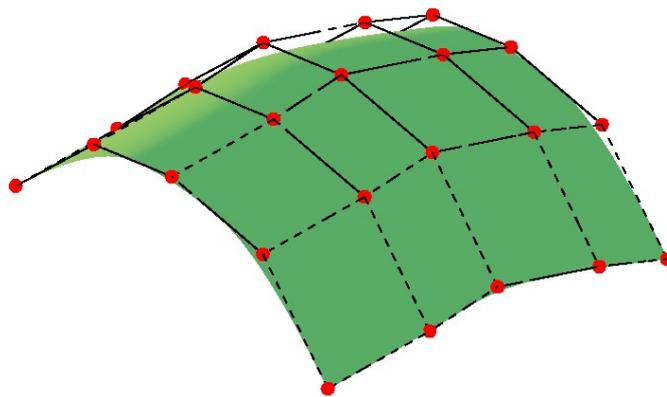
(b) Surface, control points and control mesh

Figure 2.4 piecewise quadratic curve and surface with their control mesh.

2.2.5. Parent space

In the process of numerical analysis, in order to perform the numerical integration, elements of different shapes must be allowed to take more uniform shapes, namely form of the parent element, this is done by using some mapping to transform them. In this study, we define the parent element as an independent space, that is, the parent space. The parent space is defined with respect to a natural coordinate system. The effect of parent space is same as the one in the finite element method and that is usually called as *isoparametric Elements* in the finite element method. The configuration of the parent element is depicted in Figure 2.5, where the natural coordinates $\hat{\xi}$ and $\hat{\eta}$ for this parent element the conditions $-1 \leq \hat{\xi} \leq 1$ and $-1 \leq \hat{\eta} \leq 1$.
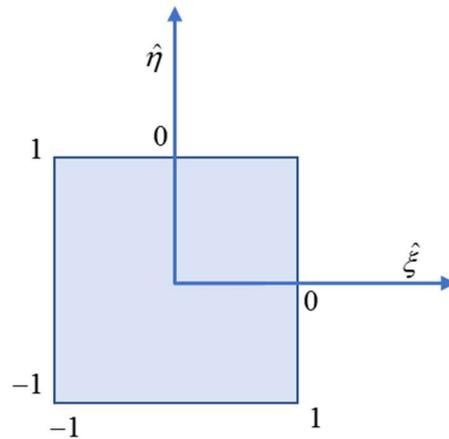
Figure 2.5 Parent space

A schematic illustration of the ideas is shown in Figure 2.5 for Isogeometric space, which is built based on Figure 2.1-2.4. And in Table 2.1 a summary of paraphernalia employed in the isogeometric analysis is also given.
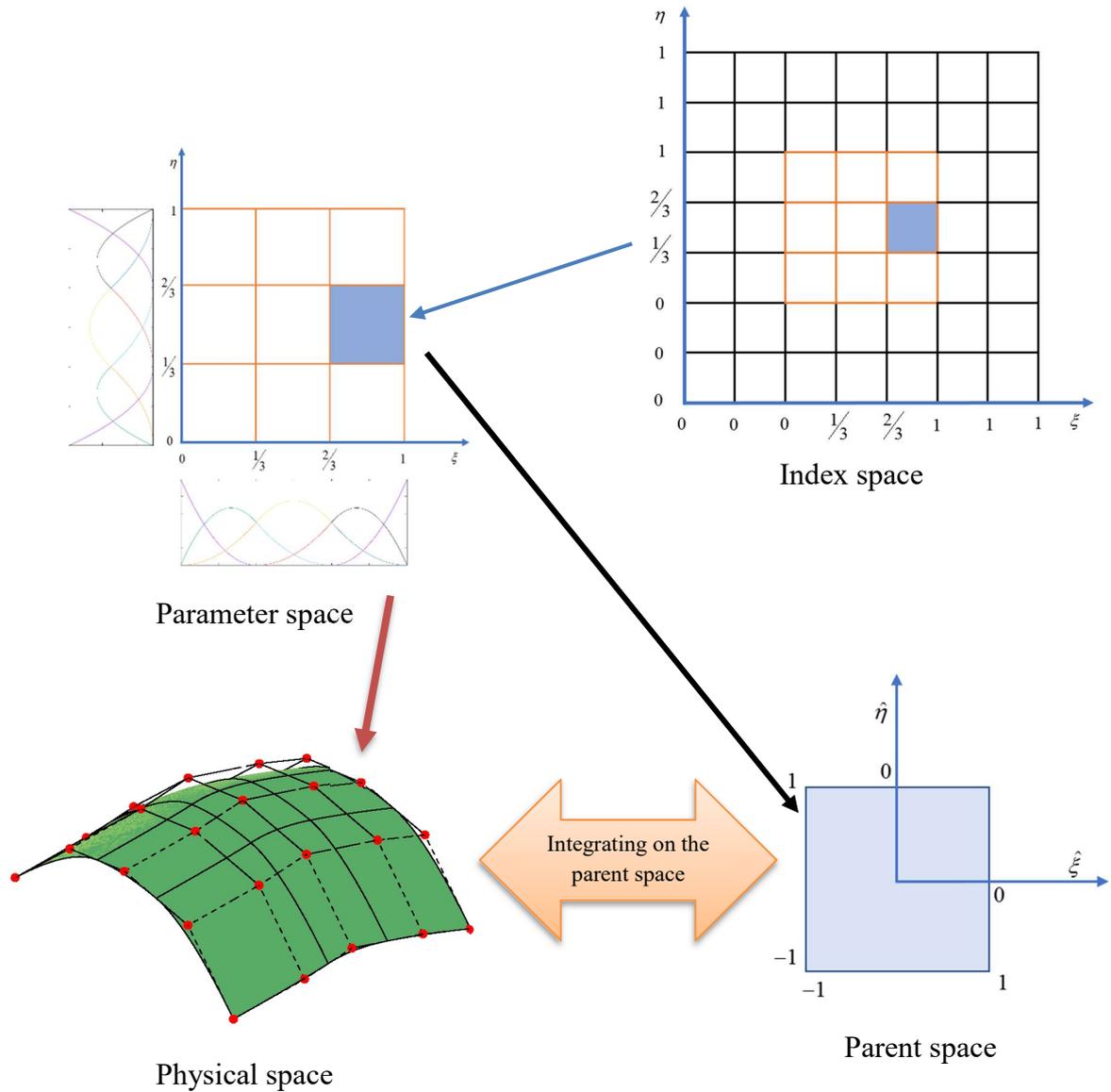
Figure 2.5 Schematic illustration of isogeometric space, that is built from knot vectors $\Xi = \{0,0,0,1/3,2/3,1,1,1\}$ and $\Pi = \{0,0,0,1/3,2/3,1,1,1\}$. Note that the parameter space plays a very important role in IGA, which is responsible for integrating and connecting the spaces.

Table 2.2 The summary of paraphernalia employed in the isogeometric analysis[5].

| Index Space		
Control Mesh	**PHYSICAL MESH**	
Multilinear control elements	Patches	Knot Spans
Topology: **1D: Straight lines defined by two consecutive control points**	Patches: Images of rectangular meshes in the parent domain mapped into the actual geometry. Patches may be thought of as macroelements or subdomains.	The topology of knots in the parent domain: 1D: Points 2D: Lines 3D: Panes
2D: Bilinear quadrilaterals defined by four control points	Topology: 1D: Curves 2D: Surfaces 3D: Volumes	The topology of knots in the physical space: 1D: Points 2D: Curves 3D: Surfaces
3D: Trilinear hexahedra defined by eight control points	Patches are decomposed into knot spans, the smallest notion of an element.	The topology of knots spans, i.e., "elements": 1D: Curved segments connecting consecutive knots 2D: Curved quadrilaterals bounded by four curves 3D: Curved hexahedra bounded by six curved surfaces

2.3. B-splines

Plenty of approaches have been used to structure the geometric model in CAD. In general, the non-uniform rational basis spline (NURBS) basis function has been used to build the geometric model, and has served as shape function in the pre/post processing of isogeometric analysis [2]. In this study, the analysis framework is built by the isogeometric analysis based on *Non-Uniform Rational B-Spline* (NURBS), in this chapter, we mainly introduce the definition of the NURBS paraphernalia and some of the main features.

2.3.1 B-splines basis function and the knot vector

Firstly, we give the definition of the B-splines basis function. The B-spline is very important for building the NURBS. The B-splines basis functions are formulated via the *Cox-de Boor recursion* formulation [6], [7], as follows:

$$N_{i,p=0}(\xi) = \begin{cases} 1 & if \ \xi_i \leq \xi < \xi_{i+1} \\ 0 & otherwise \end{cases} \tag{2.5}$$

where p is the *degree* of the B-splines basis functions[1]; $N_{i,p=0}(\xi)$ is the 0th-degree B-splines basis function and $N_{i,p}$ is the piecewise linear function; ξ are the knots of the non-descending knot vector in the parameter space, that can be mapped into the physical space, and define a physical mesh on the geometric entity by partitioning it into the elements. i is the knot index, i.e. $i = 1, 2, \cdots n + p + 1$. The corresponding knot vector is expressed by a knot vector as follows.

$$\Xi = \left\{ \xi_1, \xi_2, \cdots, \xi_{n+p+1} \right\}, \tag{2.6}$$

where the relationship between adjacent knots is

$$\xi_i \leq \xi_{i+1}, \tag{2.7}$$

For $p > 0$, the basis-functions are defined by the following equation:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi).$$ (2.8)

In Figure 2.6, the results of applying Equation (2.5) and (2.6) to a uniform knot vector are shown. The image for 0th-, 1st-, 2nd- and 3rd -order B-spline basis function with the knot vector $\Xi = \{1,2,3,4,5,6\}$ is presented in Figure 2.6, respectively. Note that, in the case where the B-spline basis functions with *p = 0* and *p = 1*, the same result as for the linear finite element functions and standard piecewise constant are respectively obtained. However, starting from a second or higher B-spline basis function is different from their FEA counterparts. For the higher-orders B-spline basis function, the basis function has smoother and higher continuity than FEM.

(b) 0th-order B-spline basis function

(a) 1st-order B-spline basis function

(d) 2nd-order B-spline basis function

(c) 3rd-order B-spline basis function
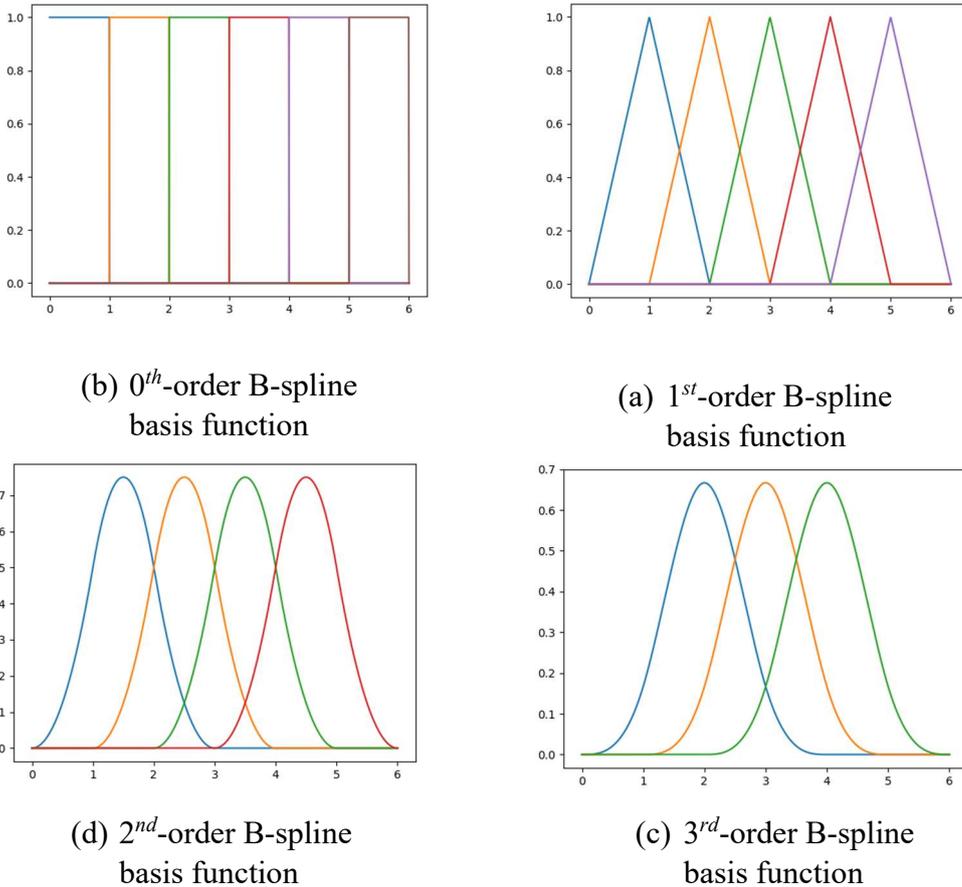
Figure 2.6 B-spline basis functions of order 0, 1, 2, 3 for uniform knot vector $\Xi = \{1,2,3,4,5,6\}$.

2.3.2 An example for computing B-spline basis function from a non-uniform knot vector

In Figure 2.6, we have given an example for uniform knot vector, but in actual analysis, we usually use a non-uniform knot vector such as open knot vector etc. Using a non-uniform knot vector can result in a richer behavior than a simple uniform knot vector. In this section, a specific example is given in regard to computing the quadratic B-spline basis function from the non-uniform knot vector. The B-spline basis function is built from the open knot vector $\Xi = \{\xi_0, \xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \xi_7, \xi_8, \xi_9, \xi_{10}\} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$ and the basis function order p equal to 2, so that the knot value of first and last at knot vector are repeated $p+1=3$ times. The specific calculation is as follows

Firstly, beginning with $p = 0$ and $i = 0$, and based on Equation (2.5), we have that

$$N_{0,0}(\xi) = \begin{cases} 1 & if \ \xi_0 \le \xi < \xi_1 \\ 0 & otherwise \end{cases}. \tag{2.9}$$

Since $\xi_0 = \xi_1 = 0$, there is no ξ, satisfying the condition, $\xi_0 \le \xi$ and $\xi < \xi_1$, and therefore $N_{0,0}(\xi) \equiv 0$. Based on the same argument, we can get

$$N_{0,0}(\xi) = 0, \tag{2.9a}$$

$$N_{1,0}(\xi) = 0. \tag{2.9b}$$

Following the same logic

$$N_{2,0}(\xi) = \begin{cases} 1 & 0 \le \xi < 1 \\ 0 & otherwise \end{cases}, \tag{2.9c}$$

$$N_{3,0}(\xi) = \begin{cases} 1 & 1 \le \xi < 2 \\ 0 & otherwise \end{cases}, \tag{2.9d}$$

$$N_{4,0}(\xi) = \begin{cases} 1 & 2 \leq \xi < 3 \\ 0 & otherwise \end{cases}, \qquad (2.9e)$$

$$N_{5,0}(\xi) = \begin{cases} 1 & 3 \leq \xi < 4 \\ 0 & otherwise \end{cases}, \qquad (2.9f)$$

$$N_{6,0}(\xi) = 0 \quad for \quad -\infty < \xi < \infty, \qquad (2.9g)$$

$$N_{7,0}(\xi) = \begin{cases} 1 & 4 \leq \xi < 5 \\ 0 & otherwise \end{cases}, \qquad (2.9h)$$

$$N_{8,0}(\xi) = 0 \quad for \quad -\infty < \xi < \infty, \qquad (2.9i)$$

$$N_{9,0}(\xi) = 0 \quad for \quad -\infty < \xi < \infty, \qquad (2.9j)$$

Not that, when $i = 11$, basis functions are zero for all the polynomial order, this is because of the length of the knot vector equal to number of the basis function + order of basis function +1. In this example, the length of the knot vector is 11, so that the number of the basis function is $11-0-1=10$. The nonzero B-spline basis functions of (2.9a) - (2.9j) are plotted in Figure 2.7.
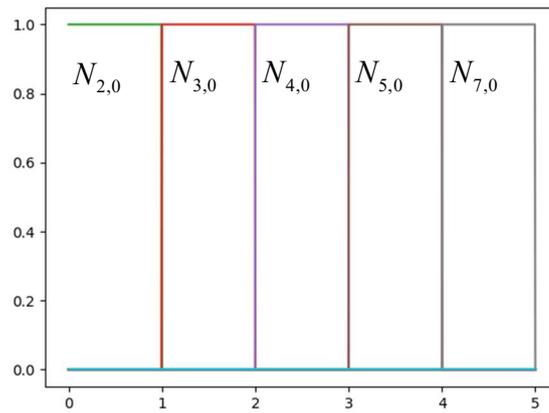

Figure 2.7 The nonzero zeroth degree B-spline basis functions with the non-uniform knot vector $\Xi = \{0,0,0,1,2,3,4,4,5,5,5\}$

Next, the first-order B-spline basis functions are computed by the Equation (2.8). The basis function is the linear functions, and its number is $11-1-1=9$. Because we have that

$$N_{0,1}(\xi) = \frac{\xi-0}{0-0}N_{0,0} + \frac{0-\xi}{0-0}N_{1,0} = 0, \qquad -\infty < \xi < \infty \qquad (2.10a)$$

$$N_{1,1}(\xi) = \frac{\xi-0}{0-0}N_{1,0} + \frac{1-\xi}{1-0}N_{2,0} = \begin{cases} 1-\xi & 0 \le \xi < 1 \\ 0 & otherwise \end{cases}, \qquad (2.10b)$$

$$N_{2,1}(\xi) = \frac{\xi-0}{1-0}N_{2,0} + \frac{2-\xi}{2-1}N_{3,0} = \begin{cases} \xi & 0 \le \xi < 1 \\ 2-\xi & 1 \le \xi < 2 \\ 0 & otherwise \end{cases}, \qquad (2.10c)$$

Further

$$N_{3,1}(\xi) = \frac{\xi-1}{2-1}N_{3,0} + \frac{3-\xi}{3-2}N_{4,0} = \begin{cases} \xi-1 & 1 \le \xi < 2 \\ 3-\xi & 2 \le \xi < 3 \\ 0 & otherwise \end{cases}, \qquad (2.10d)$$

$$N_{4,1}(\xi) = \frac{\xi-2}{3-2}N_{4,0} + \frac{4-\xi}{4-3}N_{5,0} = \begin{cases} \xi-2 & 2 \le \xi < 3 \\ 4-\xi & 3 \le \xi < 4 \\ 0 & otherwise \end{cases}, \qquad (2.10e)$$

$$N_{5,1}(\xi) = \frac{\xi-3}{4-3}N_{5,0} + \frac{4-\xi}{4-4}N_{6,0} = \begin{cases} \xi-3 & 3 \le \xi < 4 \\ 0 & otherwise \end{cases}, \qquad (2.10f)$$

$$N_{6,1}(\xi) = \frac{\xi-4}{4-4}N_{6,0} + \frac{5-\xi}{5-4}N_{7,0} = \begin{cases} 5-\xi & 4 \le \xi < 5 \\ 0 & otherwise \end{cases}, \qquad (2.10g)$$

$$N_{7,1}(\xi) = \frac{\xi-4}{5-4}N_{7,0} + \frac{5-\xi}{5-5}N_{8,0} = \begin{cases} \xi-4 & 4 \le \xi < 5 \\ 0 & otherwise \end{cases}, \qquad (2.10h)$$

$$N_{8,1}(\xi) = \frac{\xi-5}{5-5}N_{8,0} + \frac{5-\xi}{5-5}N_{9,0} = 0. \qquad -\infty < \xi < \infty \qquad (2.10i)$$

The non-zero B-spline basis functions are depicted in Figure 2.8. Note that they are piecewise linear basis functions in Figure 2.8. In addition, the knot value is repeated twice at the position of knot ξ_4, and its algebraic representation is different from other knots, which is caused by the continuity of the basis function, and which we will discuss in detail later in this chapter.
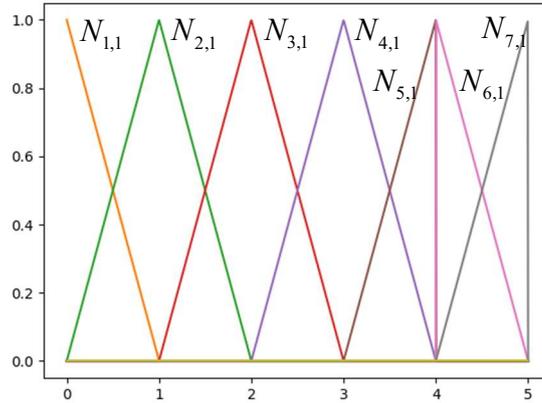

Figure 2.8 The non-zero first-degree piecewise linear B-spline basis functions,

$$\Xi = \{0,0,0,1,2,3,4,4,5,5,5\}$$

Finally, the piecewise quadratic functions are computed from (2.11a) - (2.11h). The number of basis function is the $11 - 2 - 1 = 8$. We have

$$N_{0,2}(\xi) = \frac{\xi - 0}{0 - 0} N_{0,1} + \frac{1 - \xi}{1 - 0} N_{1,1} = (1 - \xi)^2, \qquad -\infty < \xi < \infty \qquad (2.11a)$$

$$N_{1,2}(\xi) = \frac{\xi - 0}{1 - 0} N_{1,1} + \frac{2 - \xi}{2 - 0} N_{2,1} = \begin{cases} 2\xi - \frac{3}{2}\xi^2 & 0 \le \xi < 1 \\ \frac{1}{2}(2 - \xi)^2 & 1 \le \xi < 2 \end{cases}, \qquad (2.11b)$$

$$N_{2,2}(\xi) = \frac{\xi - 0}{2 - 0} N_{2,1} + \frac{3 - \xi}{3 - 1} N_{3,1} = \begin{cases} \frac{1}{2}\xi^2 & 0 \le \xi < 1 \\ -\frac{3}{2} + 3\xi - \xi^2 & 1 \le \xi < 2 \\ \frac{1}{2}(3 - \xi)^2 & 2 \le \xi < 3 \end{cases}, \qquad (2.11c)$$

$$N_{3,2}\left(\xi\right)=\frac{\xi-1}{3-1}N_{3,1}+\frac{4-\xi}{4-2}N_{4,1}=\begin{cases}\frac{1}{2}\left(\xi-1\right)^{2} & 1\le\xi<2\\ -11\frac{1}{2}+5\xi-\xi^{2} & 2\le\xi<3 \\ \frac{1}{2}\left(4-\xi\right)^{2} & 3\le\xi<4\end{cases} \qquad (2.11\text{d})$$

$$N_{4,2}\left(\xi\right)=\frac{\xi-2}{4-2}N_{4,1}+\frac{4-\xi}{4-3}N_{5,1}=\begin{cases}\frac{1}{2}\left(\xi-2\right)^{2} & 2\le\xi<3\\ -16+10\xi-\frac{3}{2}\xi^{2} & 3\le\xi<4\end{cases}, \qquad (2.11\text{e})$$

$$N_{5,2}\left(\xi\right)=\frac{\xi-3}{5-3}N_{5,1}+\frac{5-\xi}{5-4}N_{6,1}=\begin{cases}\left(\xi-3\right)^{2} & 3\le\xi<4\\ \left(5-\xi\right)^{2} & 4\le\xi<5\end{cases}, \qquad (2.11\text{f})$$

$$N_{6,2}\left(\xi\right)=\frac{\xi-4}{5-4}N_{6,1}+\frac{5-\xi}{5-4}N_{7,1}=2\left(\xi-4\right)\left(5-\xi\right),\quad 4\le\xi<5 \qquad (2.11\text{g})$$

$$N_{7,2}\left(\xi\right)=\frac{\xi-4}{5-4}N_{7,1}+\frac{5-\xi}{5-5}N_{8,1}=\left(\xi-4\right)^{2}, \qquad 4\le\xi<5, \qquad (2.11\text{h})$$

These piecewise quadratic B-spline basis functions are plotted in Figure 2.9.
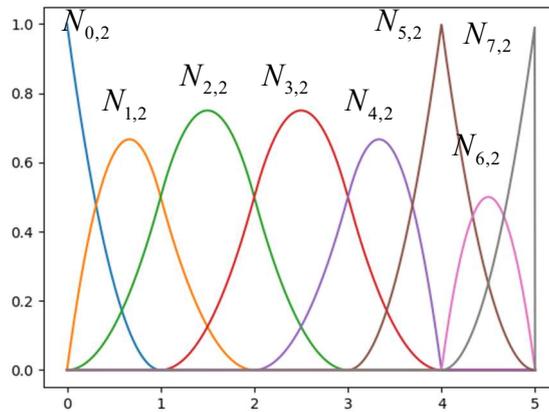

Figure 2.9 The non-zero second-degree B-spline basis functions

Remark

1. Note that the meanings of the terminologies "degree" and "order" are
    deference in the CAD and geometry community. For example, a quadratic
    polynomial in geometry is usually considered as possessing 2nd-degree and
    3rd-order, i.e. order = degree + 1. However, in the analysis community, the

quadratic polynomial will be said as 3rd-order, i.e. the meanings of degree and order are the same. In this paper, we will stick to this convention.

2.   The use of the knot vector shown in this example is non-uniform, that is, the knot values at ξ_4 and both ends are repeated 2 and 3 times, respectively. For example, in Figure 2.9, in the case where the second-order of the base function, the position of ξ_4 is C^0- continuity, and the other locations are C^1- continuity. Generally, basis functions of order p have $p - m_i$ continuous derivatives across knot ξ_i, where m_i is the multiplicity of the value of ξ_i in the knot vector [4]. When the multiplicity of a knot value is exactly p, the basis is interpolatory at that knot. When the multiplicity is $p + 1$, the basis becomes discontinuous and the patch boundary is formed.

2.3.3 Derivatives of B-spline basis functions

The derivative of the B-spline basis function is effectively represented by a B-spline low order basis function. The derivative of the ith-order B-spline basis is expressed by

$$\frac{dN_{i,p}(\xi)}{d\xi} = \frac{pN_{i,p-1}(\xi)}{\xi_{i+p} - \xi_i} - \frac{pN_{i+1,p-1}(\xi)}{\xi_{i+p+1} - \xi_{i+1}}, \qquad (2.12)$$

where p is the order or B-spline basis function. Based on the Equation (2.12), the higher derivative of the B-spline basis function can be defined as

$$\frac{d^k N_{i,p}(\xi)}{d^k \xi} = \left(\frac{p}{\xi_{i+p} - \xi_i} \times \frac{d^{k-1} N_{i,p-1}(\xi)}{d^{k-1}\xi} \right) - \left(\frac{p}{\xi_{i+p+1} - \xi_{i+1}} \times \frac{d^{k-1} N_{i+1,p-1}(\xi)}{d^{k-1}\xi} \right), \quad (2.13)$$

Equation (2.13) is another form of expression for Equation (2.14), it defines the kth-derivative of the B-spline basis function in terms of the functions $N_{i,p-k}, \cdots, N_{i+k,p-k}$ so that we have

$$\frac{d^k N_{i,p}(\xi)}{d^k \xi} = \frac{p!}{(p-k)!} \sum_{j=0}^{k} \alpha_{k,j} N_{i+j,p-k}(\xi), \qquad (2.14)$$

with

$$\alpha_{0,0} = 1,$$

$$\alpha_{k,0} = \frac{\alpha_{k-1,0}}{\xi_{i+p-k+1} - \xi_i},$$

$$\alpha_{k,j} = \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+p+j-k+1} - \xi_{i+j}} \quad j = 1,...,k-1,$$

$$\alpha_{k,k} = \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}}.$$

Note that, in the presence of a repeating knot, the denominator of several of these coefficients can be zero. As long as this happens, the coefficient is defined as zero.

2.4. B-spline curves and B-spline surfaces

In the d-dimensional space $\mathbb{R}^d$, the pth-order B-spline curve $C(\xi)$ is built by

$$C(\xi) = \sum_{i}^{n} N_{i,p}(\xi)B_i, \tag{2.15}$$

where $N_{i,p}(\xi)$ are the B-spline basis functions of order p defined on the open knot vector, which is non-periodic and non-uniform. B_i represents the control points, which are the vector-valued coefficients of the B-spline basis functions. The degree, p number of control points n, and number or knots m, are related by

$$m = n + p + 1. \tag{2.16}$$

Through Equation (2.15), we can clearly see that, a fixed point on a B-spline curve is obtained from a fixed knot value ξ. Firstly, we need to confirm the knot span in that ξ lies and compute the non-zero B-spline basis functions. Finally, the value of non-zero B-spline basis functions and the corresponding control points are multiplied.

An example piecewise quadratic B-spline curve in $\mathbb{R}^2$ is shown in Figure 2.9, where the control points are denoted by red solid circles. The dashed line connecting the control points is referred to as the control polygon or control net, which is a piecewise linear interpolation of the control points. In Figures 2.9 (a) and 1(b), the two B-spline curves are built from the open knot vector, $\Xi = \{0, 0, 0, 1, 2, 2, 3, 4, 4, 4\}$, that the positions of $\xi = 0$ and $\xi = 4$ are repeated $p+1=3$ times. Therefore, the curve is interpolatory at the first and last control points. Moreover, it is also interpolatory at the fourth control point, since the multiplicity of the knot $\xi = 2$ is equal to the order of basis function.

Note that the two B-spine curves are built by using the same knot vector, basis function and order, with the only difference being that the coordinate of control point 2, i.e., the curve in Figure 1(b) is obtained by moving the control point 2 in Figure

1(a). The corresponding is shown in Table 2.3. Additionally, Since the base function $N_{i,p}(\xi)$ is a piecewise polynomial, the curve in Figure 2.9 is a piecewise curve. In Figure 2.9 (c), we show the B-spline basis functions, and the B-spline curves for the corresponding individual knot span, which are marked with different colors.

Table 2.3 The control point coordinates in Figure 2.9 (a) and (b)

|	Control point coordinates in figure 2.9 (a)		Control point coordinates in figure 2.9 (b)	
	x	y	x	y
1	0.5	3	0.5	3
2	1.5	5.5	0	5.5
3	5	6	5	6
4	3	1.5	3	1.5
5	7.5	1.5	7.5	1.5
6	6	4	6	4
7	9	5	9	5
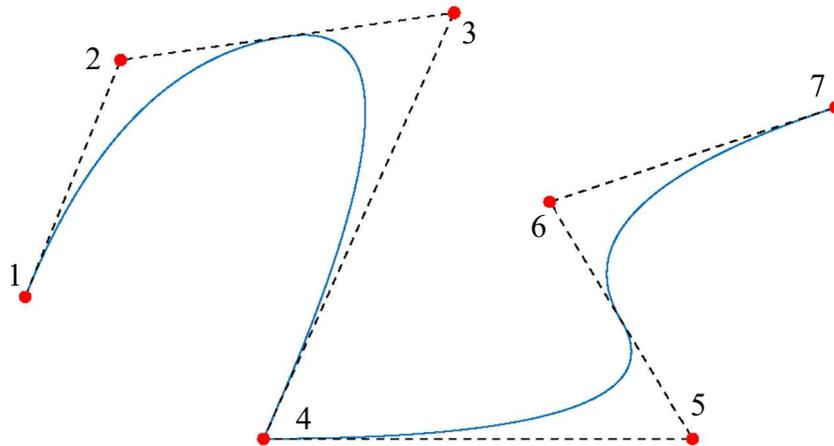
(a) Shape of curve before moving control point 2
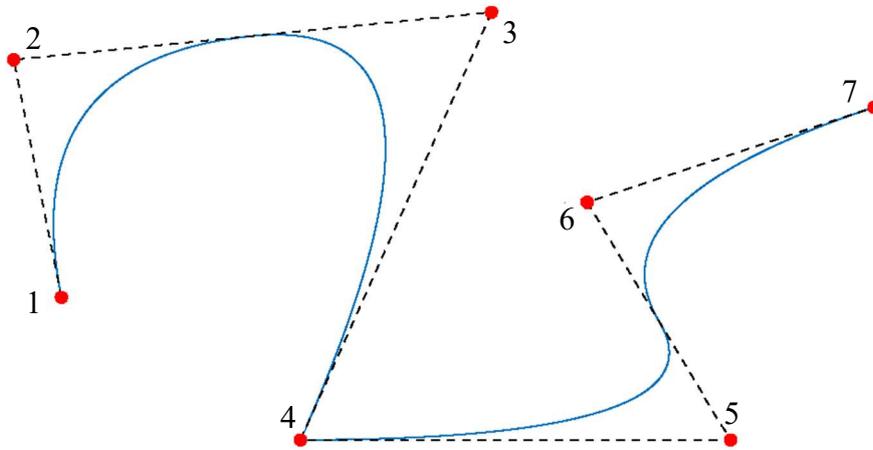
(b) Shape of curve after moving control point 2
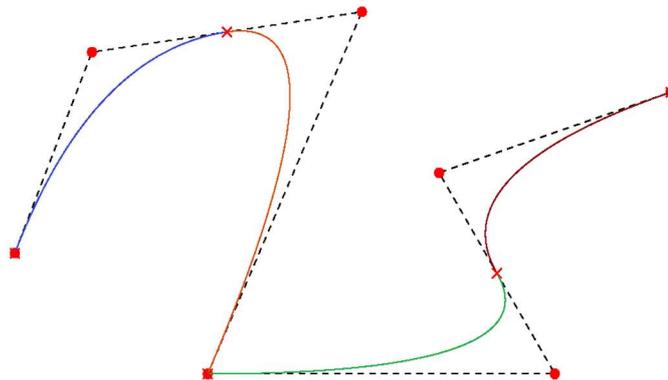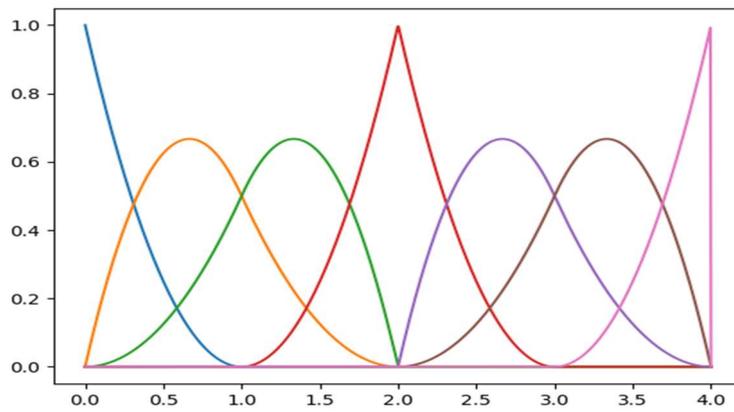
(c) Shape of curve after moving control point 2

Figure 2.9: Piecewise quadratic B-spline curve, control polygon,

and control points in $\mathbb{R}^2$.

The B-spline surface is constructed by taking the two-dimension control net $\{B_{i,j}\}, i = 1,2,\cdots,n$; $j = 1,2,\cdots,m$, the knot vectors Ξ and H, and the B-spline basis functions in the direction of the two knot vectors, as follows:

$$S(\xi, \eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta)B_{i,j}, \qquad (2.17)$$

with

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\},$$
$$H = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\},$$

where $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$ are the basis functions in the ξ – and η – directions, respectively. *p* and *q* are the order of basis function.

For computing a point on a tensor product B-spline surface at fixed parameter coordinate (ξ, η) . We first need to find the knot span in which ξ and η lies , namely $\xi \in [\xi_i, \xi_{i+1})$ and $\eta \in [\eta_i, \eta_{i+1})$, and then compute the non-zero B-spline basis functions. Finally, the value of non-zero B-spline basis functions and the corresponding control points are multiplied. Therefore, the matrix form of Equation (2.17) is

$$S(\xi, \eta) = \left[N_{k,p}(\xi) \right]^T \left[\mathbf{B}_{k,h} \right] \left[M_{h,q}(\eta) \right], \qquad (2.18)$$

with

$$i - p \leq k \leq i;$$
$$j - q \leq h \leq j;$$

where $\left[N_{k,p}(\xi) \right]^T$ is a row vector of scalars with 1×(*p*+1); $\left[M_{h,q}(\eta) \right]$ is a column vector of scalars with 1×(*q*+1), and $\left[\mathbf{B}_{k,h} \right]$ is a matrix of control points, and its size is (*p*+1) ×(*q*+1). An example of the B-spline surface is shown in Figure 2.10, where

the mesh lie in the surfaces are defined by knots in the knot vectors $\Xi = \{0,0,0,0,1/7,2/7,3/7,4/7,5/7,6/7,1,1,1,1\}$ and $H = \{0,0,0,1/8,2/8, 3/8,4/8,5/8,6/8,7/8,1,1,1\}$,that partitioned the surfaces into the elements. The element boundaries in the surface are simply the images of knot lines under the geometric mapping. An initial surface is shown in Figure 2.10 (a), that is a curved plane since it has no thickness in the other direction. Besides, in order to obtain the better visualization, the corresponding control net is marked with dotted lines and the control points are marked with solid red dots. The surface in Figure 2.10 (b) is the image after the surface in Figure 2(a) was changed, which was obtained by moving the control points A, B and C (marked in Figure 2.10 (b)) in the control net.


(a)Original geometry

(b) Geometry after moving control point B
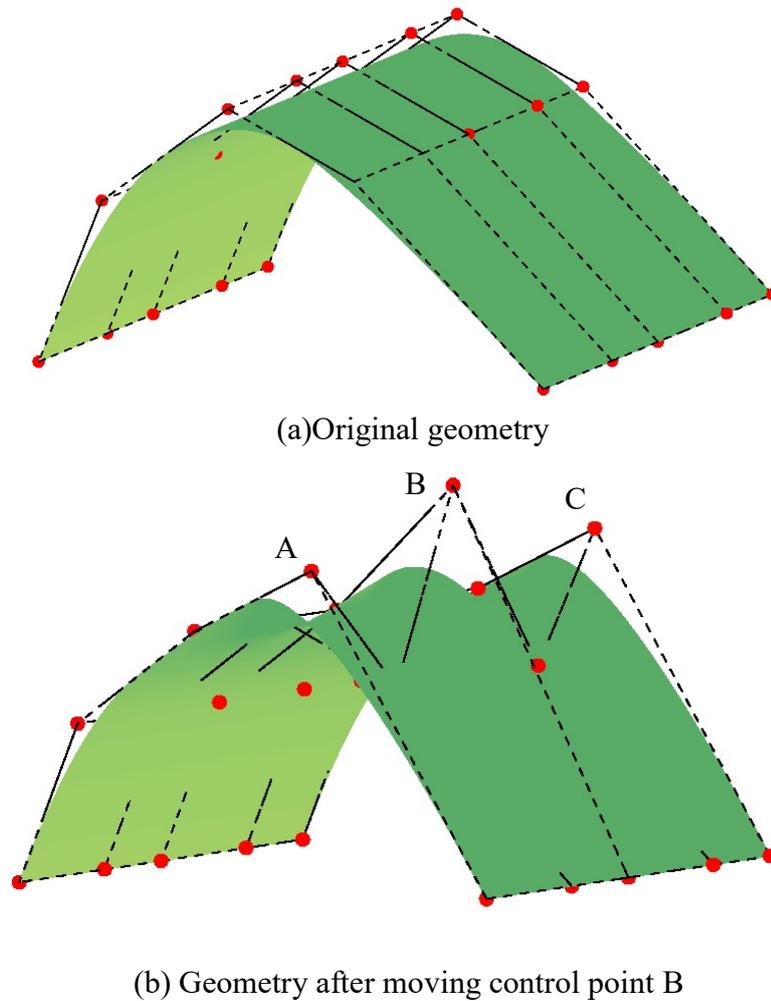
Figure 2.10: The cubic × quadratic surface B-spline surfaces and control nets in $\mathbb{R}^2$.

As described above, one of the important properties of B-splines was shown, that is its ability to directly change geometrical shape by adjusting the control points. For examples, geometry in Figures 2.9 and 2.10, by adjusting the control points, the geometric shape can be easily changed. Besides, note that due to the modification scheme property and local support property of B-splines, the area that each control point can affect is local and controllable. As shown in Figure 2.10, If control point A, B and C are moved to a new location, it only can affect the shape of the partially adjacent area on the surface and elsewhere is unaffected. In this study, this property of the B-spline was fully utilised, and the structural analysis with consideration to the uncertainty in shape was implemented by importing the parameters of uncertainty (mean and deviation etc.) Into the control point coordinates.

2.5. Properties of B-spline

In this section, some important properties of B-spline are listed, and these properties provide a theoretical basis for future numerical calculations.

i.   partition of unity [4], that is, for any knot, the sum of the basis functions must be equal to one, as follows

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1 \quad \forall \xi \tag{2.19}$$

And for each knot, its basis function is pointwise non-negative over the entire domain, namely

$$N_{i,p}(\xi) \geq 0 \quad \forall \xi \tag{2.20}$$

For example, Let the knot vector $\Xi = \{0,0,0,1,2,3,3,4,4,4\}$. We now calculate respectively the B-spline basis function of degree 0, 1, 2, and 3 at $\xi = 3.5$.

$$N_{6,0}(3.5) = 1;$$

$$N_{6,1}(3.5) = \begin{cases} 0.5 \\ 0.5 \end{cases};$$

$$N_{6,2}(3.5) = \begin{cases} 0.25 \\ 0.5 \\ 0.25 \end{cases};$$

$$N_{6,3}(3.5) = \begin{cases} 0.0625 \\ 0.4375 \\ 0.375 \\ 0.125 \end{cases};$$

From the above calculations, we can observe that for any degree, the value of base functions of knot at $\xi = 3.5$ are all non-negative and their sum is 1.

ii.   B-spline has the local support property. If knot value ξ is not in the half-open interval, $\left[\xi_i, \xi_{i+p+1}\right)$ in case of the one-dimensional topology, its corresponding basis functions are all zero, that is

$$
\begin{aligned}
N_{i,p}\left(\xi\right) &\neq 0 \quad \textit{if } \xi_i \leq \xi < \xi_{i+p+1}, \\
N_{i,p}\left(\xi\right) &= 0 \qquad \textit{otherwise.}
\end{aligned}
\tag{2.21}
$$

Through the definition of B-spline basis function Equation (2.5) and (2.6), we can observe this character. If a first-order B-spline basis function $N_{i,1}\left(\xi\right)$ is calculated, then the zeroth-order B-spline basis function $N_{i,0}\left(\xi\right)$ and $N_{i+1,0}\left(\xi\right)$ are required. And the basis functions $N_{i,0}\left(\xi\right)$ and $N_{i+1,0}\left(\xi\right)$ are non-zero at the knot span $\left[\xi_i, \xi_{i+1}\right)$ and $\left[\xi_{i+1}, \xi_{i+2}\right)$, respectively, so that the first-order basis function $N_{i,1}\left(\xi\right)$ is non-zero on these two knot spans, namely $N_{i,1}\left(\xi\right)$ is the non-zero on knot interval $\left[\xi_i, \xi_{i+2}\right)$. By analogy, because second-order B-spline basis function $N_{i,2}\left(\xi\right)$ depends on the first-order basis functions $N_{i,1}\left(\xi\right)$ and $N_{i+1,1}\left(\xi\right)$ and because they are respectively non-zero in knot interval $\left[\xi_i, \xi_{i+2}\right)$ and $\left[\xi_{i+1}, \xi_{i+3}\right)$, the second-basis function $N_{i,2}\left(\xi\right)$ is non-zero on knot interval $\left[\xi_i, \xi_{i+3}\right)$. n example of a one-dimensional B-spline base function that prove this property is shown in Figure 2.11.

In addition, in order to determine non-zero knot interval of B-spline basis function $N_{i,p}\left(\xi\right)$, we usually use a triangular computation scheme that as shown in Figure 2.12(a). Through this triangular scheme, the non-zero domain of B-spline basis function of any order can be traced back until they reach the first column in the triangular scheme. The knot interval that is ultimately covered is the non-zero domain of this basis function. for instance, suppose we want to determine the non-zero domain of the B-spline basis function $N_{0,3}\left(\xi\right)$. Based on the triangular schematic diagram in Figure 2.12, starting with $N_{0,3}\left(\xi\right)$, its

non-zero domain can be traced back in the direction indicated by the arrow until the first column of the triangular scheme is reached, that as shown with the red arrows and boxes in Figure 2.12 (b). From these, we can see that the basis function $N_{0,3}(\xi)$ is non-zero on the knot span $[\xi_0, \xi_1), [\xi_1, \xi_2), [\xi_2, \xi_3)$ and $[\xi_3, \xi_4)$, respectively, that is, its non-zero domain is on the knot interval $[\xi_0, \xi_4)$; In addition, for the two-dimensional topology follows directly from the one-dimensional topology, the basis functions are all zero, if parametric variable (ξ, η) is outside the rectangle $[\xi_i, \xi_{i+p+1}) \times [\eta_j, \eta_{j+p+1})$. This property is illustrated in Figure 2.13.
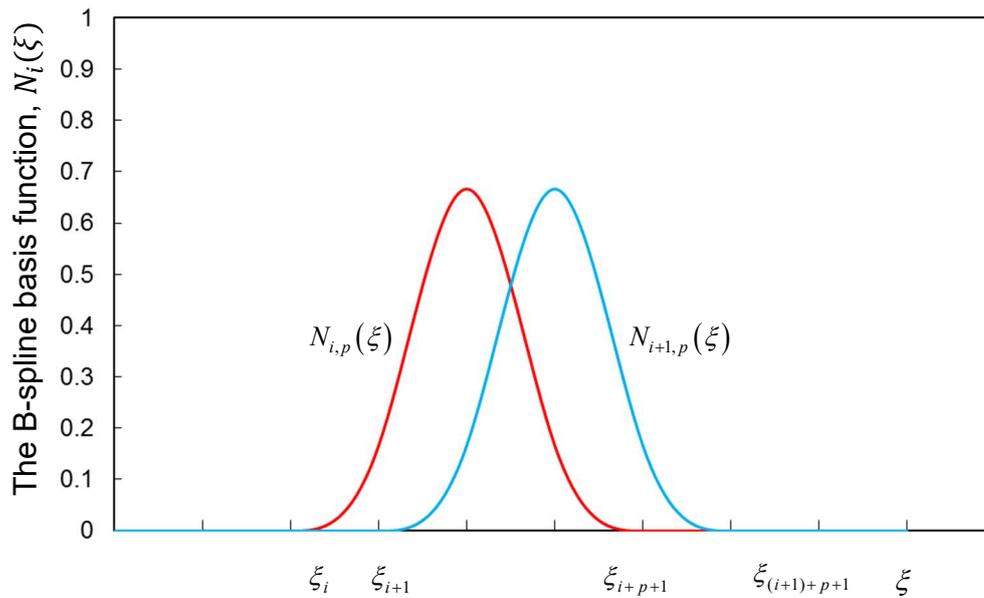


Figure 2.11 The 3rd -order B-spline basis functions.
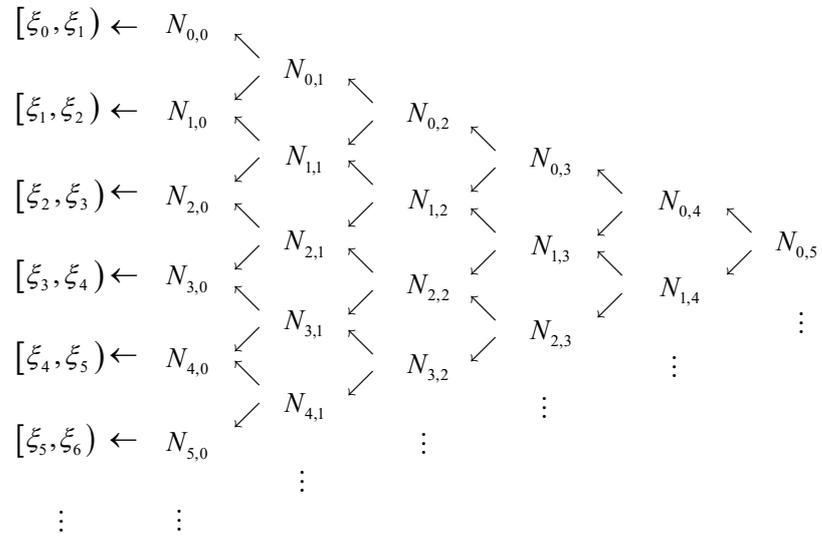
(a) Triangular schematic diagram
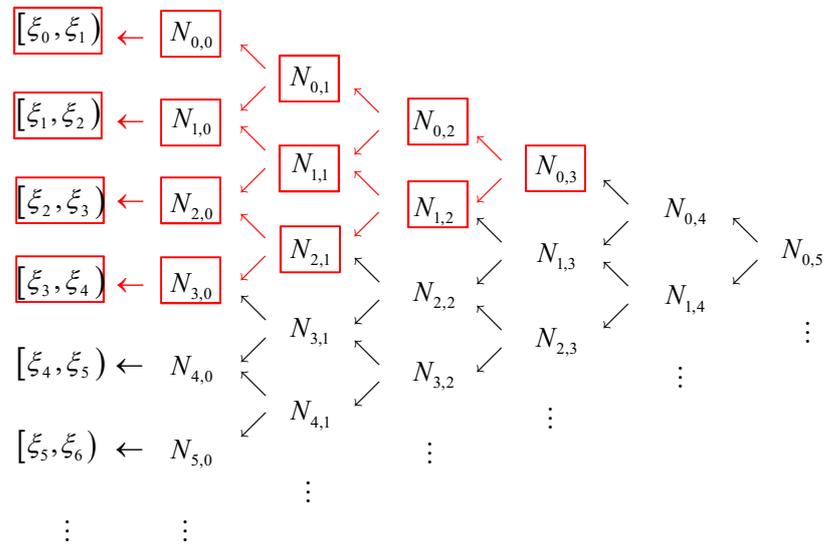
(b) The non-zero domain of the B-spline basis function

Figure 2.12 The triangular schematic diagram for the local support property of B-spline
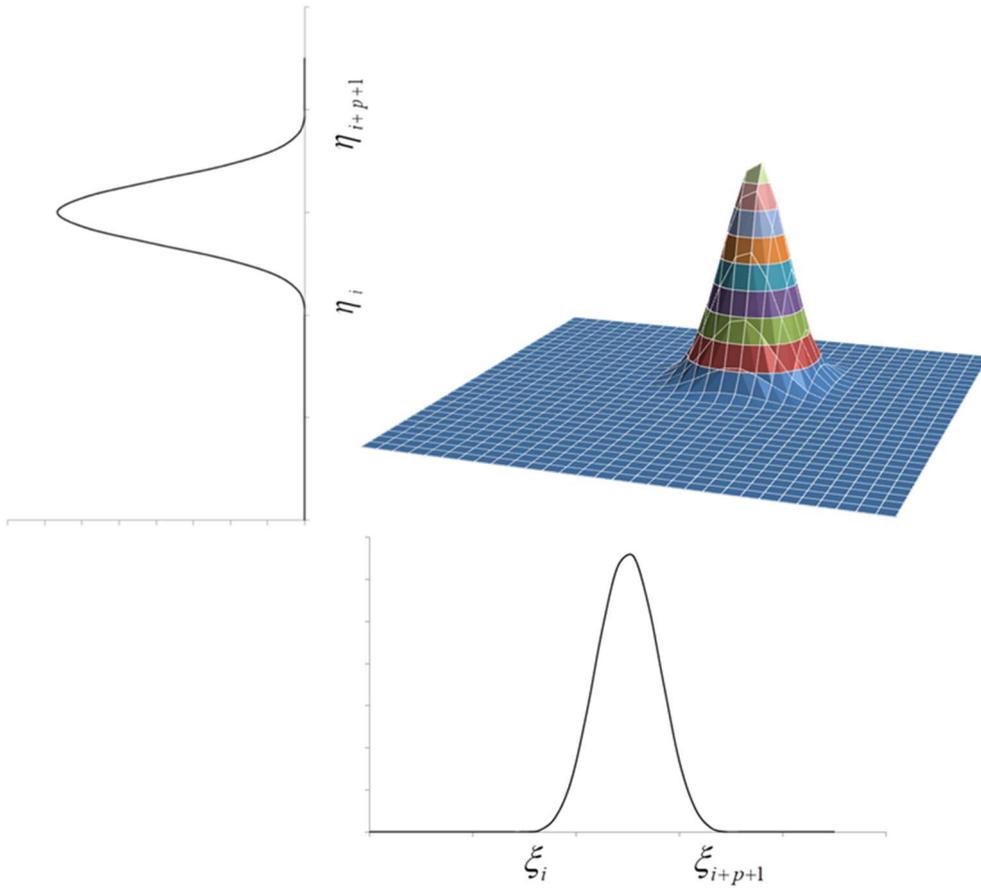
Figure 2.13 The local support property for 3rd -order B-spline basis functions in the two-dimension.

iii. In any given knot span, $\left[\xi_i, \xi_{i+1}\right)$, The maximum p+1 B-spline basis functions are non-zero, which are

$$N_{i-p,p}(\xi), \cdots, N_{i-2,p}(\xi), N_{i-1,p}(\xi), N_{i,p}(\xi)$$

Note that, if there are repeated knot values, the number of base functions is exactly the same as the degree p, and the corresponding basis functions are

$$N_{i-p,p}(\xi), \cdots, N_{i-2,p}(\xi), N_{i-1,p}(\xi)$$

Namely, given any knot span $\left[\xi_i, \xi_{i+1}\right)$, we can know which B-spline basis functions will use this knot span in their calculations. Here we likewise use a triangular schematic diagram in order to illustrate this character, that is shown in Figure 2.14. Starting with the knot span $\left[\xi_i, \xi_{i+1}\right)$ that located in the first column on the left of the triangular schematic diagram, along with the direction indicated by the arrow, we can determine the number of non-zero base functions on each degree. And all basis functions with degree *p* that are non-zero on knot span $\left[\xi_i, \xi_{i+1}\right)$ are the intersections of arrows.
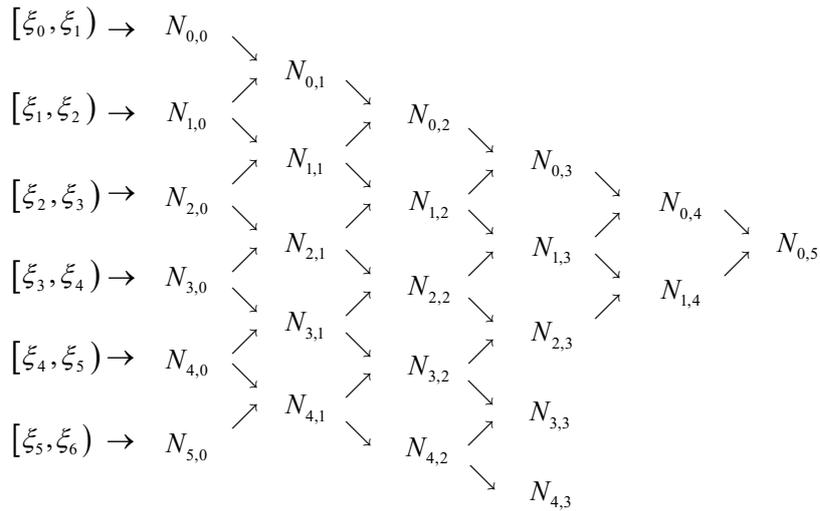

Figure 2.14 The triangular schematic diagram for property 4.

Let us look at a concrete example. Suppose we gave an open knot vector,

$$\Xi = \left\{\xi_0 = 0, \xi_1 = 0, \xi_2 = 0, \xi_3 = 1, \xi_4 = 2, \xi_5 = 3, \xi_6 = 4, \xi_7 = 5, \xi_8 = 5, \xi_9 = 5\right\} \qquad \text{and}$$

want to find out all third-degree basis functions that are non-zero on knot span $\left[\xi_3, \xi_4\right)$. As shown in Figure 2.15, we start with this knot span located in the first column, and find the non-zero basis functions in the direction indicated by the red arrow until they reach the third column in the triangular scheme diagram. In the end we can get all the 3rd -degree basis functions, $N_{0,3}\left(\xi\right)$, $N_{1,3}\left(\xi\right)$,

$N_{2,3}(\xi)$, and $N_{3,3}(\xi)$, which are non-zero on knot span $[\xi_3, \xi_4)$. In Figure 2.16, these non-zero basis functions are depicted by using lines of different colors.
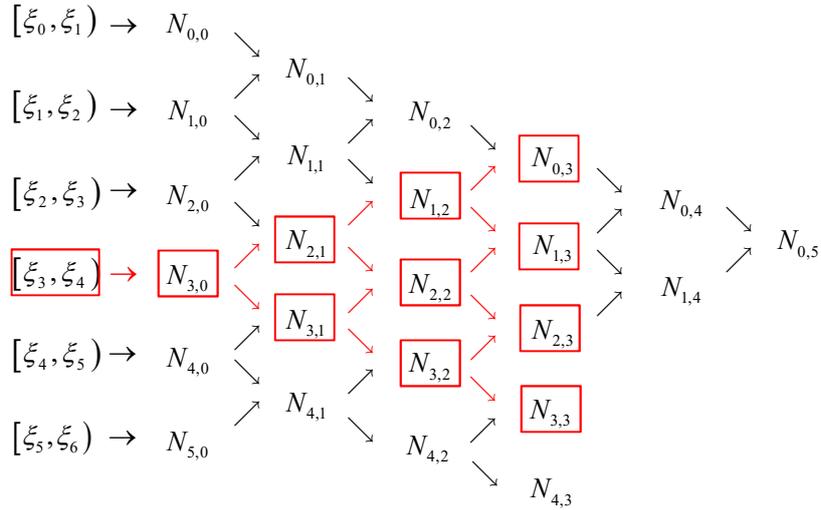

Figure 2.15 The triangular schematic diagram for finding third-degree basis functions that are non-zero on knot span, $[\xi_3, \xi_4)$
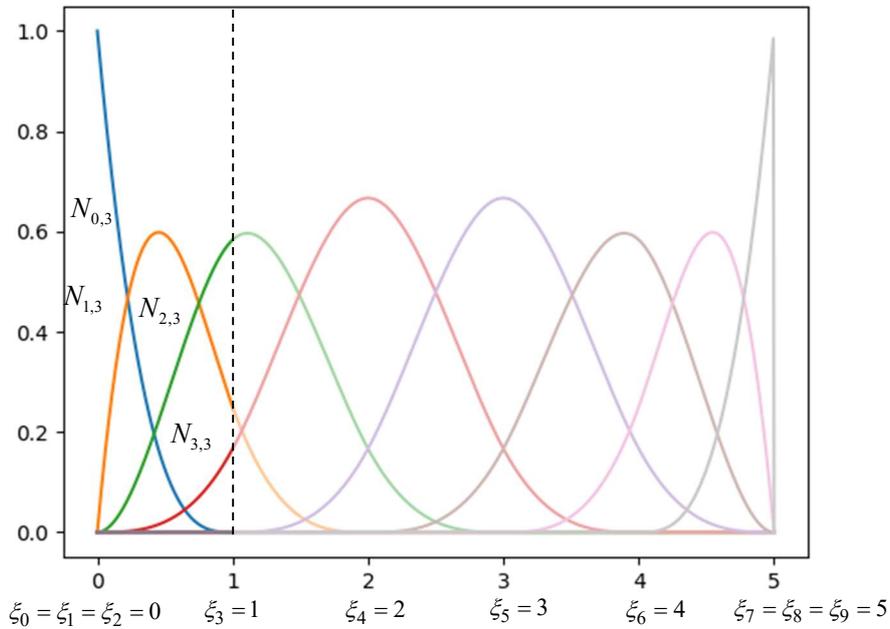

Figure 2.16 The non-zero basis functions on knot span, $[\xi_3, \xi_4)$

Moreover , we may apply the same logic to a larger knot interval. For example, as shown in Figure 2.17, we want to find all 3rd -degree basis functions that are non-zero on knot interval $[\xi_2, \xi_5)$. First of all, on the first column on the left side of triangular scheme diagram, we determine the knot interval $[\xi_2, \xi_5)$ which consists of two knot span $[\xi_2, \xi_3), [\xi_3, \xi_4)$ and $[\xi_4, \xi_5)$. Afterwards, starting from these two knot spans, the non-zero basis function is determined along the direction of the orange arrow until it reaches the column at $p = 3$. Finally, all cubic non-zero basis functions in the knot interval $[\xi_2, \xi_5)$ are , $N_{0,3}(\xi)$, $N_{1,3}(\xi)$, $N_{2,3}(\xi)$, $N_{3,3}(\xi)$, $N_{4,3}(\xi)$,and they are depicted in Figure 2.18. Note that since the use of open knot vector, there are only three 3rd -degree basis function that is non-zero on knot span $[\xi_2, \xi_3)$.
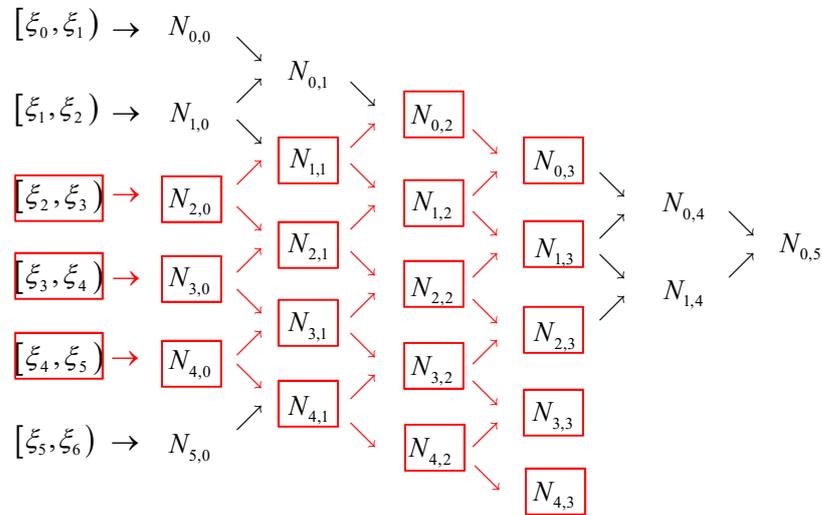

Figure 2.17 The triangular schematic diagram for finding third-degree basis functions that are non-zero in knot interval, $[\xi_2, \xi_5)$

$$\xi_0 = \xi_1 = \xi_2 = 0 \quad \xi_3 = 1 \quad \xi_4 = 2 \quad \xi_5 = 3 \quad \xi_6 = 4 \quad \xi_7 = \xi_8 = \xi_9 = 5$$
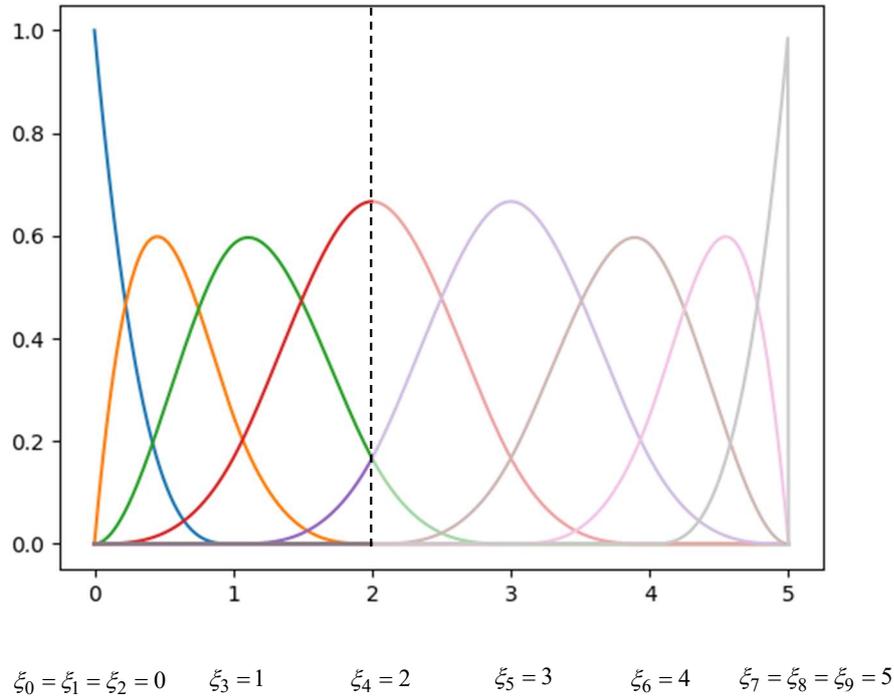
Figure 2.16 The non-zero basis functions in knot interval, $\left[\xi_2, \xi_5\right)$

iv. Endpoint interpolation; Generally, in the CAD, the curve interpolates the control points at both ends, and the B-spline surface are interpolations at the control points of the four corners. This is because the knot vectors that build these geometries are all open in the CAD. An example of a quadratic curve insertion endpoint is shown in Figure 2.17, where Both curves are depicted using exactly the same control point information. The curve in Figure 2.17 (a) is built from the open knot vector $\Xi = \{0,0,0,1,2,3,4,5,5,5\}$, so the curve is inserted into the control points at both ends; The curve in Figure 2.17 (b) is built from the knot vector $\Xi = \{1,2,3,4,5,6,7,8,9,10\}$. Since the knot vector used is not open, the control points at both ends of the curve are not interpolatory.
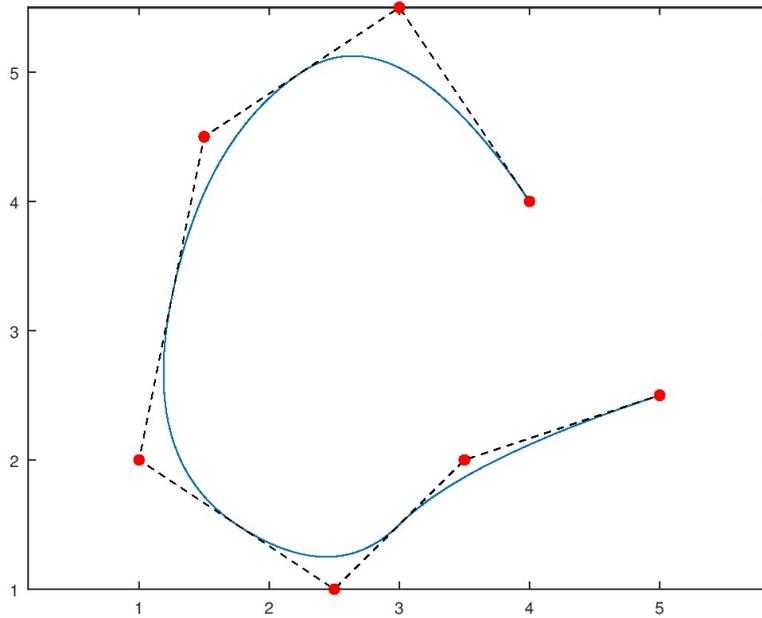
(a) The curve is inserted into the control points at both ends.
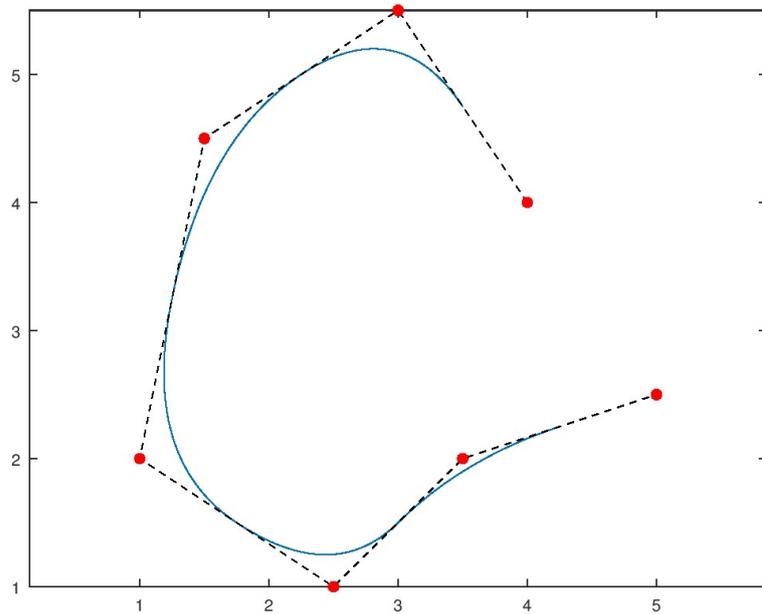
(b) The control points at both ends of the curve are not interpolatory

Figure 2.17 The quadratic curves

v. Affine invariance property; An affine transformation of a B-spline is obtained by applying the transformation directly to the control points [4]. Let τ is a point in three-dimensional Euclidean space, $\mathbb{R}^3$, and using Φ to denote an affine transformation, mapping $\mathbb{R}^3$ into $\mathbb{R}^3$, ($\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$), for any $\tau \in \mathbb{R}^3$ [8], we can get that

$$\Phi(\tau) = \mathbf{A}\tau + \vec{v} \tag{2.22}$$

where $\mathbf{A}$ is a 3×3 matrix and $\mathbf{v}$ is a vector in $\mathbb{R}^3$, For an affine transformations, it mainly includes several algebraic representations such as shears, translations, rotations, scalings, and uniform stretchings etc. Based on partition of unity property of B-spline basis function, the affine invariance property of B-spline curve can be defined by

$$
\begin{aligned}
\Phi(\tau) &= \Phi(\sum \alpha_i B_i) = A(\sum \alpha_i B_i) + \vec{v} \\
&= \sum \alpha_i AB_i + \sum \alpha_i \vec{v} \\
&= \sum \alpha_i (AB_i + \vec{v}) \\
&= \sum \alpha_i \Phi(B_i)
\end{aligned}
\tag{2.23}
$$

where $B_i \in \mathbb{R}^3$ and $\sum \alpha_i = 0$; Similarly, according to the same logic as the Equation (2.23), the affine invariance property of the B-spline surface can also be defined. Let $\tau = \sum \alpha_i \sum \beta_j B_{i,j}$, where $B_{i,j} \in \mathbb{R}^3$, and $\sum \alpha_i \sum \beta_j = 1$.

$$
\begin{aligned}
\Phi(\tau) &= \Phi(\sum \alpha_i \sum \beta_j B_{ij}) = A(\sum \alpha_i \sum \beta_j B_{ij}) + \vec{v} \\
&= \sum \alpha_i \sum \beta_j AB_{ij} + \sum \alpha_i \sum \beta_j \vec{v} \\
&= \sum \alpha_i \sum \beta_j (AB_{ij} + \vec{v}) \\
&= \sum \alpha_i \sum \beta_j \Phi(B_{ij})
\end{aligned}
\tag{2.24}
$$

vi. Strong convex hull property; If we want to draw a B-spline curve, then it must be in the convex hull of its control polygon. Namely, if ξ is in knot span $[\xi_i, \xi_{i+1})$, then B-spline curve is in the convex hull formed by control points $B_{i-p}, B_{i-p+1} \cdots, B_i$. This is because this property follows from the non-negativity and partition of unity, and local compact support properties of B-spline basis function. More specifically, if ξ is in knot span $[\xi_i, \xi_{i+1})$, From the local support property we can know that, there are only $p+1$ non-zero basis functions i.e., $N_{i,p}(\xi), \cdots, N_{i-p+1,p}(\xi), N_{i-p,p}(\xi)$ on this knot span. Because the base functions is the coefficient of the corresponding control points, only $p+1$ control points i.e. $B_{i-p}, B_{i-p+1} \cdots, B_i$ have non-zero coefficients. In addition, since on this knot span, the B-spline basis functions are non-zero and their sum equals 1, the B-spline curve must lie in the convex hull formed from control points $B_{i-p}, B_{i-p+1} \cdots, B_i$. The meaning of "strong" is that while B-spline curve still lies in the convex hull formed from all control points, it lies in a much smaller one.

Here, we consider an example which is a more thoroughgoing review of the same example in Cottrell et al.[4]. Figure 2.18 shows such convex hulls for p = 1 through p = 5. they are defined by a given set of control points. It is important to note that t the convex hull for a piecewise linear curve is just the control polygon itself. Figure 2.19 shows the corresponding curves that we obtain by pairing these control points with the different basis. As the polynomial order increases, the curves become smoother and the effect of each individual control point is diminished.
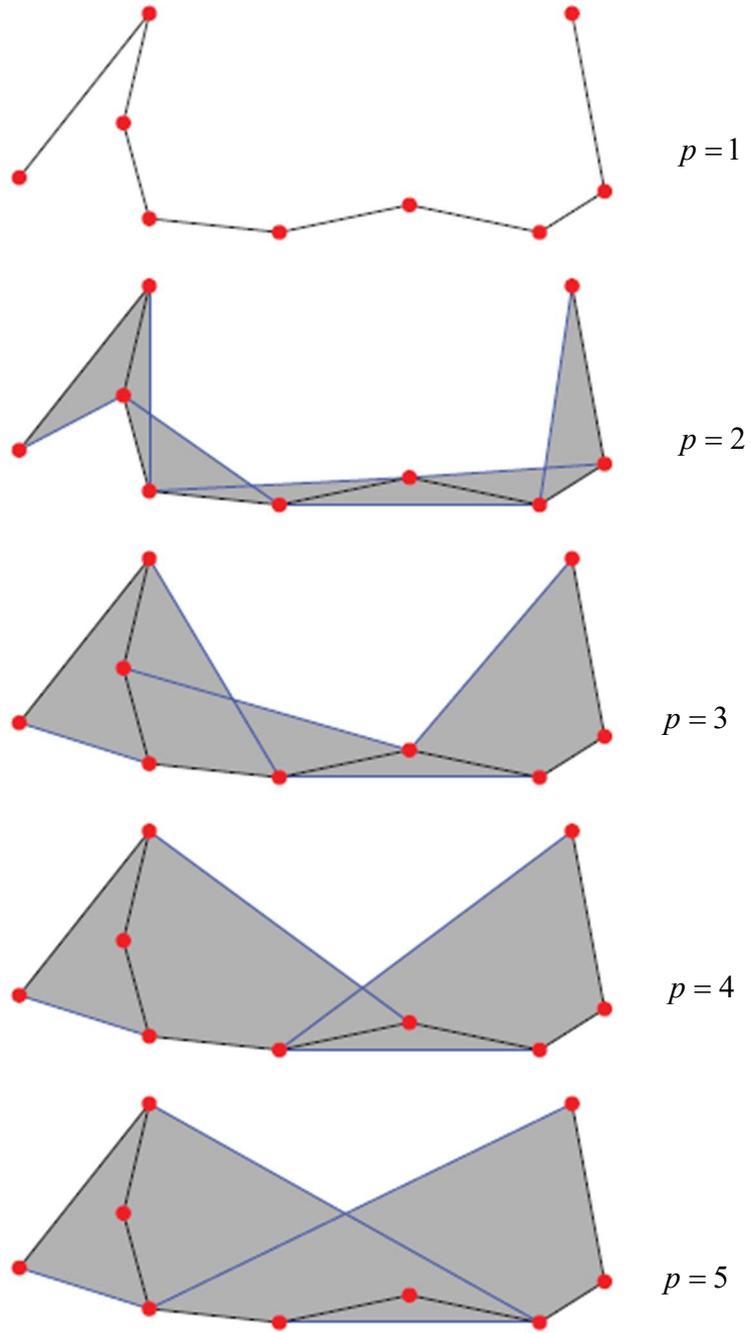
Figure 2.18 The convex hulls for $p = 1$ through $p = 5$
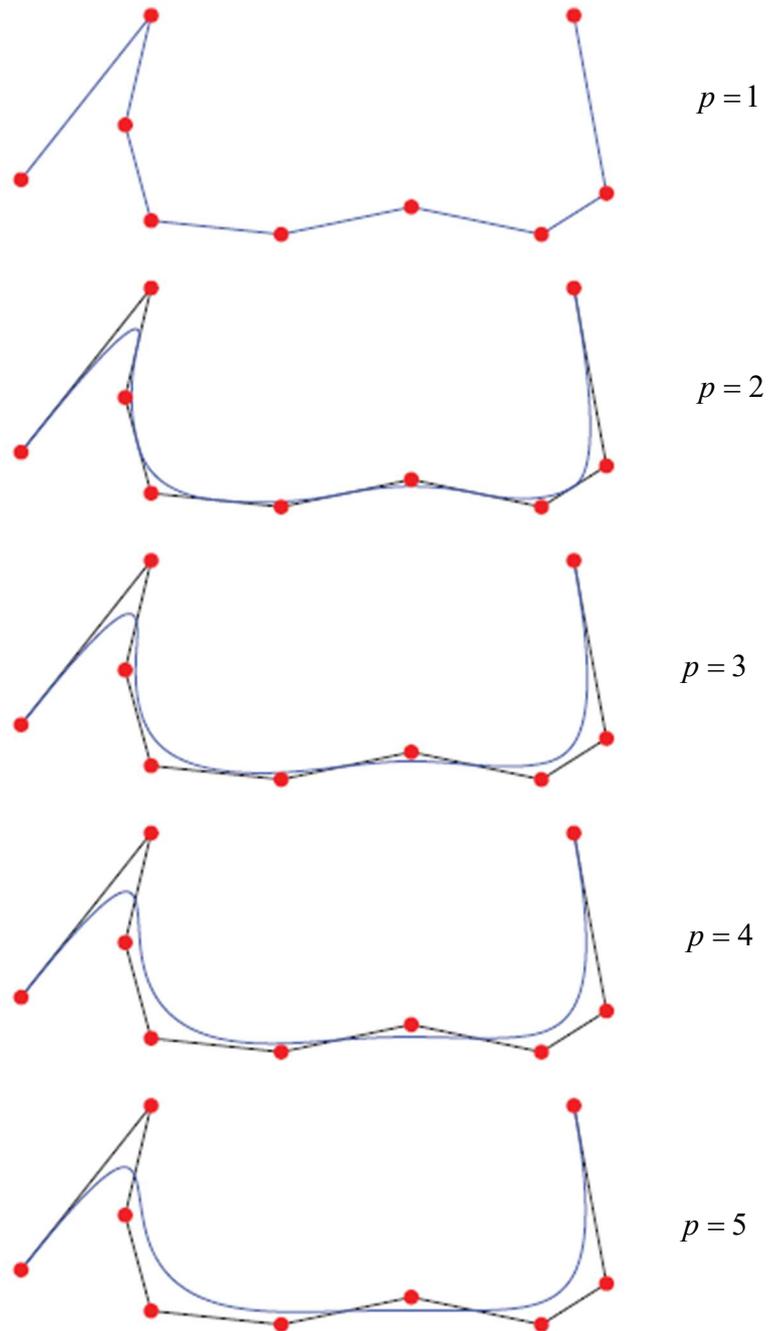
Figure 2.19 B-spline curves for $p = 1$ through $p = 5$

vii.   local modification scheme property: Due to the compact support of the B-splines basis functions, the basis function $N_{i,p}(\xi)$ is zero if ξ is outside of the interval $\left[\xi_i, \xi_{i+p+1}\right)$ in the one-dimensional topology case. The local modification scheme property of B-spline surface (two-dimensional topology) follows directly from the curve case. From the local modification scheme property, we can know that moving a single control point can affect the geometry of no more than $p+1$ elements of the curve and $(p+1)(q+1)$ elements of the surface (where p and q is the degree of basis function), respectively. The example of a surface is shown in Figure 2.20, In Figure 2.20(a), the initial surface is flat because all the control points lie in a common plane. The corresponding control mesh is offset from the surface for better visualization. If control point B marked in the red dot is moved to a new location, the Figure 2.20(b) shown that only the neighbouring area on the surface of the moved control point changes shapes and elsewhere is unchanged.
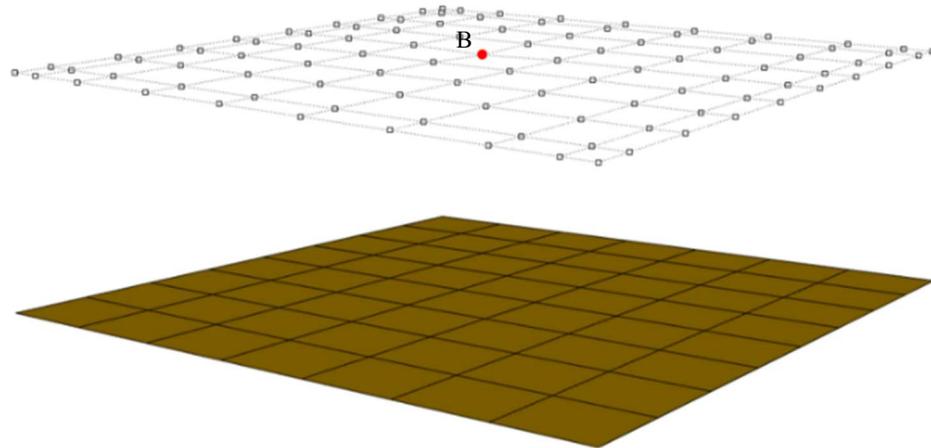
(a) A planar cubic × quadratic surface, $\Xi = \{0,0,0,0,1/7,2/7,3/7,4/7,5/7, 6/7,1,1,1,1\}$ and $\Pi = \{0,0,0,1/8,2/8,3/8,4/8,5/8,6/8,7/8,1,1,1\}$;
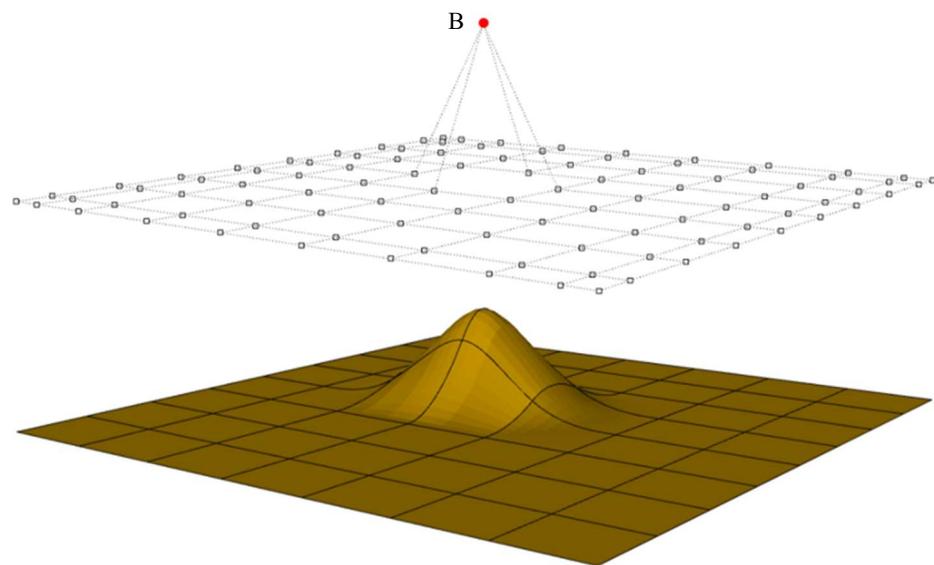
(b) The control point B (denoted by a red dot) is moved, influencing only local shape on the surface.

Figure 2.20. the B-splines surfaces.

2.6. Non-Uniform Rational B-Spline (NURBS)

In this section, we combine the concepts of the B-spline introduced in the previous sections to obtain non-uniform rational B-spline curves and surfaces. The NURBS is a more flexible modelling approach based on the B-spline concept, but without the drawbacks of B-spline; namely, NURBS allows exact the representation of geometrical shapes with conic sections, such as cylinders, ellipsoids, etc. However, it is impossible to use the B-spline to represent these simple shapes exactly.

2.6.1 NURBS curve

Firstly, a NURBS curve can be defined by

$$C(\xi) = \sum_{i}^{n} N_{i,p} \mathbf{B}_i^w \tag{2.25}$$

where $N_{i,p}(\xi)$ represents the pth-degree piecewise B-spline basis functions defined on the open knot vector $\Xi = \{\xi_0, \xi_1 \cdots, \xi_m\}$; $\mathbf{B}_i^w$ is not a geometric coordinate in the traditional sense. It is a homogeneous coordinate with weight w_i, that is to say, the geometric coordinates are weighted

$$\mathbf{B}_i^w = \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix}. \tag{2.26}$$

Note that If $w_i = 1$, then it is the Cartesian coordinates, namely, fourth component in homogeneous coordinate is rewritten as 1

$$\mathbf{B}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{2.27}$$

We expand the homogeneous coordinate with weight in Equation 2.26 and bring it into the Equation 2.25. We can get

$$\mathbf{C}^w(\xi) = \sum_{i=0}^{n} N_{i,p}(\xi)\mathbf{B}_i^w = \sum_{i=0}^{n} N_{i,p}(\xi) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} N_{i,p}(\xi)\left(w_i x_i\right) \\ \sum_{i=0}^{n} N_{i,p}(\xi)\left(w_i y_i\right) \\ \sum_{i=0}^{n} N_{i,p}(\xi)\left(w_i z_i\right) \\ \sum_{i=0}^{n} N_{i,p}(\xi)w_i \end{bmatrix}, \quad (2.28)$$

Thus, the point $\mathbf{C}^w(\xi)$ is the original B-spline curve in form of homogeneous coordinate. Now, let us convert it back to Cartesian coordinate by dividing $\mathbf{C}^w(\xi)$ with the fourth coordinate in homogeneous coordinate:

$$\mathbf{C}(\xi) = \begin{bmatrix} \dfrac{\sum_{i=0}^{n} N_{i,p}(\xi)\left(w_i x_i\right)}{\sum_{i=0}^{n} N_{i,p}(\xi)w_i} \\ \dfrac{\sum_{i=0}^{n} N_{i,p}(\xi)\left(w_i y_i\right)}{\sum_{i=0}^{n} N_{i,p}(\xi)w_i} \\ \dfrac{\sum_{i=0}^{n} N_{i,p}(\xi)\left(w_i z_i\right)}{\sum_{i=0}^{n} N_{i,p}(\xi)w_i} \\ 1 \end{bmatrix} = \sum_{i=0}^{n} N_{i,p}(\xi)\mathbf{B}_i^w = \sum_{i=0}^{n} \dfrac{N_{i,p}(\xi)w_i}{\sum_{\hat{i}=0}^{n} N_{\hat{i},p}(\xi)w_{\hat{i}}} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.29)$$

Finally, the NURBS curve in Cartesian coordinates is expressed as

$$C(\xi) = \sum_{i=1}^{n} \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^{n} N_{i,p}(\xi)w_i} \mathbf{B}_i \quad (2.30)$$

Based on Equation 2.30, the NURBS basis function is defined as follows:

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)w_{\hat{i}}} \tag{2.31}$$

where $R_{i,P}(\xi)$ represents the piecewise rational functions on the open knot vector.

From a geometric point of view, the weight w_i is applied to an affine transformation of the B-spline curves between the high-dimensional and low-dimensional spaces. So the so-called NURBS entity in $\mathbb{R}^d$ is actually a projective transformation of B-spline entity in $\mathbb{R}^{d+1}$ [4] ,where d is the number of geometric dimensions in physical space. Figure 2.21 illustrates how a NURBS curve $\mathbf{C}^w(\xi)$ in $\mathbb{R}^2$ is built by the projective transformation of a quadratic B-spline curve $\mathbf{C}(\xi)$ in $\mathbb{R}^3$.
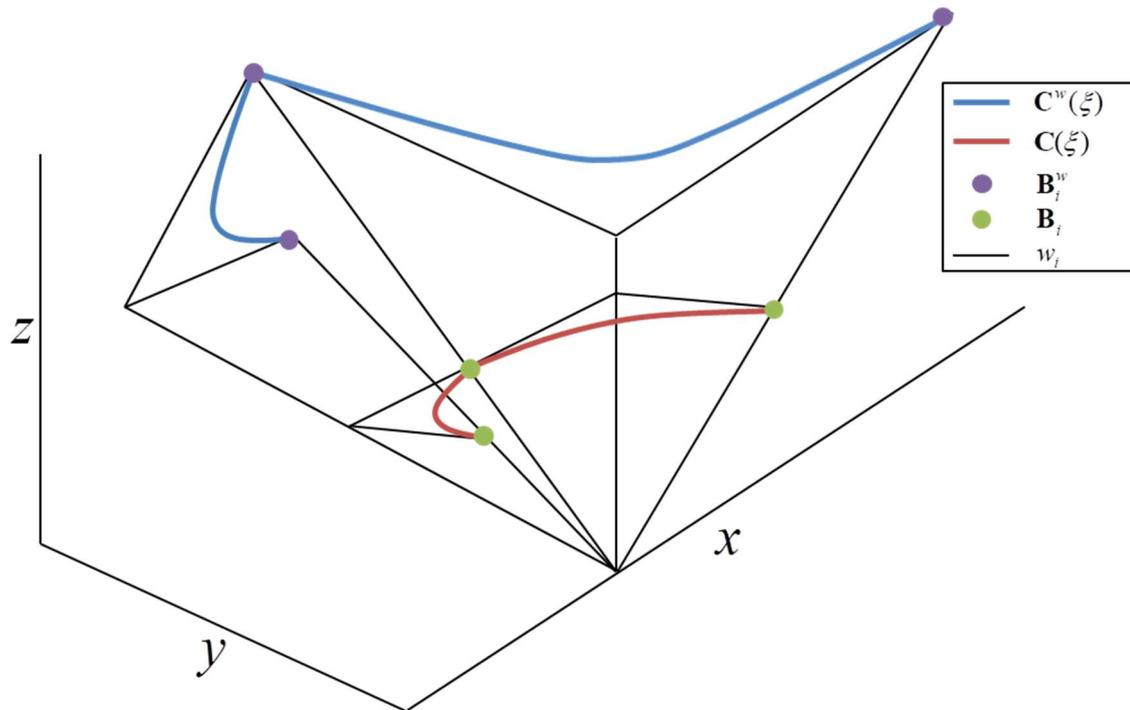
Figure 2.21 A semicircle $\mathbf{C}(\xi)$ in $\mathbb{R}^2$ constructed by the projective transformation of a piecewise quadratic B-spline in $\mathbb{R}^3$; the projective transformation of "projective control point" $\mathbf{B}_i^w$ yields control point $\mathbf{B}_i$; Weight w_i is the z-component of $\mathbf{B}_i^w$; Projective transformation of the B-spline curve $\mathbf{C}^w(\xi)$ yields the NURBS curve $\mathbf{C}(\xi)$.

Through the Equation 2.26 and Equation 2.27, we can know that, the relationship between control point $\mathbf{B}_i^w$ in homogeneous coordinate and $\mathbf{B}_i$ in Cartesian coordinate is as follows.

$$(\mathbf{B}_i)_j = \frac{(\mathbf{B}_i^w)_j}{w_i} \tag{2.32}$$

with

$$w_i = (\mathbf{B}_i^w)_{d+1}$$

where $j = 1,\dots,d$. If we would like to apply the same projective transformation to every point in the NURBS curve $\mathbf{C}^w(\xi)$, we can define the following weighting function to accomplish this one.

$$W(\xi) = \sum_i^n N_{i,p}(\xi) w_i \tag{2.33}$$

Therefor, The NURBS basis function can also be expressed as

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}} = \frac{N_{i,p}(\xi) w_i}{W(\xi)} \tag{2.34}$$

Finally, the NURBS can be rewritten as a clean form as follows:

$$\mathbf{C}(\xi) = \sum_i^n R_{i,p}\mathbf{B}_i \tag{2.35}$$

Correspondingly, the derivatives of the NURBS basis are given by

$$\frac{dR_{i,p}(\xi)}{d\xi} = w_i \frac{W(\xi)N'_{i,p}(\xi) - W'(\xi)N_{i,p}(\xi)}{(W(\xi))^2} \tag{2.36}$$

where

$$W'(\xi) = \sum_{\hat{i}}^n N'_{i,p}(\xi) w_{\hat{i}} \tag{2.37}$$

For Higher-order derivatives of the NURBS basis functions, it may be expressed in terms of lower-order derivatives as

$$\frac{d^k}{d\xi^k} R_i^p(\xi) = \frac{A_i^{(k)}(\xi) - \sum_{j=1}^{k} \binom{k}{j} W^{(j)}(\xi) \frac{d^{(k-j)}}{d\xi^{(k-j)}} R_i^p(\xi)}{W(\xi)},$$ (2.38)

where

$$A_i^{(k)}(\xi) = w_i \frac{d^k}{d\xi^k} N_{i,p}(\xi), \quad (\textit{no sum on i})$$ (2.39)

and

$$W^{(k)}(\xi) = \frac{d^k}{d\xi^k} W(\xi).$$ (2.40)

Further

$$\binom{k}{j} = \frac{k!}{j!(k-j)!}.$$ (2.41)

2.6.2 NURBS surface

Similarly, A NURBS surface with p^{th}-degree in the ξ-direction and q^{th}-degree in the η-direction is defined by the rational basis functions and corresponding control points, as follows[9]

$$S(\xi, \eta) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{\sum_{\hat{i}=1}^{n} \sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) w_{i,j}} \mathbf{B}_{i,j},$$ (2.42)

where $\{\mathbf{B}_{i,j}\}$ is a matrix form that construct a bidirectional control net, and $\{w_{i,j}\}$ are weights. (i, j) are the set of all double-indices of NURBS basis functions, that is denoted by

$$\mathbb{R} = \left\{ (i, j) : i \in \{1, \ldots, n\}, \, j \in \{1, \ldots, m\} \right\}. \tag{2.43}$$

$\left\{ N_{i,p}(\xi) \right\}$ and $\left\{ M_{j,q}(\eta) \right\}$ is the standard B-spline basis functions defined on the open knot vector Ξ and H, respectively.

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$$
$$H = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$$

Therefore, we can introduce a form of bivariate vector-valued piecewise rational basis functions

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{\sum\limits_{\hat{i}=1}^{n} \sum\limits_{j=1}^{m} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) w_{i,j}} \tag{2.44}$$

Accordingly, Equation (2.40) may be written in brevity of notations

$$S(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,p}(\xi) R_{j,q}(\eta) \mathbf{B}_{i,j} = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}(\xi\eta) \mathbf{B}_{i,j} \tag{2.45}$$

2.7. Multiple patches

2.7.1. Characteristic of multiple patch

A single geometric domain on a geometric model is referred to as the *patch*, and a geometric model composed of multiple geometric domains is referred to as *multi-patch*. In fact, in the real work, when engineers use CAD tools to design the geometry, they are usually divided into multiple patch to draw. This is because, it is difficult to use single NURBS to model the model with complex boundary or internal holes. Even if the model is successful, the physical space units generated in the process of analysis are often distorted and uneven. In addition, If different materials or physical models are used in different parts of the domain, the description of these subdomains can be simplified by different patches[4]. Moreover, if different sub-domains are to be assembled in parallel on multiprocessor machines, it is very convenient from the point of view of data structure that there is no need to split individual patches between different processors. The most common case is that the domain is completely different in topology from the cube. The tensor product structure of a patch in parameter space makes it unsuitable for representing complex, multi-joined domains. This geometry can usually be handled very simply by using multiple patches. An example of a plate with circular hole consisting of multiple patches is shown in Figure 2.22, where geometric model is built from the four independent patches. From Figure 2.22 (b), we can see that the plate with circular hole is exactly and concisely represented by four simple NURBS patches which are marked with different colours. The meshes using multiple patches and single patch are shown in Figure 2.23 (a) and (b), respectively. Compared with single patch, the geometric structure represented by multi-patch exhibits far less distortion and yields a much more "natural" mesh.
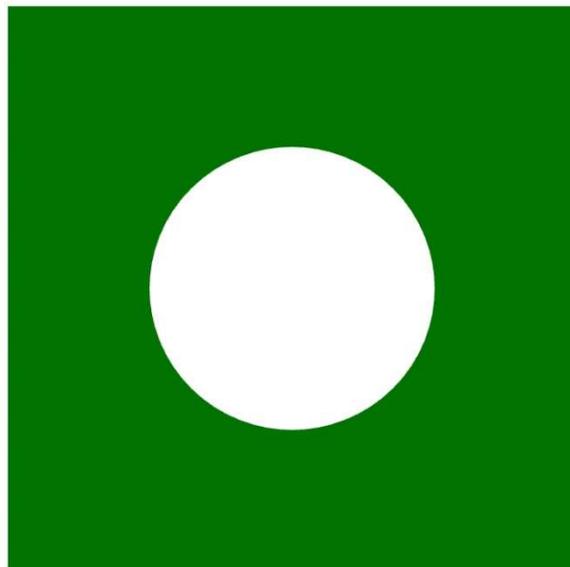
(a) The geometric model represented by a single NURBS patch
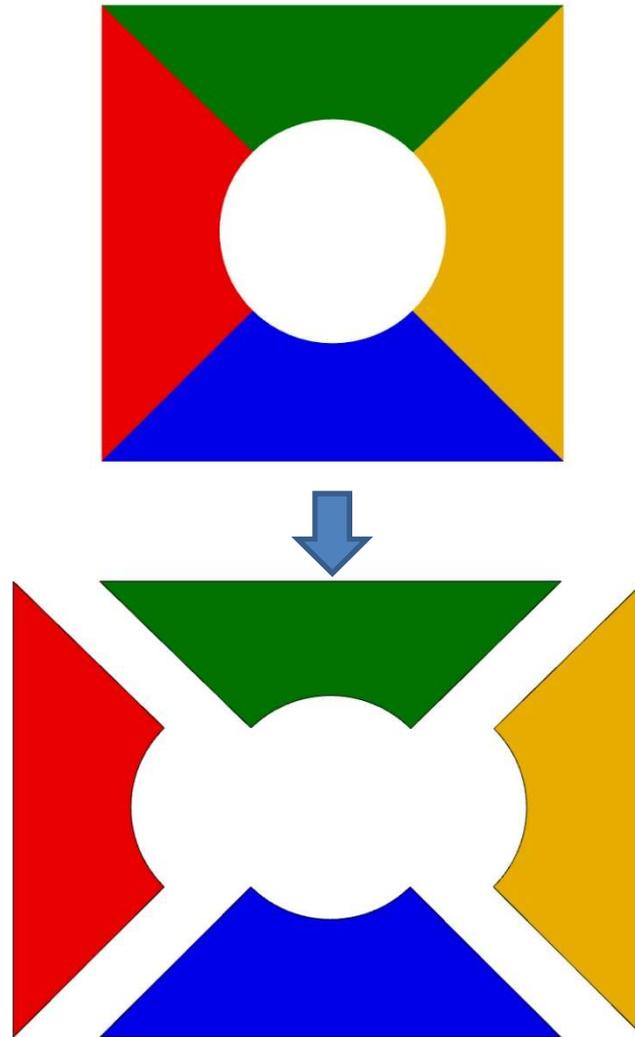
(b) the plate with circular hole is exactly and concisely represented by four simple NURBS patches

Figure 2.22 A plate with circular hole is represented by a single NURBS patch and multiple NURBS patches, respectively.
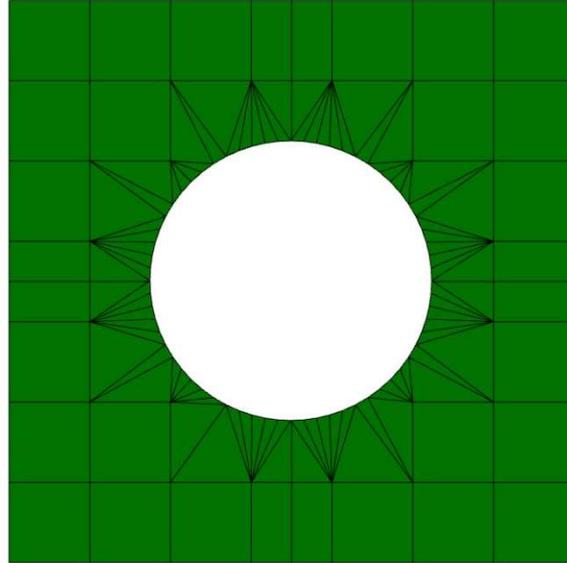
(a) The physical mesh for the single patch
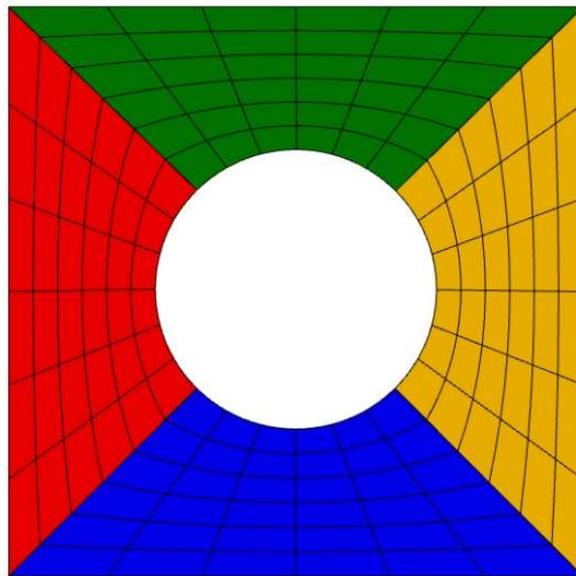
(b) The physical network is defined on four patches

Figure 2.23 The physical mesh of the plate with circular hole

Next we describe how to create multiple patch, As a prerequisite, it is assumed that the reference patch domain $S^{base}(\xi,\eta)$ is connected to patch areas of different nature $S^{add}(\xi,\eta)$. At this time, although an additional patch domain $S^{add}(\xi,\eta)$ and based patch domain $S^{base}(\xi,\eta)$ have different properties, but it is assumed as continuous. And suppose the other side is connected to a patch area of the same nature. Here, let us give a concrete example to illustrate. As shown in the Figure 2.24, we connect the additional patch area to the $Side1$ of the based patch area. Thus, a new multiple patch area $S^{com}(\xi,\eta)$ can be obtained. When the $Side3$ of the based patch area is represented by function $\psi_0(u)$, the $S^{com}(\xi,\eta)$ can be defined as following.

$$
\begin{aligned}
S^{com}(\xi,\eta) = {} & S^{base}(\xi,\eta) \\
& -\{\psi_0 + (1-\eta)S^{base}(\xi,0)\} \\
& +\{\psi_0 + (1-\eta)S^{add}(\xi,1)\},
\end{aligned}
\tag{2.46}
$$

where, $\{\psi_0 + (1-\eta)S^{base}(\xi,0)\}$ is the form of subtracting $Side1$ from the base patch domain. $\{\psi_0 + (1-\eta)S^{add}(\xi,1)\}$ is the form of adding A to the additional patch area from the based patch domain.
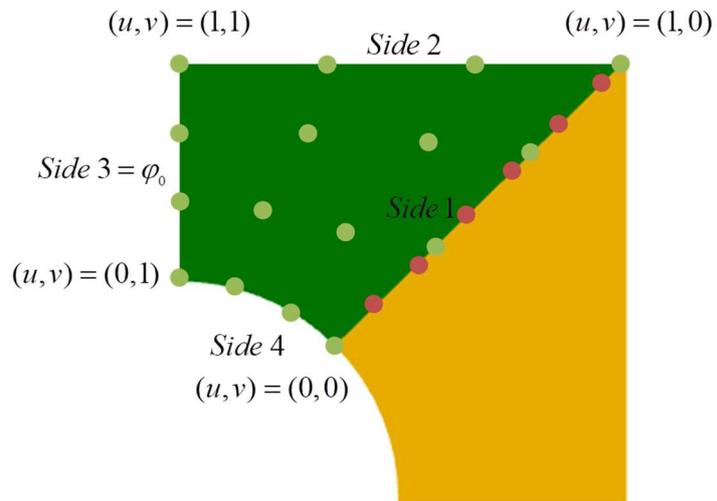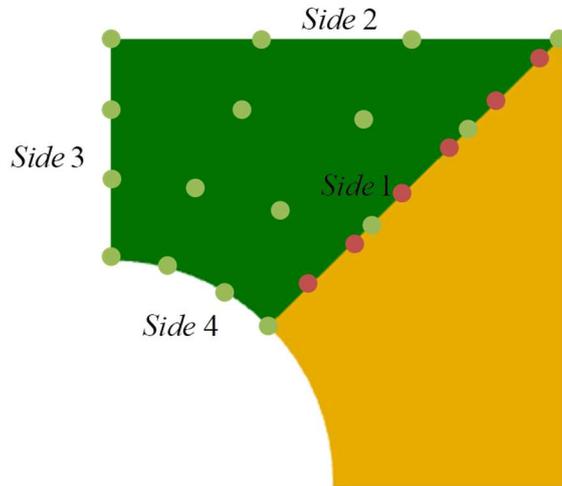
Figure 2.24 The multiple patch domain

According to figure 2.24, we can see that the green area and the pink area have different properties, so it has different control point position. Therefore, in order to unify the control points, we need a unified nature of the patch domain s. Thus, we can obtain following equation.

$$S^{com}(\xi,\eta) = S^{base}(\xi,\eta) \\ -(1-\eta)S^{base}(\xi,0) + (1-\eta)S^{add}(\xi,1). \tag{2.47}$$

And, if Equation (2.47) is expressed in the form of B spline basis function, we have following form.

$$\mathbf{S}^{com}(\xi,\eta) = \sum_{A \in \varphi} R_A(\xi,\eta)\mathbf{B}_A \\ -(1-\eta)\sum_{A \in \varphi} R_A(\xi,0)\mathbf{B}_A + (1-\eta)\sum_{A \in \varphi^{add}} R_A^{add}(\xi,1)\mathbf{B}_A, \tag{2.48}$$

where, $R_A(\xi,\eta)$ is defined as NURBS basis function in the based patch domain, φ is a set of control points constituting the based patch domain, $R_A^{add}(\xi,\eta)$ is defined as NURBS basis function in the additional patch domain, φ^{add} is a set of control points constituting the additional patch domain. And we deform Equation (2.48) as

$$\mathbf{S}^{com}(\xi,\eta) = \sum_{A \in \varphi} R_A^{com1}(\xi,\eta)\mathbf{B}_A + \sum_{A \in \varphi^{add}} R_A^{com2}(\xi,\eta)\mathbf{B}_A, \tag{2.49}$$

where,

$$\sum_{A \in \varphi} R_A^{com1}(\xi,\eta)\mathbf{B}_A = \sum_{A \in \varphi}\left\{R_A(\xi,\eta)-(1-\eta)R_A(\xi,0)\right\}\mathbf{B}_A \tag{2.50}$$

$$\sum_{A \in \varphi^{add}} R_A^{com2}(\xi,\eta)\mathbf{B}_A = \sum_{A \in \varphi^{add}}\left\{(1-\eta)R_A^{add}(\xi,1)\right\}\mathbf{B}_A \tag{2.51}$$

When, $\varphi^{com} = \varphi + \varphi^{add}$ is considered, the Equation (2.49) can ultimately be expressed as following by product of the shape functions and the control points.

$$\mathbf{S}^{com}(\xi,\eta) = \sum_{A \in \varphi^{com}} R_A^{com}(\xi,\eta)\mathbf{B}_A$$

$$= \sum_{A \in \varphi} R_A^{com1}(\xi,\eta)\mathbf{B}_A + \sum_{A \in \varphi^{add}} R_A^{com2}(\xi,\eta)\mathbf{B}_A, \tag{2.52}$$

where, $R_A^{com}(\xi,\eta)$ is the shape function of the multiple patch domain, φ^{com} is a set of control points constituting the multiple patch domain. For $R_A^{com}(\xi,\eta)$, it is defined as following.

$$R_A^{com}(\xi,\eta)\big|_{A \in \varphi} = R_A^{com1}(\xi,\eta),$$
$$R_A^{com}(\xi,\eta)\big|_{A \in \varphi^{add}} = R_A^{com2}(\xi,\eta). \tag{2.53}$$

And, in the next section, we will give formulas of connected to additional patch domain the in the side of $Side2, Side3, Side4$.

2.7.2 Definition formula for multiple patch domain

The multiple patch domain is defined as following form.

$$\mathbf{S}^{com}(\xi,\eta) = \sum_{A \in \varphi^{com}} R_A^{com}(\xi,\eta)\mathbf{B}_A, \tag{2.54}$$

where, $R_A^{com}(\xi,\eta)$ is defined as following.

$$R_A^{com}(\xi,\eta)\big|_{A \in \varphi} = R_A^{com1}(\xi,\eta),$$
$$R_A^{com}(\xi,\eta)\big|_{A \in \varphi^{add}} = R_A^{com2}(\xi,\eta). \tag{2.55}$$

For $R_A^{com1}(\xi,\eta), R_A^{com2}(\xi,\eta)$, it is dependent on the additional patch area connected to an edge of the based patch area. Thus, we have following four situations.

In the case of connection the additional patch domain in $Side1$,

$$R_A^{com1}(\xi,\eta) = R_A(\xi,\eta) - (1-\eta)R_A(\xi,0),$$
$$R_A^{com2}(\xi,\eta) = (1-\eta)R_A^{add}(\xi,1). \tag{2.56}$$

In the case of connection the additional patch domain in *Side*2 ,

$$R_A^{com1}(\xi,\eta) = R_A(\xi,\eta) - \xi R_A(1,\eta),$$
$$R_A^{com2}(\xi,\eta) = \xi R_A^{add}(0,\eta).$$

(2.57)

In the case of connection the additional patch domain in *Side*3 ,

$$R_A^{com1}(\xi,\eta) = R_A(\xi,\eta) - \eta R_A(\xi,1),$$
$$R_A^{com2}(\xi,\eta) = \eta R_A^{add}(\xi,0).$$

(2.58)

In the case of connection the additional patch domain in *Side*4 ,

$$R_A^{com1}(\xi,\eta) = R_A(\xi,\eta) - (1-\xi)R_A(0,\eta),$$
$$R_A^{com2}(\xi,\eta) = (1-\xi)R_A^{add}(1,\eta).$$

(2.59)

2.8 Conclusions

In this section, the isogeometric analysis method is introduced in detail. Some important notions which is a key to understanding IGA are described and illustrated in detail, including the physical space, parameter space, and knot vector etc. In addition, some main paraphernalias employed in isogeometric analysis be given, for example, NURBS, B-spline, and basis function etc. Finally, we also focused on some of the properties of B-spline, which played a crucial role in the development of later algorithms. NURBS, as an extension of B-spline, fully inherits these features. In order to better understand how to use NURBS, we understand them geometrically and algebraically. The former perspective provides us with insight and intuition that are invaluable in designing grids, proving theorems, and many other activities related to isogeometric analysis. The latter view is especially useful when designing algorithms and creating software, and will be the setting we use most often. Both are critical to developing a broad understanding of NURBS technology.

Reference

[1] P. Kagan, A. Fischer, and P. Z. Bar-Yoseph, 'New B-Spline Finite Element approach for geometrical design and mechanical analysis', *Int. J. Numer. Methods Eng.*, vol. 41, no. 3, pp. 435–458, 1998.

[2] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, 'Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement', *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 39–41, pp. 4135–4195, 2005.

[3] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo, 'Parametrization of computational domain in isogeometric analysis: methods and comparison', p. 25.

[4] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA, chapter 2: NURBS as a Pre-analysisTool: Geometric Design and Mesh Generation*. John Wiley & Sons, 2009.

[5] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA, chapter 1: From CAD and FEA to Isogeometric Analysis: An Historical Perspective*. John Wiley & Sons, 2009.

[6] C. de Boor, 'On calculating with B-splines', *J. Approx. Theory*, vol. 6, no. 1, pp. 50–62, Jul. 1972.

[7] M. G. Cox, *The Numerical Evaluation of B-splines*. National Physical Laboratory, Division of Numerical Analysis and Computing, 1971.

[8] L. Piegl and W. Tiller, *The NURBS Book; Chapter Three B-Spline Curves and Surfaces*. Springer Science & Business Media, 1996.

[9] L. Piegl and W. Tiller, *The NURBS Book; Chapter Four Rational B-Spline Curves and Surfaces*. Springer Science & Business Media, 1996.

3. UNCERTAINTY ANALYSIS

3.1. Concepts of Uncertainty Analysis

Stochastic analysis (Uncertainty Analysis) is also known as non-deterministic analysis, where the relationship between variables is given as a statistical value. In the real world, uncertainty is ubiquitous. For example, tiny particles floating on the surface of a liquid move continuously and disorderly, and the position of the particles at any time is uncertain; Another example is the shape and size of the electronic components or structure (Figure 3.1), which are also uncertain due to solder deformation and corrosion. Such uncertain phenomena cannot be grasped on the surface. In fact, behind certain uncertainties, there is often a certain probability law hidden. Therefore, the stochastic analysis model based on probability and mathematical statistics become a solution to such problems. It is one of the most effective tools. Here, we give a concept map (Figure 3.2) of the uncertainty analysis. When the input is a statistical distribution, the response will also obtain a statistical distribution related to the input statistical distribution through a series of analytical

methods. That is to say, from a mathematical point of view, the uncertainty characteristic of the response $u(\zeta_1, \zeta_2)$ is a function of the uncertainty characteristics of the input parameters. The response is based on the uncertainty characteristics of the input parameters in the space and the time.
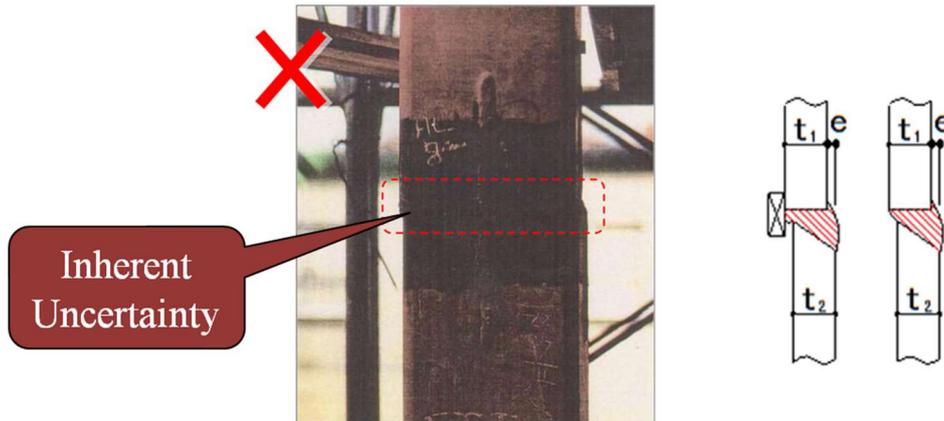
(a) Geometrical shape (Randomness in Manufacturing (the pictures cited from Reference [1])
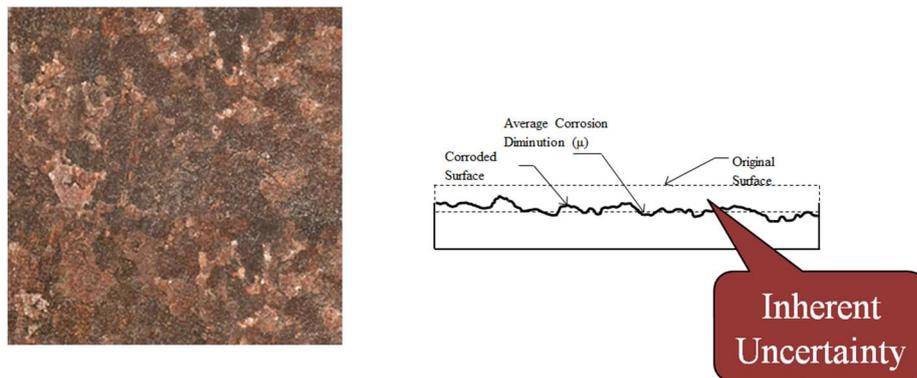
(b) Corrosion (Wastage after Years of Exposure in the Operation) (the pictures cited from Reference [2])

Figure 3.1 Image of Inherent Uncertainty
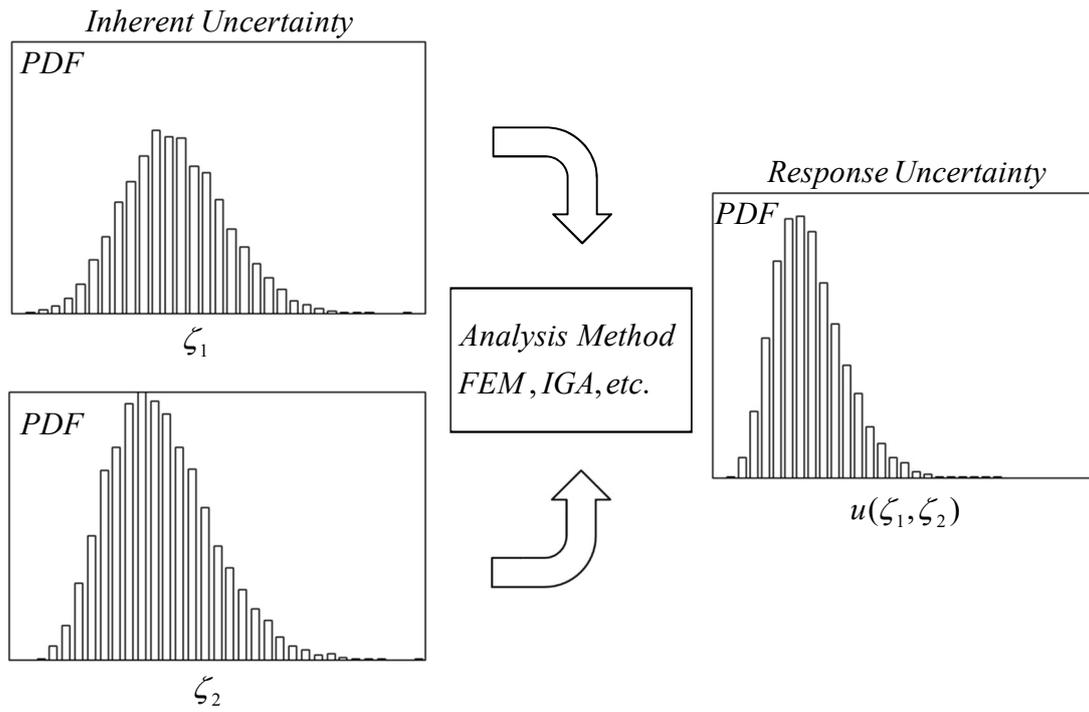
Figure 3.2 Concept Map of Uncertainty Analysis

3.2 Type of Uncertainty Analysis

As a mainstream uncertainty analysis method, it is mainly divided into two large parts, one is "the non-intrusion method", and the other is "the intrusive method". And they all introduce probability variables into an analysis method, such as finite element methods, boundary element methods, and so on. The main difference is whether to modify the analysis method itself. The so-called non-intrusive method, as literally stated, simply introduces the uncertainty parameter into the analysis method and does not make any modifications to the analysis method. The intrusion method is to express the random variable in some form, introduce to the analysis method, and re-edit and normalize the analysis method. It essentially changes the analytical method and forms a new uncertainty analysis method with random variables.
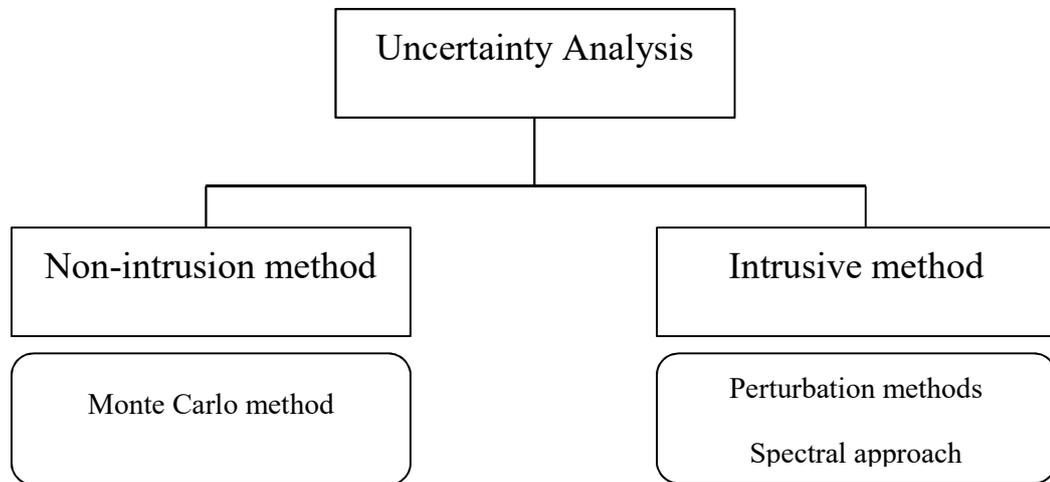
Figure 3.3 Intrusive and Non-Intrusive Formulations

3.2.1. The non-intrusion method

As a representative method of non-invasive methods, Monte Carlo method has been applied to uncertainty analysis to date. Here we give a Concept Map of the Monte Carlo method. When we assume that the input random variable follows a certain distribution (the normal distribution, the logarithmic distribution, etc.), then this distribution can be represented by *m* random samples. When we introduce the *m* samples into the analysis method, the *m* responses can be obtained, and the m responses will also follow a certain distribution to some extent. Thus, when an input parameter is considered to contain a random variable, we can obtain a random response by calculating multiple samples.
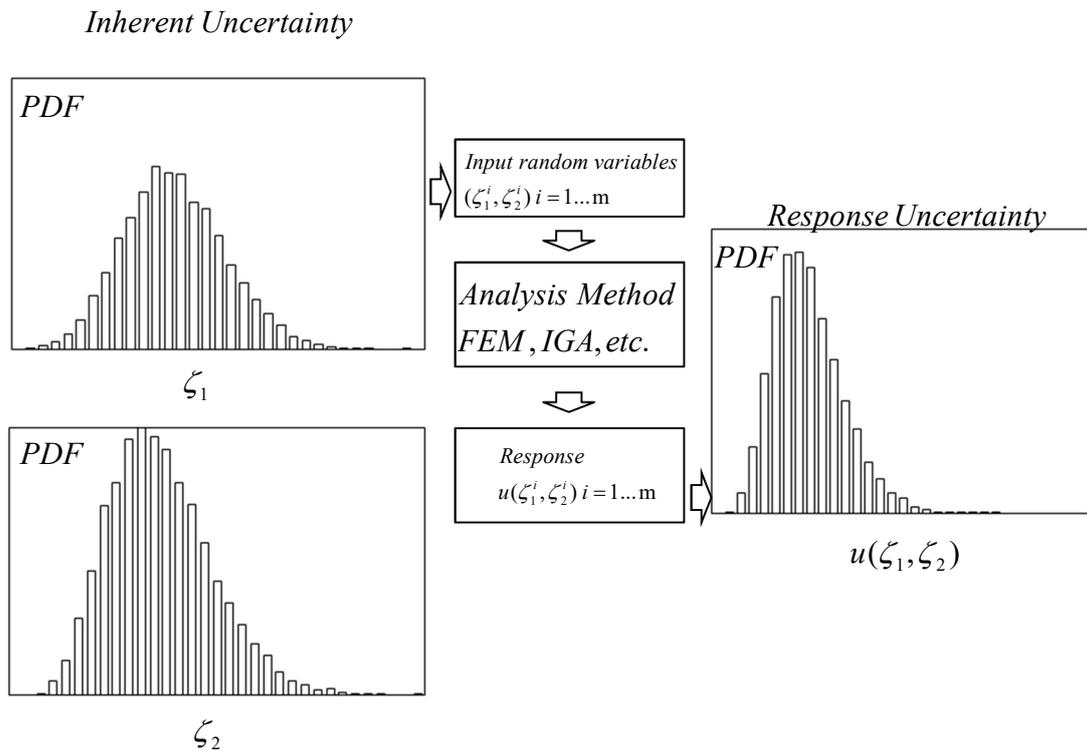
Figure 3.4 Concept Map of Monte Carlo methods

3.2.2. The intrusion method

About the intrusion method, the perturbation method was first developed, but the perturbation method can only indicate random variation within a certain small range due to the limitation of the method. Therefore, the spectral approach has been widely developed in recent years.

3.2.2.1. The perturbation method

The perturbation method uses Taylor series expansions to introduce randomness into analysis equation. Here we take the finite element method as an example to illustrate. When stiffness is considered to contain uncertainty, it can be expressed as the sum of an expected value and a variable, as shown below.

$$K = K_0 + \sum_{i=1}^{N} K_i^{\mathrm{I}} \zeta_i \,, \tag{3.1}$$

where, the K_0 is the expected value, K_i^{I} is partial derivatives, ζ_i is random variables.

3.2.2.1. Response surface method and Spectral approach

The spectral method has been introduced by Ghanem and Spanos[3] as an extension of the deterministic finite element method. Uncertainty of the Young modulus of a structure is described by using the Karhunen–Loève (K–L) expansion. However, K-L expansion can only deal with the Gaussian stochastic processes. When the inherent uncertainty of a structure is following non-Gaussian stochastic processes, K-L expansion method has become infeasible. The polynomial chaos expansion (PCE) method can represent not only a Gaussian distribution but also a non-Gaussian distribution. Sakamoto and Ghanem[4] proposed a new method to express inherent uncertainty of non-Gaussian stochastic processes using PCE method. PCE is the primary method used in this study to represent the internal uncertainty of the structure, which will be described in detail in the next chapter.

Since the intrusion method changes the analysis method internally, it often only needs to perform one analysis to get an uncertainty response. Compared to the Monte Carlo method, the calculation time and cost are greatly saved.

3.3. Polynomial Chaos Expansion (PCE)

3.3.1 Introduction

In uncertainty analysis, the stochastic expansion method is a significant alternative approach for representing uncertain parameters[5]. The purpose of stochastic expansion is to better describe the uncertainty of the system by introducing a series of polynomials characterizing the characteristics of the stochastic system.

In this study, the PCE was employed in order to represent inherent uncertainty in structure model. PCE approach uses a random space composed of polynomial bases to describe the uncertainty of system with PDF form. The basic idea is to approximately represent uncertainty by using the sum of the orthogonal polynomial chaos containing independent random variables, and the key step is to determine the coefficients of each polynomial. The PCE with multiple random variables for a Gaussian random response is defined as follows[6]:

$$
\begin{aligned}
u(\zeta) = a_0 \Psi_0 &+ \sum_{i_1=1}^{\infty} a_{i_1} \Psi_1\left(\vartheta_{i_1}(\zeta)\right) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2} \Psi_2\left(\vartheta_{i_1}(\zeta), \vartheta_{i_2}(\zeta)\right) \\
&+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} \Psi_3\left(\vartheta_{i_1}(\zeta), \vartheta_{i_2}(\zeta), \vartheta_{i_3}(\zeta)\right) + \cdots ,
\end{aligned}
\tag{3.2}
$$

where $u(\zeta)$ is a random process that can express the input parameter and can approximate the response; $a_{i_1},\cdots,a_{i_p}$ are polynomial coefficients; $\vartheta_{i_1}(\zeta),\ldots,\vartheta_{i_p}(\zeta)$ is a set of random variables in the sample space; $\Psi_p\left(\vartheta_{i_1}(\zeta),\ldots,\vartheta_{i_p}(\zeta)\right)$ is a set of orthogonal polynomials, which are functions of the random variable $\vartheta_{i_p}(\zeta)$.

Equation (3.2) can be reduced to:

$$
u(\zeta) = \sum_{i=0}^{q} a_i \Psi_i\left(\vec{\vartheta}(\zeta)\right) ,
\tag{3.3}
$$

where $\vec{\vartheta}(\zeta)$ is the random vector built from random variables. q is the number of terms of polynomial chaos, and is determined by the following formula:

$$q = \frac{(p+n)!}{p!\,n!} - 1 \quad, \tag{3.4}$$

where p is the maximal order of polynomial chaos, and n is the number of random variables. According to the Askey-scheme[7], there are different optimal polynomials for different probability density functions, i.e., the choice of polynomial depends on the probability density function of the random variable. In this study, the stochastic response $u(\zeta)$ was approximated by using the Hermite polynomials as optimal polynomials, because uncertainty in shape was hypothetically defined as following the normal random distribution. The multi-dimensional Hermite polynomials are expressed as follows:

$$\Psi_p\left(\zeta_{i_1}, \cdots, \zeta_{i_p}\right) = e^{-\frac{1}{2}\vec{\zeta}^T\vec{\zeta}} (-1)^p \frac{\partial^p}{\partial\zeta_{i_1}\cdots\partial\zeta_{i_p}} e^{\frac{1}{2}\vec{\zeta}^T\vec{\zeta}} \quad. \tag{3.5}$$

Note that the basis-functions constituting PCE are all orthogonal polynomials. Therefore, we used the orthogonal property of polynomials to deal with stiffness equations with random variables in this study. The orthogonal property of the Hermite polynomials is defined by the inner product of the weight functions as follows:

$$\left\langle \Psi_m(\zeta), \Psi_k(\zeta) \right\rangle = \int_D \Psi_m(\zeta)\Psi_k(\zeta)W(\zeta)d\zeta = k!\delta_{km} \, , \tag{3.6}$$

where δ_{km} is the Kronecker delta and D is the domain of the standard normal probabilistic variable, and $W(\zeta)$ is the weight function. Generally, the weight function is the same as the probability density function, in order to ensure that the PCE converges exponentially to a random variable. For the random variable whose probability density function is a Gauss function, the weight function is expressed as follows:

$$W(\zeta) = \frac{1}{\sqrt{2\pi}} e^{\frac{-\zeta^2}{2}} \ ,$$ (3.7)

and $\Psi_m(\zeta)$ is derived by the following characteristic:

$$\left\langle \Psi_m(\zeta) \right\rangle = \begin{cases} 1 & m = 0 \\ 0 & if \ m > 0 \end{cases} .$$ (3.8)

As mentioned above, if the response surface is obtained by the PCE; then, the mean value and standard deviation of the stochastic response can be approximated by the following formulas:

$$\begin{aligned} E[u] &= \left\langle \sum_{j=0}^{P} u_j \Psi_j(\zeta) \right\rangle \\ &= u_0 \left\langle \Psi_0(\zeta) \right\rangle + \sum_{j=1}^{P} u_j \left\langle \Psi_j(\zeta) \right\rangle \\ &= u_0 \ , \end{aligned}$$ (3.9)

$$\begin{aligned} Var[u] &= \left\langle \left(u - E[u] \right)^2 \right\rangle \\ &= \left\langle \left(\sum_{j=0}^{P} u_j \Psi_j(\zeta) - u_0 \right)^2 \right\rangle \\ &= \sum_{j=1}^{P} u_j^2 \left\langle \Psi_j(\zeta)^2 \right\rangle . \end{aligned}$$ (3.10)

Furthermore, the probability density function can also be obtained by the output responses.

When we assume that a random variable follows a different probability distribution, its corresponding orthogonal polynomial is also different[7]. Here, as shown in the table 3.1, we give a detailed introduction which includes random variables are considered as discrete cases and continuous cases.

Table 3.1 Weiner-Askey Scheme

	Random variable	Polynomial	Support
Continuous	Gaussian	Hermite	$(-\infty, \infty)$
	Gamma	Laguerre	$[0, \infty)$
	Beta	Jacobi	$[a, b]$
	Uniform	Legendre	$[a, b]$
Discrete	Poisson	Charlier	$\{0, 1, 2, ...\}$
	Binomial	Krawtchouk	$\{0, 1, ..., N\}$
	Negative Binormial	Meixner	$\{0, 1, 2, ...\}$
	Hypergeometric	Hahn	$\{0, 1, ..., N\}$

According to the table, we can clearly understand that we should use the Hermite polynomial when we assume that the random variable follows the Gaussian distribution, we should use the Laguerre polynomial when the random variable follows the gamma distribution, and so on. As an example, in the table below, we give the form of the Hermite polynomial of the one dimensional and two dimensional, respectively

Table 3.2 One-Dimensional Hermite polynomial chaoses[5]

i	**Order of the Basis function**	i^{th} **Basis function** Ψ_i	$\langle \Psi_i^2 \rangle$
0	0	1	1
1	1	ζ	1
2	2	$\zeta^2 - 1$	2
3	3	$\zeta^3 - 3\zeta$	6
4	4	$\zeta^4 - 6\zeta^2 + 3$	24

Table 3.3 Two-Dimensional Hermite polynomial chaoses[5]

i	Order of the Basis function	ith Basis function Ψ_i	$\langle \Psi_i^2 \rangle$
0	0	1	1
1	1	ζ_1	1
2		ζ_2	1
3	2	$\zeta_1^2 - 1$	1
4		$\zeta_1 \zeta_2$	2
5		$\zeta_2^2 - 1$	2
6	3	$\zeta_1^2 \zeta_2 - \zeta_2$	2
7		$\zeta_1 \zeta_2^2 - \zeta_1$	2
8		$\zeta_1^3 - 3\zeta_1$	6
9		$\zeta_2^3 - 3\zeta_2$	6

Comparing Table 3.2 with Table 3.3, we can find that when there is one random variable, the four order base functions only need five polynomials to achieve. When two random variables are considered, the three order base functions require nine polynomials. That is, when more random variables are considered, in order to better represent random characteristics, we may need to use more polynomials.

Thus far, some important aspects of PCE have been described with respect to uncertainty analysis. In the following chapters, we will introduce the application of PCE, and give a practical example to understand application more deeply.

3.3.2 Application of Polynomial Chaos

This section describes the application of polynomial chaos [7]. First, we consider a stochastic differential equation that input random variable (ζ) .

$$\mathcal{L}\big(X(\mathbf{x},\zeta);u(\mathbf{x},\zeta)\big) = f\big(\mathbf{x},\zeta\big), \tag{3.11}$$

where, $\mathcal{L}$ is a general differential equation and it can be represented as linear or non-linear. $X(\mathbf{x},\zeta)$ represents the input parameter, which is a function of the random variable ζ, and $u\big(\mathbf{x},\zeta\big)$ is the response term, which also contain a random variable because the input parameters contain random variables. $f\big(\mathbf{x},\zeta\big)$ is the source term (such as Load in finite element equation) which may also contain random variables ζ or no random variables ζ.

When we consider that the geometry of the structure, the material properties, etc. contain uncertainty, we can introduce this uncertainty into the system. So we can find that the response will also contain some uncertainty because the input parameters contain uncertainty, and the response $u\big(\mathbf{x},\zeta\big)$ can be represented by Polynomial Chaos Expansion (PCE) as follows:

$$u\big(\mathbf{x},\zeta\big) = \sum_{i=0}^{P} u_i\big(\mathbf{x}\big)\Psi_i\big(\zeta\big) . \tag{3.12}$$

where $u_i\big(\mathbf{x}\big)$ are the unknown coefficients of the PCE approximation and P is the number of order, and $\Psi_i\big(\zeta\big)$ are the polynomial basis functions (Table 3.2 and Table 3.3). Substituting Equation(3.12) into Equation(3.11), the equation is Expressed as follows:

$$\mathcal{L}\left(X(\mathbf{x},\boldsymbol{\theta}); \sum_{i=0}^{P} u_i\big(\mathbf{x}\big)\Psi_i\big(\boldsymbol{\theta}\big) \right) = f\big(\mathbf{x},\boldsymbol{\theta}\big), \tag{3.13}$$

where, because the polynomial basis functions $\Psi_i(\boldsymbol{\theta})$ contain polynomial orthogonal properties, we can obtain the following equation by multiplying both sides of Equation(3.13) by $\Psi_t(\boldsymbol{\theta})W(\boldsymbol{\theta})$, and integrating the equation.

$$\left\langle \mathcal{L}\left(\mathbf{x}(\boldsymbol{\theta}); \sum_{i=0}^{P} u_i(\mathbf{x})\Psi_i(\boldsymbol{\theta}) \right), \Psi_t(\boldsymbol{\theta}) \right\rangle = \left\langle f(\mathbf{x}(\boldsymbol{\theta})), \Psi_t(\boldsymbol{\theta}) \right\rangle . \qquad (3.13)$$

When the $t = 0,1,\ldots P$, we can obtain a set of $(P + 1)$ simultaneous equations for each unknown coefficients $u_i(\mathbf{x})$ as follows.

$$\left\langle \mathcal{L}\left(\mathbf{x}(\boldsymbol{\theta}); \sum_{i=0}^{P} u_i(\mathbf{x})\Psi_i(\boldsymbol{\theta}) \right), \Psi_0(\boldsymbol{\theta}) \right\rangle = \left\langle f(\mathbf{x}(\boldsymbol{\theta})), \Psi_0(\boldsymbol{\theta}) \right\rangle$$

$$\left\langle \mathcal{L}\left(\mathbf{x}(\boldsymbol{\theta}); \sum_{i=0}^{P} u_i(\mathbf{x})\Psi_i(\boldsymbol{\theta}) \right), \Psi_1(\boldsymbol{\theta}) \right\rangle = \left\langle f(\mathbf{x}(\boldsymbol{\theta})), \Psi_1(\boldsymbol{\theta}) \right\rangle$$

$$\left\langle \mathcal{L}\left(\mathbf{x}(\boldsymbol{\theta}); \sum_{i=0}^{P} u_i(\mathbf{x})\Psi_i(\boldsymbol{\theta}) \right), \Psi_2(\boldsymbol{\theta}) \right\rangle = \left\langle f(\mathbf{x}(\boldsymbol{\theta})), \Psi_2(\boldsymbol{\theta}) \right\rangle$$

$$\vdots$$

$$\left\langle \mathcal{L}\left(\mathbf{x}(\boldsymbol{\theta}); \sum_{i=0}^{P} u_i(\mathbf{x})\Psi_i(\boldsymbol{\theta}) \right), \Psi_P(\boldsymbol{\theta}) \right\rangle = \left\langle f(\mathbf{x}(\boldsymbol{\theta})), \Psi_P(\boldsymbol{\theta}) \right\rangle$$

$$(3.14)$$

Through Equation (3.14), we can find that this simultaneous equations contains P+1 unknown coefficients $u_0(\mathbf{x}), u_1(\mathbf{x}),\ldots,u_P(\mathbf{x})$ and P+1 equations. So we can solve the unknown coefficients $u_0(\mathbf{x})$, $u_1(\mathbf{x})$,..., $u_P(\mathbf{x})$ by solving this simultaneous equations(Eq.(3.14)).

In Figure 3.5, we show process of application of polynomial chaos for make the application of PCE easier to understand. When the inherent of model contains uncertainty and it obeying two different random variables (θ_1,θ_2), the input parameters $H(\boldsymbol{\theta})$ is expressed as a polynomial form. And the response $u(\boldsymbol{\theta})$ can be approximated as polynomial function about the input random variables $\boldsymbol{\theta}=\{\theta_1,\theta_2\}$. Then we apply the expressed input parameters and the approximated output response

to the governing equations, the governing equations will be redefined as simultaneous equations, the unknown coefficients u_i can be solved. The response surface $u(\boldsymbol{\theta})$ can be obtained when the coefficients u_i is determined. And when we want to obtain the probability density function of the response, we can bring the M samples of standard random variable $\boldsymbol{\theta}$ into the resulting response surface $u(\boldsymbol{\theta})$ to obtain the probability density function.

Input parameters

ζ_1 ζ_2

The input parameter H is represented as polynomial form.

$$H(\zeta) = \sum_{i=0}^{P} H_i \Psi_i(\zeta)$$

Output parameters

The response can be assumed as polynomial form.

$$u(\zeta) = \sum_{i=0}^{P} u_i \Psi_i(\zeta)$$

Governing equations

solving unknown coefficients u_i

obtained response surface $u(\zeta)$

Response uncertainty
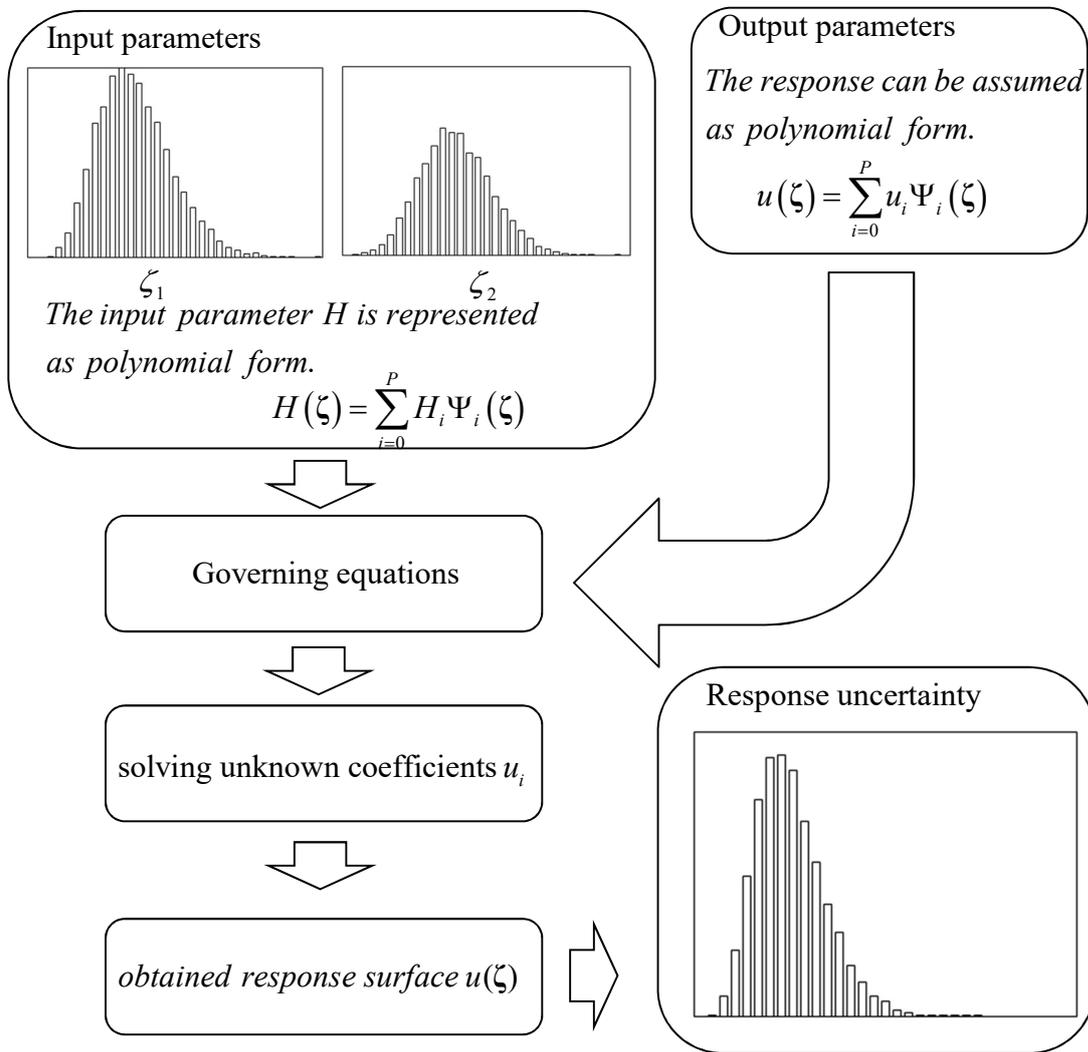
Figure 3.5 Process of Application of Polynomial Chaos

3.3.3 Numerical Example

In this section, we give a simple numerical example of a continuous free-standing beam deflection problem for a better understanding of the application of polynomial chaos. The deflection problem of a beam is discussed considering Young's modulus with uncertainty. The governing differential equation (Equation(3.15)) of this problem is give follow, and $3EI/L^3$ corresponding to $X(\mathbf{x}, \zeta)$ of Eq.(3.11), δ corresponding to $u(\mathbf{x}, \zeta)$ of Eq.(3.11) and P corresponding to $f(\mathbf{x}, \zeta)$ of Eq.(3.11). And we can see that P ($f(\mathbf{x}, \zeta)$) does not contain uncertainty in this simple example.

$$\frac{3EI\delta}{L^3} = P \,, \qquad (3.15)$$
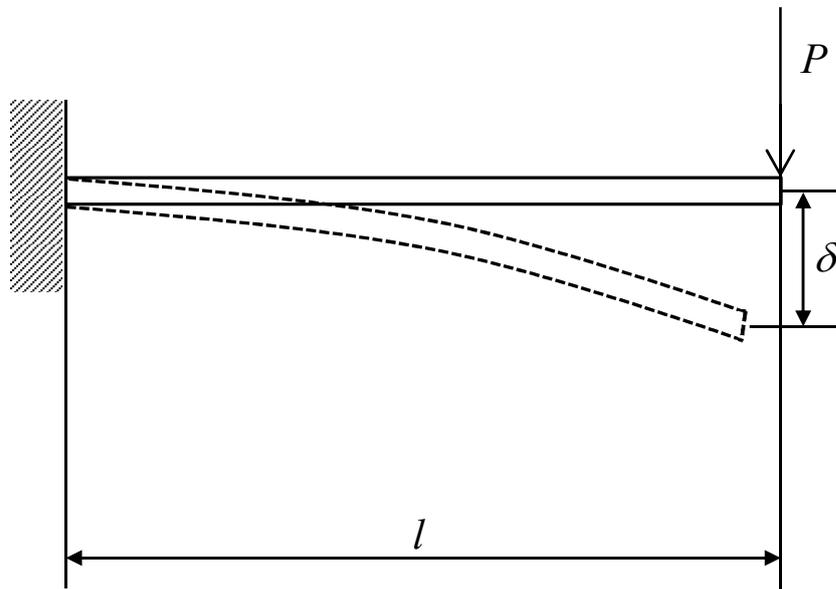
Figure 3.6: Free-standing beam

where I is moment of inertia of area, we consider it as solid cylinder, then I is

$I = \dfrac{\pi R^4}{4}$, the R is radius of the solid cylinder. And L is the length of beam, δ is the

maximum deflection, P is load. They are all considered as deterministic parameters. As uncertainty parameters, the elastic modulus E is assumed to following Gaussian distribution with mean E_μ and standard deviation of E_σ. So the E can be expressed as follows:

$$E = E_\mu + E_\sigma \zeta, \tag{3.16}$$

where, ζ is a standard normal random variable. In other words, Young's modulus can be represented as follows by Hermite polynomial of 1st order because of $\Psi_0(\zeta) = 1, \Psi_0(\zeta) = \zeta$ (From *Table 3.2*).

$$\begin{aligned} E &= E_\mu \Psi_0(\zeta) + E_\sigma \Psi_1(\zeta) \\ &= \sum_{i=0}^{1} E_i \Psi_i(\zeta) \end{aligned} \tag{3.17}$$

In this problem, the maximum deflection (δ) is considered as response of the system. It will contain some uncertainty because we assume that there is uncertainty in Young's modulus. So that the response δ can be approximate by using a one dimensional PCE of the random variable ζ of n order, and show following.

$$\delta(\zeta) = \sum_{i=0}^{n} \delta_i \Psi_i(\zeta), \tag{3.18}$$

where $\Psi_i(\zeta)$ are the orthogonal basis function which is Hermite polynomials of the variables (ζ) , and δ_i are the unknown coefficients of PCE, ζ is a Gaussian random variable. The main idea to approximate the response $\delta(\zeta)$ by using orthogonal polynomials with unknown coefficients δ_i . At this point, it is quite

obvious that we need to estimate δ_i for to obtain statistics of the response completely. This can be achieved as described following.

First we substitute Equation(3.17) and Equation(3.18) into Equation(3.15) as follows.

$$\frac{3I}{L^3}\sum_{i=0}^{1}E_i\Psi_i\left(\zeta\right)\sum_{i=0}^{n}\delta_i\Psi_i\left(\zeta\right)=P, \tag{3.19}$$

And consider the orthogonality of polynomials, we multiply both sides of Equation(3.19) by $\Psi_t(\zeta)w(\zeta)$, and integrating the equation, the following equations are obtained.

$$\frac{3I}{L^3}\sum_{i=0}^{1}\sum_{j=0}^{n}E_i\delta_j\left\langle\Psi_i\left(\zeta\right)\Psi_j\left(\zeta\right)\Psi_t\left(\zeta\right)\right\rangle=P\left\langle\Psi_t\left(\zeta\right)\right\rangle, \tag{3.20}$$

where the Young's modulus E is approximated by 1th order polynomial as shown in Equation(3.17). Thus, in this problem, we can assume that the order (n) of PCE approximation of Equation(3.18) is 1. And the $t=0,1$, we can obtain the simultaneous equations, it can be derived as follows:

$$\begin{bmatrix}\dfrac{3I}{L^3}\sum_{i=0}^{1}\sum_{j=0}^{1}E_i\delta_j\left\langle\Psi_i\left(\zeta\right)\Psi_j\left(\zeta\right)\Psi_0\left(\zeta\right)\right\rangle\\[4mm]\dfrac{3I}{L^3}\sum_{i=0}^{1}\sum_{j=0}^{1}E_i\delta_j\left\langle\Psi_i\left(\zeta\right)\Psi_j\left(\zeta\right)\Psi_1\left(\zeta\right)\right\rangle\end{bmatrix}=\begin{bmatrix}P\left\langle\Psi_0\left(\zeta\right)\right\rangle\\[4mm]P\left\langle\Psi_1\left(\zeta\right)\right\rangle\end{bmatrix}, \tag{3.21}$$

We can see that there are two equations with two unknowns δ_0, δ_1. Therefore, we can find that we can be obtained the response surface $\delta\left(\zeta\right)$ by solving only two deterministic equations. When we need to know more statistics, such as the mean and standard deviation, this information can be obtained by Eq.(3.9) and Eq.(3.10) when the response surface is obtained. Moreover, when we need to get the probability distribution function of the response, it can be obtained by using large number of realizations of standard normal random variable (ζ) and plugging them into Equation(3.18). Therefore, we can clearly understand that this method avoids the multiple computations of MCS very well, because this method directly obtains the response surface with random

variables.

And for clarity we show first order PCE (n=1). The inner product values are shown in the following tables:

Table 3.4 Inner Products

$$\left\langle \Psi_i(\theta)\Psi_j(\theta)\Psi_0(\theta)\right\rangle$$

i / j	0	1
0	1	0
1	0	1

$$\left\langle \Psi_i(\theta)\Psi_j(\theta)\Psi_1(\theta)\right\rangle$$

i / j	0	1
0	0	1
1	1	0

Here, we try to give the specific values of deterministic parameters and uncertain parameters to evaluate the uncertainty of maximum deflection.

$$L = 1000mm$$
$$E = Normal(\mu_X = 205800MPa, \sigma_X = 1000MPa)$$
$$q = 1MPa / mm^2$$
$$R = 1mm \; (I = \frac{\pi}{4}mm^4)$$

When Table3.4 is considered, we can obtain the following equation.

$$\frac{3I}{L^3}\left(E_0\delta_0 + E_1\delta_1\right) = P \,,$$
$$\frac{3I}{L^3}\left(E_1\delta_0 + E_0\delta_1\right) = 0 \,. \tag{3.22}$$

Bring in parameter values to the equation Eq.(3.22)

$$\frac{3}{1000^3}\frac{\pi}{4}\left(205800\delta_0 + 1000\delta_1\right) = 1 \,,$$
$$\frac{3}{1000^3}\frac{\pi}{4}\left(1000\delta_0 + 205800\delta_1\right) = 0 \,. \tag{3.23}$$

We can get δ_0 and δ_1 by solving this simultaneous equations.

$$\delta_0 = 2062.38$$
$$\delta_0 = -10.0209$$

Thus, the response surface $\delta(\zeta)$ can be written as following.

$$\delta(\zeta) = 2062.38 - 10.0209\zeta \,. \tag{3.24}$$

And he following Table shows the mean and standard deviation of the response $\delta(\zeta)$ which is obtained from MCS and PCE and Eq.(3.9) and Eq.(3.10). The result of MCS is obtained by running 10000 simulations of the deterministic problem in

regard to the Young's modulus of a given 10,000 samples. And, the result of PCE is obtained by introducing 10,000 random variables into Eq.(3.24). We can find that the results of MCE and PCE are very similar, and the mean and standard deviation equations calculated according to polynomial statistical Eq.(3.9) and Eq.(3.10) are also close to MCS to some extent. Therefore, when we get the response surface, we can easily and quickly calculate the statistical properties through Eq.(3.9) and Eq.(3.10). And we can find that this method avoids multiple calculations of MCS and greatly saves costs.

Table 3.4 Comparative statistics of response $\delta(\zeta)$

Statistics of $\delta(\zeta)$	MCS (10000 samples)	PCE (10000 samples)	PCE (By Eq.)
Mean (mm)	2062.26	2062.33	2062.38
Standard dev. (mm)	10.10	10.10	10.02

In Figure 3.7, the probability density distribution is shown by MCS and PCE. We can be find that PCE is shown to prove that the results are well consistent with MCS results. And we can easily get the probability distribution density function by using this method.
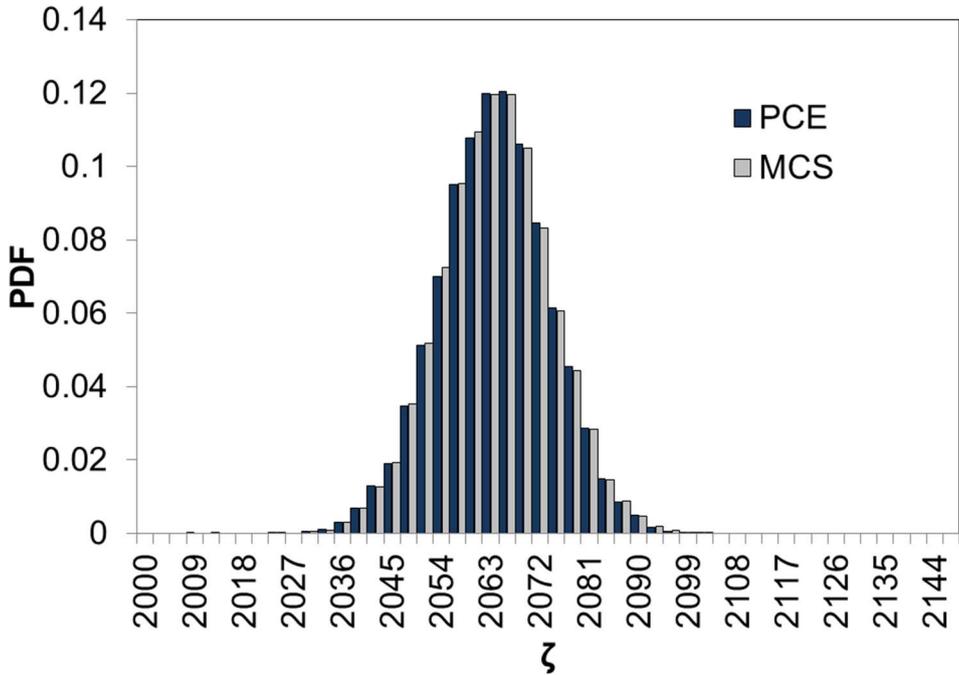
Figure 3.7: The probability density distributions

3.4. Conclusions

In this chapter, we mainly describe the relevant knowledge of stochastic analysis, including the concept of stochastic analysis, the types of stochastic analysis, and the knowledge of the current mainstream polynomial expansion method.

We can learn from this chapter that stochastic analysis is mainly divided into two categories: intrusive method and non-intrusive method. For stochastic analysis, intrusion method will become the main development direction because it saves computation time and cost more effectively. In recent years, polynomial expansion methods have often been used to represent the inherent randomness of structures. It is applied to analytical methods, and a new method is formulated. In this chapter, the properties, types, applications, and specific examples of polynomial expansion methods are given.

Reference

[1] Point of management for steel frame construction, Japan Federation of Construction Contractors, 2015. (In Japanese)

[2] M. M. Htun, Y. Kawamura, and M. Ajiki, 'A Study on Random Field Model for Representation of Geometry of Corroded Plates and Estimation of Stochastic Properties of Their Strength', *J. Jpn. Soc. Nav. Archit. Ocean Eng.*, vol. 18, pp. 91–99, 2013.

[3] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Courier Corporation, 2003.

[4] Sakamoto Shigehiro and Ghanem Roger, 'Polynomial Chaos Decomposition for the Simulation of Non-Gaussian Nonstationary Stochastic Processes', *J. Eng. Mech.*, vol. 128, no. 2, pp. 190–201, Feb. 2002.

[5] S.-K. Choi, R. V. Grandhi, and R. A. Canfield, *Reliability-based structural design*. London: Springer, 2007.

[6] D. Xiu and G. E. Karniadakis, *Modeling uncertainty in flow . . .* 2002.

[7] D. Xiu and G. Karniadakis, 'The Wiener--Askey Polynomial Chaos for Stochastic Differential Equations', *SIAM J Sci Comput*, vol. 24, pp. 619–644, Oct. 2002.

4. STOCHASIC ISOGEOMETRIC ANALYSIS (SIGA) METHOD FOR UNCERTAINY IN SHAPE

4.1. Overview

There are various uncertainties in the field of contemporary engineering, such as corrosion surfaces, material properties, and geometric shapes. These uncertainties may be caused by the analytic object itself or by external causes. Therefore, many analytical methods for the problem of uncertainty have been developed[1]–[3]. However, in these analytical methods, it is rarely aimed at the analysis and research of uncertainty in shape.

However, the shape uncertainty problem is an inherent problem that is inevitable in actual engineering design and has a great impact on the safety of the engineering structure. In figure 4.1, an example of a hole that is drilled in a flat metal sheet is given. Because of the processing technique, the hole drilled has inherent uncertainty

in shape. This uncertainties in shape caused by such processing will cause great potential safety hazards in the operation of machinery and equipment.

Figure 4.1 A hole is drilled in a flat metal sheet

Actually, there are many more examples like this one with regard to generating shape uncertainty in the process of product development and design such as shape uncertainty caused by corrosion, and shape errors that occur during processing, etc. Therefore, this study focuses on the development of analytical methods for uncertainty in shape. We make full use of the geometric characteristics of the control points, that is, the geometric shapes can be manipulated freely and accurately by

moving the control points. And the structural analysis with consideration to the uncertainty in shape was implemented by importing the parameters of uncertainty (mean and deviation etc.) into the control point coordinates. Thus, the Stochastic isogeometric analysis method for uncertainty in shape is developed. This study explored one possible extension of IGA method in the field of computational stochastic mechanics, that is the extension of classical deterministic isogeometric analysis method into a probabilistic analytical framework to evaluate the uncertainty in shape.

In the previous two chapters we introduced the NURBS as a pre-analysis tool and some of the main stochastic analysis techniques employed in this study, in detail. These important concepts and knowledge will become important tools for building stochastic isogeometric analysis framework. More specifically, we use NURBS in IGA to construct analytic geometry model and isogeometric space and parameterization etc. The stochastic analysis techniques are utilized to achieve the uncertainty represent in shape. A schematic illustration of the ideas is presented in Figure 4.2 in regard to the stochastic isogeometric analysis method for uncertainty in shape. The random variable as the input data is introduced control point in CAD, and the NURBS is applied to numerical analysis frameworks and geometric designs as shape functions and basis functions, respectively. Finally, the parameters of the computational domain are used to analyze the uncertainties in shape.
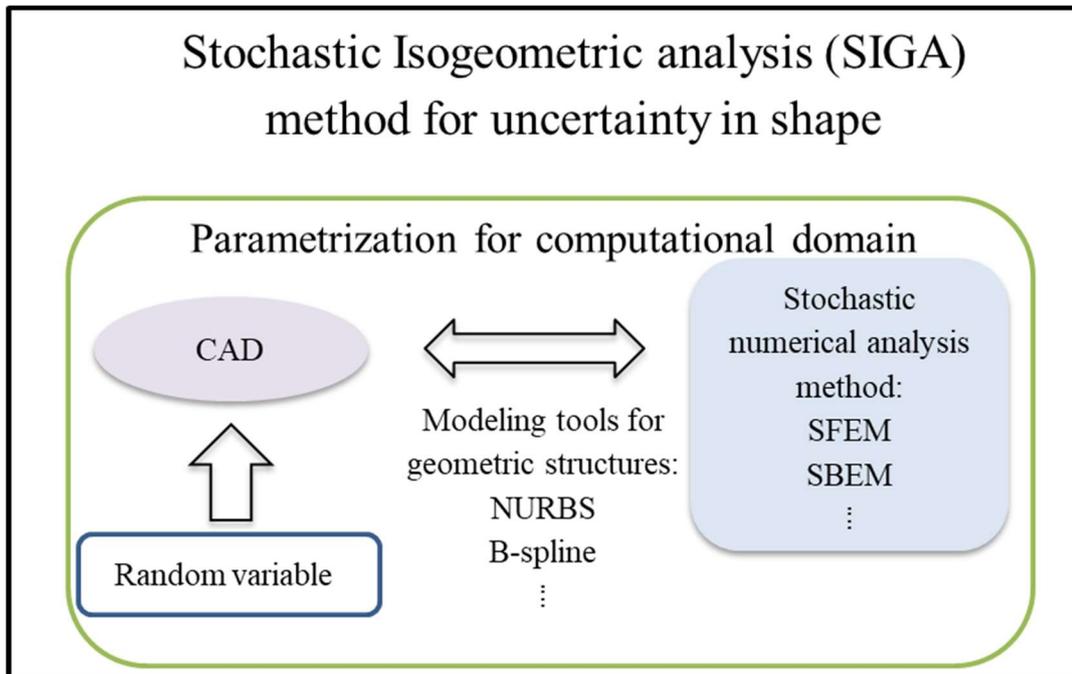
Figure 4.2 Schematic illustration of the stochastic isogeometric analysis method for uncertainty in shape

4.2. Establishment of analytical viewpoints

In this study, we combined IGA method and the stochastic methodology in order to create a new method for estimating the uncertainty in shape. Moreover, in this section, we use two terminologies ("physically-based" and "FEA-based") in order to describe the different analytic viewpoint between the proposed method and the SFEM. These two terminologies have been used in earlier methods integrating FEM and computational geometry [4]. Attempts by early engineering technicians to combine finite element method with computer aided geometric design are referred to as *physically-based modeling*. And try to connect them to solve the actual engineering problems[5]–[7]. In this study, we redefine this proper noun, namely uncertainty in shape was considered from a physically-based point of view, instead of the classic FEA-based point of view. The following sections briefly discuss the application of these two analytic points of view within uncertainty analysis framework, respectively. Subsequently, the formulation of SIGA for uncertainty in shape will be defined.

4.2.1. Classic FEA-based Point of View for Uncertainty Analysis

In general, the use of traditional SFEM to discuss uncertainty in shape is based on the FEA-based point of view, i.e. the SFEM is built from the analytical framework of the classic FEM [8-10]. Thus, it inherits all the characteristics of FEA in the term of geometry and algebra. However, this kind of inheritance is always a double-edged sword, which has both advantageous advantages and inevitable limitations. SFEM inherits a stable analytical framework and a relatively complete theoretical basis in traditionally deterministic finite elements. But it is regrettable that the some of the characteristics inherited may hinder the further development of SFEM in terms of uncertainty in shape, although they may not have any impact on the numerical analysis in the FEM or other SFEM. Among them, a major development bottleneck is the geometric representation of the analysis object. The requirements for the performance of the geometric shapes are usually relatively high in analyzing

116

uncertainty in shape, especially for complex geometric structures. However, throughout the probability analysis process, SFEM use a geometry approximated by FE-mesh to represent the uncertainty in shape, namely, the random field is actually acted upon an approximate geometry instead of an actual geometric structure, so that leads to geometric errors, inevitably. Notwithstanding, through some mesh refinement methods inherited from the FEM, the analytical object can be as much as possible conform to the actual geometric structure, but this typical geometric error cannot be reduced indefinitely. Moreover, the analytical model is partitioned into the elements connected by nodes, and through linear interpolation shape functions to achieve interpolation within the elements. Therefore, all the information is tied to the finite element mesh, in the SFEM, the uncertain parameter information is introduced into nodal coordinates as well, in order to represent the uncertainty in shape. As the position of the corresponding node changes, the shape of the mesh and elements also changes. Thus, in order to avoid affecting the accuracy of the analytical results, the appropriate FE-mesh must be restructured by remeshing, repeatedly. This increases the difficulty and time of analysis. Furthermore, the FE-mesh is usually insensitive to detecting shape changes at the geometrical boundaries containing surface or curve, or even failing to represent the deformation process. For instance, there is a plate with a circular hole and we assume that the random field is acted on the location of the circular hole. At present, within the SFEM analytical framework, the uncertainty at hole can only be considered as changes in the size of the radius [8]. If we want to assume that the uncertainty representation is the shape change from circular to other irregularities, which is difficult to be achieved in the current SFEM analytical framework, let alone more complex geometric structures. If it is in the case of FEM, the geometric shape can be artificially and arbitrarily changed by the geometric visualization tools, and then repartition it into the FE mesh. However, in the actual analysis and formulation, the shape of the analytical object cannot be controlled artificially and visually, instead, use the algebraic method to deal with the analysis process. Therefore, in the analysis process of SFEM, the uncertainty representation in shape cannot be achieved simply by changing the coordinates of nodes with uncertain parameters, and also need to consider its influence on other elements and

nodes. This influence is difficult to be controlled and determined for complex uncertainty representation. In Reference [8-10], Chen et al. by setting the coefficients of the degree of change at each node to represent the uncertainty in shape and gave the corresponding algorithm, but that cannot perform well for geometric structures with curves and surfaces due to the drawbacks inherited from the FEM mentioned above. Besides, these coefficients of the degree of change also there is an impact on the analytical results to a certain degree. Namely, if the bad coefficients are determined, a bad FE-mesh will be generated, resulting in an inaccuracy result that may be obtained [11].

Through the discussion of the above classic FEA-based point of view, we can know that despite the powerful FEM has been widely expanded and applied to various engineering fields, and has achieved prominent achievements. However, for the uncertainty analysis in shape based on FEM analysis framework, some inherent characteristics of FEM limit its development and greatly increase the difficulty and cost of analysis, Therefore, in order to break these bottlenecks, in this study we introduced the physically-based analytical point of view to deal with the uncertainty in shape.

4.2.2 .Physically-based point of view for uncertainty analysis

In this study, a novel method is proposed in the aspect of reliability analysis for uncertainty in shape, and it is carried out from a physically-based point of view, i.e. the entire probability analysis uses an actual geometric structure to represent uncertainty in shape instead of an approximate one. Similarly, the SIGA for uncertainty in shape is an extension from the classical isogeometric analysis framework. Some of the characteristics inherited from IGA make up for the shortcomings of traditional SFEM on the uncertainty analysis in shape. The SIGA takes full advantage of these the native characteristics of IGA, that is using the same geometry description in engineering design and analysis processes, respectively; Thus the random field is acted upon the geometric entity, the geometric errors are eliminated to the utmost, especially, the more complex the geometry, the more

obvious this advantage, for example, circle and ellipse, and irregular geometry, etc. And, since a physical mesh on the geometric entity is used for discretization in the uncertainty analysis process, it does not need to be like FEM the mesh repartition due to shape change and determining coefficients of the degree of shape at each node, so the calculation load is reduced. In addition, because the NURBS basis function is used as the shape function of the analytic part in this study, the shape function used in this study is smoother and more continuous than the one in SFEM. In Figure 4.3, we show, 1st, 2nd, 3rd and 4th -degree NURBS base functions defined by knot vector $\Xi = \{0, 0, 1, 1, 2, 3, 4, 4, 5\}$. We can observe that the 2nd - degree NURBS basis function is C^1 continuity, although it is repeated at position $\xi = 1$ and $\xi = 4$. And with the increase of the number of degree, its continuity and smoothness are higher. In FEM, the continuity of the base function is always C^0, which is similar to the NURBS base function of the first degree.
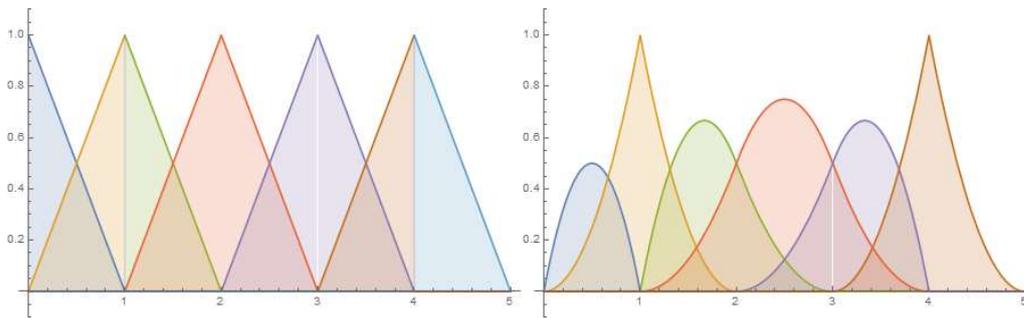
Degree 1   Degree 2
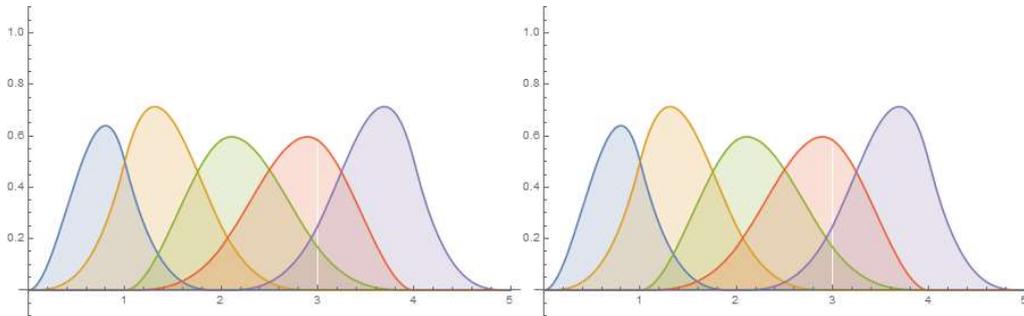
Degree 3   Degree 4

Figure 4.3 1st -,2nd -, 3rd -and 4th -degree NURBS base functions defined by knot

vector $\Xi = \{0, 0, 1, 1, 2, 3, 4, 4, 5\}$

Another major advantage of the physically-based point of view, besides being precision in representing the complex geometric structure and its less computationally expensive, is that it can define the analytical boundaries flexibly, exactly and easily. This advantage follows from the fact that the local support property and the local modification scheme property of NURBS [12], coupled with the power of NURBS which ability to directly manipulate the geometric shape by adjusting the control points. These features of NURBS have been described in detail in Chapter 2. Therefore, the uncertainty in shape can be represented arbitrarily, and region and degree of shape change can be controlled easily, flexibly and exactly. The specific example shown in Figure 4.4 consists of one-quarter of a circular disk with a circular hole at the centre. It was assumed that the shape of the circular hole entails uncertainty. As shown in Figure 4.4(a), the control points 1,2, and 3 control the shape of the hole directly, and any one of them can import uncertain parameters to represent the uncertainty in shape at the circular hole. Therefore, the determination and control of the analytical boundary for the region of interest became dramatically easier. We do not need to consider moving other control points to represent uncertainty in shape and the impact on the ones.
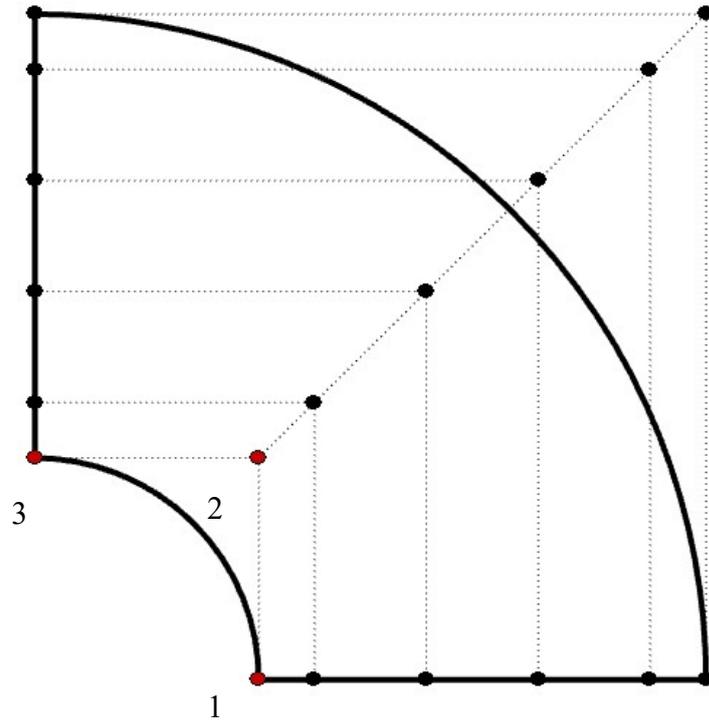
(a) Control points 1, 2 and 3 for controlling circular hole at the centre.
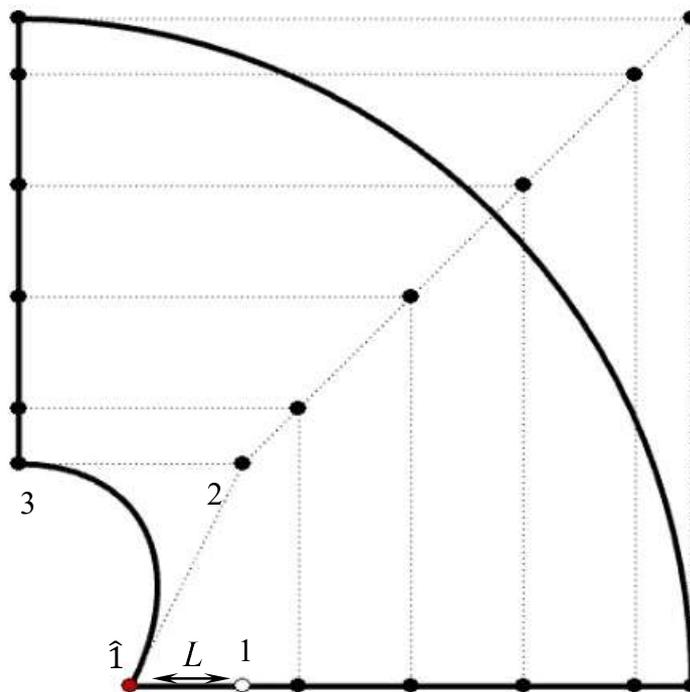
(b) Shape of circular hole at centre is changed by moving control point 1 to $\hat{1}$.

Figure 4 .4: One quarter of circular disk with circular hole at centre

Through the above discussion, it is obvious that the natural geometric features of NURBS provides great convenience for uncertainty analysis in shape. Here, it should be mentioned again that the ability of control points manipulating geometric shape intuitively and freely was adequately exploited to achieve representation of uncertainty in shape in this study. Thereby, degrees of freedom and uncertain parameters are located at the control points. And based on an affine invariance property that follows from the partition of unity property in the NURBS [12,13], ensures that the random field acts precisely on the geometric entity through the control points, in spite of the fact that the control points are commonly not interpolatory in NURBS, unlike in standard stochastic FEM.

4.2.3. Identifying degree and regions of the actual domain influenced by each of control points.

In this study, part of the power of NURBS was utilized, that is the ability to intuitively change their shape by adjusting the control points. However, compared with nodal uncertainty in SFEM, the identification of uncertainty of control points in SIGA is a challenging problem. This is because control points do not typically lie on the shape. In this study, we utilize several properties of NURBS (i.e. partition of unity property, local support property and the local modification scheme property) to process identification of uncertainty of control points in SIGA. A complete discussion of these properties is beyond the scope of this chapter, but a good introduction in the context of NURBS can be found in Chapter 2. Next, let us elaborate on how we identified the uncertainty of control points in this study. In Chapter 2 of this paper, we have introduced the definition of NURBS curves and surfaces and their properties. A geometric model in the physical space is constructed by taking a linear combination of NURBS basis functions. Note that the elements and mesh are also constructed while building geometric models. Note also that for the definition of element and mesh, the SFEM and SIGA are the difference. In stochastic finite element method, there is one notion of a mesh and one notion of an

element. Elements are usually defined by their nodal coordinates and the degrees-of-freedom, that is, if we move a node, it can only affect the geometric shape of the element it is in. The FE mesh is structured from elements connected by nodes, and the random field acts directly on the FE mesh as the object of analysis. Therefore, for SFEM, the identification of shape uncertainty is easy. However, in NURBS, there are two notions of mesh, the control mesh (that is also known as the "control net") and the physical mesh. Piecewise linear interpolation of the control points gives the so-called control mesh. The control mesh consists of multilinear elements, in one dimension they are line segments, in two dimensions they are bilinear quadrilateral elements. The control mesh has the look of a typical finite element mesh of multilinear elements, but the control mesh does not conform to the actual geometry. The physical mesh is constructed based on the knot span and it is in line with the geometric entities in physical space, namely the knots in the parameter space, by geometric mapping, partition the geometric entity into elements. Thus, the elements have representations in both a parameter space and physical space. Note that these elements are the smallest geometric entities in physical space we deal with, and so they also are thought of as *micro-element.* Note also that, when we talk about "elements" without further description, we usually mean the elements defined by the knot span in physical mesh. In this study, we introduce uncertain parameters into the control points to implement shape uncertainty analysis, but these control points are usually inserted into the control mesh, not in the physical mesh. Therefore, we must confirm the relationship between the control point and the physical mesh in order to identify the effect of the control point containing the random variable on the actual geometric shape. Namely, we should know the regions of the actual domain where control point with uncertainty can influence.

Let's take a closer look at how does uncertainty in shape of geometry represent if the parameters of uncertainty are introduced a control point. In NURBS, the control points are usually not interpolatory, unlike in standard finite element analysis. However, uncertainty parameters in this study and degrees of freedom are thought to as control variables are located at the control points, and the information located at

the control points can be precisely mapped by the affine transformations into the geometric entity. This is because the NURBS possess an affine invariance property following from the partition of unity property [12], i.e. an affine transformation is applied to the geometric entity by applying it directly to the control points. Affine transformations include translations, rotations, scalings, and uniform stretchings and shearings. And based on the affine invariance property, ensures that the random field acts precisely on the geometric entity through the control points. This property is very important for our work. Based on this property we can exactly represent uncertainty in actual geometry through introduce uncertainty into the control points. Additionally, from a geometric point of view, we can consider that the control points imported uncertainty can be expressed in the form of a randomly varying geometric vector. Therefore, it needs to be discussed from two aspects of direction and distance how the uncertainty introduced into the control point is identified on the actual geometry. For the convenience of explanation, this part is mainly discussed based on the NURBS curve, and the NURBS surfaces follow directly from the curve case.

The local modification scheme property discussed on an earlier section states that changing the position of the control point B_i only affects $p + 1$ elements at actual geometry, where p is the degree of a NURBS curve. In fact, the shape change is translational in the direction of the control point being moved. The Figure 4.5 illustrated this effect. Both the curve in Figure 4.5(a) and (b) are the NURBS curves of degree 2 described by 9 control points (denoted by the red dots), and the knots (marked with red crosses). These knots defined 7 knot spans (elements) were denoted in figure 4.5 using 7 different colored curve segments. The Figure 4.5 (a) showed the original curve. If the control point $B(5)$ at the original curve is moved to a new position $B'(5)$ as indicated by the red vector in Figure 4.5(b), the curve is moved in the same direction. The results were shown in Figure 4.5(b), we took respectively two points located in the knots $\xi = 3/7$ and $\xi = 4/7$ on the curve as the reference for better visualization. It is obvious that $C(3/7)$ and $C(4/7)$ are moved in the

same direction to $C'(3/7)$ and $C'(4/7)$, as shown by both the blue vectors. Note, as you can see, moving the control point $B(5)$ only affects elements 3, 4 and 5 on the curve, their position is changed to curve segments $3'$, $4'$ and $5'$. The elements 1, 2, 6 and 7 are not affected.
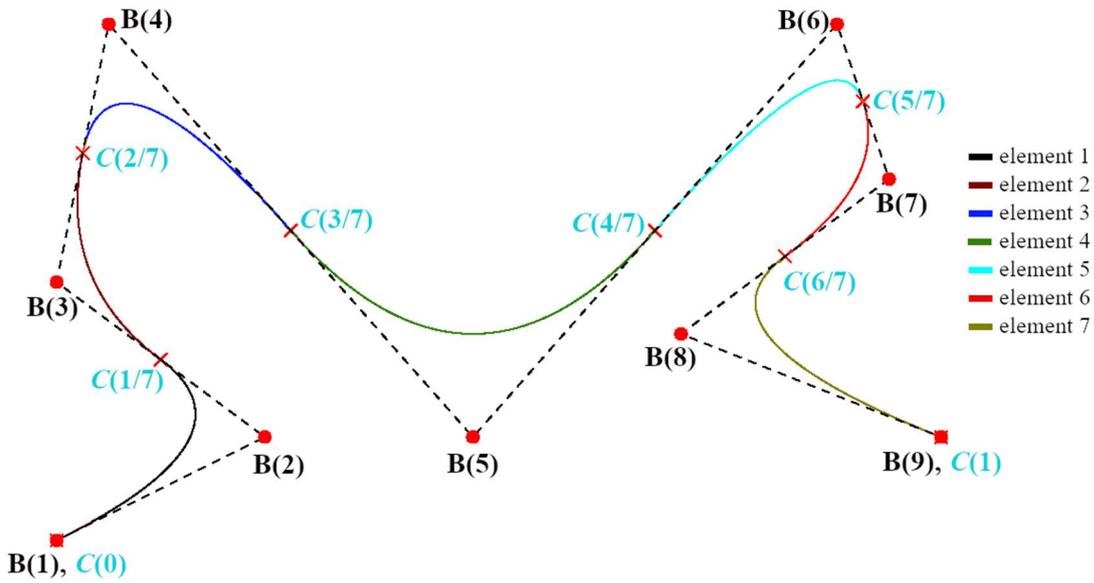
(a) original curve is defined by knot vector

$$\Xi = \{0,0,0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1, 1, 1\}$$ and 9 control points. The curve is

partitioned into elements, are denoted by the curve segments of different colours.
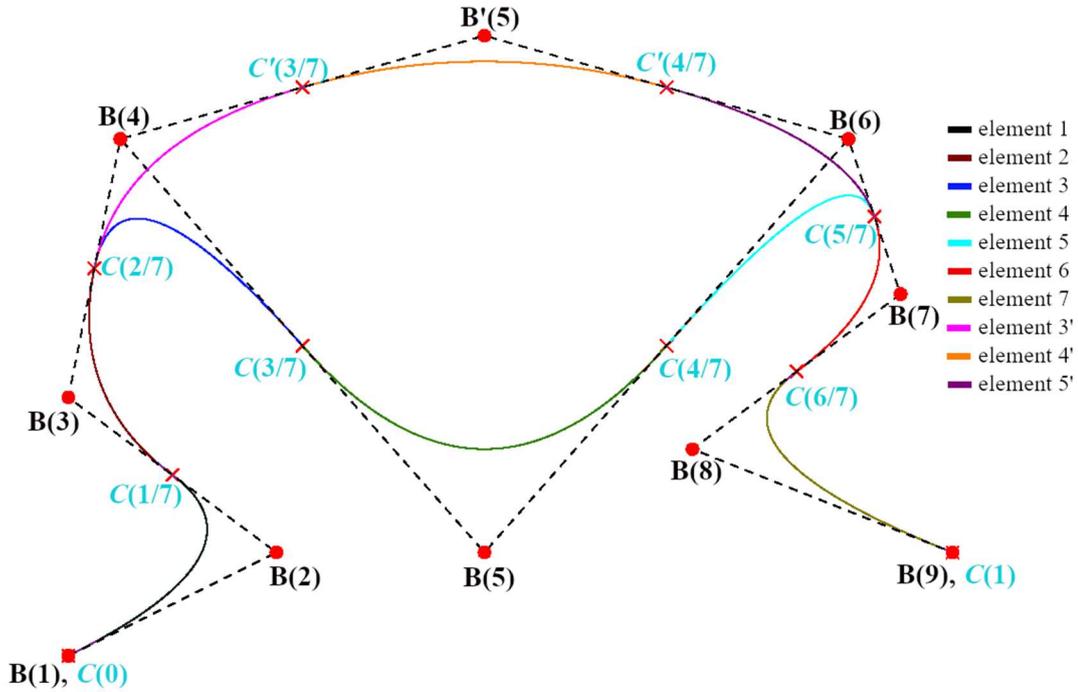
(a) The shape of curve is changed by the moving the control point $B(5)$ to $B'(5)$.

Figure 4.5. The piecewise quadratic NURBS curve.

Let us take a look at some details from the algebraic perspective. Through Equation (2.35) in the Chapter 2, a piecewise-polynomial NURBS curve of degree p can be given as follows:

$$C(\xi) = \sum_{i}^{n} R_{i,p}(\xi) B_i \qquad \text{(by 2.35)}$$

Let control point B_k be introduced uncertainty, i.e. the control point B_k is moved to a new position $B_k + \vec{v}$, where $\vec{v}$ is randomly varying geometric vector, that gives both the direction and length of this move. Since the new NURBS curve is defined $B_0, B_1, \cdots, B_k + \vec{v}, \cdots, B_n$ by its equation $C'(\xi)$ is the following.

$$
\begin{aligned}
C'(\xi) &= \sum_{i=0}^{k-1} R_{i,p}(\xi) B_i + R_{k,p}(\xi)(B_k + \vec{v}) + \sum_{i=k+1}^{n} R_{i,p}(\xi) B_i \\
&= \sum_{i=0}^{n} R_{i,p}(\xi) B_i + R_{k,p}(\xi) \vec{v} \qquad\qquad (4.1) \\
&= C(\xi) + R_{k,p}(\xi) \vec{v}
\end{aligned}
$$

Therefore, only the k-th term uses a control point containing uncertainty, $B_k + \vec{v}$, after regrouping we know that the new curve is the sum of the original curve and a randomly varying geometric vector $R_{k,p}(\xi)\vec{v}$. Since $R_{k,p}(\xi)$ is non-zero on the interval $\left[\xi_i, \xi_{i+p+1} \right)$, i.e. if ξ outside this interval, this extra term is zero (local support property of NURBS basis function), moving a control point only affects the shape of a section of the given curve.

In the above, if control point B_i is moved in certain direction to a new position $B_i{}'$, then a point on a NURBS curve at a fixed ξ value, where ξ is in $\left[\xi_i, \xi_{i+p+1} \right)$, will be moved in the same direction from B_i to $B_i{}'$. However, the distance moved is different from points on the original curve to ones on the new curve, i.e. the degree of curve change in $\left[\xi_i, \xi_{i+p+1} \right)$ is different. We can clearly know from the Equation (4.1) that the moved distance of any point on the curve is the $\left| R_{k,p}(\xi)\vec{v} \right|$. More precisely, give a ξ, we have fixed point $C(\xi)$ on the original curve and $C'(\xi)$ on the new curve, and $C'(\xi) = C(\xi) + R_{i,p}\vec{v}$. Since $\vec{v}$ gives the direction of movement, $C'(\xi)$ is the result of moving $C(\xi)$ in the same direction as the moved direction of

control point. The length of this translation is, of course, the length of vector $R_{k,p}(\xi)\vec{v}$. Therefore, when $R_{k,p}(\xi)$ reaches its maximum, the change of curve is the largest. For example, let us calculate the distance between $C(3/7)$ and $C'(3/7)$, that is the length of vector $R_{5,2}(3/7)\vec{v} = 0.5\vec{v}$ where $\vec{v}$ is the vector containing parameters of uncertainty and $R_{5,2}(3/7)$ is solved by Equation(2.34) in the Chapter 2, i.e.

$$R_{i,p}(\xi) = \frac{N_{i,p}(3/7)w_i}{\sum_{\hat{i}=1}^{n} N_{i,p}(3/7)w_{\hat{i}}} \quad . \tag{4.2}$$

Therefore, the distance moved from $C(3/7)$ to $C'(3/7)$ is about 50% of the distance between the original control point $B(5)$ and the new control point $B'(5)$ as shown in Figure 4.5 (b). The distance at any point on the curve is calculated in the same way, scilicet, the corresponding point of ξ on the new curve is obtained by translating the corresponding point of ξ on the original curve in the direction of vector $\vec{v}$ with a distance of $\left| R_{k,p}(\xi)\vec{v} \right|$.

Next, let us discuss the case of NURBS surfaces. For better illustrative on this point, we suppose a two-dimensional geometry was structured from the Equation (2.42), and the let control point $B_{k,h}$ be introduced uncertainty, i.e. the control point $B_{k,h}$ was moved to a new position $B_{k,h} + \vec{v}$, where $\vec{v}$ is randomly varying geometric vector used to represent uncertainty in shape. Since the new NURBS geometric structure was defined by $B_{1,1}, \cdots, B_{k,h} + \vec{v}, \cdots, B_{n,m}$, as follows:

$$S'(\xi, \eta) = \sum_{i=1}^{k-1}\sum_{j=1}^{h-1} R_{i,p}(\xi)R_{j,q}(\eta)B_{i,j} +$$

$$R_{k,p}(\xi)R_{h,q}(\eta)(B_{k,h}+\vec{v}) + \sum_{i=k+1}^{n}\sum_{j=h+1}^{m} R_{i,p}(\xi)R_{j,q}(\eta)B_{i,j}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,p}(\xi)R_{j,q}(\eta)B_{i,j} + R_{k,p}(\xi)R_{h,q}(\eta)(B_{k,h}+\vec{v})$$

$$= S(\xi, \eta) + R_{k,p}(\xi)R_{h,q}(\eta)\vec{v}. \tag{4.3}$$

It can be clearly known from Equation (4.3), only the *k,h*-th term used a control point containing uncertainty, $B_{k,h}+\vec{v}$, after regrouping, the new geometric structure is the sum of the original one and a $R_{k,p}(\xi)R_{h,q}(\eta)\vec{v}$. Furthermore, the randomly varying geometric vector $\vec{v}$ can be expressed by coordinates of the movement of control point

$$\vec{v} = (\hat{x}-x, \hat{y}-y), \tag{4.4}$$

where *x* and *y* are the control point coordinates before the shape change. $\hat{x}$ and $\hat{y}$ are the coordinates after the shape change. And its representation form regard with uncertainty can specifically be given by coordinate of the corresponding control points and a linear function with a random variable, ζ, as follows:

$$\hat{x} = x + L_x = x + \sum_{i=0}^{n} L_{x_i}\Psi(\zeta), \tag{4.5}$$

$$\hat{y} = y + L_y = y + \sum_{i=0}^{n} L_{y_i}\Psi(\zeta), \tag{4.6}$$

where $\Psi(\zeta)$ is the polynomial (Hermite polynomials) chaos with random variables. L_x and L_y are the deviation lengths in the *x*- and *y*-direction, respectively. In this study, the representation of uncertainty in shape is assumed to follow a normal distribution, such that the deviation lengths (L_x and L_y) can be rewritten as:

$$L_x = \sum_{i=0}^{n} L_{x_i} \Psi(\zeta) = L_x^{\mu} + L_x^{\sigma} \zeta, \tag{4.7}$$

$$L_y = \sum_{i=0}^{n} L_{y_i} \Psi(\zeta) = L_y^{\mu} + L_y^{\sigma} \zeta, \tag{4.8}$$

where $L_{x\,or\,y}^{\mu}$ and $L_{x\,or\,y}^{\sigma}$ are the mean and standard deviation, respectively, the subscripts *x* and *y* represent the *x*- and *y*-direction in the coordinate space. Figure 4.4 (b) shows an example of uncertainty representation with regard to geometrical shape. We assumed that the deformation of structure model occurs along the inner circular hole. In this example, the uncertainty in shape of the model was represented by introducing the uncertain parameter into control point 1, which is denoted by the hollow circle. Control point $\hat{1}$(red solid node) is the location of control point 1 after shape changed, and the bidirectional arrow (in black) indicates the deviation length (*L*). As shown in Figure 4.4 (b), we can see that the shape of the inner circular hole can be changed simply by moving control point 1, and without having to move other points. Additionally, the deviation length (*L*) is introduced directly into the coordinate of the control point in order to geometrically represent the uncertainty in shape.

We can know through the above discussion. In order to identify uncertainty of control points and can successfully represent this uncertainty in the actual geometry. We first made use of knot span to structure the mesh in the actual geometry. And then through affine invariance property of NURBS, we can know that the probability characteristics is not changed in the affine transformation from control points to actual geometry. We utilize the local modification scheme property of NURBS to determine the region of the actual domain where control point with uncertainty can influence. In addition, the degree of influence of the control point containing uncertainty on the actual geometry can be determined from both a geometric perspective and an algebraic one. The former viewpoint gives us insight and intuition that will prove invaluable in designing geometric models, identifying uncertainty, and a host of other activities related to stochastic isogeometric analysis. The latter

viewpoint is particularly useful in designing algorithms, and will be the setting in which we most frequently work. In the design algorithms, we use the connectivity arrays to link the control points, element and knots etc., in order to implement the introduction of uncertainty and exactly control uncertainty representation in the actual domain.

4.3. Preprocessing for formalization of stochastic isogeometric analysis (SIGA) method for uncertainty in shape

In this study, the SIGA framework was developed by utilising the intrusive method. The formulation of SIGA was defined by incorporating the PCE into the original IGA formulation framework. Therefore, Stiffness equation in original IGA formulation was redefined in the form of PCE. In this section, the formulation of SIGA is given with respect to a two-dimensional linear elasticity problem with a single random variable. Therefore, Before describing the stochastic isogeometric analysis method for uncertainty in shape, it is necessary to review the basic equations of linear elastic mechanics and the related energy principles.

4.3.1. Basic equation of linear elastic mechanics
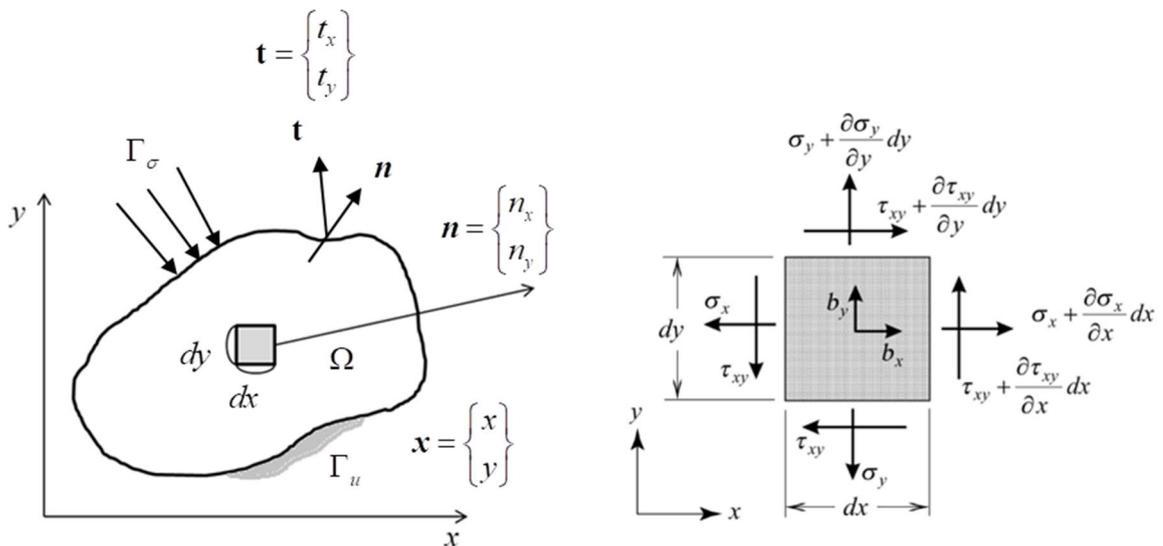
Figure 4.6 Model of an elastic body

It is considered that the number n_{sd} of space dimension of the problem is 2. A general point is denoted by $\boldsymbol{x}$. we will identify the point $\boldsymbol{x}$ with its position vector emanating. The unit outward normal vector to Γ is denoted by $\boldsymbol{n}$.

Firstly, the equilibrium equation for two-dimensional elastic problem is defined by

$$
\begin{aligned}
\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + F_x = \rho \frac{\partial^2 u}{\partial t^2}, \\
\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + F_y = \rho \frac{\partial^2 v}{\partial t^2},
\end{aligned}
\tag{4.9}
$$

where F_x and F_y are body forces. σ_x, σ_y and τ_{xy} are the only non-zero stress compoonents; u and v are the displacement component; $\rho \dfrac{\partial^2 u}{\partial t^2}$ and $\rho \dfrac{\partial^2 v}{\partial t^2}$ are the inertia forces, in the static balance, they are zero, that is

$$
\begin{aligned}
\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + F_x = 0, \\
\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + F_y = 0,
\end{aligned}
\tag{4.10}
$$

Secondly, geometric equation; The relationship between strain ε and displacement at any point in the elastomer is expressed as follows

$$
\begin{aligned}
\varepsilon_x &= \frac{\partial u}{\partial x}, \\
\varepsilon_y &= \frac{\partial v}{\partial y}, \\
\gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}.
\end{aligned}
\tag{4.11}
$$

Let $\dfrac{\partial u}{\partial x} = u_{,x}$ and $\dfrac{\partial u}{\partial y} = u_{,y}$, then the matrix form of Equation (4.11) is

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \tau_{xy} \end{bmatrix} = \partial \mathbf{u} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \end{bmatrix}. \tag{4.12}$$

The following relationship exists between the three strain components of Equation (4.11), called the deformation coordination condition, which is

$$\frac{\partial^2 \varepsilon_y}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial y^2} = \frac{\partial^2 \gamma_{xy}}{\partial x \partial y}. \tag{4.13}$$

Assuming no initial stress or strains, stresses σ are related to strains ε as follow

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \mathbf{D}\boldsymbol{\varepsilon} = \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & 1/2(1-v) \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \tau_{xy} \end{bmatrix}. \tag{4.14}$$

Where $\mathbf{D}$ is the constitutive matrix of plane stress, E is the elastic modulus and v is the Poisson's ratio. If the problem is in a state of plane strain, E is replaced by

$$\mathbf{D} = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1-v & v & 0 \\ v & 1-v & 0 \\ 0 & 0 & 1/2(1-2v) \end{bmatrix}. \tag{4.15}$$

In linear elastic problem, boundary conditions are classified into stress boundary conditions, displacement boundary conditions, or a mixture of the two. If the surface forces $\overline{F}_x$ and $\overline{F}_y$ are known, the stress boundary conditions can be written.

$$\begin{aligned} \overline{F}_x &= \sigma_x l + \tau_{yx} m, \\ \overline{F}_y &= \tau_{xy} l + \sigma_y m. \end{aligned} \tag{4.16}$$

If the displacement at the surface of the object is known, it is called the displacement boundary condition and can be written

$$u = \overline{u}, \quad v = \overline{v}, \tag{4.17}$$

Generally, for a correct solution to an elastic mechanics problem, all equations and boundary conditions of Equations (4.9) to (4.17) above should be satisfied. In this study we will also refer to these formulas for normalization.

4.3.2. Principle of Virtual Work

In the process of formalization, we need to use the principle of virtual work to construct the element matrix. Investigate the static equilibrium state of a deformed body under the action of external force. If the object is under the action of another new force system, a new deformation and displacement corresponding to the constraint condition is generated, which is called virtual displacement. The so-called "virtual" means that the displacement is not generated by the original force system. In this way, the original external force and internal force must work on the virtual displacement, which is called virtual work. According to the condition that the resultant force is zero when all the forces acting on any element in an elastic body are balanced, the condition that the sum of virtual work is zero is derived. Therefore, the definition of a virtual work principle for a suitable structural mechanics is as follows.

$$\iint \left(q_x u^* + q_y v^* \right) dS + \iiint \left(F_x u^* + F_y v^* \right) dV$$
$$= \iiint \sigma_x \varepsilon_x^* + \sigma_y \varepsilon_y^* + \sigma_{xy} \gamma_{xy}^* dV . \tag{4.18}$$

where u^* and v^* are principle of virtual displacements; $\varepsilon_x^*, \varepsilon_y^*$ and γ_{xy}^* are principle of virtual strain; The surface integral of the first term of the Equation (4.18) is the external force virtual work exerted by the surface force q_x and q_y on the surface S of the elastomer. The second term represents the external force virtual work by the volumetric force F_x and F_y; the right side of the formula represents the sum of the internal forces and virtual work inside the elastomer. According to Equation (4.18),

we can know that if the object is in equilibrium under the action of external force, the external force virtual work acting on the elastic body is equal to the internal force virtual work. For the convenience of application, the Equation (4.18) can also be written in a matrix form. The following matrix symbols are introduced

$$\{q\} = [q_x \; q_y]^T , \tag{4.19}$$

$$\{P\} = [F_x \; F_y]^T , \tag{4.20}$$

$$\{\delta^*\} = [u^* \; v^*]^T \tag{4.21}$$

$$\{\sigma\} = [\sigma_x \; \sigma_y \; \tau_{xy}]^T , \tag{4.22}$$

$$\{\varepsilon^*\} = [\varepsilon_x^* \; \varepsilon_y^* \; \gamma_{xy}^*]^T . \tag{4.23}$$

Then the formula (4.18) can be abbreviated as

$$\iint \{\delta^*\}^T \{q\} \, dS + \iiint \{\delta^*\}^T \{P\} \, dV = \iiint \{\varepsilon^*\}^T \{\sigma\} \, dV . \tag{4.24}$$

4.3.3. Structural Domain

Firstly, we need to define a domain posed over a NURBS geometry to be calculated in the linear static elastic problem. In this thesis, we use Ω to represent a whole computational domain. If the computational domain is composed of multiple patches, the corner mark P, i will be added, i.e. $\Omega^{P,i}$ where i is the patch number. The entire computational domain for multiple patches is represented by

$$\Omega_{num}^{M} = \sum_{i=1}^{n=num} \Omega^{P,i}, \tag{4.25}$$

where subscript *num* is the total number of patches. specifically, as shown in Figure 4.7.
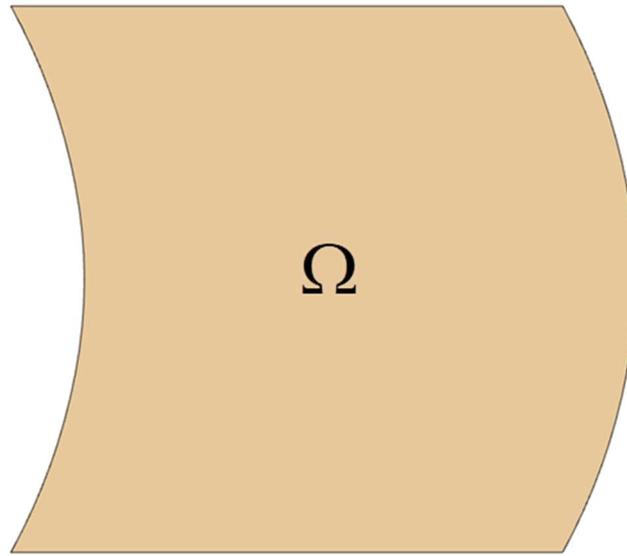
(a) A computational domain defined by single patch
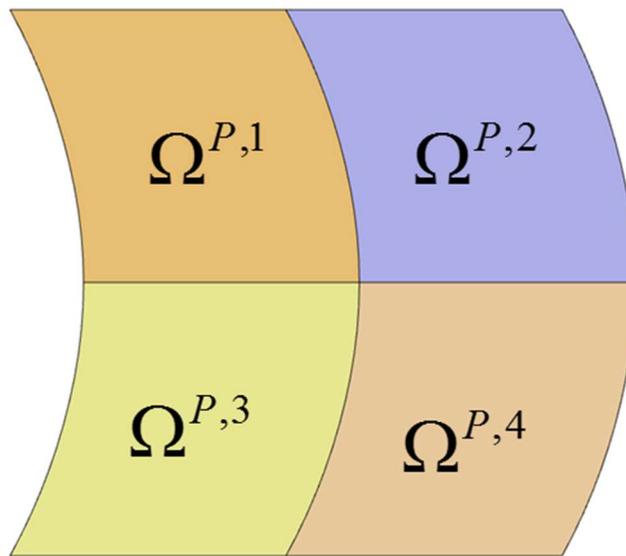
(b) A computational domain consisting of multiple patches.

Figure 4.7 computational domain posed over a NURBS geometry

In addition, the smallest computational domain on the NURBS geometry entity (i.e element defined by physical mesh) can be marked as $\Omega_{E,j}$, where subscript E is the meaning of the element, and j represents the number of elements. If the computational domain is composed of multiple patches, the element is represented as $\Omega_{E,j}^{P,i}$. Therefore, the relationships between different computational domains are

$$\Omega_{num}^{M} = \sum_{i=1}^{n=num} \Omega^{P,i} = \sum_{i=1}^{n=num} \sum_{j=1}^{m=eum} \Omega_{E,j}^{P,i} \tag{4.25}$$

where *eum* is the total number of elements in each patch. An example of element partitioning over a NURBS geometry entity is shown in Figure 4.8. These elements are divided from the knot vector $\{\xi_{p+n+1}\}$ and $\{\eta_{p+n+1}\}$ in the $\xi-$ and $\eta-$ direction, respectively.
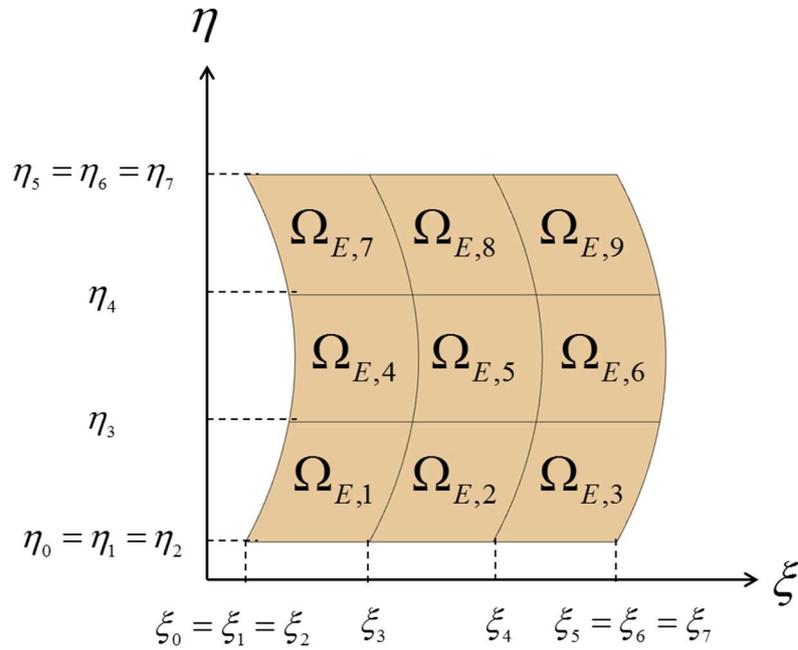
(a) The notation $\Omega_{E,j}; j = 1, 2 \cdots 9$ is used to represent elements defined by span in a single patch.
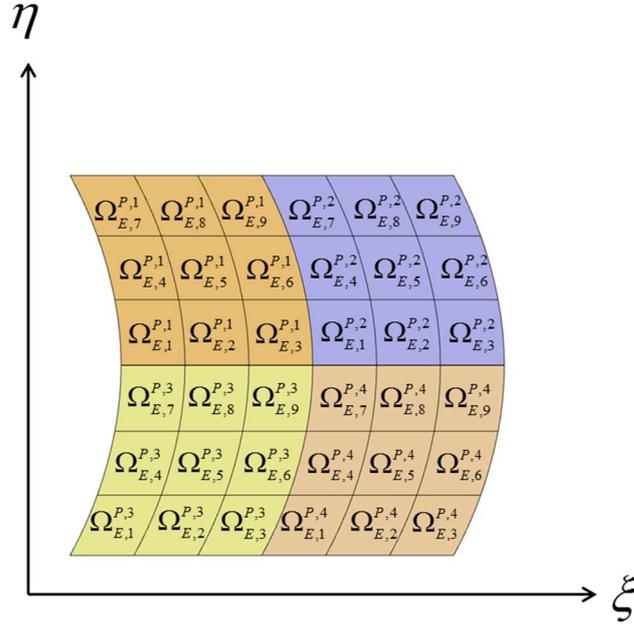
(b) The elements in multiple patches are represented by symbol $\Omega_{E,j}^{P,i}$, where

$$i = 1, \cdots, 4; \; j = 1, 2 \cdots 9;$$

Figure 4.8 Elements are marked in single patch and multiple patches, respectively.

NURBS geometric entity actually has two coordinate systems, one is homogeneous coordinate and the other is parametric coordinate. homogeneous coordinates are standard spatial coordinates, and parametric coordinates conform to geometric entity. Therefore, the boundaries of NURBS geometric entities are defined by parametric coordinates. As shown in Figure 4.9, we want to define the boundary on the right side of the NURBS geometry, ie, the BD sides. Then the boundary BD is defined by the parameter coordinates $\{\xi, \eta\}$ as follows

$$\xi = \xi_{p+n+1} \; , \; \; \eta_1 \le \eta \le \eta_{q+m+1} \tag{4.26}$$

By Equation 2.42, boundary BD on geometric entities is defined by

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta)\mathbf{B}^{i,j}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} 1 \times M_{j,q}(\eta)\mathbf{B}^{i,j} \qquad (4.27)$$

$$= \mathbf{S}(\xi).$$
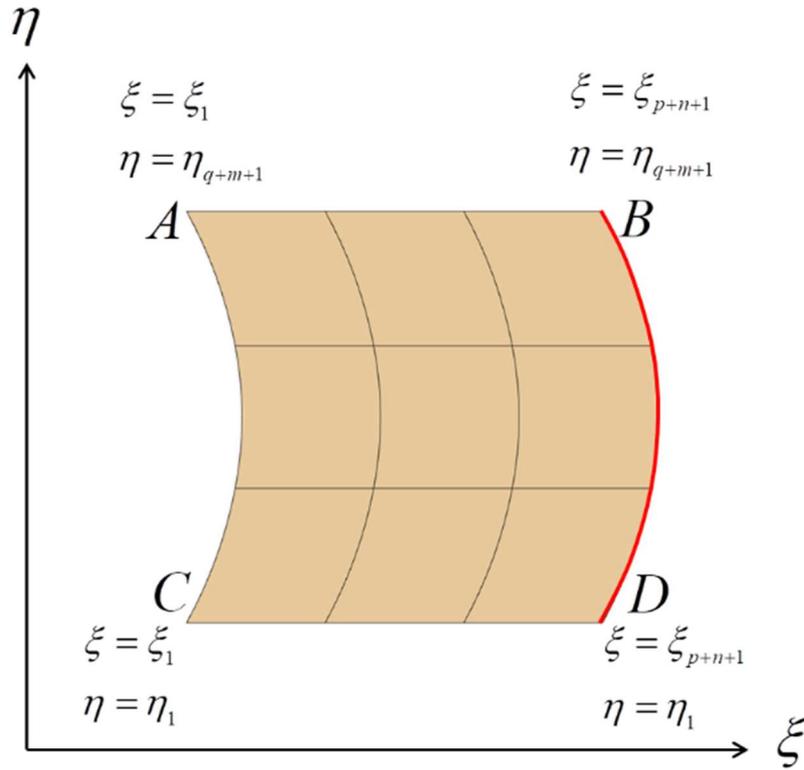
Figure 4.9 The boundary BD on geometric entities

By Equation 4.27, we can know that the two-dimensional surface equation $\mathbf{S}(\xi,\eta)$ eventually becomes a formula with a single variable. Applying the same logic to the remaining three boundaries of NURBS entity, the boundary CD is

$$\xi_1 \le \xi \le \xi_{p+n+1}, \quad \eta = \eta_1 \qquad (4.28)$$

with

$$\mathbf{S}(\xi,\eta) = \mathbf{S}(\xi) \qquad (4.29)$$

The boundary AC

$$\xi = \xi_1, \quad \eta_1 \le \eta \le \eta_{q+m+1} \qquad (4.30)$$

$$\mathbf{S}(\xi,\eta) = \mathbf{S}(\eta) \qquad (4.31)$$

and boundary AB is defined by

$$\xi_1 \le \xi \le \xi_{p+n+1}, \quad \eta = \eta_{q+m+1} \qquad (4.32)$$

$$\mathbf{S}(\xi,\eta) = \mathbf{S}(\xi) \qquad (4.33)$$

In Figure 4.10, we give a more detailed example.
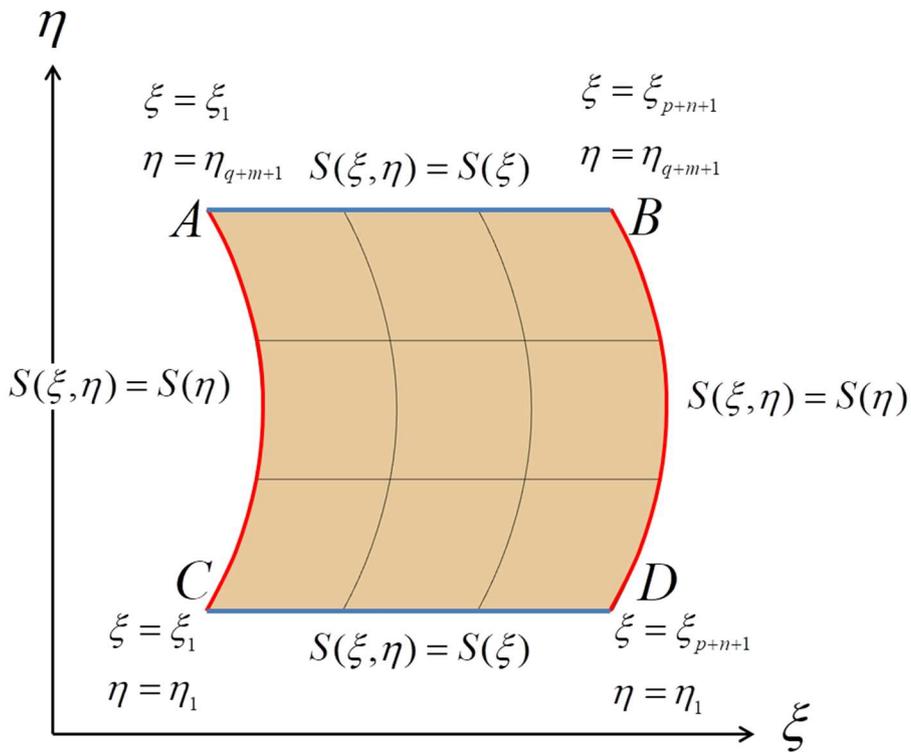
Figure 4.10 Geometric boundary on NURBS entity

4.3.4. Boundary value problems (BVPs)

In the linear static elasticity problem, the domain to be calculated is defined as Ω, and the boundary of the domain is set as Γ (specifically, see Figure 4.11). However, the displacement constraint condition which is the boundary condition of the displacement is represented by Γ_g and the load condition which is the boundary condition of the load is indicated by the symbol Γ_h.
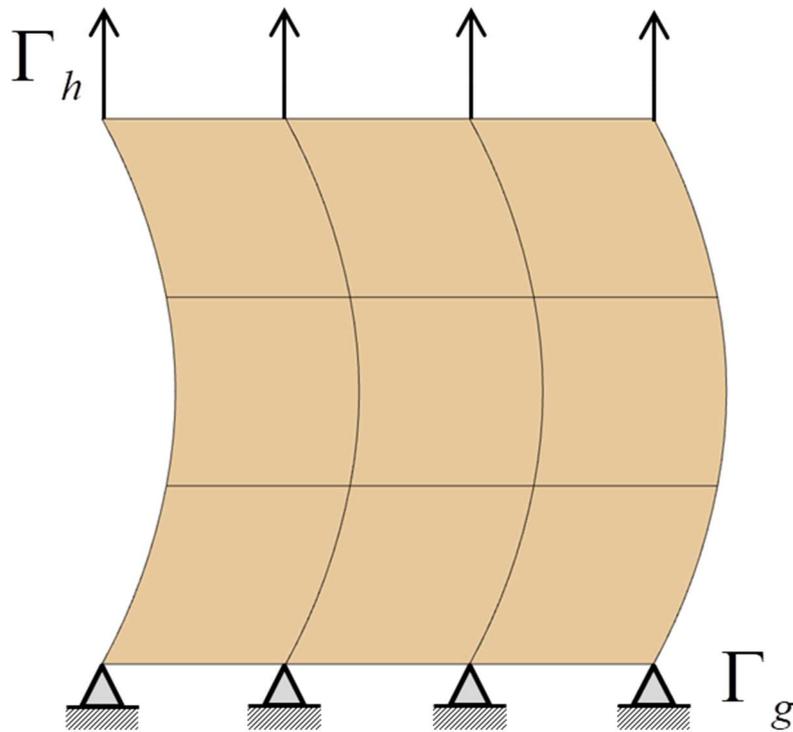
Figure 4.11 The boundary condition for load and displacement.

When the region is two-dimensional, the stress is defined as follows (planar stress state is considered)

$$\begin{Bmatrix} \sigma_x & \tau_{xy} \\ \tau_{yx} & \sigma_y \end{Bmatrix} = \begin{Bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{Bmatrix} \tag{4.34}$$

Displacement and volume force are expressed as follows.

$$\mathbf{u} = \{u_1 \quad u_2\} \tag{4.35}$$

$$\mathbf{f} = \{f_1 \quad f_2\} \tag{4.36}$$

By Hooke's Law, the following equations can be defined.

$$\sigma_{ij} = c_{ijkl} u_{kl} \tag{4.37}$$

Based on equilibrium conditional equations of stress

$$\sigma_{ij,j} + f_i = 0 \quad (in\ \Omega) \tag{4.38}$$

The boundary conditions of displacement and load are defined by

$$u_i = g_i \quad (on\ \Gamma_{g_i}) \tag{4.39}$$

$$u_i = g_i \quad (on\ \Gamma_{g_i}) \tag{4.40}$$

The corresponding vector form is

$$\mathbf{g} = \{g_1 \quad g_2\} \tag{4.41}$$

$$\mathbf{h} = \{h_1 \quad h_2\} \tag{4.42}$$

4.3.5. Establish weak formulation

In section 4.3.3, the strong form for the 2D linear elasticity is derived. In this section, we will derive the weak form by the strong form.

Here, the strong form in the x-direction, $\dfrac{\partial \sigma_x}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + b_x = 0$, is multiplied by an arbitrary weight function $w_x(x, y)$ and integrated over the domain:

$$\int_\Omega w_x \frac{\partial \sigma_x}{\partial x} d\Omega + \int_\Omega w_x \frac{\partial \tau_{xy}}{\partial y} d\Omega + \int_\Omega w_x b_x d\Omega = 0. \qquad (4.43)$$

We expand Equation (4.43), and the Equation (4.44) can be obtained as show following.

$$\int_\Gamma w_x \sigma_x n_x d\Gamma - \int_\Omega \frac{\partial w_x}{\partial x} \sigma_x d\Omega + \int_\Gamma w_x \tau_{xy} n_y d\Gamma - \int_\Omega \frac{\partial w_x}{\partial y} \tau_{xy} d\Omega + \int_\Omega w_x b_x d\Omega = 0. \quad (4.44)$$

The *x*-component of the traction vector, **t**, was given by following equation.

$$\mathbf{t} = \begin{Bmatrix} t_x \\ t_y \end{Bmatrix} = \begin{Bmatrix} \sigma_x n_x + \sigma_{xy} n_y \\ \sigma_{xy} n_x + \sigma_y n_y \end{Bmatrix}. \qquad (4.45)$$

This makes it possible to write the above equation as.

$$\int_\Gamma w_x t_x d\Gamma - \left(\int_\Omega \frac{\partial w_x}{\partial x} \sigma_x + \int_\Omega \frac{\partial w_x}{\partial y} \tau_{xy} \right) d\Omega + \int_\Omega w_x b_x d\Omega = 0. \qquad (4.46)$$

Analogous manipulations are carried out for the strong form in the *y*-direction, and hereby the equation for y-directions is given:

$$\int_\Gamma w_y t_y d\Gamma - \left(\int_\Omega \frac{\partial w_y}{\partial y} \sigma_y + \int_\Omega \frac{\partial w_y}{\partial x} \tau_{xy} \right) d\Omega + \int_\Omega w_y b_y d\Omega = 0. \qquad (4.47)$$

We can obtain a new equation by addition of the two equations.

$$\int_\Gamma \left(w_x t_x + w_y t_y \right) d\Gamma + \int_\Omega \left(w_x b_x + w_y b_y \right) d\Omega$$
$$- \left(\int_\Omega \frac{\partial w_x}{\partial x} \sigma_x + \int_\Omega \frac{\partial w_x}{\partial y} \tau_{xy} + \int_\Omega \frac{\partial w_y}{\partial y} \sigma_y + \int_\Omega \frac{\partial w_y}{\partial x} \tau_{xy} \right) d\Omega = 0. \qquad (4.48)$$

We can use the following vectors.

$$\boldsymbol{b} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix}, \quad \boldsymbol{w} = \begin{Bmatrix} w_x \\ w_y \end{Bmatrix}, \quad \nabla \boldsymbol{w} = \begin{Bmatrix} \dfrac{\partial w_x}{\partial x} \\ \dfrac{\partial w_y}{\partial y} \\ \dfrac{\partial w_x}{\partial x} + \dfrac{\partial w_y}{\partial y} \end{Bmatrix}, \quad \boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_x \\ \tau_{xy} \end{Bmatrix}. \tag{4.49}$$

The Equation (4.4) can be written as the weak form of the equilibrium equations in matrix notation:

$$\int_\Omega (\nabla w)^{\mathrm{T}} \sigma d\Omega = \int_\Gamma w^{\mathrm{T}} t d\Gamma + \int_\Omega w^{\mathrm{T}} b d\Omega, \tag{4.50}$$

where, the term $\int_\Gamma w^{\mathrm{T}} t d\Gamma$ is boundary loads, the term $\int_\Omega w^{\mathrm{T}} b d\Omega$ is volume loads.

Here, since Equation (4.50) is cumbersome, it is rewritten in an easy-to-understand manner using matrix notation. For the left side of the Equation (4.50), we have

$$\int_\Omega (\nabla \mathbf{w})^{\mathrm{T}} \sigma d\Omega = \int_\Omega \varepsilon (\mathbf{w})^T \cdot \mathbf{D} \, \varepsilon (\mathbf{u}) d\Omega$$
$$= a(\mathbf{w}, \mathbf{u}). \tag{4.51}$$

The right side of the equation (4.50) is rewritten to

$$\int_\Omega \mathbf{w}^{\mathrm{T}} \mathbf{b} \, d\Omega = (\mathbf{w}, \mathbf{f}) \tag{4.52}$$

$$\int_\Gamma \mathbf{w}^{\mathrm{T}} \mathbf{t} d\Gamma = (\mathbf{w}, \mathbf{h})_\Gamma \tag{4.53}$$

The final Equation (4.50) is simplified to

$$a(\mathbf{w}, \mathbf{u}) = (\mathbf{w}, \mathbf{f}) + (\mathbf{w}, \mathbf{h})_\Gamma \tag{4.54}$$

In later chapters, we usually use Equation (4.54) to define the matrix form.

4.3.6. Galerkin approximation

In this section, we will use Galerkin approximation for weak forms. In order to make it easier to understand, we first give the approximate form of each function and the weak form of the approximate form.

Approximation of displacement:

$$\mathbf{u} \approx \mathbf{u}^h = \mathbf{v}^h + \mathbf{g}^h \ . \tag{4.55}$$

Approximation of weight:

$$\mathbf{w} \approx \mathbf{w}^h = \{w^h{}_1, w^h{}_2, w^h{}_3\}. \tag{4.56}$$

where, $\mathbf{g}^h$ is defined according to boundary conditions. $\mathbf{v}^h$ is the item which is need solved. $\mathbf{g}^h$ and $\mathbf{v}^h$ are given as following.

$$\mathbf{v}^h = \{v^h{}_1, v^h{}_2, v^h{}_3\}, \tag{4.57}$$

$$\mathbf{g}^h = \{g^h{}_1, g^h{}_2, g^h{}_3\}. \tag{4.58}$$

Thus, we can be approximated weak form by using Equation (4.57) and (4.58).

$$a(\mathbf{w}, \mathbf{u}) = (\mathbf{w}, \mathbf{f}) + (\mathbf{w}, \mathbf{h})_\Gamma \ , \tag{4.59}$$

$$a(\mathbf{w}^h, \mathbf{v}^h) = (\mathbf{w}^h, \mathbf{f}) + (\mathbf{w}^h, \mathbf{h})_\Gamma - a(\mathbf{w}^h, \mathbf{g}^h). \tag{4.60}$$

In the following sections, we will elaborate on the approximation methods of each function.

4.3.7. Approximation of displacement $\mathbf{u} \approx \mathbf{u}^h$

The displacement is approximated by using shape function. Where, we use n_{sd} to represent the spatial dimension of the object domain. When $n_{sd} = 2$, it is considered as a plane stress problem.

Displacement:

$$\mathbf{u} = \{u_1, u_2\}, \tag{4.61}$$

Body Forces:

$$\mathbf{f} = \{f_1, f_2\}, \tag{4.62}$$

Surface Force:

$$\mathbf{h} = \{h_1, h_2\}, \tag{4.63}$$

Virtual Displacement:

$$\mathbf{w} = \{w_1, w_2\}. \tag{4.64}$$

Then, Approximation of displacement $\mathbf{u}$ can be shown in following.

$$\mathbf{u}^h = \{u^h_1, u^h_2\}. \tag{4.65}$$

The $\mathbf{u}^h$ can be presented as following by using B-spline basis function.

$$u_i \approx u_i^h = \sum_{A \in \varphi} R_A d_{A,i}, \tag{4.66}$$

where, the R_A is shape function, $d_{A,i}$ is the degree of freedom corresponding to the control point of B-spline. A is the DOF number corresponding to the control point. φ is a set of degrees of freedom corresponding to all control points of domain Ω. We can find that we can define u_i^h when all $d_{A,i}$ constituting domain Ω is given.

4.3.8. Definition of $\mathbf{g}^h$, $\mathbf{v}^h$

As shown in the boundaries assumed in Section 4.3.3, the displacement constraints are expressed by the following equations bases on $\mathbf{u} \approx \mathbf{u}^h$.

$$u_i = g_{i(on\,\Gamma_{g_i})}. \tag{4.67}$$

It can be written in the following form.

$$u^h_{\,i} = g_{i(on\,\,\Gamma_{g_i})}. \tag{4.68}$$

When the following functions Ψ_i are considered, we have:

$$\Psi_i = \int_{\Gamma_{g_i}} (u_i - g_i)^2 d\Gamma . \tag{4.69}$$

When $\Psi_i = 0$, we can find that displacement constraints can be fully satisfied. And when Ψ_i is very small, the conditions close to displacement constraints can also be applied to u_i. Thus the Equation (4.69) can be transformed into the following forms.

$$\begin{aligned}
\Psi_i &= \int_{\Gamma_{g_i}} (u_i - g_i)^2 d\Gamma \\
&= \int_{\Gamma_{g_i}} u_i^2 - 2u_i g_i + g_i^2 d\Gamma \\
&= \int_{\Gamma_{g_i}} \left\{ \left(\sum_{A \in \varphi_{g_i}} (R_A\, d_{i,A}) \right)^2 - 2\sum_{A \in \varphi_{g_i}} (R_A\, d_{i,A})g_i + g_i^2 \right\} d\Gamma.
\end{aligned} \tag{4.70}$$

For $d_{i,B}$ whose DOF number is B, we consider Ψ_i to be the smallest.

$$\frac{\partial \Psi_i}{\partial d_{i,B}} = 0. \tag{4.71}$$

According to Equation (4.70), we have:

$$\frac{\partial \Psi_i}{\partial d_{i,B}} = \int_{\Gamma_{gi}} \{2 \sum_{A \in \varphi_{g_i}} (R_A d_{i,A}) R_B - 2 R_B g_i\} d\Gamma$$

$$= 2 \sum_{A \in \varphi_{g_i}} <R_A R_B> d_{i,A} - 2 <R_B g_i>,$$

(4.72)

where, $\int_{\Gamma_{gi}} R_A d\Gamma = <R_A>$, $\int_{\Gamma_{gi}} g_i d\Gamma = <g_i>$, $\int_{\Gamma_{gi}} R_A g_i d\Gamma = <R_A g_i>_{(\because d_{i,A} \text{は定数})}$.

Thun, the $d_{i,B}$ should satisfy the following conditions.

$$\sum_{A \in \varphi_{g_i}} <R_A R_B> d_{i,A} = <R_B g_i>.$$

(4.73)

The set of degree of freedom corresponding to the control point of boundary Γ_{gi} is expressed by φ_{gi}. When the number of degrees of freedom is $n^{g_i node}$, it is equivalent to the displacement constraint in the whole range of $B \in \varphi_{gi}$. The matrix can be written in the following form.

$$\begin{pmatrix} <R_1 R_1> & <R_1 R_2> & \dots & <R_1 R_{n^{g_i node}}> \\ <R_2 R_1> & <R_2 R_2> & \dots & <R_2 R_{n^{g_i node}}> \\ & & \dots & \\ <R_{n^{g_i node}} R_1> & <R_{n^{g_i node}} R_2> & \dots & <R_{n^{g_i node}} R_{n^{g_i node}}> \end{pmatrix} \begin{pmatrix} d_{i,1} \\ d_{i,2} \\ \\ d_{i,n^{g_i node}} \end{pmatrix} = \begin{pmatrix} R_1 g_i \\ R_2 g_i \\ \\ R_{n^{g_i node}} g_i \end{pmatrix}. \quad (4.74)$$

We can obtain the value of $\left(d_{i,1}, d_{i,2}, \dots, d_{i,n^{g_i node}}\right)^T$ by solving the Equation (4.74). And in order to distinguish, we use $\left(d^g_{i,1}, d^g_{i,2}, \dots, d^g_{i,n^{g_i node}}\right)^T$ form.

Especially when the displacement is completely constrained ($u^h_i = g_i = 0_{(on \ \Gamma_{gi})}$), Eq.(4.74) is considered as the following form, and we can find that it is completely satisfied with Form $\Psi_i = 0$.

$$\left(d_{i,1}, d_{i,2}, \dots, d_{i,n^{g_i node}}\right)^T = \mathbf{0}.$$

(4.75)

As a result, we can judge from the constraints that it only needs to satisfy Equation (4.75).

According to the above conclusion, we can decompose the displacement into "the part solved according to the boundary conditions g_i^h" and "the part of freedom that should be solved v_i^h". It can be written in the following form.

$$
\begin{aligned}
u_i^h &= \sum_{A\in\varphi-\varphi_{gi}} R_A d_{i,A} + \sum_{A\in\varphi_{gi}} R_A \; d_{i,A}^g \\
&= v_i^h + g_i^h,
\end{aligned}
\tag{4.76}
$$

where, φ is the set of degrees of freedom corresponding to the control points constituting domain Ω. φ_{gi} is the set of degrees of freedom corresponding to the control points on boundary Γ_{gi}.

4.3.9. Approximation of Virtual Displacement $\mathbf{w} \approx \mathbf{w}^h$

In the same way as displacement $\mathbf{u}$, we use B-spline basis function to approximate the virtual displacement $\mathbf{w}$.

$$
\mathbf{w} \approx \mathbf{w}^h = \{w^h{}_1, w^h{}_2, w^h{}_3\}.
\tag{by 4.56}
$$

When we use φ_{gi}, we have:

$$
\begin{aligned}
w_i^h &= \sum_{A\in\varphi-\varphi_{gi}} R_A c_{i,A} + \sum_{A\in\varphi_{gi}} R_A \times 0 \\
&= \sum_{A\in\varphi-\varphi_{gi}} R_A c_{i,A},
\end{aligned}
\tag{4.77}
$$

where, $c_{i,A}$ is a constant. And it satisfies the following equation.

$$
w^h{}_i = 0_{(on\ \ \Gamma_{gi})}.
\tag{4.78}
$$

4.3.10. Space transformation

In Chapter 2, we have introduced the concepts and definitions of parameter, physical,

index and parent space in detail, and given the relationships that exist between each (see figure 2.5). In this section, in order to better understand the discretization process for the computational domains, we will briefly introduce how the relationships between these spaces are constructed. Usually, we connect these spaces through mapping (i.e. changing the variables and coordinate system). In conventional SFEM, the elements divided by FE-mesh in the physical space is directly mapped to the parent space, as shown in Figure 4.12. The parent space is an uniformized parent elements and parent element has its own mapping from the element in FE-mesh.
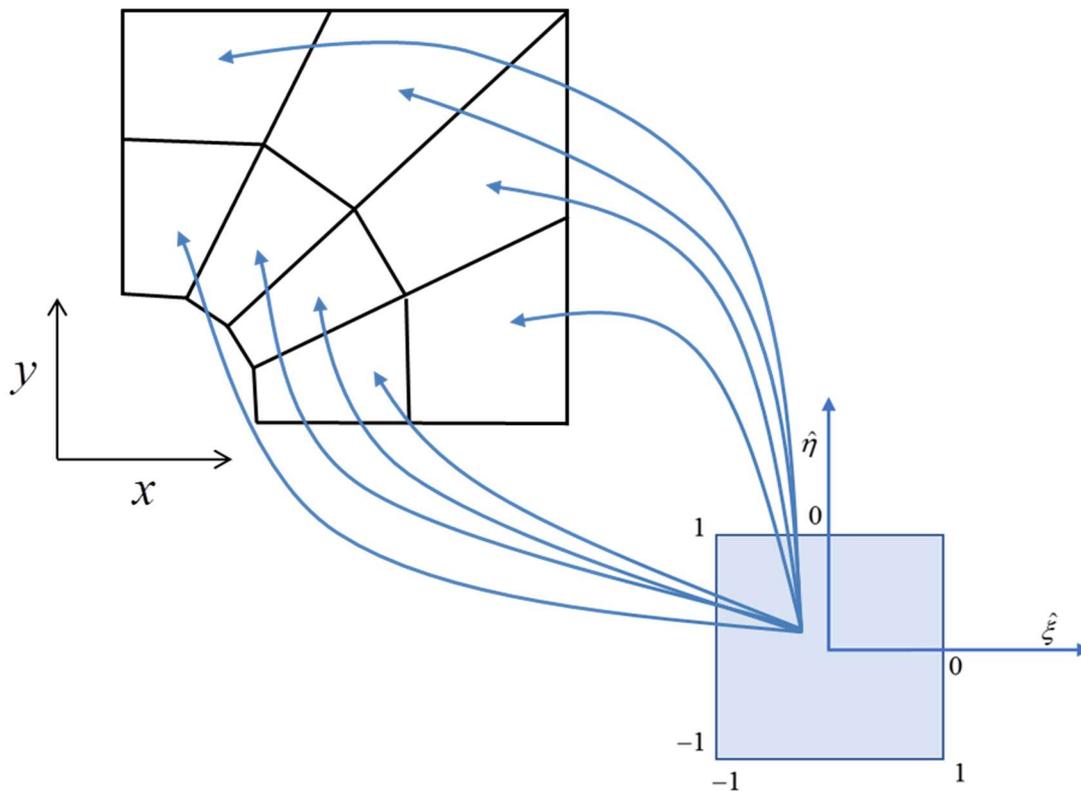
Figure 4.12 the mapping is performed from the parent space to element in FE-mesh

In the SIGA, the discretization of the computational domain located on the NURBS geometric entity is performed using the knot vector in the parameter space. Moreover, the elements partitioned by the physical mesh in the physical space are in one-to-one correspondence with the knot spans in the parameter space (specifically, refer to Figure 4.13). For the NURBS entity constructed by multiple patch, each patch has its own separate parameter space. The definition of multi-patch mapping is shown in Figure 4.14.

when the domain Ω is surrounded by two functions $\psi_1(u)$, $\psi_2(u)$, using the mapping function definition domain is the most simple to define transfinite mappings.

$$P(u,v) = (1-v)\psi_1(u) + v\psi_2(u), \tag{4.79}$$

where, for the random variable u, v, it needs to satisfy the following conditions.

$$|u| \leq 1, |v| \leq 1. \tag{4.80}$$

Ultimately, we can use Equation (4.79) to define a multiple patch domain.
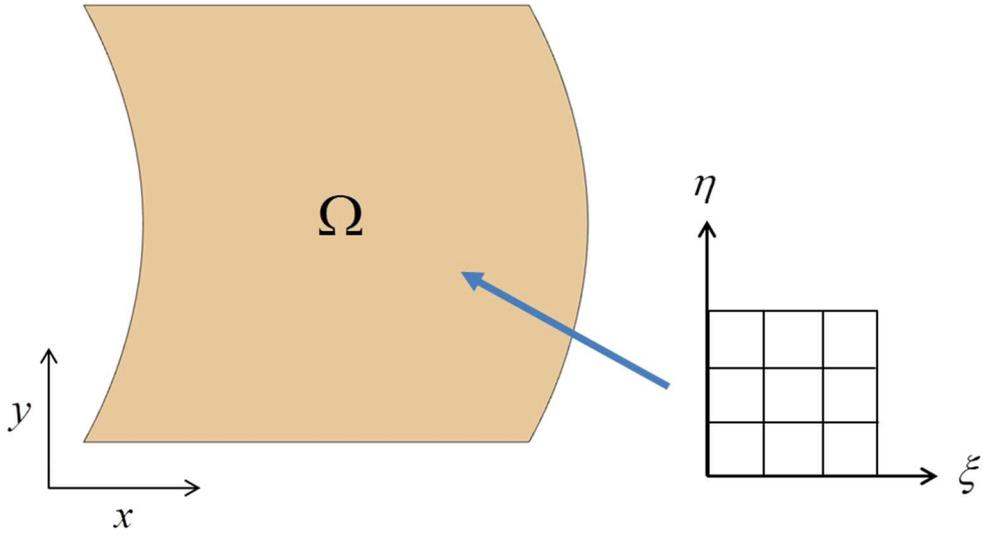
(a) A single B-spline map takes the patch from the parameter space to the physical space.
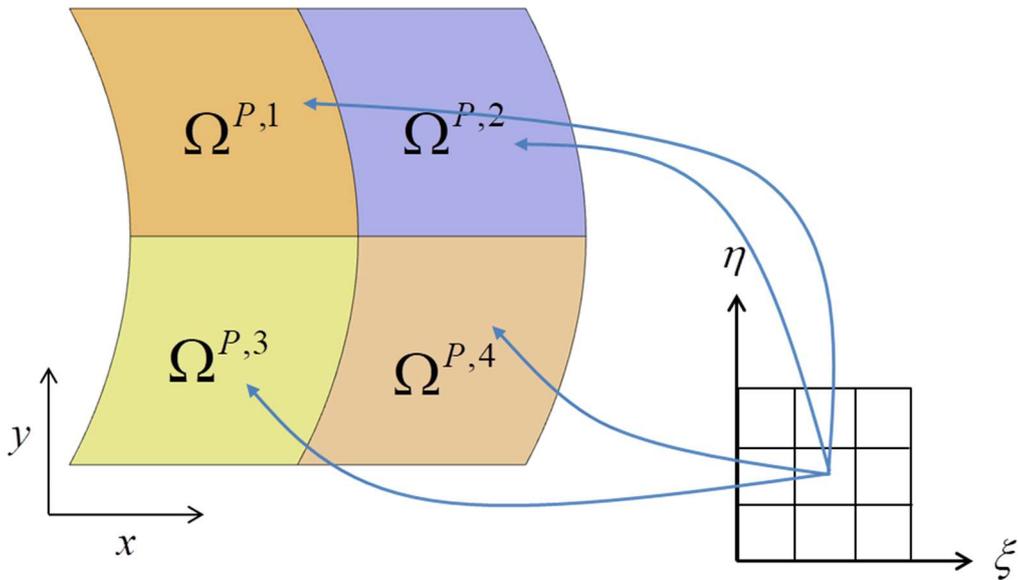
(b) The multi- patch NURBS entity possess multiple separate parameter space

Figure 4.13 the NURBS geometric entity is discretized by the parameter space

From the above discussion, we can see that in the process of discretization of NURBS geometric entities using SIGA method, an additional space needs to be considered, that is, parameter space, which does not exist in classical FEM method. Therefore, in SIGA framework, three space transformations need to be considered, i.e., an affine mapping from the parent space to the parameter space ($f : \tilde{\Omega} \rightarrow \hat{\Omega}$); a geometrical mapping from the parameter space to the physical space ($g : \hat{\Omega} \rightarrow \Omega$). Then, these two mappings are combined to constitute space mapping from the parent space into the physical space ($\gamma : f \circ g$). The mapping processes in the SIGA are illustrated in Figure 4.14. For a more detailed and thorough discussion of these space concepts, see [4].
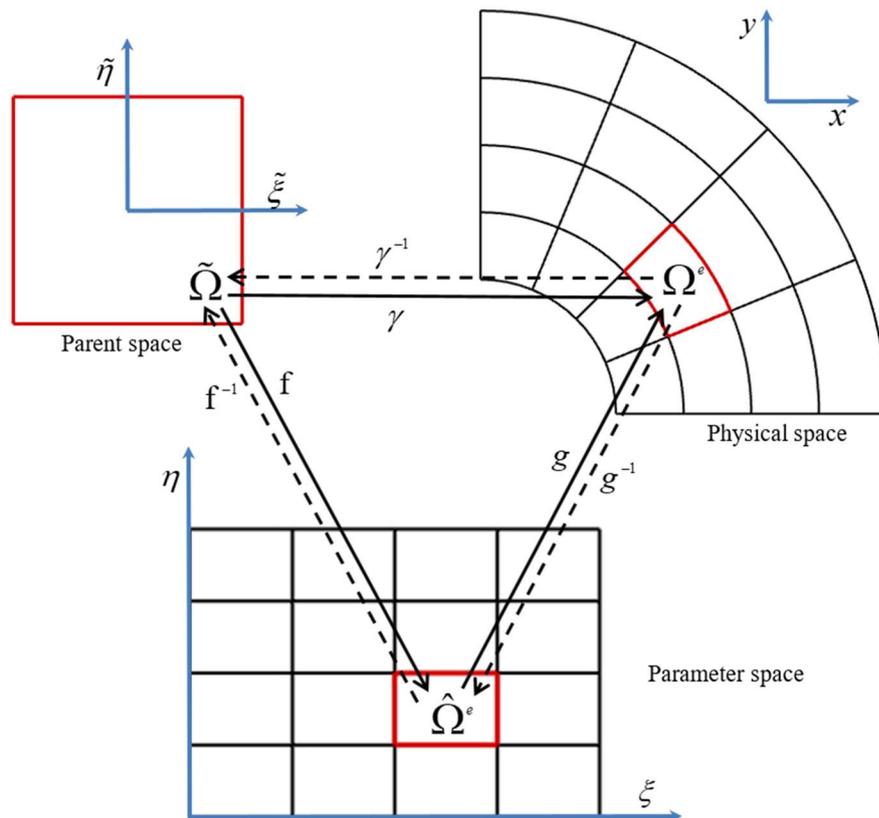
Figure 4.14 Diagram of mappings in analysis process. Dashed line with arrowhead denotes the inverses of the mappings.

4.4. Formulation of Stochastic Isogeometric Analysis For Uncertainty In Shape

In this study, the SIGA framework was developed by utilising the intrusive method. The formulation of SIGA was defined by incorporating the PCE into the original IGA formulation framework. Therefore, Stiffness equation in original IGA formulation was redefined in the form of PCE. In this section, the formulation of SIGA based on NURBS is given with respect to a two-dimensional linear elasticity problem with a single random variable.

For the sake of actual calculation, we convert the weak form shown in Equation 4.54 into a matrix form.

First of all, the unit vector $\mathbf{e}_i$ for two-dimension is defined as follows (this unit vector will be used without further notice).

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.81}$$

Through this unit vector, each function (vector form) can be rewritten as follows according to the summation rule.

$$\mathbf{w}^h = w^h_{\ i} \mathbf{e}_i, \tag{4.82}$$

$$\mathbf{v}^h = v^h_{\ i} \mathbf{e}_i, \tag{4.83}$$

$$\mathbf{g}^h = g^h_{\ i} \mathbf{e}_i. \tag{4.84}$$

As shown in Section 4.3.7, each component of the virtual displacement $\mathbf{w}^h$ is expressed as follows.

$$w_i^{\ h} = \sum_{A \in \varphi - \varphi_{g_i}} R_A c_{i,A}. \tag{4.85}$$

Further, $\mathbf{v}^h$ and $\mathbf{g}^h$ can be rewritten as

$$v_i^h = \sum_{A \in \phi - \phi_{g_i}} R_A \cdot d_{i,A},$$
(4.86)

$$g_i^h = \sum_{A \in \phi_{g_i}} R_A \cdot d_{i,A}^g,$$
(4.87)

where ϕ is the set of degrees of freedom corresponding to the control points constituting the computational domain; ϕ_{gi} is a set of degrees of freedom corresponding to the control points related to the boundary condition on the boundary Γ_{gi}. From the above, the Equation (4.54) can be transformed as follows by the Equations (4.85), (4.86) and (4.87).

$$\sum_{j=1}^{n_{sd}} (\sum_{B \in \phi - \phi_{g_j}} a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) d_{j,B}) = (R_A \mathbf{e}_i, \mathbf{f}) + (R_A \mathbf{e}_i, \mathbf{h})_{\Gamma}$$

$$-\sum_{j=1}^{n_{sd}} (\sum_{B \in \phi_{g_i}} a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) d_{j,B}^g).$$
(4.88)

Ultimately, the matrix equation can be constructed as follows:

$$\mathbf{Kd} = \mathbf{F},$$
(4.89)

where

$$\mathbf{K} = a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) = \mathbf{e}_i^T \int_{\Omega} \mathbf{B}_A^T \mathbf{D} \mathbf{B}_B d\Omega \, \mathbf{e}_j,$$
(4.90)

$$\mathbf{F} = (R_A \mathbf{e}_i, \mathbf{f}) + (R_A \mathbf{e}_i, \mathbf{h})_{\Gamma} - \sum_{j=1}^{nsd} (\sum_{B \in \phi_{g_j}} a(R_A \mathbf{e}_i, R_B \mathbf{e}_j) d_{j,B}^g),$$
(4.91)

and **K** is the global stiffness matrix, **F** is the global force vector, and **d** is the unknown displacement response in this formulation. Note that global stiffness equation is defined in the whole computational domains Ω.

We can apply the same logic to define patch stiffness equation. The patch matrix form will be used when the computational domain composed of multiple patches is analyzed. The patch stiffness equation in computational domain $\Omega^{P,i}$ can be defined by

$$\mathbf{k}^P \mathbf{d}^P = \mathbf{f}^P, \tag{4.92}$$

where $\mathbf{k}^P$ is the patch stiffness matrix, $\mathbf{f}^P$ is the patch force vector, and $\mathbf{d}^P$ is the unknown displacement response in this formulation. Note that A single patch can be considered as a special form of multiple patches. Finally, the patch analysis system is assembled from the local stiffness matrices k_e and force vectors f_e established over the elements Ω_E.

In the SIGA formulation, the local stiffness matrix containing the random variables ζ is expressed as follows:

$$k_e(\zeta) = \int_{\Omega^e} \boldsymbol{B}^T(\zeta) \boldsymbol{D} \boldsymbol{B}(\zeta) |\boldsymbol{J}(\zeta)| d\Omega^e \tag{4.93}$$

where $\boldsymbol{B}(\zeta)$ and $\boldsymbol{J}(\zeta)$ are the strain matrix and the Jacobian matrix which involve random variable ζ, respectively, **D** is the material property matrix.

We have given the definition of each spatial transformation in Section 4.3.9, and the mapping between them is achieved by using the Jacobian determinant. Therefore, in the SIGA. the mapping relationship between the coordinate system $[x, y]$ in the physical space and the coordinate system $[\xi, \eta]$ in the parameter space can be defined by the Jacobian determinant including the random variable, as follows

$$|\boldsymbol{J}(\zeta)| = \begin{vmatrix} \dfrac{\partial \hat{x}}{\partial \xi} \dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial \hat{x}}{\partial \eta} \dfrac{\partial \eta}{\partial \tilde{\eta}} \\[4mm] \dfrac{\partial \hat{y}}{\partial \xi} \dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial \hat{y}}{\partial \eta} \dfrac{\partial \eta}{\partial \tilde{\eta}} \end{vmatrix}$$

$$= \begin{vmatrix} \dfrac{\partial}{\partial \xi}\left[x + \sum_{i=0}^{n} L_{x_i} \Psi(\zeta) \right]\dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial}{\partial \eta}\left[x + \sum_{i=0}^{n} L_{x_i} \Psi(\zeta) \right]\dfrac{\partial \eta}{\partial \tilde{\eta}} \\[4mm] \dfrac{\partial}{\partial \xi}\left[y + \sum_{i=0}^{n} L_{y_i} \Psi(\zeta) \right]\dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial}{\partial \eta}\left[y + \sum_{i=0}^{n} L_{y_i} \Psi(\zeta) \right]\dfrac{\partial \eta}{\partial \tilde{\eta}} \end{vmatrix}, \tag{4.94}$$

where $\hat{x}$ and $\hat{y}$ are the coordinates involving the random variable ζ in the physical space (see Equations (4.5) and (4.6)). ξ and η are the parametric coordinates in the parameter space, and are obtained by the knot vectors and parent element coordinates of the Gauss points, as follows:

$$\xi = \frac{\left((\xi_{i+1} + \xi_i) + \tilde{\xi}(\xi_{i+1} - \xi_i) \right)}{2}, \tag{4.95}$$

$$\eta = \frac{\left((\eta_{i+1} + \eta_i) + \tilde{\eta}(\eta_{i+1} - \eta_i) \right)}{2}, \tag{4.96}$$

where $\tilde{\xi}$ and $\tilde{\eta}$ are the parent element coordinates. In practice, the Equations (4.95) and (4.96) involves a mapping which from the parent space to the parameter space $f : \tilde{\Omega} \rightarrow \hat{\Omega}$. Additionally, the mapping from the parameter space to the physical space $g : \hat{\Omega} \rightarrow \Omega$ is calculated from the NURBS basis functions and the control point coordinates. The transformation matrix for mapping $g : \hat{\Omega} \rightarrow \Omega$ is represented by Equations (2.30), (4.5) and (4.6)

$$J_g(\zeta) = \begin{bmatrix} \dfrac{\partial \hat{x}}{\partial \xi} & \dfrac{\partial \hat{x}}{\partial \eta} \\[2ex] \dfrac{\partial \hat{y}}{\partial \xi} & \dfrac{\partial \hat{y}}{\partial \eta} \end{bmatrix}$$

$$= \begin{bmatrix} \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \xi}\left[x_i + \sum_{j=0}^{n} L_{x_j}\Psi(\zeta)\right] & \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \eta}\left[x_i + \sum_{j=0}^{n} L_{x_j}\Psi(\zeta)\right] \\[3ex] \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \xi}\left[y_i + \sum_{j=0}^{n} L_{y_j}\Psi(\zeta)\right] & \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \eta}\left[y_i + \sum_{j=0}^{n} L_{y_j}\Psi(\zeta)\right] \end{bmatrix},$$

$$(4.97)$$

where n_e is the number of the nonzero NURBS basis function over element e, which is equal to $(p+1)^2$ in this formulation, and where p is the order of the basis function. The associated Jacobian determinant for this mapping is denoted as $\left| J_g(\zeta) \right|$. Thus, the matrix form of Equation (4.94) is represented by

$$J(\zeta) = \begin{bmatrix} \dfrac{\partial \hat{x}}{\partial \xi}\dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial \hat{x}}{\partial \eta}\dfrac{\partial \eta}{\partial \tilde{\eta}} \\[2ex] \dfrac{\partial \hat{y}}{\partial \xi}\dfrac{\partial \xi}{\partial \tilde{\xi}} & \dfrac{\partial \hat{y}}{\partial \eta}\dfrac{\partial \eta}{\partial \tilde{\eta}} \end{bmatrix}$$

$$= \begin{bmatrix} \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \xi}\left[x_i + \sum_{j=0}^{n} L_{x_j}\Psi(\zeta)\right]\dfrac{\partial \xi}{\partial \tilde{\xi}} & \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \eta}\left[x_i + \sum_{j=0}^{n} L_{x_j}\Psi(\zeta)\right]\dfrac{\partial \eta}{\partial \tilde{\eta}} \\[3ex] \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \xi}\left[y_i + \sum_{j=0}^{n} L_{y_j}\Psi(\zeta)\right]\dfrac{\partial \xi}{\partial \tilde{\xi}} & \displaystyle\sum_{i=1}^{n_e} \dfrac{\partial R_i}{\partial \eta}\left[y_i + \sum_{j=0}^{n} L_{y_j}\Psi(\zeta)\right]\dfrac{\partial \eta}{\partial \tilde{\eta}} \end{bmatrix}$$

$$(4.98)$$

Note that, in this study, uncertainty in shape is considered as the variable random, and represented by introducing PCE into the control points coordinates in the physical space. Therefore, uncertainty representation exists in this mapping process.

In addition, the orthogonal property of PCE is needed to be used in order to perform the formulation. Therefore, the $\left| J(\zeta) \right|$ needs to be rewritten as Hermite polynomial form, and obtained by the following procedure: substituting Equations (4.7) and (4.8) into Equation (4.98), solving the Jacobian determinant, and then combining like terms with respect to ζ. Finally, the $\left| J(\zeta) \right|$ is expressed as:

$$\left|\boldsymbol{J}(\zeta)\right|=\left|\boldsymbol{J}_0\right|+\left|\boldsymbol{J}_1\right|\zeta+\ldots+\left|\boldsymbol{J}_n\right|\zeta^n=\sum_{i=0}^{n}\left|\boldsymbol{J}\right|\Psi(\zeta), \qquad (4.99)$$

where n is the polynomial order. it is equal to 2, since, in this study, the representation of uncertainty in shape was assumed to follow a normal distribution.

Here, the implementation of polynomial transforms of $\sum_{i=0}^{n}\left|\boldsymbol{J}_i\right|\zeta^i \to \sum_{i=0}^{n}\left|\boldsymbol{J}_i\right|\Psi(\zeta)$ is based on the algorithm of paper [11], which we refer to as follows.

❖ Chen [11] developed an algorithm to realize automatic transformation from the general polynomial form to the Hermite polynomial form. Here, as an example, the polynomial transforms of the m order will be explained such as the following equation.

$$\sum_{i=0}^{m}a_i\theta^i \to \sum_{i=0}^{m}b_i\psi_i(\theta). \qquad (4.100)$$

Table 4.1 The Coefficients t_{ij} of θ of Hermite Polynomial $\psi_i(\theta)$

j i	θ^0	θ^1	θ^2	θ^3	...
0	1	0	0	0	
1	0	1	0	0	
2	-1	0	1	0	
3	0	-3	0	1	
...					

Here, the coefficients $a_i (i=1,2,\cdots m)$ and the coefficients $b_i (i=1,2,\cdots m)$ of Equation(4.100) represents an arbitrary coefficient, and the θ represents an arbitrary random variable. And a table is also shown in which i is order of

161

polynomial, j is the jth power of θ, and the values t_{ij} is coefficients of θ^j in each order, it is corresponding to the basis function of Table 2.1. The coefficient b_i of the Equation (4.100) can be computed by the following procedure.

(i)

Combined the Table 3.3, firstly, the coefficient a_m of the highest order term (θ^m) is considered. The $a_m\theta^m$ can be expressed as follows.

$$a_m\theta^m = a_m\psi_m(\theta) - a_m\sum\nolimits_{j=0}^{m-1}t_{mj}\theta^j \tag{4.101}$$

where, $b_m = a_m$. However, it is find that Equation (4.101) has a remainder term $a_m\sum\nolimits_{j=0}^{m-1}t_{mj}\theta^j$. This coefficient need be added to $a_j(j=0,1,\cdots m-1)$ that is, the coefficient a_j will be rewritten as follows.

$$a_j = a_j - a_m t_{mj} \quad (j=0,1,\cdots m-1). \tag{4.102}$$

For example: when $m=3$ is considered, the $a_3\theta^3$ can be rewritten as $a_3\theta^3 = a_3(\theta^3 - 3\theta) + 3a_3\theta$ in which $\psi_3(\theta) = \theta^3 - 3\theta$, that is $t_{30} = 0$, $t_{31} = -3$, $t_{32} = 0$. By adding the remainder term $3a_3\theta$ to the coefficient a_1 of θ^1 on the left side of Equation (4.100), a new coefficient $a_1 = a_1 + 3a_3$ will be obtained and execute the following steps.

(ii)

Secondly , the coefficient a_{m-1} of second highest order term θ^{m-1} is considered. By using the same method as (i), $b_{m-1} = a_{m-1}$ can be obtained. Thus,

the coefficient a_j on the left side of Equation (4.100) can be expressed as

$a_j = a_j - a_{m-1}t_{m-1j}$, $(j = 0,1,2,\cdots m-2)$ because terms less than *m-2* is remained.

(iii)

Finally, the coefficients of θ^{m-j} $(j = 2,3,\cdots m)$ can be obtained in order by reducing the order.

Thus, we can find that by the above algorithm, the polynomial of *n* order can be automatically transformed into Hermite polynomial.

Moreover, in order to obtain the strain matrix $\boldsymbol{B}(\zeta)$, the derivatives of the basis-functions with respect to the physical coordinates must be calculated. Thus, we first used Equations (4.95) and (4.96) to calculate the parametric coordinates with respect to the quadrature points in the parent space. subsequently, the derivatives of the basis-functions with respect to the parameter coordinates were calculated by Equation (2.31). Finally, the derivatives in the physical space were obtained by applying the chain-rule, as follows:

$$\frac{\partial R(\xi,\eta)}{\partial \hat{x}} = \frac{\partial R}{\partial \xi}\frac{\partial \xi}{\partial \hat{x}} + \frac{\partial R}{\partial \eta}\frac{\partial \eta}{\partial \hat{x}}, \tag{4.103}$$

$$\frac{\partial R(\xi,\eta)}{\partial \hat{y}} = \frac{\partial R}{\partial \xi}\frac{\partial \xi}{\partial \hat{y}} + \frac{\partial R}{\partial \eta}\frac{\partial \eta}{\partial \hat{y}}, \tag{4.104}$$

where $\partial \xi/\partial \hat{x}$, $\partial \xi/\partial \hat{y}$ are obtained by calculating the inverse of the mapping from the parameter space to the physical space, and the inverse of this mapping is denoted by $1/\left|\boldsymbol{J}_g(\zeta)\right|$. In this expression, the denominator contains the random variable ζ ; therefore, it cannot be directly calculated by applying the PCE. However, $1/\left|\boldsymbol{J}_g(\zeta)\right|$ can be derived approximately by using orthogonally of Hermite polynomials, as follows

$$\left| \boldsymbol{J}_g \left(\zeta \right) \right|^{-1} = \sum_{a=0}^{r} \Upsilon_a \Psi_a \left(\zeta \right), \tag{4.105}$$

where $\Psi_a \left(\zeta \right)$ represents the *p*th-order Hermite polynomials, and Υ_a represents the unknown coefficients with respect to the approximate polynomials. We used an algorithm based on literature [11] in order to calculate the unknown coefficients. Here, we will briefly describe the methods described in literature [11].

Here, it is assumed that $1/\left| \boldsymbol{J}_g \left(\zeta \right) \right|$ can be approximated by Hermite PCE as follows.

$$\frac{1}{\left| \boldsymbol{J} \left(\zeta \right) \right|} = \sum_{j=0}^{r} \Upsilon_j \Psi_j \left(\zeta \right), \tag{4.106}$$

where, Υ_j are unknown coefficients of approximated polynomial, $\Psi_j \left(\zeta \right)$ are the basis functions, represented by Hermite polynomial. The unknown coefficients Υ_j can be decided as follows. Firstly, Equation (4.106) is rewritten as follows

$$\sum_{i=0}^{n} \left| \boldsymbol{J}_i \right| \Psi(\zeta) \times \sum_{j=0}^{r} \Upsilon_j \Psi_j \left(\zeta \right) = 1, \tag{4.107}$$

Multiplying $\Psi_t(\zeta) w(\zeta)$ to both sides of Equation (4.107), and integrating the equation, we can be obtained following equation.

$$\int_{-\infty}^{\infty} \sum_{i=0}^{n} \left| \boldsymbol{J}_i \right| \Psi_i(\zeta) \times \sum_{j=0}^{r} \Upsilon_j \Psi_j \left(\zeta \right) \Psi_t(\zeta) w(\zeta) d\zeta = \int_{-\infty}^{\infty} 1 \times \Psi_t(\zeta) w(\zeta) d\zeta. \tag{4.108}$$

when the $\int_D (\text{Equation}) \Psi_t(\zeta) w(\zeta) d\zeta = \left\langle (\text{Equation}) \Psi_t(\zeta) \right\rangle$ is considered, this equation can be depicted as follows

$$\sum_{i=0}^{n} |\boldsymbol{J}_i| \sum_{j=0}^{r} \Upsilon_j \left\langle \Psi_i(\zeta)\Psi_j(\zeta)\Psi_t(\zeta) \right\rangle = \left\langle \Psi_t(\zeta) \right\rangle. \qquad (4.109)$$

Thus, the following simultaneous equations about the coefficients Υ_j are obtained when the $t=0, ..., r$ or n ($t=r$ when $r<n$, $t=n$ when $n<r$).

$$\begin{bmatrix} J'_{00} & J'_{01} & \cdots & J'_{0n} \\ J'_{10} & J'_{11} & \cdots & J'_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ J'_{n0} & J'_{n1} & \cdots & J'_{nn} \end{bmatrix} \begin{bmatrix} \Upsilon_0 \\ \Upsilon_1 \\ \vdots \\ \Upsilon_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad (4.110)$$

where J'_{jt} is $\sum_{i=0}^{n} |\boldsymbol{J}_i| \left\langle \Psi_i(\zeta)\Psi_j(\zeta)\Psi_t(\zeta) \right\rangle$. Here, $\left\langle \Psi_i(\zeta)\Psi_j(\zeta)\Psi_t(\zeta) \right\rangle$ can be evaluated by numerical integration (Equation (4.111a)) or by using Equation (4.111b) [15]

$$\left\langle \Psi_i(\zeta)\Psi_j(\zeta)\Psi_t(\zeta) \right\rangle = \int_D \left\langle \Psi_i(\zeta)\Psi_j(\zeta)\Psi_t(\zeta) \right\rangle w(\zeta) d\zeta, \qquad (4.111a)$$

$$\left\langle \Psi_i(\zeta)\Psi_j(\zeta)\Psi_t(\zeta) \right\rangle = \begin{cases} 0 & i+j+t = odd, \max(i,j,t) > s \\ \dfrac{i!\,j!\,t!}{(s-i)!(s-j)!(s-t)!} & otherwise \quad (s = \dfrac{i+j+t}{2}) \end{cases}. \qquad (4.111b)$$

Then the unknown coefficients of approximation polynomial, Υ_j, is derived by solving simultaneous equations. And, we apply it to the solution of matrix $\boldsymbol{B}(\zeta)$ in Equation (4.93). Thus, $\boldsymbol{B}(\zeta)$ is expressed by Equations (4.103), (4.104) and (4.105), as follows:

$$\boldsymbol{B}(\zeta) = \begin{bmatrix} \dfrac{\partial R(\xi,\eta)}{\partial \hat{x}} & 0 \\ 0 & \dfrac{\partial R(\xi,\eta)}{\partial \hat{y}} \\ \dfrac{\partial R(\xi,\eta)}{\partial \hat{y}} & \dfrac{\partial R(\xi,\eta)}{\partial \hat{x}} \end{bmatrix} = \left| \boldsymbol{J}_g(\zeta) \right|^{-1} \begin{bmatrix} \dfrac{\partial R(\xi,\eta)}{\partial \xi} + \dfrac{\partial R(\xi,\eta)}{\partial \eta} & 0 \\ 0 & \dfrac{\partial R(\xi,\eta)}{\partial \xi} + \dfrac{\partial R(\xi,\eta)}{\partial \eta} \\ \dfrac{\partial R(\xi,\eta)}{\partial \xi} + \dfrac{\partial R(\xi,\eta)}{\partial \eta} & \dfrac{\partial R(\xi,\eta)}{\partial \xi} + \dfrac{\partial R(\xi,\eta)}{\partial \eta} \end{bmatrix}. \quad (4.112)$$

We substituted Equation (4.104) into Equation (4.105) in order to rewrite the strain matrix $\boldsymbol{B}(\zeta)$ as PCE form with respect to the random variable ζ :

$$\boldsymbol{B}(\zeta)=\frac{1}{\left|\boldsymbol{J}_g(\zeta)\right|}\times\boldsymbol{B}_0+\frac{1}{\left|\boldsymbol{J}_g(\zeta)\right|}\times\boldsymbol{B}_1\times\zeta+\cdots+\frac{1}{\left|\boldsymbol{J}_g(\zeta)\right|}\times\boldsymbol{B}_a\times\zeta=\frac{1}{\left|\boldsymbol{J}_g(\zeta)\right|}\times\sum_{a=0}^{r}\boldsymbol{B}_a\zeta^a. \quad (4.113)$$

Recall that in the above equations, the strain matrix and Jacobian matrix were all expressed in PCE form. Therefore, we were able to obtain the PCE representation form of Equation (4.93) as follows:

$$\boldsymbol{k}_e(\zeta)=\sum_{j=0}^{n+r}\boldsymbol{k}_{e,j}\Psi_j(\zeta). \quad (4.114)$$

Subsequently, we assembled the local stiffness matrix $\boldsymbol{k}_e$ into the global stiffness matrix $\mathbf{K}$ and rewrote it in PCE form, as follows:

$$\mathbf{K}=\sum_{j=0}^{n+r}\mathbf{K}\Psi_j(\zeta). \quad (4.115)$$

Similarly, the local force vector is expressed as follows:

$$\boldsymbol{f}_e(\zeta)=\int_{\Omega^e}R(\xi)\hbar_e\left|J(\zeta)\right|d\Omega^e, \quad (4.116)$$

where $\hbar_e$ is the surface force vector. The PCE form is denoted as:

$$\boldsymbol{f}_e(\zeta)=\sum_{i=0}^{n}\boldsymbol{f}_{e,i}\Psi_i(\zeta) \quad (4.117)$$

The derivation process of Equation (4.117) was the same as that of Equation (4.114); however, due to the applied loads on the single-side, the $\boldsymbol{f}_e$ was the obtained by the one-dimensional NURBS basis functions. Then, the global force vector was assembled and represented in PCE form, as follows:

$$\mathbf{F}=\sum_{i=0}^{n}\mathbf{F}\Psi_i(\zeta). \quad (4.118)$$

Moreover, the unknown displacement response vector **d** in Equation (4.89) could also be derived approximately by utilising orthogonal property of Hermite polynomials. Thus, we first represented it in PCE form:

$$\mathbf{d} = \sum_{k=0}^{n+r} \mathbf{d}_k \Psi_k(\zeta).$$
(4.119)

Note that the order of PCE for Equation (4.119) is assumed in accord with that of global stiffness matrix (see Equation (4.115)), which is needed in order to solve the global stiffness matrix later. The substitution of Equations (4.115), (4.118), and (4.119) into Equation (32), yields:

$$\sum_{j=0}^{n+r} \mathbf{K} \Psi_j(\zeta) \times \sum_{k=0}^{n+r} \mathbf{d}_k \Psi_k(\zeta) = \sum_{i=0}^{n} \mathbf{F} \Psi_i(\zeta).$$
(4.120)

To solve Equation (4.120), we utilised the same algorithm used to solve Equation (4.105). According to the algorithm [11], the orthogonal property of PCE was utilised in order to solve the unknown displacement response vector **d**. First the two-sides of Equation (4.120) were multiplied by $\Psi_m(\zeta)W(\zeta)$ and were then integrated, simultaneously. Thereby, we obtained:

$$\int_{-\infty}^{+\infty} \sum_{j=0}^{n+r} \mathbf{K} \Psi_j(\zeta) \times \sum_{k=0}^{n+r} \mathbf{d}_k \Psi_k(\zeta) \Psi_m(\zeta)W(\zeta)d\zeta = \int_{-\infty}^{+\infty} \sum_{i=0}^{n} \mathbf{F} \Psi_i(\zeta) \Psi_m(\zeta)W(\zeta)d\zeta. \quad (4.121)$$

Thus, the inner-product form can be given by

$$\sum_{j=0}^{n+r} \mathbf{K} \left\langle \Psi_k(\zeta) \Psi_m(\zeta) \Psi_j(\zeta) \right\rangle \sum_{k=0}^{n+r} \mathbf{d}_k = \sum_{i=0}^{n} \mathbf{F} \left\langle \Psi_i(\zeta) \Psi_m(\zeta) \right\rangle, \quad (4.122)$$

where

$$\left\langle \Psi_m(\zeta) \Psi_a(\zeta) \Psi_i(\zeta) \right\rangle = \int_D \Psi_m(\zeta) \Psi_i(\zeta) \Psi_a(\zeta) W(\zeta)d\zeta, \quad (4.123)$$

where $\left\langle \Psi_m(\zeta) \right\rangle$ is calculated by the Equation (3.8).

Subsequently, Equation (4.120) could be rewritten in a form of simultaneous equations by Equation (4.121), (4.122), and (4.123), as follows

$$
\begin{bmatrix}
\mathbf{K}_{0,0}\langle\cdot\rangle & \mathbf{K}_{0,1}\langle\cdot\rangle & \cdots & \mathbf{K}_{0,n+r}\langle\cdot\rangle \\
\mathbf{K}_{1,0}\langle\cdot\rangle & \mathbf{K}_{1,1}\langle\cdot\rangle & \cdots & \mathbf{K}_{1,n+r}\langle\cdot\rangle \\
\vdots & \vdots & \vdots & \vdots \\
\mathbf{K}_{n+r,0}\langle\cdot\rangle & \mathbf{K}_{n+r,1}\langle\cdot\rangle & \cdots & \mathbf{K}_{n+r,n+r}\langle\cdot\rangle
\end{bmatrix}
\begin{bmatrix}
\mathbf{d}_0 \\
\mathbf{d}_1 \\
\vdots \\
\mathbf{d}_{n+r}
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{F}_0\langle\cdot\rangle \\
\mathbf{F}_1\langle\cdot\rangle \\
\vdots \\
\mathbf{F}_{n+r}\langle\cdot\rangle
\end{bmatrix},
\tag{4.124}
$$

where $\langle\cdot\rangle$ denotes the inner-product with respect to PCE. Finally, the stochastic response of the displacement representing uncertainty in shape was obtained by Equation (4.124).

After we find the displacement based on the Equations (4.119) and (4.120), the corresponding strain and stress can be calculated by

$$
\varepsilon = \boldsymbol{Bu}, \tag{4.125}
$$

$$
\sigma = \boldsymbol{DBu}. \tag{4.126}
$$

The corresponding forms of uncertainty are as follows

$$
\begin{aligned}
\boldsymbol{\varepsilon}(\zeta) &= \left(\sum_{u=0}^{n} A_u \Psi_u(\zeta) \right) \left(\sum_{j=0}^{1} \boldsymbol{B}_j \Psi_j(\zeta) \right) \left(\sum_{i=0}^{r} \boldsymbol{u}_i \Psi_i(\zeta) \right) \\
&= \sum_{i=0}^{n+r+1} \boldsymbol{\varepsilon}_i \Psi_i(\zeta),
\end{aligned}
\tag{4.127}
$$

$$
\begin{aligned}
\boldsymbol{\sigma}(\zeta) &= \boldsymbol{D}\boldsymbol{\varepsilon}(\zeta) \\
&= \boldsymbol{D} \left(\sum_{u=0}^{n} A_u \Psi_u(\zeta) \right) \left(\sum_{j=0}^{1} \boldsymbol{B}'_j \Psi_j(\zeta) \right) \left(\sum_{i=0}^{r} \boldsymbol{u}_i^{*'} \Psi_i(\zeta) \right) \\
&= \sum_{i=0}^{n+r+1} \boldsymbol{\sigma}_i \Psi_i(\zeta),
\end{aligned}
\tag{4.128}
$$

In this study, we used the C++ programming language in order to implement the formulation of proposed method. Moreover, all C++ program flow charts have been added to Appendix A.

Remark

(1)   In this formulation, the uncertain parameters were only imported into some specific control points in order to represent the uncertainty in shape. These specific control points were determined according to geometrical shape and range that needed to be analysed in the structure model, and these control points had ability to intuitively change shape within this range. Therefore, the uncertainty in shape for structure model could be accurately presented by just importing the uncertain parameters into one or more control points. In this context, this method greatly reduces the computational cost.

(2)   The role of PCE is important in the intrusive SIGA formulation procedure. Recalling that the stiffness matrix, force vector and displacement were all rewritten as a representation form of PCE, because we needed to use the orthogonal property of PCE to solve the global stiffness equation.

(3)   Note that the algorithms developed so far only apply to Gaussian stochastic fields and processes, and that program of SIGA formulation is implemented based on a single-patch.

4.5. CONCLUSIONS

In this section, we describe in detail the formalization process for stochastic isogeometric analysis method for uncertainty in shape. The formalization constructed is an intrusive formulation procedure. The deterministic isogoemetric analysis framework was rewritten as an uncertainty form based on PCE, and the orthogonal properties of PCE were fully utilised in order to solve the stiffness matrix.

The idea of SIGA with uncertainty in shape proceeds mainly as follows:

(1) The inherent randomness of a system is considered as an input, which is defined by the probability density function (PDF) of the random variable.

(2) IGA is used in conjunction with the PCE of the Gaussian random field describing the analysis process.

(3) The responses of the system are obtained by assuming that it is a function for inputting random variables.

Note that, the role of the PCE in this study is very important, since it is used to represent uncertain parameters.

 In addition, we propose a new analytical perspective to deal with the shape uncertainty problem in engineering analysis, namely, in this study, a novel method is proposed in the aspect of reliability analysis for uncertainty in shape, and it is carried out from a physically-based point of view. Because, the uncertainty in shape of structure model was represented by directly introducing stochastic parameters into the control points in the physical space, the new analytical viewpoint overcomes many shortcomings of the traditional SFEM method.

References

[1] M. Papadrakakis and N. D. Lagaros, 'Reliability-based structural optimization using neural networks and Monte Carlo simulation', *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 32, pp. 3491–3507, Jun. 2002.

[2] J. Argyris, M. Papadrakakis, and G. Stefanou, 'Stochastic finite element analysis of shells', *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 41, pp. 4781–4804, Sep. 2002.

[3] T. D. Hien and H.-C. Noh, 'Stochastic isogeometric analysis of free vibration of functionally graded plates considering material randomness', *Comput. Methods Appl. Mech. Eng.*, vol. 318, pp. 845–863, May 2017.

[4] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA, chapter 1: From CAD and FEA to Isogeometric Analysis: An Historical Perspective*. John Wiley & Sons, 2009.

[5] D. Terzopoulos and H. Qin, 'Dynamic NURBS with geometric constraints for interactive sculpting', *ACM Trans. Graph.*, vol. 13, no. 2, pp. 103–136, Apr. 1994.

[6] P. Kagan, A. Fischer, and P. Z. Bar-Yoseph, 'New B-Spline Finite Element approach for geometrical design and mechanical analysis', *Int. J. Numer. Methods Eng.*, vol. 41, no. 3, pp. 435–458, 1998.

[7] G. Celnikera and D. Gossardb, *Deformable Curve and Surface Finite-Elements for Free-Form Shape Design*. .

[8] L. Piegl and W. Tiller, *The NURBS Book*. Springer Science & Business Media, 1996.

[9] Chen X, Kawamura Y, Okada T. A Study on the Method of Structural Analysis with Uncertainty in Shape by Stochastic Finite Element Method (In Japanese). Journal of the Japan Society of Naval Architects and Ocean Engineers 2015; Vol. 22 (2015);187-195

[10] Chen X, Kawamura Y, Okada T. Stochastic finite element method based on response surface methodology considering uncertainty in shape of structures. 13th International Symposium on Practical Design of Ship and Other Floating Structures, Copenhagen, 2016.

[11] Chen X, Kawamura Y, Okada T. Development of structural analysis method with uncertainty in shape to follow non-normal distribution by stochastic finite element method (In Japanese). Transactions of the Japan Society for Computational Engineering and Science 2016; Vol. 2016 (2016) :20160019

[12] Chen X, Kawamura Y, Okada T. Development of Stochastic Finite Element Method for Problems with Uncertainty in Shape Following Non-normal Distribution (In Japanese). Conference of JASNAOE 2017; 437-442.

[13]   L. Piegl and W. Tiller, The NURBS Book. Springer Science & Business Media, 1996.

[14] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, chapter 2: NURBS as a Pre-analysisTool: Geometric Design and Mesh Generation. John Wiley & Sons, 2009.

[15] Szego G., Orthogonal Polynomials, fourth ed., American Mathematical Society, Providence, 1975.

5. NUMERICAL EXAMPLES

In this chapter, we apply the stochastic isogeometric analysis framework to deal with two-dimensional static elastic problems. Four numerical examples are thoroughly explored within this section. For the first numerical example which is a quarter-circular cantilever beam model, the freshly proposed generalized stochastic isogeometric analysis based polynomial chaos expansion is rigorously verified against the well-established theoretical results. Subsequently, the proposed stochastic isogeometric analysis framework is further implemented for the stochastic static analysis of localized corrosion in the second example. The NURBS geometric models in the above two examples are all built with a single patch. For multi-patch NURBS geometry, we give examples of the infinite plate with a circular hole and butt joint，respectively. These examples fully verify the stability of the multi-patch SIGA analytical framework and the possibility of its application to more practical engineering problems. In all cases, the model problems considered here were linear-elastic problems. Their solutions compare directly to that of MCS used as a reference value. The all calculations are implemented using C++ programming language, and timed on a Windows 10 pro 64bit with Intel(R) CPU E5-2603 v3 @ 1.6 GHz processor (12 CPUs) and 64GB RAM. Additionally, in the figure, the Poisson's ratio and Young's modulus are denoted by υ and E, respectively.

5.1. Example 1: quarter-circular cantilever beam

First, we present a verification example with regard to a quarter-circular cantilever beam shown in Figure 5.1. The analytical model is designed based on NURBS geometric modeling tool in CAD. The relevant model design dimensions are shown in Figure 5.2, where the inner and outer radii of the circular beam are 10 (mm) and 30 (mm), they are denoted as R_{in}, R_{out}, respectively. In addition, the setup of the static elasticity problem is also illustrated in Figure 5.2, the hash marks on the bottom edge denote the Dirichlet boundary conditions. The beam was subjected to a uniform pressure ($P_x = 1\text{N/mm}^2$) on its left-edge. Additionally, the material properties of the quarter-circular cantilever beam Poisson's ratio υ and elastic coefficient E are also marked in Figure 5.2, respectively. The NURBS base used in shape design and analysis is constructed from the knot vector Ξ and Π in ξ - and η - direction, respectively，as follows

$$\Xi = \{0,0,0,1,1,1\}$$

$$\Pi = \{0,0,0,0.250671,\ 0.501018\ ,\ 0.750020,1,1,1\}$$

The control mesh of the geometric model is shown in Figure 5.3, and the corresponding control point coordinates are listed in Table 5.1.

In this two-dimensional example, we assumed that the changes in shape take place in the inner circle. As shown in Figure 5.4, moving any one of the control points 1, 2, and 3 can change the shape of the inner circle. Here, we imported the uncertainty parameters into the coordinate of control point 2 in order to intuitively manipulate the shape of the inner circle, as Figure 5.5, while L_x and L_y denote the deviation length of control point 2 in the x- and y-direction, respectively. The corresponding standard deviation and mean of L_x and L_y in control point 2 were set to 0.5 and 0, respectively.
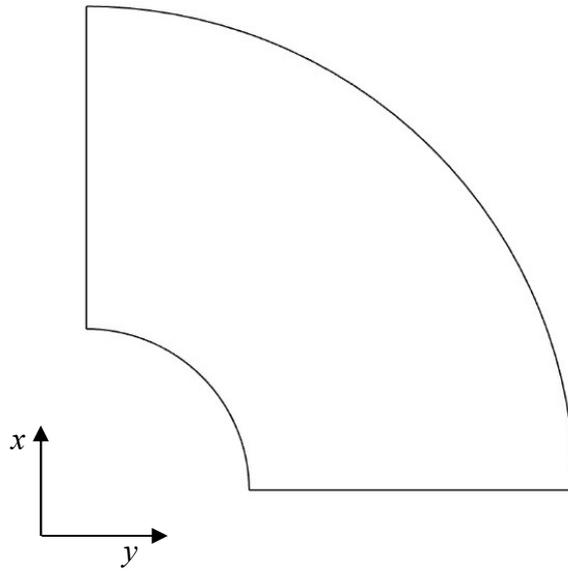
Figure 5.1 The geometry of the circular beam.
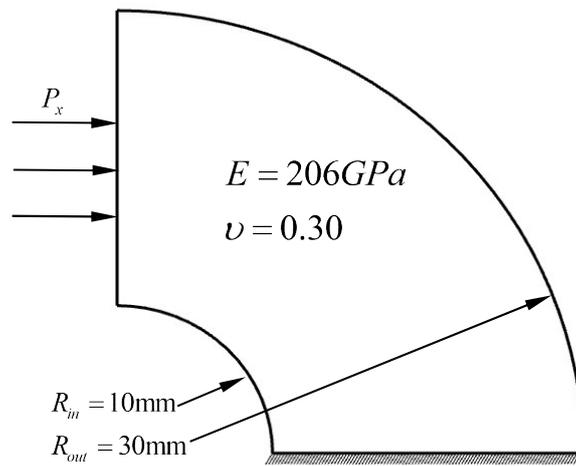
Figure 5.2 The geometry of the circular beam with material properties, boundary conditions and uniform pressure ($P_x = 1\text{N/mm}^2$)
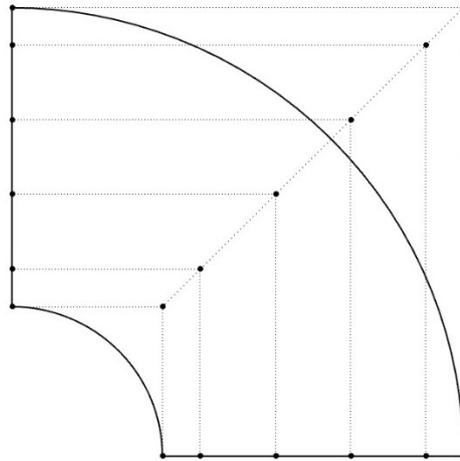
Figure 5.3 Control points and control net of quarter-circular cantilever beam

Table 5.1 Control point coordinates for quarter-circular cantilever beam

i	j	B_{ij}	w_{ij}
1	1	(0, 10)	1
2	2	(10, 10)	0.707
3	3	(10, 0)	1
1	4	(0, 12.5)	1
2	5	(12.5,12.5)	0.707
3	6	(12.5, 0)	1
1	1	(0, 17.5)	1
2	2	(17.5, 17.5)	0.707
3	3	(17.5, 0)	1
1	4	(0, 22.5)	1
2	5	(22.5, 22.5)	0.707
3	6	(22.5, 0)	1
1	1	(0, 27.5)	1
2	2	(27.5, 27.5)	0.707
3	3	(27.5, 0)	1
1	4	(0, 30)	1
2	5	(30, 30)	0.707
3	6	(30, 0)	1
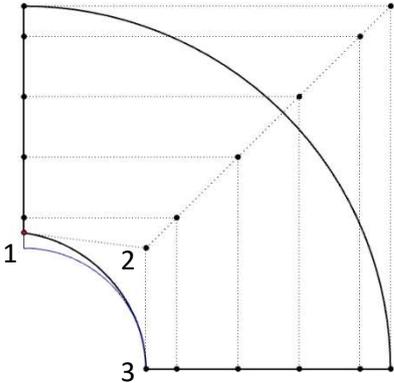
(a) Moving control point 1
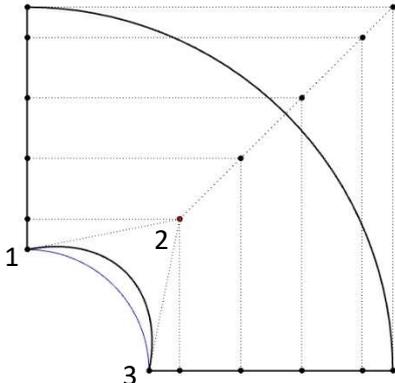
(b) Moving control point 2
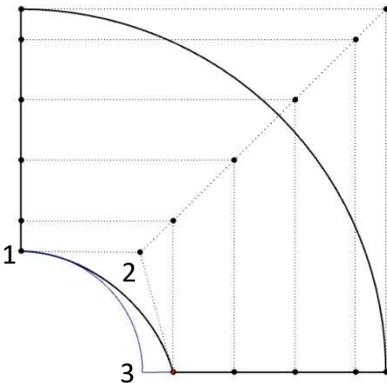
(c) Moving control point 3

Figure 5.4 Moving the control points 1, 2, 3, on the NURBS geometry
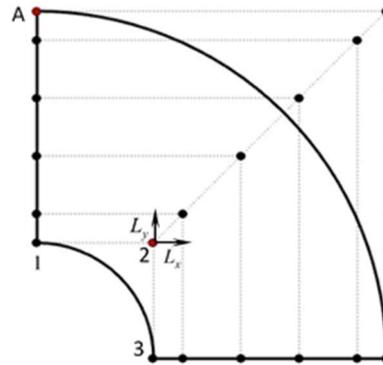
Figure 5.5 The uncertainty parameters are introduced into the coordinate of control point 2

In this example, we present the analysis results for the displacement response surface in the position of control point A in the position of control point 3 (see, Figure 5.5), which is the position with the largest displacement, and compare them to the analysis results obtained by the MCS and the IGA methods, respectively.

First, in the following, we give the deformation of the structure and the corresponding displacement results of IGA. In this case, the stochastic shape at the circular hole is arbitrary and controllable by introducing stochastic parameters to the corresponding control points. As shown in Figure 5.5, the stochastic parameters are introduced into coordinate of control point 2 in the x and y directions to represent the uncertainty. According to the stochastic parameters (the standard deviation and mean were 0.5 and 0, respectively) setting of Figure 5.5. In the Figure 5.6 to Figure 5.14, we respectively give stochastic shapes with the stochastic variables of -4, -3, -2, -1, 0, 1, 2, 3 and 4. That is the deformation distances are -2.0mm, -1.5mm, -1.0mm, -0.5mm, 0mm, 0.5mm, 1.0mm, 1.5mm, 2.0mm, respectively. According to these figures, we can find that when the circular hole expands outward, the displacement of point A becomes larger. In reality, for the normal distribution, the values less than one standard deviation away from the mean account for 68.27% of the set; while two standard deviations from the mean account for 95.45%; and three standard deviations account for 99.73%. Therefore, the probability of four standard deviations appearing is very small.
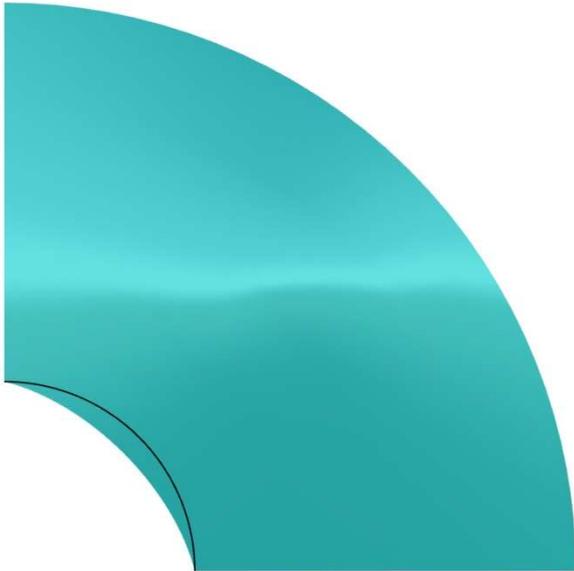
(a) The deformation
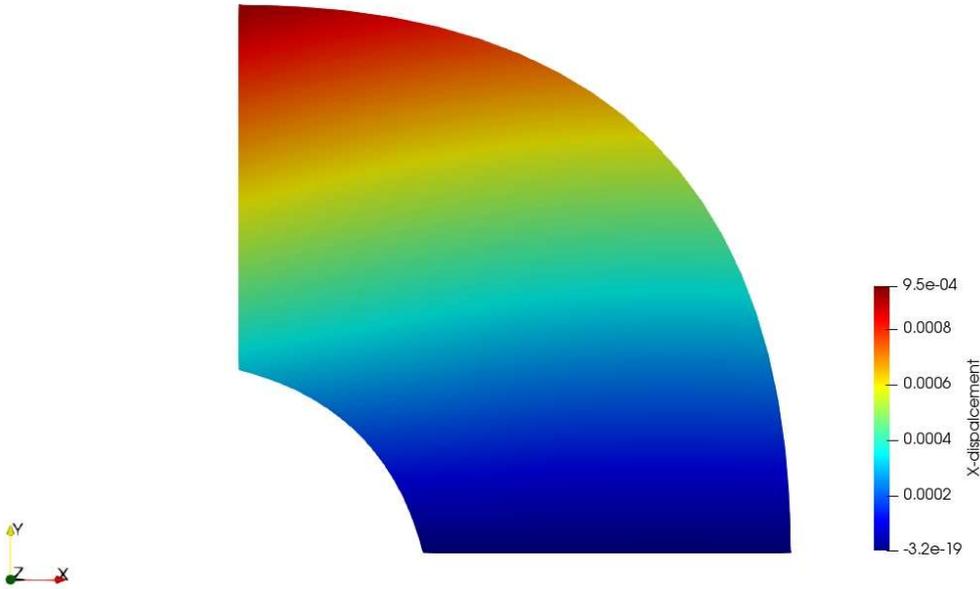
(b) The corresponding displacement results of IGA

Figure 5.6 The deformation distance -2.0mm
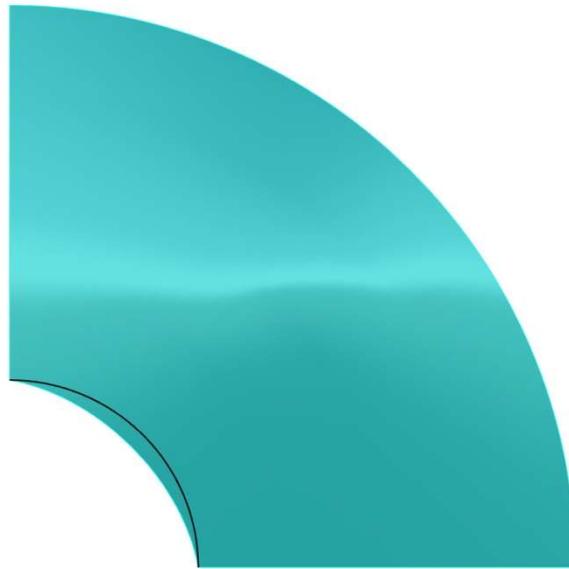
(a) The deformation
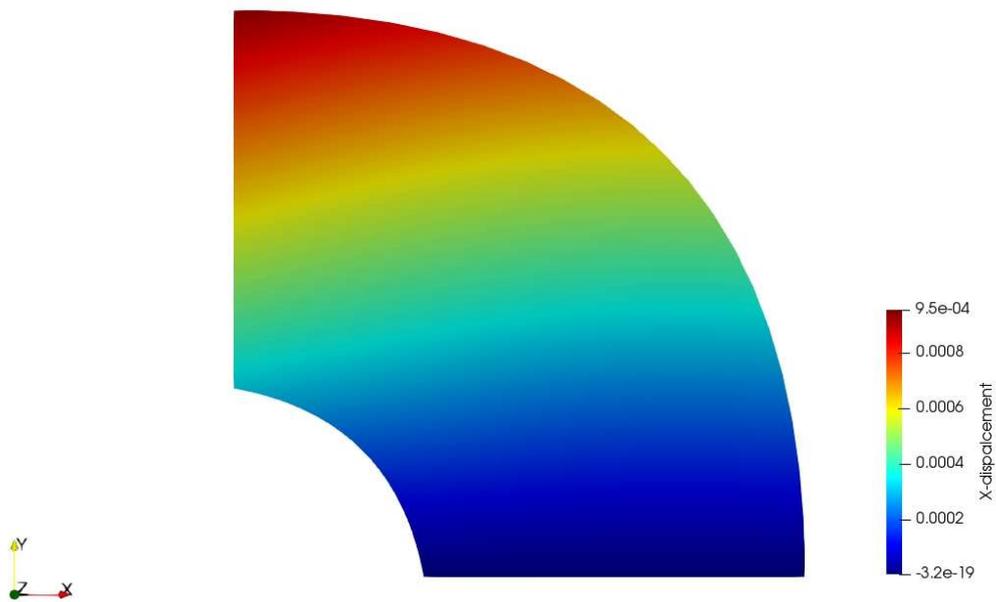
(b) The corresponding displacement results of IGA

Figure 5.7 The deformation distance -1.5mm
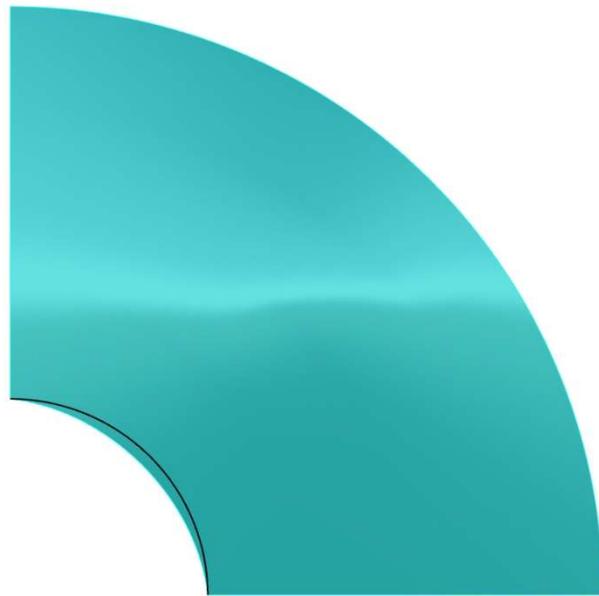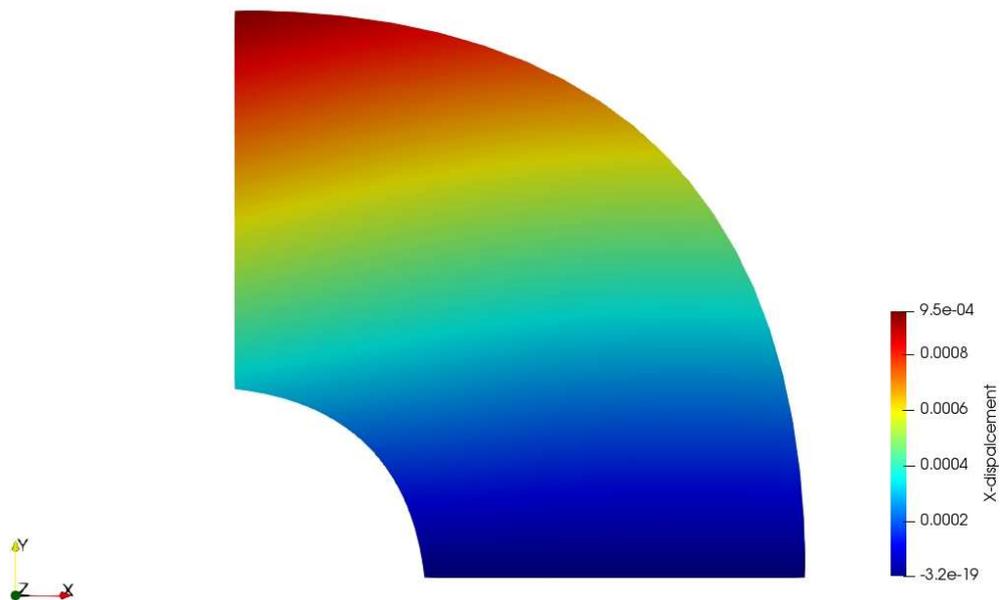
(a) The deformation

(b) The corresponding displacement results of IGA

Figure 5.8 The deformation distance -1.0mm
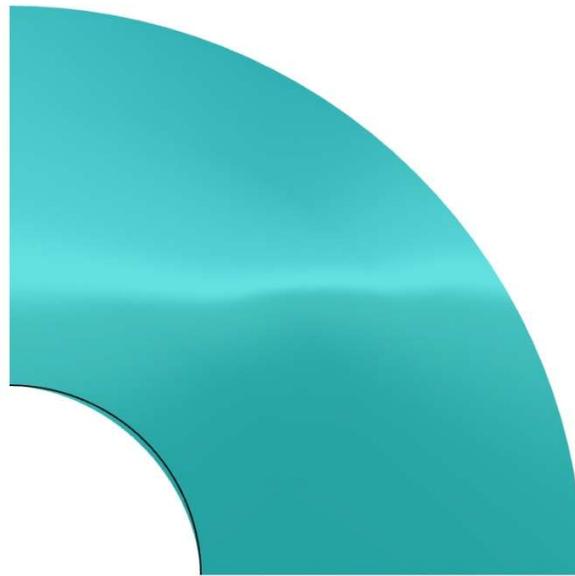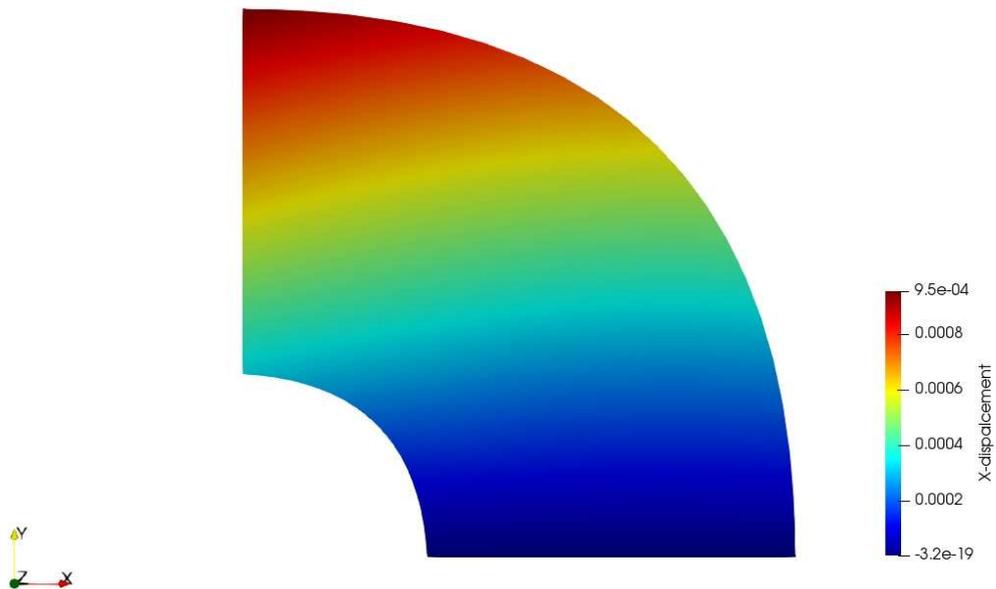
(a) The deformation

(b) The corresponding displacement results of IGA

Figure 5.9 The deformation distance -0.5mm
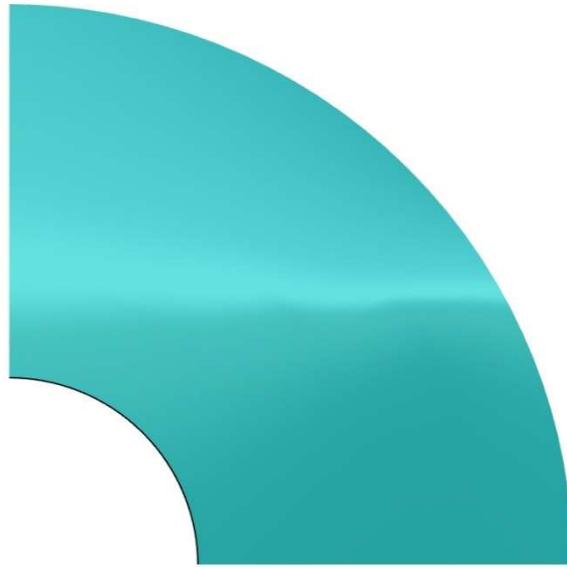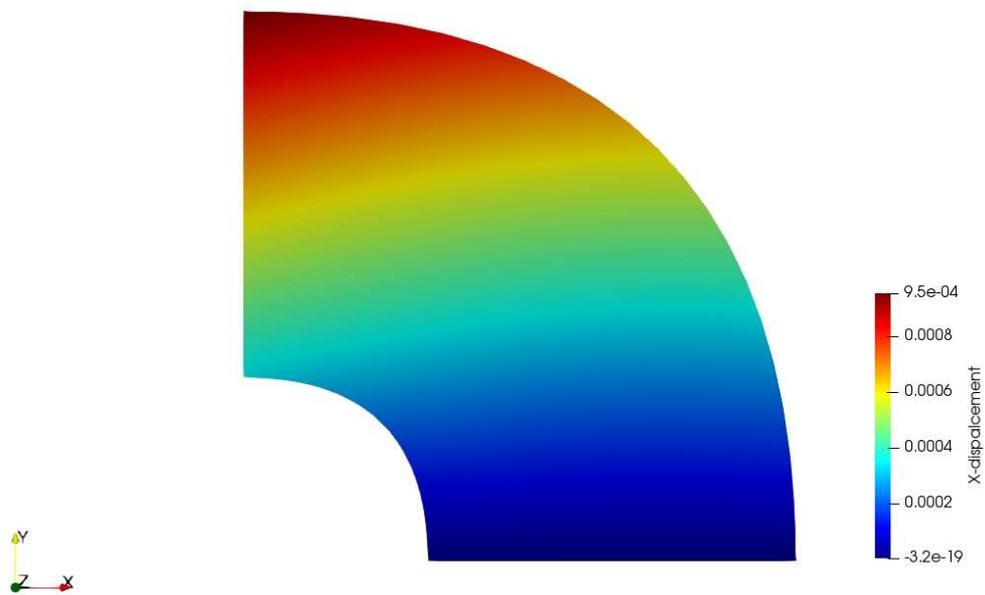
(a) The deformation

(b) The corresponding displacement results of IGA

Figure 5.10 The deformation distance 0mm
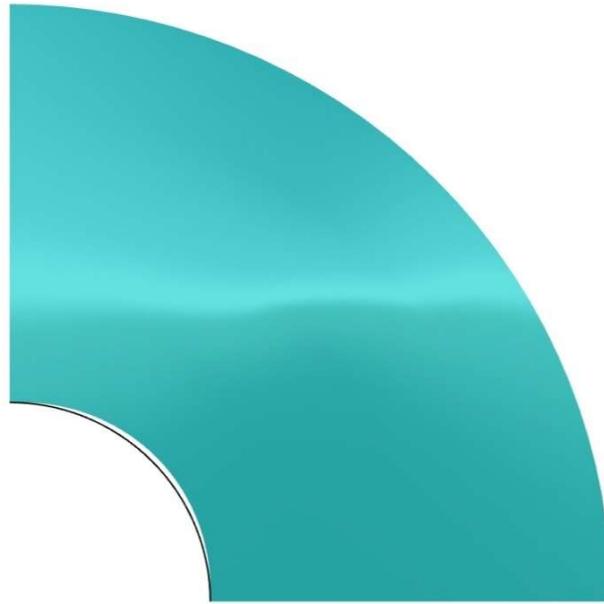
(a) The deformation
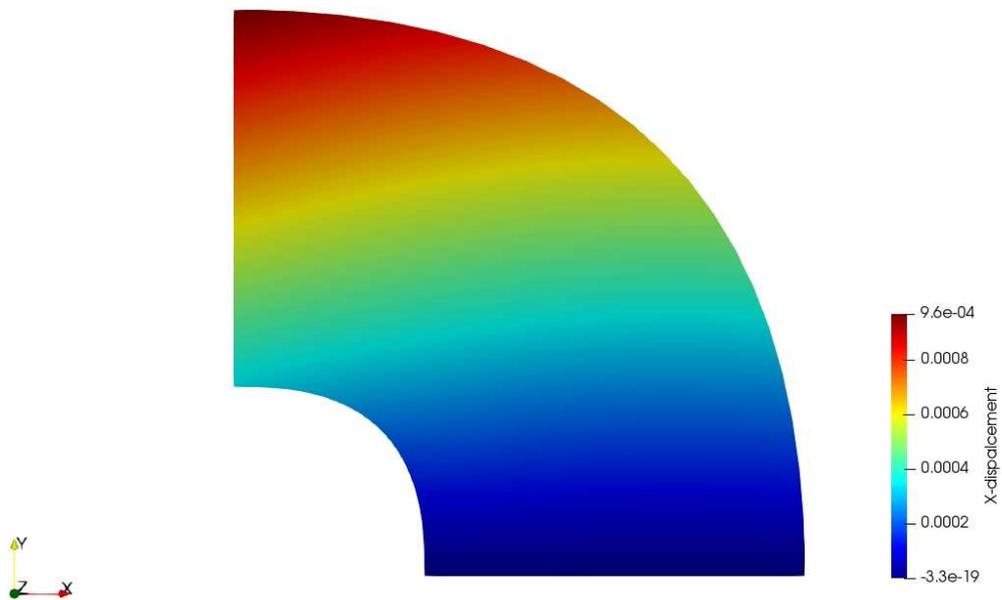
(b) The corresponding displacement results of IGA

Figure 5.11 The deformation distance 0.5mm
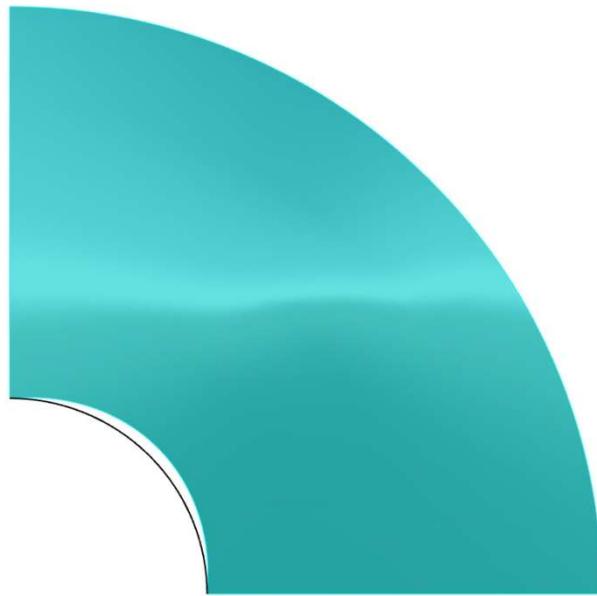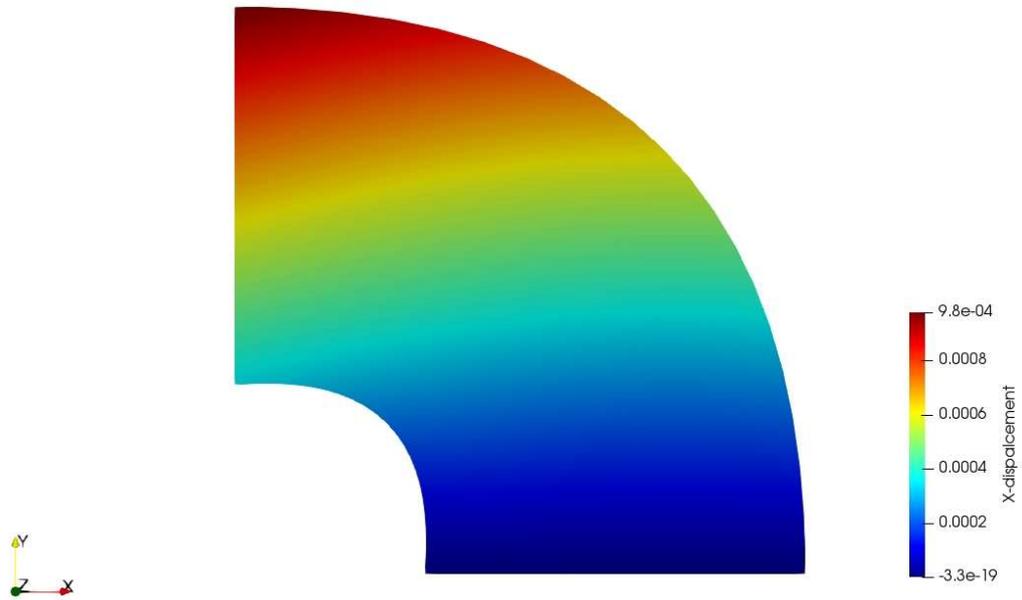
(a) The deformation

(b) The corresponding displacement results of IGA

Figure 5.12 The deformation distance 1.0mm
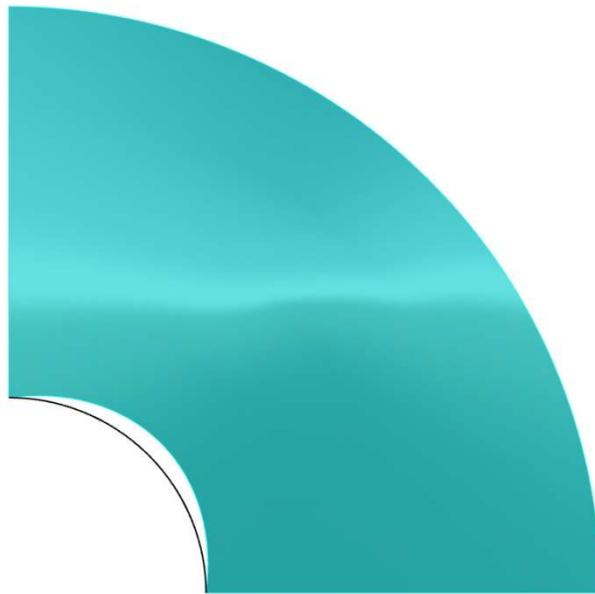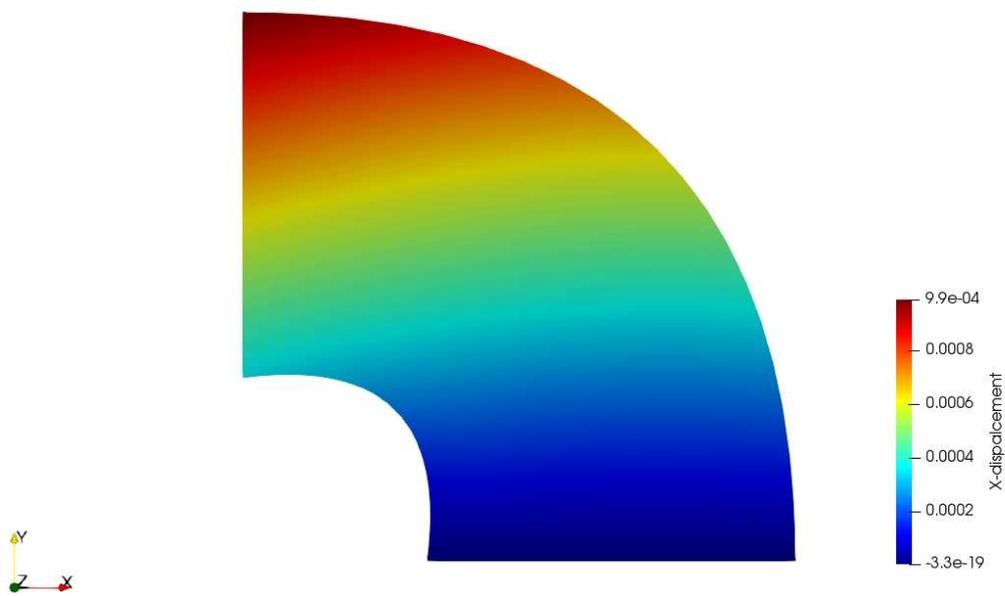
(a) The deformation

(b) The corresponding displacement results of IGA

Figure 5.13 The deformation distance 1.5mm
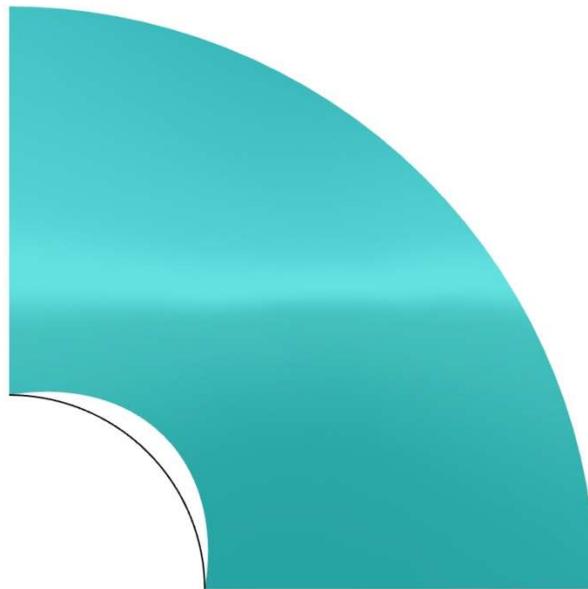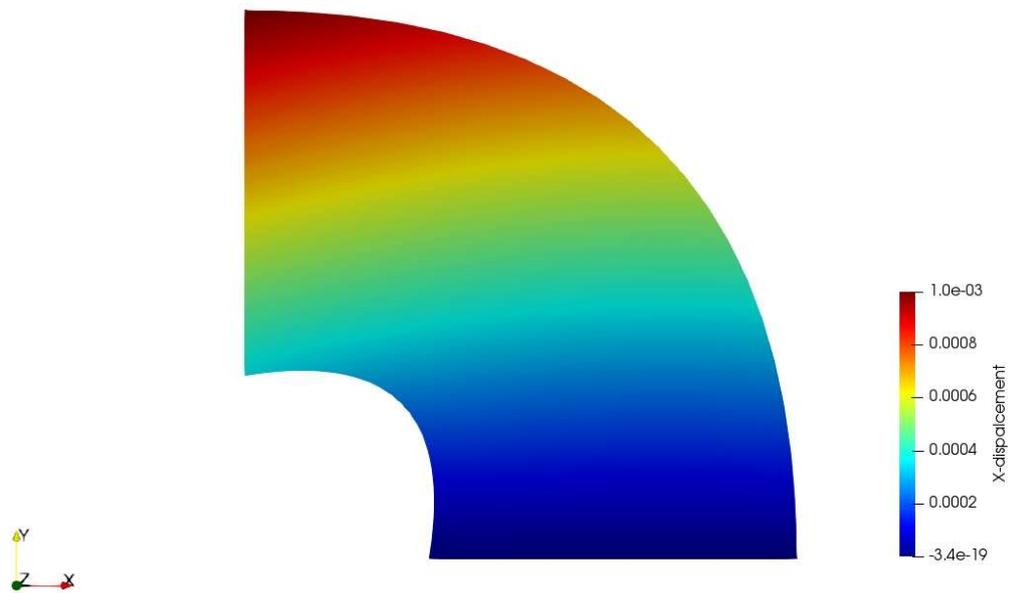
(a) The deformation

(b) The corresponding displacement results of IGA

Figure 5.14 The deformation distance 2.0mm

In order to demonstrate the accuracy of the analysis results obtained by the proposed method, the results of SIGA and IGA were compared, as shown in Figure 5.15, which shows that the proposed method had very high precision. The x-axis in Figure 5.15 represents the stochastic variable ζ , while the y-axis represents the corresponding displacement value. The numerical results of IGA were obtained by modifying coordinate of control point 2 in the x- and y- direction as indicated in Table 5.2. The response surface of SIGA was computed by using Equation (4.119). In this example, the order of the Hermite polynomials used to calculate the respond surface was taken as the fourth-order. The polynomial coefficients within the range of 0th to 8th-order are shown in Table 5.3 and denoted by a_i . They were derived by Equation (4.124) in Section 4. Thus, the response surface in this example was obtained by:

$$
\begin{aligned}
d(\zeta) = {}& 1.\times10^{-3}\Psi_0(\zeta) + 9.83\times10^{-6}\Psi_1(\zeta) + \\
& 1.21\times10^{-7}\Psi_2(\zeta) + 2.1\times10^{-9}\Psi_3(\zeta) - 3.31\times10^{-10}\Psi_4(\zeta)
\end{aligned}
\tag{5.1}
$$
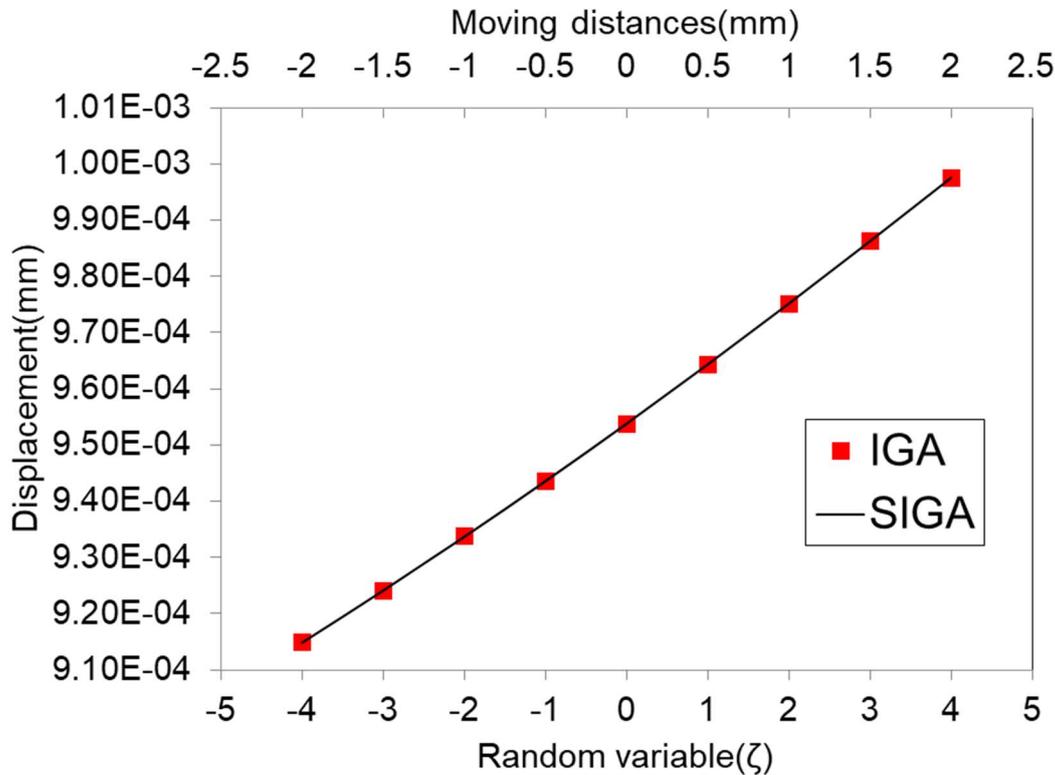
Figure 5.15 Displacement response obtained by SIGA and numerical results of IGA

In fact, the obtained response surface converged to an exact optimal solution as the order of the polynomial increased. Figure 5.16, shows error between the analysis results of IGA and SIGA. It can be seen that their errors are very small from the fourth-order of polynomials onwards. Thereby, the demand for precision of analysis results is met and this method is shown to be feasible and accurate. Additionally, the polynomial coefficients of response surface are also very small from fourth-order onwards and can be ignored (see Table 5.3). Based on these considerations, the polynomial of Equation (4.124) is only computed to the fourth-order.

Table 5.2: Moving distances for control point 2, and corresponding displacement of control point A, indicated in Figure 6(b).

Moving distances	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2
x-coordinate	8.00	8.50	9.00	9.50	10.00	10.5	11	11.5	12
y-coordinate	8.00	8.50	9.00	9.50	10.00	10.5	11	11.5	12
Displacement (10^{-4})	9.67	9.76	9.85	9.94	10.04	10.14	10.24	10.34	10.45

Table 5.3: Polynomial coefficients of response surface, (a_i).

Order (*i*th)	0	1	2	3	4	5	6	7	8
a_i	1.0×10^{-3}	9.8×10^{-6}	1.2×10^{-7}	-2.1×10^{-9}	-3.3×10^{-10}	-2.1×10^{-11}	-2.2×10^{-12}	2.5×10^{-12}	-1.5×10^{-13}
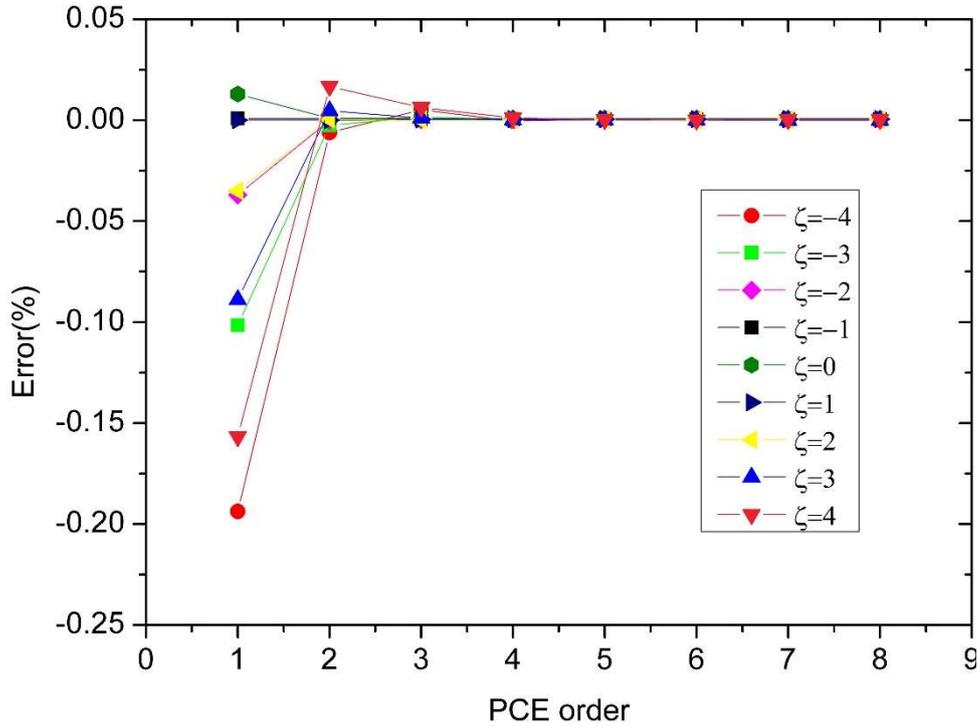
Figure 5.16 Error measured by numerical results of IGA and SIGA.

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.17, we present the displacement probability density functions for the control point A in the x-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples and the degrees of freedom in a single sample are 36. The average running time of the program for MSC is 99.196 second. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables ζ into the Equation (5.1). The degrees of freedom are 324 and the average running time of the program is the 0.093 second. The comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS, but at a much lower computational cost. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.4. It can be easily seen that they are almost equal. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).

Table 5.4 Statistics of the displacement

	SIGA	MCS	PCE
MEAN (10^{-4})	9.5393	9.5392	9.5405
STDEV (10^{-5})	1.0363	1.0468	1.0378
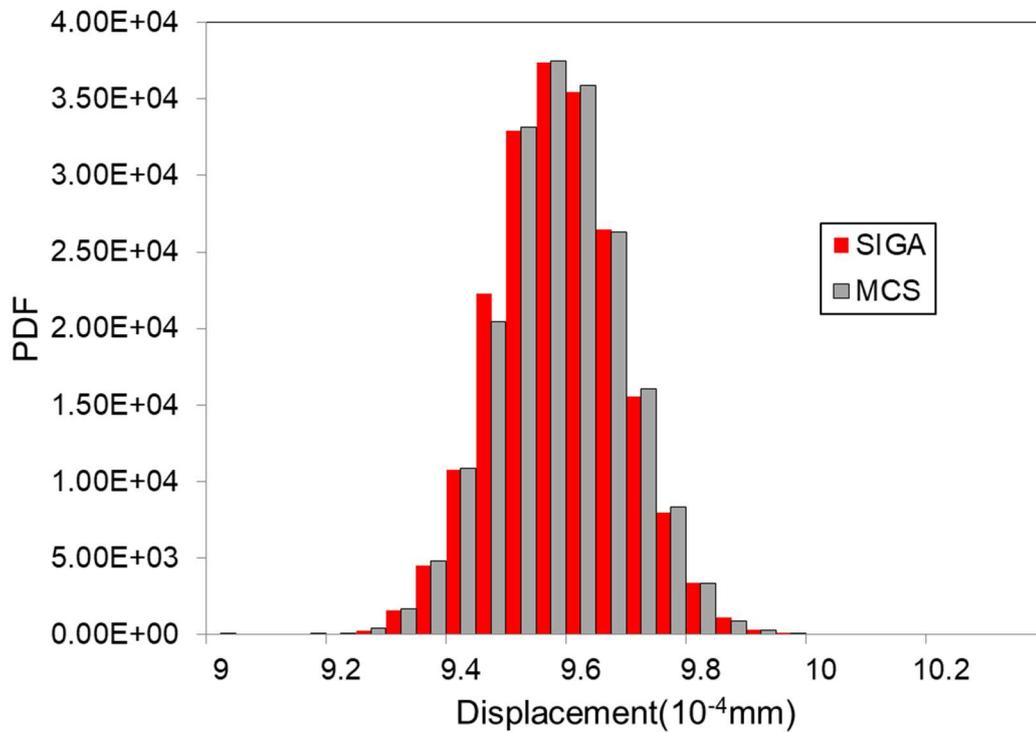
Figure 5.17: Displacement probability density functions for the control point A in the x-direction

Similarly, in the next part, we will give the stress situation for the control point 3 in the y-direction. In the Figure 5.18, we respectively give stochastic shapes with the stochastic variables of -4, -3, -2, -1, 0, 1, 2, 3 and 4. That is the deformation distances are -2.0mm, -1.5mm, -1.0mm, -0.5mm, 0mm, 0.5mm, 1.0mm, 1.5mm, 2.0mm, respectively. And in order to demonstrate the accuracy of the analysis results obtained by the proposed method, the results of SIGA and IGA were compared, as

shown in Figure 5.19, which shows that the proposed method had very high precision. We can find that when the random variable changes from - 4 to 4, that is, the deformation distance changes from – 2mm to 2mm, the stress along the Y axis of point 3 tends to decrease.
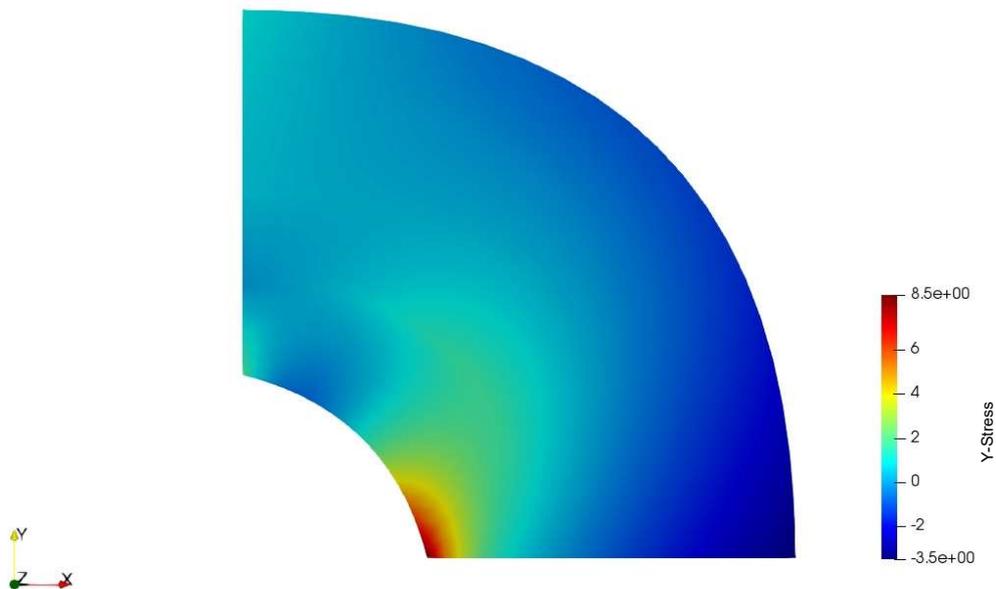
(a) The deformation distance -2.0mm
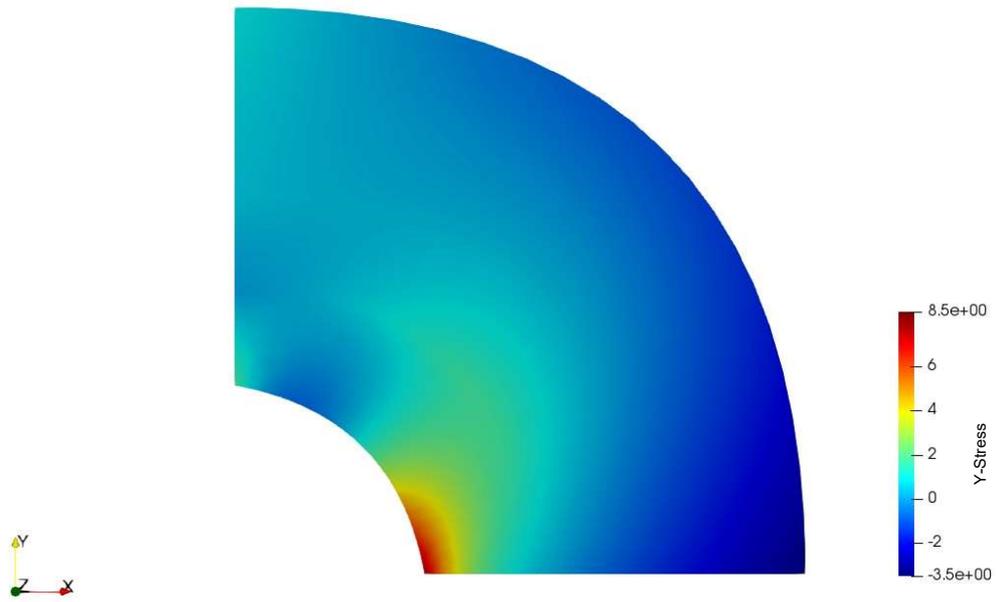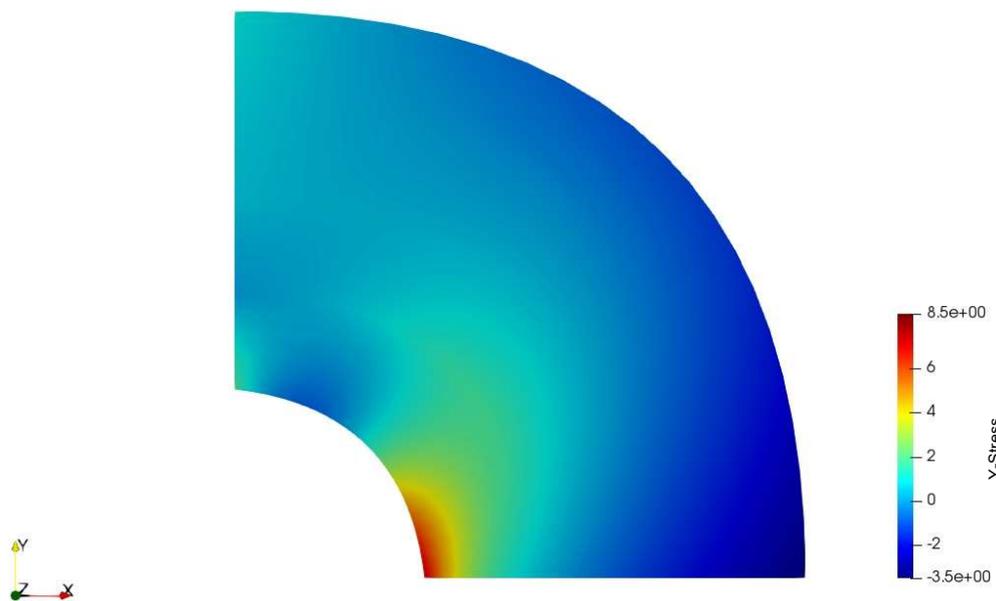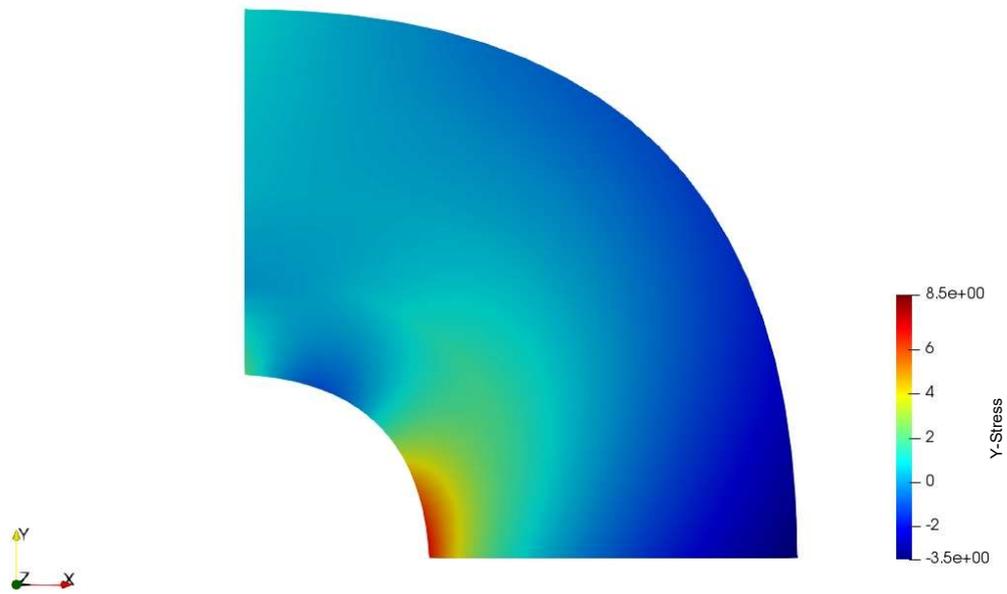
(b) The deformation distance -1.5mm
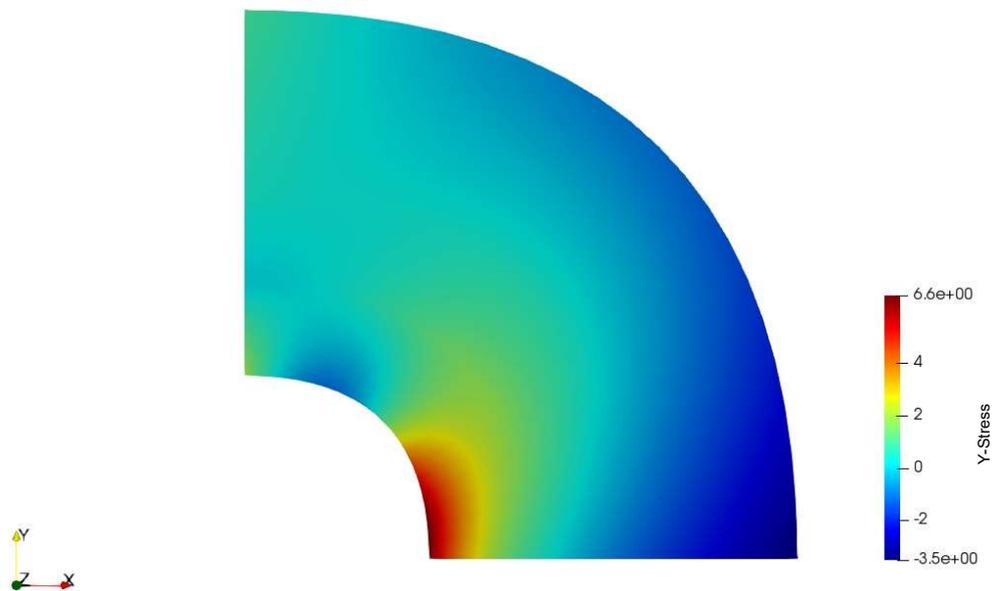
(c) The deformation distance -1.0mm

(d) The deformation distance -0.5mm
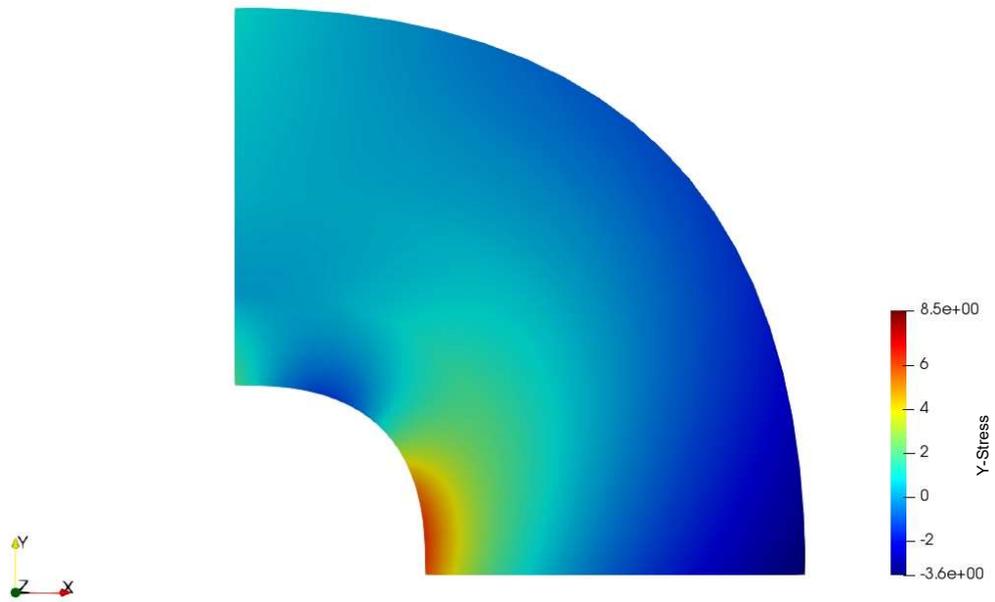
(e) The deformation distance 0.0mm
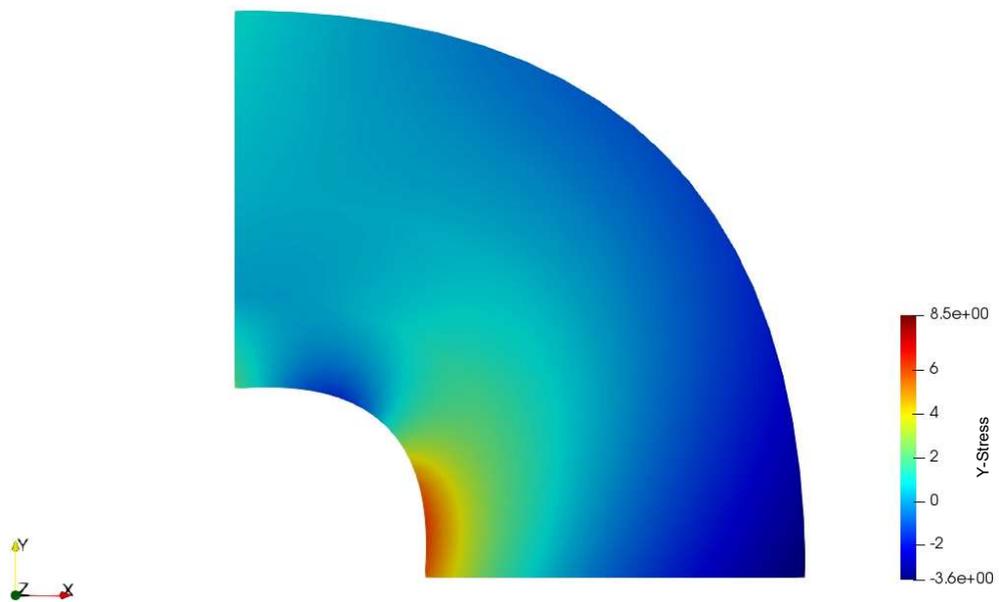
(f) The deformation distance 0.5mm

(g) The deformation distance 1.0mm
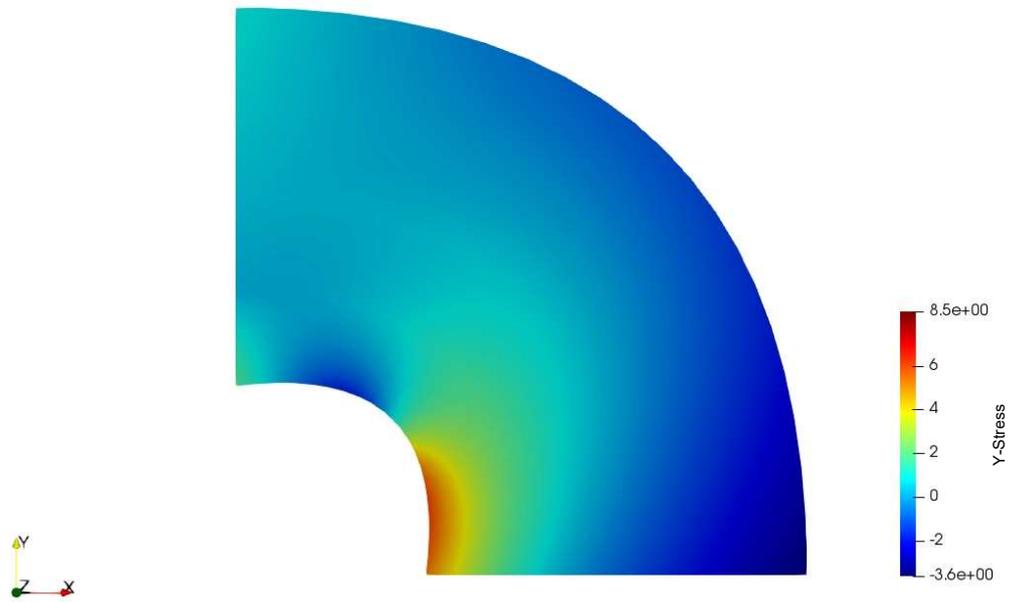
(h) The deformation distance 1.5mm
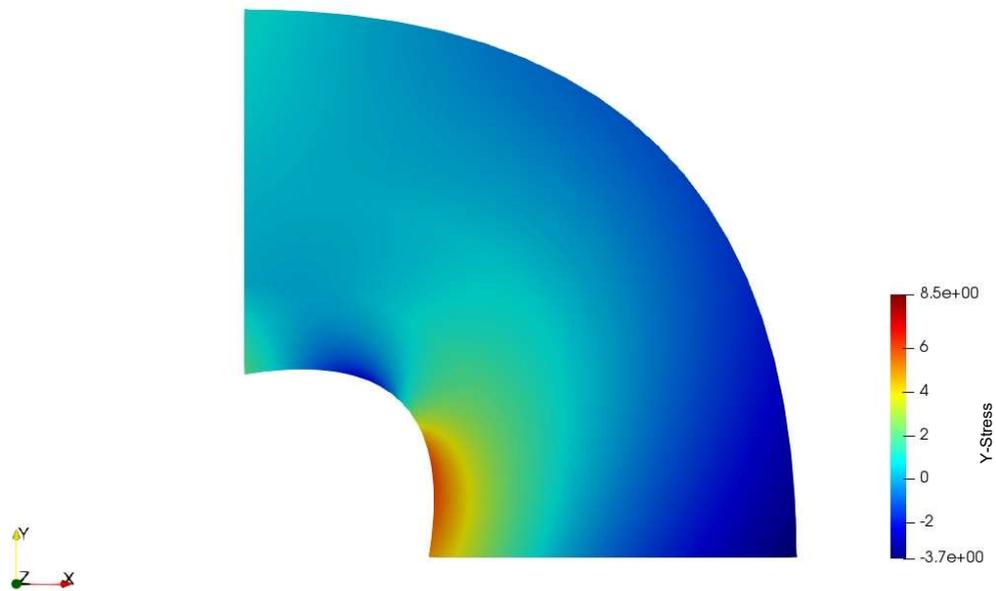
(i) The deformation distance 2.0mm

Figure 5.18 The corresponding displacement results of IGA
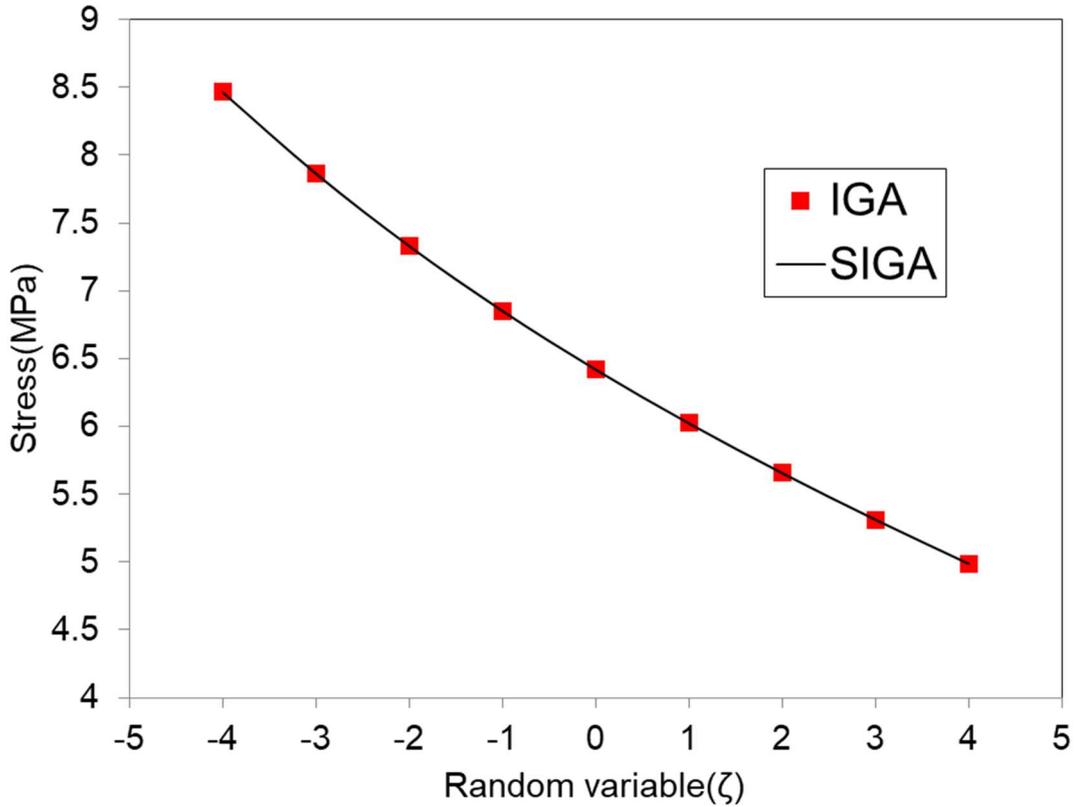
Figure 5.19 Stress response obtained by SIGA and numerical results of IGA

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.20, we present the stress probability density functions for the control point 3 in the y-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples. The average running time of the program for MSC is 4940.52 second. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables into the stress response equation. The average running time of the program is the 1.236 second. The comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS, but at a much lower computational cost. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.5. It can be easily seen that they are almost equal. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).
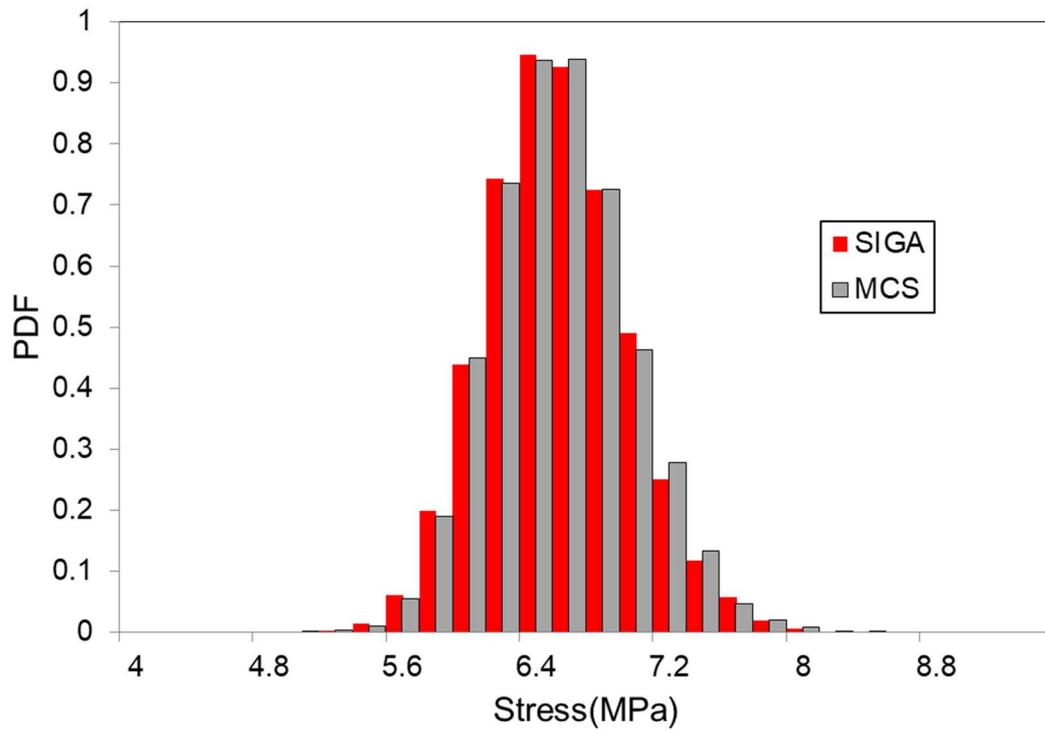
Figure 5.20: Stress probability density functions for the control point 3 in the y-direction

Table 5.5 Statistics of the Stress

	SIGA	MCS	PCE
MEAN	6.4363	6.4440	6.4637
STDEV	0.4164	0.4183	0.4179

The results in this case provide a possibility for uncertainty estimation of the response using the proposed method. And we also can find that the stress for the

control point 3 in the y-direction is reduced when the radius of point 2 becomes larger, the stress is reduced from 8.5Mpa to 5Mpa.

5.2. Example 2: localized corrosion

The second example involves a potential application of this method to the problem of uncertainty in shape caused by localized corrosion. The controllability and scalability of proposed method are demonstrated by this example. The relevant setup of the boundary conditions and material properties is illustrated in Figure 5.21(a), where the analysis model was subjected to a uniform tensile force ($T_x = 100\text{N/mm}^2$) in the x-direction on the right edge. The corresponding control net and control points is shown in Figure 5.21 (b) where L_y^m and L_y^n are the deviation lengths of the control points m and n in the y-direction, respectively.
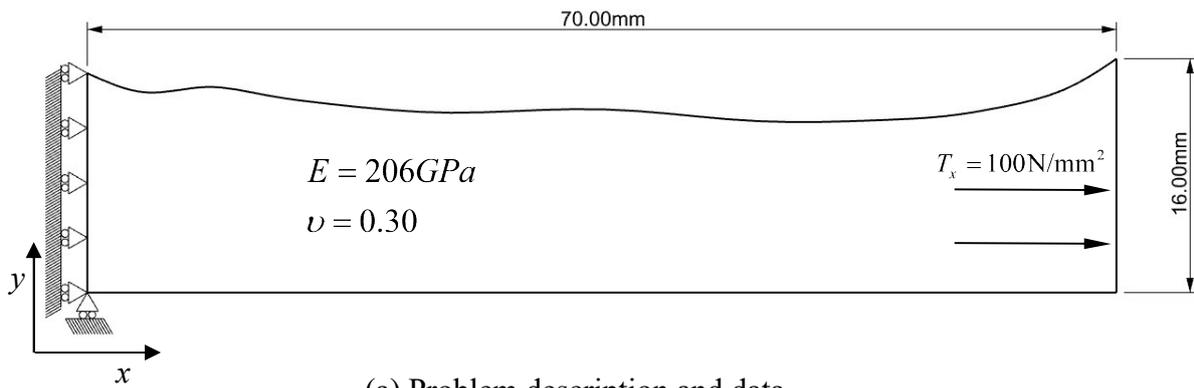
(a) Problem description and data.
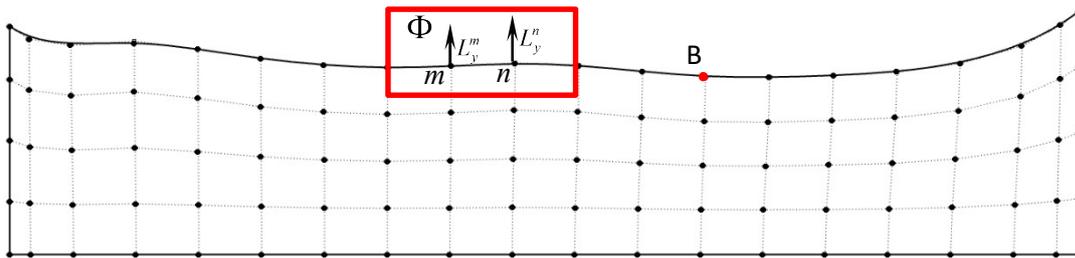
(b) Control net and control points. Rectangular region indicates location where localized corrosion phenomena occurred.

Figure 5.21: Localized corrosion problem

In this example, we assumed that localized corrosion phenomena appeared on the surface of region Φ shown in Figure 5.21 (b). The uncertainty in shape in region Φ

can be represented through the introduction of the uncertainty parameters into the control points m and n, since these two control points have the ability to directly change the geometry of this area (see Figure 5.21 (b)). The corresponding stochastic parameters are shown in Table 4, where acceptable different values are assigned to the mean and standard deviation for the y-coordinate of control points m and n, respectively.

Table 5.6: Stochastic parameters (L_y^m and L_y^n) for the region Φ with uncertainty

| y-direction	Mean	Standard deviation
L_y^m (control point *m*)	0	0.5
L_y^n (control point *n*)	0	0.6

In what follows, we present the displacement nephogram under different deformation distance in Figure 5.22 and the shape variables of the red box in figure B can be observed. It is obtained by using IGA methods. In Table 5.7, we present the coefficients of displacement response obtained from proposed method in the x-direction for the position of B point. By these coefficients, we computed the displacement response surface shown in Figure 5.23 and denoted by the solid grey line, while the red-dots denote the reference values obtained by determinate IGA method. by comparing the displacements obtained by SIGA and IGA. It can be easily seen that the analysis results of SIGA are in good agreement with the reference values.

*Table 5.7: Coefficients of displacement response, (a_i).*

| Order(*i*th)	0	1	2	3
a_i	5.1×10^{-2}	-1.6×10^{-3}	1.0×10^{-4}	-5.0×10^{-6}
4	5	6	7	8
2.0×10^{-7}	-6.8×10^{-9}	1.9×10^{-10}	-2.9×10^{-11}	2.0×10^{-12}
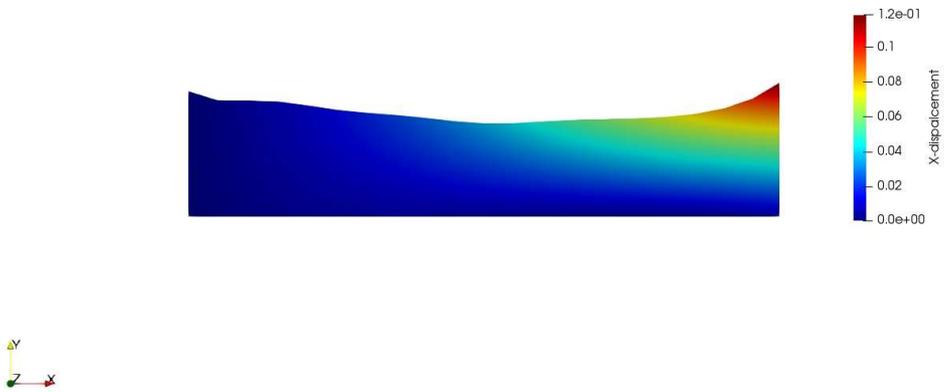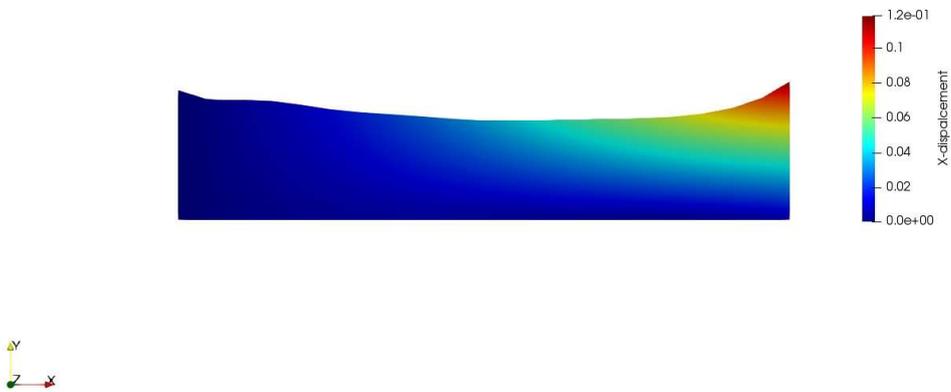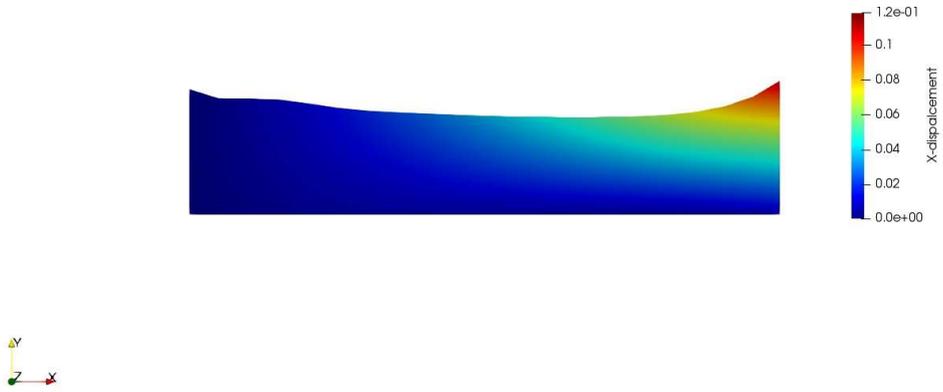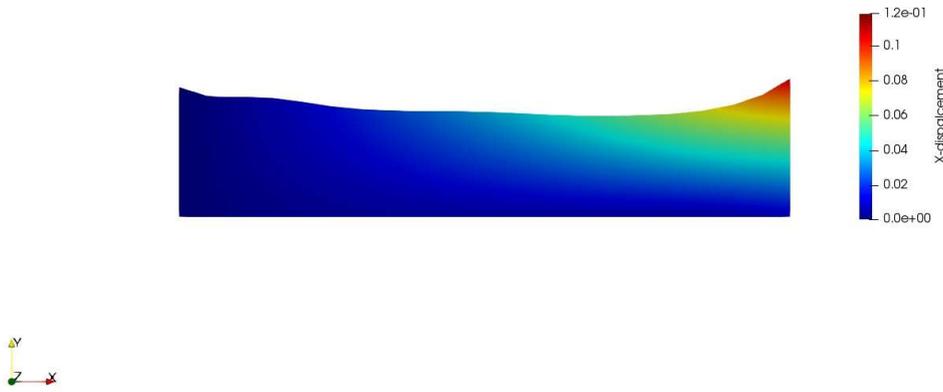
(a) The random variable ζ=-4
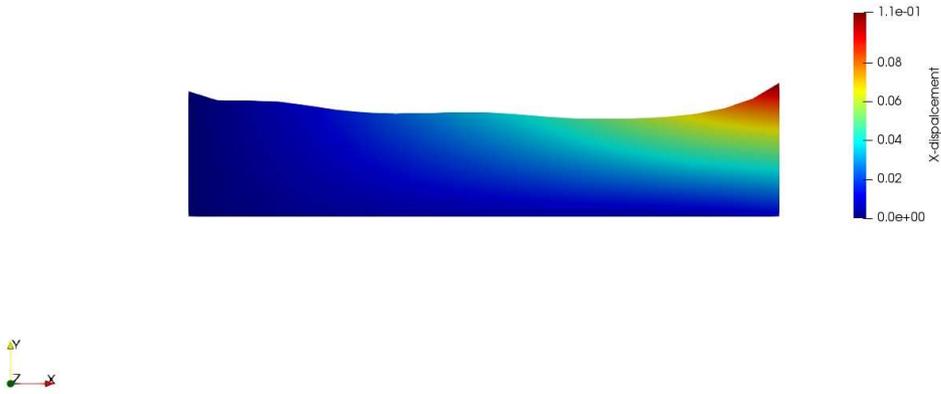
(b) The random variable ζ=-3
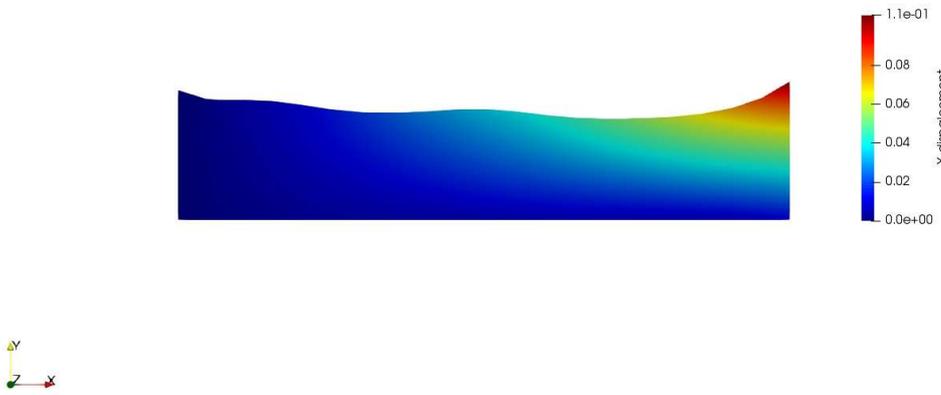
(c) The random variable ζ=-2


(d) The random variable ζ=-1
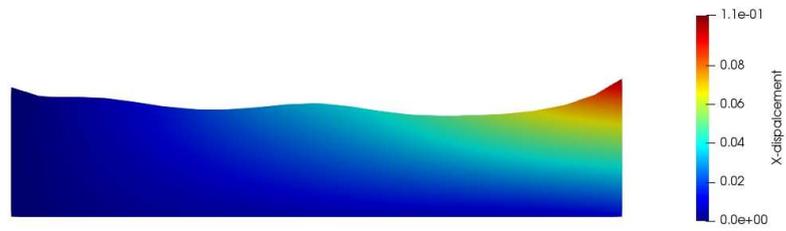
(e) The random variable ζ=0
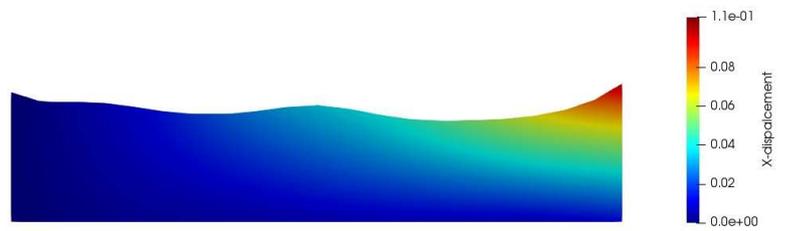
(f) The random variable ζ=1

(g) The random variable ζ=2

(h) The random variable ζ=3
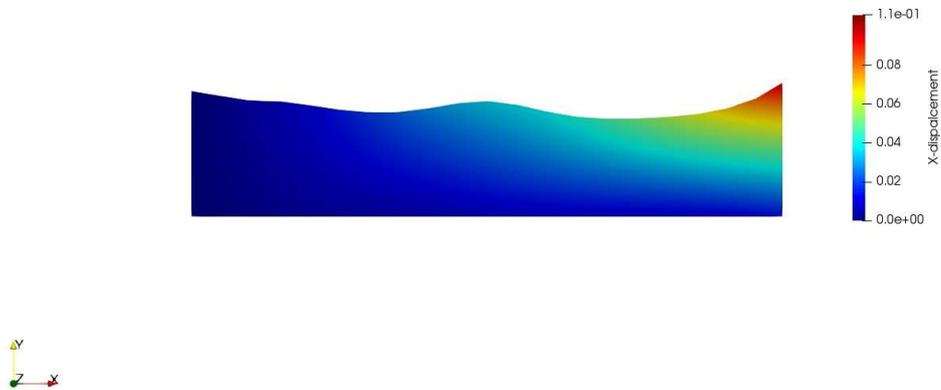
(i) The random variable ζ=4

Figure 5.22 The corresponding displacement results of IGA
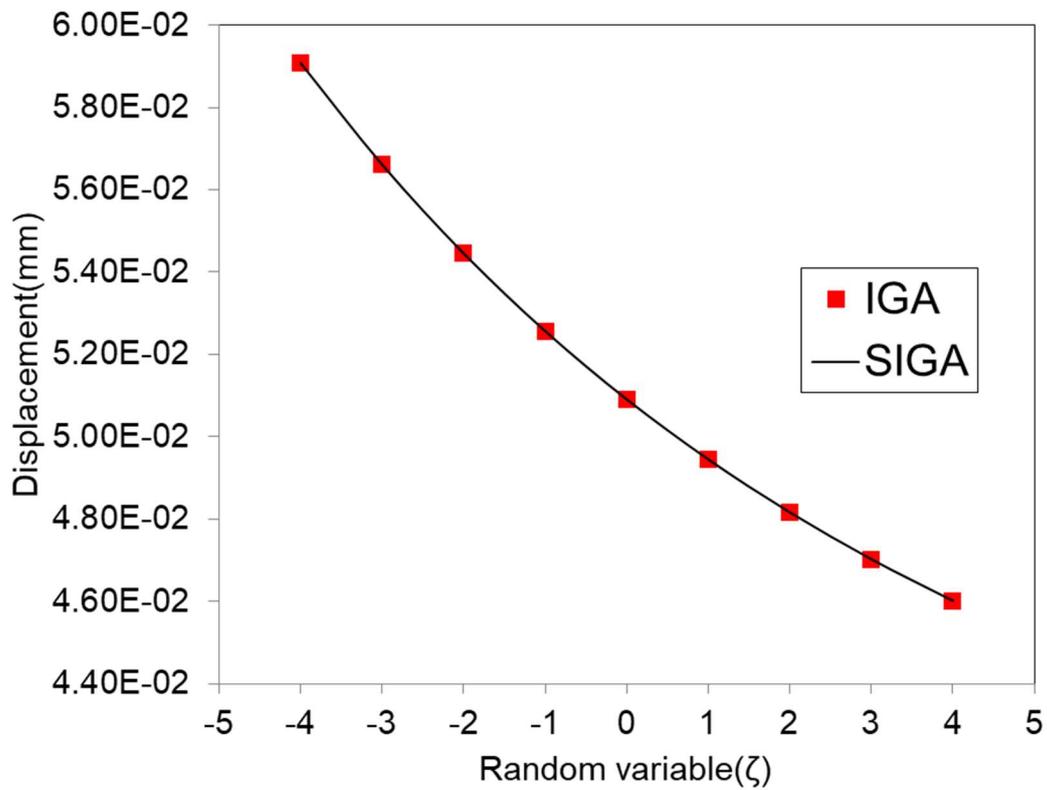
Figure 5.23: Displacement response from SIGA and numerical results of IGA

Furthermore, in Figure 5.24, we provide the probability density functions of displacement in the x-direction in the position of B point, they were obtained by MCS and SIGA methods, respectively. The results of MCS obtained by running for 10,000 iterations are denoted by the grey-column in Figure 5.24. The results of SIGA were obtained by generating 10,000 random variables ζ in the same Gaussian random field as MCS, and then introducing the response surface. In comparison to MCS, the performance of SIGA was better in terms of computational efforts. Moreover, the mean and standard deviation for MCS，SIGA and PCE are also shown in Table 5.8, where it can be seen that their degree of dispersion is almost the same.

This example demonstrates flexibility of the proposed method. In the SFEM analysis framework, we may need to consider the deformation of the mesh and possible interference between the adjacent nodes, because these factors can affect the accuracy of the analytical results. The SIGA overcame these problems, we could choose the analysis positions and range of model freely and flexibly according to the actual situation during the process of analysing uncertainty. Especially, for dealing with complicacy geometry entity, this flexibility and efficiency is even more important.
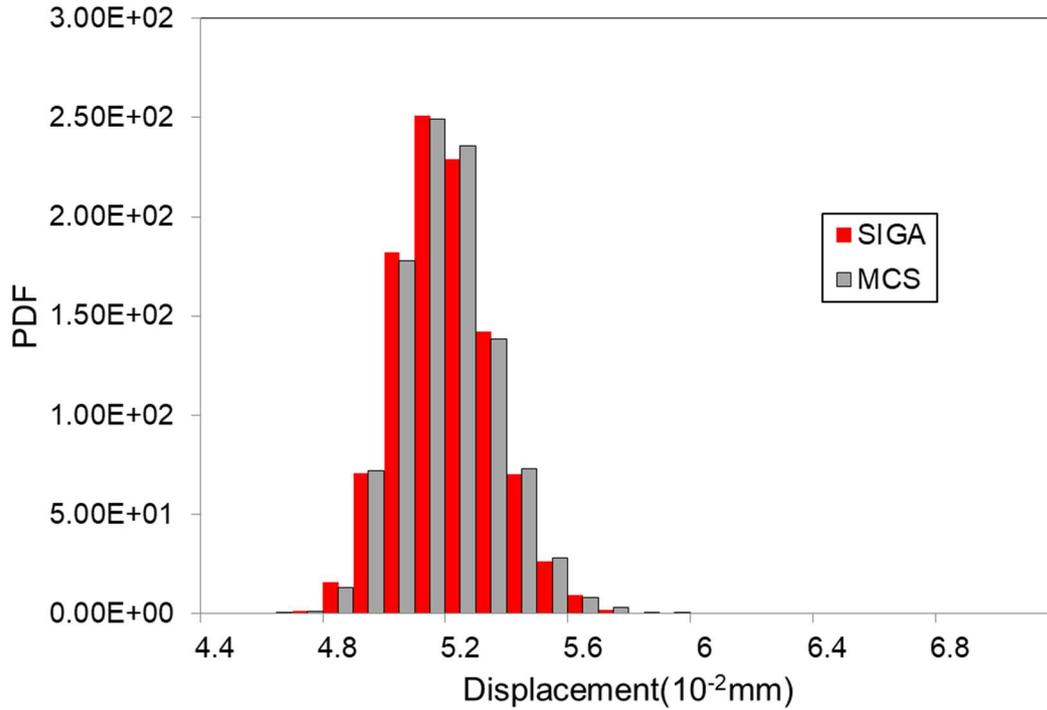
Figure 5.24: Displacement probability density functions for B point position in x-direction. Mean and standard deviation for MCS ，SIGA and PCE.

Table 5.8 Statistics of the Displacement

	SIGA	MCS	PCE
MEAN(10^{-2})	5.1008	5.1037	5.1058
STDEV(10^{-3})	1.5680	1.5775	1.5750

Similarly, in the next part, we will give the stress situation for the point B in the x-direction. In order to demonstrate the accuracy of the analysis results obtained by the proposed method for calculating stress, the results of SIGA and IGA were compared, as shown in Figure 5.25, which shows that the proposed method had very high

precision. We can find that when the random variable changes from - 4 to 4, the stress along the X axis of point B tends to increase and it also shows a linear change.
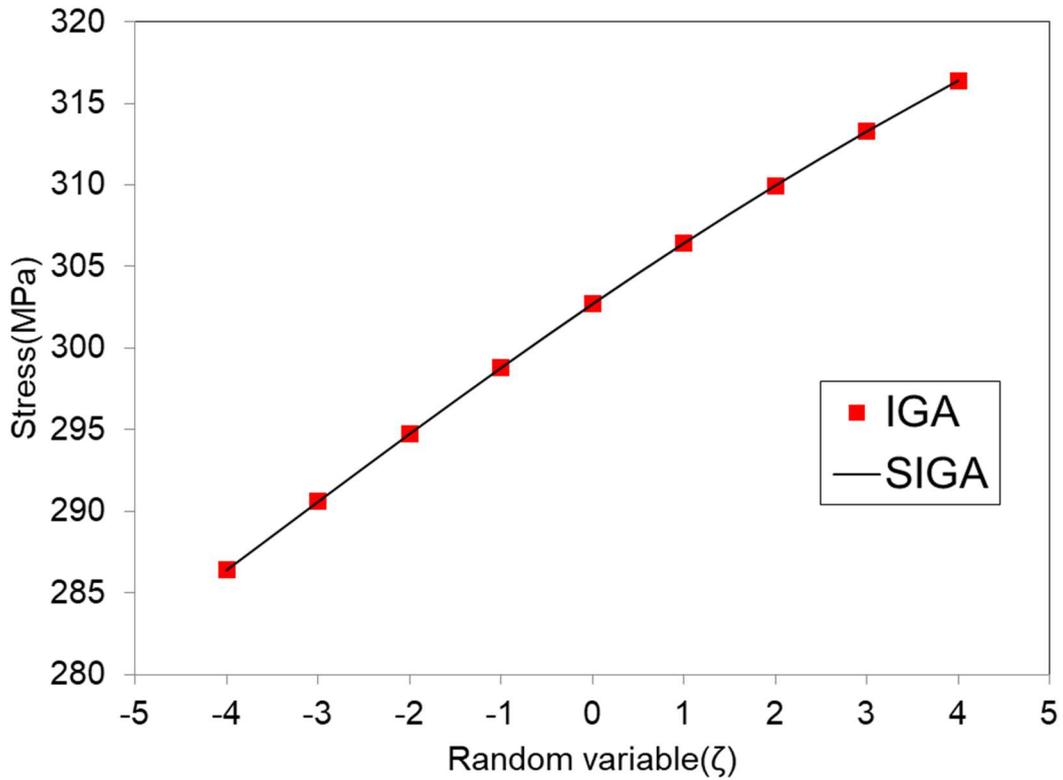
Figure 5.25: Stress response from SIGA and numerical results of IGA

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.26, we present the stress probability density functions for the control point B in the x-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples. The average running time of the program for MSC is 28424.1 second. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables into the stress response equation. The average running time of the program is the 4.977 second. The

comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS, but at a much lower computational cost. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.9. It can be easily seen that they are almost equal. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).
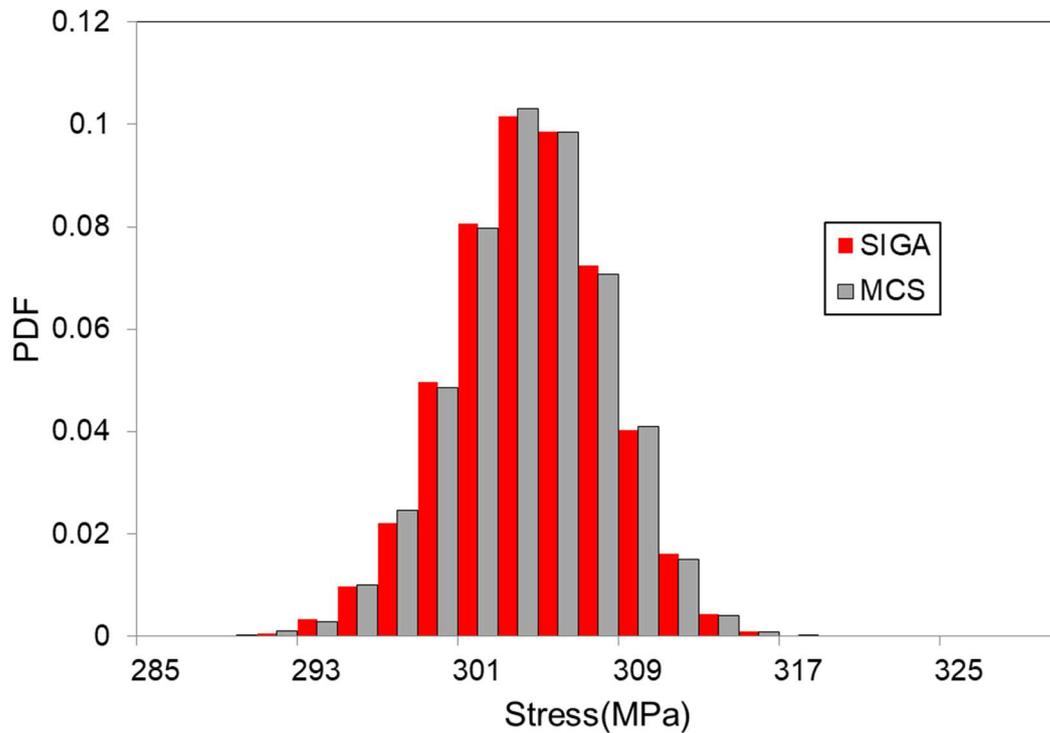
Figure 5.26: Stress probability density functions for B point position in x-direction.

Table 5.9 Statistics of the Stress

	SIGA	MCS	PCE
MEAN	302.61	302.54	302.66
STDEV	3.8053	3.8144	3.8153

The results in this case provide a possibility for uncertainty estimation of the response using the proposed method. And we also can find that the stress for the control point B in the x-direction is increased with the deepening of corrosion depth.

5.3. Example 3: A Plate with a Circular Hole

As a multi-patch example, we present a verification example with regard to quarter-circular a plate with a circular hole shown in Figure 5.27. According to the properties of multi-patch, we can find that quarter-circular a plate with a circular hole model is actually composed of two patches. The analytical model is designed based on NURBS geometric modeling tool in CAD. The relevant model design dimensions are shown in Figure 5.28, where the radii of the plate with a circular hole are 1mm, and the length and width of the plate with a circular hole are 5mm. In addition, the setup of the static elasticity problem is also illustrated in Figure 5.28, the hash marks on the bottom edge and right edge denote the symmetry boundary condition. The plate was subjected to a uniform tension (P_y) on its upper-edge. Additionally, the material properties of the plate with a circular hole Poisson's ratio υ and elastic coefficient E are also marked in Figure 5.28, respectively. The NURBS base used in shape design and analysis is constructed from the knot vector Ξ and Π in ξ- and η-direction, respectively, as follows

$$\Xi = \{0,0,0,0.5,1,1,1\}$$
$$\Pi = \{0,0,0,0.5,1,1,1\}$$

The control network of the geometric model is shown in Figure 5.29.

In this two-dimensional example, we assumed that the changes in shape take place in the inner circle. As illustrated in Example 1, moving any control point around the circular hole can change the shape of the circular hole. In this example, as shown in Figure 5.30, we move point 1 to change the shape of the circular hole. Here, we imported the uncertainty parameters into the coordinate of control point 1 in order to intuitively manipulate the shape of the inner circle, as Figure 5.30 while L_x denote the deviation length of control point 1 in the x-direction. The corresponding mean and standard deviation of L_x in control point 1 were set to 0 and 0.12, respectively.
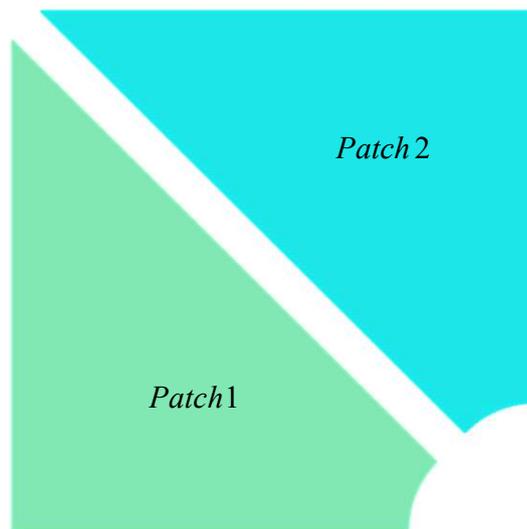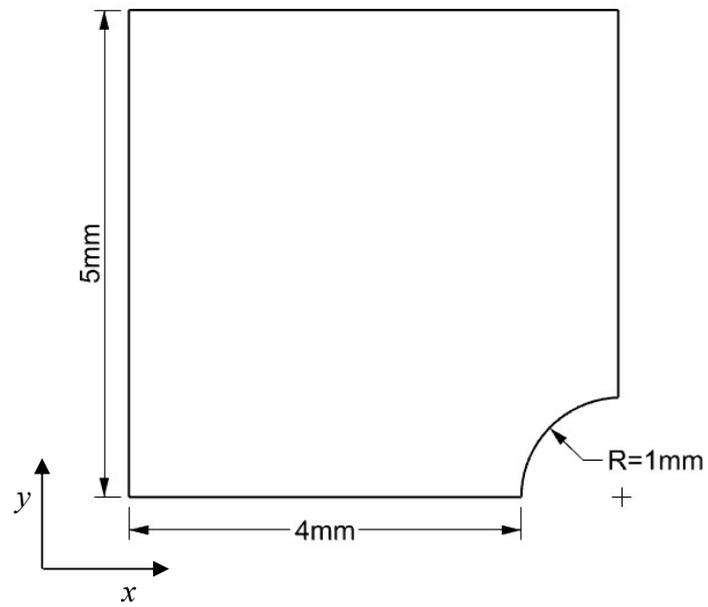
Figure 5.27 The geometry of a plate with a circular hole.
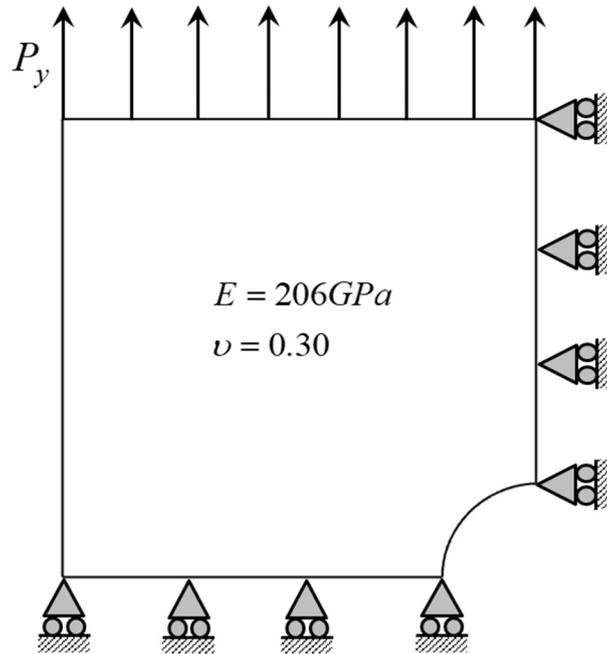
Figure 5.28 The geometry of a plate with a circular hole with material properties,

boundary conditions and uniform pressure ($P_y = 70 \text{N/mm}^2$)
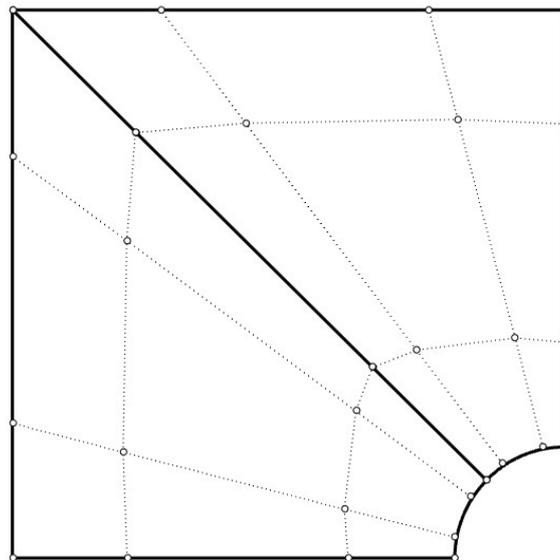
Figure 5.29   Control points and control net of the plate with a circular hole
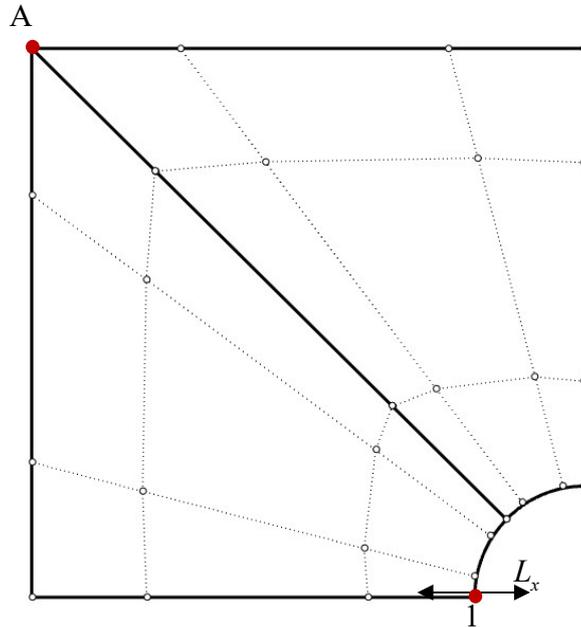
Figure 5.30 Moving the control points 1, on the NURBS geometry

In this example, we present the analysis results for the displacement response surface of the position of control point A and stress response surface of the position of control point 1 in y-direction, which is a stress concentration position, and compare them to the analysis results obtained by the MCS and the IGA methods, respectively.

First, similarly, in the following, we give the displacement results under different deformation conditions of IGA. In this case, the stochastic shape at the circular hole is arbitrary and controllable by introducing stochastic parameters to the corresponding control points. As shown in Figure 5.30, the stochastic parameters are introduced into coordinate of control point 1 in the x directions to represent the uncertainty. According to the stochastic parameters (the mean and standard deviation were 0 and 0.12, respectively) setting of Example 3, in the Figure 5.31, we respectively give stochastic shapes with the stochastic variables of -4, -3, -2, -1, 0, 1, 2, 3 and 4. That is the deformation distances are -0.48mm, -0.36mm, -0.24mm, -0.12mm, 0mm, 0.12mm, 0.24mm, 0.36mm, 0.48mm, respectively. According to the figure, we can find that the deformation of the circular hole is quite obvious when we move the control point A in the x-direction. And in order to demonstrate the accuracy

of the analysis results obtained by the proposed method, the results of SIGA by using the response surface Equation (5.2) and IGA were compared, as shown in Figure 5.32, which shows that the proposed method had very high precision. We can find that when the random variable changes from - 4 to 4, that is, the deformation distance changes from – 0.48mm to 0.48mm, the displacement along the Y axis of point A tends to increase, and it tends to be flat.
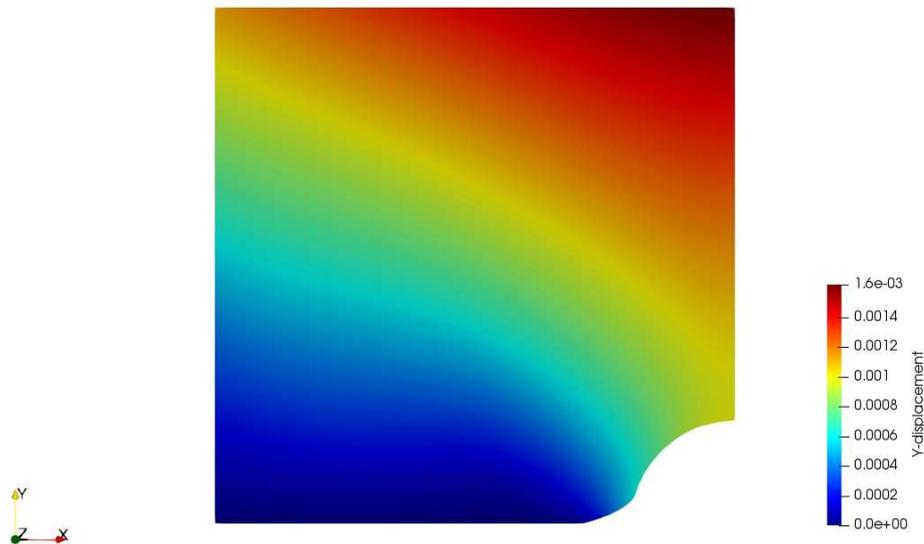
(a) The deformation distance -0.48mm
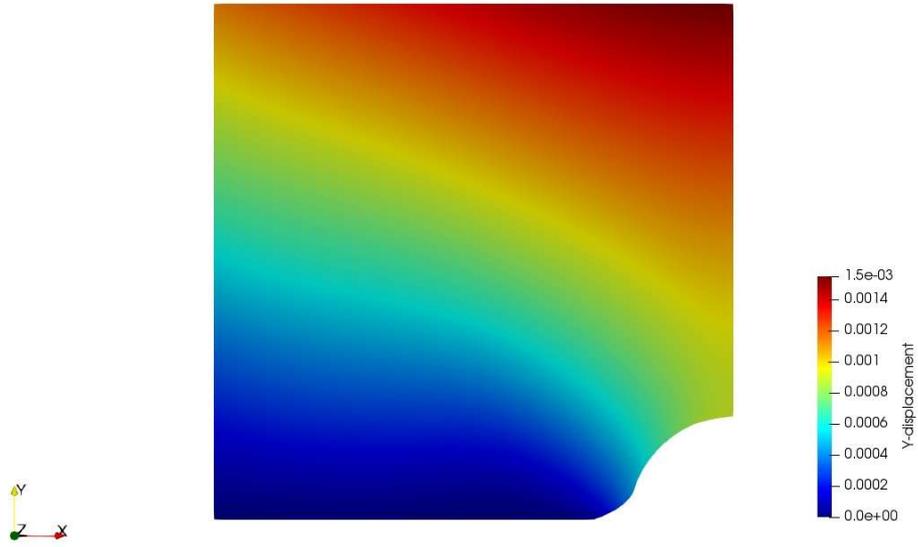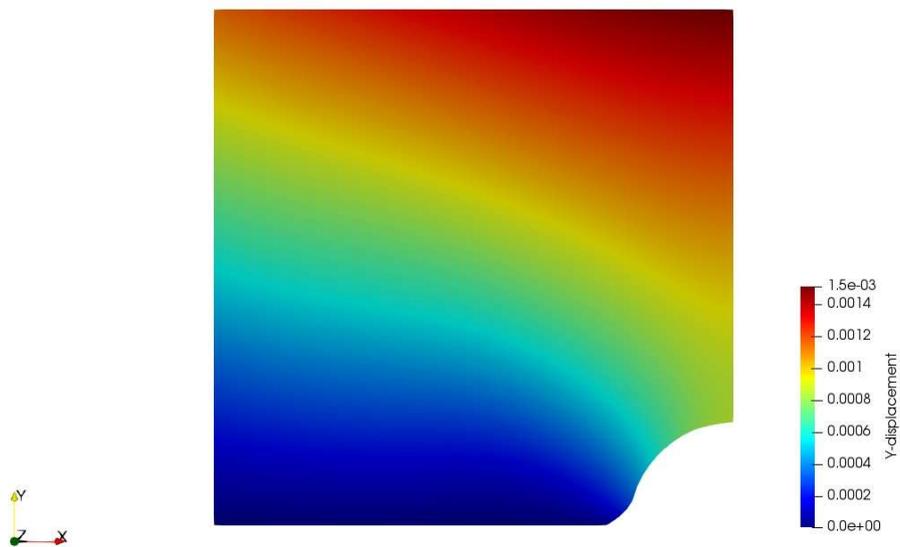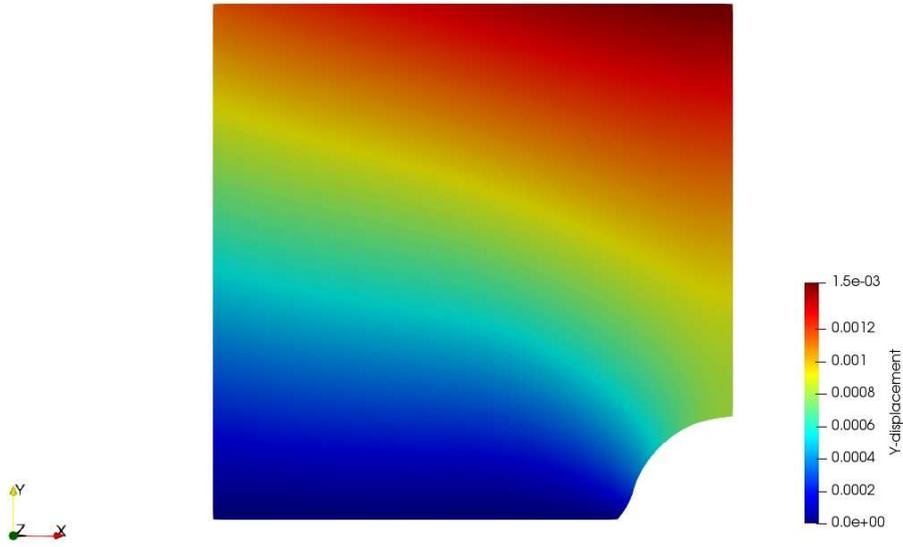
(b) The deformation distance -0.36mm
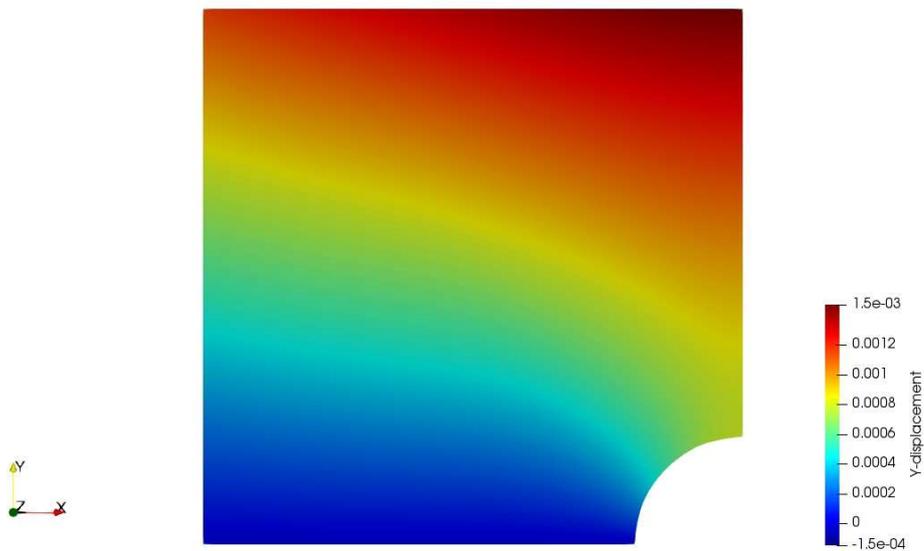
(c) The deformation distance -0.24mm

(d) The deformation distance -0.12mm
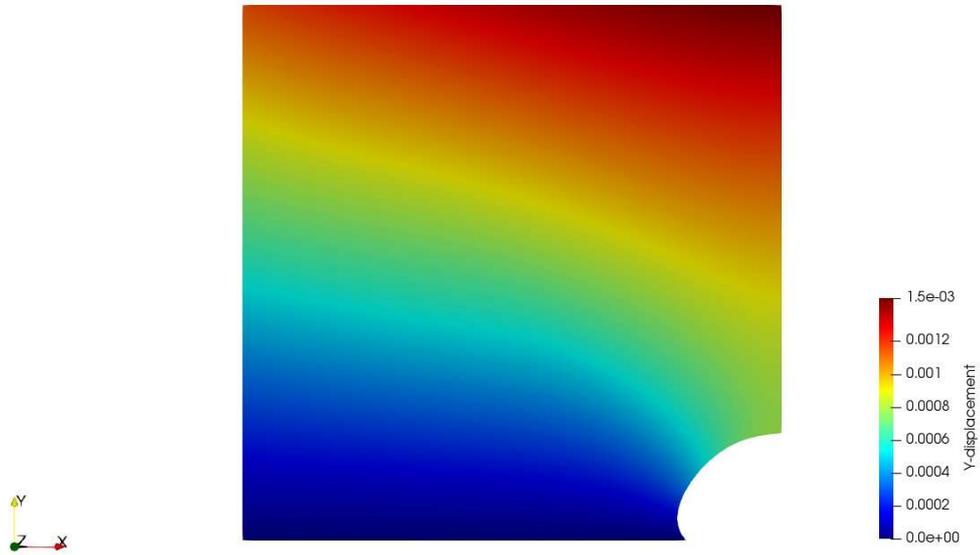
(e) The deformation distance 0.0mm
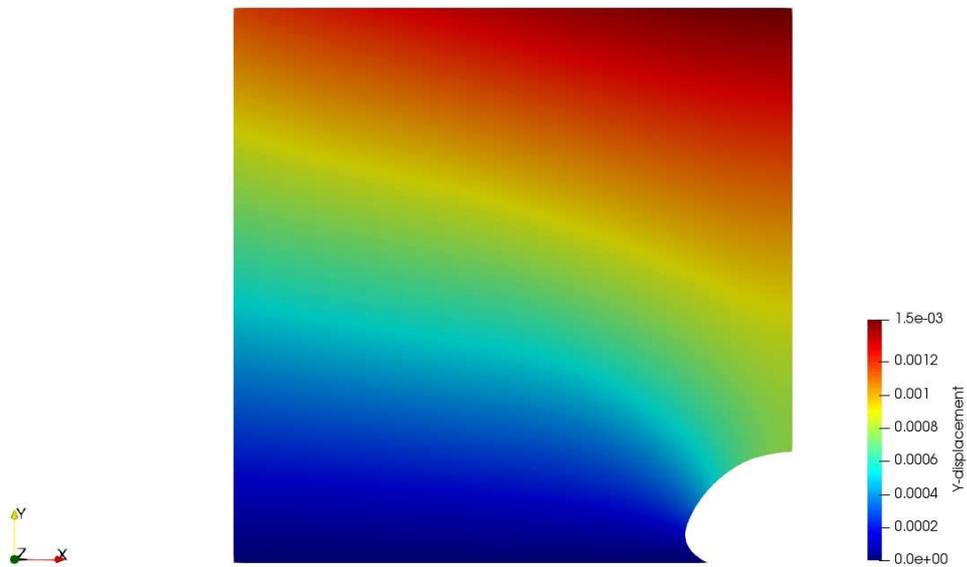
(f) The deformation distance 0.12mm

(g) The deformation distance 0.24mm
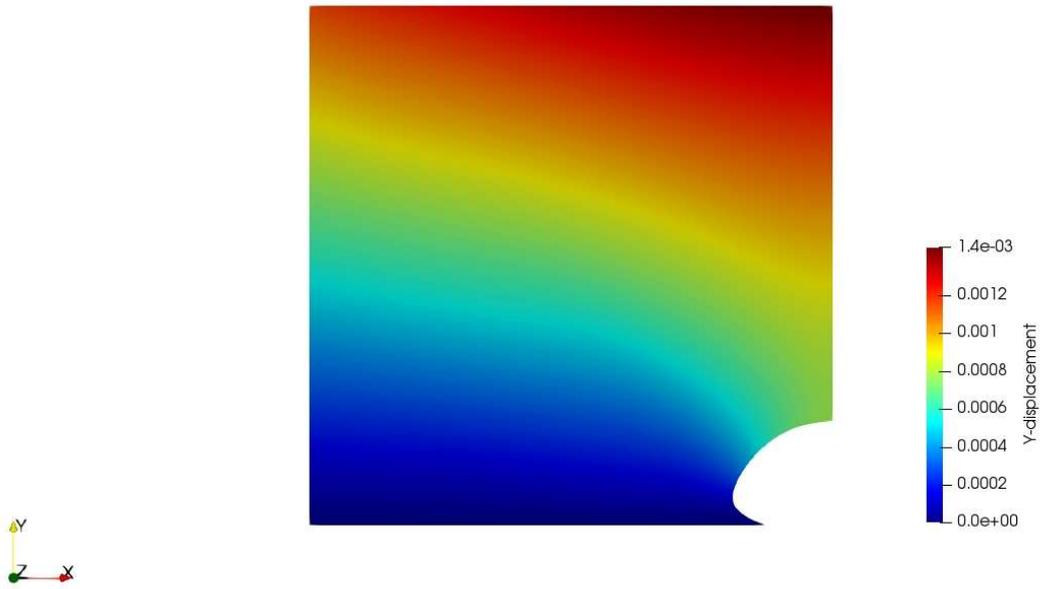
(h) The deformation distance 0.36mm

(i) The deformation distance 0.48mm

Figure 5.31 The corresponding displacement results of IGA

$$d(\zeta) = 1.14 \times 10^{-3} \Psi_0(\zeta) + 1.99 \times 10^{-6} \Psi_1(\zeta) -$$
$$5.28 \times 10^{-7} \Psi_2(\zeta) + 9.48 \times 10^{-8} \Psi_3(\zeta) - 1.22 \times 10^{-8} \Psi_4(\zeta) \qquad (5.2)$$
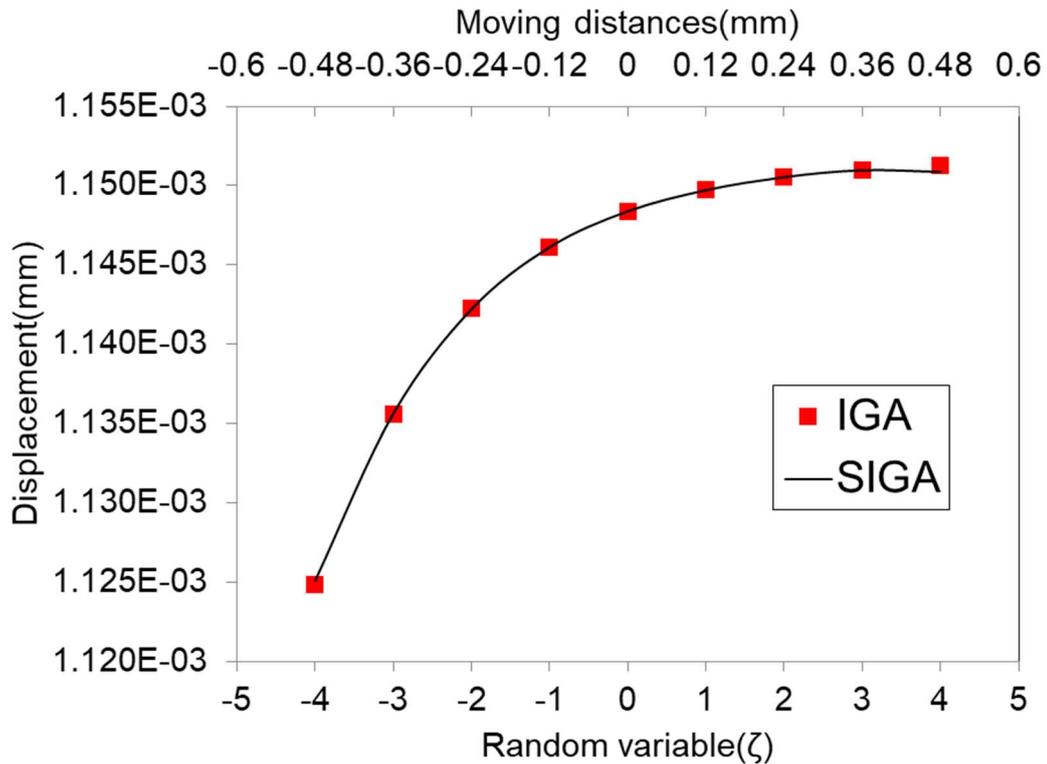
Figure 5.32 Displacement response obtained by SIGA and numerical results of IGA

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.33, we present the displacement probability density functions for the control point A in the y-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables ζ into the Equation (5.2). The comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS to some extent because we can see that the probability density function of the right part of the mean has a slight error. We

investigate the error and find that when the random variable ζ is greater than 0, the change rate of the response surface becomes very small (see Figure 5.32). That is to say, when the deformation of the circular hole continues to increase in the direction of radius, the change of displacement at point A is very small. So the response surface has a different trend. This is taken into account that the probability density function begins to show errors when it is larger than the average value. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.10. It can be found that the mean value is almost equal but the standard deviation of MCS is slightly larger than that of SIGA. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).

Table 5.10 Statistics of the displacement

	SIGA	MCS	PCE
MEAN (10^{-3})	1.1479	1.1479	1.1479
STDEV (10^{-6})	2.1166	2.1567	2.1359
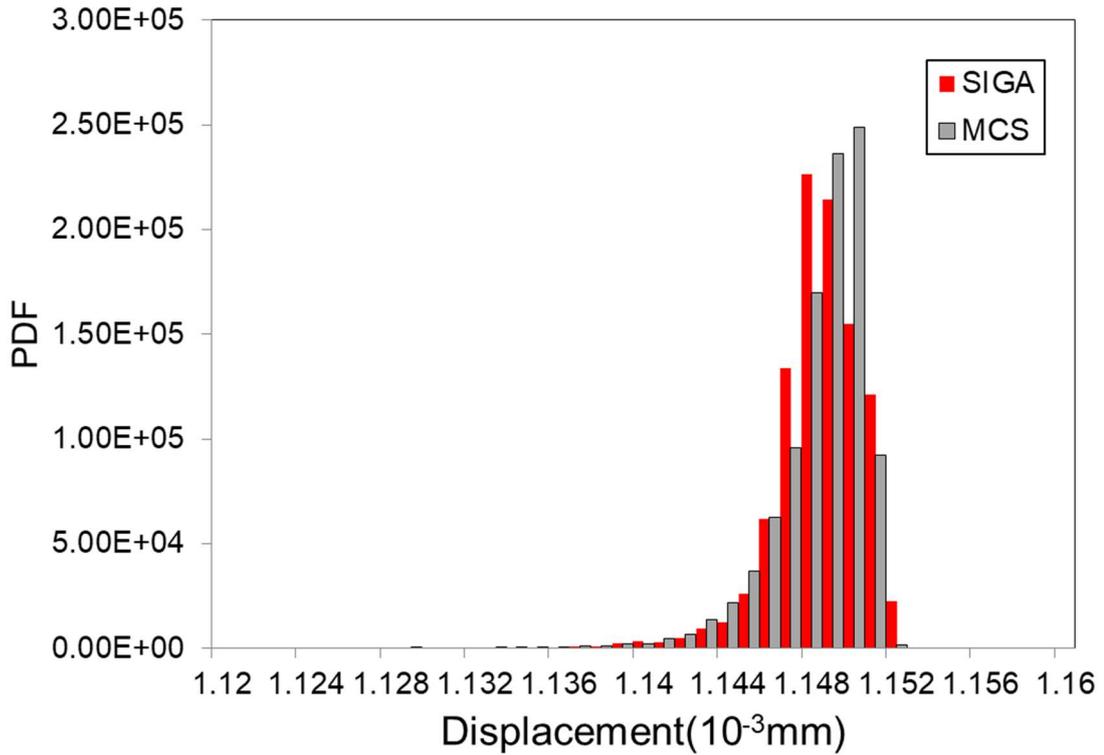
Figure 5.33: Displacement probability density functions for the control point A in the x-direction

Similarly, in the next part, we will give the stress situation for the control point 1 in the y-direction. In the Figure 5.34, we respectively give stochastic shapes with the stochastic variables of -4, -3, -2, -1, 0, 1, 2, 3 and 4, respectively. Comparing with Figure 5.34, we can find that the maximum stress point moves upward from point 1 as the random variable increases. That is, with the change of the circular hole, the maximum stress point also changes.

And in order to demonstrate the accuracy of the analysis results obtained by the proposed method, the results of SIGA and IGA were compared, as shown in Figure 5.35, which shows that the proposed method had very high precision. But we can find that when the random variables are -4 and 4, the results of the two methods are not very consistent. Observing Figure 5.34(a), (i), the circular hole deformation in these two cases is very large, and when the random variable is 4(the deformation distance 0.48mm), the stress value of point1 is negative in the y direction. We

consider that the method has some errors in accuracy for relatively large deformation. However, we consider that the probability of occurrence of the random variable 4 and -4 is almost absent, so it does not have any influence on the probability characteristics.
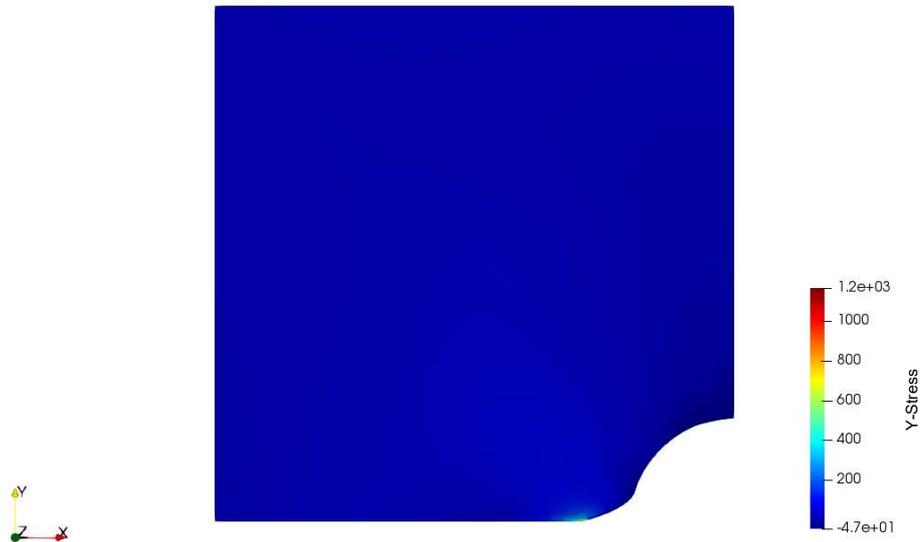
(a) The deformation distance -0.48mm
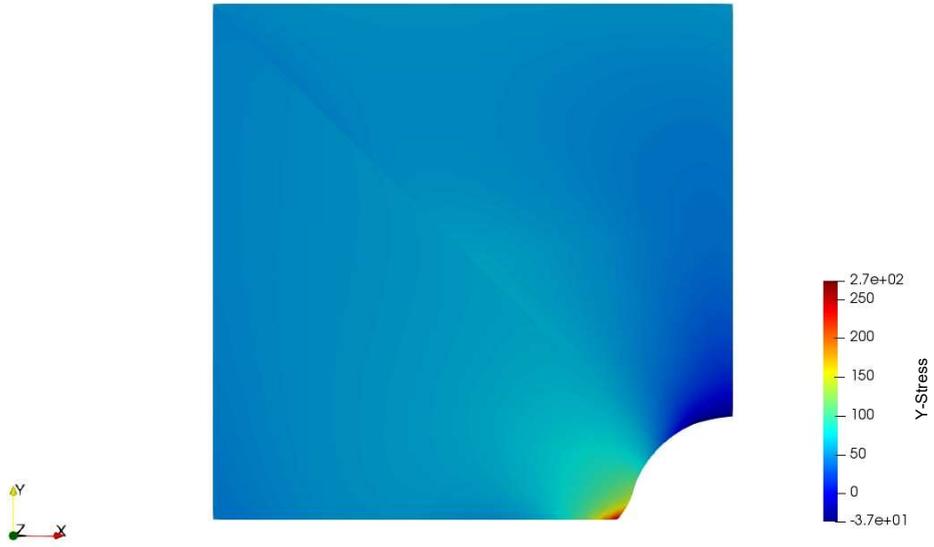
(b) The deformation distance -0.36mm
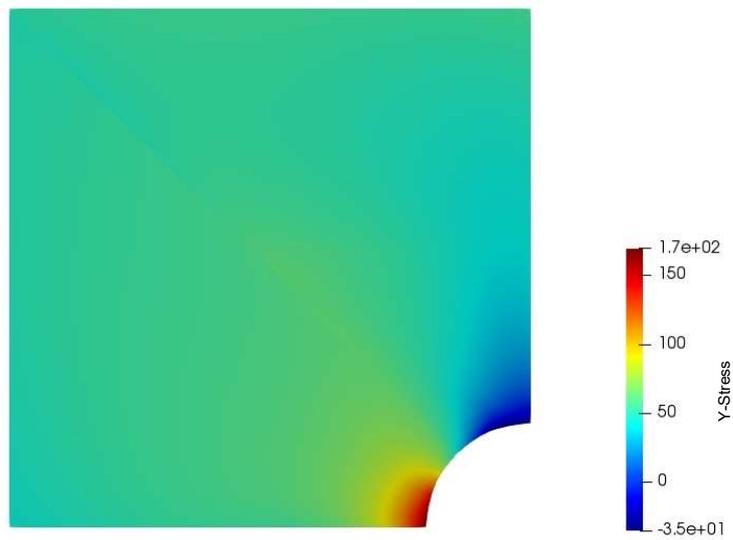
(c) The deformation distance -0.24mm

(d) The deformation distance -0.12mm
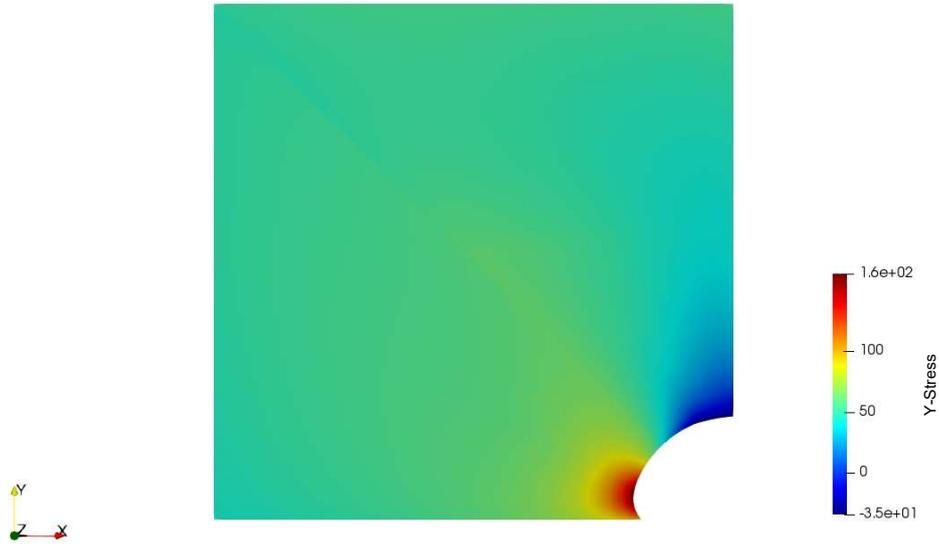
(e) The deformation distance 0.0mm
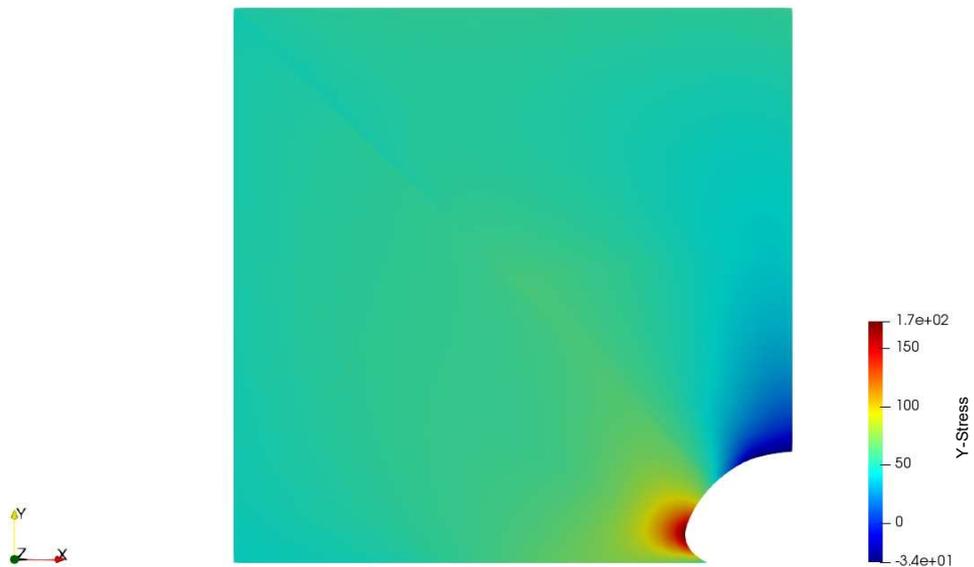
(f) The deformation distance 0.12mm

(g) The deformation distance 0.24mm
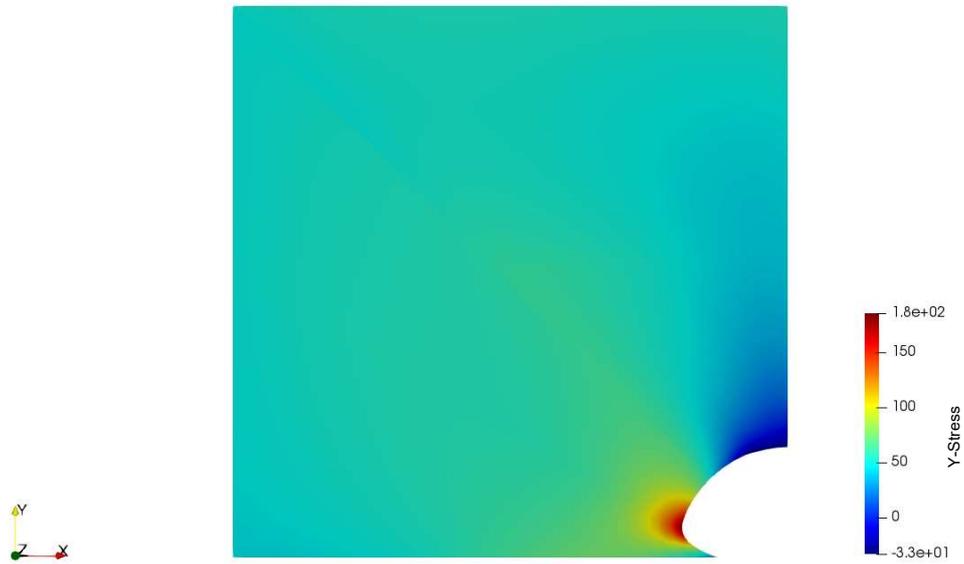
(h) The deformation distance 0.36mm
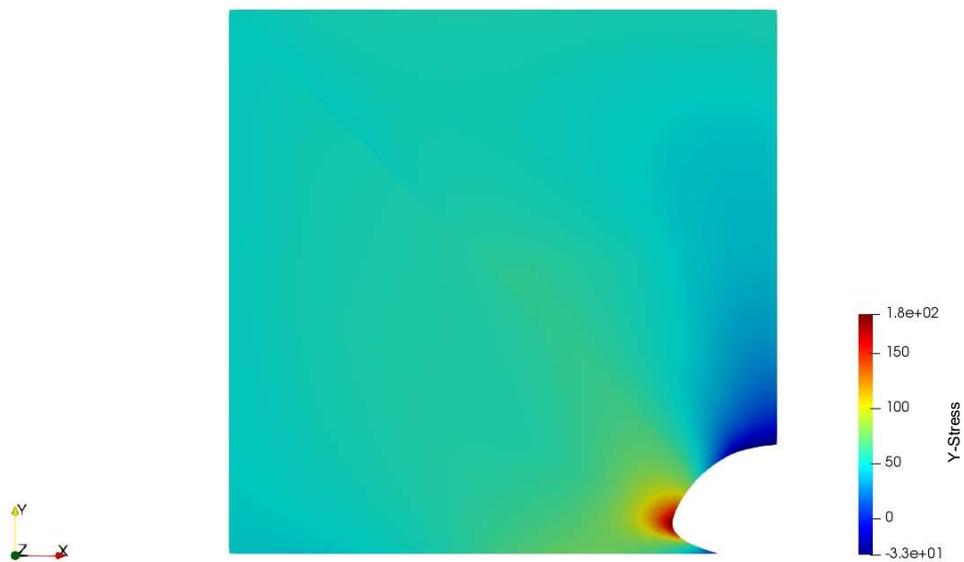
(i) The deformation distance 0.48mm

Figure 5.34 The corresponding displacement results of IGA
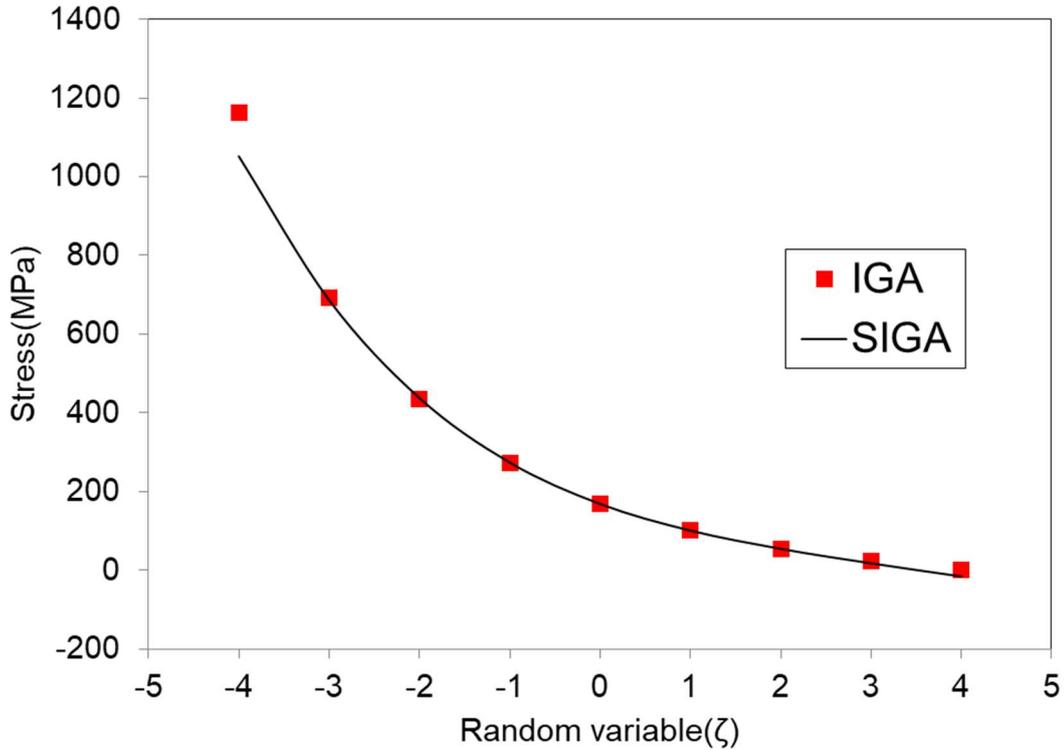
Figure 5.35 Stress response obtained by SIGA and numerical results of IGA

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.36, we present the stress probability density functions for the control point 1 in the y-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples. The average running time of the program for MSC is 6828.59 second. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables into the stress response equation. The average running time of the program is the 1.126 second. The comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS, but at a much lower computational cost. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.11. It can be easily seen that they are almost equal. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).
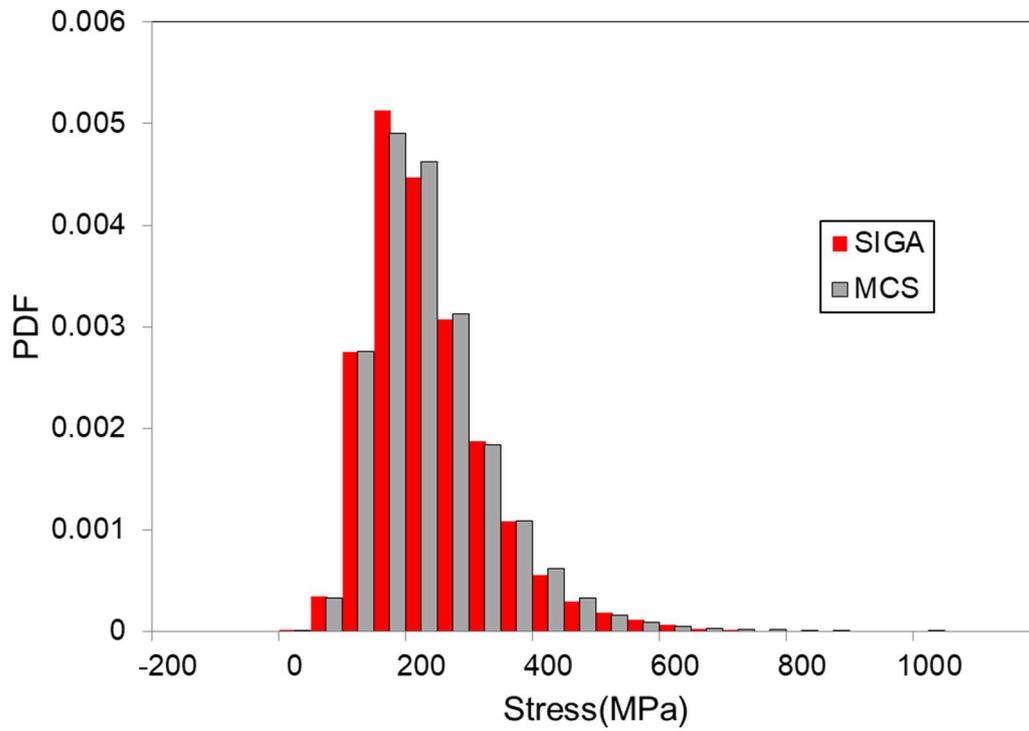
Figure 5.36: Stress probability density functions for the control point 1 in the y-direction

Table 5.11 Statistics of the Stress

	SIGA	MCS	PCE
MEAN	187.19	189.05	187.42
STDEV	95.886	97.843	97.128

The results in this case provide a possibility for uncertainty estimation of the response using the proposed method. And we also can find that the stress for the

control point 1 in the y-direction is reduced when the radius of point 1 becomes small, the stress is reduced from 1000Mpa to 0Mpa because the position of stress concentration changes with the deformation of circular hole.

5.4. Example 4: A Butt Joint

As other multi-patch example, we present a verification example with regard to a butt joint shown in Figure 5.37. According to the properties of multi-patch, we can find that a butt joint model is actually composed of three patches. The analytical model is designed based on NURBS geometric modeling tool in CAD. The relevant model design dimensions are shown in Figure 5.37, where the length and width of the butt joint are 55mm and 10mm, respectively. In addition, the setup of the static elasticity problem is also illustrated in Figure 5.38, the hash marks on the left edge denote the symmetry boundary condition and the point at the lower left is fixed. The plate was subjected to a uniform tension (P_x) on its right-edge. Additionally, the material properties of the plate with a circular hole Poisson's ratio υ and elastic coefficient E are also marked in Figure 5.38, respectively. The NURBS base used in shape design and analysis is constructed from the knot vector Ξ and Π in ξ- and η-direction, respectively，as follows

$$\Xi = \{0,0,0,0.5,1,1,1\}$$
$$\Pi = \{0,0,0,0.5,1,1,1\}$$

The control network of the geometric model is shown in Figure 5.39.

In this two-dimensional example, we assumed that the changes in shape take place in the welding part, moving any control point around the welding part can change the shape of the welding part. In this example, as shown in Figure 5.30, we move point 1 to change the shape of the welding part. Here, we imported the uncertainty parameters into the coordinate of control point 1 in order to intuitively manipulate the shape of the welding part, as Figure 5.40 while L_y denote the deviation length of control point 1 in the y-direction. The corresponding mean and standard deviation of L_y in control point 1 were set to 0 and 0.5, respectively.
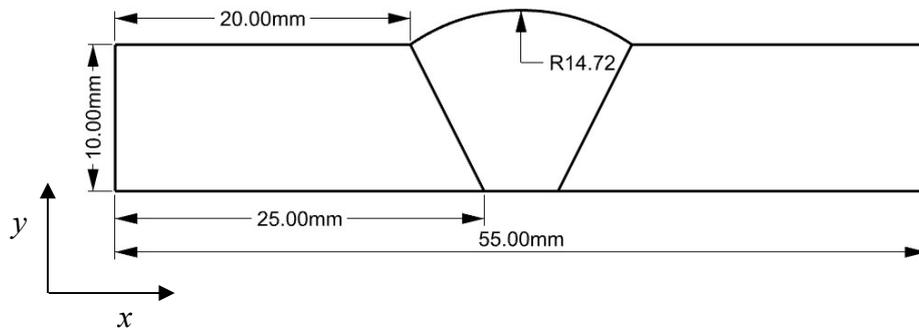
Figure 5.37 The geometry of a butt joint.

$$E = 206 GPa$$
$$\upsilon = 0.30$$

P_x
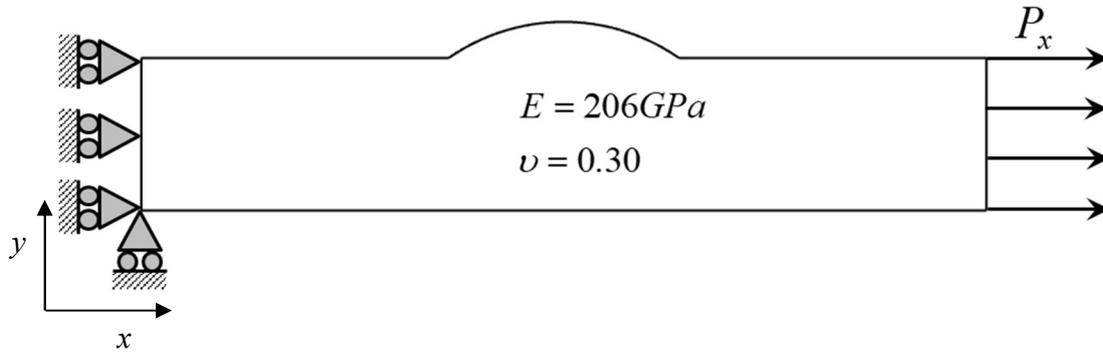
Figure 5.38 The geometry of the butt joint with material properties, boundary conditions and uniform pressure ($P_x = 50\,\text{N}/\text{m}\,\text{m}^2$)
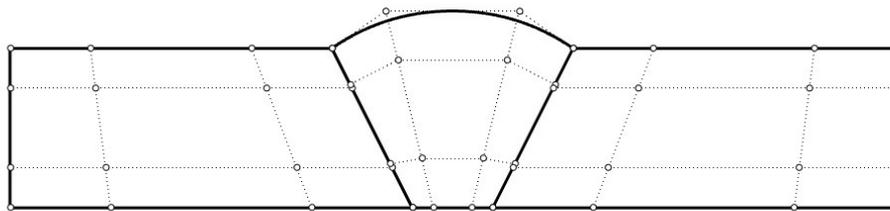
Figure 5.39 Control points and control net of the butt joint
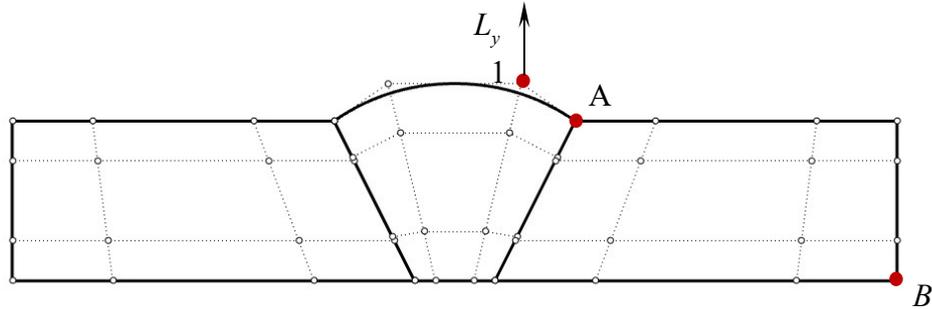
Figure 5.40 Moving the control points 1, on the NURBS geometry

In this example, we present the analysis results for the displacement response surface of the position of control point B and stress response surface of the position of control point A in x-direction, which is a stress concentration position, and compare them to the analysis results obtained by the MCS and the IGA methods, respectively.

First, similarly, in the following, we give the displacement results under different deformation conditions of IGA. In this case, the stochastic shape at the butt joint is arbitrary and controllable by introducing stochastic parameters to the corresponding control points. As shown in Figure 5.40, the stochastic parameters are introduced into coordinate of control point 1 in the y directions to represent the uncertainty. According to the stochastic parameters (the mean and standard deviation were 0 and 0.5, respectively) setting of Example 4. In the Figure 5.41, we respectively give stochastic shapes with the stochastic variables of -4, -3, -2, -1, 0, 1, 2, 3 and 4. That is the deformation distances are -2.0mm, -1.5mm, -1.0mm, -0.5mm, 0mm, 0.5mm, 1.0mm, 1.5mm, 2.0mm, respectively. According to these figures, we can find that the deformation of the welding part is quite obvious when we move the control point 1 in the y-direction. And in order to demonstrate the accuracy of the analysis results obtained by the proposed method, the results of SIGA by using the response surface Equation (5.3) and IGA were compared, as shown in Figure 5.42, which shows that the proposed method had very high precision. We can find that when the random

variable changes from - 4 to 4, that is, the deformation distance changes from - 2.0mm to 2.0mm, the displacement along the X axis of point A tends to increase, and it shows nonlinearity.

$$d(\zeta) = 1.39 \times 10^{-2} \Psi_0(\zeta) + 3.12 \times 10^{-5} \Psi_1(\zeta) -$$
$$3.27 \times 10^{-6} \Psi_2(\zeta) + 2.47 \times 10^{-7} \Psi_3(\zeta) - 1.45 \times 10^{-8} \Psi_4(\zeta) \qquad (5.3)$$
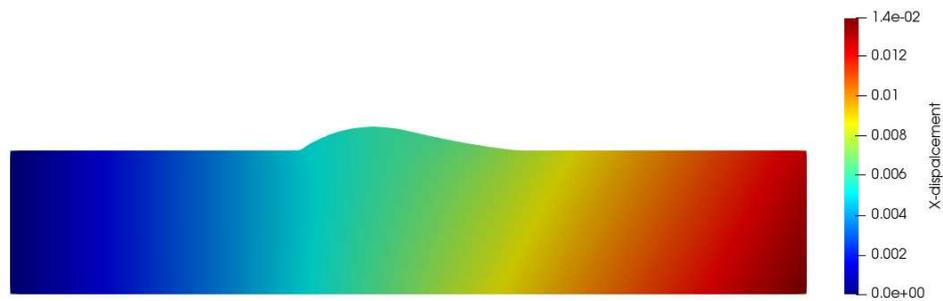
(a) The deformation distance -2mm
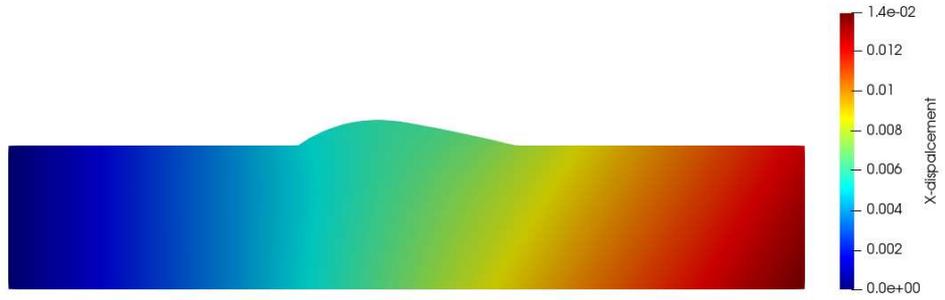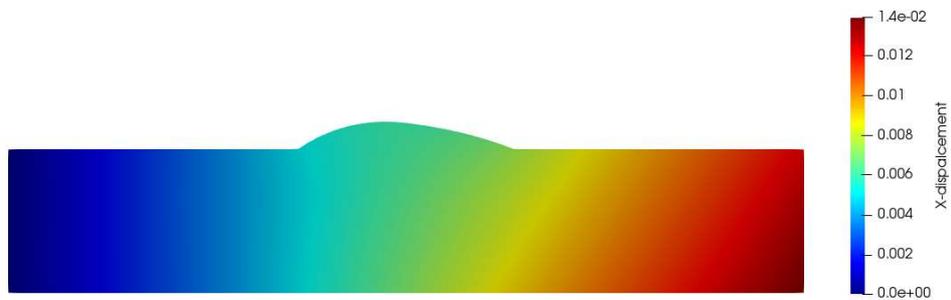
(b) The deformation distance -1.5mm
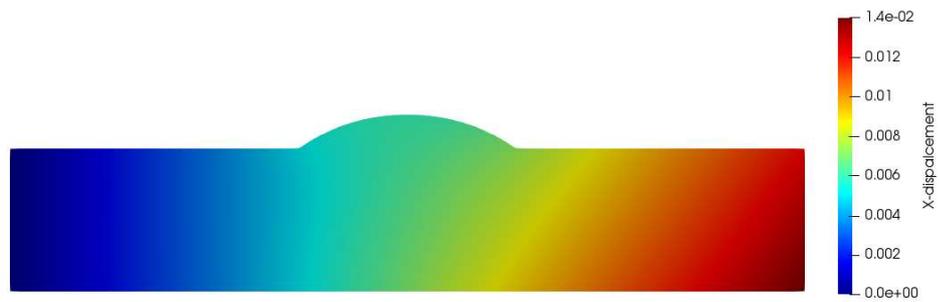
(c) The deformation distance -1.0mm

(d) The deformation distance -0.5mm

(e) The deformation distance 0.0mm
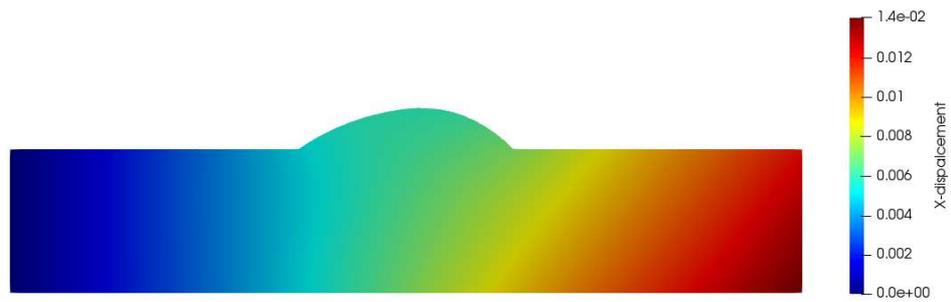
(f) The deformation distance 0.5mm

(g) The deformation distance 1.0mm
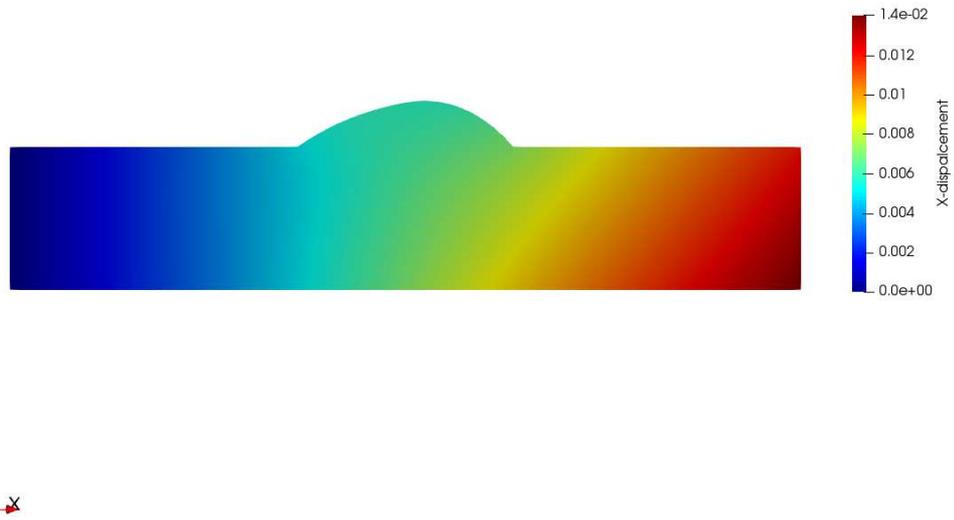
(h) The deformation distance 1.5mm
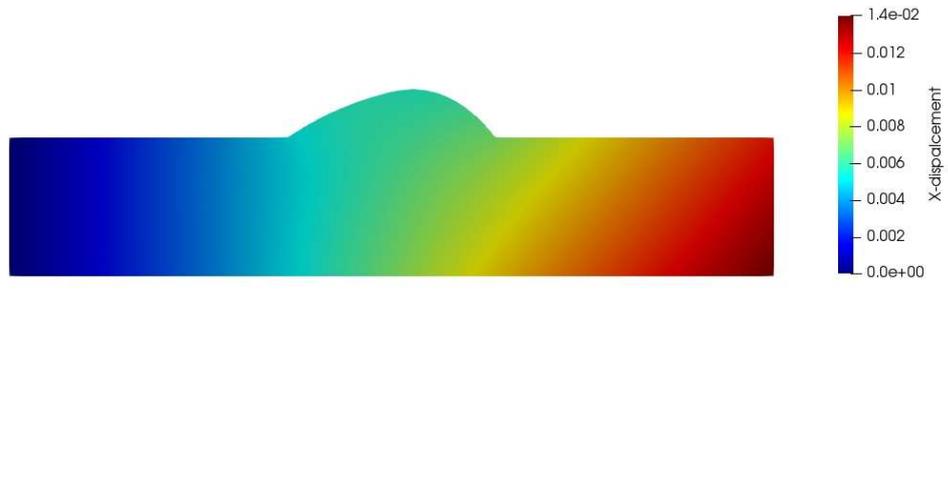
(i) The deformation distance 2.0mm

Figure 5.41 The corresponding displacement results of IGA
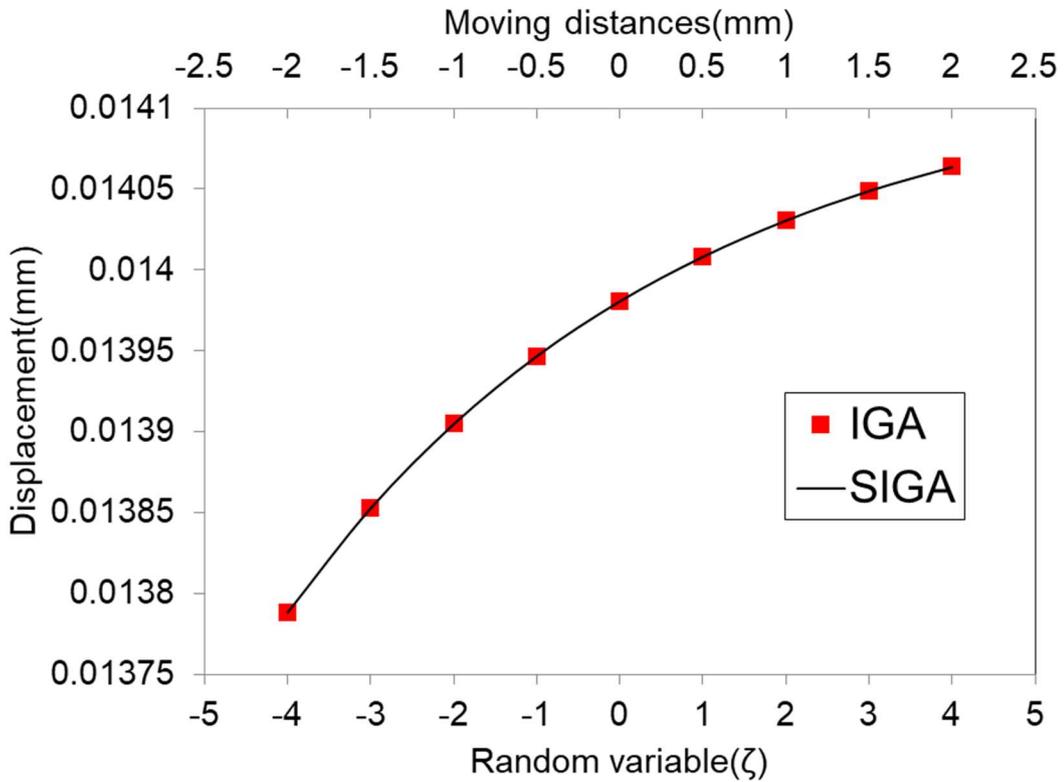
Figure 5.42 Displacement response obtained by SIGA and numerical results of IGA

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.43, we present the displacement probability density functions for the control point B in the x-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables ζ into the Equation (5.3). The comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.12. It can be found that the mean value is almost equal but the standard deviation of MCS is slightly larger than that of SIGA. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).

Table 5.12 Statistics of the displacement

	SIGA	MCS	PCE
MEAN (10^{-2})	1.3977	1.3977	1.3977
STDEV (10^{-5})	3.1362	3.1666	3.1563
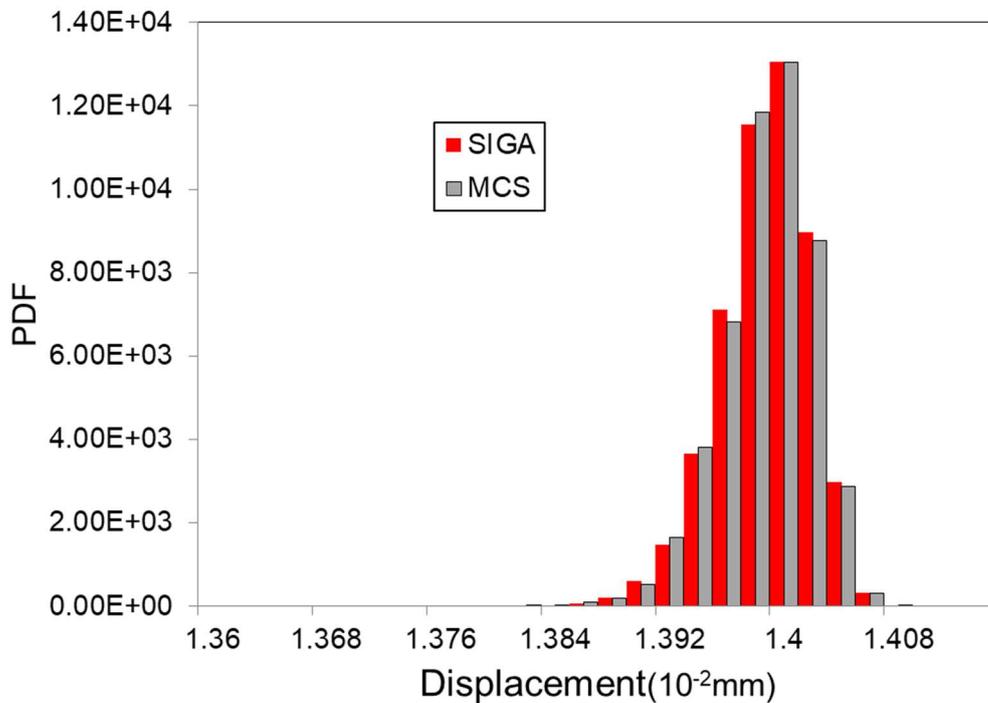
Figure 5.43: Displacement probability density functions for the control point B in the x-direction

And in order to demonstrate the accuracy of the analysis results obtained by the proposed method, the results of SIGA and IGA were compared, as shown in Figure 5.44, which shows that the proposed method had very high precision.

242
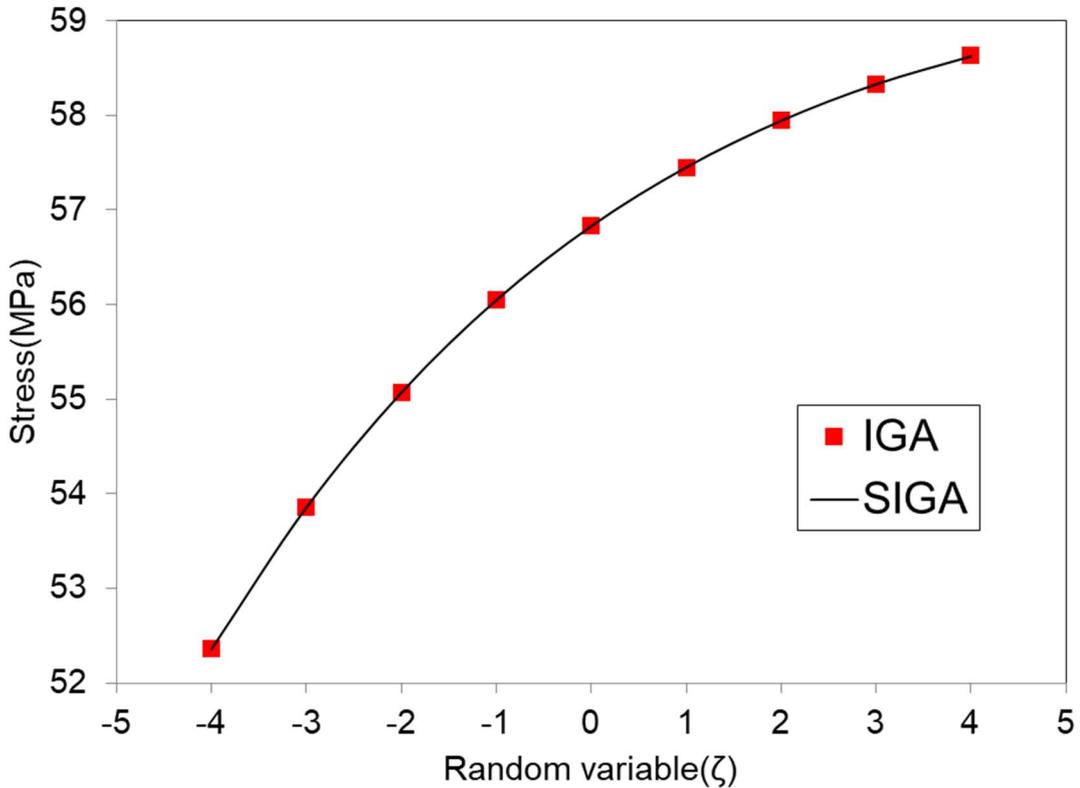
Figure 5.44 Stress response obtained by SIGA and numerical results of IGA

Additionally, the statistical characteristics of the proposed method were investigated by comparing with Monte-Carlo simulation. In Figure 5.45, we present the stress probability density functions for the control point A in the x-direction. The reference values (gray clustered column) were obtained by repeatedly performing Monte Carlo simulations with 10,000 model samples. The average running time of the program for MSC is 6967.7 second. For SIGA, the probability density function of the system was computed by substituting 10,000 random variables into the stress response equation. The average running time of the program is the 1.547 second. The comparison of these results to those obtained by MCS revealed that the probability distribution obtained by SIGA is almost the same as that of MCS, but at a much lower computational cost. Moreover, the mean and standard deviation from these two methods and PCE are listed in Table 5.13. It can be easily seen that they are almost equal. Note that the mean and standard deviation of PCE are derived by Equations (3.9) and (3.10).
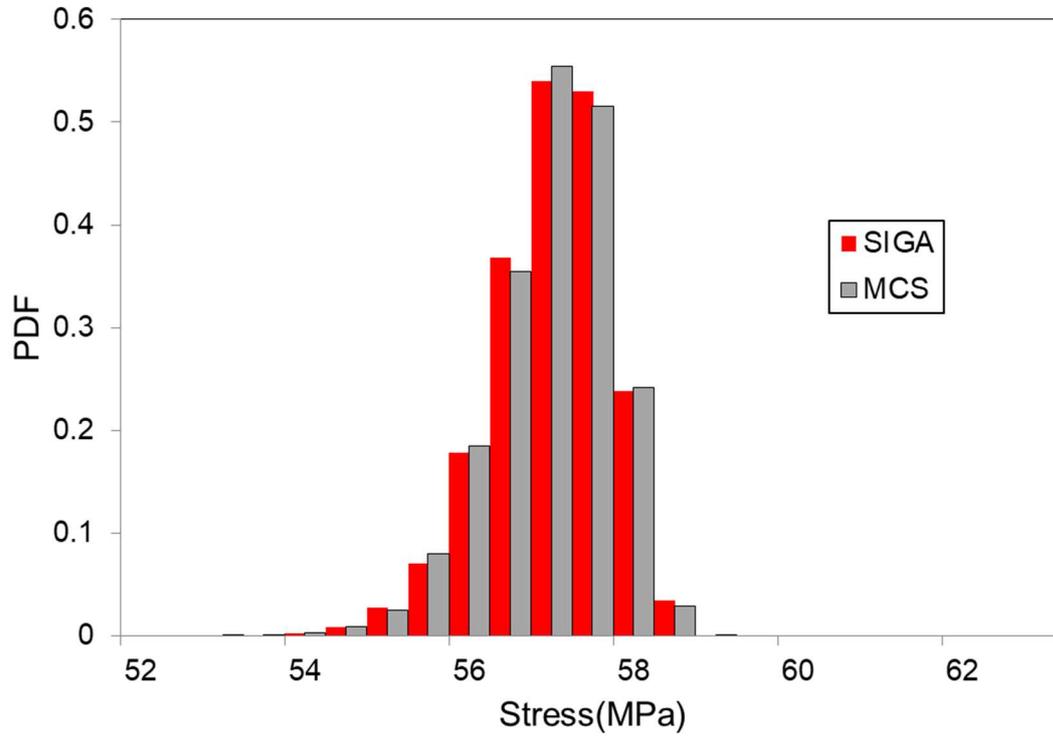
Figure 5.45: Stress probability density functions for the control point 1 in the y-direction

Table 5.13 Statistics of the Stress

	SIGA	MCS	PCE
MEAN	56.751	56.737	56.759
STDEV	0.7175	0.7249	0.7223

The results in this case provide a possibility for uncertainty estimation of the response using the proposed method. And we also can find that the stress for the control point 1 in the y-direction is reduced when the radius of point 1 becomes small,

the stress is reduced from 1000Mpa to 0Mpa because the position of stress concentration changes with the deformation of circular hole.

5.5. Conclusions

In this chapter, in order to demonstrate the applicability, accuracy and effectiveness of the proposed SIGA analysis scheme, for numerical examples are thoroughly explored within this section. Firstly, a numerical example for quarter-circular cantilever beam is given. In this example, the freshly proposed generalized stochastic isogeometric analysis based polynomial chaos expansion is rigorously verified against the well-established theoretical results. Subsequently, the proposed stochastic isogeometric analysis analysis framework is further implemented for the stochastic static analysis of a localized corrosion in the second example. In order to apply to more complex analytical models, we present examples of two multi-patch NURBS geometry models i.e. infinite plate with circular hole and butt joint, these ones fully demonstrate the ability of SIGA to handle multiple patch geometries. In addition, in these example, the validity and accuracy of the results are assessed by comparing them to the results obtained by Monte Carlo simulation (MCS) based on the IGA algorithm. In addition, we give the calculation time of each analytical example in detail. By comparison, we can see that the proposed method is superior to the traditional MCS method in terms of computational cost and efficiency.

6. CONCLUSION

This paper presented an innovative numerical method for estimating uncertainty in shape by conjugating the isogoemetric analysis framework and probability theory. In this study, the uncertainty in shape was estimated from a "physically-based" point of view, rather than from the classic "FEA-based" point of view, namely, the uncertainty in shape of structure model was represented by directly introducing stochastic parameters into the control points in the physical space. The proposed method effectively overcomes some of the classical SFEM. Despite classical SFEM has excellent analytical performance and a sound analytical system, there are still some inevitable limitations. Especially, in the uncertainty analysis in shape, because of what its use of a geometry approximated by a finite element mesh (FE-mesh), some of its innate disadvantages have been exposed. In many situations, this geometry approximated can cause errors in the analytical results during the performing uncertainty analysis on some complex and sensitive geometric structures. In order to overcome the shortcomings in the SFEM mentioned above, based on the natural characteristics of NURBS, we proposed the stochastic isogeometric analysis

method to deal with the problem of uncertainty in shape. Firstly, throughout the probabilistic analysis, an exact geometric entity was used to represent uncertainty in shape, which effectively reduced errors in terms of geometry. From the algebraic aspect, the NURBS basis function used for discretization is a smoother, highly continuous basis function, thus that greatly improved the accuracy and reliability of the analysis. Furthermore, unlike typically FE-mesh in the classical SFEM, usually, the NURBS uses the control points to manipulate geometric shape, and consequently, the problem of mesh repartitioning does not need to be considered, the costs and difficulty of the analysis were diminished, significantly. Secondly, based on excellent geometric properties of NURBS, the boundaries of the analytical region containing random field can be identified exactly, easily and flexibly. Finally, this study explored one possible extension of IGA in the field of computational stochastic mechanics, that is the extension of classical deterministic IGA into a probabilistic analytical framework to evaluate the uncertainty in shape.

Additionally, we elaborated on the formulation of SIGA, which is an intrusive formulation procedure. The deterministic isogoemetric analysis framework was rewritten as an uncertainty form based on PCE, and the orthogonal properties of PCE were fully utilised in order to solve the stiffness matrix. Ultimately In the formulation process, the global stiffness matrix is newly defined as a Hermite polynomial form. Moreover we used the C++ programming language to implement this formulation, and obtained the response surface for the displacement stress. In order to demonstrate the validity and practical value of the proposed method, the four numerical examples were offered. By these two numerical examples, it could be seen that the numerical solutions from the SIGA were in good agreement with those obtained by IGA and MCS. And, the calculation time is far superior to the traditional MCS method, which greatly reduces the computational cost. Moreover, these examples allowed us to investigate the scalability and applicability of the proposed method to very large problems in two- and three-dimensions as well as in parallel implementations.

Additionally, at present, this research is just in its infancy, so there are still many limitations. For the proposed analytical method, there are mainly the following limitations in the actual uncertainty analysis.

- For the current research, all the analytical models we use are built in two-dimensional geometric space, and are not suitable for three-dimensional geometric structure. However, based on the defined two-dimensional formalization framework, we can easily extend the method to multi-dimensional space.

- In this research, as a first step of SIGA that can deal with uncertainty in shape, the formulation was defined by assuming that shape uncertainty is represented by one random variable. In addition, by referring to the formalization process of the current single random variable, we can know that the proposed method can be applied to the probability problem with multiple random variables, but it is necessary to import different polynomials and change the current program.

- In this study, we assume that the shape uncertainty of the geometry is based on a normal distribution. For non-normal distribution we can do this by introducing different polynomials.

- At the present stage of research, the formulation method developed can only be applied to two-dimensional static linear-elastic problems, but not to other numerical analysis problems.

The analysis framework of IGA is developed from FEM and it has become an important branch in the field of numerical analysis. Although IGA has some good performance in numerical analysis, there are still many shortcomings. Because of this, IGA is still in a stage to be perfected, and it cannot be like classic FEM what is widely used and brings remarkable advantages and benefits to modern engineering applications at present stage. For example, the idea of using spline control points to parameterize the shapes is not necessarily always an advantage. The problem of parameterization is also a major bottleneck restricting the development of IGA technology and many researchers have done a lot of work to make up for this

deficiency. In this study, we just explored one possible extension of IGA in the field of computational stochastic mechanics, that is incorporating of classical response surface methodology into the IGA analytical framework to evaluate the uncertainty problem in shape in stochastic analysis. Therefore, there are still many deficiencies, which are largely inherited from IGA. In the subsequent research work, overcoming these shortcomings and perfecting our analytical methods is our main research direction.

In future research work, we plan to apply this method to more complex analytical models, especially for the handing of multi-patch and complex CAD boundaries problems, such as multi-patch Coons etc. Therefore, it is indispensable to construct an analysis-suitable parameterization method for SIGA. On the other aspect, we will introduce more stochastic variables into the analytical model in order to represent the more complex stochastic shape. In addition. normally, in the IGA, the stiffness matrix is the band matrix, therefore, it is easy to store and assembling. But in this study, the stiffness matrix is not a band matrix, this is because the order of PCE approximate is introduced into the equation as a parameter. And we use the same Gauss' elimination method as IGA to solve the stiffness equation, we think this is not a good solution for solving matrix in the SIGA. In the current research, we have not optimized the algorithm in this aspect, so SIGA is higher in time and space complexity than a single IGA program.

Appendix A

C++ code for stochastic isogeometric analysis based on NURBS

A.1. Overview

In this study, we used the C++ programming language in order to implement the formulation of proposed method and use the program compilation tool Visual Studio 2017 to execute the code written. This program is only applicable to the 2-dimensional static elasticity problem under normal conditions. And it can handle the single patch and multi-patched NURBS geometry well. Here we mainly give the flow chart of the main program and some sub-function programs. Here we mainly give the flow charts of the main program and some sub-function programs. Based on

these flow charts, users can write programs for different linear elasticity problems, and can rewrite programs with code they are good at.

A.2. Code architecture for determinate isogeometric Analysis

First of all, let us consider the architecture of a determinate isogeometric Analysis programe. The flowchart for a typical example of determinate isogeometric Analysis is given in Figure A1. The C++ program begins with the data "Read input data " that including the boundary value problem, and all of the geometrical data being read from files. These basic data are usually read from external files and geometrical data from the CAD. Once the data has been read, it is possible to connect the associated information by establishing a connection matrix and allocate memory for all major global matrices, which is then initialized to zero. After completing these pre-processing steps, the modules of the program begin to assemble, and a basic loop begins with all the elements in the physical mesh. In each element, the element stiffness matrix and the element force vector are initialized, and then the code enters the loop through the trapezoidal box as shown in Figure A1. for each quadrature point, it calls a routine to compute all the basis functions and any necessary derivatives. The number of basis functions is the same as the control points, and their calculation methods and steps are the same as the geometry design software. We need a separate routine to get these values when needed. After calculating these values, we can continue to build the local stiffness matrix and force vector. With each quadrature point, the local arrays can be assembled. At this point, the information is sequentially added to the global stiffness matrix and the force vector using the connectivity matrix, and the program automatically loops to the next element. After all of the elements are assembled, the global arrays are complete. We then solve the program system, Output result to a file, perform post-processing, then the whole analysis is finished. In addition, in Figure A2, we also give the corresponding multi-patch program flow chart. Throughout the program, a sub-loop of patch is added.

```
                    ╭─────────────╮
                    │    Start     │
                    ╰─────────────╯
                          │
                          ▼
                  ╱─────────────────╱
                 ╱  Read input data ╱
                ╱─────────────────╱
                          │
                          ▼
        ┌──────────────────────────────────────────┐
        │ Build connectivitise and allocate global arrays │
        └──────────────────────────────────────────┘
                          │
                          ▼
              ┌───────────────────────┐
              │   K = 0 and F = 0     │
              └───────────────────────┘
                          │
                          ▼
                ╱───────────────────────╲
               ╱  Loop through elements   ╲
              ╱─────────────────────────────╲
                          │
                          ▼
              ┌───────────────────────┐
              │  Kᵉ = 0 and Fᵉ = 0    │
              └───────────────────────┘
                          │
                          ▼
             ╱────────────────────────────╲
            ╱  Loop through quadrature points ╲
           ╱───────────────────────────────────╲
                          │
                          ▼
        ┌──────────────────────────────────────────┐
        │  Evaluate basis functions and derivatives │
        └──────────────────────────────────────────┘
                          │
                          ▼
            ┌────────────────────────────┐
            │ Add contributions to Kᵉ and Fᵉ │
            └────────────────────────────┘
                          │
                          ▼
           ╲────────────────────────────────╱
            ╲  End loop over quadrature points ╱
             ╲────────────────────────────────╱
                          │
                          ▼
            ┌──────────────────────────────┐
            │ Assemble Kᵉ → K and Fᵉ → F   │
            └──────────────────────────────┘
                          │
                          ▼
             ╲────────────────────────────╱
              ╲  End loop over elements    ╱
               ╲──────────────────────────╱
                          │
                          ▼
                ┌──────────────────┐
                │  Solve Kd = F    │
                └──────────────────┘
                          │
                          ▼
                ╱──────────────────╱
               ╱  Write output data ╱
              ╱──────────────────╱
                          │
                          ▼
                    ╭─────────────╮
                    │    Stop      │
                    ╰─────────────╯
```
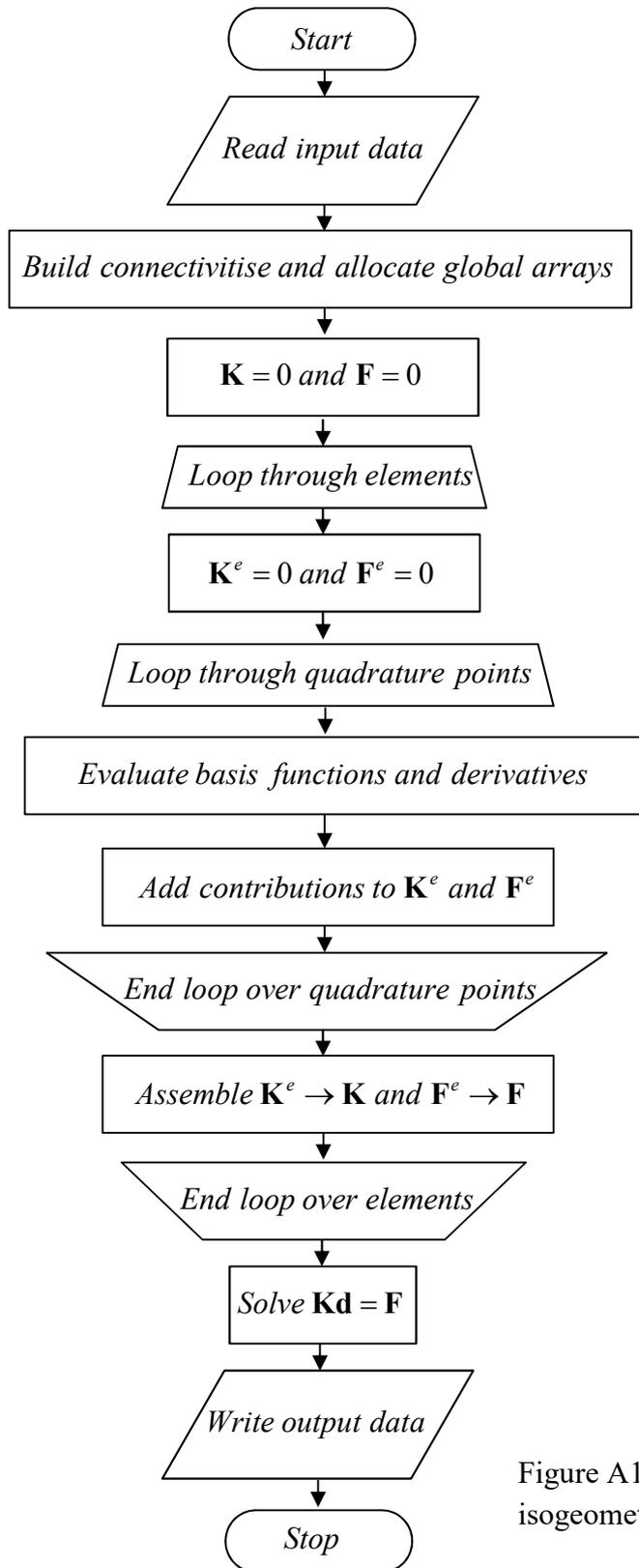
Figure A1 Flowchart of a determinate single-patch isogeometric Analysis programe

```
                 ┌─────────┐
                 │  Start  │
                 └─────────┘
                      │
            ┌──────────────────────┐
            │ Read global input data│
            └──────────────────────┘
                      │
            ┌──────────────────────┐
            │ Build connectivitise │
            │ and allocate global  │
            │      arrays          │
            └──────────────────────┘
                      │
            ┌──────────────────────┐
            │  K = 0 and F = 0     │
            └──────────────────────┘
```
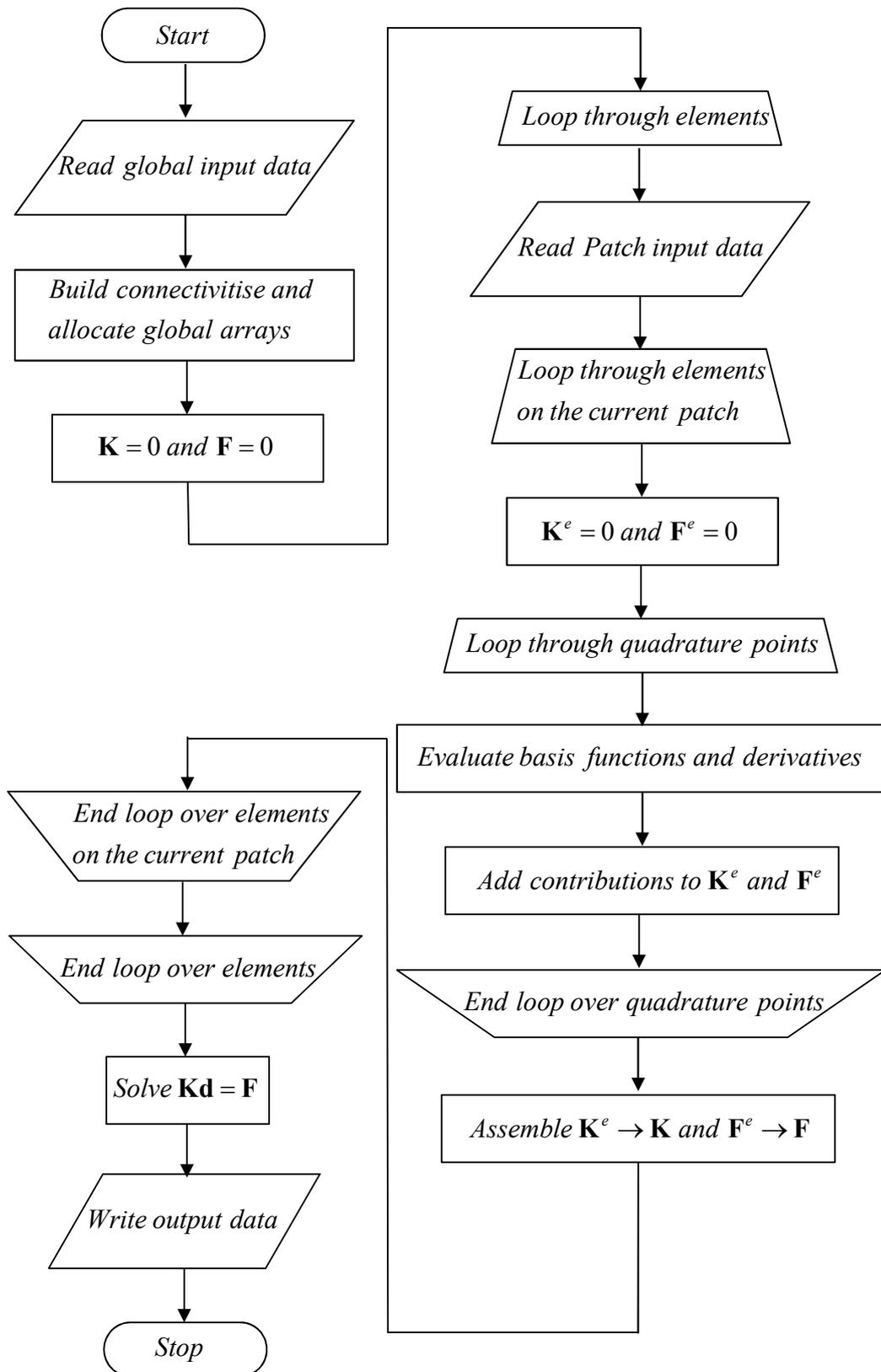
Figure A2 Flowchart of a determinate multi-patch isogeometric Analysis programe

A.3. Code architecture for stochastic isogeometric analysis based on a single patch.

Appendix A.1 shows a flowchart of program of the single patch SIGA for uncertainty in shape. Clearly, the SIGA program architecture is similar to the IGA program architecture, and the different portions of the program are the those shown in blue in Figure A3. We convert an existing a single-patch isogeometric analysis based on NURBS code to SIGA isogeometric analysis code, the only portions of the code that require modification. The program begins with the READ INPUT box that read the information containing the geometrical data, stochastic parameter, etc. After the pre-processing steps are completed, the PCE is introduced into element loop and assembly of system, and that plays an important role in these algorithms. Then, the calculation results in regard to displacement response surface and normal probability distribution, etc. are recorded.

Start

Read input

Build arrays for connectivity and allocation

Initialize the global stiffness matrix **K** *and force vector* **F**

Loop over elements

Initialize the local stiffness matrix **k** *and force vector* **f**

Loop through Gauss points

Compute the parameter coordinate $[\xi, \eta]$

Compute $\partial\xi/\partial\tilde{\xi}; \partial\eta/\partial\tilde{\eta}$

Evaluate basis functions and derivatives

Compute $|\mathbf{J}(\zeta)|$ *using orthogonal property of PCE*

Numerical approximate inverse of $|\mathbf{J}_g(\zeta)|$ *using the Hermite polynomials*

Geometrical data : *coordinate of control points, knot vector, etc.* *Stochastic parameter* : *mean and standard deviation.* *Material properties.* *Boundary conditions.*

Form the strain – displacement matrix B and reweitten the PCE form

Build the local stiffness matrix $\mathbf{k}_e$ *and force vector* $\mathbf{f}_e$ *in PCE form*

End loop over Gauss points

Assemble $\mathbf{k}_e, \mathbf{f}_e$ *to global stiffness matrix* **K** *and force vector* **F**

End loop over elements

Solve global system $\mathbf{Kd} = \mathbf{F}$ *with respect to displacement*

Write output

Stop

Displacement response surface. *Normal probability distribution.*

Figure A3 Flowchart of program for stochastic isogeometric analysis for uncertainty in shape, which applies to single-patch. Routines in blue denote differences from IGA program.

A.4. Code architecture for multi-patch SIGA

A multi-patch stochastic isogeometric analysis code can be made to conform with the flowchart in Figure A3. In practice, the design flow of the whole program is based on the flow chart of a single patch shown in Figure A3. In the multi-patched SIGA analysis program, we need to enter enough associated information, especially for multiple patches. The entire program patch begins to loop, rather than starting with elements as before. It should be noted here that each control point containing the uncertainty parameter variable is only associated with one patch when executing a patch loop. We can enter these associated information in each patch loop and only read the information related to the patch we are currently using. After entering the patch loop, the element loop in the patch is started. At the beginning of this step, everything is analyzed as in the case of a single patch. Finally, we assemble the patch stiffness matrix based on each patch, and then pass these patch stiffness matrix is assembled into a total stiffness matrix and the matrix is solved.

Start

Read input

Geometrical data :
coordinate of control
points, knot vector, etc.
Stochastic parameter :
mean and standard deviation.
Material properties.
Boundary conditions.

Build arrays for connectivity and allocation

Initialize the global stiffness matrix **K** *and force vector* **F**

Read Patch input data

Loop through elements on the current patch

Loop over elements

Initialize the local stiffness matrix **k** *and force vector* **f**

Loop through Gauss points

Compute the parameter coordinate$[\xi, \eta]$

Compute $\partial \xi / \partial \tilde{\xi}; \partial \eta / \partial \tilde{\eta}$

Evaluate basis functions and derivatives

Compute $\left| \mathbf{J}(\zeta) \right|$ *using orthogonal property of PCE*

Numerical approximate inverse of $\left| \mathbf{J}_g(\zeta) \right|$ *using the Hermite polynomials*

Form the strain – displacement matrix B and rewritten the PCE form

Build the local stiffness matrix $\mathbf{k}_e$ *and force vector* $\mathbf{f}_e$ *in PCE form*

End loop over Gauss points

Assemble $\mathbf{k}_e, \mathbf{f}_e$ *to global stiffness matrix* **K** *and force vector* **F**

End loop over elements

End loop over elements on the current patch

Solve global system **Kd** = **F** *with respect to displacement*

Write output

Displacement response surface.
*Normal probability distribution.*

Stop

Figure A4 Flowchart of program for stochastic isogeometric analysis for uncertainty in shape, which applies to multi-patch. Routines in blue denote differences from IGA program.