ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ &
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΠΜΣ "ΔΟΜΟΣΤΑΤΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΩΝ ΚΑΤΑΣΚΕΥΩΝ"
ΔΠΜΣ "ΕΦΑΡΜΟΣΜΕΝΗ ΜΗΧΑΝΙΚΗ"

POST-GRADUATE THESIS
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

NON-LINEAR ISOGEOMETRIC ANALYSIS WITH BOUC-WEN HYSTERESIS MODEL

ΦΟΙΤΗΤΕΣ: ΚΑΤΣΑΒΡΙΑΣ ΕΥΑΓΓΕΛΟΣ & ΚΑΤΣΑΒΡΙΑ ΑΝΑΣΤΑΣΙΑ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΚΟΥΜΟΥΣΗΣ ΒΛΑΣΙΟΣ, ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π.

ΝΟΕΜΒΡΙΟΣ 2012

ΑΘΗΝΑ

PROLOGUE

The advances in computer graphics (introduction of new representation tools like NURBS, coons, T-splines) have recently triggered the development of the isogeometric method, which unifies the fields of CAD (Computer Aided Design) and FEA (Finite Element Analysis). After getting insight into the method and its capabilities, many analysts got interested in integrating it to the existing finite element computer programs. This is the case with our supervisor Prof. Koumousis Vlassis as well. One of Prof Koumousis' main interests is the Bouc – Wen hysteresis model, which he and his Phd. student Triantafyllou S. have already used into FEA. Soon after isogeometric analysis and NURBS got widely known, our supervisor wondered how efficient it might be integrating Bouc – Wen with the isogeometric analysis. The idea was assigned to us as a topic for a thesis.

Hope you enjoy the journey through this knowledge as we did,

Evangelos Katsavrias & Anastasia Katsavria
vageng@gmail.com, anast.katsavria@hotmail.com

ΠΡΟΛΟΓΟΣ

Οι εξελίξεις στον χώρο των γραφικών (εισαγωγή νέων εργαλείων αναπαράστασης όπως τα NURBS, τα coons και τα T-splines) πρόσφατα πυροδότησαν την ανάπτυξη της ισογεωμετρικής μεθόδου, η οποίο ενώνει τα πεδία του CAD (σχεδίαση με την βοήθεια του υπολογιστή) και της FEA (ανάλυση με χρήση πεπερασμένων στοιχείων). Πολλοί αναλυτές, έχοντας λάβει γνώση της ισογεωμετρικής μεθόδου και των δυνατοτήτων της, ενδιαφέρθηκαν για τη δυνατότητα ενσωμάτωσή της στους ήδη υπάρχοντα προγράμματα πεπερασμένων στοιχείων. Ένας εξ' αυτών είναι και ο επιβλέπων καθηγητής μας, Κουμούσης Βλάσσης. Ένα από τα βασικά ενδιαφέροντα του καθ. Κουμούση είναι και το υστερητικό μοντέλο το οποίο έχει ήδη εφαρμόσει σε FEA με τον Δρ. Τριανταφύλλου Σάββα που είχε υπό την επίβλεψη του. Σύντομα αφού έγινε ευρέως γνωστή η ισογεωμετρική μέθοδος και τα NURBS, διερωτήθηκε ο επιβλέπων καθηγητής μας κατά πόσο οι νέες αυτές δυνατότητες είναι αποδοτικές όταν χρησιμοποιηθούν σε μία ανάλυση με χρήση του μοντέλου Bouc – Wen. Αυτό αποτέλεσε έναυσμα για τον ορισμό ενός θέματος διπλωματικής εργασίας μεταπτυχιακού, το οποίο και μας ανατέθηκε.

Ευχόμαστε για ένα ευχάριστο ταξίδι μέσα στη γνώση σαν το δικό μας,

Κατσαβριάς Ευάγγελος & Κατσαβριά Αναστασία

vageng@gmail.com, anastas86@hotmail.com

SUMMARY

The aim of this thesis is to study the efficiency of the isogeometric analysis method combined with the material nonlinear analysis by the hysteresis model Bouc – Wen. Three main parts have been created; the first containing computational geometry, the second the Isogeometric analysis method and the third the combination of the Isogeometric analysis method with the Bouc-Wen hysteresis model.

In the first part, the NURBS basis functions are gradually build. Firstly, the importance of interpolation and approximation methods in computer graphics and computational geometry is stated. Furthermore, it is pointed out the difference between them; to this end, we present some interpolatory and some approximatory curves. The presentation of the latter curves starts from the Bernstein polynomials, which set a basis for the construction of Bézier curves (rational or non). A number of these curves are joined one another to form a B-Spline (rational or non, respectively). While studying the construction method of all these approximation curves, the CAGD routines and plotting routines was developed in Matlab, through which curves, as well as surfaces and solids, and the k$^{\text{th}}$ ($k \leq p$, where p is the degree) derivative at any point of the domain can be constructed and plotted. To this a very helpful reference is (Les Piegel, Wayne Tiller, 1997). Therefore, taking one step at a time, the construction of NURBS basis functions and a full library of plotting routines was reached.

In the second part, the reader is provided with important mathematical background and theory of the finite elements method. Once the necessary mathematical background is presented, the basic features of the geometric basis functions are analyzed to relate them with the known theory. Afterwards, a number of differences between the geometric basis functions and the classical basis functions (Lagrange) are discussed. Then, the kinematic matrices for the plane stress problem are formed and presented in two examples of linear plane stress problems. Additionally, a research on the numerical integration of the geometric basis functions by Legendre polynomials is conducted. Finally, parametric analyses were done by hp-refinement, so as to verify the soundness of the code we developed.

In the last part of this thesis the isogeometric analysis method is combined with the hysteresis model Bouc – Wen. The reader may gain some general information on the phenomenon of hysteresis by reading the introduction. Afterwards, certain parts of the doctoral thesis of Triantafyllou Savvas are attached about the mathematical modeling of Bouc-Wen hysteresis model in force terms and in the complete stress space. Then, it is shown that the generalized Bouc-Wen model in the complete stress

space is compatible with the Isogeometric analysis. The main matrices are developed for such an analysis and two applications are presented under monotonic and cyclic load cases. At the end of the part, the results obtained from a multivariate research on the efficiency of the isogeometric analysis method combined with the Bouc – Wen hysteresis model are attached with certain conclusions.

ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της διπλωματικής εργασίας είναι η μελέτη της αποδοτικότητας της μεθόδου ισογεωμετρικής ανάλυσης όταν συνδυάζεται με πλαστική ανάλυση με το υστερητικό μοντέλο Bouc – Wen. Τρία είναι τα κύρια μέρη που απαρτίζουν την εργασία. Το πρώτο μέρος, περιέχει την παρουσίαση των μεθόδων στην υπολογιστική γεωμετρία, το δεύτερο τη μέθοδο ισογεωμετρικής ανάλυσης και το τρίτο τον συνδυασμό του υστερητικού μοντέλου με την ισογεωμετρική ανάλυση.

Στο πρώτο μέρος, κατασκευάζονται σταδιακά τα NURBS. Πρώτα, σημειώνεται πόσο σημαντικές είναι οι μέθοδοι της παρεμβολής και της προσέγγισης στα γραφικά υπολογιστών και την υπολογιστική γεωμετρία. Επιπλέον, τονίζεται η διαφορά μεταξύ των δύο μεθόδων· προς αυτήν την κατεύθυνση, παρατίθενται μερικές καμπύλες παρεμβολής και μερικές προσέγγισης. Η παρουσίαση των τελευταίων καμπυλών ξεκινά με τα πολυώνυμα Bernstein, τα οποία αποτελούν βάση για την κατασκευή των καμπυλών Bézier (ρητών ή μη). Ενώνοντας δύο ή περισσότερες τέτοιες καμπύλες προκύπτει μία καμπύλη B-Spline (ρητή ή μη, αντίστοιχα). Ταυτόχρονα με την μελέτη των μεθόδων κατασκευής όλων αυτών των καμπυλών προσέγγισης, αναπτύχθηκαν κώδικες στη Matlab που υπολογίζουν και εκτυπώνουν καμπύλες, επιφάνειες και στερεά, καθώς και την κ-οστη ($\kappa \leq p$, όπου p είναι η τάξη της πολυωνυμικής συνάρτησης) παράγωγο σε οποιοδήποτε σημείο του πεδίου ορισμού. Συνεπώς, κάνοντας ένα βήμα τη φορά, επετεύχθη η κατασκευή μιας βιβλιοθήκης από κώδικες Matlab για κατασκευή και εκτύπωση των συναρτήσεων βάσης B-Splines (ρητών NURBS και μη ρητών) και των αντίστοιχων γεωμετριών.

Στο δεύτερο μέρος, παρατίθενται στον αναγνώστη σημαντικά μέρη από το μαθηματικό υπόβαθρο και τη θεωρία των πεπερασμένων στοιχείων. Στη συνέχεια, γίνεται μια προσπάθεια σύνδεσης της θεωρίας των πεπερασμένων στοιχείων με τις γεωμετρικές συναρτήσεις βάσης. Έπειτα, σχολιάζεται ένα πλήθος από διαφορές που εμφανίζουν οι γεωμετρικές και οι συμβατικές συναρτήσεις βάσης στη μέθοδο των πεπερασμένων στοιχείων. Στη συνέχεια, κατασκευάζουμε τα κινηματικά μητρώα για το πρόβλημα επίπεδης έντασης και παρουσιάζουμε δύο παραδείγματα γραμμικών προβλημάτων επίπεδης έντασης. Ακόμα, διερευνήθηκε η μέθοδος αριθμητικής ολοκλήρωσης με πολυώνυμα Legendre των γεωμετρικών συναρτήσεων βάσης. Τέλος, έγιναν παραμετρικές αναλύσεις με πύκνωση της διακριτοποίησης και αύξηση του βαθμού συνέχειας των γεωμετρικών συναρτήσεων σχήματος, ώστε να εξακριβώσουμε την ορθότητα του κώδικα που αναπτύξαμε.

Στο τελευταίο μέρος της εργασίας αυτής η ισογεωμετρική ανάλυση συνδυάζεται με το υστερητικό μοντέλο Bouc – Wen. Ο αναγνώστης μπορεί να ενημερωθεί για το

φαινόμενο της υστέρησης από την εισαγωγή. Στη συνέχεια ακολουθούν αποσπάσματα της διδακτορική διατριβής του Τριανταφύλλου Σάββα για την μαθηματική μοντελοποίηση του μοντέλου υστέρησης Bouc-Wen σε επίπεδο δυνάμεων και στον πλήρες χώρο των τάσεων. Ακολούθως γίνεται η τεκμηρίωση της νομιμότητας της εφαρμογής του γενικευμένου υστερητικού μοντέλου Bouc-Wen στην ισογεωμετρική ανάλυση. Αναπτύσσονται τα βασικά μητρώα για μια τέτοια ανάλυση και παρουσιάζονται εφαρμογές σε μονοτονική και ανακυκλιζόμενη φόρτιση. Στο τέλος, γίνονται παραμετρικές διερευνήσεις για την απόδοση της μεθόδου και εξάγονται συμπεράσματα.

WORK DIVISION

The contents of our thesis extend to many fields that are connected, this implies that both authors need to understand everything and communicate their opinion. Furthermore, a distribution of the entire effort is needed in order to enhance the overall treatment of the subject within the given time span for a postgraduate thesis. This was arranged as follows:

As far as the first part is concerned, the research through the bibliography and the writing was made by Anastasia under the guidelines of Evangelos. The text of the second part was written by Evangelos with the support of Anastasia in certain cases. The writing of the third part was done by Evangelos exclusively. Now turning to the task of code writing, the programming of the basis functions and the main routines in CAGD methods was made by both students and achieved a full library of functions constructing, both rational and non, Bézier and B-Splines basis functions. The curve, surface and solid plotters made by Evangelos. Moreover, Anastasia tested the code and provided Evangelos with any necessary feedback in order to eliminate any errors within the codes. The codes of the Isogeometric analysis, with their post-processors, for the linear and non-linear plane stress-strain problems were written by Evangelos. Anastasia contributed to the analysis of examples and all the parametric analyses in this work. All the subjects was examined and verified by both the students.

CONTENTS

PART I

COMPUTATIONAL GEOMETRY

1 INTRODUCTION

The term ***computational geometry*** refers to a branch of computer science which is devoted to the study of algorithms which can be stated in terms of geometry. The main impetus for the development of computational geometry as a discipline was progress in computer graphics and computer-aided design and manufacturing (CAD/CAM), but many problems in computational geometry are classical in nature, and may come from mathematical visualization.

Other important applications of computational geometry include robotics (motion planning and visibility problems), geographic information systems (GIS) (geometrical location and search, route planning), integrated circuit design (IC geometry design and verification), computer-aided engineering (CAE) (mesh generation and analysis), computer vision (3D reconstruction).

The main branches of computational geometry are:

- Combinatorial computational geometry, also called algorithmic geometry, which deals with geometric objects as discrete entities.
- Numerical computational geometry, also called machine geometry, computer-aided geometric design (CAGD), or geometric modelling, which deals primarily with representing real-world objects in forms suitable for computer computations in CAD/CAM systems. This branch may be seen as a further development of descriptive geometry and is often considered a branch of computer graphics or CAD. The term "computational geometry" in this meaning has been in use since 1971.

Within the context of this thesis, the advances in the numerical computational geometry are being used. The most important instruments here are parametric curves and parametric surfaces, such as Bezier curves, spline curves and surfaces. Application areas include shipbuilding, aircraft, and automotive industries. The modern ubiquity and power of computers means that even perfume bottles and shampoo dispensers are designed using techniques unheard of by shipbuilders of 1960s. Let’s take a short trip in time.

The design of complex curved shapes has always posed a problem, and the technology employed has changed radically over the last century. In the past, curves were generated using mechanical devices such as sweeps (large French curves) or by bending thin laths of metal or wood round pegs or ducks on the loft floor and were drawn by tracing chalk round the laths or scribing on sheets of aluminium. The laths assumed a shape which was both aesthetically pleasing and mechanically sound, being the curve of minimum strain energy, and they were known as splines, an East Anglian dialect word.

In lofting, the shape of a design was represented by a physical artifact, the actual layout, and this was vulnerable to damage. It is said that one of the motivations for a mathematical description came from the realization in the Second World War that a bomb near the loft could disrupt or destroy the master definition of an aircraft. The mathematical method adopted was conic lofting: conic sections were well established mathematically so that geometric constructions for drafting the curves from numerical data or for designing the curves *ab* initio to fit prescribed criteria were available. Several textbooks describing these techniques appeared in the 1940s and make fascinating reading today.

During the same period, the notion of the *mathematical spline* was introduced by Schoenberg, largely for the actuarial fitting of life tables. It was derived from the physical spline by observing that, for small deflections, the shape assumed by the physical spline was a piecewise cubic polynomial. In the early 1960s the limitations of conic lofting became apparent in the civil aircraft industry, and mathematical splines were introduced into lofting by Ferguson at Boeing and later by Sabin at the British Aircraft Corporation, using the interpolatory basis. They made one significant change to Schoenberg's approach: the splines were *vector – valued* rather than *scalar – valued.* This was crucial in two senses: it permitted curves and surfaces to be defined in an axis-independent manner, and it allowed curves to be multi – valued, even closed, and to have slopes of 90°. Unfortunately, it also meant that the spline world split into two camps with the approximation theory community concentrating on *functional* approximation and the geometric design community on *shape* approximation, a totally different problem still today deficient in mathematical underpinnings.

With splines back in the repertoire of the geometric modeller, the next important development lay in the use of vector – valued Bernstein polynomials by Bezier at Renault and de Casteljau at Citröen. From Bézier curves to B-splines was a natural progression, B-splines being the piecewise analog of the Bernstein polynomials. These techniques, which rely on approximation, have the important property, missing in any interpolation method, of guaranteeing the construction of smooth shapes. This is called ***variation diminishing property*** and is crucial to the approximation and design of shapes and to the confidence of users.

The variation diminishing property is particularly striking when compared with the behaviour of standard Lagrange polynomials. An example is illustrated in Figure 1.1

and Figure 1.3, where eight data points in $\mathbb{R}^2$ are being approximated by NURBS and interpolated using Lagrange functions, respectively. The variation diminishing property leads the B-spline curves in Figure 1.1 to be monotone, a property that proves to be useful in analysis. The difference between oscillatory Figure 1.2 and the monotone Figure 1.1 is in whether we have interpreted the data as nodes in classical finite element sense, or as control points. It is the pointwise positivity and the non-interpolatory nature of the B-spline basis that makes this latter interpretation possible.

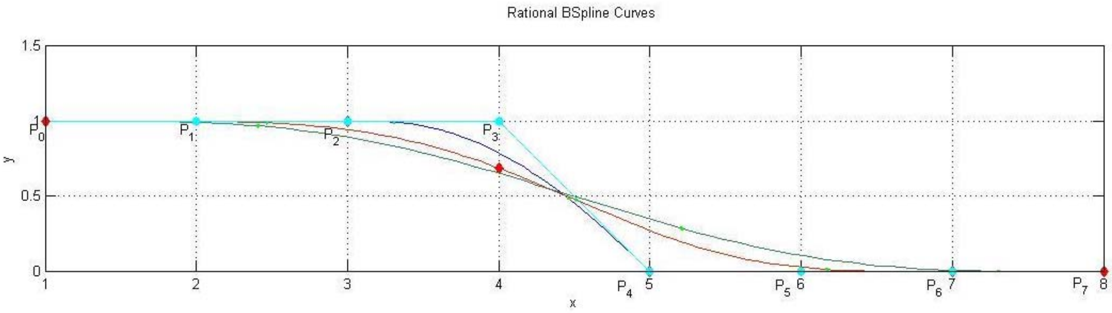


Figure 1.1 NURBS of degrees p=3 (the inner curve- blue coloured), p=5 (the middle curve - red coloured), p=7 (the outer curve - green coloured) approximating discontinuous data

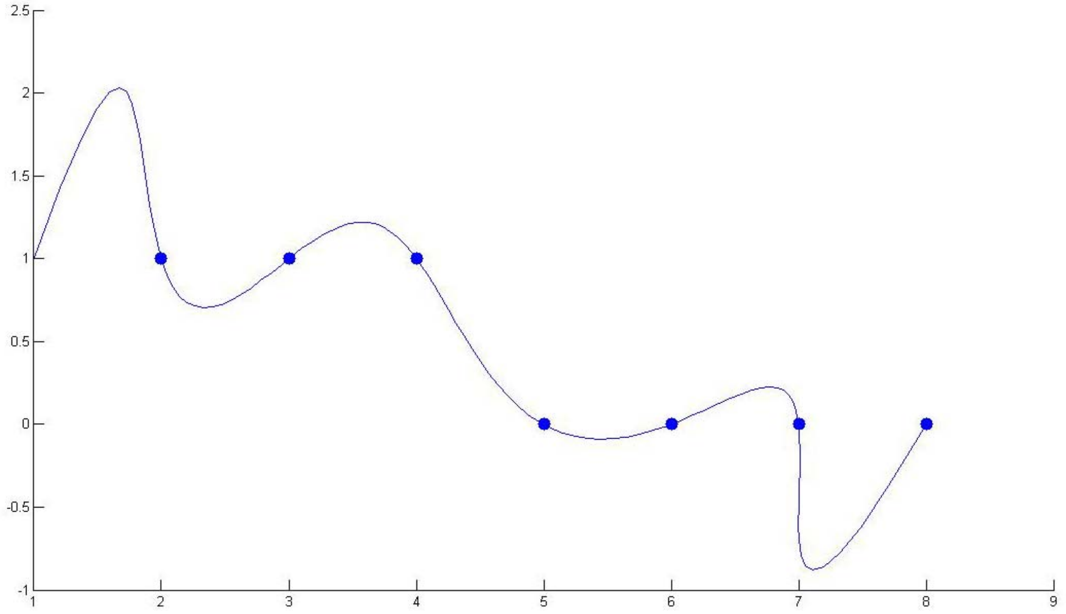


Figure 1.2 Interpolation of the same discontinuous data as in Figure 1.1 using Lagrange functions and a uniform knot vector

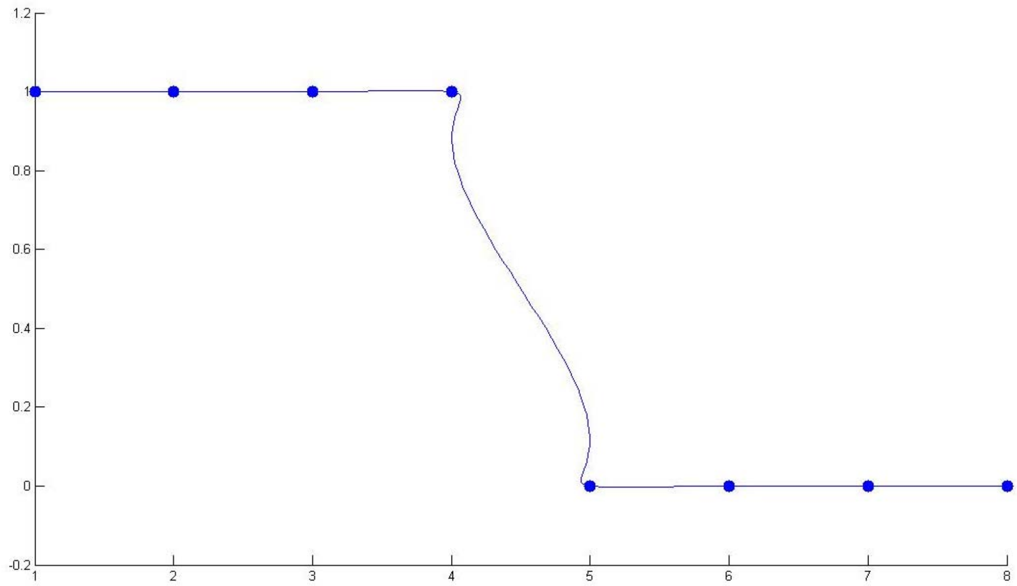


Figure 1.3 Interpolation of the same discontinuous data as in Figure 1.1 using Lagrange functions and a non-uniform knot vector

These geometry entities hold some more properties which are very interesting and useful for the design. Firstly, a p^{th} order basis function is C^{p-1} continuous. Another important property is that of the convex hull, which sets bounds to the range of the geometric object to be constructed. Moreover, part of the power of B-splines is the ability to intuitively change their shape by adjusting the control points Last but not least, B-splines could be refined through knot insertion. All these properties are discussed in the oncoming text.

2 CURVES AND SURFACES BASICS

2.1 IMPLICIT AND PARAMETRIC FORMS (LES PIEGEL, WAYNE TILLER, 1997)

The two most common methods of representing curves and surfaces in geometric modelling are implicit equations and parametric functions. Among the various methods which curves and surfaces in geometric modelling could be represented by, implicit equations and parametric functions appear to be the most common.

The implicit equation of a curve lying in the *xy* plane has the form *f(x,y)* = 0. This equation describes an implicit relation between the x and y coordinates of the points lying on the curve. For a given curve the equation is unique up to a multiplicative constant. An example is the circle of unit radius centred at the origin, specified by the equation $f(x,y) = x^2 + y^2 - 1 = 0$ (Figure 2.1).

In parametric form, each of the coordinates of a point on the curve is represented separately as an explicit function of an independent parameter

Eq. 2.1

$$C(u) = \big(x(u), y(u)\big) \quad a \leq u \leq b$$

Thus, C(u) is a vector-valued function of the independent variable, u. Although the interval $[a, b]$ is arbitrary it is usually normalized to [0, 1]. The first quadrant of the circle shown in Figure 2.1 is defined by the parametric functions

Eq. 2.2

$$x(u) = cos(u)$$

Eq. 2.3

$$y(u) = sin(u) \quad 0 \leq u \leq \frac{\pi}{2}$$

Setting $t = tan({}^{u}/_{2})$, one can derive the alternate representation

Eq. 2.4

$$x(t) = \frac{1 - t^2}{1 + t^2}$$

Eq. 2.5

$$y(t) = \frac{2t}{1 + t^2} \quad 0 \leq t \leq 1$$

Thus, the parametric representation of a curve is not unique.

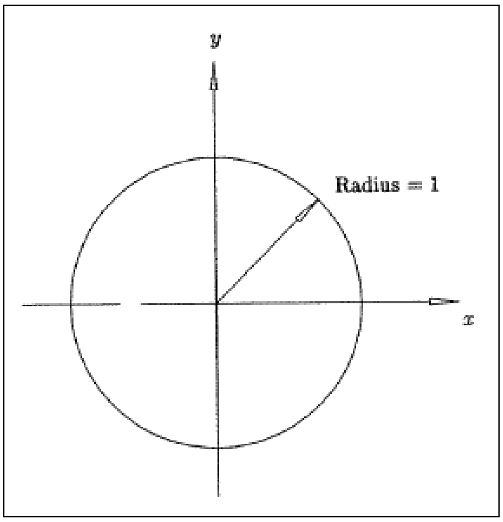


Figure 2.1 A circle of radius 1, centred at the origin

It is instructive to think of $C(u) = (x(u), y(u))$ as the path traced out by a particle as a function of time; u is the time variable, and $[a, b]$ is the time interval. The first and second derivatives of $C(u)$ are the velocity and acceleration of the particle, respectively. Differentiating Eq. 2.2 and Eq. 2.3 once yields the velocity functions

Eq. 2.6

$$C'(u) = (x'(u), y'(u)) = (-\,sin(u), cos(u))$$

Eq. 2.7

$$C'(t) = \left(x'(t), y'(t)\right) = \left(\frac{-4t}{(1+t^2)^2}, \frac{2(1-t^2)}{(1+t^2)^2}\right)$$

Notice that the magnitude of the velocity vector, $C'(u)$, is a constant

Eq. 2.8

$$|C'(u)| = \sqrt{sin^2\,(u)\ +\ cos^2\,(u)}\ =\ 1$$

i.e., the direction of the particle is changing with time, but its speed is constant. This is referred to as a *uniform parameterization.* Substituting t = 0 and t = 1 into C'(t) yields C'(0) = (0,2) and C'(1) = (-1,0), i.e., the particle's starting speed is twice its ending speed (Figure 2.2).

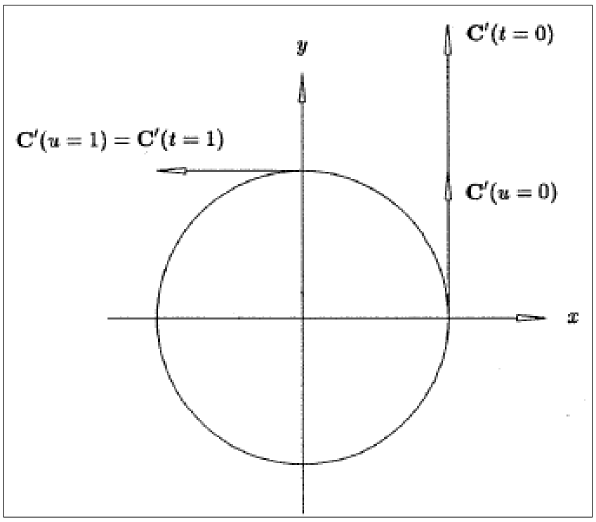


Figure 2.2 Velocity vectors C'(u) and C'(t) at u, t = 0, and 1

A surface is defined by an implicit equation of the form *f(x, y, z)* = 0. An example is the sphere of unit radius centred at the origin, shown in Figure 2.3 and specified by the equation $x^2 + y^2 + z^2 - 1 = 0$. A parametric representation (not unique) of the same sphere is given by $S(u,v) = (x(u,v), y(u,v), z(u,v))$, where

Eq. 2.9

$$x(u,v) = sin(u)cos(v)$$

$$y(u,v) = sin(u)\,sin(v)$$

$$z(u,v) = cos(u) \quad 0 \le u \le \pi, \qquad 0 \le v \le 2\pi$$

Notice that two parameters are required to define a surface. Holding *u* fixed and varying *v* generates the latitudinal lines of the sphere; holding *v* fixed and varying *u* generates the longitudinal lines.

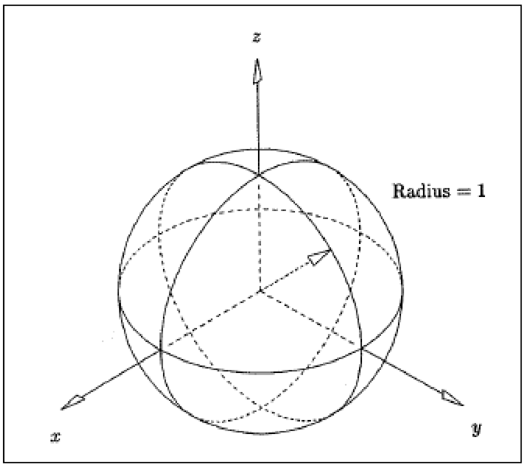


Figure 2.3 A sphere of radius 1, centred at the origin

Denote the partial derivatives of $S(u,v)$by

$S_u(u,v) = (x_u(u,v), y_u(u,v), z_u(u,v))$ and $S_v(u,v) = (x_v(u,v), y_v(u,v), z_v(u,v))$, i.e., the velocities along latitudinal and longitudinal lines. At any point on the surface where the vector cross product $S_u \times S_v$ does not vanish, the unit normal vector, N, is given by (Figure 2.4)

Eq. 2.10

$$N = \frac{Su \times Sv}{|Su \times Sv|}$$

The existence of a normal vector at a point, and the corresponding tangent plane, is a geometric property of the surface independent of the parameterization. Different parameterizations give different partial derivatives, but Eq. 2.10 always yields N provided the denominator does not vanish. From Eq. 2.9 it can be seen that for all v, $0 \leq v \leq 2\pi$, $S_v(O,v) = S_v(\pi,v) = 0$, that is, S_v vanishes at the north and south poles of the sphere. Clearly, normal vectors do exist at the two poles, but under this parameterization Eq. 2.10 cannot be used to compute them.

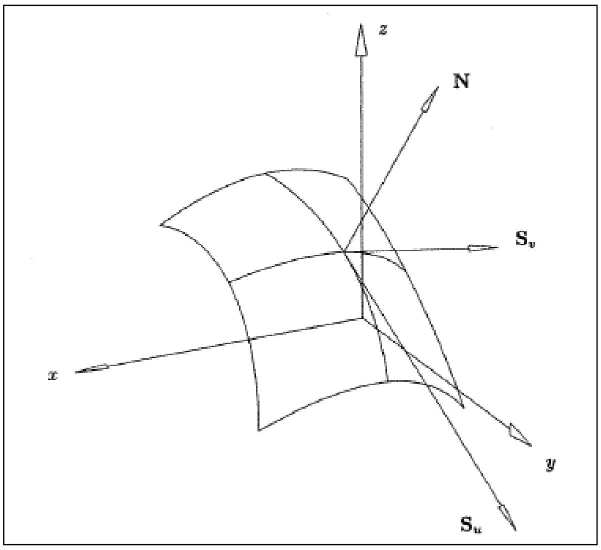


Figure 2.4 Partial derivative and unit normal vectors of S(u,v)

Of the implicit and parametric forms, it is difficult to maintain that one is always more appropriate than the other. Both have their advantages and disadvantages. Successful geometric modelling is done using both techniques. A comparison of the two methods follows:

- By adding a *z* coordinate, the parametric method is easily extended to represent arbitrary curves in three-dimensional space, $C(u) = (x(u), y(u), z(u))$; the implicit form only specifies curves in the *xy* (or *xz* or *yz)* plane;
- It is cumbersome to represent bounded curve segments (or surface patches) with the implicit form. However, boundedness is built into the parametric

form through the bounds on the parameter interval. On the other hand, unbounded geometry (e.g., a simple straight line given by $f(x,y) = ax + by + c = 0$) is difficult to implement using parametric geometry;

- Parametric curves possess a natural direction of traversal (from $C(a)$ to $C(b)$ if $a \leq u \leq b$); implicit curves do not. Hence, it is easy to generate ordered sequences of points along a parametric curve. A similar statement holds for generating meshes of points on surfaces;
- The parametric form is more natural for designing and representing shape in a computer. The coefficients of many parametric functions, e.g., Bezier and B-spline, possess considerable geometric significance. This translates into intuitive design methods and numerically stable algorithms with a distinctly geometric flavour;
- The complexity of many geometric operations and manipulations depends greatly on the method of representation. Two classic examples are:
  - compute a point on a curve or surface - difficult in the implicit form;
  - given a point, determine if it is on the curve or surface - difficult in the parametric form;
- In the parametric form, one must sometimes deal with parametric anomalies which are unrelated to true geometry. An example of this is the unit sphere (see Eq. 2.9). The poles are parametric critical points which are algorithmically difficult, but geometrically the poles are no different than any other point on the sphere.

2.2 CURVE DEFINITION (RICHARD H. BARTELS, JOHN C. BEATTY, BRIAN A. BARSKY, 1987)

There are a variety of ways to define a specific curve. Each can be broadly classified as being based on ***"interpolation"*** or on ***"approximation"***. In both cases one begins by specifying a sequence of points, which we will represent in illustrations by a circle "•" or a "+" sign. In the case of interpolation the curve is required to pass through data points, $\boldsymbol{P}_i$ in sequence order as shown in Figure 2.5. For those techniques based on approximation the curve is required only to pass "near" data points, $\boldsymbol{P}_i$, in the order shown in Figure 2.6. Exactly what "near" means depends on the particular approximation technique used.

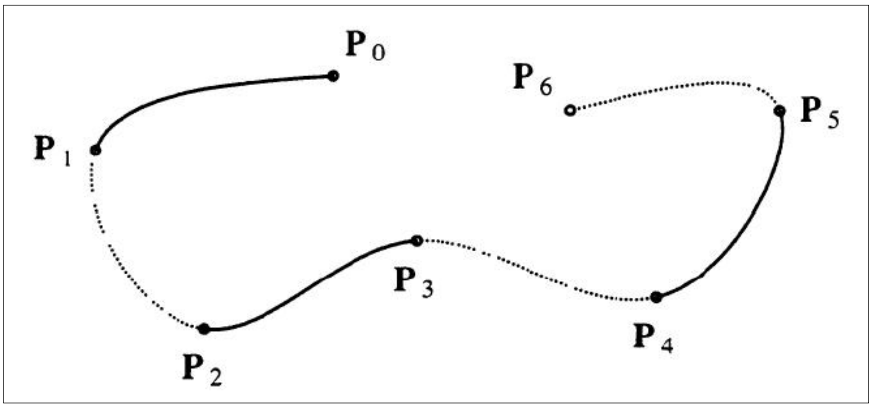


Figure 2.5 A curve defined by interpolation

In either interpolation or approximation, moving the points alters the curve. We will concentrate on a specific method of approximation (B-splines), to be introduced in 1.2, for which the computational cost of constructing the curve is very low and for which only a portion of the curve changes whenever a single point is moved.

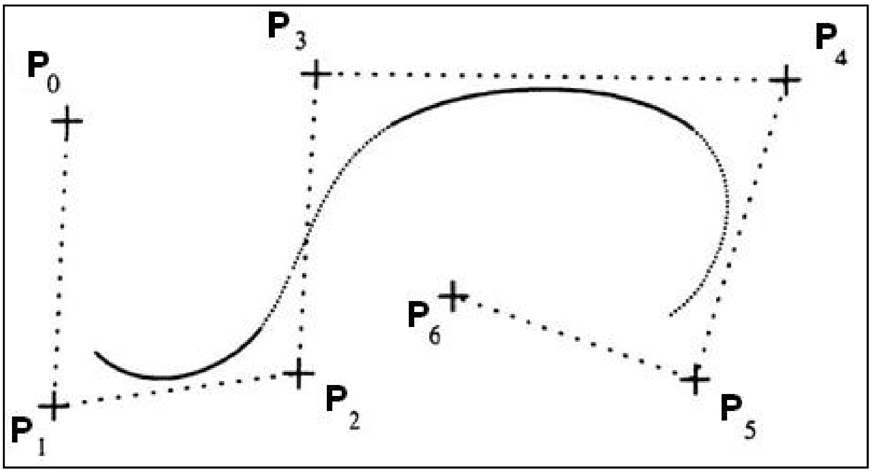


Figure 2.6 A curve that approximates a sequence of points. The solid and heavily dotted curves represent distinct curve segments. Each is a single parametric cubic. The point at which two successive segments meet is called a joint. The value of the parameter u which corresponds to a joint is called a knot.

Although polynomial are computationally efficient and easy to work with, it is not usually possible to define a satisfactory curve using single polynomials for x(u) and y(u). Instead it is customary to break the curve into some number of pieces called ***segments***, each defined by separate polynomials, and join the segments together to form a ***piecewise polynomial*** curve. Thus, as the parameter u varies between some initial minimum value u_{min} and some final maximum value u_{max} to define the curve, certain distinguished values of u, called ***knots***, will be encountered that correspond to the ***joints*** between the polynomial segments. The sequence of knot values is required to be non descending, so that

$$u_0 \leq \cdots \leq u_j = u_{min} \leq \cdots \leq u_l = u_{max} \leq \cdots \leq u_{last}$$

(Note that some of the knots may lie to the left or the right of the range of parameter values defining the curve. Reasons for permitting this will become clear later). The sequence of knot values

$$u_0, \dots, u_j, \dots, u_l, \dots \leq u_{last}$$

is called the ***knot sequence*** or the ***knot vector*.**

Thus the parametric functions x(u) and y(u) are each composed of polynomial pieces, the first covering the interval of u ranging from u_j to the next distinct knot to the right, the second covering values from this next knot to the subsequent distinct knot further to the right, and so on. Usually x(u) and y(u) are required to satisfy some continuity constraints at the joints between successive polynomial segments; if the 0^{th} through d^{th} derivatives are everywhere continuous (in particular, at the joints), then x and y are said to be C^d continuous. Later we will discover that issues of continuity can be arranged by admitting multiple knots; that is, by letting successive members of the knot sequence be equal, which causes certain of the intervals $[u_i, u_{i+1})$ to be vacuous. A sequence like that, with knots being not at constant distance to each other, is called a ***nonuniform knot sequence***. In contrast, sometimes we will assume that the knots are all distinct and a constant distance apart,

$$u_{i+1} = u_i + \Delta$$

This is called a ***uniform knot sequence***. The frequent choice of convenience will be $u_i = i$, for which, clearly, Δ=1. It is clear that non-uniform knot vectors allow us to obtain much richer behaviour than is possible with uniform one.

An ***open uniform*** knot vector is used when constructing a B-spline. It is characterized by the multiplicity of knot values at the ends, which is equal to the order k of the B-spline basis function. Internal knot values are evenly spaced.

Formally, an open uniform knot vector is given by

$$x_i = 0 \quad 1 \leq i \leq k$$

$$x_i = i - k \quad k + 1 \leq i \leq n + 1$$

$$x_i = n - k + 2 \quad n + 2 \leq i \leq n + k + 1$$

It should not be overlooked that the resulting open uniform basis functions yield curves that behave most nearly like Bézier curves. When the number of control polygon vertices is equal to the order of the B-spline basis and an open uniform knot vector is used, the B-spline basis reduces to the Bernstein basis. Hence, the resulting B-spline curve is a Bézier curve. It that case, the knot vector is just k zeros followed by k ones. For example, for four polygon vertices the fourth order (k=4) open uniform knot vector is [0 0 0 0 1 1 1 1] . A cubic Bézier/B-spline curve results.

It will often be simpler to express x and y on the interval from u_i to u_{i+1} as functions of the local parametrization given by

$$\bar{u} = \frac{u - u_i}{u_{i+1} - u_i}$$

rather than as functions of u.

2.3 HERMITE AND CUBIC SPLINE INTERPOLATION (RICHARD H. BARTELS, JOHN C. BEATTY, BRIAN A. BARSKY, 1987)

Suppose that we have m+1 data points $\boldsymbol{P}_0, \dots, \boldsymbol{P}_m$ through which we wish to draw a curve such as that shown in Figure 2.7 (in which m=6).

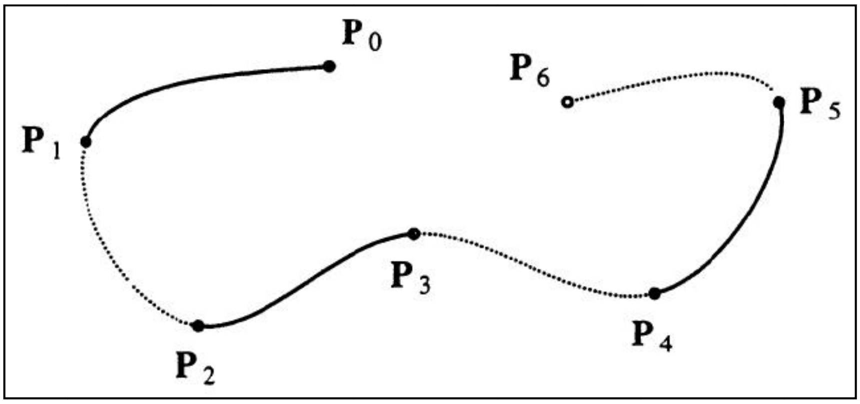


Figure 2.7 An interpolating cubic spline

Each successive pair of data points is connected by a distinct curve segment. The ith segment runs from $\boldsymbol{P}_i$ to $\boldsymbol{P}_{i+1}$, and we will assume that the parameter u runs correspondingly from the knot u_i to the knot u_{i+1} to generate this segment. This corresponds to the knot sequence and parameter range outlined in paragraph 2.2 with the special choices $u_0 = u_j = u_{min}$ and $u_{max} = u_l = u_m = u_{last}$. Since each such segment $\boldsymbol{C_i}(u)$ is represented parametrically as $\left(X_i(u), Y_i(u)\right)$, we are really concerned with how the $X_i(u)$ and $Y_i(u)$are determined by the points

$$\boldsymbol{P}_i = (x_i\,, y_i).$$

In general, the x -coordinates $X(u)$ of points on a curve are determined solely by the x-coordinates $x_0, \dots, x_m$ of the data points, and similarly $Y(u)$ is determined solely by the y -coordinates of the data points. Since both $X(u)$ and $Y(u)$ are treated in the same way we will discuss only $Y(u)$; indeed, to obtain curves in three dimensions we simply define a $Z(u)$ as well and let $\boldsymbol{C_i}(u)$ be given by $\left(X_i(u), Y_i(u), Z_i(u)\right)$.

For ease of computation we will limit ourselves to the use of polynomials in defining $X_i(u)$, $Y_i(u)$ and $Z_i(u)$. Indeed cubic polynomials usually provide sufficient flexibility for many applications at reasonable cost. For the curve in Figure 2.7, then, $Y(u)$ is shown in Figure 2.8.

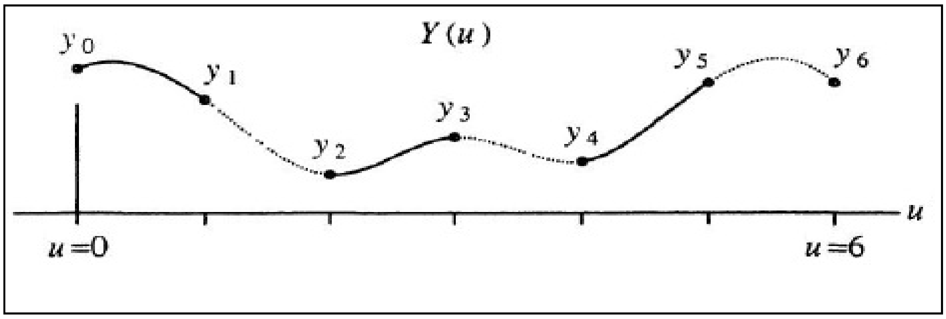


Figure 2.8 Y(u) for the curve shown in Figure 2.7 above. In this example we have rather arbitrary chosen to use uniform knot spacing, so that the knot sequence is [0,1,2,3,4,5,6].

It will be easiest to continue the discussion by re-parameterizing each segment Y_i separately by substituting $\bar{u}$ for u as was described earlier. This means that $\bar{u} = u_i - i$ for the knot sequence given in Figure 2.8. Each $Y_i(\bar{u})$ is a cubic polynomial in the parameter $\bar{u}$. We know two things in particular about

$$Y_i(\bar{u}) = a_i + b_i\bar{u} + c_i\bar{u}^2 + d_i\bar{u}^3 ,$$

namely that

$$Y_i(0) = y_i = a_i$$

$$Y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i .$$

Because we have four coefficients to determine, we need two other constraints to completely determine a particular $Y_i(\bar{u})$. One easy way to do this is to simply pick, arbitrarily, first derivatives D_i of $Y(\bar{u})$ at each knot u_i, so that

$$Y_i^{(1)}(0) = D_i = b_i$$

$$Y_i^{(1)}(1) = D_{i+1} = b_i + 2c_i + 3d_i .$$

These four equations can be solved symbolically, once and for all, to yield

Eq. 2.11

$$a_i = y_i$$

$$b_i = D_i$$

$$c_i = 3(y_{i+1} - y_i) - 2D_i - D_{i+1}$$

$$d_i = 2(y_i - y_{i+1}) + D_i + D_{i+1}$$

Since we use D_i as the derivative at the left end of the i^{th} segment $\left(i.e., as\ Y_i^{(1)}(0)\right)$ and at the right end of the $(i-1)^{th}$ segment $\left(as\ Y_{i-1}^{(1)}(1)\right)$, $Y(\bar{u}\ )$ has a continuous first derivative.

This technique is called ***Hermite interpolation***. It can be generalized to higher-order polynomials.

How are the D_i specified? One possibility is to compute them automatically, perhaps by fitting a parabola through y_{i-1}, y_i and y_{i+1} and using its derivative at y_i as D_i ; arbitrary values (such as 0) can be used at the end points. Or one can use for D_i the y component of a weighted average of the vector from $\boldsymbol{P}_{i-1}$to $\boldsymbol{P}_i$ and the vector from $\boldsymbol{P}_{i+1}$ to $\boldsymbol{P}_i$. Or the user may specify derivative vectors directly.

It is possible to arrange that successive segments match second as well as first derivatives at joints, using only cubic polynomials. Suppose, as above, that we want to interpolate the (m + 1) points $\boldsymbol{P}_0, \ldots, \boldsymbol{P}_m$ by such a curve. Each of the m segments $Y_0(\bar{u}\ ), \ldots, Y_{m-1.}(\bar{u}\ )$ is a cubic polynomial determined by four coefficients. Hence we have 4m unknown values to determine. At each of the (m-1) interior knots $u_1, \ldots, u_{m-1}$ (where two segments meet) we have four conditions:

$$Y_{i-1}(1) = y_i, \qquad Y_{i-1}^{(1)}(1) = Y_i^{(1)}(0)$$

$$Y_i(0) = y_i, \qquad Y_{i-1}^{(2)}(1) = Y_i^{(2)}(0)$$

Since we also require that

$$Y_0(0) = y_0$$

$$Y_{m-1}(1) = y_m$$

we have a total of 4(m - 1) + 2 = 4m - 2 conditions from which to determine our 4m unknowns. Thus, we need two more conditions. These may be chosen in a variety of ways. A common choice is simply to require that the second derivatives at the endpoints u_0 and u_m both be zero; these conditions yield what is called a ***natural cubic spline***. Figure 2.10 is actually a natural cubic spline.

2.4 PRACTICAL CONSIDERATIONS-COMPUTING NATURAL CUBIC SPLINES (RICHARD H. BARTELS, JOHN C. BEATTY, BRIAN A. BARSKY, 1987)

We do not need to solve 4m equations directly – the problem can be simplified. Notice that a natural cubic spline is actually a special case of Hermite interpolation; we may simply choose first derivative vectors so as to match second derivatives as well. If we can compute the needed D_i , we have already obtained definitions of the a_i, b_i, c_i and d_i in terms of the D_i.

Thus at each internal joint we want to choose D_i so that

$$Y_{i-1}^{(2)}(1) = Y_i^{(2)}(0)$$

or

$$2c_{i-1} + 6d_{i-1} = 2c_i.$$

Substituting in our earlier solutions Eq. 2.11 forc_{i-1}, d_{i-1} and c_i, we have

$$2[3\,(y_i - y_{i-1}) - 2D_{i-1} - D_i\,] + 6[2\,(y_{i-1} - y_i) + D_{i-1} + D_i\,]$$
$$= 2[3\,(y_{i+1} - y_i) - 2D_i - D_{i+1}].$$

Simplifying, and moving the unknowns to the left, we have

Eq. 2.12

$$D_{i-1} + 4D_i + D_{i+1} = 3(y_{i+1} - y_{i-1}).$$

Since there are m - 1 internal joints, there are m - 1 such equations. Requiring that the second derivative at the beginning of the curve be zero implies that

$$2c_0 = 0$$
$$2[3\,(y_1 - y_0) - 2D_0 - D_1] = 0$$
$$2D_0 + D_1 = 3\,(y_1 - y_0)$$

Requiring that the second derivative at the end of the curve be zero similarly results in

$$D_{m-1} + 2D_m = 3\,(y_m - y_{m-1})$$

We now have m+1 equations in m+1 unknowns. Representing them in matrix form we have

$$\begin{bmatrix} 2 & 1 & & & & & & \\ 1 & 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & & \\ & & 1 & 4 & 1 & & & \\ & & & . & . & . & . & \\ & & & & & 1 & 4 & 1 \\ & & & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ . \\ . \\ . \\ . \\ D_m \end{bmatrix} = \begin{bmatrix} 3\,(y_1 - y_0) \\ 3\,(y_2 - y_0) \\ . \\ . \\ . \\ 3\,(y_m - y_{m-2}) \\ 3\,(y_m - y_{m-1}) \end{bmatrix}$$

Beginning at the top, the first 1 in each row is eliminated using the row immediately above and the diagonal is scaled:

$\gamma_0 \leftarrow 1/2$

```
$for\ i \leftarrow 1\ step\ 1\ until\ m-1\ do$
    $\gamma_i \leftarrow 1/(4-\gamma_{i-1})$
$endfor$
$\gamma_m \leftarrow 1/(2-\gamma_{m-1})$
```

Corresponding operations are carried out on the right-hand-side entries; e.g., for the y components shown above:

```
$\delta_0 \leftarrow 3\,(y_1 - y_0)\gamma_0$
$for\ i \leftarrow 1\ step\ 1\ until\ m-1\ do$
    $\delta_i \leftarrow (\,3\,(y_{i+1} - y_{i-1}) - \delta_{i-1})\gamma_i$
$endfor$
$\delta_m \leftarrow (\,3\,(y_m - y_{m-1}) - \delta_{m-1})\gamma_m$
```

The result of this *forward elimination* process will be

$$\begin{bmatrix} 1\ \gamma_0 & & & & & & \\ & 1\ \gamma_1 & & & & & \\ & & 1\ \gamma_2 & & & & \\ & & & . \quad . \quad . & & & \\ & & & & \gamma_{m-2} & & \\ & & & & & 1 \quad \gamma_{m-1} & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ . \\ . \\ . \\ . \\ D_m \end{bmatrix} = \begin{bmatrix} \delta_0 \\ \delta_1 \\ . \\ . \\ . \\ . \\ \delta_m \end{bmatrix}$$

This directly yields the value of D_m, and it is then a simple matter to solve successively for $D_{m-1}, \dots, D_0$ in a process of backward *substitution*:

```
$D_m \leftarrow \delta_m$
$for\ i \leftarrow m-1\ step\ -1\ until\ 0\ do$
    $D_i \leftarrow \delta_i - \gamma_i D_{i+1}$
$endfor$
```

The multiplicative factors γ_i that accomplish the forward substitution need only be computed once. The δ_is must be computed and the backward substitution performed separately for each coordinate. When a data point is moved, the values $\delta_j, \dots, \delta_m$ must be recomputed and the entire backward substitution again performed.

2.5 OTHER END CONDITIONS FOR CUBIC INTERPOLATING SPLINES

There are many other ways in which to determine the additional two constraints needed to define a C^2 continuous interpolating cubic spline fully. These conditions are most commonly applied to the ends of a curve, hence the name ***end conditions***; the natural cubic splines offer an example of this. However, all that is really necessary is to provide the missing two conditions. Any two linear equations that are independent of those provided by the interpolation conditions could be used. They could involve data points or derivatives interior to the curve as well as at the ends. Whatever conditions are used, they will have some influence over the shape of the entire curve. For example, instead of fixing the second derivatives at the first and last knot to zero, we may fix the first derivatives there to be zero.

2.6 KNOT SPACING

Although the end conditions discussed above affect the entire curve, their principal influence is felt at the endpoints. Gross changes to a curve's shape can be made anywhere, without moving the interpolation points, by varying the knot spacing. (See Figure 2.9.)

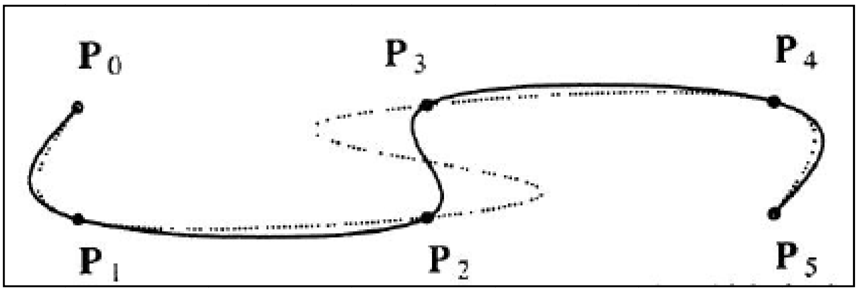


Figure 2.9 The solid line is a natural cubic interpolating spline in which the knots are spaced one unit apart. Unit knot spacing is used also in the dotted curve except for the parametric interval corresponding to the segment between P_2 and P_3, for which the knots are spaced four units apart.

With the single exception of Figure 2.9, we have used a uniform knot sequence in defining the interpolating cubic spline curves discussed above. The knot vector for the solid curve in Figure 2.9 is 0, 1, 2, 3, 4, 5 while the dotted curve interpolates the same data points, but for the knot vector 0, 1, 2, 6, 7, 8.

Thus knot spacing can be used to influence shape; the more difficult question is how that influence can be controlled intuitively.

Uniform knot spacing is one obvious way to define a knot sequence. The Euclidean distance between data points is a second natural choice for the length of the parametric interval over which u varies in defining a segment.

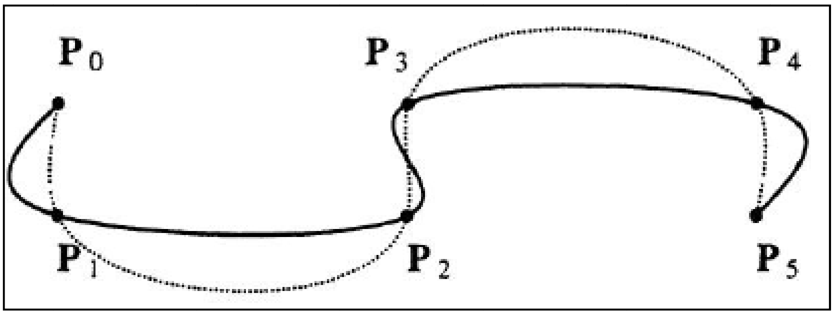


Figure 2.10 The solid line in the above figure is a natural cubic interpolating spline in which the knots are spaced a unit apart. In the case of the dotted curve, the knots corresponding to two successive data points differ in value by the Euclidean distance separating the two points.

2.7 CLOSED CURVES

It is sometimes useful to generate closed curves such as in Figure 2.11. In this case, Eq. 2.12 applies at each of the m points, with the caveat that indices must be computed modulo m + 1. The system of equations that results looks a little different:

$$\begin{bmatrix} 4 & 1 & & & & & & 1 \\ 1 & 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & & \\ & & 1 & 4 & 1 & & & \\ & & & . & . & . & . & \\ & & & & & 1 & 4 & 1 \\ 1 & & & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ . \\ . \\ . \\ . \\ D_m \end{bmatrix} = \begin{bmatrix} 3\,(y_1 - y_m) \\ 3\,(y_2 - y_0) \\ . \\ . \\ . \\ 3\,(y_m - y_{m-2}) \\ 3\,(y_0 - y_{m-1}) \end{bmatrix}$$

Basically one solves this system as one solved for the D_i for an open curve. During forward elimination, however, it is necessary to compute and save nonzero values for entries in the rightmost column and to successively cancel the leftmost nonzero value in the bottom row. The analogous change must be made to the back substitution process as well.

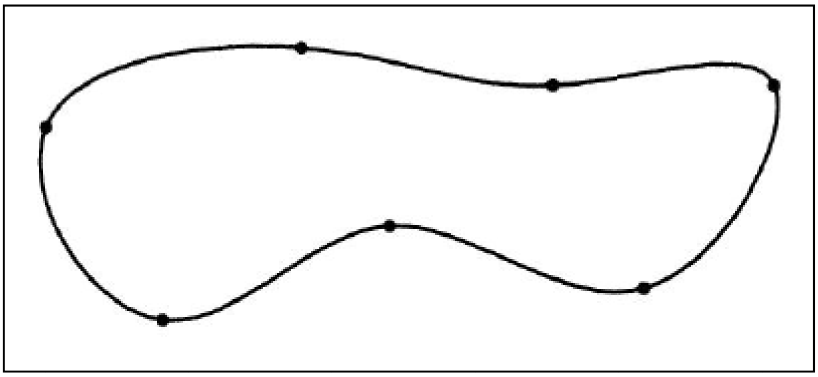


Figure 2.11 A closed interpolating cubic spline

3 FUNCTION SPACES (FARIN, 2002)

It is probably already clear to the reader that the interpolation or approximation of the data is done with the use of a number of functions with different properties. Here, we provide the reader with some useful definitions of functional analysis.

Let $C[a,b]$ be the set of all real-valued continuous functions defined over the interval $[a,b]$ of the real axis. We can define addition and multiplication by a constant for elements $f, g \in C[a,b]$ by setting $(\alpha f + \beta g)(t) = \alpha f(t) + \beta g(t)$ for all $t \in [a,b]$. With these definitions, we can easily show that $C[a,b]$ forms a *linear space* over the reals. The same is true for the sets $C^k[a,b]$ the sets of all real- valued functions defined over $[a,b]$ that are *k-times continuously differentiable*. Furthermore, for every k, C^{k+1} is a subspace of C^k.

We say that *n* functions $f_i, \ldots, f_n \in C[a,b]$ are *linearly independent* if $\sum c_i f_i = 0$ for all $t \in [a,b]$ implies $c_1 = \ldots = c_n = 0$.

We mention some subspaces of $C[a,b]$ that will be of interest later. The spaces P^n of all *polynomials* of degree *n*

$$p^n(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n \; ; \; t \in [a,b]$$

For fixed n, the dimension of P^n is $n + 1$: each $p^n \in P^n$ is determined uniquely by the $n + 1$ coefficients $a_0, \ldots, a_n$. These can be interpreted as a vector in $(n + 1)$-dimensional linear space $\mathbb{R}^{n+1}$ which has dimension $n + 1$. We can also name a *basis* for P^n: the *monomials* $1, t, t^2, \ldots, t^n$ are $n + 1$ linearly independent functions and thus form a basis.

Another interesting class of subspaces of $C[a,b]$ is given by piecewise linear functions: let $a = t_0 < t_1 < \cdots < t_n = b$ be a *partition* of the interval $[a,b]$. A continuous function that is linear on each subinterval $[t_i, t_{i+1}]$ is called a *piecewise linear function*. Over a fixed partition of $[a,b]$, the piecewise linear functions form a linear function space. A basis for this space is given by the *hat functions*: a hat function $H_i(t)$ is a piecewise linear function with $H_i(t_i) = 1 \; and \; H_j(t_j) = 0 \; if \; i \neq j$. A piecewise linear function f with $f(t_j) = f_j$ can always be written as

$$f(t) = \sum_{j=0}^{n} H_j(t)\, f_j$$

Figure 3.1 illustrates an example.

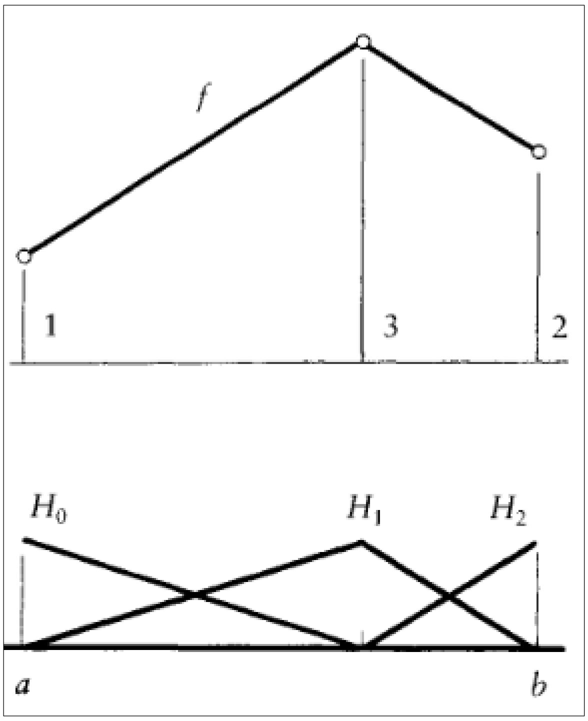


Figure 3.1 Hat functions : the piecewise linear function f can be written as f =H_0+3H_1+2H_2

We will also consider *linear operators* that assign a function $A[f]$ to a given function f. An operator $A : C[a,b] \to C[a,b]$ is called *linear* if it leaves linear combinations invariant:

$$A(\alpha f + \beta g) = \alpha A[f] + \beta A[g];\ a, \beta \in \mathbb{R}.$$

An example is given by the derivative operator that assigns the derivative f to a given function f: $Af = f'$.

4 CONVEX HULL (FARIN, 2002)

As was mentioned in the introduction, the tools which are used in the following paragraphs hold the important property called the convex hull property; hence we consider it is useful to learn little about the mathematical structure called convex hull.

As a first step towards a mathematical description of an object, one defines a coordinate system in which it will be described analytically. A point in a coordinate system identifies a location, often relative to other objects. Upon points we may apply a number of operations. For example, translation $\mathbf{a}+\mathbf{w}; \mathbf{a} \in \mathbb{E}^3, \mathbf{w}$ is an arbitrary vector, and subtraction – this operation yields a vector. Moreover, additionlike operations are defined for points: they are barycentric combinations. These are weighted sums of points where the weights sum to one:

Eq. 4.1

$$\mathbf{b} = \sum_{j=0}^{n} a_j \mathbf{b_j} \;\mid\; \mathbf{b} \in \mathbb{E}^{\mathbf{3}},\; \mathbf{b_j} \in \mathbb{E}^{\mathbf{3}},\; \sum_{j=0}^{n} a_j = 1$$

At first glance, this looks like an undefined summation of points, but we can rewrite Eq. 4.1 as

$$\mathbf{b} = \mathbf{b}_0 + \sum_{j=1}^{n} a_j (\mathbf{b_j} - \mathbf{b}_0),$$

which is clearly the sum of a point and a vector.

An example of a barycentric combination is the centroid g of a triangle with **a, b, c,** given by

$$\mathrm{g} = \frac{1}{3}\mathbf{a} + \frac{1}{3}\mathbf{b} + \frac{1}{3}\mathbf{c}.$$

An important special case of barycentric combinations are the *convex combinations*. These are barycentric combinations where the coefficients a_j , in addition to summing to one, are also nonnegative. A convex combination of points is always "inside" those points, which is an observation that leads to the definition of the convex hull of a point set: this is the set that is formed by all convex combinations of a point set. Figure 4.1 gives an example. More intuitively, the *convex hull* of a set is formed as follows: for a 2D set, imagine a string that is loosely circumscribed around the set, with nails driven through the points in the set. Now pull the string tight—it will become the boundary of the convex hull.

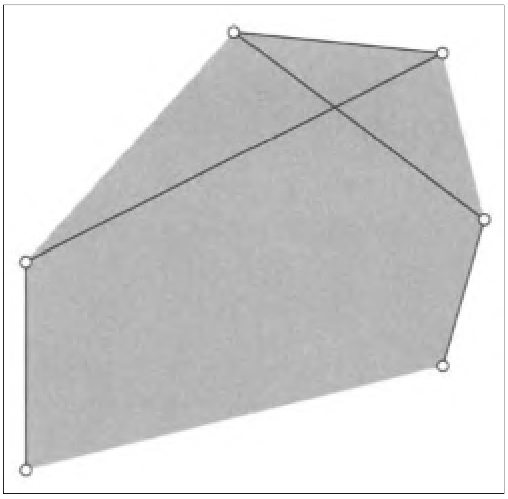


Figure 4.1 Convex hulls: a point set (a polygon) and its convex hull, shown shaded

The convex hull of a point set is a *convex set*. Such a set is characterized by the following: for any two points in the set, the straight line connecting them is also contained in the set. Examples are ellipses or parallelograms. It is an easy exercise to verify that affine (see section 5) maps preserve convexity.

Let us return to barycentric combinations, which generate points from points. If we want to generate a *vector* from a set of points, we may write

$$\mathbf{v} = \sum_{\mathrm{j}=0}^{\mathrm{n}} a_j \mathbf{b}_j \mid \mathbf{v} \in \mathbb{V}^3 \quad \mathbf{b}_\mathbf{j} \in \mathbb{E}^3 \ \ and \quad \sum_{\mathrm{j}=0}^{\mathrm{n}} a_j = 0$$

where we have a new restriction on the coefficients: now we must demand that the a_j sum to zero.

5 AFFINE MAPS (FARIN, 2002)

Most of the transformations that are used to position or scale an object in a computer graphics or CAD environment are affine maps. The term affine map is due to L. Euler; affine maps were first studied systematically by F. Moebius. It is a transformation which preserves straight lines (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances between points lying on a straight line (e.g., the midpoint of a line segment remains the midpoint after transformation). It does not necessarily preserve angles or lengths, but does have the property that sets of parallel lines will remain parallel to each other after an affine transformation. Examples of affine transformations include translation, geometric contraction, expansion, homothety, reflection, rotation, shear mapping, similarity transformation, and spiral similarities and compositions of them. An affine transformation expressed in mathematics is equivalent to a linear transformation followed by a translation.

The fundamental operation for points is the barycentric combination. We will thus base the definition of an affine map on the notion of barycentric combinations. *A map* $\boldsymbol{\Phi}$ *that maps* $\mathbb{E}^3$ *into itself is called an affine map if it leaves barycentric combinations invariant.* So if

$$\boldsymbol{x} = \sum \alpha_j \mathbf{a}_{\boldsymbol{j}} \; ; \; \sum \alpha_j = 1, \boldsymbol{x}, \mathbf{a}_{\boldsymbol{j}} \in \mathbb{E}^3$$

and $\boldsymbol{\Phi}$ is an affine map, then also

Eq. 5.1

$$\boldsymbol{\Phi x} = \sum \alpha_j \boldsymbol{\Phi} \mathbf{a}_{\boldsymbol{j}} \; ; \; \boldsymbol{\Phi x}, \boldsymbol{\Phi} \mathbf{a}_{\boldsymbol{j}} \in \mathbb{E}^3$$

This definition looks fairly abstract, yet it has a simple interpretation. The expression $\boldsymbol{x} = \sum \alpha_j \mathbf{a}_{\boldsymbol{j}}$ specifies how we have to weight the points $\mathbf{a}_{\boldsymbol{j}}$ such that their weighted average is $\boldsymbol{x}$. This relation is still valid if we apply an affine map to all points $\mathbf{a}_{\boldsymbol{j}}$ and to $\boldsymbol{x}$. As an example, the midpoint of a straight line segment will be mapped to the midpoint of the affine image of that straight line segment. Also, the centroid of a number of points will be mapped to the centroid of the image points.

Let us now be more specific. In a given coordinate system, a point $\boldsymbol{x}$ is represented by a coordinate triple, which we also denote by $\boldsymbol{x}$. An affine map now takes on the familiar form

Eq. 5.2

$$\boldsymbol{\Phi x} = A\boldsymbol{x} + \boldsymbol{v}$$

where A is a 3x3 matrix and $\boldsymbol{v}$ is a vector from $\mathbb{R}^3$.

It is easily proved that the inverse of our initial statement is true as well: every map of the form Eq. 5.2 represents an affine map.

Some examples of affine maps are as follows:

The identity. It is given by $\boldsymbol{v} = \boldsymbol{0}$, the zero vector, and by $A = I$, the identity matrix.

A translation. It is given by $A = I$, and a *translation vector* **v**.

A scaling. It is given by $\boldsymbol{v} = \boldsymbol{0}$ and by a diagonal matrix A. The diagonal entries define by how much each component of the pre-image **x** is to be scaled.

A rotation. If we rotate around the z-axis, then $\boldsymbol{v} = \boldsymbol{0}$ and $A = \begin{bmatrix} cosa & -sina & 0 \\ sina & cosa & 0 \\ 0 & 0 & 1 \end{bmatrix}$

A shear. An example is given by $\boldsymbol{v} = \boldsymbol{0}$ and $A = \begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$

This family of shears maps the x, y-plane onto itself while "tilting" the z-axis.

A parallel projection. All of $\mathbb{E}^3$is projected onto the x, y-plane if we set

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and $\boldsymbol{v} = \boldsymbol{0}$. Note that A may also be viewed as a scaling matrix.

An important special case of affine maps are the euclidean maps, also called rigid body motions. They are characterized by orthonormal matrices A, which are defined by the property $A^T A = I$. Euclidean maps leave lengths and angles unchanged; they are either rotations or translations.

Affine maps can be combined, and a complicated map may be decomposed into a sequence of simpler maps. Every affine map can be composed of translations, rotations, shears and scalings.

The rank of A has an important geometric interpretation: if rank (A) = 3, then the affine map $\boldsymbol{\Phi}$ maps 3D objects to 3D objects. If the rank is less than three, $\boldsymbol{\Phi}$ is a parallel projection onto a plane (rank = 2) or even onto a straight line (rank = 1).

An affine map of $\mathbb{E}^2$ to $\mathbb{E}^2$ is uniquely determined by a (nondegenerate) triangle and its image. Thus any two triangles determine an affine map of the plane onto itself. In $\mathbb{E}^3$, an affine map is uniquely defined by a (nondegenerate) tetrahedron and its image.

We may also define affine maps of vectors. If $\boldsymbol{w} = \boldsymbol{b} - \boldsymbol{a}$ is a vector, and $A\boldsymbol{x} + \boldsymbol{v}$ represents an affine map $\boldsymbol{\Phi}$, then

$$\boldsymbol{\Phi}(\boldsymbol{w}) = A\boldsymbol{w}$$

is the image of $\boldsymbol{w}$ under $\boldsymbol{\Phi}$. As expected, the translational part $\boldsymbol{v}$ of the affine map is of no consequence when mapping vectors to vectors.

6 ONE SEGMENT NON-RATIONAL AND RATIONAL POLYNOMIAL CURVES, SURFACES AND SOLIDS

There are trade-offs when implementing a geometric modelling system. The ideal situation is to restrict ourselves to a class of functions which

- are capable of precisely representing all the curves the users of the system need;
- are easily, efficiently, and accurately processed in a computer, in particular:
  - the computation of points and derivatives on the curves is efficient;
  - numerical processing of the functions is relatively insensitive to floating point round-off error;
  - the functions require little memory for storage;
- are simple and mathematically well understood.

A widely used class of functions is the polynomials. Although they satisfy the last two criteria in this list, there are a number of important curve (and surface) types which cannot be precisely represented using polynomials; these curves must be approximated in systems using polynomials. In this section, we study two common methods of expressing polynomial functions, power basis and Bézier. Although mathematically equivalent, we will see that the Bezier method is far better suited to representing and manipulating shape in a computer.

6.1 POWER BASIS CURVES (LES PIEGEL, WAYNE TILLER, 1997)

An nth-degree power basis curve is given by

Eq. 6.1

$$\boldsymbol{C}(u) = \big(x(u), y(u), z(u)\big) = \sum_{i=0}^{n} \boldsymbol{a}_i u^i \qquad 0 \le u \le 1$$

The $\boldsymbol{a}_i = (x_i, y_i, z_i)$ are vectors, hence

Eq. 6.2

$$x(u) = \sum_{i=0}^{n} x_i u^i \quad y(u) = \sum_{i=0}^{n} y_i u^i \quad z(u) = \sum_{i=0}^{n} z_i u^i$$

In matrix form Eq. 6.1 is

Eq. 6.3

$$\boldsymbol{C}(u) = [\boldsymbol{a_0}\ \ \boldsymbol{a_1}\ \ \dots\ \ \boldsymbol{a}_n] \begin{bmatrix} 1 \\ u \\ \vdots \\ u^n \end{bmatrix} = [\boldsymbol{a_i}]^T [u^i]$$

Differentiating Eq. 6.1 yields

Eq. 6.4

$$\boldsymbol{a_i} = \frac{\boldsymbol{C}^{(i)}(u)\big|_{u=0}}{i!}$$

Where $\boldsymbol{C}^{(i)}(u)\big|_{u=0}$ is the ith derivative of $\boldsymbol{C}(u)$ at $u = 0$. The n+1 functions, $\{u^i\}$, are called the ***basis (or blending) functions***, and the $\{\boldsymbol{a_i}\}$ the coefficients of the power basis representation.

Given u_0, the point $\boldsymbol{C}(u_0)$ on a power basis curve is most efficiently computed using Horner's method

- for degree = 1 : $\boldsymbol{C}(u_0) = \boldsymbol{a_1} u_0 + \boldsymbol{a_0}$
- degree = 2 : $\boldsymbol{C}(u_0) = (\boldsymbol{a_2} u_0 + \boldsymbol{a_1}) u_0 + \boldsymbol{a_0}$
- $\vdots$
- degree = n :

Eq. 6.5

$$\boldsymbol{C}(u_0) = ((\dots((\boldsymbol{a_n} u_0 + \boldsymbol{a_{n-1}}) u_0 + \boldsymbol{a_{n-2}}) u_0 + \cdots) u_0 + \boldsymbol{a_1}) u_0 + \boldsymbol{a_0}$$

6.2 TENSOR PRODUCT SURFACES – SOLIDS (LES PIEGEL, WAYNE TILLER, 1997)

The curve $\boldsymbol{C}(u)$ is a vector-valued function of one parameter. It is a mapping (deformation) of a straight line segment into Euclidean three-dimensional space. A surface is a vector-valued function of two parameters, u and v, and represents a mapping of a region, $\mathcal{R}$, of the uv plane into Euclidean three-dimensional space. Thus it has the form $\boldsymbol{S}(u, v) = (x(u, v), y(u, v), z(u, v)), (u, v) \in \mathcal{R}$. There are many schemes for representing surfaces. They differ in the coordinate functions used and the type of region $\mathcal{R}$. Probably the simplest method, and the one most widely used in geometric modelling applications, is the ***tensor product scheme***. This is the method used in this thesis.

The tensor product method is basically a bidirectional curve scheme. It uses basis functions and geometric coefficients. The basis functions are bivariate functions of u and v, which are constructed as products of univariate basis functions.

The geometric coefficients are arranged (topologically) in a bidirectional, $n \times m$ net. Thus a tensor product surface has the form

Eq. 6.6

$$\boldsymbol{S}(u,v) = \big(x(u,v), y(u,v), z(u,v)\big) = \sum_{i=0}^{n}\sum_{j=0}^{m} f_i(u) g_j(v) \boldsymbol{b}_{i,j}$$

where

$$\begin{cases} \boldsymbol{b}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ 0 \le u, v \le 1 \end{cases}$$

Note that the (u,v) domain of this mapping is a square (a rectangle, in general).Note also that $S(\,u,v)$ has a matrix form

Eq. 6.7

$$\boldsymbol{S}(u,v) = [f_i(u)]^T \big[\boldsymbol{b}_{i,j}\big]\big[g_j(v)\big]$$

where $[f_i(u)]^T$ is a $(1) \times (n+1)$ row vector, $\big[g_j(v)\big]$ is a $(m+1) \times (1)$ column vector, and $\big[\boldsymbol{b}_{i,j}\big]$ is a $(n+1) \times (m+1)$ matrix of three-dimensional points.

As an example we consider the power basis surface

Eq. 6.8

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} \boldsymbol{a}_{i,j} u^i v^j = \big[u^i\big]^T \big[\boldsymbol{a}_{i,j}\big]\big[v^j\big] \qquad \begin{cases} \boldsymbol{a}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ 0 \le u, v \le 1 \end{cases}$$

We have $f_i(u) = u^i$ and $g_j(v) = v^j$, and the basis functions are the products, $\{u^i v^j\}$. If we fix $u = u_0$, then

Eq. 6.9

$$\boldsymbol{C}_{u_0}(v) = \boldsymbol{S}(u_0, v) = \sum_{j=0}^{m}\left(\sum_{i=0}^{n} \boldsymbol{a}_{i,j} u_0^i\right) v^j = \sum_{j=0}^{m} \boldsymbol{b}_j(u_0)\, v^j$$

where $\boldsymbol{b}_j(u_0) = \sum_{i=0}^{n} \boldsymbol{a}_{i,j} u_0^i$

is a power basis curve lying on the surface, $\boldsymbol{S}(u,v)$. Similarly, $\boldsymbol{C}_{v_0}(u)$ is a power basis curve lying on $\boldsymbol{S}(u,v)$; and the curves $\boldsymbol{C}_{u_0}(v)$ and $\boldsymbol{C}_{v_0}(u)$ intersect at the surface point, $\boldsymbol{S}(u_0, v_0)$. These curves are called ***isoparametric curves (****or* ***isocurves)****.* $\boldsymbol{C}_{u_0}(v)$ is called a v curve, $\boldsymbol{C}_{v_0}(u)$ a u curve.

Differentiating Eq. 6.8, we obtain

Eq. 6.10

$$\boldsymbol{S}_u(u,v) = \sum_{i=1}^{n}\sum_{j=0}^{m} i\boldsymbol{a}_{i,j}u^{i-1}v^{j} \qquad \boldsymbol{S}_v(u,v) = \sum_{i=0}^{n}\sum_{j=1}^{m} j\boldsymbol{a}_{i,j}u^{i}v^{j-1}$$

Notice that for fixed (u_0, v_0), $\boldsymbol{S}_u(u_0, v_0) = \boldsymbol{C}'_{v_0}(u_0)$ and $\boldsymbol{S}_v(u_0, v_0) = \boldsymbol{C}'_{u_0}(v_0)$. The normal vector, **N**, is computed using Eq. 2.10.

Analogously, a tensor product solid could be defined.

6.3 BEZIER CURVES (LES PIEGEL, WAYNE TILLER, 1997)

Bézier techniques bring sophisticated mathematical concepts into a highly geometric and intuitive form. From a practical standpoint, this form facilitates the creative design process. Equally as important, Bézier techniques are an excellent choice in the context of numerical stability of floating point operations. For these reasons, Bézier techniques are at the core of 3D Modelling or Computer Aided Geometric Design (CAGD).

Since the power basis and Bézier forms both use polynomials for their coordinate functions, they are mathematically equivalent; i.e., any curve that can be represented in one form can also be represented in the other form. However, the Bézier method is superior to the power basis form for geometric modelling. Our presentation of Bézier curves is rather informal; for a more rigorous and complete treatment the reader should consult other references.

The power basis form has the following disadvantages:

- it is unnatural for interactive shape design; the coefficients $\{\boldsymbol{a}_i\}$ convey very little geometric insight about the shape of the curve. Furthermore, a designer typically wants to specify end conditions at both ends of the curve, not just at the starting point;
- algorithms for processing power basis polynomials have an algebraic rather than a geometric flavor (e.g., Horner's method);
- numerically, it is a rather poor form; e.g., Horner's method is prone to round-off error if the coefficients vary greatly in magnitude.

The Bézier method remedies these shortcomings.

6.3.1 PARABOLAS (FARIN, 2002)

We give a simple construction for the generation of a parabola; the straightforward generalization will then lead to Bézier curves. Let $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ be any three points in $\mathbb{E}^3$, and let $u \in \mathbb{R}$. Construct

$$\mathbf{b}_0^1(u) = (1-u)\mathbf{b}_0 + u\mathbf{b}_1,$$

$$\mathbf{b}_1^1(u) = (1-u)\mathbf{b}_1 + u\mathbf{b}_2,$$

$$\mathbf{b}_0^2(u) = (1-u)\mathbf{b}_0^1(u) + u\mathbf{b}_1^1(u).$$

Inserting the first two equations into the third one, we obtain

Eq. 6.11

$$\mathbf{b}_0^2(u) = (1-u)^2\mathbf{b}_0 + 2u(1-u)\mathbf{b}_1 + u^2\mathbf{b}_2$$

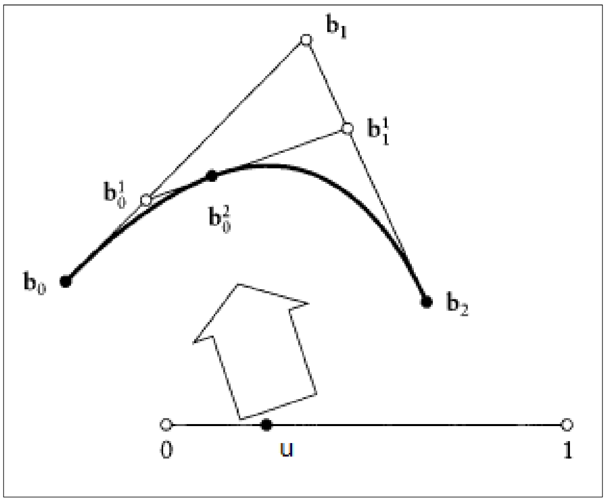


Figure 6.1 Parabolas construction by repeated linear interpolation

This is a quadratic expression in *u* (the superscript denotes the degree), and so $\mathbf{b}_0^2(u)$ traces out a *parabola* as *u* varies from $-\infty$ to $+\infty$. We denote this parabola by $\mathbf{b}^2$. This construction consists of ***repeated linear interpolation***; its geometry is illustrated in Figure 6.1. For *u* between 0 and 1, $\mathbf{b}^2(u)$ is inside the triangle formed by $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$; in particular $\mathbf{b}^2(0) = \mathbf{b}_0$ and $\mathbf{b}^2(1) = \mathbf{b}_2$.

Inspecting the ratios of points in Figure 6.1, we see that

$$ratio(\mathbf{b}_0, \mathbf{b}_0^1, \mathbf{b}_1) = ratio(\mathbf{b}_1, \mathbf{b}_1^1, \mathbf{b}_2) = ratio(\mathbf{b}_0^1, \mathbf{b}_0^2, \mathbf{b}_1^1) = {}^{u}/_{(1-u)}.$$

Thus our construction of a parabola is *affinely invariant,* since piecewise linear interpolation is affinely invariant.

We also note that a parabola is a plane curve, since $\mathbf{b}^2(u)$ is always a barycentric combination of three points, as is clear from inspecting Eq. 6.11. A parabola is a special case of *conic sections*.

Finally we state a theorem from analytic geometry, closely related to our parabola construction. Let **a**,**b**,**c** be the three distinct points on a parabola. Let the tangent at **b** intersect the tangents at **a** and **c** in **e** and **f**, respectively. Let the tangents at **a** and **c** intersect in **d**. Then ratio(**a**,**e**,**d**)=ratio(**e**,**b**,**f**)=ratio(**d**,**f**,**c**). This *three tangent theorem* describes a property of parabolas; the de Casteljau algorithm can be viewed as the constructive counterpart. Figure 6.1, although using a different notation, may serve as an illustration of the theorem.

6.3.2 THE DE CASTELJAU ALGORITHM (FARIN, 2002)

Parabolas are plane curves. However, many applications require true space curves. For those purposes, the previous construction for a parabola can be generalized to generate a polynomial curve of arbitrary degree n:

de Casteljau algorithm:

Given: $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{E}^3$ and $u \in \mathbb{R}$, ***set***

Eq. 6.12

$$\mathbf{b}_i^r(u) = (1-u)\mathbf{b}_i^{r-1}(u) + u\mathbf{b}_{i+1}^{r-1}(u) \quad r = 1, \dots, n \; and \; i = 0, \dots, n-r$$

and $\mathbf{b}_i^0(u) = \mathbf{b}_i$. Then $\boldsymbol{b}_0^n(u)$ is the point with the parameter value *u* on the ***Bézier curve*** $\mathbf{b}^n$, hence $\mathbf{b}^n(u) = \mathbf{b}_0^n(u)$.

The polygon **P** formed by $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ is called the ***Bézier polygon*** or ***control polygon*** of the curve $\mathbf{b}^n$. Similarly, the polygon vertices $\mathbf{b}_i$ are called ***control points*** or ***Bézier points.*** Figure 6.2 illustrates the cubic case.

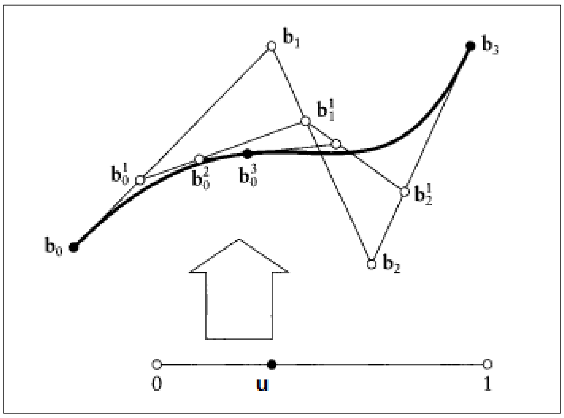


Figure 6.2 The de Casteljau algorithm: the point $\mathbf{b}_0^3(\mathbf{u})$ is obtained from repeated linear interpolation. The cubic case n=3 is shown for u=1/3

Properties

Affine invariance. An important property of Bézier curves is that they are invariant under affine maps, which means that the following two procedures yield the same result: (1) first, compute the point $\mathbf{b}^n(u)$ and then apply an affine map to it; (2) first, apply an affine map to the control polygon and then evaluate the mapped polygon at parameter value u.

Affine invariance is, of course, a direct consequence of the de Casteljau algorithm: the algorithm is composed of a sequence of linear interpolations (or, equivalently, of a sequence of affine maps). These are themselves affinely invariant, and so is a finite sequence of them.

Let us discuss a practical aspect of affine invariance. Suppose we plot a cubic curve $\mathbf{b}^3$ by evaluating at 100 points and then plotting the resulting point array. Suppose now that we would like to plot the curve after a rotation has been applied to it. We can take the 100 computed points, apply the rotation to each of them, and plot. Or, we can apply the rotation to the 4 control points, then evaluate 100 times and plot. The first method needs 100 applications of the rotation, whereas the second needs only 4!

Invariance under affine parameter transformations. Very often, one thinks of a Bézier curve as being defined over the interval [0,1]. This is done because it is convenient, not because it is necessary: the de Casteljau algorithm is "blind" to the actual interval that the curve is defined over because it uses ratios only. One may therefore think of the curve as being defined over any arbitrary interval $a \leq u \leq b$ of the real line—after the introduction of local coordinates $\bar{u} = (u — a)/(b — a)$, the algorithm proceeds as usual. This property is inherited from the linear interpolation process. The corresponding generalized de Casteljau algorithm is of the form:

Eq. 6.13

$$\mathbf{b}_i^r(\bar{u}) = \frac{b-\bar{u}}{b-\bar{u}}\mathbf{b}_i^{r-1}(\bar{u}) + \frac{\bar{u}-a}{b-a}\mathbf{b}_{i+1}^{r-1}(\bar{u}).$$

The transition from the interval [0,1] to the interval $[a, b]$ is an ***affine map***. Therefore, we can say that Bézier curves are invariant under affine parameter transformations. Sometimes, one sees the term ***linear parameter transformation*** in this context, but this terminology is not quite correct: the transformation of the interval [0,1] to $[a, b]$ typically includes a translation, which is not a linear map.

Convex hull property. For $u \in [0,1], \mathbf{b}^n(u)$ lies in the convex hull of the control polygon. This follows since every intermediate $\mathbf{b}_i^r$ is obtained as a convex barycentric combination of previous $\mathbf{b}_j^{r-1}$—at no step of the de Casteljau algorithm do we produce points outside the convex hull of the $\mathbf{b}_i$.

A simple consequence of the convex hull property is that a planar control polygon always generates a planar curve.

The importance of the convex hull property lies in what is known as ***interference checking***. Suppose we want to know if two Bézier curves intersect each other—for example, each might represent the path of a robot arm, and our aim is to make sure that the two paths do not intersect, thus avoiding expensive collisions of the robots. Instead of actually computing a possible intersection, we can perform a much cheaper test: circumscribe the smallest possible box around the control polygon of each curve such that it has its edges parallel to some coordinate system. Such boxes are called *minmax boxes,* since their faces are created by the minimal and maximal coordinates of the control polygons. Clearly each box contains its control polygon, and, by the convex hull property, also the corresponding Bézier curve. If we can verify that the two boxes do not overlap (a trivial test), we are assured that the two curves do not intersect. If the boxes do overlap, we would have to perform more checks on the curves. The possibility for a quick decision of no interference is extremely important, since in practice one often has to check one object against thousands of others, most of which can be labeled "no interference" by the minmax box test.

Endpoint interpolation. The Bézier curve passes through $\mathbf{b_0}$ and $\mathbf{b_n}$: we have $\mathbf{b}^n(0) = \mathbf{b}_0, \mathbf{b}^n(1) = \mathbf{b}_n$. In a design situation, the endpoints of a curve are certainly two very important points. It is therefore essential to have direct control over them, which is assured by endpoint interpolation.

Variation diminishing property. no straight line intersects a curve more times than it intersects the curve's control polygon (for a three dimensional Bezier curve, replace the words 'straight line' with the word 'plane'). This expresses the property that a Bezier curve follows its control polygon rather closely and does not wiggle more than its control polygon.(Figure 6.3)

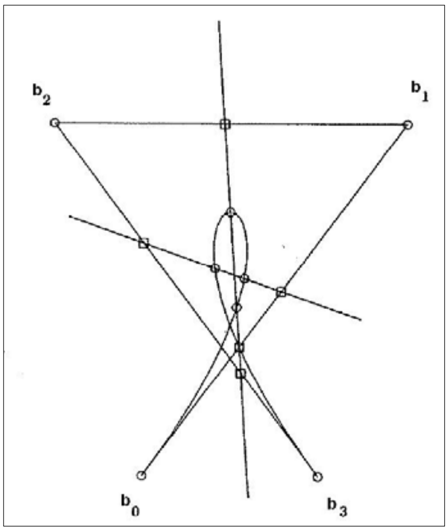


Figure 6.3 Cubic Bézier curve

6.3.3 BERNSTEIN POLYNOMIALS

We can also express Bézier curves in an explicit form. An nth-degree Bézier curve is defined by a basis of Bernstein polynomials and the n+1 data points to be approximated

Eq. 6.14

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} B_{i,n}(u)\, \boldsymbol{P_i} \quad 0 \le u \le 1$$

The basis (blending) functions, $\{B_{i,n}(u)\}$, are the classical nth-degree Bernstein polynomials given by

Eq. 6.15

$$B_{i,n}(u) = \frac{n!}{i!\,(n-i)!} u^i (1-u)^{n-i} = \binom{n}{i} u^i (1-u)^{n-i}$$

Or the recursive formula

Eq. 6.16

$$B_{i,n}(u) = (1-u)B_{i,n-1}(u) + uB_{i-1,n-1}(u)\,, \;\; B_{i,n} \equiv 0 \; if \; i < 0 \; or \; i > n$$

The geometric data of this form, $\{\boldsymbol{P_i}\}$, are called ***control points***, and are the data points in the physical space $\mathbb{E}^3$to be approximated. By weighting all the data points with the Bernstein polynomials the point $\boldsymbol{C}(u)$ is produced in the same space as the data points, thus $\boldsymbol{C}(u) \in \mathbb{E}^3$ for a specific parametric ordinate u. This mapping procedure from the parametric domain of u to the physical space $\mathbb{E}^3$ gives a full curve in the physical space when the entire parametric domain $0 \le u \le 1$ is mapped.

These functions have the properties following:

P1.1 partition of unity: $\sum_{i=0}^{n} B_{i,n}(u) = 1$ for all $0 \leq u \leq 1$ this property of the Bernstein basis are used to form barycentric combinations of the data points; thus any affine map imposed on the curve leaves intact the basis and is applied directly on the control points (see section 5). This property is the most important in CAGD, where affine mappings are used frequently (rotate, scale, stretch, shear, translation, mirror, etc.). This property seems to be crucial for the analysis too, since in linear elasticity problems any infinitesimal particle of a body can undergo three basic affine mappings (translation, stretch and shear). The non-linear kinematics can be described by some overlapped affine transformations which are essentially the higher degree Bernstein basis functions;

P1.2 non-negativity: $B_{i,n}(u) \geq 0$ for all i, n and $0 \leq u \leq 1$ this property of the Bernstein basis lends to formulation of convex combinations of the data points. Thus, the convex hull property of the Bezier curves is well illustrated. This property renders stable numerical procedures;

P1.3 $B_{0,n}(0) = B_{n,n}(1) = 1$ This is a property that gives rise to an easy self-definition of CAGD and analysis boundary conditions

P1.4 $B_{i,n}(u)$ attains exactly one maximum in the interval [0, 1] that is, at u=i/n;

P1.5 symmetry: for any n, the set of polynomials $\{B_{i,n}(u)\}$ is symmetric with respect to u = ½ , thus the Bernstein basis can form symmetric matrices in analysis;

P1.6 recursive definition: $B_{i,n}(u) = (1-u)B_{i,n-1}(u) + uB_{i-1,n-1}(u)$; we define $B_{i,n}(u) \equiv 0 \; if \; i < 0 \; or \; i > n$;

P1.7 derivatives:

$$B'_{i,n}(u) = \frac{dB_{i,n}(u)}{du} = n(B_{i-1,n-1}(u) - B_{i,n-1}(u))$$

with $B_{-1,n-1}(u) \equiv B_{n,n-1}(u) \equiv 0$

$$\sum_{i=0}^{n} B^{(k)}{}_{i,n}(u) = 0$$

This property gives the ability to form easily vectors using the derivative basis and the same data points, thus vector fields (gradients) can be easily defined in analysis procedures.

In any curve (or surface) representation formula, the choice of the basis functions determines the geometric characteristics of the curve. Figure 6.4 and Figure 6.5 show the Bernstein basis functions $\{B_{i,n}(u)\}$ for *n* =3.

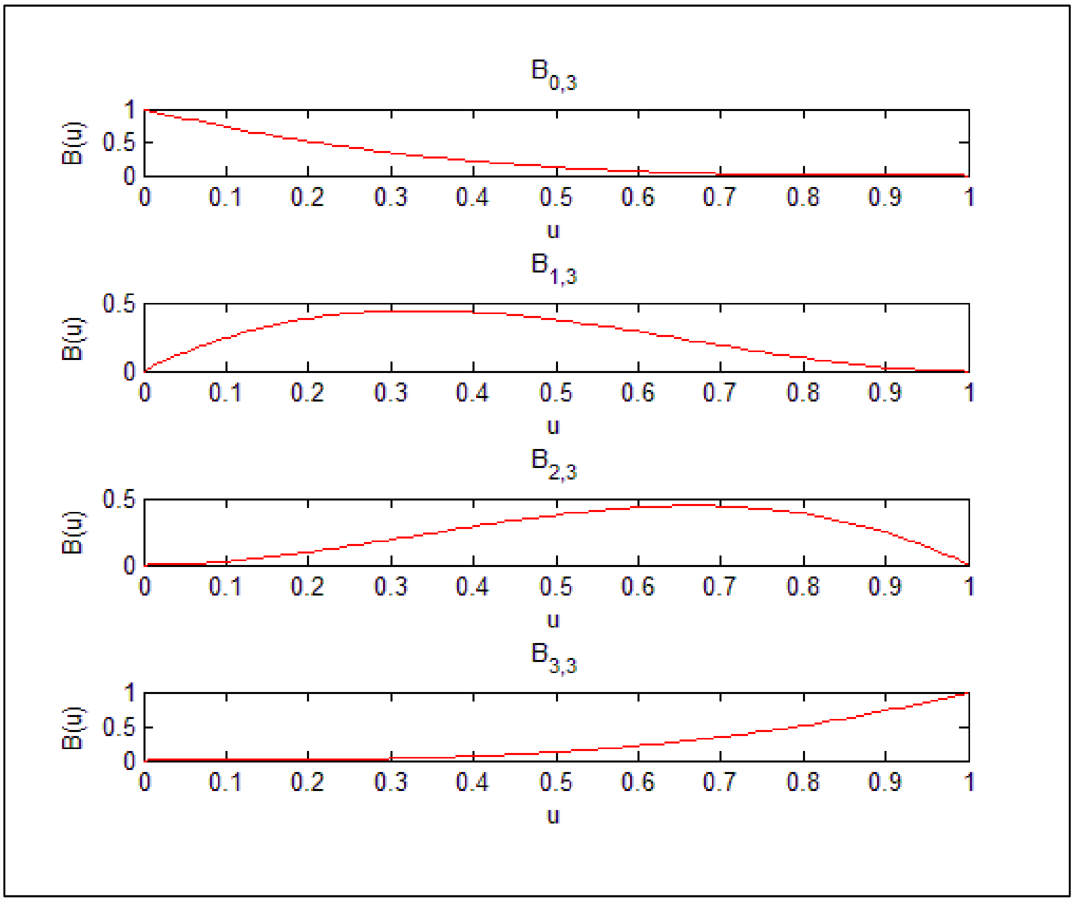


Figure 6.4 The Bernstein basis functions of degree 3, one by one

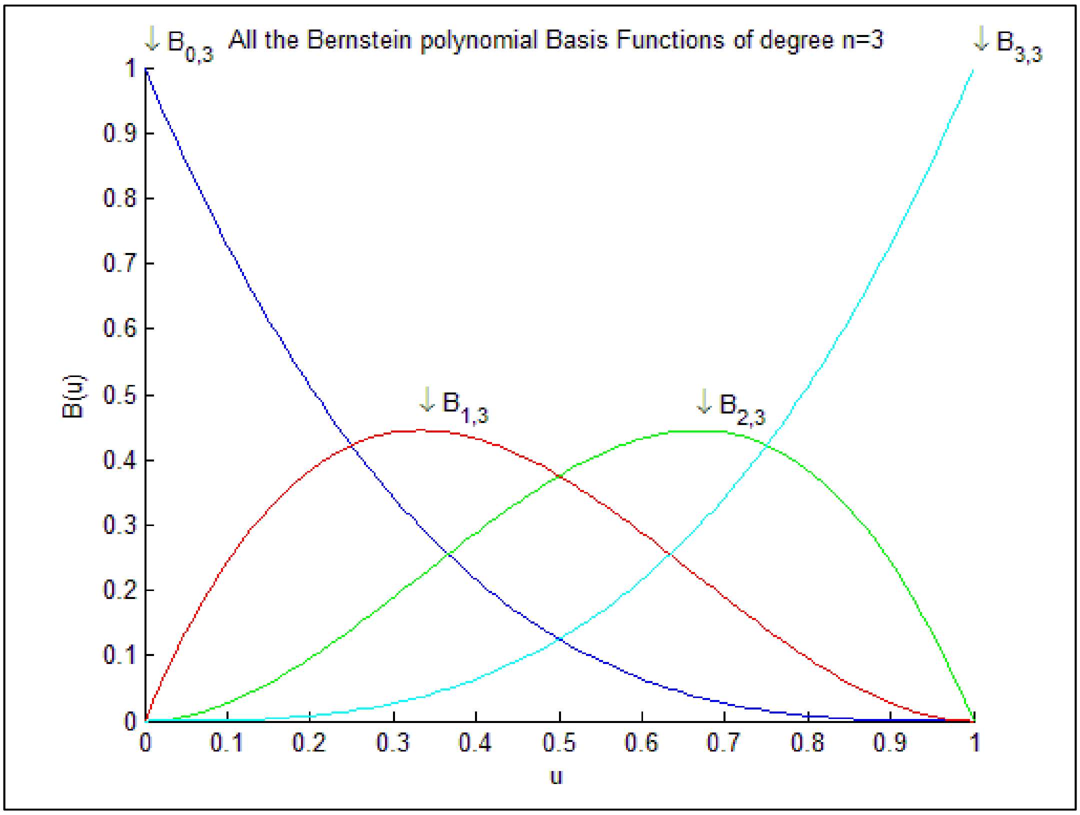


Figure 6.5 All the Bernstein Basis Functions of degree 3

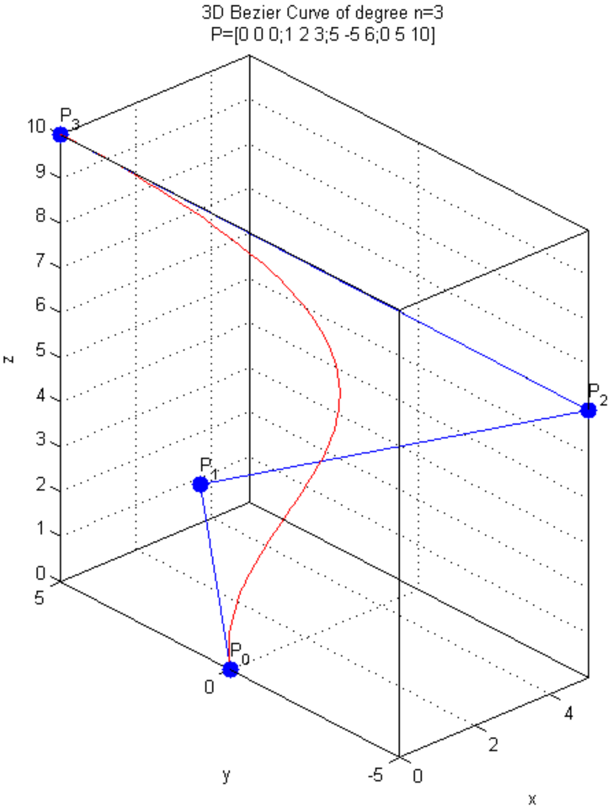


Figure 6.6 A 3rd degree Bezier curve, built using the Bernstein basis functions of Figure 6.5

The recursive formula for generating the Bernstein polynomials of Eq. 6.16 is described by the linear interpolation of two Bernstein polynomials of one degree lower. That is a linear transformation and thus an affine transformation. This sequential affine transformation of the basis conserves all the highly important properties of an initial affine transformation. This initial basis is the linear interpolation of the identity function. The identity function maps a point to itself. Thus, possessing a collection of constant functions (corresponding to data points), one can apply a linear interpolation to the points, consequently to the constant functions. This is the beginning of blossoming new basis functions by linear interpolations of the latest ones, exactly as it is geometrically represented by the De Casteljau algorithm (see section 6.3.2). In figures below, the procedure described above is shown graphically beginning with the initial constant function and ending up to the 3rd degree of Bernstein polynomials.

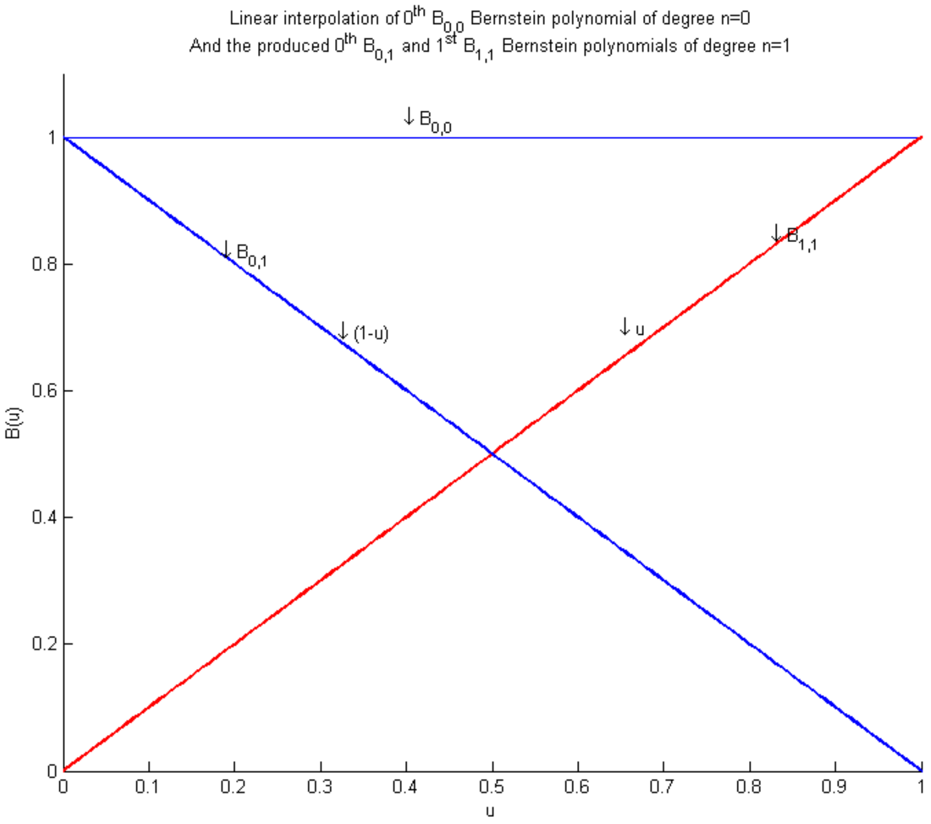


Figure 6.7

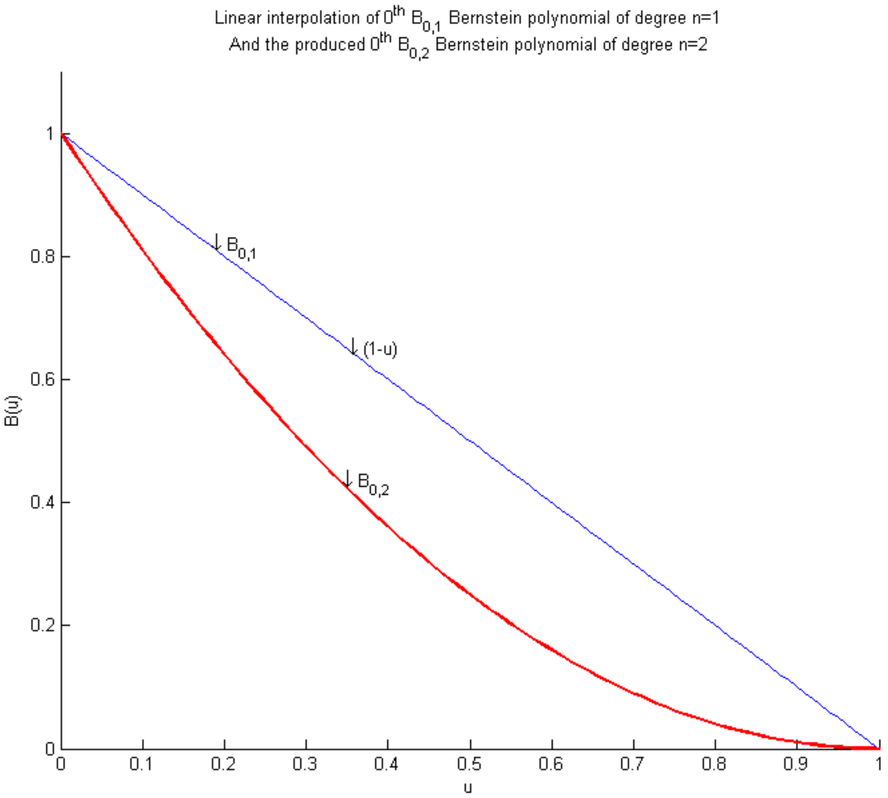


Figure 6.8

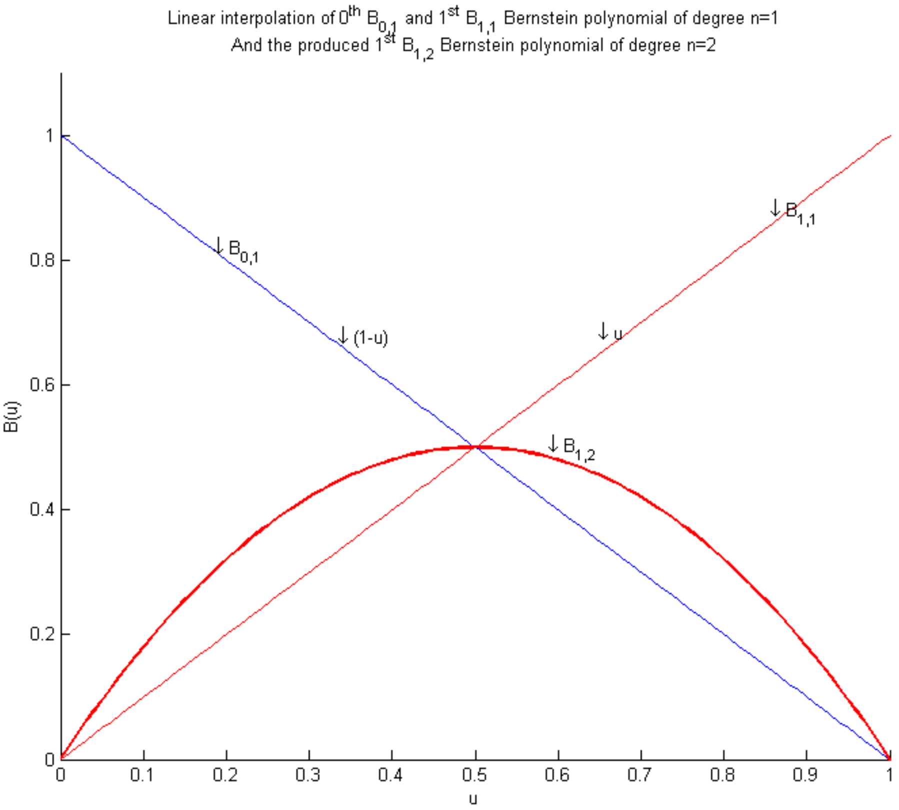


Figure 6.9

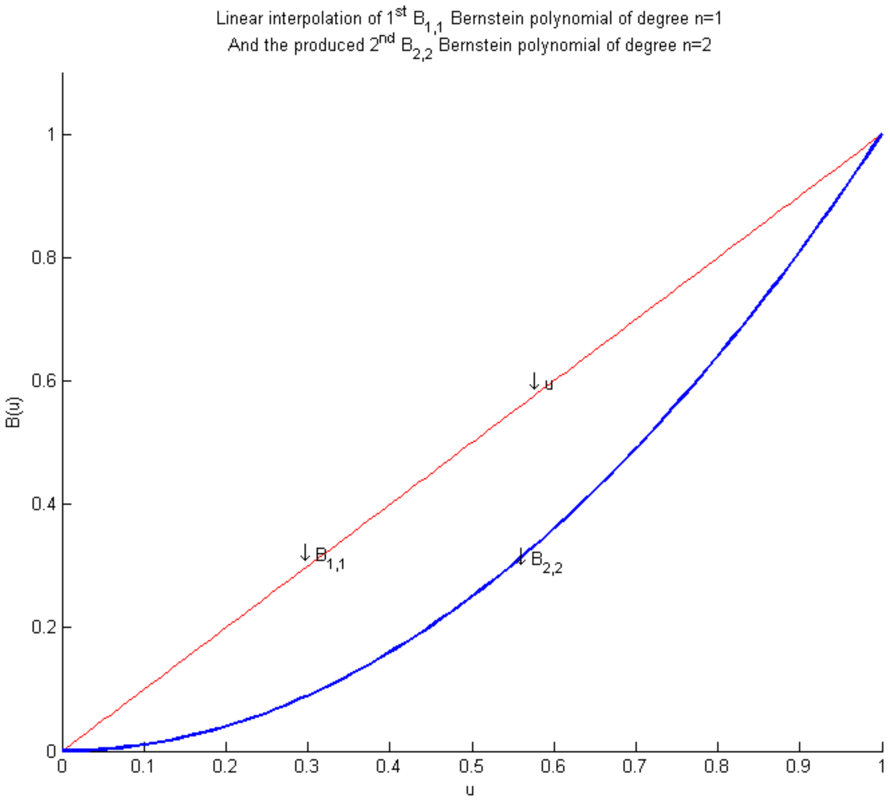


Figure 6.10

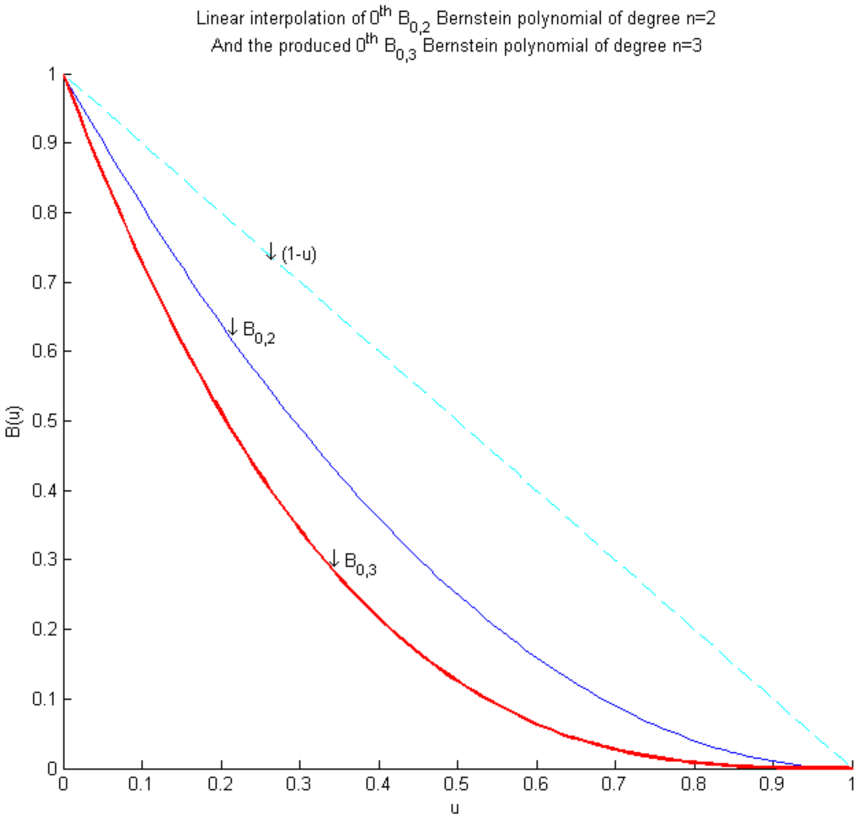


Figure 6.11

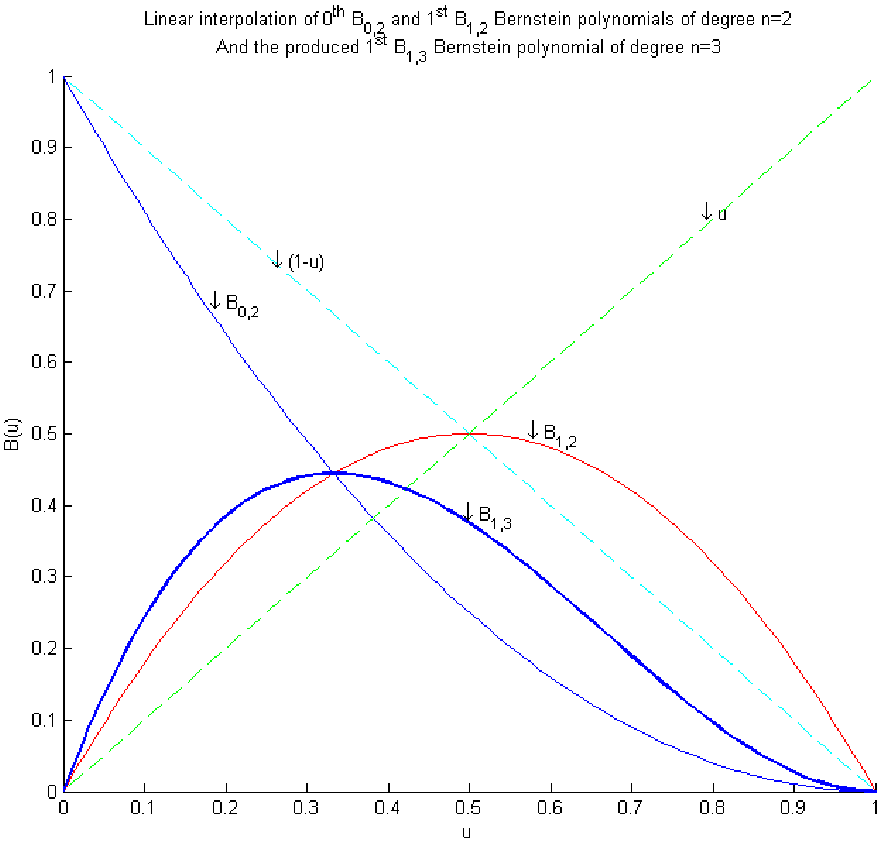


Figure 6.12

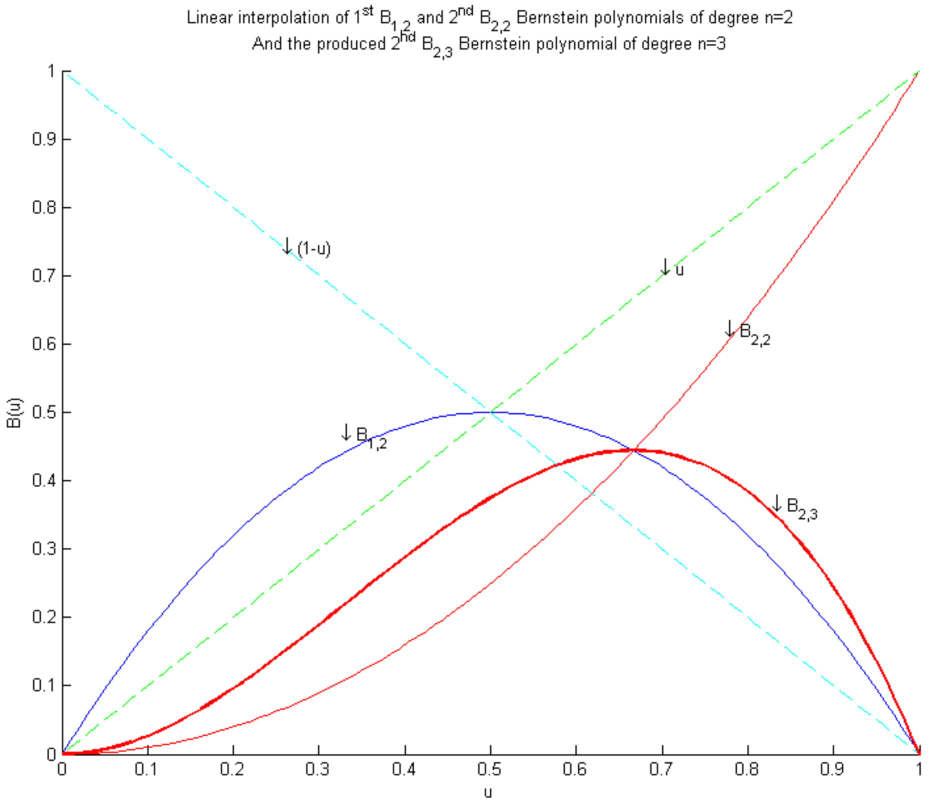


Figure 6.13

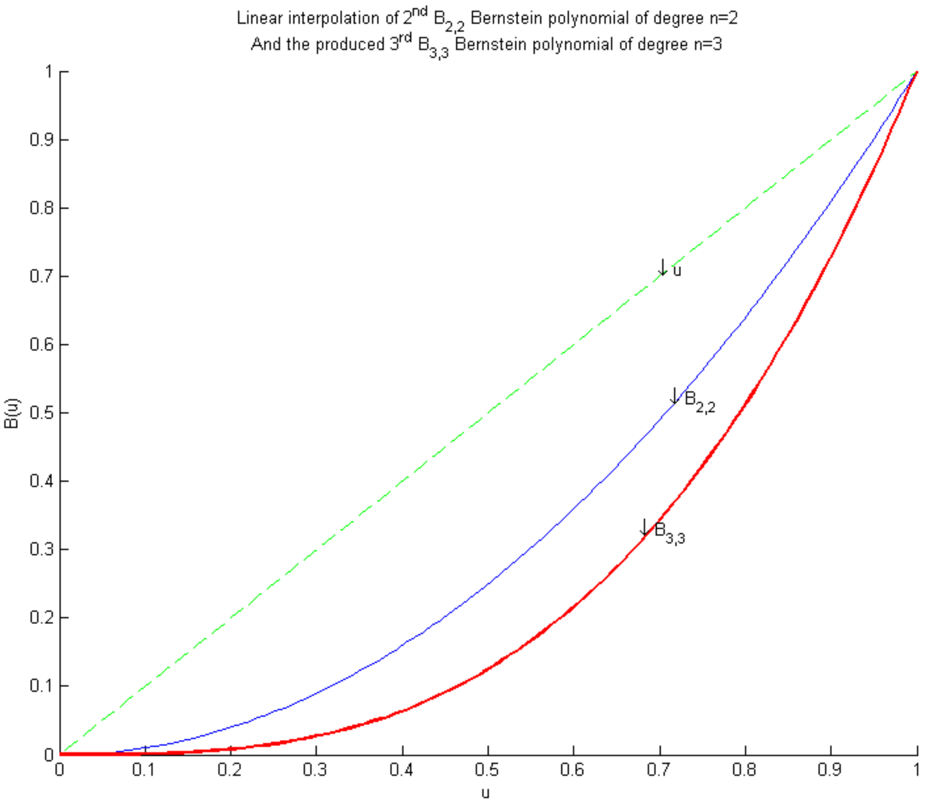


Figure 6.14

Addition properties of Bézier Curves

The properties mentioned in section 6.3.2, which were derived using geometric arguments, could be also derived using algebraic arguments along with a number of some other properties.

Symmetry. Looking at the examples in Figure 6.15, it is clear that it does not matter if the Bézier points are labelled $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$ or $\mathbf{P}_n, \mathbf{P}_{n-1}, \dots, \mathbf{P}_0$.The curves that correspond to the two different orderings look the same; they differ only in the direction in which they are traversed. Written as a formula:

Eq. 6.17

$$\sum_{i=0}^{n} \mathbf{P}_i B_{i,n}(u) = \sum_{i=0}^{n} \mathbf{P}_{n-i} B_{i,n}(1-u)$$

This follows from the identity

Eq. 6.18

$$B_{i,n}(u) = B_{n-i,n}(1-u),$$

which follows from inspection of Eq. 6.15. We say that Bernstein polynomials are symmetric with respect to u and 1 — u.

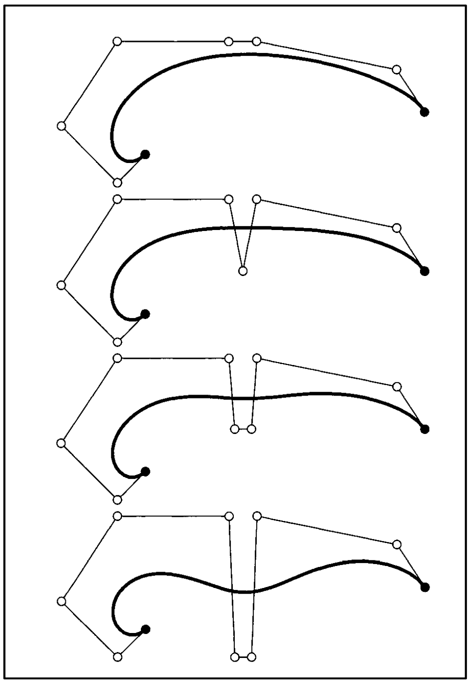


Figure 6.15 Bezier curves: some examples of how the shape of the curve is affected by the distribution of the control points

Pseudolocal control. The Bernstein polynomial $B_{i,n}$ has only one maximum and attains it at $u = i/n$. This has a design application: if we move only one of the control polygon vertices, say, $\mathbf{P}_i$ then the curve is mostly affected by this change in the region of the curve around the parameter value i/n. This makes the effect of the change reasonably predictable, although the change does affect the whole curve. As a rule of thumb, the maximum of each $B_{i,n}$ is roughly 1/3; thus a change of $\mathbf{P}_i$ by three units will change the curve by one unit.

Using property P1.7, it is easy to derive the general expression for the derivative of a Bézier curve

Eq. 6.19

$$C'(u) = \frac{d\left(\sum_{i=0}^{n} B_{i,n}(u)\,\mathbf{P}_i\right)}{du} = \sum_{i=0}^{n} B'_{i,n}(u)\,\mathbf{P}_i$$

$$= \sum_{i=0}^{n} n\left(\left(B_{i-1,n-1}(u) - B_{i,n-1}(u)\right)\right)\mathbf{P}_i = n\sum_{i=0}^{n-1} B_{i,n-1}(u)(\mathbf{P}_{i+1} - \mathbf{P}_i)$$

From Eq. 6.19 we easily obtain formulas for the end derivatives of a Bézier curve, e.g.

Eq. 6.20

$$C'(0) = n(\mathbf{P}_1 - \mathbf{P}_0) \qquad C''(0) = n(n-1)(\mathbf{P}_0 - 2\mathbf{P}_1 + \mathbf{P}_2)$$

$$C'(1) = n(\mathbf{P}_n - \mathbf{P}_{n-1}) \quad C''(1) = n(n-1)(\mathbf{P}_n - 2\mathbf{P}_{n-1} + \mathbf{P}_{n-2})$$

Notice from Eq. 6.19 and Eq. 6.20 that

- the derivative of an nth degree Bézier curve is an (n-1)th degree Bézier curve;
- the expressions for the end derivatives at $u = 0$ and $u = 1$ are symmetric (due,of course, to the symmetry of the basis functions);
- the kth derivative at an endpoint depends (in a geometrically very intuitive manner) solely on the k+1 control points at that end.

6.3.4 RATIONAL BÉZIER CURVES

Conic sections may be expressed as rational quadratic (Bézier) curves, and their generalization to higher-degree rational curves is quite straightforward: a rational Bézier curve of degree n in $\mathbb{E}^3$ is the projection of an n^{th} degree Bézier curve in $\mathbb{E}^4$ into the hyperplane $w = 1$. We may view this 4D hyperplane as a copy of $\mathbb{E}^3$; we assume that a point in $\mathbb{E}^4$ is given by its coordinates $[x\ y\ z\ w]^T$. It can be easily proved that an nth degree rational Bézier curve is given by

Eq. 6.21

$$\boldsymbol{C}(u) = \frac{w_0 \boldsymbol{P}_0 B_{0,n}(u) + \cdots + w_n \boldsymbol{P}_n B_{n,n}(u)}{w_0 B_{0,n}(u) + \cdots + w_n B_{n,n}(u)};\ \boldsymbol{C}(u), \boldsymbol{P}_i \in \mathbb{E}^3$$

w_i are called ***weights***; the $\boldsymbol{P}_i$form the control polygon. It is the projection of the 4D control polygon $[w_i \boldsymbol{P}_i\ \ w_i]^T$ of the nonrational 4D preimage of $\boldsymbol{C}(u)$.

If each weight equals to one, we obtain the standard nonrational Bézier curve, since the denominator is identically equal to one (This is also true if the weights are not unity, but are equal to each other—a common factor does not matter.). If some w_i are negative, singularities may occur; we will therefore deal only with nonnegative w_i. Rational Bézier curves enjoy all the properties that their nonrational counterparts possess; for example, they are affinely invariant. We can see this by rewriting Eq. 6.21 as

Eq. 6.22

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} \boldsymbol{P}_i \frac{w_i B_{i,n}(u)}{\sum_{i=0}^{n} w_i B_{i,n}(u)}.$$

We see that the basis functions

Eq. 6.23

$$\frac{w_i B_{i,n}(u)}{\sum_{i=0}^{n} w_i B_{i,n}(u)}$$

sum to one identically, thus asserting affine invariance. If all w_i are nonnegative, these basis functions hold the convex hull property. We also have symmetry, invariance under affine parameter transformations, endpoint interpolation, and the variation diminishing property.

The w_i are typically used as ***shape parameters***. If we increase one of the w_i, the curve is pulled toward the corresponding $\boldsymbol{P}_i$, as illustrated in Figure 6.16. Note that the effect of changing a weight is different from that of moving a control vertex. If we let all weights tend to infinity at the same rate, we do not approach the control polygon since a common (if large) factor in the weights does not matter.

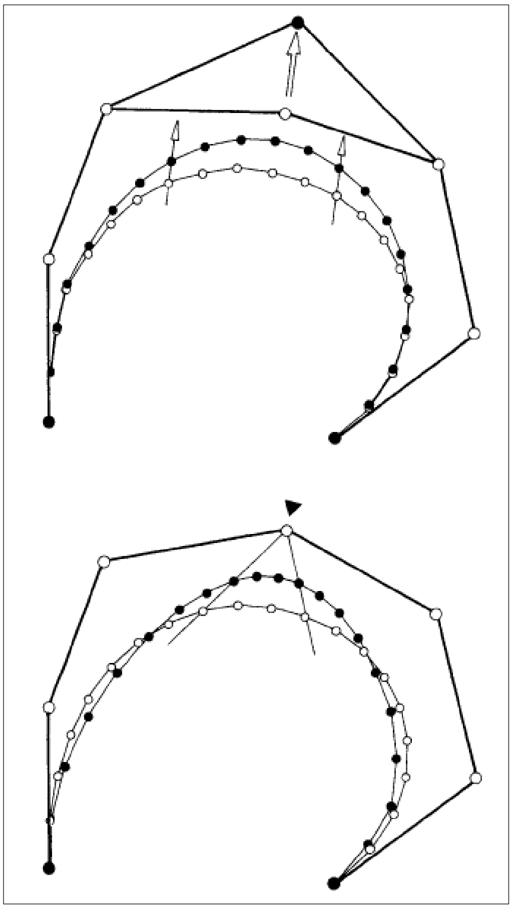


Figure 6.16 Influence of the weights: top, changing one control point; bottom, changing one weight

Two properties differ from the nonrational case. First, we have ***projective*** invariance; that is, if a rational Bézier curve is transformed by a projective transformation, we could just as well apply that transformation to the control polygon (using its weights to write it in homogeneous form) and would end up with the same curve. Note that nonrational curves have this property only for a subset of all projective maps, that is, the affine maps. The second difference is the ***linear precision property***. Rational curves may have all Bezier points $\boldsymbol{P}_i$ distributed on a straight line in a totally arbitrary fashion:

$$\boldsymbol{P}_i = (1 - a_i)\boldsymbol{P}_0 + a_i\boldsymbol{P}_n$$

with arbitrary real numbers a_i. We can still find weights w_i such that the resulting curve traces out the straight line $\overline{\boldsymbol{P}_0\boldsymbol{P}_n}$ in a ***linear*** fashion. They are given by $w_0 = 1$ and

$$w_i = \frac{i}{n+1-i}\frac{1-a_{i-1}}{a_i}w_{i-1}; i = 1, \dots, n\,.$$

Rational Bézier curves may be modified in another way. Let us define ***weight points*** $\boldsymbol{q}_i$, by setting

Eq. 6.24

$$\boldsymbol{q}_i = \frac{w_i\boldsymbol{P}_i + w_{i+1}\boldsymbol{P}_{i+1}}{w_i + w_{i+1}}.$$

These points are defined via the weights; they may be used as shape parameters for the curve. If we change the location of one of the $\boldsymbol{q}_i$, we may recompute a new set of weights by setting $w_0 = 1$ and using Eq. 6.24 as a recursion for the w_i.

In the following figures, $R_{i,j}$ denotes a rational Bernstein polynomial basis function, which is computed by Eq. 6.23.

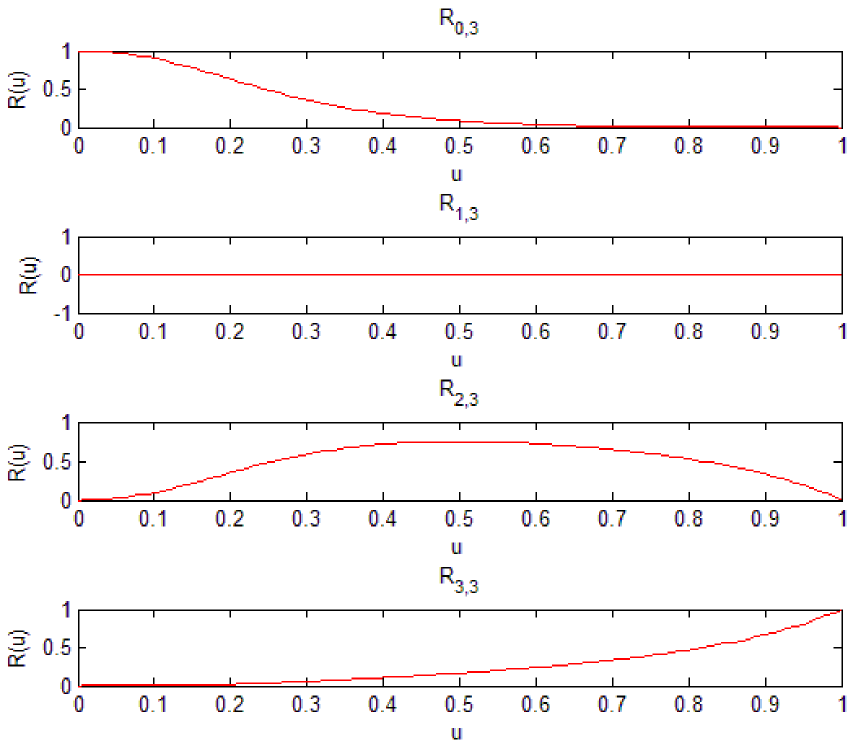


Figure 6.17 Rational Bernstein polynomial Basis Functions, weight vector : [1 0 3 2]

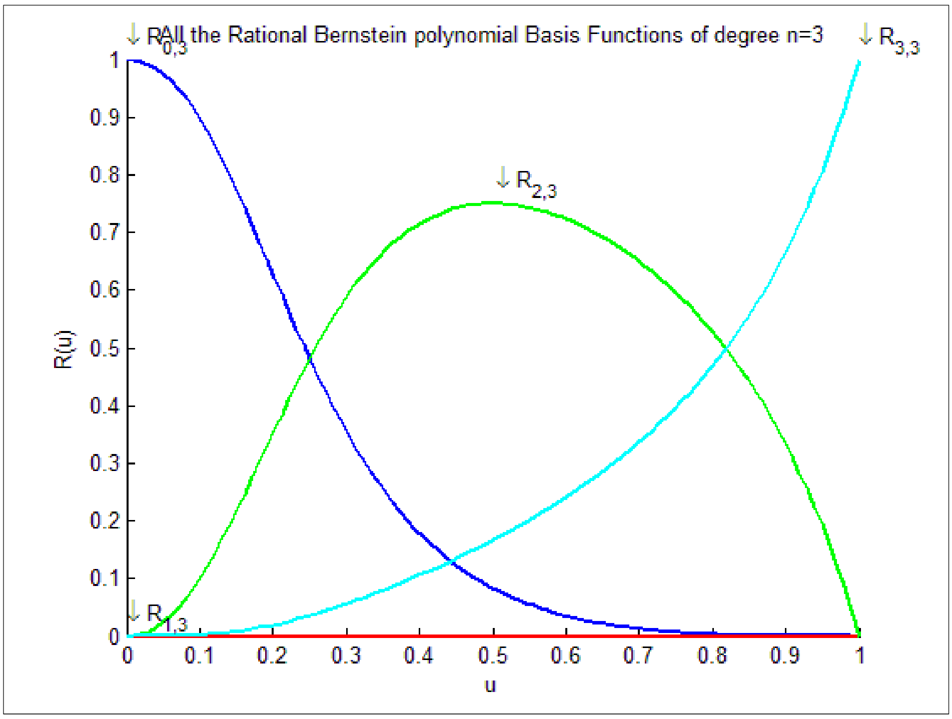


Figure 6.18 Rational Bernstein polynomial basis functions of degree 3. weight vector : [1 0 3 2]

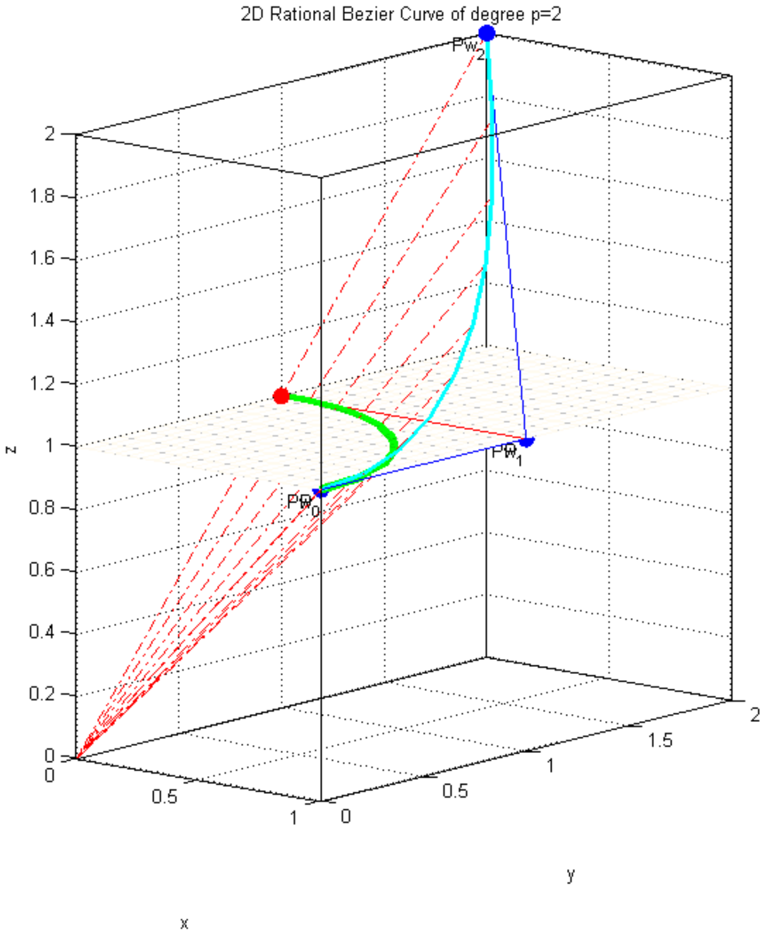


Figure 6.19 An arc of 90° computed by mapping a curve from the 3D space into the 2D

6.3.5 BÉZIER SURFACES

Nonrational Bézier surfaces are obtained by considering a bidirectional net of control points and products of the univariate Bernstein polynomials

Eq. 6.25

$$\boldsymbol{S}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}B_{i,n}(u)B_{j,m}(v)\boldsymbol{P}_{i,j}\qquad 0\le u,v\le 1$$

The basis function $B_{0,3}(u)B_{1,4}(v)$ is shown in Figure 6.20, and Figure 6.21 shows a cubic x quatric Bézier surface.

For fixed $u=u_0$

$$\boldsymbol{C}_{u_0}(v)=\boldsymbol{S}(u_0,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}B_{i,n}(u_0)B_{j,m}(v)\boldsymbol{P}_{i,j}=\sum_{j=0}^{m}B_{j,m}(v)\left(\sum_{i=0}^{n}B_{i,n}(u_0)\boldsymbol{P}_{i,j}\right)$$
$$=\sum_{j=0}^{m}B_{j,m}(v)\boldsymbol{C}_j(u_0)$$

where $\boldsymbol{C}_j(u_0)=\sum_{i=0}^{n}B_{i,n}(u_0)\boldsymbol{P}_{i,j}$, $j=0,\dots,m$ is a Bézier curve lying on the surface. Analogously, $\boldsymbol{C}_{v_0}(u)=\sum_{i=0}^{n}B_{i,n}(u)\boldsymbol{C}_i(v_0)$ is a Bézier u isocurve lying on the surface.

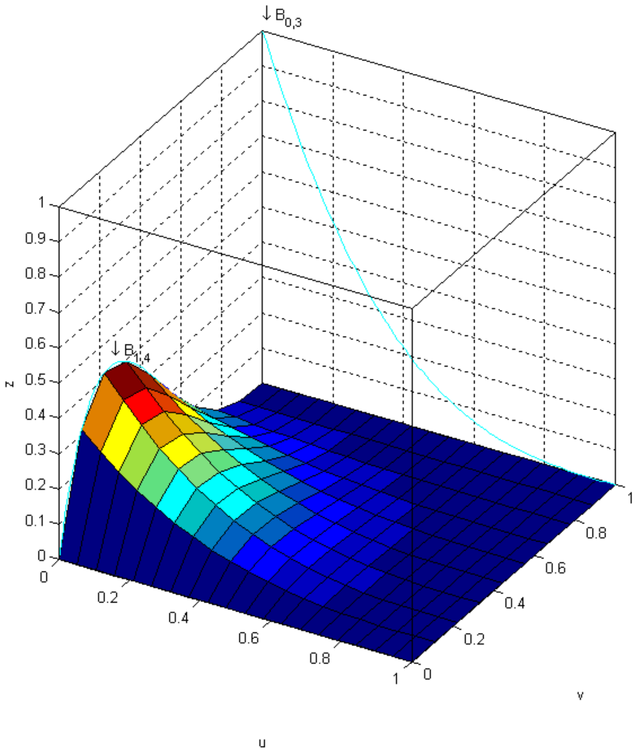


Figure 6.20 A bivariate Bézier basis function, $B_{0,3}(u)B_{1,4}(v)$, is a product of two univariate Bézier basis functions

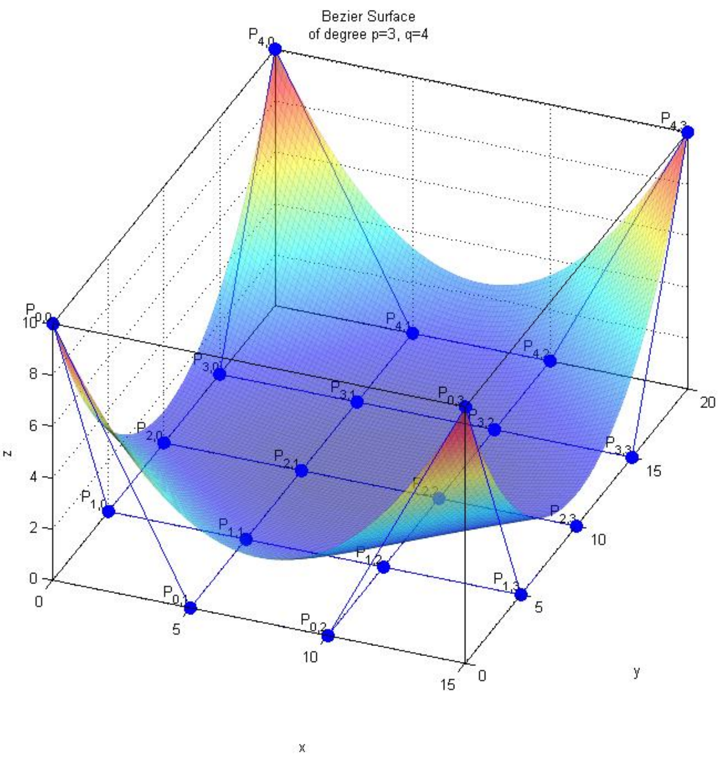


Figure 6.21 A Bézier surface computed on a bivariate basis

As is the case for curves, because of their excellent properties, Bézier surfaces are better suited for geometric modelling applications than power basis surfaces. In particular,

- nonnegativity : $B_{i,n}(u)B_{j,m}(v) \geq 0$ for all i, j, u, v;
- partition of unity : $\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = 1$ for all u and v;
- $\boldsymbol{S}(u, v)$ is contained in the convex hull of its control points;
- transformation invariance;
- the surface interpolates the four corner control points;
- when triangulated, the control net forms a planar polyhedral approximation to the surface.

It is interesting to note that there is no known variation diminishing property for Bézier surfaces.

We define a ***rational Bézier surface*** to be the perspective projection of a four-dimensional polynomial Bézier surface

Eq. 6.26

$$\boldsymbol{S}^{\boldsymbol{w}}(u, v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v)\, \boldsymbol{P}_{i,j}^{\boldsymbol{w}}$$

and

$$\boldsymbol{S}(u,v) = H\{\boldsymbol{S}^{\boldsymbol{w}}(u,v)\} = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v)w_{i,j}\,\boldsymbol{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v)\,w_{i,j}} = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)\boldsymbol{P}_{i,j}$$

where $R_{i,j}(u,v) = \frac{B_{i,n}(u)B_{j,m}(v)w_{i,j}}{\sum_{r=0}^{n}\sum_{s=0}^{m} B_{r,n}(u)B_{s,m}(v)w_{r,s}}$

Notice that the $R_{i,j}(u,v)$ are rational functions, but they are not products of other basis functions. Hence, $\boldsymbol{S}(u,v)$ is not a tensor product surface, but $\boldsymbol{S}^{\boldsymbol{w}}(u,v)$ is. As with curves, we generally work with Eq. 6.26 and project the results.

Assuming $w_{i,j} > 0$ for all i and j, the properties listed previously for nonrational Bézier surfaces (and the product functions $B_{i,n}(u)B_{j,m}(v)$) extend naturally to rational Bézier surfaces. Furthermore, if $w_{i,j} = 1$ for all i and j, then $R_{i,j}(u,v) = B_{i,n}(u)B_{j,m}(v)$, and the corresponding surface is nonrational.

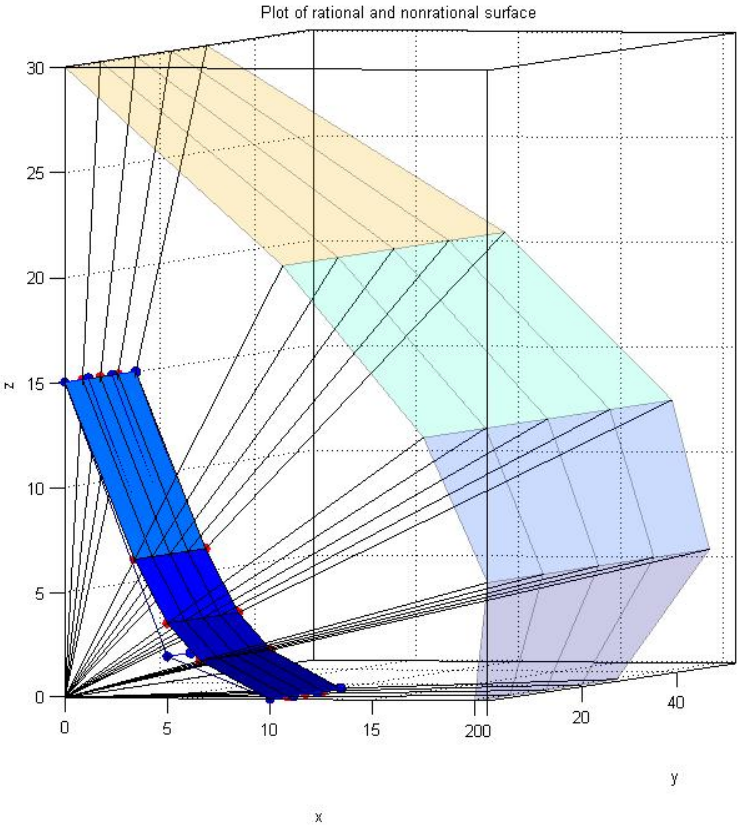


Figure 6.22 The nonrational surface (transparent) in homogeneous coordinates projected to a rational Bézier surface

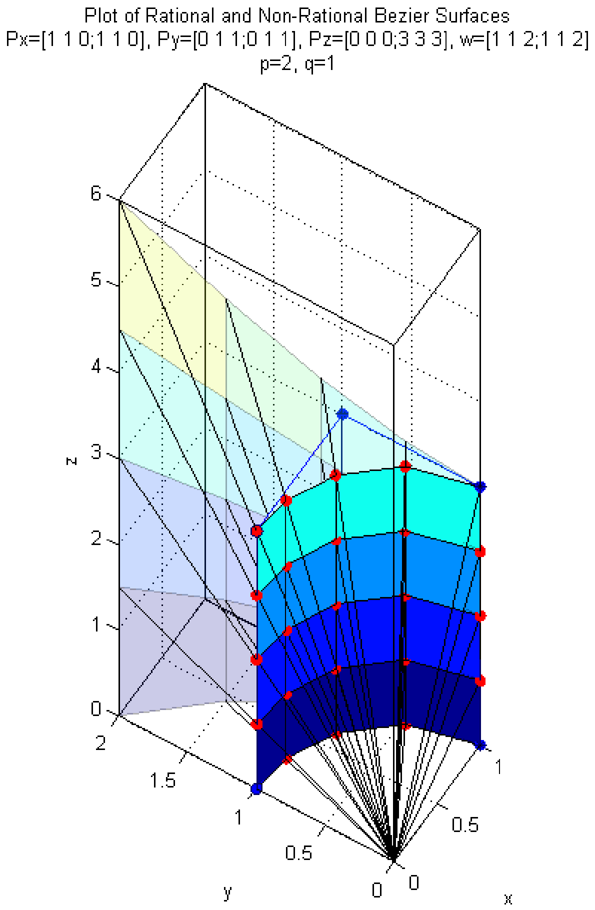


Figure 6.23 The projective non-rational Bezier surface (transparent) and the corresponding rational Bezier surface generated, representing a quarter of cylinder

7 B-SPLINES (PIECEWISE RATIONAL AND NON-RATIONAL POLYNOMIALS)

Curves consisting of just one polynomial or rational segment are often inadequate, since they have some shortcomings:

- A high degree is required in order to satisfy a large number of constraints. However, high degree curves are inefficient to process and are numerically unstable.
- A high degree is required to accurately fit some complex shapes.
- Single-segment curves (surfaces) are not well-suited to interactive shape design. Although Bezier curves can be shaped by means of their control points and weights, the control is not sufficiently local.

The solution is to use curves (surfaces) which are ***piecewise polynomial***, or ***piecewise rational***. Such curves (surfaces) are defined on $u \in [0,1]$ and parameterized by some internal values u_i, which are called breakpoints, $0 \leq u_{i-1} \leq u_i \leq u_{i+1} \leq 1$. They map into the endpoints of the polynomial segments. We denote the segments by $\mathbf{C}_\mathrm{i}(\mathrm{u}),\ 1 \leq i \leq m$ when the curve $\mathbf{C}(\mathrm{u})$ consists of m polynomial segments of the same degree. The segments are constructed so that they join with some level of continuity (not necessarily the same at every breakpoint). Let $\mathbf{C}_\mathrm{i}{}^{(j)}(\mathrm{u})$ denote the j$^{\text{th}}$ derivative of $\mathbf{C}_\mathrm{i}$, then $\mathbf{C}(u)$ is said to be C^k continuous at the breakpoint u_i if $\mathbf{C}_\mathrm{i}{}^{(j)}(\mathrm{u}_i) = \mathbf{C}_{\mathrm{i}+1}{}^{(j)}(\mathrm{u}_i)$ for all $0 \leq j \leq k$.

Any of the standard polynomial forms can be used to represent each segment $\mathbf{C}_\mathrm{i}(\mathrm{u})$.Figure 7.2 shows the curve of Figure 7.1 with the three segments in cubic Bézier form. $\boldsymbol{P}_i^j$ denotes the i$^{\text{th}}$ control point of the j$^{\text{th}}$ segment.

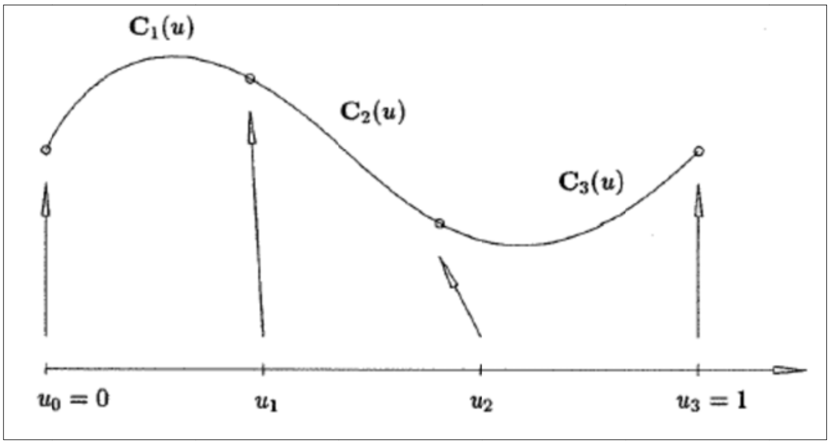


Figure 7.1 A piecewise cubic polynomial curve with three segments

If the degree equals three and the breakpoints $U = \{u_0, u_1, u_2, u_3\}$ remain fixed, and if we allow the twelve control points, $\boldsymbol{P}_i^j$, to vary arbitrarily, we obtain the vector space, V, consisting of all piecewise cubic polynomial curves on U. V has dimension twelve, and a curve in V may be discontinuous at u_1 or u_2. Now suppose we specify (as in Figure 7.2) that $\boldsymbol{P}_3^1 = \boldsymbol{P}_0^2$ and $\boldsymbol{P}_3^2 = \boldsymbol{P}_0^3$. This gives rise to V^0, the vector space of all piecewise cubic polynomial curves on U which are at least C^0 continuous everywhere. V^0 has dimension ten, and $V^0 \subset V$.

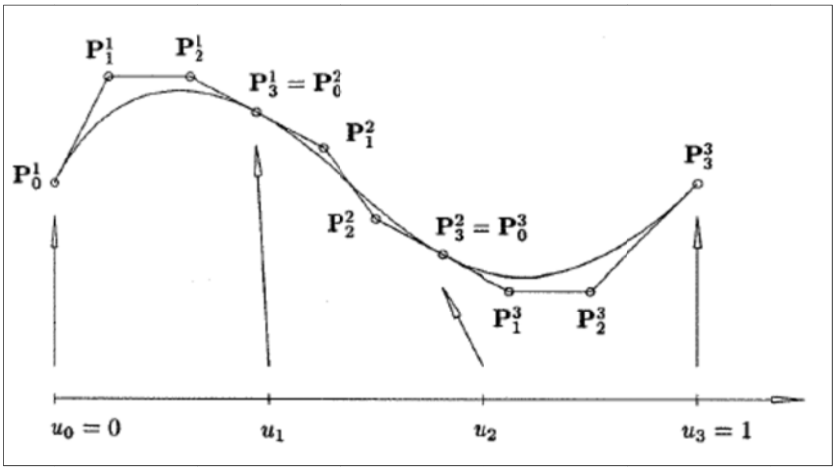


Figure 7.2 The curve of Figure 7.1 shown with the polynomial segments represented in Bézier form

Imposing C^1 continuity is a bit more involved. Let us consider $u = u_1$. Assume that $\boldsymbol{P}_3^1 = \boldsymbol{P}_0^2$. Let

$$v = \frac{u - u_0}{u_1 - u_0} \text{ and } w = \frac{u - u_1}{u_2 - u_1}$$

be local parameters on the intervals $[u_0, u_1]$ and $[u_1, u_2]$, respectively. Then $0 \le v, w \le 1$. C^1 continuity at u_1 implies

$$\frac{1}{u_1 - u_0} C_1^{(1)}(v = 1) = C_1^{(1)}(u_1) = C_2^{(1)}(u_1) = \frac{1}{u_2 - u_1} C_2^{(1)}(w = 0)$$

And from Eq. 6.20 it follows that

$$\frac{3}{u_1 - u_0}(\boldsymbol{P}_3^1 - \boldsymbol{P}_2^1) = \frac{3}{u_2 - u_1}(\boldsymbol{P}_1^2 - \boldsymbol{P}_0^2)$$

Thus

Eq. 7.1

$$\boldsymbol{P}_3^1 = \frac{(u_2 - u_1)\boldsymbol{P}_2^1 + (u_1 - u_0)\boldsymbol{P}_1^2}{u_2 - u_0}$$

Eq. 7.1 is interpreted as that $\boldsymbol{P}_3^1$ and $\boldsymbol{P}_3^2$ can be written in terms of $\boldsymbol{P}_2^1, \boldsymbol{P}_1^2$ and $\boldsymbol{P}_2^2, \boldsymbol{P}_1^3$, respectively. Hence, V^1, the vector space of all C^1 continuous piecewise cubic polynomial curves on U, has dimension eight, and $V^1 \subset V^0 \subset V$.

This makes it clear that storing and manipulating the individual polynomial segments of a piecewise polynomial curve is not the ideal method for handling such curves. First, redundant data must be stored: twelve coefficients, where only eight are required for C^1 continuous cubic curves, and only six for C^2 continuous cubic curves. Second, for the Bézier form the continuity of $C(u)$ depends on the positions of the control points, hence there is little flexibility in positioning control points while maintaining continuity. If a designer wants C^1 continuity and is satisfied with the segments $C_1(u)$ and $C_3(u)$, but wants to modify the shape of $C_2(u)$, he is out of luck: none of $C_2(u)'s$ control points can be modified. Third, determining the continuity of a curve requires computation (such as Eq. 7.1).

We want a curve representation of the form

Eq. 7.2

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} f_i(u)\boldsymbol{P}_i$$

Where the $\boldsymbol{P}_i$ are ***control points,*** and the $\{f_i(u), i = 0, \dots, n\}$ are ***piecewise polynomial functions*** forming a basis for the vector space of all piecewise polynomial functions of the desired degree and continuity (for a fixed breakpoint sequence, $U = \{u_i\}, 0 \leq i \leq m$). Note that continuity is determined by the basis functions, hence the control points can be modified without altering the curve's continuity. Furthermore, the $\{f_i\}$ should have the "usual" nice analytic properties. This ensures that the curves defined by Eq. 7.2 have nice geometric properties similar to Bézier curves, e.g., convex hull, variation diminishing, transformation invariance. Another important property that we seek in our basis functions is that of ***local support;*** this implies that each $f_i(u)$ is nonzero only on a limited number of subintervals, not the entire domain $[u_0, u_m]$. Since $\boldsymbol{P}_i$ is multiplied by $f_i(u)$, moving $\boldsymbol{P}_i$ affects curve shape only on the subintervals where $f_i(u)$ is nonzero.

In the following paragraph, we use the Cox-de-Boor recursion formula in order to define the piecewise polynomial functions $f_i(u)$; this is the point of view adopted in (Les Piegel, Wayne Tiller, 1997). Nonetheless, there is another point of view on how a B-spline basis function is constructed. This methodology is based on bonding polynomial curve segments and is described in (Richard H. Bartels, John C. Beatty, Brian A. Barsky, 1987). We include this theory at the end of this part, in the Appendix A (Pg.109).

7.1 B - SPLINE CURVE DEFINITION

Letting ***C(u)*** be the position vector along the curve as a function of the parameter u, a B-spline curve is given by

Eq. 7.3

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{P_i} \qquad a \le u \le b$$

where the $\{\boldsymbol{P_i}\}$ are the position vectors of the n + 1 control polygon vertices, control points, and the pth-degree B-spline basis functions defined on the nonperiodic (and nonuniform) knot vector

$$U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}$$

(m+1 knots). Unless it is stated otherwise, we assume that $a = 0$ and $b = 1$. The polygon formed by the $\{\boldsymbol{P_i}\}$ is called the ***control polygon***.

$N_{i,p}(u)$ is the ith B-spline basis function of p-degree (order p+1) and is defined by the Cox-de-Boor recursion formula

Eq. 7.4

$$N_{i,0}(u) = \begin{cases} 1 & if\ u_i \le u \le u_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Note that

- $\boldsymbol{C}(u)$ is a piecewise polynomial curve (since $N_{i,p}(u)$ are piecewise polynomials); the degree p, the number of control points, n+1, and number of knots, m+1, are related by $m = n + p + 1$;
- the half-open interval, $[u_i, u_{i+1})$, is called the ith knot span; it can have zero length, since knots need not be distinct;
- for p>0, $N_{i,p}(u)$ is a linear combination of two (p-1)-degree basis functions;

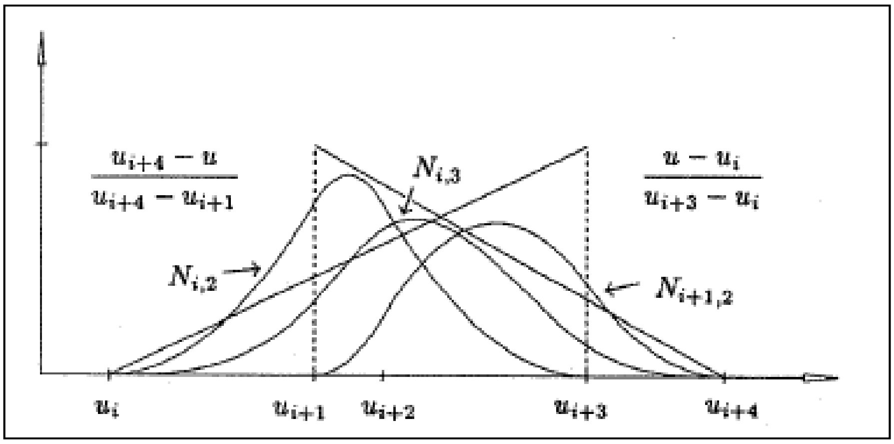


Figure 7.3 The recursive definition of B-spline basis functions

- computation of a set of basis functions requires specification of a knot vector, U, and the degree p;
- Eq. 7.4 can yield the quotient $\frac{0}{0}$; we define this quotient to be zero;
- the $N_{i,p}(u)$ are piecewise polynomials, defined on the entire real line; generally only the interval $[u_0, u_m]$ is of interest;
- the computation of the pth-degree functions generates a truncated triangular table

$$\begin{array}{cccc}
N_{0,0} & & & \\
 & N_{0,1} & & \\
N_{1,0} & & N_{0,2} & \\
 & N_{1,1} & & N_{0,3} \\
N_{2,0} & & N_{1,2} & \\
 & N_{2,1} & & N_{1,3} \\
N_{3,0} & & N_{2,2} & \vdots \\
 & N_{3,1} & \vdots & \\
N_{4,0} & \vdots & & \\
\vdots & & &
\end{array}$$

We now list a number of important properties of the B-spline basis functions. It is these properties which determine the many desirable geometric characteristics in B-spline curves and surfaces. Assume degree p and a knot vector $U = \{u_0, .., u_m\}$.

P2.1 $N_{i,p}(u) = 0$ if u is outside the interval $[u_i, u_{i+p+1})$ (local support property). This is illustrated by the trianglular scheme shown here.

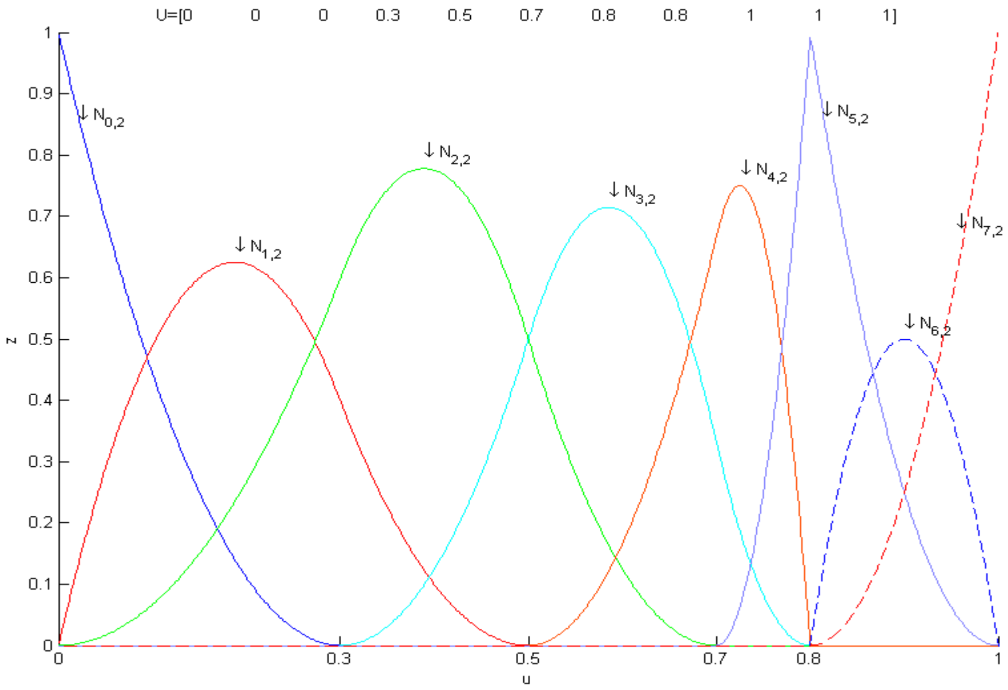


Figure 7.4 The nonzero second degree basis functions, U={0,0,0,0.3,0.5,0.7,0.8,0.8,1,1,1}

The local support of the basis functions yields the local control property of the B-Spline geometries. This property is demonstrated in Figure 7.5, where a curve defined by the basis shown above and the control points P=[0 0;3 2;4 5;10 8;15 5;20 -2;10 -5;10 1] is pertubated by moving a single control point to the position P_1=[6 0]. The P_1 control point is conjugate to the $N_{1,2}$ basis function, which is extended on the knot spans [u_2, u_3) and [u_3, u_4), thus there are affected only these two knot spans as is shown in figure below.

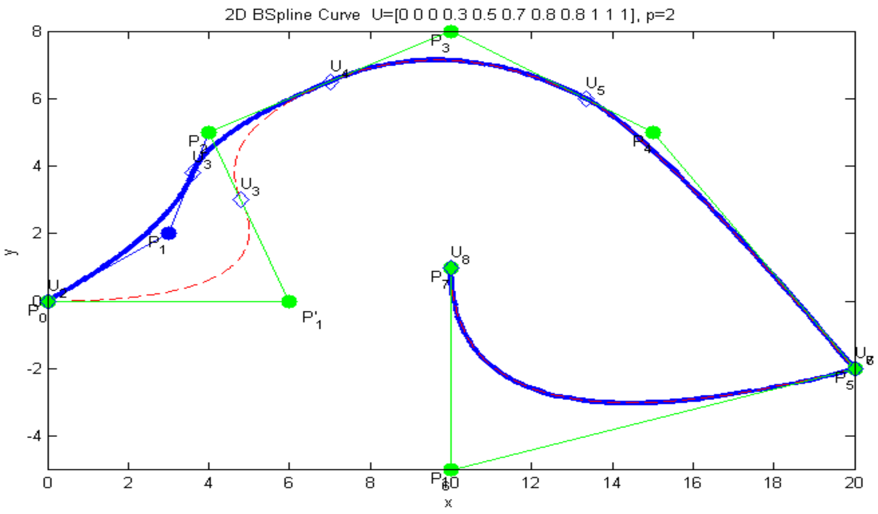


Figure 7.5 The local control property is illustrated by moving the control point P_1

Notice that $N_{1,3}$ is a combination of $N_{1,0}, N_{2,0}, N_{3,0}\ and\ N_{4,0}$. Thus, $N_{1,3}$ is nonzero only for $u = [u_1, u_5)$

$$\begin{tikzcd}
N_{1,0} & & N_{0,2} & \\
 & & & \\
 & N_{1,1} \arrow[uul] \arrow[ddl] & & \\
 & & & N_{0,3} \\
N_{2,0} & & N_{1,2} \arrow[uul] \arrow[ddl] & \\
 & & & \\
 & N_{2,1} \arrow[uul] \arrow[ddl] & & N_{1,3} \arrow[uul] \arrow[ddl] \\
 & & & \\
N_{3,0} & & N_{2,2} \arrow[uul] \arrow[ddl] & \\
 & & & \\
 & N_{3,1} \arrow[uul] \arrow[ddl] & & N_{2,3} \\
 & & & \\
N_{4,0} & & N_{3,2} &
\end{tikzcd}$$

P2.2 In any given knot span, $[u_j, u_{j+1})$, at most p+1 of the $N_{i,p}$ are nonzero, namely the functions $N_{j-p,p}, \dots, N_{j,p}$. On $[u_3, u_4)$ the only nonzero zeroth-degree function is $N_{3,0}$. Hence, the only cubic functions not zero on $[u_3, u_4)$ are $N_{0,3}, \dots, N_{3,3}$. This property is illustrated here

$$\begin{tikzcd}
 & N_{1,1} & & \\
 & & & N_{0,3} \\
N_{2,0} & & N_{1,2} \arrow[ur] \arrow[ddr] & \\
 & & & \\
 & N_{2,1} \arrow[uur] \arrow[ddr] & & N_{1,3} \\
 & & & \\
N_{3,0} \arrow[uur] \arrow[ddr] & & N_{2,2} \arrow[uur] \arrow[ddr] & \\
 & & & \\
 & N_{3,1} \arrow[uur] \arrow[ddr] & & N_{2,3} \\
 & & & \\
N_{4,0} & & N_{3,2} \arrow[uur] \arrow[ddr] & \\
 & & & \\
 & N_{4,1} & & N_{3,3}
\end{tikzcd}$$

P2.3 $N_{i,p}(u) \geq 0$ for all i, p and u(nonnegativity). This is proven by induction on p. It is clearly true for $p = 0$; assume it is true for $p - 1, p \geq 0$, with i and u arbitrary. By definition

Eq. 7.5

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

By P2.1, $N_{i,p-1}(u) = 0$ if $u \notin [u_i, u_{i+p})$. But $u \in [u_i, u_{i+p})$ implies $\frac{u-u_i}{u_{i+p}-u_i}$ is nonnegative. The same is true for the second term, and hence the $N_{i,p}(u)$ are nonnegative;

P2.4 For an arbitrary knot span $[u_i, u_{i+1})$,

Eq. 7.6

$\sum_{j=i-p}^{i} N_{j,p}(u) = 1$ for all $u \in [u_i, u_{i+1})$ (partition of unity).

To prove this, consider

$$\sum_{j=i-p}^{i} N_{j,p}(u) = \sum_{j=i-p}^{i} \frac{u-u_j}{u_{j+p}-u_j} N_{j,p-1}(u) + \sum_{j=i-p}^{i} \frac{u_{j+p+1}-u}{u_{j+p+1}-u_{j+1}} N_{j+1,p-1}(u)$$

Changing the summation variable in the second sum from $i-p$ to $i-p+1$, and considering that $N_{i-p,p-1}(u) = N_{i+1,p-1}(u) = 0$, we have

$$\sum_{j=i-p}^{i} N_{j,p}(u) = \sum_{j=i-p+1}^{i} \left[\frac{u-u_j}{u_{j+p}-u_j} + \frac{u_{j+p}-u}{u_{j+p}-u_j}\right] N_{j,p-1}(u)$$

$$= \sum_{j=i-p+1}^{i} N_{j,p-1}(u)$$

Applying the same concept recursively yields

$$\sum_{j=i-p}^{i} N_{j,p}(u) = \sum_{j=i-p+1}^{i} N_{j,p-1}(u) = \sum_{j=i-p+2}^{i} N_{j,p-2}(u)$$

$$= \cdots = \sum_{j=i}^{i} N_{j,0}(u) = 1$$

P2.5 All derivatives of $N_{i,p}(u)$ exist in the interior of a knot span (while it is a polynomial). At a knot $N_{i,p}(u)$ is $p-k$ times continuously differentiable, where k is the multiplicity of the knot. Hence, increasing degree increases continuity, and increasing knot multiplicity decreases continuity;

P2.6 Except for the case $p = 0$, $N_{i,p}(u)$ attains exactly one maximum value.

It is important to understand the effect of multiple knots. Consider the functions $N_{0,2}, N_{1,2}, N_{2,2}, N_{5,2}$ and $N_{6,2}$. Recalling that $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, from Eq. 7.4

and P2.1, we see that these functions are computed on the following knot spans and are zero outside these spans

$$N_{0,2} : \{0,0,0,1\}$$

$$N_{1,2} : \{0,0,1,2\}$$

$$N_{2,2} : \{0,1,2,3\}$$

$$N_{5,2} : \{3,4,4,5\}$$

$$N_{6,2} : \{4,4,5,5\}$$

Now the word ‘multiplicity’ is understood in two different ways:

- the multiplicity of a knot in the knot vector;
- the multiplicity of a knot with respect to a specific basis function.

For example, $u = 0$ has multiplicity three in the previous knot vector U. But with respect to the functions $N_{0,2}, N_{1,2}, N_{2,2}$ and $N_{5,2}$, $u = 0$ is a knot of multiplicity 3,2,1 and 0, respectively. From P2.5, the continuity of these functions at $u = 0$ is $N_{0,2}$ discontinuous; $N_{1,2}$ C^0 continuous; $N_{2,2}$ C^1 continuous; $N_{5,2}$ totally unaffected ($N_{5,2}$ and all its derivatives are zero at $u = 0$, from both sides). $N_{1,2}$ ‘sees’ $u = 0$ as a double knot, hence it is C^0 continuous. $N_{2,2}$ ‘sees’ all its knots with multiplicity 1, thus it is C^1 continuous everywhere. Clearly, another effect of multiple knots (as seen by the functions) is to reduce the number of ‘apparent’ intervals on which a function is nonzero; e.g., $N_{6,2}$ is nonzero only on $u \in [u_4, u_5)$, and it is only C^0 continuous at $u = 4$ and $u = 5$.

P2.7 A knot vector of the form

$$U = \{0, \dots ,0,1, \dots ,1\},$$

where there are p+1 0s and 1s, yields the Bernstein polynomials of degree p

P2.8 Let $m + 1$ be the number of knots. Then there are $n + 1$ basis functions, where $n = m - p - 1$; $N_{0,p}(a) = 1$ and $N_{n,p}(b) = 1$. For example, $N_{0,p}(a) = 1$ folloes from the fact that $N_{0,0}, \dots , N_{p-1,0} = 0$, since this implies that $N_{0,p}(a) = N_{p,0}(a) = 1$ From P2.4 it follows that $N_{i,p}(a) = 0$ for $i \neq 0$, and $N_{i,p}(b) = 0$ for $i \neq n$

The derivative of a basis function is given by

Eq. 7.7

$$N'_{i,p} = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Now let $N_{i,p}^{(k)}$ denote the k^th derivative of $N_{i,p}(u)$. Repeated differentiation of Eq. 7.7 produces the general formula

Eq. 7.8

$$N_{i,p}^{(k)} = p\left(\frac{N_{i,p-1}^{(k-1)}}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p+1} - u_{i+1}}\right)$$

These derivative basis functions have the property of

$$\sum_{j=i-p}^{i} N_{j,p}^{(k)}(u) = 0 \text{ for all } u \in [u_i, u_{i+1})$$

That is the basis functions to generate vectors (gradients) out of points (see section 4).

Remarks on Eq. 7.8 :

- k should not exceed p (all higher derivatives are zero);
- the denominators involving knot differenced can become zero; the quotient is defined to be zero in this case.

Example of derivative basis functions are shown in figure below, corresponding to the basis functions of Figure 7.4

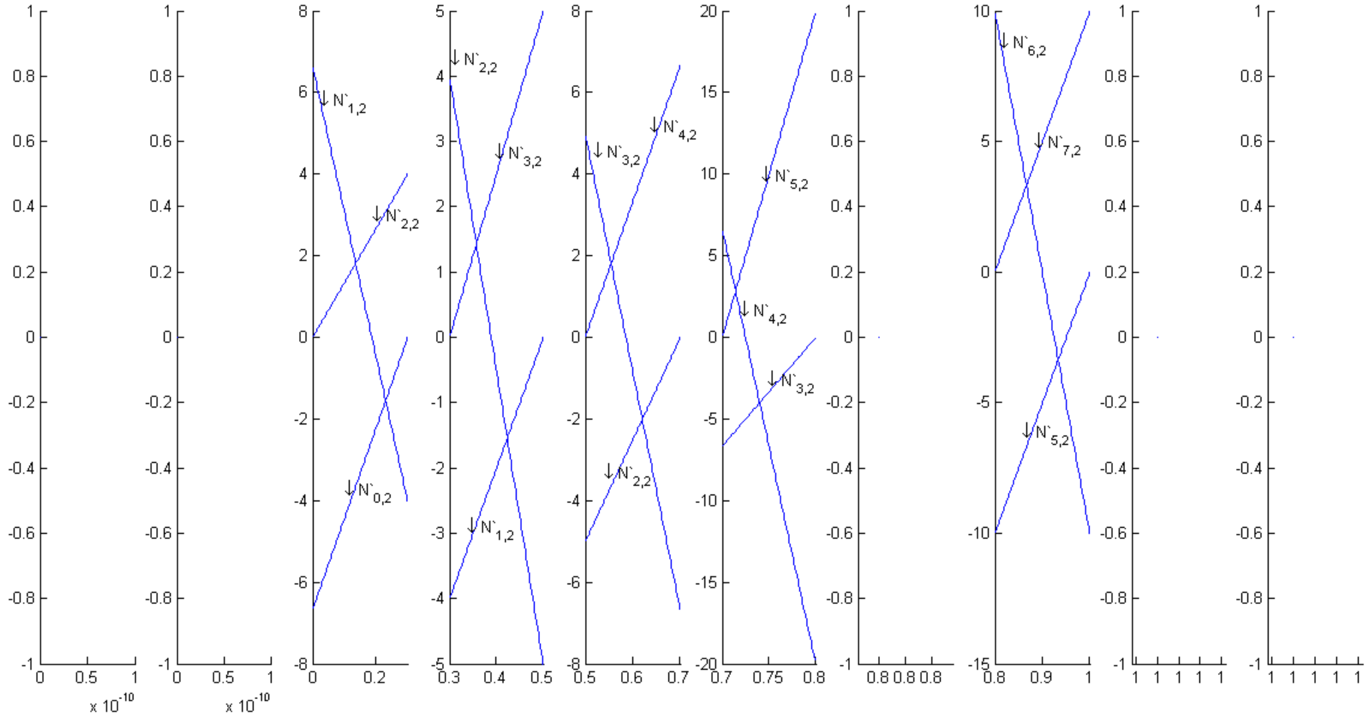


Figure 7.6 Derivative B-Spline basis functions are plotted in each knot span of the corresponding 2^nd degree basis of the Figure 7.4

In figure below a curve is presented with the basis of figure Figure 7.4 and its derivatives produced by the basis of the Figure 7.6. The control points that are used are P=[0 0;3 2;4 5;10 8;15 5;20 -2;10 -5;10 1]. In next figures the first derivative curve is plotted with its derivatives (2nd derivatives) and the graphs that illustrate the rate of changing of each coordinate with respect to the parametric space.

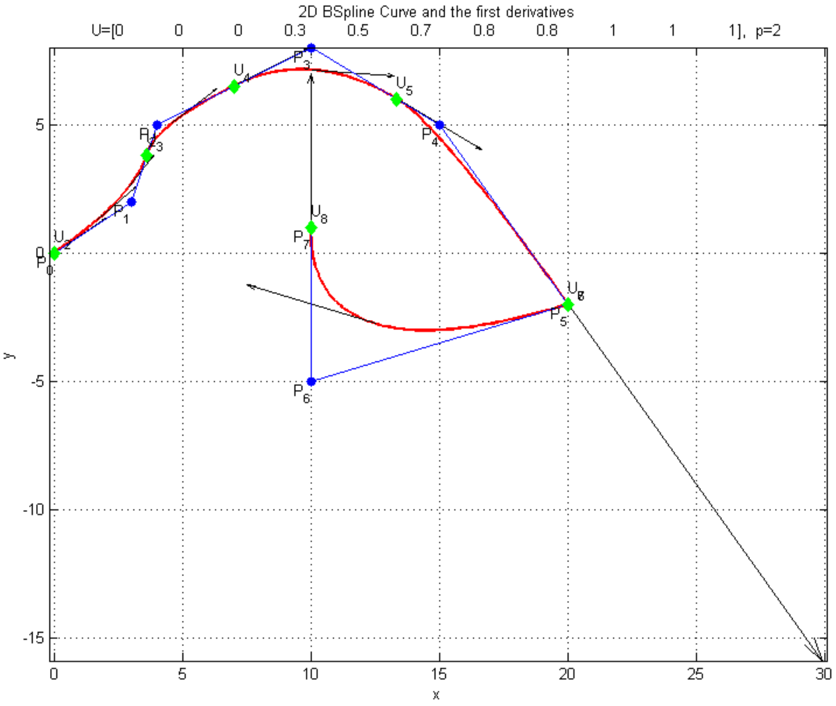


Figure 7.7 A 2D BSpline curve with its derivatives w.r.t. the parametric space in random points

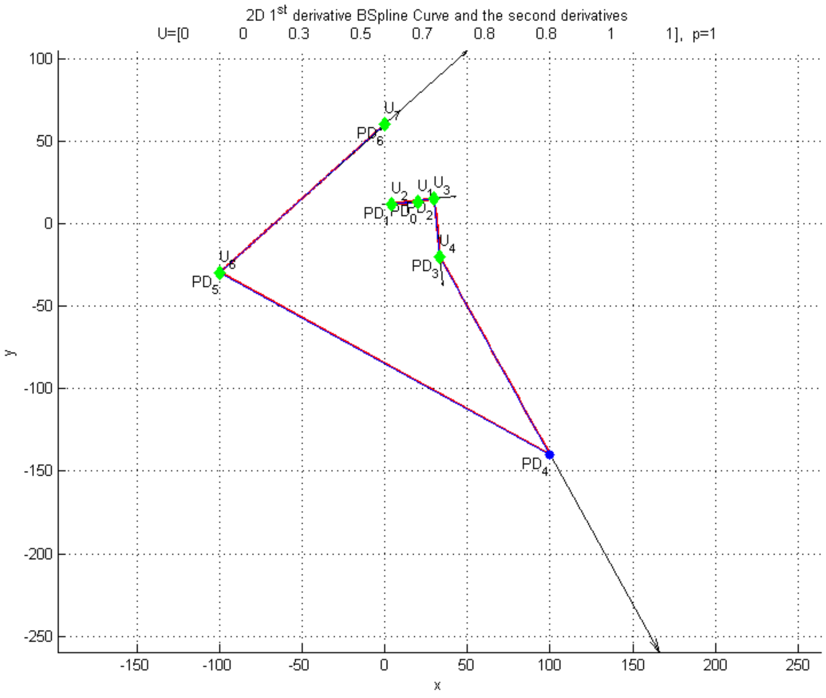


Figure 7.8 The derivative curve of the above figure and the 2nd derivatives

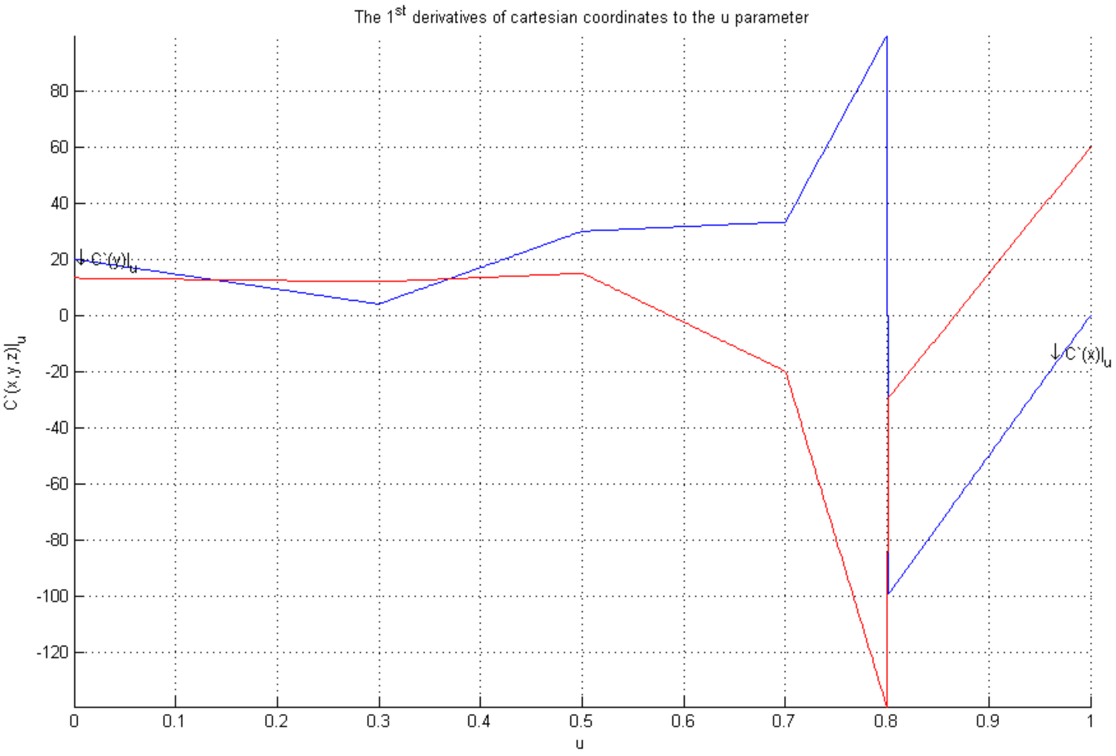


Figure 7.9 The rates of change of coordinates as traveling along the curve

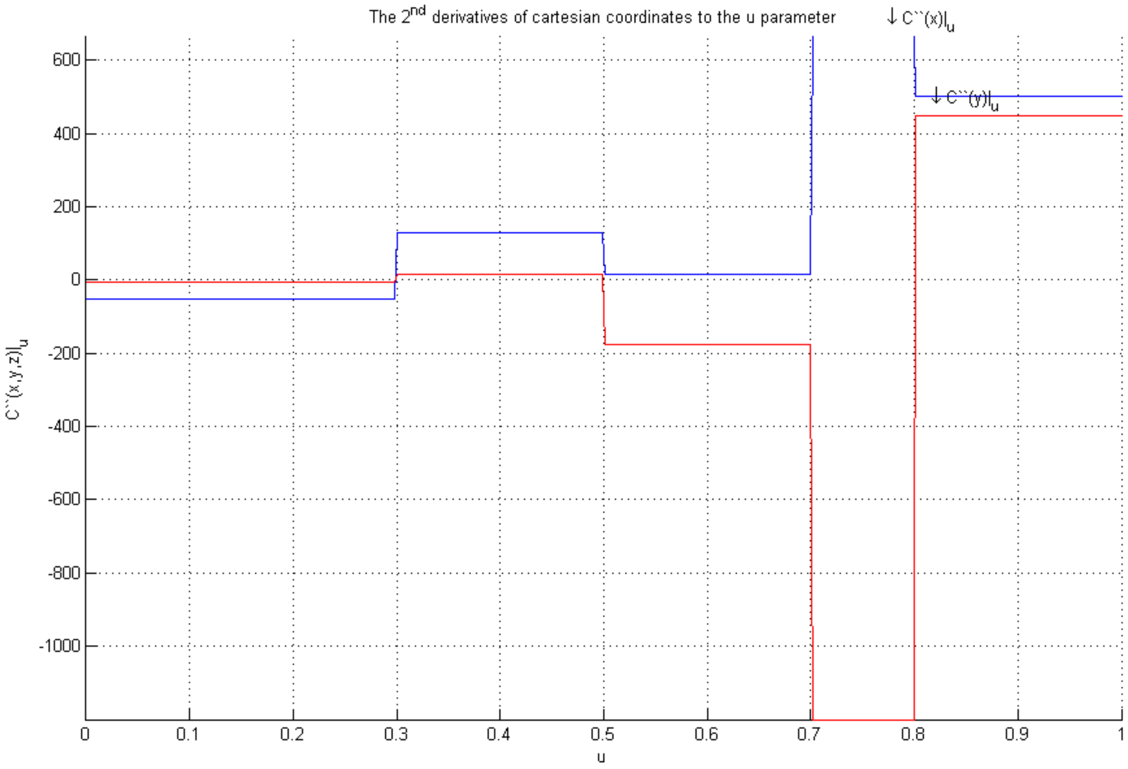


Figure 7.10 The second rates of change of coordinates as traveling along the curve

7.2 UNIFORM BICUBIC B-SPLINE SURFACES

We want to form surfaces as a scaled sum of basis functions, as in (Eq. 1.5), but now X, Y, and Z must be functions of two independent parameters:

$$\sum_i \boldsymbol{P}_i N_i(u,v)$$

and we need basis functions $N_i(u,v)$ that are piecewise in the two independent parameters u and v. These basis functions should be bicubic, nonzero only on a local parametric region, nonnegative, and should sum to one. The local-support regions of all the basis functions should cover the u,v-plane in some regular fashion, and the indexing scheme used for the control vertices $\boldsymbol{P}_i$ will have to reflect the geometric arrangement of this covering. Much research is currently appearing on general ways of constructing such basis functions, which are called ***multivariate B-splines***.

We will concentrate on a special, classical construction process for surfaces that is a natural and straightforward generalization of the uniform cubic B-spline curves. Recall that a spline curve is formed by piecing together successive curve segments; we will form a spline surface by piecing together rectangular surface patches to form a composite surface in much the same way that one constructs a patchwork quilt. We will specify the continuity with which these patches meet, just as we specify the continuity with which curve segments meet.

Our generalization works in the following way. A surface will be formed as a scaled sum of basis functions, using for scale factors the x -, y - and z -coordinates of a ***topologically rectangular array*** of control vertices, called the ***control mesh*** or ***control graph***, near which the surface is to pass. Since the control vertices are arranged in a rectangular topology, $\boldsymbol{P}_{i,j}$ the surface can be expressed by a double summation:

Eq. 7.9

$$\boldsymbol{S}(u,v) = \sum_i \sum_j \boldsymbol{P}_{i,j} N_{i,j}(u,v) = \sum_i \sum_j \left(x_{i,j} N_{i,j}(u,v), y_{i,j} N_{i,j}(u,v), z_{i,j} N_{i,j}(u,v) \right)$$

An easy way to cause the parametric region of locality of the basis functions to be rectangular is to let $N_{i,j}(u,v) = N_i(u) N_j(v)$, where $N_i(u)$ and $N_j(v)$ are simply the univariate cubic B-splines defined by (Eq. 1.4). In the mathematical literature the basis function $N_{i,j}(u,v)$ formed in this way is called a ***tensor product B-spline***. (Figure 7.11).

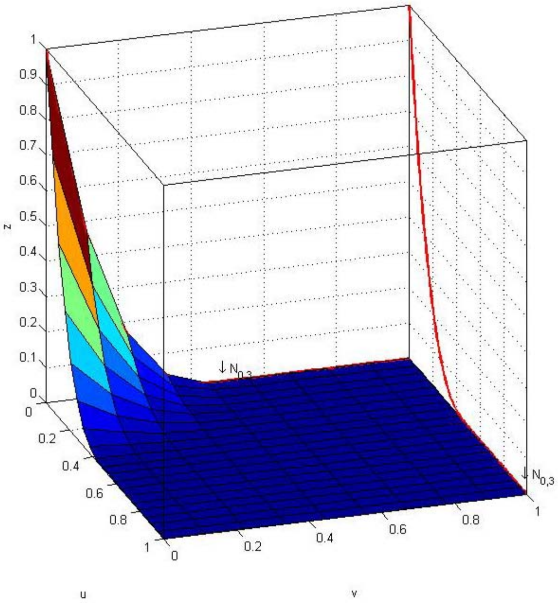


Figure 7.11 The $N_{0,0}(u, v)$ bivariate cubic B-spline basis function

There are some details that we need to discuss. We now have two knot sequences - one is a sequence of knot values for u and one is a sequence of knot values for v. Together they form a grid in parameter space. The bivariate B-spline $N_{0,0}(\mathrm{u}, \mathrm{v})$ is formed over the mesh of Figure 7.12 from the univariate B-splines $N_0(\mathrm{u})$ and $N_0(\mathrm{v})$. The former is nonzero over the range (u_0, u_4) and the latter is nonzero over the range (v_0, v_4) so $N_{0,0}(u, v)$ will be nonzero whenever $u_0 < u < u_4$ and $v_0 < v < v_4$ - that is, for the sixteen square ***parametric regions*** shown in Figure 7.12, and nowhere else.

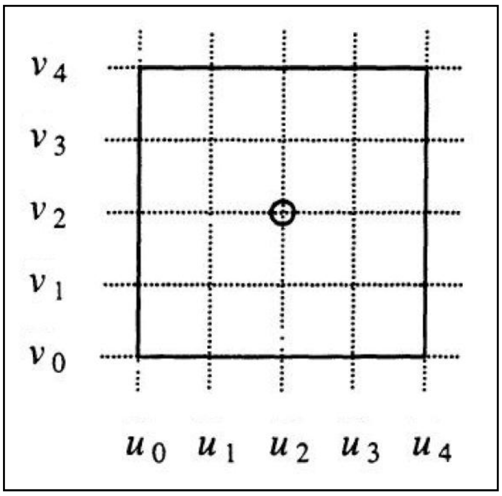


Figure 7.12 The parametric mesh used to construct the bivariate uniform bicubic B-spline $N_{0,0}(u, v)$.

Because $N_0(u)$ and $N_0(v)$ are piecewise functions of u and v respectively, $N_{0,0}(u, v)$ is a piecewise function of both. The parametric region with corners at (u_i, v_j) and

(u_{i+1}, v_{j+1}) (a "dotted square" in Figure 7.12) defines an area in which $N_{0,0}(u, v)$ is a pure bicubic polynomial. The polynomials for adjacent regions are distinct, but at a shared boundary they agree in position and have identical first and second partial derivatives - that is, they meet with C^2 continuity. This follows from the continuity properties of $N_0(u)$ and $N_0(v)$ and the fact that $N_{0,0}(u, v) = N_0(u)N_0(v)$.

We can define another B-spline by adding another value to one of the knot vectors. Suppose that we add a knot u_5. The resulting parametric grid is shown in Figure 7.13.

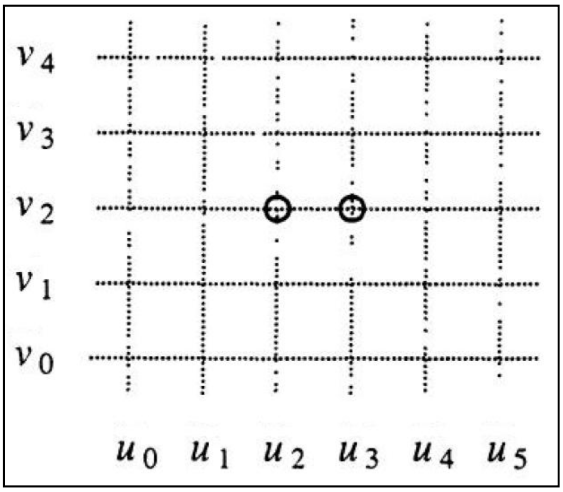


Figure 7.13 This grid allows us to define a second B-spline, centred at (u_3, v_2) and nonzero for $u_1 < u < u_5$ and $v_0 < v < v_4$

Adding an additional knot v_5 to the "vertical" knot sequence would then add two B-splines above the two of Figure 7.12, and so on. So long as we continue using uniform knot sequences, the $N_i(u)$ and $N_j(v)$ all have the same shape, and consequently the surface basis functions $N_{i,j}(u, v)$ all have the same shape. In effect we simply translate the one basis function so that it is centred over a variety of grid intersections.

Now suppose that we extend the knot sequences sufficiently, and ask ourselves how many of the $N_{i,j}(u, v)$ will be nonzero four or more regions distant from the ends of the knot sequences. Since each B-spline is nonzero over sixteen regions, sixteen of them will be nonzero on any such region.

Thus, exactly sixteen B-splines and their corresponding control vertices are required to define a single surface patch. For example, the parameter grid needed to define the sixteen B-splines that are nonzero when $u_3 < u < u_4$ and $v_3 < v < v_4$ is shown in Figure 7.14.

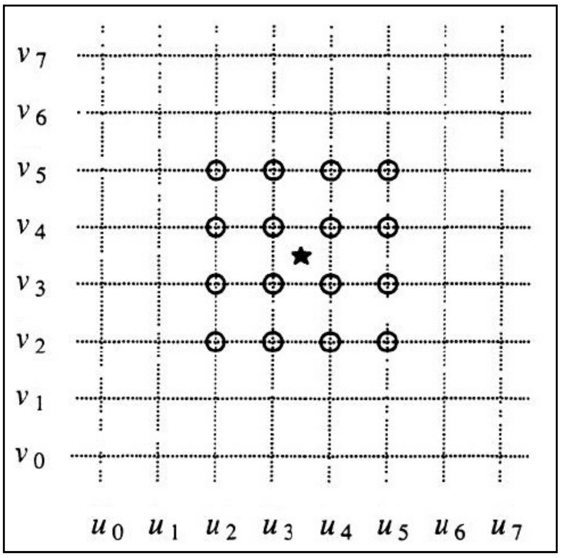


Figure 7.14 The parametric mesh used to construct the sixteen bivariate B-splines that are nonzero when $u_0 < u < u_4$ and $v_0 < v < v_4$. The peaks of these sixteen B-splines are circled. The parametric region in which all sixteen are nonzero is marked with a star.

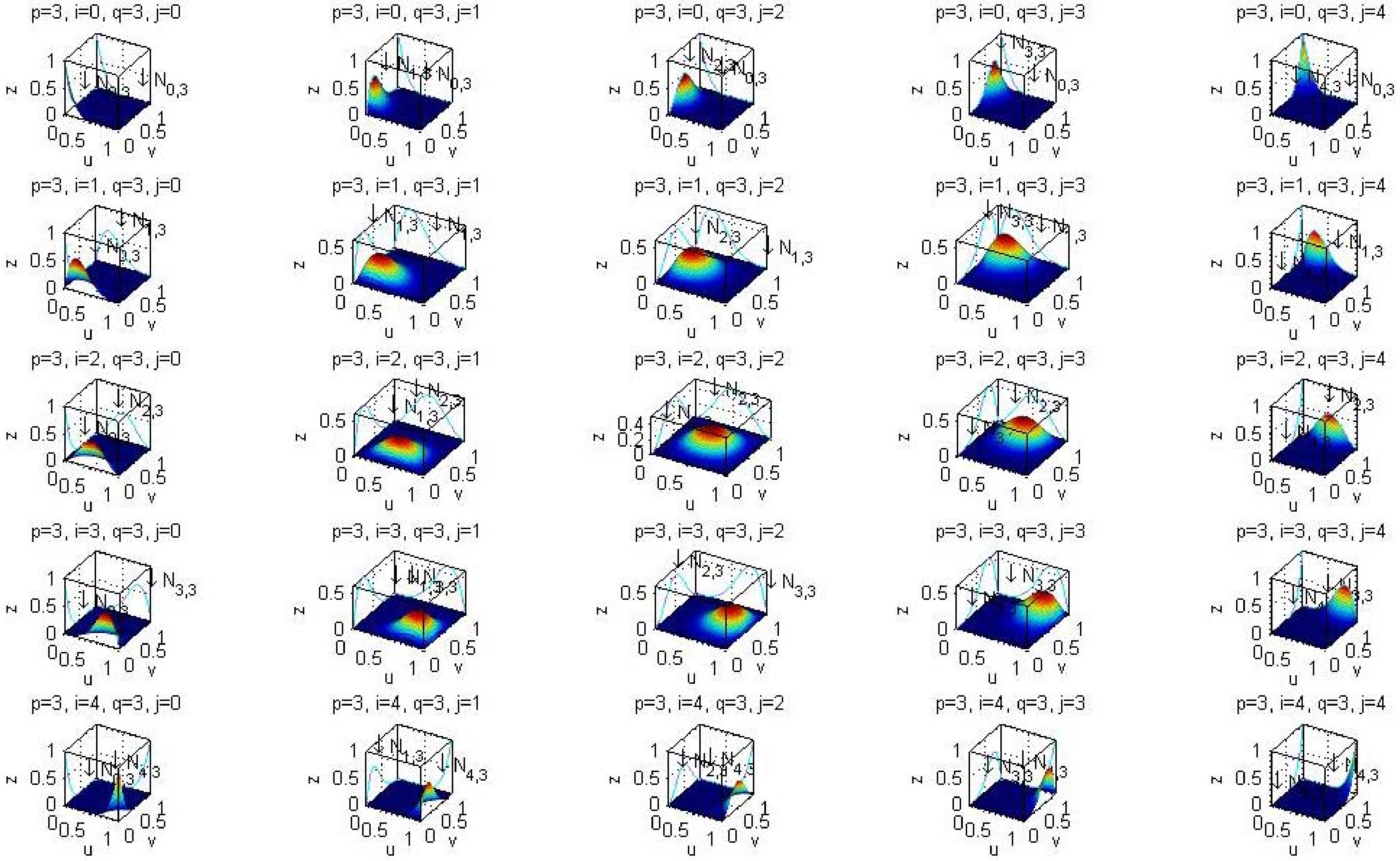


Figure 7.15 The bicubic B-spline Tensor Basis functions in a whole on the grid $[\mathbf{0\ 0\ 0\ 0\ 0.5\ 1\ 1\ 1\ 1}] \times [\mathbf{0\ 0\ 0\ 0\ 0.5\ 1\ 1\ 1\ 1}]$

In summary, then, since $N_i(u)$ and $N_j(v)$ are each nonzero only over four successive intervals, if $u_i < u < u_{i+1}$ and $v_j < v < v_{j+1}$, and if we adopt the convention that the portion of $\boldsymbol{C}(u,v)$ defined by this set of values for $u$and v is denoted by $\boldsymbol{C}_{i,j}(u,v)$, we can rewrite Eq. 7.9 as

Eq. 7.10

$$\boldsymbol{C}_{\boldsymbol{i},\boldsymbol{j}}(u,v) = \sum_{r=-3}^{0}\sum_{s=-3}^{0} \boldsymbol{P}_{i+r,j+s}\, N_{i+r}(u) N_{j+s}(v)$$

Or, in terms of $\bar{u}$ and $\bar{v}$,

$$\boldsymbol{C}_{\boldsymbol{i},\boldsymbol{j}}(\bar{u},\bar{v}) = \sum_{r=-3}^{0}\sum_{s=-3}^{0} \boldsymbol{P}_{i+r,j+s}\, N_{i+r}(\bar{u}) N_{j+s}(\bar{v})$$

This is said to be the ***tensor product*** of two univariate B-spline curve segments.

The basic properties of B-spline curves that we have discussed in Sections 1.3 through 1.5 carry over easily to B-spline surfaces.

The perspective transformation, on the other hand, does not preserve the shape of B-spline curves or surfaces. That is, the surface obtained by computing points on a surface and then applying the perspective transformation is not identical to the surface obtained by applying the perspective transformation to the control vertices and then computing points on the surface defined by the transformed control vertices. In fact, the perspective transformation of a cubic curve or surface is not necessarily expressible as a cubic. It is instead a rational polynomial, namely the quotient of two cubic polynomials, and it is easy to construct examples of numerators and denominators that are relatively prime.

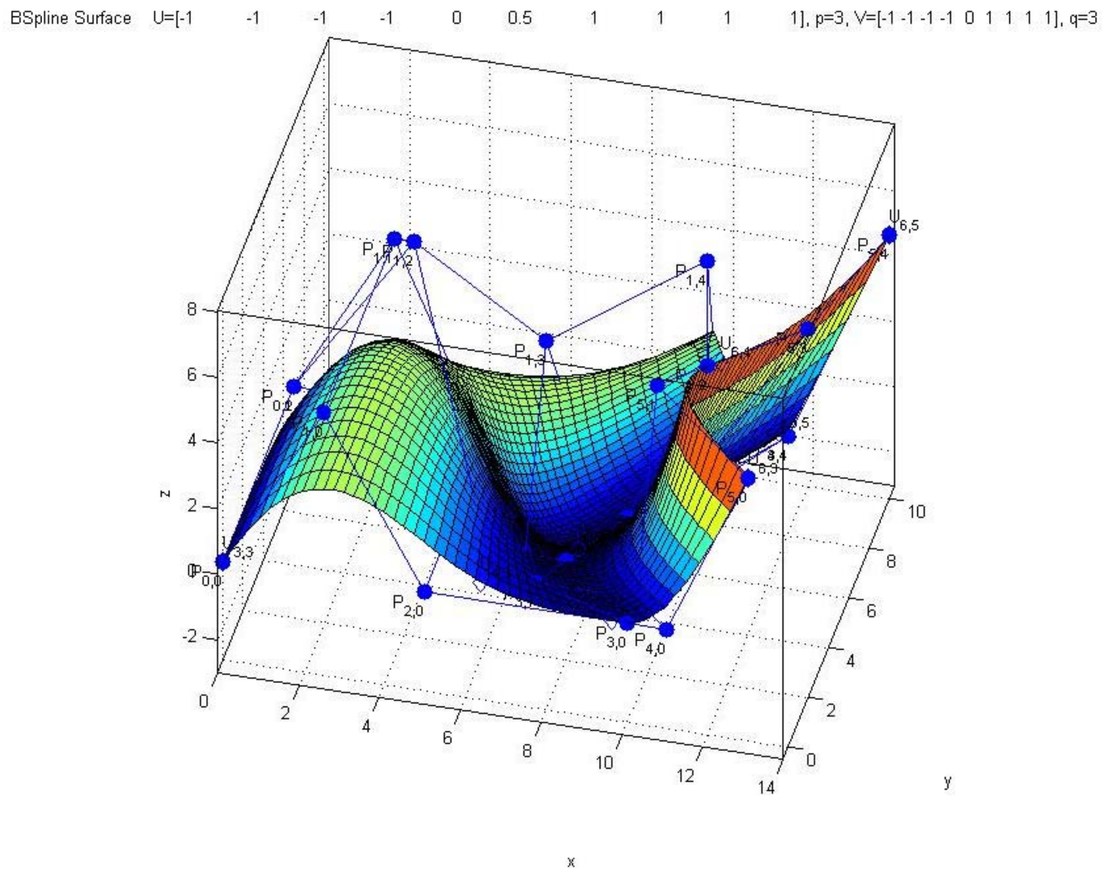


Figure 7.16 A B-Spline surface

7.3 B-SPLINE SOLIDS

Tensor product B-spline solids are defined in analogous fashion to B-spline surfaces. Given a ***control lattice*** $\{\boldsymbol{P}_{i,j,k}\}, i = 1,2, \dots, n, j = 1,2, \dots, m, k = 1,2, \dots, l$, polynomial orders p,q and r, and knot vectors $U = \{u_1, u_2, \dots, u_{n+p+1}\}, V = \{v_1, v_2, \dots, v_{m+q+1}\}$ and $\Omega = \{\omega_1, \omega_2, \dots, \omega_{l+r+1}\}$, a B-spline solid is defined by

$$\boldsymbol{S}(u, v, \omega) = \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{m} \sum_{k=1}^{l} N_{i,p}(u) N_{j,q}(v) N_{k,r}(\omega)\, \boldsymbol{P}_{i,j,k}$$

The properties of a B-spline solid are trivariate generalizations of those for B-spline surfaces.

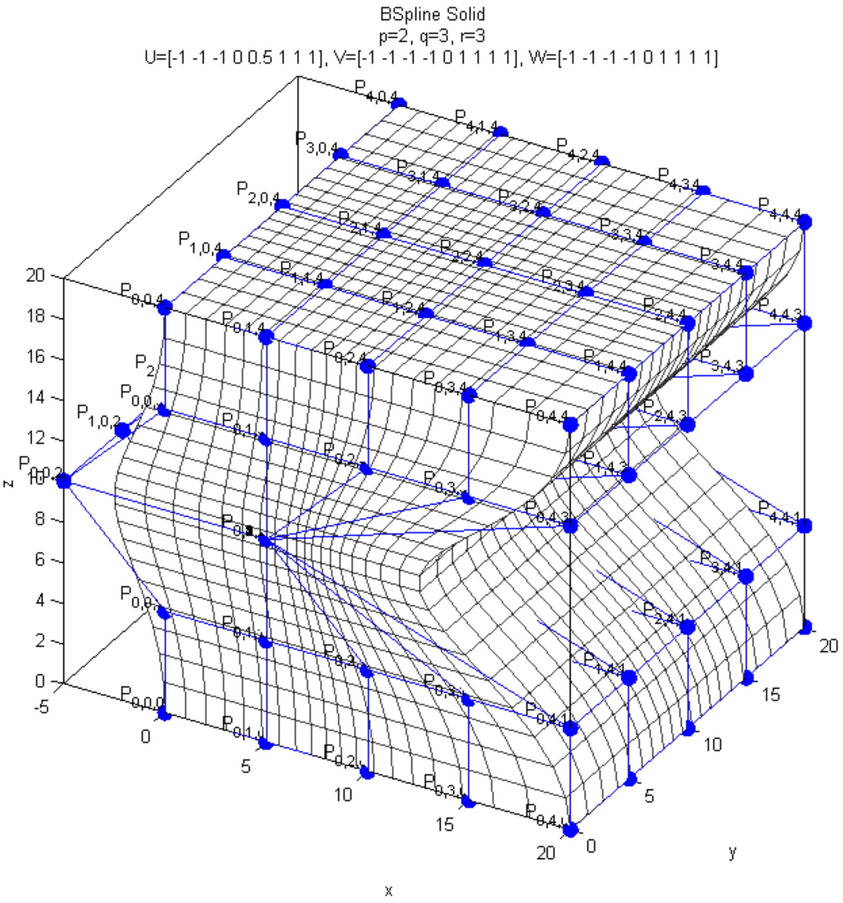


Figure 7.17 A tensor product solid based on B-splines of degrees 2,3 and 3 in x,y and z direction respectively

8 NURBS

B-spline curves and surfaces grew out of the pioneering work of Pierre Bézier in the early 1970s. Perhaps one can consider B-spline curves and surfaces the children of Bézier curves and surfaces, and non-uniform rational B-splines, or **NURBS**, the grandchildren.

The rational splines are (roughly speaking) a quotient of two splines. A key advantage of this form is the ability to represent conic curves and quadric surfaces, such as cylinders, spheres, and cones. Moreover, the free-form curves and surfaces we discuss are all a special case of the rational splines since the constant function I is a perfectly respectable spline, and can be used as the denominator in a rational representation. Another advantage is that the rational formulation is invariant under projective transformations, such as perspective projection.

NURBS have become the de facto industry standard for the representation, design, and data exchange of geometric information processed by computers. This enormous success behind NURBS is largely due to the fact that

- NURBS provide a unified mathematical basis for representing both analytic shapes, such as conic sections and quadric surfaces, as well as free-form entities, such as car bodies and ship hulls;
- designing with NURBS is intuitive; almost every tool and algorithm has an easy-to-understand geometric interpretation;
- NURBS algorithms are fast and numerically stable;
- NURBS curves and surfaces are invariant under common geometric transformations, such as translation, rotation, parallel and perspective projections;
- NURBS are generalizations of nonrational B-splines and rational and nonrational Bézier curves and surfaces.

8.1 THE DE CASTELJAU ALGORITHM (ROGERS, 2001)

A rational B-spline curve is the projection of a nonrational (polynomial)B-spline curve defined in four-dimensional (4D) homogeneous coordinate space back into three-dimensional (3D) physical space. Specifically

Eq. 8.1

$$\boldsymbol{C}(u) = \sum_{i=1}^{n+1} \boldsymbol{P}_i^w N_{i,k}(u)$$

where the $\boldsymbol{P}_i^w$s are the four-dimensional homogeneous control polygon vertices for the nonrational four-dimensional B-spline curve. $N_{i,k}(u)$ is the nonrational B-spline basis function previously given in Eq. 7.4.

Projecting back into three-dimensional space by dividing through by the homogeneous coordinate yields the rational B-spline curve

Eq. 8.2

$$\boldsymbol{C}(u) = \frac{\sum_{i=1}^{n+1} P_i w_i N_{i,k}(u)}{\sum_{i=1}^{n+1} w_i N_{i,k}(u)} = \sum_{i=1}^{n+1} P_i R_{i,k}(u)$$

where the P_is are the three-dimensional control polygon vertices for the rational B-spline curve and the

Eq. 8.3

$$R_{i,k}(u) = \frac{w_i N_{i,k}(u)}{\sum_{i=1}^{n+1} w_i N_{i,k}(u)}$$

are the rational B-spline basis functions. Here, $w_i \geq 0$ for all values of i.

In figures below one can observe the effect of a weight to the conjugate basis function and to the neighbour basis functions. The basis parameterization through the weights, affects the produced geometry in analogous way. For the physical effect of the weights see section 6.3.4.

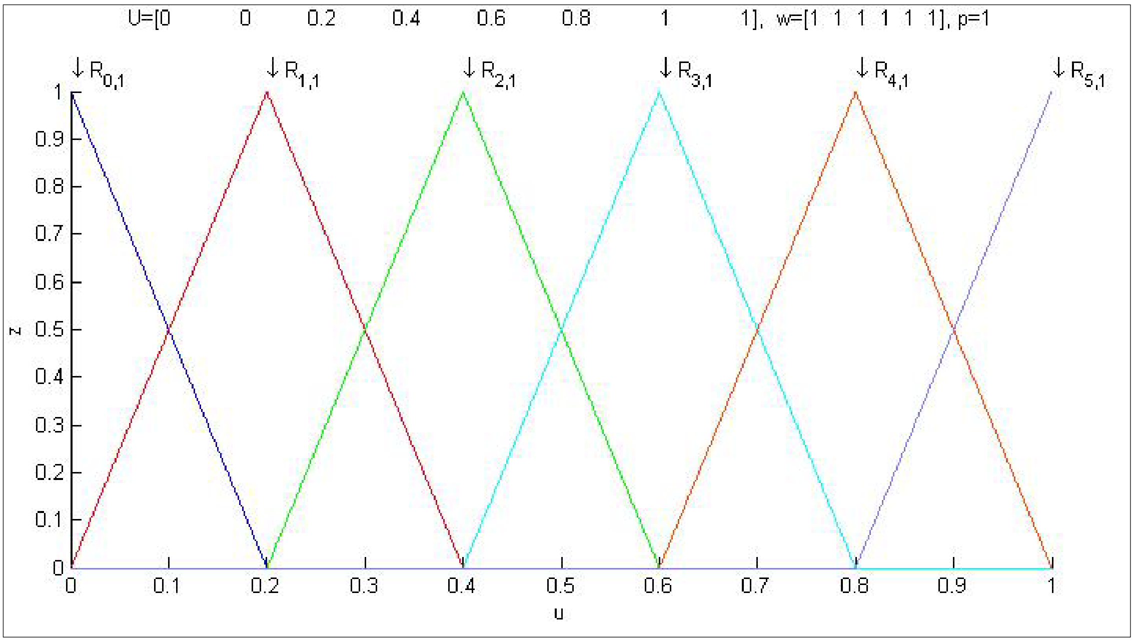


Figure 8.1 The nonrational B-spline basis functions of degree 1. Alternatively, rational B-spline basis functions with w=[1 1 1 1 1 1]

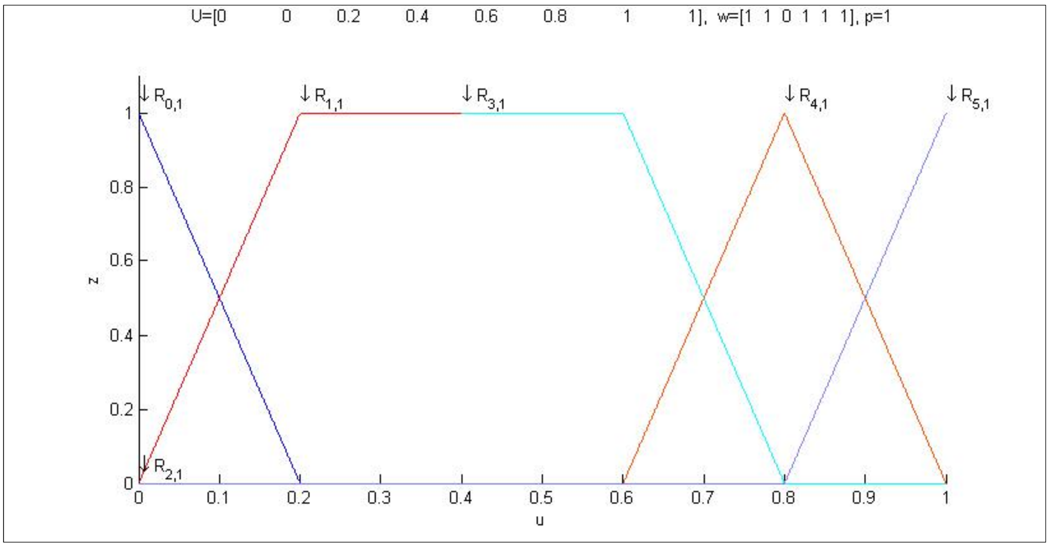


Figure 8.2 Rational B-spline basis functions as in Figure 8.1, but considering a w =[1 1 0 1 1 1]

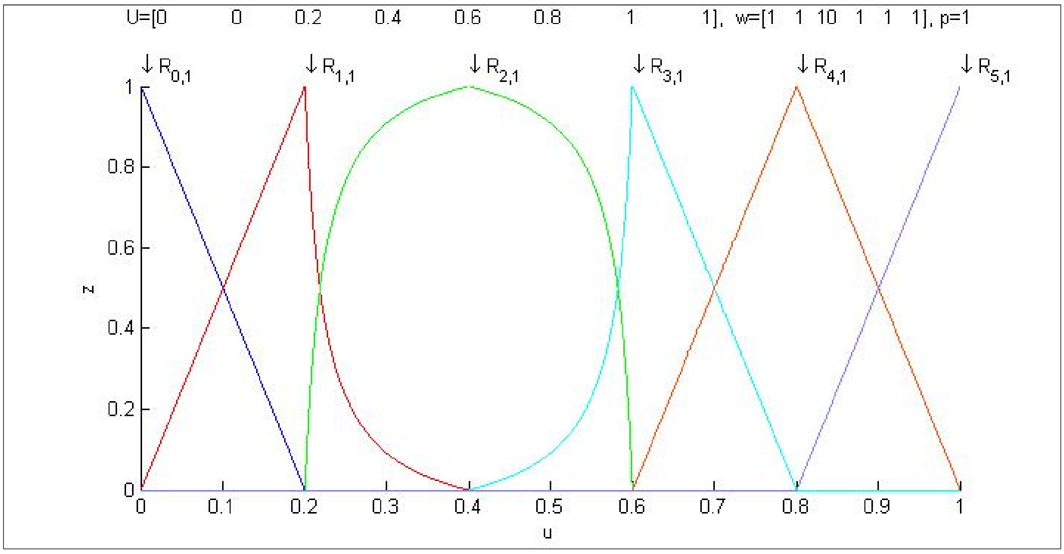


Figure 8.3 Rational B-spline basis functions as in Figure 8.1, but considering a w =[1 10 1 1 1]

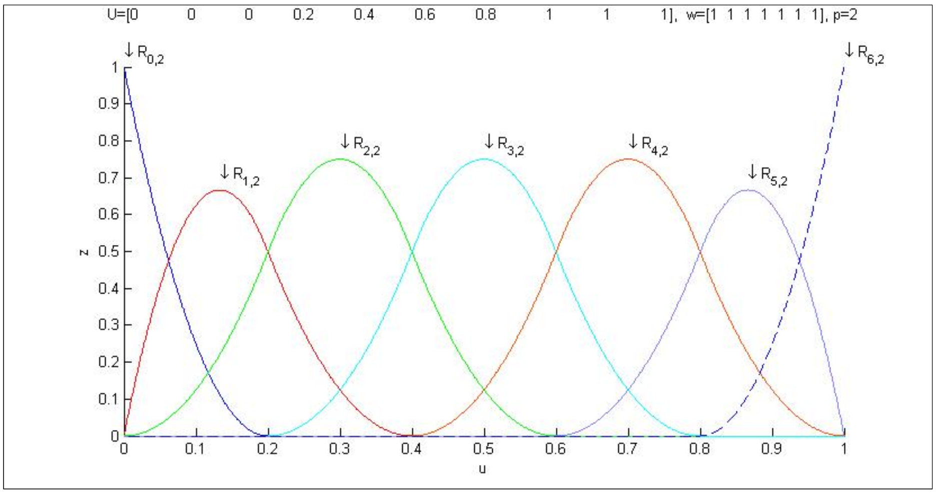


Figure 8.4 The nonrational B-spline basis functions of degree 2. Alternatively, rational B-spline basis functions with w=[1 1 1 1 1 1 1]

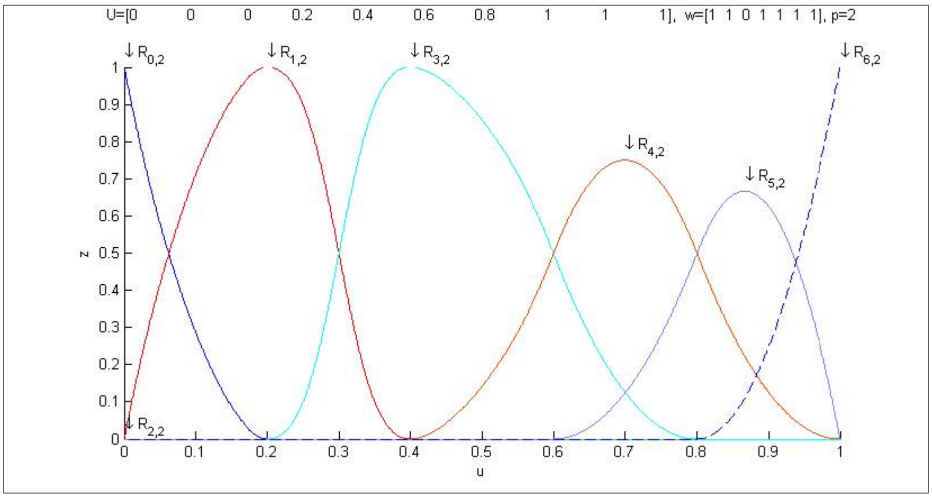


Figure 8.5 Rational B-spline basis functions as in Figure 8.4, but considering a w =[1 1 0 1 1 1 1]

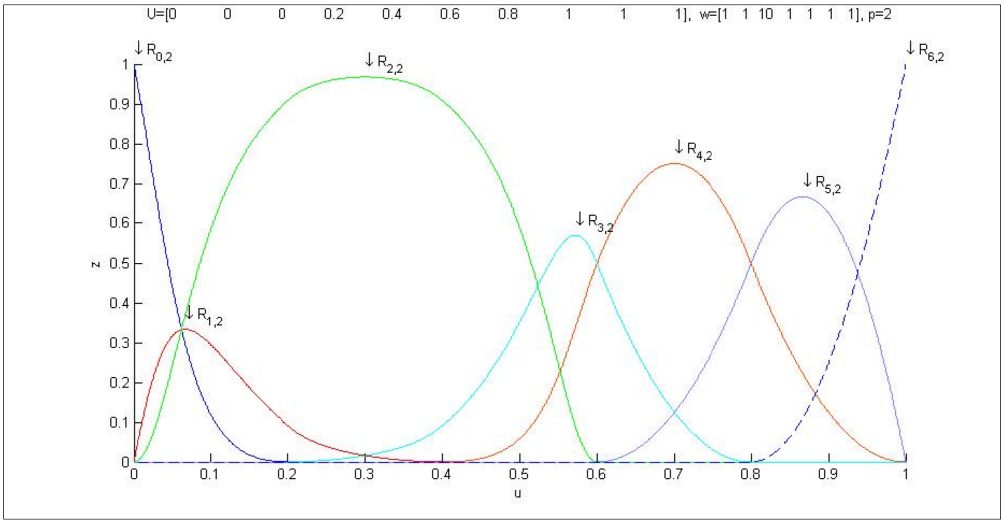


Figure 8.6 Rational B-spline basis functions as in Figure 8.4, after applying a w =[1 1 10 1 1 1 1]

8.2 CHARACTERISTICS OF NURBS

Rational B-spline basis functions and curves are a generalization of nonrational B-spline basis functions and curves. Thus, they carry forward nearly all the analytic and geometric characteristics of their nonrational B-spline counterparts. In particular :

- Each rational basis function is positive or zero for all parameter values, i.e., $R_{i,k} \geq 0$.
- The sum of the rational B-spline basis functions for any parameter value t is one, i.e. $\sum_{i=1}^{n+1} R_{i,k}(u) = 1$
- Except for first-order basis functions, i.e., k = 1, each rational basis function has precisely one maximum.
- A rational B-spline curve of order k (degree k-1) is C^{k-2} continuous everywhere.

- The maximum order of the rational B-spline curve is equal to the number of control polygon vertices.
- A rational B-spline curve exhibits the variation-diminishing property.
- A rational B-spline curve generally follows the shape of the control polygon.
- For $w_i > 0$, a rational B-spline curve lies within the union of convex hulls formed by k successive control polygon vertices.
- Any *projective* transformation is applied to a rational B-spline curve by applying it to the control polygon vertices; i.e., the curve is invariant with respect to a *projective* transformation. Note that this is a stronger condition than that for a nonrational B-spline, which is only invariant with respect to an *affine* transformation.

From Eq. 7.6 and Eq. 8.3, it is clear that when all $w_i = 1$, $R_{i,k}(u) = N_{i,k}(u)$.Thus, nonrational B-spline basis functions and curves are included as a special case of rational B-spline basis functions and curves. Furthermore, it is easy to show that an open rational B-spline curve with order equal to the number of control polygon vertices is a rational Bézier curve. For the case of all $w_i = 1$, the rational Bézier curve reduces to a nonrational Bézier curve. Thus, both rational and nonrational Bézier curves are included as special cases of rational B-spline curves.

8.3 DERIVATIVES OF NURBS CURVES (ROGERS, 2001)

The derivatives of rational B-spline curves are obtained by formal differentiation of Eq. 8.2 and Eq. 8.3. Specifically

Eq. 8.4

$$C'(t) = \sum_{i=1}^{n+1} P_i R'_{i,k}(t)$$

with

Eq. 8.5

$$R'_{i,k}(t) = \frac{w_i N'_{i,k}(t)}{\sum_{i=1}^{n+1} w_i N_{i,k}} - \frac{w_i N_{i,k} \sum_{i=1}^{n+1} w_i N'_{i,k}}{\left(\sum_{i=1}^{n+1} w_i N_{i,k}\right)^2}$$

Evaluating these results at t=0 and t=n-k+2 yields

Eq. 8.6

$$P'(0) = (k-1)\frac{w_2}{w_1}(B_2 - B_1)$$

Eq. 8.7

$$P'(n-k+2) = (k-1)\frac{w_n}{w_{n+1}}(B_{n+1}-B_n)$$

which shows that the direction of the slope is along the first and last polygon spans, respectively. Higher – order derivatives are obtained in a similar manner.

8.4 APPLICATION OF RATIONAL BÉZIER AND B-SPLINE CURVES – REPRESENTATION OF CONIC SECTIONS

As mentioned previously, rational Bézier and B-spline curves are used to represent all the conic sections. Furthermore, they provide a single mathematical description capable of blending the conic sections into free-form curves. Because the conic sections are described by quadratic equations, it is convenient to first consider a quadratic rational Bézier (n = 2) defined by three polygon vertices (n+1=3) , with knot vector U=[0 0 0 1 1 1]. Writing this out yields

Eq. 8.8

$$C(u) = \frac{w_0 N_{0,2}(u) P_0 + w_1 N_{1,2}(u) P_1 + w_2 N_{2,2}(u) P_2}{w_0 N_{0,2}(u) + w_1 N_{1,2}(u) + w_2 N_{2,2}(u)}$$

which, in fact, is a second degree rational Bézier curve (Figure 8.7).

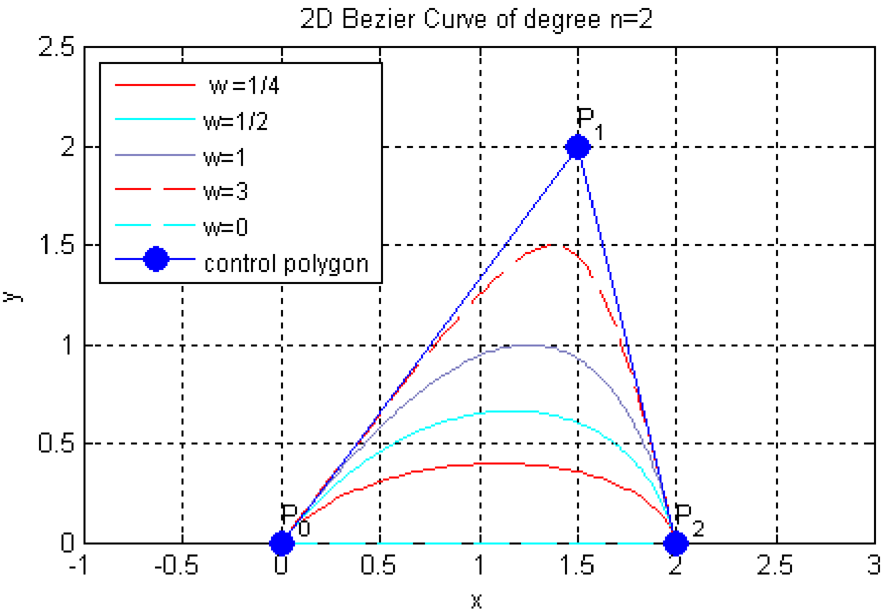


Figure 8.7 Conic sections defined by rational Bézier (or B-spline over one knot span) curves. The weight is imposed on the control point P_1. (a) Straight line, $w_1 = 0$; (b) elliptic, $w_1 = 1/4$ and $w_1 = 1/2$; (c) parabolic, $w_1 = 1$; (d) hyperbolic, $w_1 = 3$.

It is convenient to assume $w_0 = w_2 = 1$. Eq. 8.8 then reduces to

Eq. 8.9

$$C(u) = \frac{N_{0,2}(u) P_0 + w_1 N_{1,2}(u) P_1 + N_{2,2}(u) P_2}{N_{0,2}(u) + w_1 N_{1,2}(u) + N_{2,2}(u)}$$

Now, if $w_1 = 0$, a straight line between P_0 and P_2 results. If $w_1 \rightarrow \infty$, the control polygon is reproduced. When $w_1 = 0$ and $u = {}^1\!/_2$, the midpoint of the line P_0P_2, M (see Figure 8.12) is obtained. Similarly, when $w_1 \rightarrow \infty$, $u = {}^1\!/_2$ yields the polygon point at P_1. For $0 < w_1 < \infty$, the point S corresponding to the point at $u = {}^1\!/_2$ on the curve C(u) moves along the straight line connecting the midpoint M and P_1. S is called the shoulder point. The value of w_1 determines the type of conic section. It is shown that if

$w_1 = 0$	a straight line results
$0 < w_1 < 1$	an elliptic curve segment results
$w_1 = 1$	a parabolic curve segment results
$w_1 > 1$	a hyperbolic curve segment results

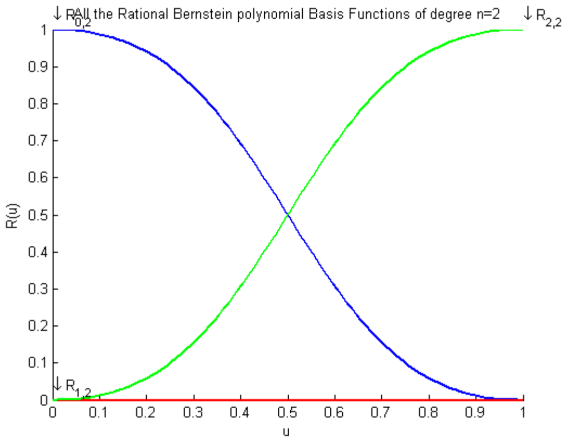


Figure 8.8 The Rational Bernstein polynomial Basis Functions used to construct the conic section Figure 8.7 - $w_1=0$

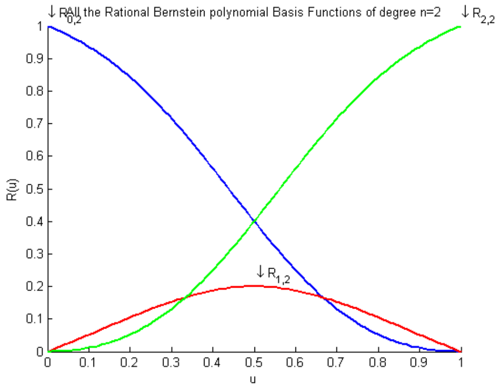


Figure 8.9 The Rational Bernstein polynomial Basis Functions used to construct the conic section Figure 8.7- $w_1=1/4$. As $R_{1,2}$ gets inflated, $R_{0,2}$ and $R_{2,2}$ are deflated

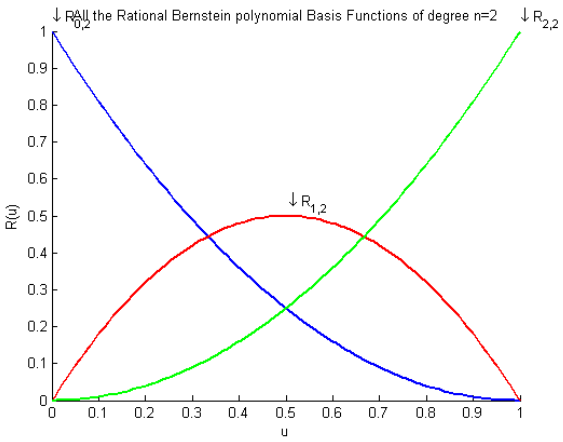


Figure 8.10 The Rational Bernstein polynomial Basis Functions used to construct the conic section Figure 8.7 - $w_1=1$

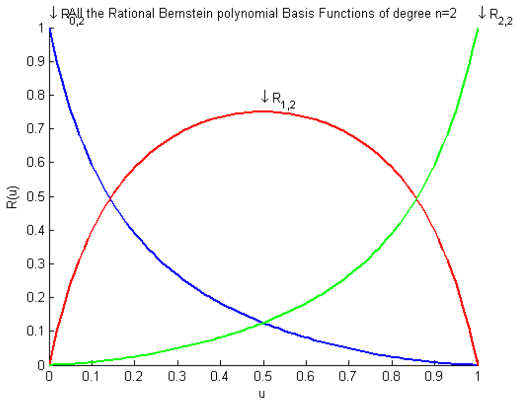


Figure 8.11 The Rational Bernstein polynomial Basis Functions used to construct the conic section Figure 8.7- $w_1=3$

Using the basis functions of Eq. 7.4 into Eq. 8.8 yields

$$C(u) = \frac{(1-u)^2 P_0 + 2w_1 u(1-u)P_1 + u^2 P_2}{(1-u)^2 + 2w_1 u(1-u) + u^2}$$

For u = 1/2, *C (u)* = S, which yields

Eq. 8.10

$$S = \frac{1}{1+w_1}\frac{P_0 + P_2}{2} + \frac{w_1}{1+w_1}P_1 = \frac{M}{1+w_1} + \frac{w_1}{1+w_1}P_1$$

Writing the parametric equation of the straight line between M and P_1 gives

Eq. 8.11

$$S(s) = (1-s)M + sP_1$$

where s is the parameter. Equating coefficients of Eq. 8.10 and Eq. 8.11 shows that

Eq. 8.12

$$s = \frac{w_1}{1+w_1} \quad \text{and} \quad \mathrm{w}_1 = \frac{s}{1-s} = \frac{M-S}{S-P_1}$$

The parameter s controls the shape of the curve and its conic form. Hence, it is a good design tool.

Because a circle is a special case of an ellipse, for a particular value of w_1 Eq. 8.9 yields a circular arc. Because of symmetry, P_0, P_1 and P_2 for a circular arc form an isosceles triangle, as shown in Figure 8.12. The required value of w_1 is determined from the geometry shown in Figure 8.12.

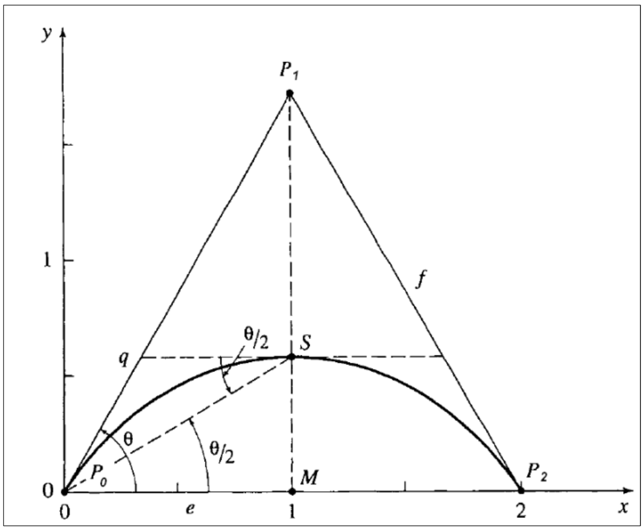


Figure 8.12 Circular arc formed as a rational B-spline curve

Because the triangle $P_0P_1P_2$ is isosceles, S is the maximum point on the curve. Hence, the tangent at S is parallel to the line P_0P_2. The triangle P_0qS is also isosceles, with equal base angles $\angle SP_0q$ and $\angle P_0Sq$, labelled θ/2 in Figure 8.12. Because the tangent at S is parallel to the line P_0P_2, the angles $\angle qSP_0$ and $\angle SP_0M$ are equal. Thus, the angle $\angle SP_0M = \theta/2$ is half the base angle of the isosceles triangle formed by $P_0P_1P_2$.

From Eq. 8.12 and these results, w_1 is

$$w_1 = \frac{M - S}{S - P_1} = \frac{(MS)}{(P_1M) - (MS)} = \frac{e \cdot tan(\frac{\theta}{2})}{f \cdot sin\theta - e \cdot tan(\theta/2)}$$

Recalling that $\tan\left(\frac{\theta}{2}\right) = sin\theta/(1 + cos\theta)$ yields

Eq. 8.13

$$w_1 = \frac{\frac{e \cdot sin(\theta)}{1 + cos\theta}}{f \cdot sin\theta - \frac{e \cdot sin(\theta)}{1 + cos\theta}} = \frac{e}{f(1 + cos\theta) - e} = \frac{e}{f} = cos\theta$$

The portion of the circle subtended by the arc is twice the angle θ. For an arc of 120°, θ=60° and h_2 =1/2. For this particular case, the radius of the circle is 2(S- M).

A full circle is formed by piecing together multiple segments. Specifically, a full circle is given by the three rational quadratic B-spline curve segments, each subtending an arc of 120°. The control polygon vertices form an equilateral triangle, as shown in Figure 8.13.The nonuniform knot and weights vectors are

$$U = [0\ 0\ 0\ 1\ 1\ 2\ 2\ 3\ 3\ 3]$$

$$w = [1\ \tfrac{1}{2}\ 1\ \tfrac{1}{2}\ 1\ \tfrac{1}{2}\ 1]$$

Similarly, a full circle is also given by the four rational quadratic B-spline curve segments, each subtending an arc of 90° with the control polygon forming a square, as shown in Figure 8.14. Here, the nonuniform knot and weights vectors are

$$U = [0\ 0\ 0\ 1\ 1\ 2\ 2\ 3\ 3\ 4\ 4\ 4]$$

$$w = \left[1\ \sqrt{2}/_2\ 1\ \sqrt{2}/_2\ 1\ \sqrt{2}/_2\ 1\ \sqrt{2}/_2\ 1\right]$$

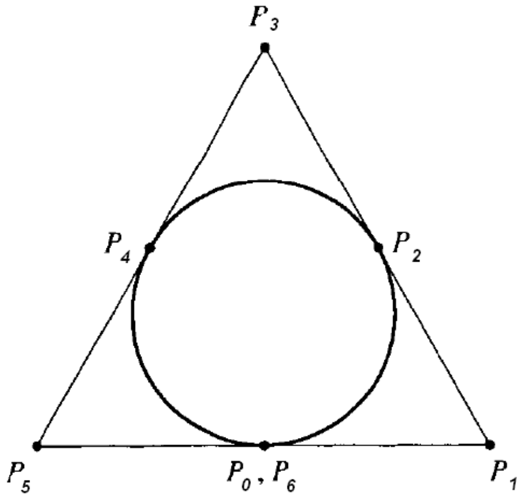


Figure 8.13 Rational B-spline circle consisted of three 120° segments

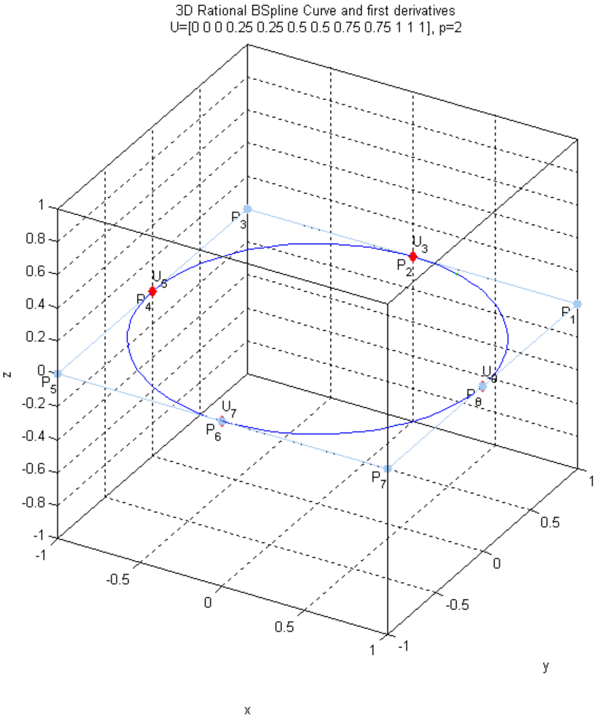


Figure 8.14 Rational B-spline circle assembled by four 90° segments

8.5 Definition and Properties of NURBS Surfaces and Solids (Les Piegel, Wayne Tiller, 1997)

A NURBS surface of degree p in the u direction and degree q in the v direction is a bivariate vector-valued piecewise rational function of the form

$$\boldsymbol{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\,\boldsymbol{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\,w_{i,j}} \qquad 0 \le u,v \le 1$$

The $\{\boldsymbol{P}_{i,j}\}$ form a bidirectional control net, the $\{w_{i,j}\}$ are the weights, and the $\{N_{i,p}(u)\}$ and $\{N_{j,q}(v)\}$ are the nonrational B-spline basis functions defined on the knot vectors

$$U = \{\underbrace{0,\dots,0}_{p+1}, u_{p+1},\dots,u_{r-p-1}, \underbrace{1,\dots,1}_{p+1}\}$$

$$V = \{\underbrace{0,\dots,0}_{q+1}, u_{p+1},\dots,u_{r-p-1}, \underbrace{1,\dots,1}_{q+1}\}$$

where $r = n + p + 1$ and $s = m + q + 1$.

Introducing the piecewise rational basis functions

$$R_{i,j}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m} N_{k,p}(u)N_{l,q}(v)\,w_{k,l}}$$

The surface can be written as

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)\,\boldsymbol{P}_{i,j}$$

The important properties of the functions $R_{i,j}(u,v)$ are roughly the same as those given in Section 6.2 for the nonrational basis functions, $N_{i,p}(u)N_{j,q}(v)$ .We summarize them here.

- Nonnegativity : $R_{i,j}(u,v) \ge 0$ for all $i, j, u,$ and $v;$
- Partition of unity: $\sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v) = 1$ for all $(u,v) \in [0,1] \times [0,1]$;
- Local support: $R_{i,j}(u,v) = 0$ if (u,v) is outside the rectangle given by $[u_i, u_{i+p+1}) \times [u_j, u_{j+p+1})$;
- In any given rectangle of the form $[u_{i_0}, u_{i_0+1}) \times [u_{j_0}, u_{j_0+1})$, at most (p + 1)(q + 1) basis functions are nonzero, in particular the $R_{i,j}(u,v)$ for $i_0 - p \le i \le i_0$ and $j_0 - q \le j \le j_0$ are nonzero;

- Extrema : if p > 0 and q > 0, then $R_{i,j}(u,v)$ attains exactly one maximum value;
- $R_{0,0}(0,0) = R_{n,0}(1,0) = R_{0,m}(0,1) = R_{n,m}(1,1) = 1$;
- Differentiability : interior to the rectangles formed by the u and v knot lines, all partial derivatives of $R_{i,j}(u,v)$ exist. At a u knot (v knot) it is $p - k(q - k)$ times differentiable in the u (v) direction, where k is the multiplicity of the knot;
- If all $w_{i,j} = \mathrm{a}$ for $0 \le i \le n, 0 \le j \le m$, and $a \neq 0$, then $R_{i,j}(u,v) = N_{i,p}(u)N_{j,q}(v)$ for all i,j.

The properties stated above yield the following important geometric properties of NURBS surfaces :

- Corner point interpolation: $\boldsymbol{S}(0,0) = \boldsymbol{P}_{\mathbf{0,0}}, \boldsymbol{S}(1,0) = \boldsymbol{P}_{\boldsymbol{n},\mathbf{0}}, \boldsymbol{S}(0,1) = \boldsymbol{P}_{\mathbf{0},\boldsymbol{m}}$ and $\boldsymbol{S}(1,1) = \boldsymbol{P}_{\boldsymbol{n},\boldsymbol{m}}$;
- Affine invariance : an affine transformation is applied to the surface by applying it to the control points;
- Strong convex hull property : assume $w_{i,j} \ge 0$ for all i,j. If $(u,v) \in [u_{i_0}, u_{i_0+1}) \times [u_{j_0}, u_{j_0+1})$,then $\boldsymbol{S}(u,v)$ is in the convex hull of the control points $\boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}}$, $i_0 - p \le i \le i_0$ and $j_0 - q \le j \le j_0$;
- Local modification: if $\boldsymbol{P}_{\boldsymbol{i},\boldsymbol{j}}$, is moved, or $w_{i,j}$ is changed, it affects the surface shape only in the rectangle $[u_i, u_{i+p+1}) \times [u_j, u_{j+p+1})$;
- Nonrational B-spline and Bezier and rational Bezier surfaces are special cases of NURBS surfaces;
- Differentiability : $\boldsymbol{S}(u,v)$ is $p - k(q - k)$ times differentiable with respect to $u(v)$ at a u knot (v knot) of multiplicity k.

We remark that there is no known variation diminishing property for NURBS surfaces (see [Prau92]).

We can use both control point movement and weight modification to locally change the shape of NURBS surfaces.

It is convenient to represent a NURBS surface using homogeneous coordinates, that is

$$\boldsymbol{S}^{\boldsymbol{w}}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\boldsymbol{P}_{i,j}^{w}$$

where $\boldsymbol{P}_{i,j}^{w} = (w_{i,j}x_{i,j}, w_{i,j}y_{i,j}, w_{i,j}z_{i,j}, w_{i,j})$. Then $\boldsymbol{S}(u,v) = H\{\boldsymbol{S}^{\boldsymbol{w}}(u,v)\}$. We refer interchangeably to either $\boldsymbol{S}^{\boldsymbol{w}}(u,v)$ or $\boldsymbol{S}(u,v)$ as the NURBS surface. Strictly speaking,

$\boldsymbol{S}^{\boldsymbol{w}}(u,v)$ is a tensor product, piecewise polynomial surface in four – dimensional space. $\boldsymbol{S}(u,v)$ is a piecewise rational surface in three – dimensional space; it is not a tensor product surface, since the $R_{i,j}(u,v)$ are not products of univariate basis functions.

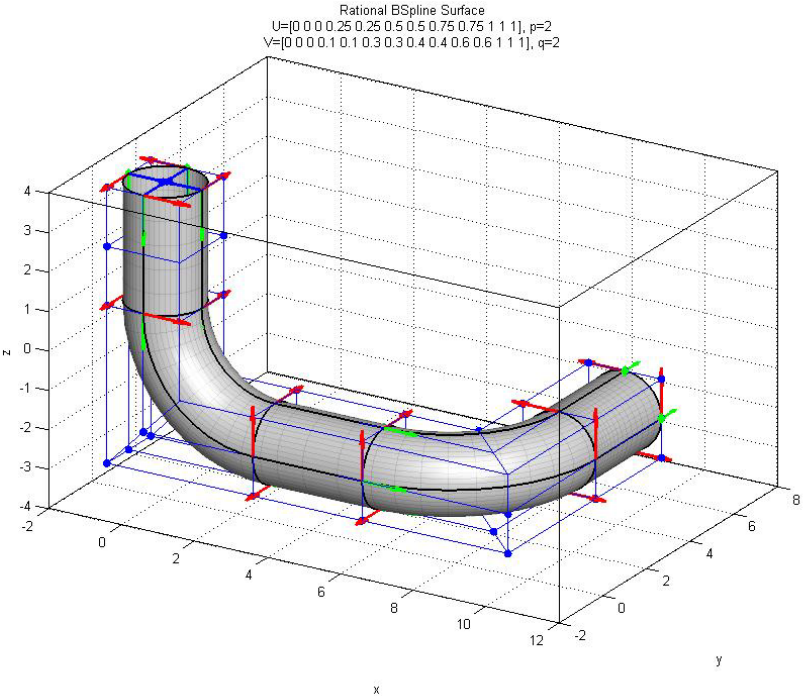


Figure 8.15 A pipe constructed on bivariate NURBS basis (surface geometry) and the derivatives of the physical coordinates w.r.t. the parametric space

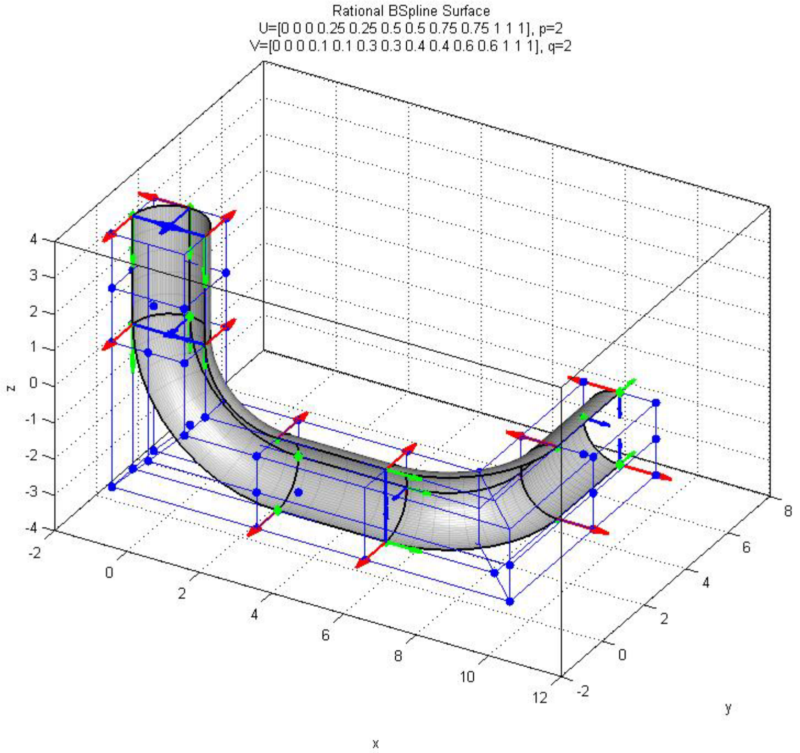


Figure 8.16 The pipe of Figure 8.15 cut in half

Analogously, rational solids are defined in terms of the trivariate rational basis functions

$$R_{i,j,k}(u,v,\omega)=\frac{N_{i,p}(u)N_{j,q}(v)N_{k,r}(\omega)w_{i,j,k}}{\sum_{a=1}^{\mathrm{n}}\sum_{b=1}^{m}\sum_{c=1}^{l}N_{a,p}(u)N_{b,q}(v)N_{c,r}(\omega)\,w_{k,l}}$$

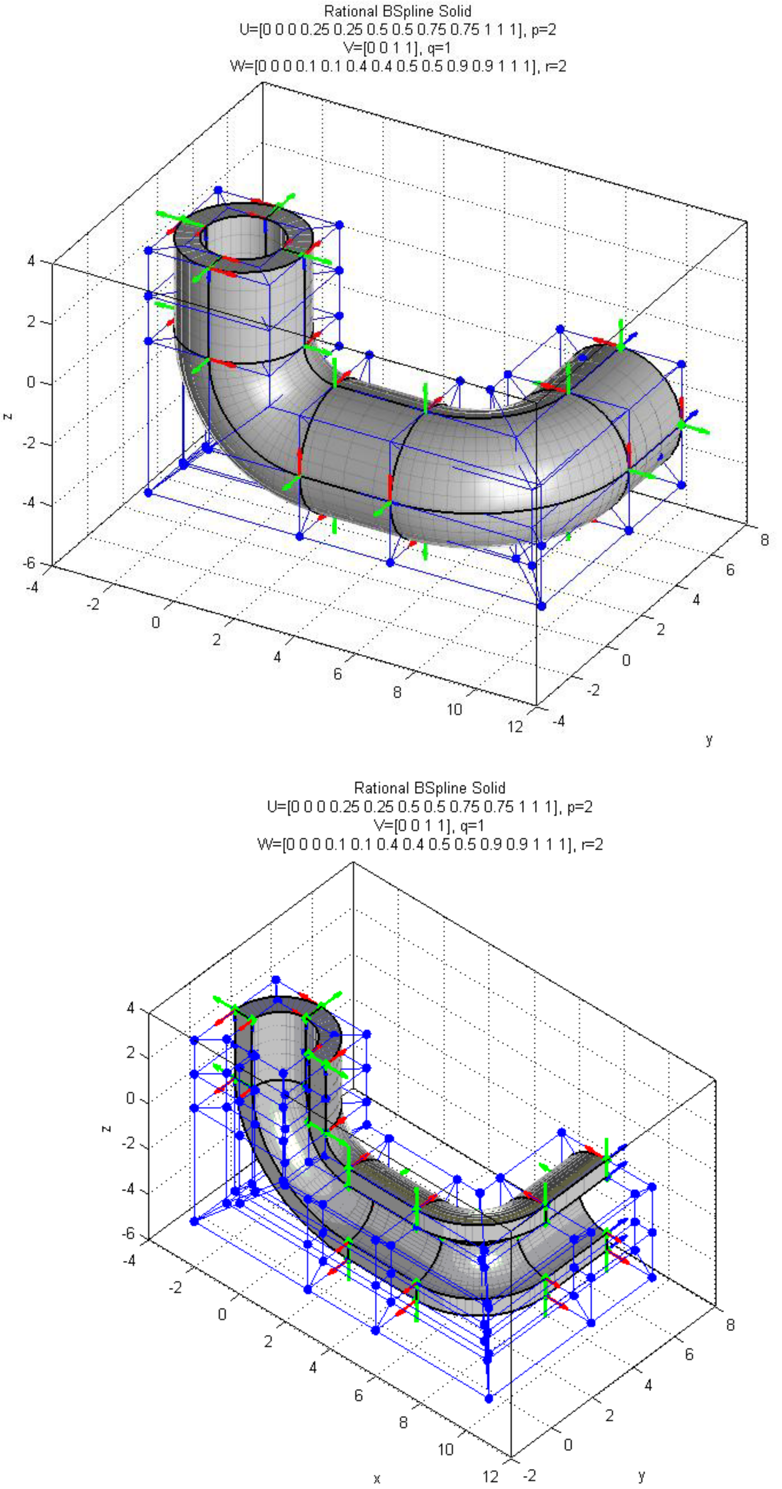


Figure 8.17 The pipe of Figure 8.15 with a thickness of 0.75; the elements used are solid and the basis consists of NURBS functions

8.6 MULTIPLE PATCHES (J. AUSTIN COTTRELL, THOMAS J.R. HUGHES, YURI BAZILEVS, 2009)

In almost all practical circumstances, it will be necessary to describe domains with multiple NURBS patches. For example, if different material or physical models are to be used in different parts of the domain, it might simplify things to describe these subdomains by different patches. Also, if different subdomains are to be assembled in parallel on a multiple processor machine, it is convenient from the point of view of data structures to not have a single patch split between different processors. Most common is the case where the domain simply differs topologically from a cube. The tensor product structure of the parameter space of a patch makes it poorly suited for representing complex, multiply connected domains. Such geometries can frequently be handled quite simply by using multiple patches (see, *e.g.*, Figure 2.29).

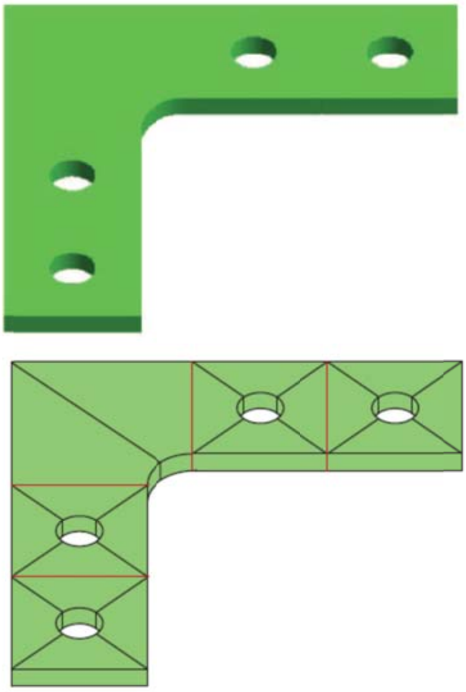


Figure 8.18 The bracket on the top is exactly and concisely represented by five simple NURBS patches (patch boundaries are shown in red, element boundaries in black). The patches match geometrically and parametrically on the internal faces where they meet.

8.7 REFINEMENT (J. AUSTIN COTTRELL, THOMAS J.R. HUGHES, YURI BAZILEVS, 2009)

Automatic adaptive mesh refinement has not been as widely adopted in industry as one might assume from the extensive academic literature, because mesh refinement requires access to the exact geometry and thus seamless and automatic communication with CAD, which simply does not exist. Without accurate geometry and mesh adaptivity, convergence and high-precision results are impossible.

One of the most interesting aspects of B-splines is the myriad of ways in which the basis may be enriched while leaving the underlying geometry and its parameterization intact. To fully recognise the many possibilities, we must first understand the subtle ways in which the basic mechanisms of B-spline refinement differ from their finite element counterparts. These differences lead to more richness in the overall refinement space. In particular, not only do we have control over the element size and the order of the basis, but we can control the continuity of the basis as well.

8.7.1 KNOT INSERTION

The first mechanism by which one can enrich the basis is ***knot insertion***. Knots may be inserted without changing a curve geometrically or parametrically. Given a knot vector $U = \{u_1, u_2, \dots, u_{n+p+1}\}$, we introduce the notion of an extended knot vector $\bar{U} = \{\bar{u}_1 = u_1, \bar{u}_2, \dots, \bar{u}_{n+m+p+1} = u_{n+p+1}\}$, such that $U \subset \bar{U}$. As before, the new n+m functions are formed by Eq. 7.4 , now by applying them to the new knot vector $\bar{U}$. The new n+m control points, $\bar{\boldsymbol{P}} = \{\bar{\boldsymbol{P}}_1, \bar{\boldsymbol{P}}_2, \dots, \bar{\boldsymbol{P}}_{n+m}\}^T$ are formed from linear combinations of the original control points, $\boldsymbol{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_n\}^T$, by

Eq. 8.14

$$\bar{\boldsymbol{P}} = T^p\, \boldsymbol{P}$$

where

$$T_{ij}^0 = \begin{cases} 1 & \bar{u}_i \in [u_j, u_{j+1}) \\ 0 & otherwise \end{cases}$$

and

$$T_{ij}^{q+1} = \frac{\bar{u}_{i+q} - u_j}{u_{j+q} - u_j} T_{ij}^q + \frac{u_{j+q+1} - \bar{u}_{i+q}}{u_{j+q+1} - u_{j+1}} T_{ij+1}^q \quad for\ q = 0,1,2, \dots, p-1$$

Knot values already present in the knot vector may be repeated in this way, thereby increasing their multiplicity, but the continuity of the basis will be reduced. However the continuity of the curve is preserved by choosing the control points as in Eq. 8.14.

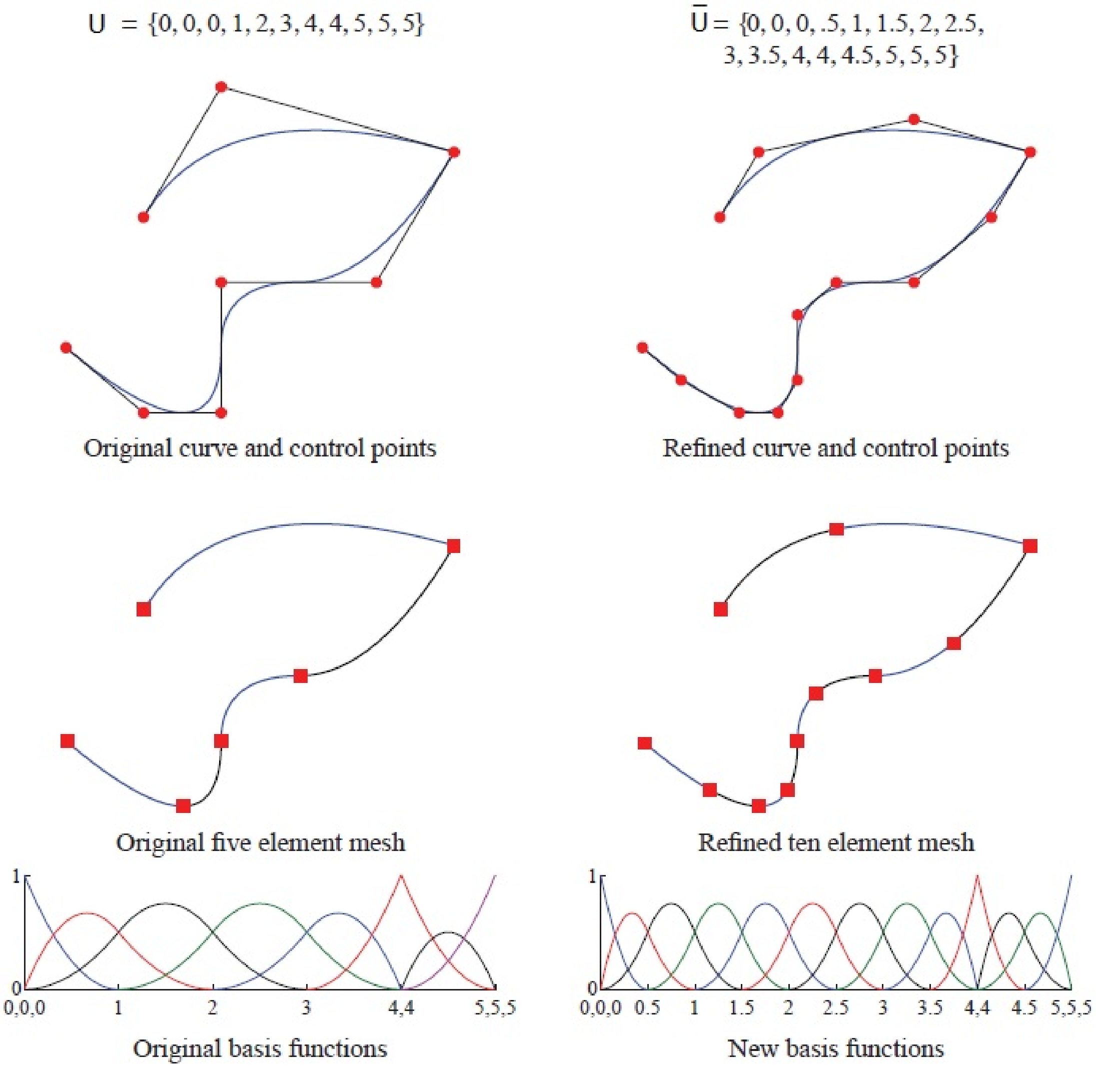


Figure 8.19 Knot insertion. Control points are denoted by●. The knots, which define a mesh by partitioning the curve into elements, are denoted by ■. Each element has been evenly split in the parametric domain.

Insertion of the new knot values clearly has similarities with the classical h-refinement strategy in finite element analysis as it splits existing elements into new ones. It differs, however, in the number of the new functions that are created, as well as in the continuity of the basis across the newly created element boundaries (C^{p-1} in this case). To perfectly replicate h-refinement, one would need to insert each of the new knot values p times so that the functions will be C^0 across the new boundary. The alternative to inserting new knot values – increasing the multiplicity of existing knot values to decrease the continuity of the basis without creating new elements – does not have an analogue in FEA, as FEA meshes have C^0 element boundaries to begin with. In this way, knot insertion is very closely related, but not identical to h-refinement.

8.7.2 ORDER ELEVATION

The second mechanism by which one can enrich the basis is ***order elevation*** (sometimes also called “degree elevation”). As its name implies, the process involves raising the polynomial order of the basis functions used to represent the geometry. Recalling that the basis has $p - m_i$ continuous derivatives across element boundaries, it is clear that when p is increased, m_i must also be increased if we are to ***preserve*** the discontinuities in the various derivatives already existing in the original curve. During order elevation, the multiplicity of each knot value is increased by one, but no new knot values are added. As with knot insertion, neither the geometry nor the parameterization are changed.

The process for order elevation begins by replicating existing knots until their multiplicity is equal to the polynomial order, thus effectively subdividing the curve into many Bézier curves by knot insertion. The next step is to elevate the order of the polynomial on each of these individual segments. Lastly, excess knots are removed to combine the segments into one, order-elevated, B-spline curve. Several efficient algorithms exist which combine the steps so as to minimize the computational cost of the process. For a thorough treatment, see (Les Piegel, Wayne Tiller, 1997).

Order elevation clearly has much in common with the classical *p*-refinement strategy in finite element analysis as it increases the polynomial order of the basis. The major difference is that *p*-refinement always begins with a basis that is C^0 everywhere, while order elevation is compatible with any combination of continuities that exist in the unrefined B-spline mesh. This flexibility leads us to a new higher-order technique that is unique to isogeometric analysis.

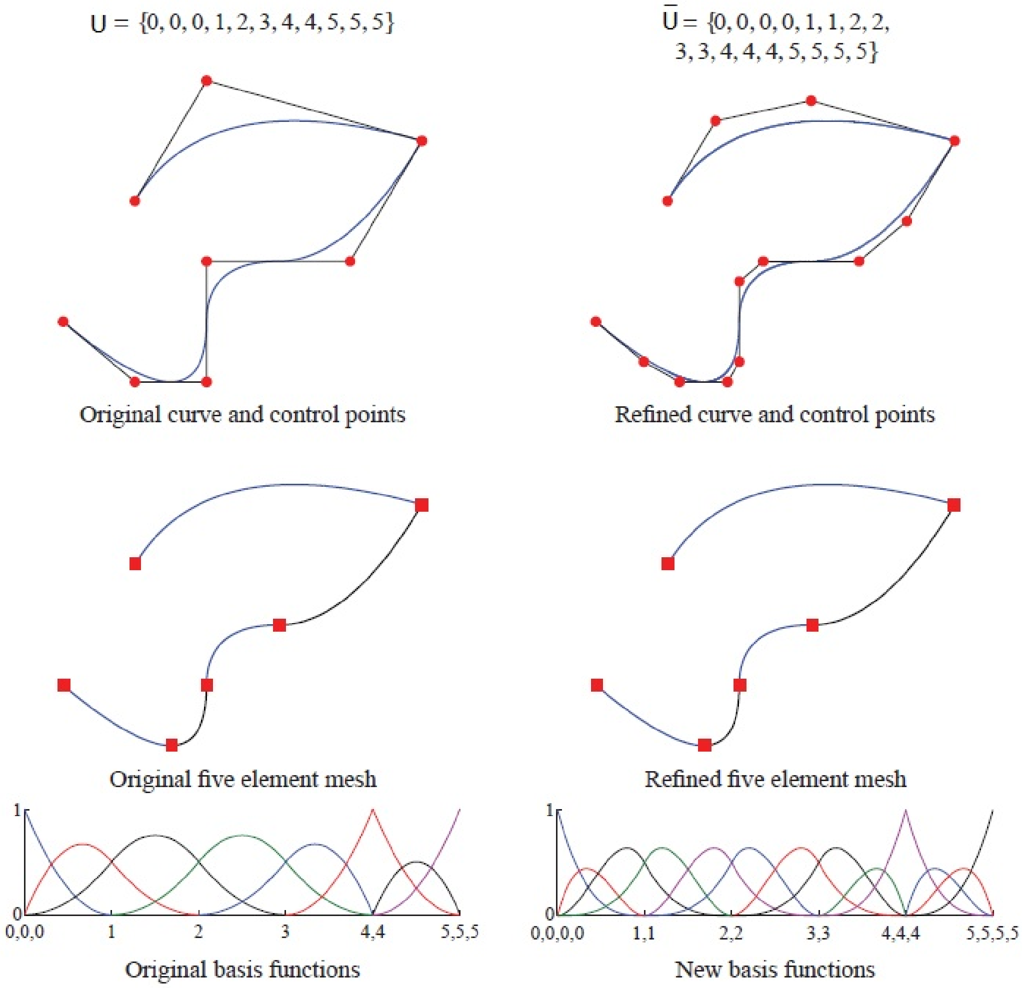


Figure 8.20 Order elevation. Control points are denoted by●. The knots, which define a mesh by partitioning the curve into elements, are denoted by ◼. Note the increased multiplicity of internal knots. This is done to preserve discontinuities in the appropriate derivatives of the curve.

8.7.3 k-REFINEMENT: HIGHER ORDER AND HIGHER CONTINUITY

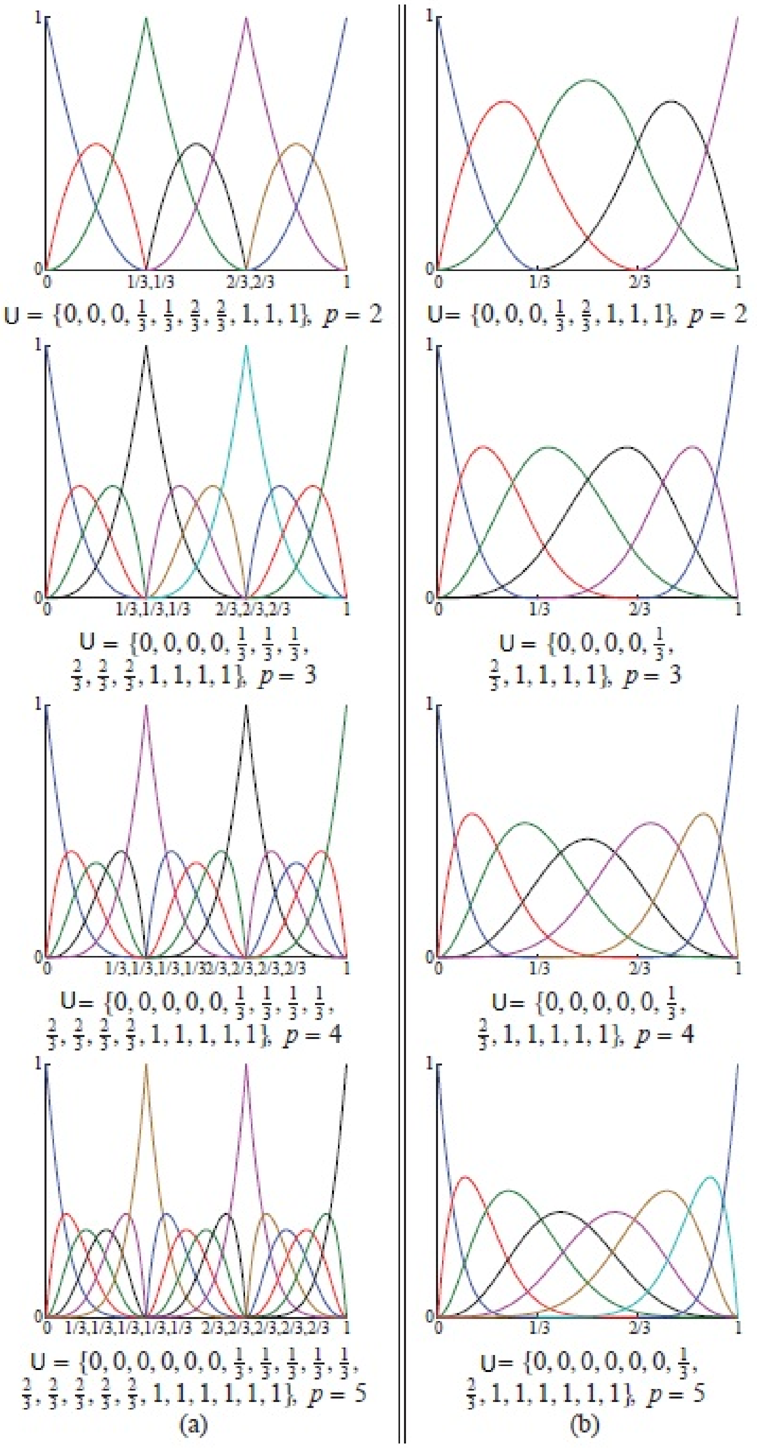
As we have seen, the two primitive refinement operations for B-splines are knot insertion and order elevation. Knot insertion is similar to *h*-refinement, but for it to be a perfect analogue each new knot value would have to be inserted with multiplicity m_i - p to ensure a C^0 basis everywhere. Similarly, if we begin with a mesh in which all of the functions are already C^0 across element boundaries, order elevation coincides exactly with the traditional notion of *p*-refinement. Knot insertion and order elevation, however, provide us with more to work with than do the two standard notions of refinement.

As mentioned above, we can insert new knot values with multiplicities equal to one to define new elements across whose boundaries functions will be C^{p-1}. We can also repeat existing knot values to lower the continuity of the basis across existing element boundaries. This makes knot insertion a more flexible process than simple *h*-refinement. Similarly, we have a more flexible higher-order refinement as well. It stems from the fact that the processes of order elevation and knot insertion do not commute. If a unique knot value, $\bar{\xi}$, is inserted between two distinct knot values in a curve of order p, the number of continuous derivatives of the basis functions at $\bar{\xi}$ is $p - 1$. If we subsequently elevate the order to q, the multiplicity of every distinct knot value (including the knot just inserted) is increased so that discontinuities in the p^{th} derivative of the basis are preserved. That is, the basis still has $p - 1$ continuous derivatives at $\bar{\xi}$, although the polynomial order is now q. If, instead, we elevated the order of the original, coarsest curve to q and only then inserted the unique knot value $\bar{\xi}$, the basis would have $q - 1$ continuous derivatives at $\bar{\xi}$. We refer to this latter procedure as ***k-refinement***. We know of no analogous practice in standard finite element analysis.

It is important that we point out that this notion of *k*-refinement is not the same as the "*k*-convergence" in which the position of the knots is altered. It bears more in common with the "*k*-version finite element method" in that *k* refers to continuity, but the motivations are different. The increased continuity is required so that a least-squares finite element approach is possible. Such an approach requires that the solution space have the same number of continuous derivatives as found in the highest order derivative of the differential operator. Our motivations for using basis functions of higher continuity are efficiency and robustness of the solution space in a classical Galerkin finite element formulation of the problem.

The concept of *k*-refinement is potentially a superior approach to high-precision analysis than *p*-refinement. In traditional *p*-refinement there is a very inhomogeneous structure to arrays due to the different basis functions associated with surface, edge, vertex and interior nodes. In addition, there is a proliferation in the number of nodes because C^0-continuity is maintained in the refinement process. In *k*-refinement, there is a homogeneous structure within patches and growth in the number of control variables is limited.

It is also important to note that "pure" *k*-refinement, where all functions maintain maximal C^{p-1} continuity across element boundaries, is only possible if the coarsest mesh is comprised of a single element. If the initial mesh places constraints on the continuity across certain element boundaries, these constraints will exist on all meshes. In general, though some such constraints will exist, the number of elements desired for analysis will be much higher than the number needed for modeling the geometry. Refinements may be performed such that the functions have $p - 1$ continuous derivatives across these new element boundaries and the benefits of *k*-refinement will still be significant.



$U = \{0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1\},\ p = 2$

$U = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\},\ p = 2$

$U = \{0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1\},\ p = 3$

$U = \{0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1\},\ p = 3$

$U = \{0, 0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1, 1\},\ p = 4$

$U = \{0, 0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1, 1\},\ p = 4$

$U = \{0, 0, 0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1, 1, 1, 1\},\ p = 5$

(a)

$U = \{0, 0, 0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1, 1, 1\},\ p = 5$

(b)

Figure 8.21 Three element, higher-order meshes for *p*- and *k*-refinement. (a) The *p*-refinement approach results in many functions that are C^0 across element boundaries. (b) In comparison, *k*-refinement results in a much smaller number of functions, each of which is C^{p-1} across element boundaries.

8.7.4 THE *hpk*-REFINEMENT SPACE

As we have shown, knot insertion and order elevation are the primitive operations by which classical *h*- and *p*-refinements, as well as the new *k*-refinement, can be implemented. Recognizing their flexibility as compared with classical refinement procedures makes feasible the notion of an *hpk*-refinement space. Recalling that B-spline curves may have no more than $p-1$ continuous derivatives across an element boundary, the set of possible refinements may be characterized as in Figure 8.22.

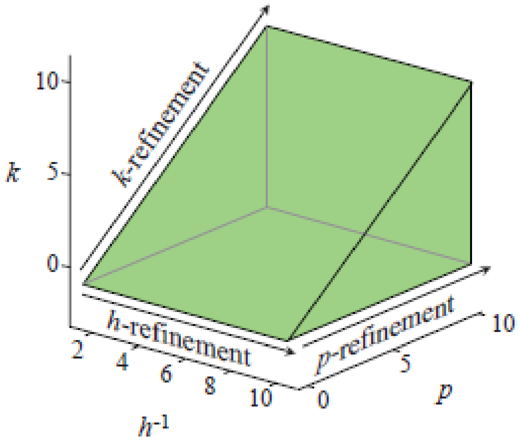


Figure 8.22 The hpk-space. The set of all allowable refinements is contained in the region shown in green. Note that this region extends in the direction of the arrows

Pure *k*-refinement keeps *h* fixed but increases the continuity along with the polynomial order. Pure *p*-refinement increases the polynomial order while the basis remains C^0. Increasing the multiplicity of existing knot values decreases the continuity without introducing new elements. Inserting new knot values with a multiplicity of *p* results in classical *h*-refinement, whereby new elements are introduced that have C^0 boundaries. Inserting new knot values with a multiplicity of 1 decreases *h* without decreasing the minimum continuity already found in the mesh. Considering all of the aforementioned techniques results in a multitude of refinement options beyond simple *h*-, *p*- and *k*-refinement.

1 APPENDIX A

In this appendix we introduce a different way for the construction of the B-splines basis functions. In (Richard H. Bartels, John C. Beatty, Brian A. Barsky, 1987) each basis function is constructed of a set of piecewise spline segments bonded under some end conditions at knots.

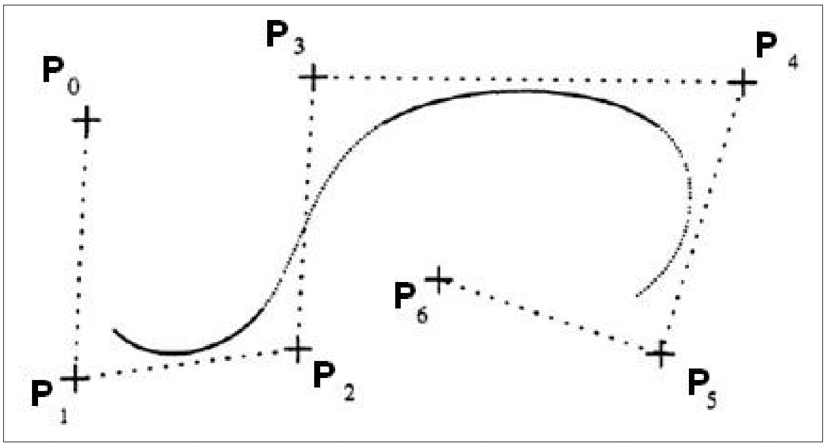


Figure 1.1 An example of a curve defined by a sequence of control vertices, represented here by "+" signs, near which the curve passes. The lightly dotted line connecting the control vertices forms the control polygon

1.1 LINEAR B-SPLINES (RICHARD H. BARTELS, JOHN C. BEATTY, BRIAN A. BARSKY, 1987)

The way in which local control is obtained is most easily explained by considering first a piecewise linear interpolation of the control vertices. Consider the "curve" shown in Figure 1.2.

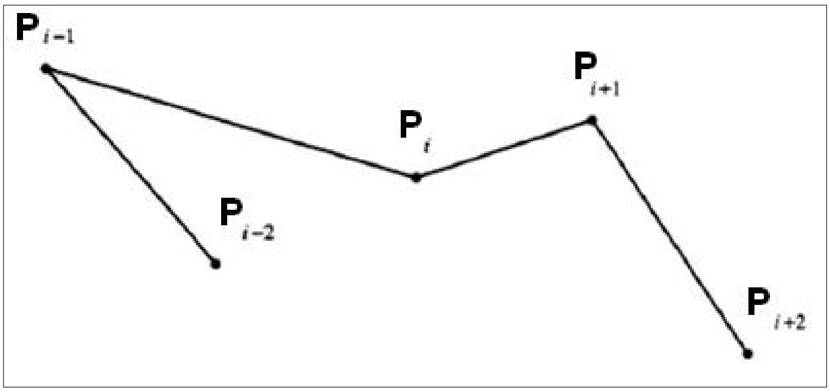


Figure 1.2 A piecewise linear curve

If we represent the segments of this curve in the obvious way we have

$$\begin{aligned}\boldsymbol{C}_{i-1}(\bar{u}) &= \left(X_{i-1}(\bar{u}), Y_{i-1}(\bar{u})\right)\\ &= (1-\bar{u})\boldsymbol{P}_{i-2} + \bar{u}\boldsymbol{P}_{i-1} \quad for\ \bar{u} = (u-u_{i-1})/(u_i - u_{i-1})\end{aligned}$$

$$\begin{aligned} \boldsymbol{C}_i(\bar{u}) &= \big(X_i(\bar{u}), Y_i(\bar{u})\big) \\ &= (1-\bar{u})\boldsymbol{P}_{i-1} + \bar{u}\boldsymbol{P}_i \;\; for\ \bar{u} = (u-u_i)/(u_{i+1}-u_i) \end{aligned}$$

$$\begin{aligned} \boldsymbol{C}_{i+1}(\bar{u}) &= \big(X_{i+1}(\bar{u}), Y_{i+1}(\bar{u})\big) \\ &= (1-\bar{u})\boldsymbol{P}_i + \bar{u}\boldsymbol{P}_{i+1} \;\; for\ \bar{u} = (u-u_{i+1})/(u_{i+2}-u_{i+1}) \end{aligned}$$

$$\begin{aligned} \boldsymbol{C}_{i+2}(\bar{u}) &= \big(X_{i+2}(\bar{u}), Y_{i+2}(\bar{u})\big) \\ &= (1-\bar{u})\boldsymbol{P}_{i+1} + \bar{u}\boldsymbol{P}_{i+2} \;\; for\ \bar{u} = (u-u_{i+2})/(u_{i+3}-u_{i+2}) \end{aligned}$$

Where

$$X_{i-1}(\bar{u}) = (1-\bar{u})x_{i-2} + \bar{u}x_{i-1}$$

$$Y_{i-1}(\bar{u}) = (1-\bar{u})y_{i-2} + \bar{u}y_{i-1}$$

$$X_i(\bar{u}) = (1-\bar{u})x_{i-1} + \bar{u}x_i$$

$$Y_i(\bar{u}) = (1-\bar{u})y_{i-1} + \bar{u}y_i$$

$$X_{i+1}(\bar{u}) = (1-\bar{u})x_i + \bar{u}x_{i+1}$$

$$Y_{i+1}(\bar{u}) = (1-\bar{u})y_i + \bar{u}y_{i+1}$$

$$X_{i+2}(\bar{u}) = (1-\bar{u})x_{i+1} + \bar{u}x_{i+2}$$

$$Y_{i+2}(\bar{u}) = (1-\bar{u})y_{i+1} + \bar{u}y_{i+2}$$

Altering $\boldsymbol{P}_i$ clearly affects only the two segments $\boldsymbol{C}_i(\bar{u})$ and $\boldsymbol{C}_{i+1}(\bar{u})$ which are adjacent to it: $\boldsymbol{P}_i$ does not appear in the formulas for any other segments. Let us represent our piecewise linear curve so as to isolate the individual influence of each separate control vertex. Doing this will make straightforward the generalization to higher-order, smoother, piecewise polynomial curves.

If we plot $Y(u)$ as a function of u, and represent the contribution of y_i to $Y(u)$ by a dashed line, we obtain Figure 1.3

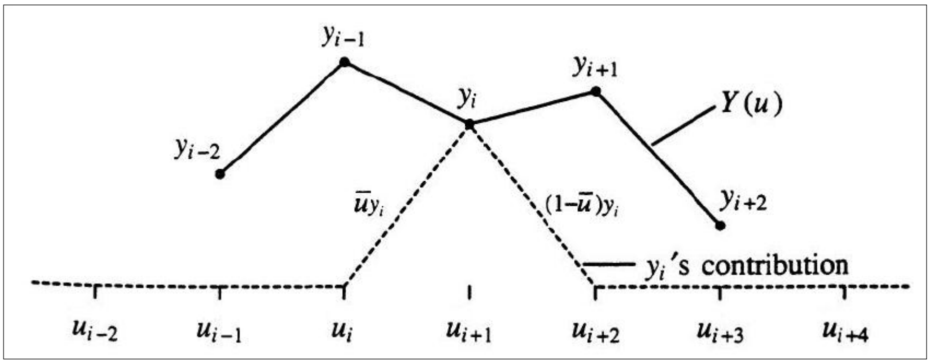


Figure 1.3 The contribution of y_i to Y(u)

Notice that this contribution is zero both to the left of u_i and to the right of u_{i+2}. Similarly, the contribution of y_{i-1} is shown in Figure 1.4.

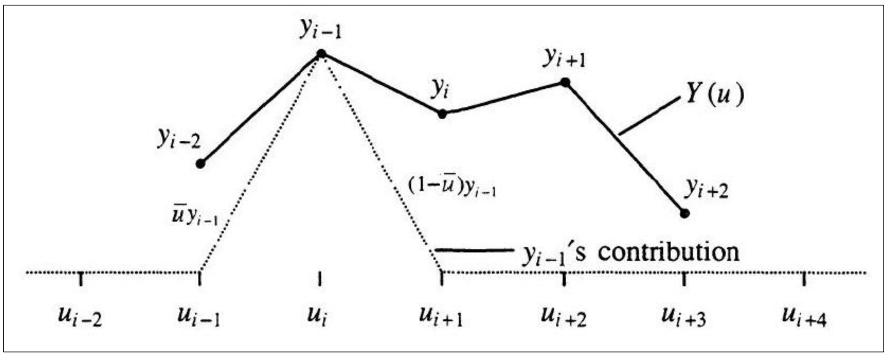


Figure 1.4 The contribution of y_{i-1} to Y(u)

Plotting these two ***hat functions*** together gives us, in Figure 1.5, a graphical representation of the fact that $Y_i\ (\bar{u}) = (1-\bar{u})y_{i-1} + \bar{u}y_i$. (For the sake of clarity we will stop extending these hat functions to the left and right by zero when we draw them because these extensions would all be drawn on top of one another).

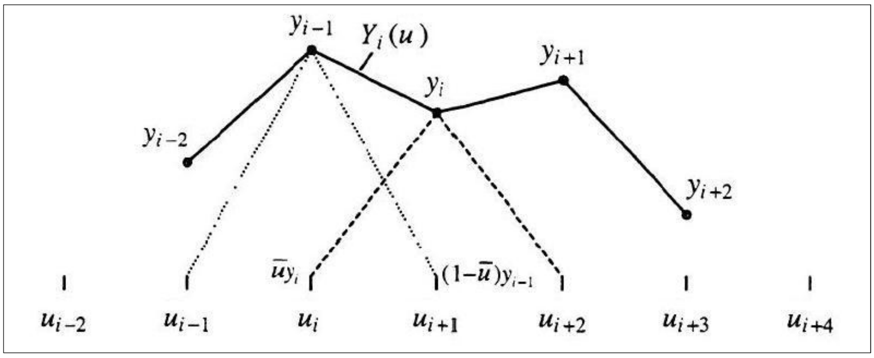


Figure 1.5 A simultaneous look at the contributions of y_{i-1} and y_i to the curve in general, and to $Y_i(u)$ in particular

It is useful to think of y_{i-1} and y_i as each scaling a corresponding unit hat function (see below) whose maximum height is one. These hat functions are all translations of one another. It is also useful to think of y_{i-1} and y_i as each being weighted by a corresponding unit hat function. We make the distinction between weighting and scaling to emphasize that the y 's can have any value: positive, negative, or zero. As such, they act like scale factors. The hat functions, on the other hand, are designed to have only a nonnegative value for any value of u and, as such, they act like weights.

As u increases from u_{i-1} , the contribution of y_{i-1} grows from nothing at $u = u_{i-1}$, peaks at $u = u_i$, and dies away to nothing again at $u = u_{i+1}$. The contribution of y_i is similar on the interval $u_i \le u < u_{i+2}$. More profoundly, we have seen that $\boldsymbol{C}(\mathrm{u})$ is entirely determined by $\boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_i$ alone in the interval $\mathrm{u}_i \le \mathrm{u} < \mathrm{u}_{i+1}$. In this interval $Y(\mathrm{u})$ is just a weighted average of y_{i-1} and y_i namely

$$Y_i(u) = \left(\frac{u_{i+1} - u}{u_{i+1} - u_i}\right) y_{i-1} + \left(\frac{u - u_i}{u_{i+1} - u_i}\right) y_i = (1-\bar{u})y_{i-1} + \bar{u}y_i\ .$$

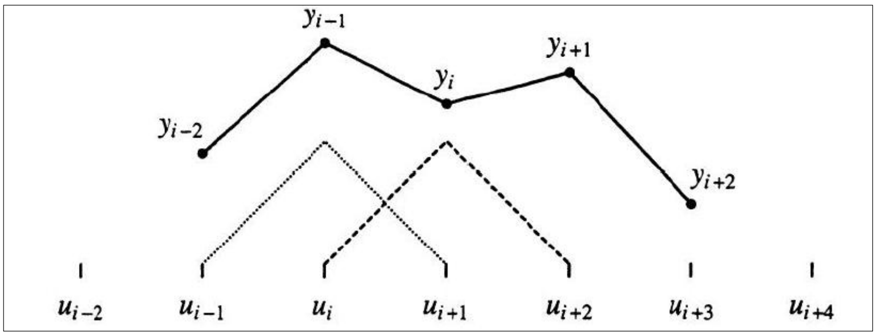


Figure 1.6 Multiplying the two unit (height one) hat functions shown here by y_{i-1} and y_i yields the scaled hat functions shown in Figure 1.5

If we call the "dotted" unit hat function $B_{i-1}(u)$ and the "dashed" unit hat function $B_i(u)$ (to be compatible with later material we name a hat function after the knot at its left extremity), then the line segment attaching y_{i-1} to y_i may be written as

Eq. 1.1

$$Y_i(u) = y_{i-1}B_{i-1}(u) + y_iB_i(u) \;\; for\; u_i \le u < u_{i+1}$$

where

Eq. 1.2

$$B_i(u) = \begin{cases} \dfrac{u - u_i}{u_{i+1} - u_i} & u_i \le u < u_{i+1} \\ \dfrac{u_{i+2} - u}{u_{i+2} - u_{i+1}} & u_{i+1} \le u < u_{i+2} \end{cases}$$

We can represent the other segments of a piecewise linear curve in the same way; Eq. 1.1 is quite general. In Figure 1.7 we show all the hat functions $B_{i-2}(u), \dots, B_{i+2}(u)$ that define our example "curve". Depending on the point of view we wish to take, we may speak of Eq. 1.3 as a ***linear combination*** of the functions B_i, or as a ***weighted sum*** of the control vertices $\boldsymbol{P_i}$. For any particular i, Eq. 1.3 reduces to Eq. 1.1 since all the hat functions except $B_{i-1}(u)$ and $B_i(u)$are zero inside the interval from $u_i \; to \; u_{i+1}$. With enough hat functions, we can represent any piecewise linear curve in this way. The unit hat functions $B_i(u)$ are called ***basis functions*** for this reason. We may now turn our argument around: any particular vertex $\boldsymbol{P_i}$ contributes to the curve we are defining only where $B_i(u)$ is nonzero. Since $B_i(u)$ is nonzero only over the two successive intervals $[u_i, u_{i+1}]$ and $[u_{i+1}, u_{i+2}]$ the actual position of $\boldsymbol{P_i}$ can influence only the two corresponding segments $\boldsymbol{C_i}(u)$ and $\boldsymbol{C_{i+1}}(u)$ of the curve. The result is local control.

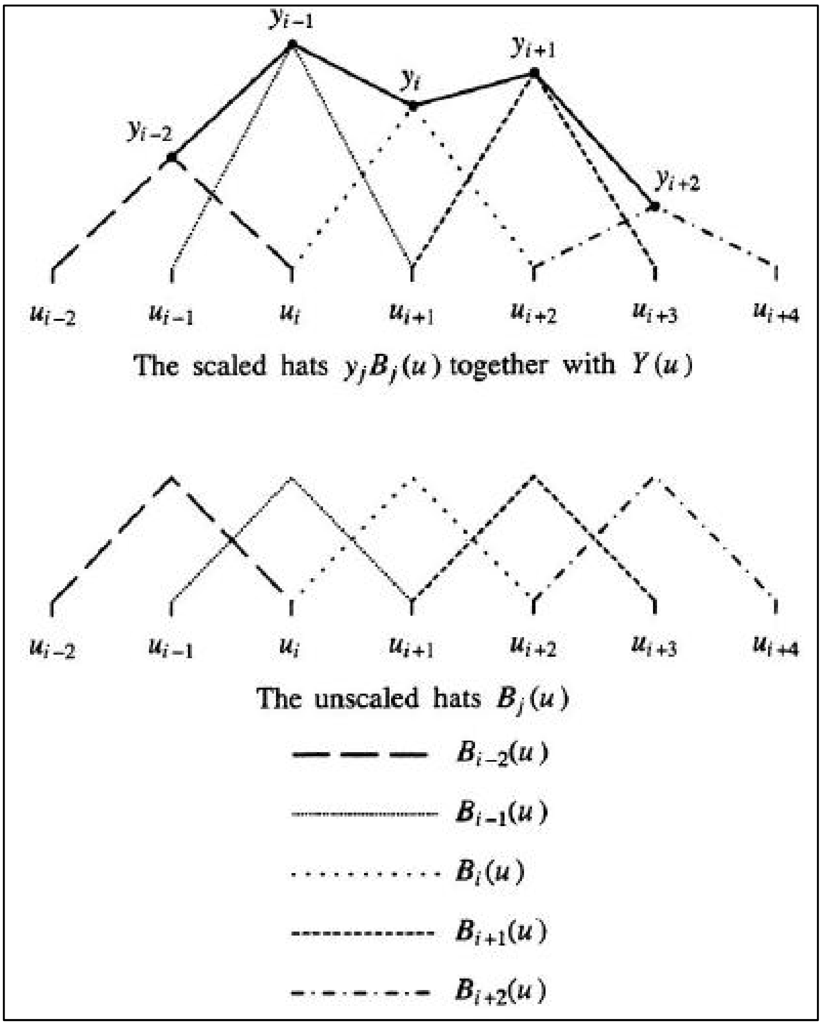


Figure 1.7 Representing a piecewise linear curve as a linear combination of hat functions

The entire curve can now be written as

Eq. 1.3

$$\boldsymbol{C}(u) = \sum_i \boldsymbol{P}_i B_i(u) = \sum_i (x_i B_i(u), y_i B_i(u))\,.$$

Notice that we have made use of the half-open intervals $[u_i\,, u_{i+1})$ and $[u_{i+1}\,, u_{i+2})$ so that $\boldsymbol{C}_i(u)$ defines the curve for values of u up to but not including u_{i+1} because the first interval is open at the right. $\boldsymbol{C}_{i+1}(u)$ then takes over at u_{i+1} itself because the second interval is closed at the left.

The hat functions that we have introduced are continuous, although their derivatives usually have jumps at knots (the technical term is C^0 continuous). Consequently, when we use them to weight control vertices and sum them using Eq. 1.3 we obtain a curve that is continuous, but whose first derivative vector may be discontinuous at knots: a piecewise linear curve, as we knew from the beginning.

1.2 UNIFORM CUBIC B-SPLINES

Our real objective, of course, is to define curves like the one in Figure 1.8 by assembling pieces that are curved rather than straight. As in Chapter 2.3, and for the same reasons, we choose to consider piecewise cubic curves.

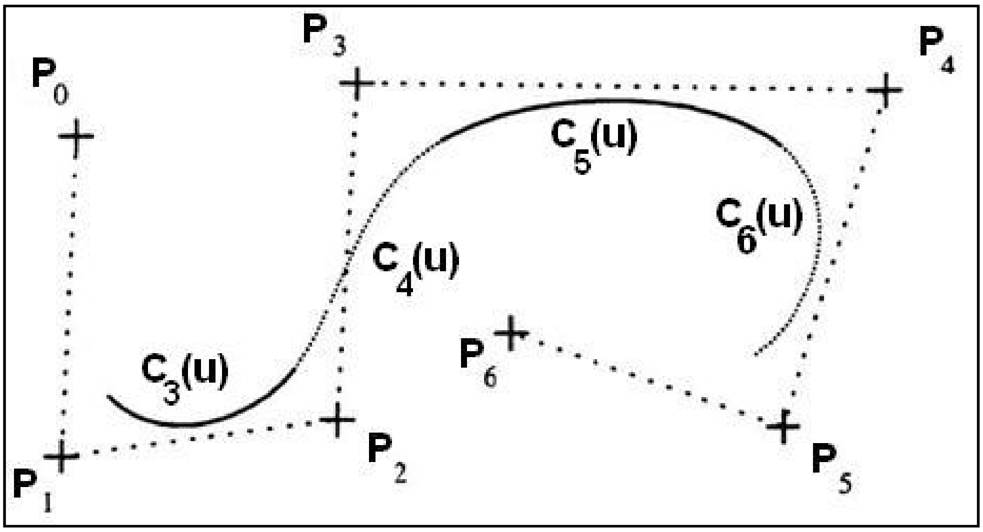


Figure 1.8 The curve shown is constructed from cubic segments so as to approximate the indicated vertices smoothly. The reason for numbering the curve segments as shown will become clear later.

The technique we are now developing does not, in general, interpolate the control vertices - that is a special property of the piecewise linear curves we have considered. Instead, each sequence of control vertices defines a curve that "passes near" those vertices. As before, we may restrict our attention to a single coordinate such as $Y(u)$, as shown in Figure 1.9.

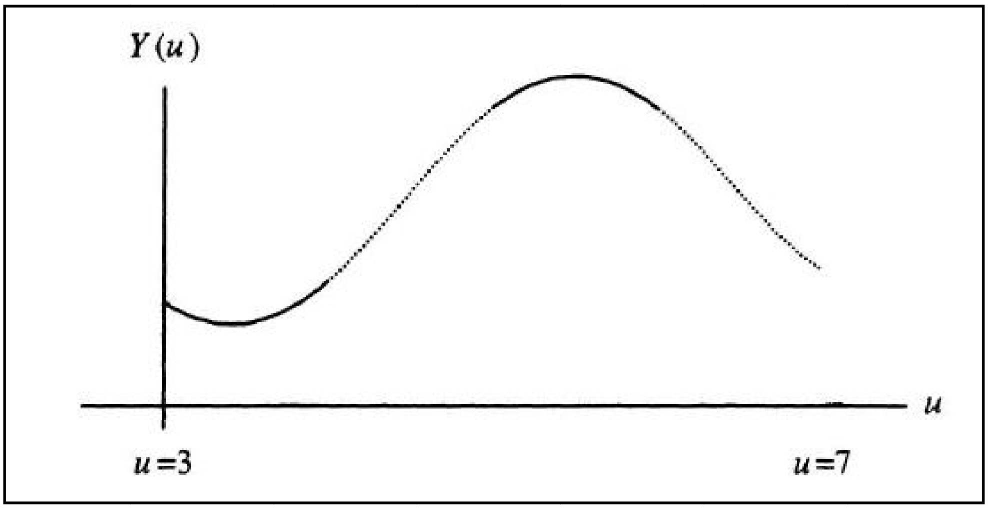


Figure 1.9 $Y(u)$, the y component of the curve in Figure 1.8

We focus on piecewise cubic polynomial curves assembled from cubic polynomials $X_i(\bar{u})$ and $Y_i(\bar{u})$ that have positional, first derivative and second derivative continuity (C^2 continuity) at the joints between successive segments, so that they satisfy the equations

$$\boldsymbol{C}_{i-1}(u_i) = \boldsymbol{C}_i(u_i)$$

$$\boldsymbol{C}_{i-1}^{(1)}(u_i) = \boldsymbol{C}_i^{(1)}(u_i)$$

$$\boldsymbol{C}_{i-1}^{(2)}(u_i) = \boldsymbol{C}_i^{(2)}(u_i)$$

In particular, this implies that

$$Y_{i-1}(u_i) = Y_i(u_i)$$

$$Y_{i-1}^{(1)}(u_i) = Y_i^{(1)}(u_i)$$

$$Y_{i-1}^{(2)}(u_i) = Y_i^{(2)}(u_i)$$

and similarly for $X(u)$. We can achieve the desired continuity if the basis functions with which we define $X(u)$ and $Y(u)$ are themselves C^2 continuous piecewise cubic polynomials with knots at the u_i , since a linear combination (scaled sum) of such basis functions will also be a C^2 continuous piecewise cubic polynomial. Much as for the piecewise linear case, locality can be obtained if all but a small number of the parametric polynomial segments defining a basis function are identically zero. The basis functions we use will be smoother, and it turns out that this means they have to be nonzero on a somewhat wider interval, but the construction is otherwise quite analogous to the linear case we have already considered. For example, $Y(\mathrm{u})$ for the curve of Figure 1.8 can be represented in the following way as a sum of scaled C^2 continuous piecewise cubic basis functions.

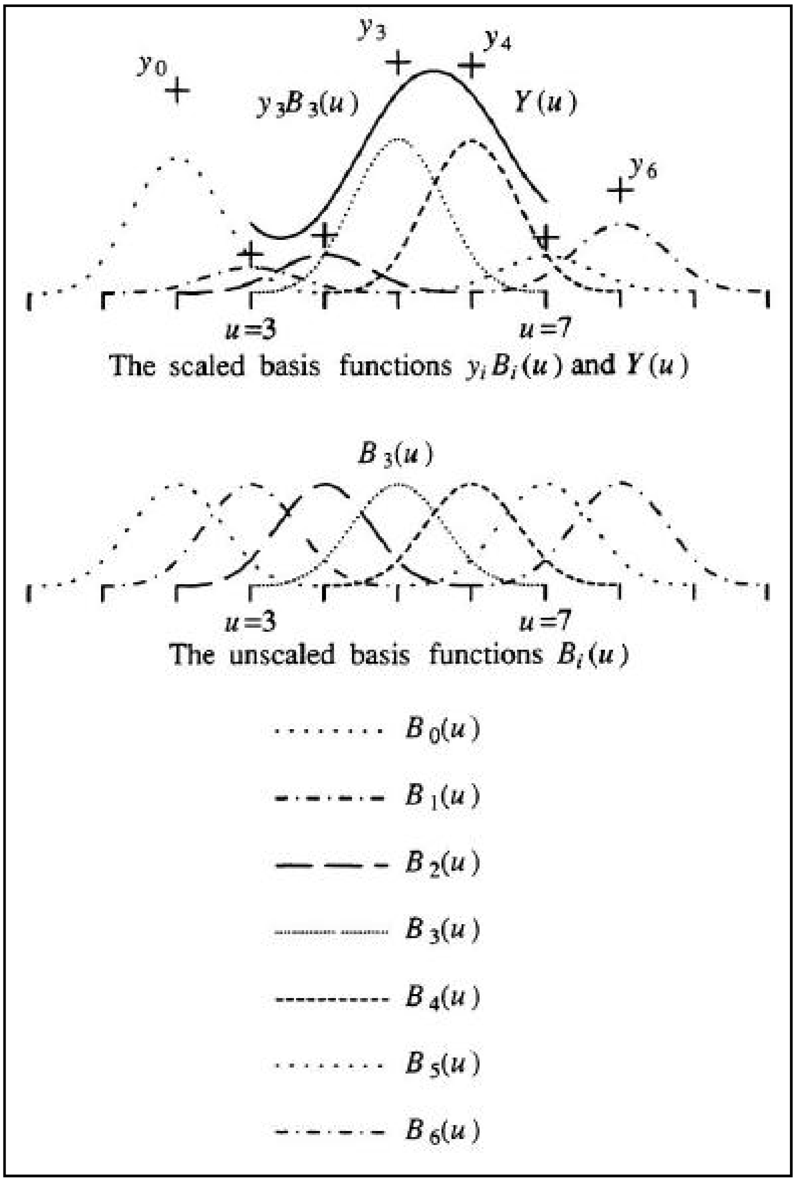


Figure 1.10 The curve of Figure 1.9 as a scaled sum of basis functions

Figure 1.10 illustrates several conventions. We choose to index control vertices from zero through m (here 6). As we will see, it requires four basis functions to properly define each cubic curve segment. Hence there are three more basis functions (and three more control vertices) than there are curve segments. Each basis function is nonzero over four parametric intervals. The leftmost basis function extends three additional intervals to the left of the curve, and the rightmost basis function extends three additional intervals to the right. Summarizing: there are m + 1 control vertices, m + 1 basis functions, m - 2 curve segments bounded by m - 1 knots, and m - 1 + 3 + 3 = m + 5 knots altogether. The curve is generated (swept out) as u runs from u_3 to u_{m+1}.

$$u_0 < u_1 < u_2 < u_3 = u_{min}$$

$$u_{min} = u_3 < u_4 < \cdots < u_{m+1} = u_{max}$$

$$u_{max} = u_{m+1} < u_{m+2} < u_{m+3} < u_{m+4}$$

Let us see how we might actually define these basis functions. Using a little foresight, we suppose each basis function to be nonzero over four successive intervals (which for convenience we assume all have length one), as shown in Figure 1.11, and ask that within each interval a basis function be defined by a cubic polynomial

$$a_j + b_j u + c_j u^2 + d_j u^3, \quad i - 3 \le j \le i.$$

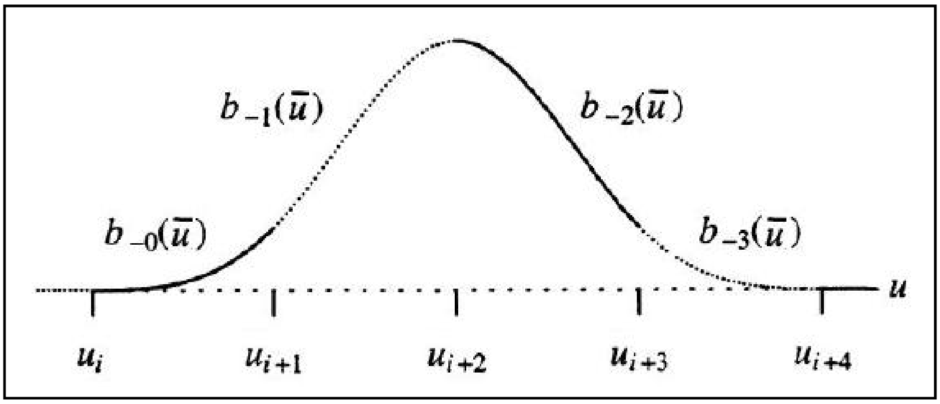


Figure 1.11 The uniform cubic B-spline $B(u)$ is a cubic C^2 basis function centred at u_{i+2}. It is zero for $u \le u_i$and for $u \ge u_{i+4}$. The nonzero portion of $B_i(u)$ is composed of the four polynomial segments $b_{-0}(\bar{u}), b_{-1}(\bar{u}), b_{-2}(\bar{u})$ and $b_{-3}(\bar{u})$

Since the nonzero portion of our cubic basis function $B(u)$ consists (from left to right) of four basis segments $b_{-0}(\bar{u}), b_{-1}(\bar{u}), b_{-2}(\bar{u})$ and $b_{-3}(\bar{u})$, and since each segment has four coefficients, there are sixteen coefficients to determine. By assumption $B_i(u)$ is identically zero for $u \le u_i$ and for $u \ge u_{i+4}$, so the first and second derivatives $B_i^{(1)}(u)$ and $B_i^{(2)}(u)$ are also identically zero outside the interval (u_i, u_{i+4}). The requirement that positions, first derivatives, and second derivatives match at each knot u_j then implies that

Table 1

$0 = b_{-0}(0)$	$0 = b_{-0}^{(1)}(0)$	$0 = b_{-0}^{(2)}(0)$
$b_{-0}(1) = b_{-1}(0)$	$b_{-0}^{(1)}(1) = b_{-1}^{(1)}(0)$	$b_{-0}^{(2)}(1) = b_{-1}^{(2)}(0)$
$b_{-1}(1) = b_{-2}(0)$	$b_{-1}^{(1)}(1) = b_{-2}^{(1)}(0)$	$b_{-1}^{(2)}(1) = b_{-2}^{(2)}(0)$
$b_{-2}(1) = b_{-3}(0)$	$b_{-2}^{(1)}(1) = b_{-3}^{(1)}(0)$	$b_{-2}^{(2)}(1) = b_{-3}^{(2)}(0)$
$b_{-3}(1) = 0$	$b_{-3}^{(1)}(1) = 0$	$b_{-3}^{(2)}(1) = 0$

where for simplicity each segment is individually parameterized so that $\bar{u} = 0$ corresponds to its left endpoint and $\bar{u} = 1$ corresponds to its right endpoint. These constitute fifteen constraints. We will see that it is convenient to require that

$$b_{-0}(0) + b_{-1}(0) + b_{-2}(0) + b_{-3}(0) = 1$$

Because $b_{-0}(0) = 0$ this simplifies to

$$b_{-1}(0) + b_{-2}(0) + b_{-3}(0) = 1.$$

Because our knots are equally spaced, this amounts to assuming that when we add together an unscaled sequence of basis functions B_i, each of which is a copy of *B* shifted so that its ***support*** (the parameter values for which it is nonzero) begins at u_i, the three basis functions B_{j-3}, B_{j-2} and B_{j-1} that are nonzero at u_j sum to one (Figure 1.12). Such an assumption is said to be a ***normalizing condition*** and serves to define the function $B(u)$ uniquely. Rather miraculously, this normalizing condition will in fact hold at all other values of u as well; that is,

$$b_{-0}(\bar{u}) + b_{-1}(\bar{u}) + b_{-2}(\bar{u}) + b_{-3}(\bar{u}) = 1, \text{ for all } 0 \leq \bar{u} < 1$$

(Notice that our hat functions also summed to one.)

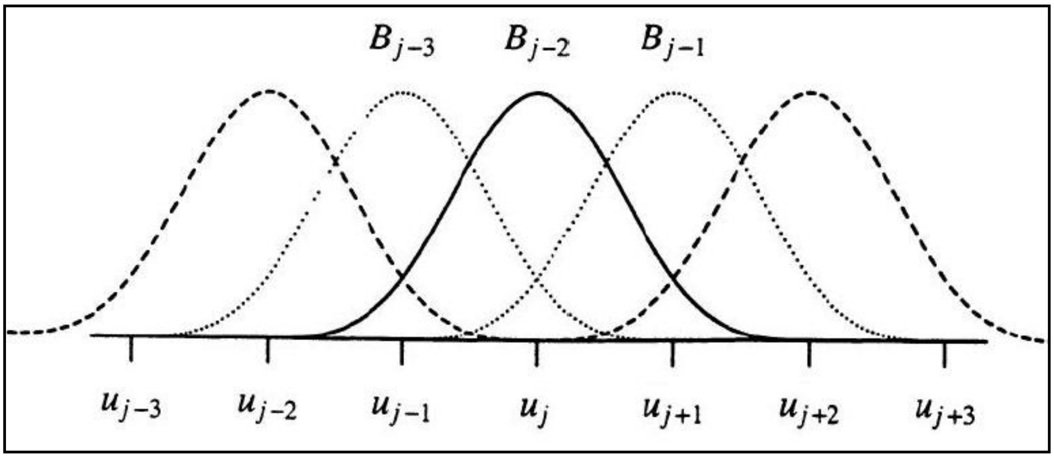


Figure 1.12 The basis functions that are not 0 at u_j are indicated by labels

We now have sixteen equations in sixteen unknowns (that is why we assumed that our basis function had four cubic segments), and we may solve for the coefficients a_j, b_j, c_j and d_j of the four segments b_{-0}, b_{-1}, b_{-2}, and b_{-3} comprising our basis function *B*. Doing so yields the polynomials

Eq. 1.4

$$b_{-0}(\bar{u}) = \frac{1}{6}\,\bar{u}^3$$

$$b_{-1}(\bar{u}) = \frac{1}{6}(1 + 3\bar{u} + 3\bar{u}^2 - 3\bar{u}^3)$$

$$b_{-2}(\bar{u}) = \frac{1}{6}(4 - 6\bar{u}^2 + 3\bar{u}^3)$$

$$b_{-3}(\bar{u}) = \frac{1}{6}(1 - 3\bar{u} + 3\bar{u}^2 - \bar{u}^3)$$

These four segments define the ***uniform cubic B-spline***; again, the term *uniform* means that the knots are equally spaced. The "B" is short for "Basis", which is appropriate because, given a sufficient number of them, they can be used to represent any C^2 spline over a uniform knot sequence. It is easy to verify that these segments have the continuity necessary to qualify them as C^2 splines. Consider, for example, the joint between $b_{-2}(\bar{u})$ and $b_{-3}(\bar{u})$. So far as positional continuity is concerned, we have

$$b_{-2}(1) = b_{-3}(0) = \frac{1}{6}.$$

Consider the first parametric derivative at their common joint. We have

$$b_{-2}^{(1)}(\bar{u}) = \frac{1}{6}(-12\bar{u} + 9\bar{u}^2)$$

$$b_{-3}^{(1)}(\bar{u}) = \frac{1}{6}(-3 + 6\bar{u} - 3\bar{u}^2)$$

and $$b_{-2}^{(1)}(1) = b_{-3}^{(1)}(0) = -\tfrac{1}{2}.$$

Their second parametric derivatives are given by

$$b_{-2}^{(2)}(\bar{u}) = (-2 + 3\bar{u})$$

$$b_{-3}^{(2)}(\bar{u}) = (1 - \bar{u})$$

so that

$$b_{-2}^{(2)}(1) = b_{-3}^{(2)}(0) = 1$$

However,

$$b_{-2}^{(3)}(\bar{u}) = 3$$

$$b_{-3}^{(3)}(\bar{u}) = -1$$

so that their common third parametric derivatives are not equal. Notice that we also have

$$b_{-3}(1) = b_{-3}^{(1)}(1) = b_{-3}^{(2)}(1) = 0$$

Since the basis function (and consequently all its derivatives) are identically zero to the right of $b_{-3}(1)$, we have positional as well as first and second derivative continuity at the right end of $b_{-3}(\bar{u})$ as well.

To determine a curve, we select a set of control vertices $\boldsymbol{P}_i$ and use them to define the curve

Eq. 1.5

$$\boldsymbol{C}(u) = \sum_i \boldsymbol{P}_i B_i(u) = \sum_i (x_i B_i(u), y_i B_i(u))$$

in which each B_i is simply a copy of B, shifted so that its support extends from u_i to u_{i+4}, and the coefficients in the summation are given by the control vertices

$$\boldsymbol{P}_i = (x_i, y_i).$$

Notice that because the basis functions are nonzero on only four successive intervals, if $u_i \leq u \leq u_{i+4}$ then

Eq. 1.6

$$\boldsymbol{C_i}(u) = \sum_{r=-3}^{r=0} \boldsymbol{P}_{i+r} B_{i+r}(u)$$
$$= \boldsymbol{P}_{i-3} B_{i-3}(u) + \boldsymbol{P}_{i-2} B_{i-2}(u) + \boldsymbol{P}_{i-1} B_{i-1}(u) + \boldsymbol{P}_{i-0} B_{i-0}(u).$$

If we replace each basis function$B_j(u)$ by the particular segment that pertains to the interval$[u_i, u_{i+1})$, then (Eq. 1.6) can be written as

Eq. 1.7

$$\boldsymbol{C_i}(\bar{u}) = \sum_{r=-3}^{r=0} \boldsymbol{P}_{i+r} b_r(\bar{u}) = \boldsymbol{P}_{i-3} b_{-3}(\bar{u}) + \boldsymbol{P}_{i-2} b_{-2}(\bar{u}) + \boldsymbol{P}_{i-1} b_{-1}(\bar{u}) + \boldsymbol{P}_{i-0} b_{-0}(\bar{u}).$$

Notice that the segments of our basis function are numbered from right to left because that is the order in which they appear when summed to form a curve: the leftmost control vertex scales the rightmost basis segment, and so on. Eq. 1.7 also reflects the convenience of parameterizing each basis segment from $\bar{u} = 0$ at its left end; since the basis functions are all translates of one another, this convention allows us to use the same formulas in defining each basis function, and hence in computing each curve segment.

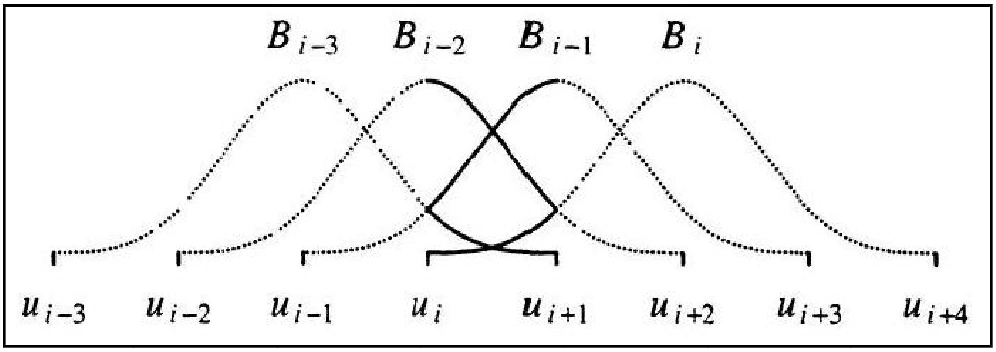


Figure 1.13 The four uniform cubic B-splines that are nonzero on the ith interval $[u_i, u_{i+1})$.

1.3 THE CONVEX HULL PROPERTY

It is a consequence of the way in which we have constructed the B_i (specifically, a result of their normalization) that the ith segment of a uniform cubic B-spline curve lies within the convex hull of the vertices $\boldsymbol{P}_{i-3}, \boldsymbol{P}_{i-2}, \boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_i$, as shown in Figure 1.14. (see section 4)

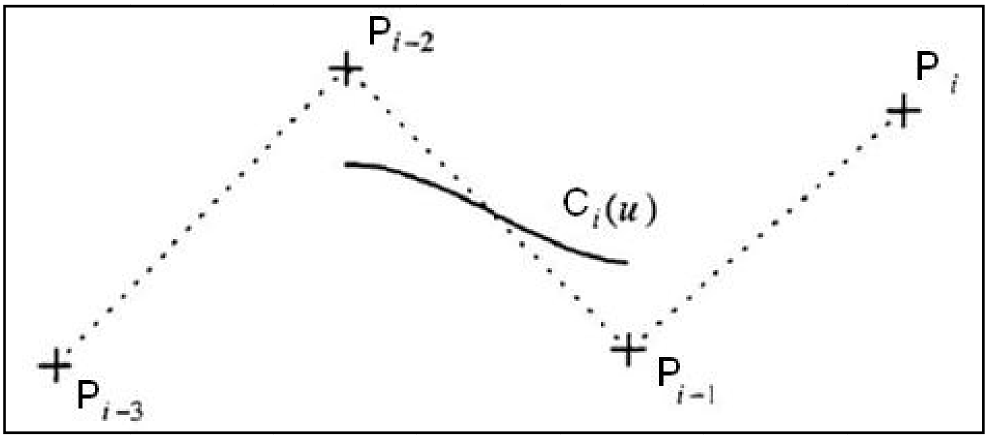


Figure 1.14 The ith segment lies within the convex hull of $\boldsymbol{P}_{i-3}, \boldsymbol{P}_{i-2}, \boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_i$

This is true because, although we only required that the basis functions sum to one at the knots, it is easy to verify directly by summing equations (Eq. 1.4) that

Eq. 1.8

$$\sum_{r=-3}^{r=0} b_r(\bar{u}) = b_{-3}(\bar{u}) + b_{-2}(\bar{u}) + b_{-1}(\bar{u}) + b_{-0}(\bar{u}) = 1, \text{ for all } 0 \leq \bar{u} < 1$$

It follows from equation (Eq. 1.8) and from the fact that the values of the b's are nonnegative that the ith segment of a uniform cubic B-spline curve is created as a convex combination of the control vertices $\boldsymbol{P}_{i-3}, \boldsymbol{P}_{i-2}, \boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_i$, with the values of the basis segments acting as the w_i in the discussion above. Hence, the ith segment lies within the convex hull of these control vertices. Thus if four successive control vertices of such a curve are visible on a display screen, just as we would see any straight line segment within the area surrounded by the vertices, we will also see the spline segment they define. An entire curve "follows" the control vertices in the sense that each successive segment lies within the convex hull of the next group of four control vertices; as we go from one segment to the next, the "oldest" is dropped because it no longer contributes to the curve, and a new vertex is picked up.

It also follows from this discussion that we may consider the B-spline curves as a "parameter dependent, varying convex combination" or "running average" of the control vertices.

1.4 TRANSLATION INVARIANCE

It is highly desirable that translating all the control vertices by the same amount not change the shape of the curve defined. Like the convex hull property, this is an easy consequence of (Eq. 1.8).

Suppose that we translate the control vertices by $\boldsymbol{t} = (dx\,, dy)$. Let $\boldsymbol{C}(u)$ be the curve defined by the control vertices $\boldsymbol{P}_i$, and let $\boldsymbol{C_t}(u)$ be the curve defined by the control vertices $\boldsymbol{P}_i + \boldsymbol{t}$. From (Eq. 1.5) we have

$$\boldsymbol{C_t}(u) = \sum_i (\boldsymbol{P}_i + \boldsymbol{t})B_i(u) = \sum_i \boldsymbol{P}_i B_i(u) + \boldsymbol{t}\sum_i B_i(u)$$

From Eq. 1.8, then, we have

$$\boldsymbol{C_t}(u) = \sum_i \boldsymbol{P}_i B_i(u) + \boldsymbol{t} = \boldsymbol{C}(u) + \boldsymbol{t}$$

Thus, we may either translate the control vertices and then compute the curve they define, or compute the curve first and then translate the points lying on it - the result is the same.

1.5 ROTATION AND SCALING INVARIANCE

It is also important that we be able to rotate a curve without changing its shape. Suppose that we rotate the control vertices by some angle θ. Let **R** be the matrix accomplishing this rotation. Again $\boldsymbol{C}(u)$ is the curve defined by the control vertices $\boldsymbol{P}_i$, and let $\boldsymbol{C_r}(u)$ be the curve defined by the control vertices $\mathbf{R}\boldsymbol{P}_i$. From (Eq. 1.5) we have

$$\boldsymbol{C_r}(u) = \sum_i \mathbf{R}\boldsymbol{P}_i B_i(u)$$

Since for any matrix **M** and vectors **a** and **b**, **Ma+Mb = M(a+b)**, we have

$$\boldsymbol{C}_r(u) = \mathbf{R}\sum_i \boldsymbol{P}_i B_i(u) = \mathbf{R}\boldsymbol{C}(u)$$

Thus we may either rotate the control vertices and then compute the curve they define, or compute the curve first and then rotate the points lying on it - the result is the same.

Since scaling can be represented as a matrix operation, a similar argument establishes that the shape of a cubic B-spline curve is not affected by scaling the control vertices - the same curve is obtained if we scale points on the curve instead.

1.6 END CONDITIONS FOR CURVES

1.6.1 CURVATURE

One of the properties in which we are interested is curvature: whether, or how much, a curve "bends" at some point. Curvature is defined quantitatively in the following way.

At a given point $\boldsymbol{P}$ on a parametrically defined curve $\boldsymbol{C}(u)$, the circle that has the same first and second derivative vectors as the curve is called the ***osculating circle***. The centre and radius of this circle are called the ***centre of curvature*** $c(u)$ and the ***radius of curvature*** $\rho(u)$, respectively, at this point; the ***curvature*** $\kappa(u)$ at this point is the reciprocal, $1/\rho(u)$ of the radius of curvature. Thus if the osculating circle has a large radius, the curvature is small, as our intuition tells us. The ***curvature vector*** $\boldsymbol{K}(u)$ has a magnitude equal to the curvature and points from $\boldsymbol{P}$ towards the centre of curvature.

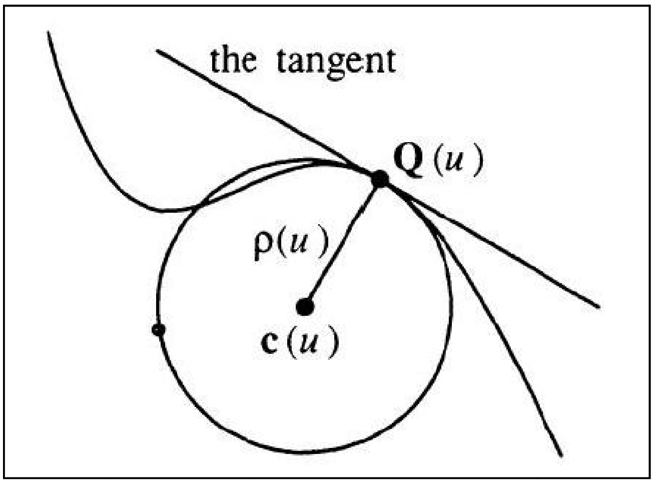


Figure 1.15 The osculating circle

With the use of a bit of differential geometry it is possible to show that the vector

$$\frac{\boldsymbol{C}^{(1)}(u) \times \boldsymbol{C}^{(2)}(u)}{|\boldsymbol{C}^{(1)}(u)|^3}$$

has magnitude equal to the curvature. However, this vector is perpendicular to the plane containing the osculating circle (the ***osculating plane***). An additional cross-product with

$$\frac{\boldsymbol{C}^{(1)}(u)}{|\boldsymbol{C}^{(1)}(u)|}$$

results in a vector of the same length lying in the osculating plane, which is the curvature vector:

Eq. 1.9

$$\boldsymbol{K}(\bar{u}) = \frac{\left(\boldsymbol{C}^{(1)}(u) \times \boldsymbol{C}^{(2)}(u)\right) \times \boldsymbol{C}^{(1)}(u)}{|\boldsymbol{C}^{(1)}(u)|^4}$$

From Eq. 1.9 it follows that:

- if the second derivative vector is zero, then the curvature is zero;
- if the first and second derivative vectors are nonzero but linearly dependent (collinear), then the curvature is zero;
- if the first and second derivative vectors are linearly independent (not collinear), then the curvature is nonzero.

1.6.2 DOUBLE VERTICES

Suppose that we double the first and last vertices. That is, the user specifies the sequence of *m + 1* vertices $\boldsymbol{P}_0, \boldsymbol{P}_1, \dots, \boldsymbol{P}_{m-1}, \boldsymbol{P}_m$, but we actually compute a curve of *m* segments from the sequence of *m + 3* vertices $\boldsymbol{P}_0, \boldsymbol{P}_0, \boldsymbol{P}_1, \dots, \boldsymbol{P}_{m-1}, \boldsymbol{P}_m, \boldsymbol{P}_m$. By adding a vertex to each end of the curve, we add an additional segment to each end as well. The new segments have the form

Eq. 1.10

$$\boldsymbol{C}_{\boldsymbol{2}}(\bar{u}) = \boldsymbol{P}_0[b_{-3}(\bar{u}) + b_{-2}(\bar{u})] + \boldsymbol{P}_1 b_{-1}(\bar{u}) + \boldsymbol{P}_2 b_{-0}(\bar{u})$$

Eq. 1.11

$$\boldsymbol{C}_{\boldsymbol{m+1}}(\bar{u}) = \boldsymbol{P}_{m-2} b_{-3}(\bar{u}) + \boldsymbol{P}_{m-1} b_{-2}(\bar{u}) + \boldsymbol{P}_m[b_{-1}(\bar{u}) + b_{-0}(\bar{u})]$$

If we evaluate these at $\bar{u} = 0$ and $\bar{u} = 1$, respectively, to obtain the first and last points on the curve, we find that

$$\boldsymbol{P_s} = \boldsymbol{C_2}(0) = \frac{1}{6}(5\boldsymbol{P}_0 + \boldsymbol{P}_1) = \left(1 - \frac{1}{6}\right)\boldsymbol{P}_0 + \frac{1}{6}\boldsymbol{P}_1$$

$$\boldsymbol{P_e} = \boldsymbol{C_{m+1}}(1) = \frac{1}{6}(\boldsymbol{P}_{m-1} + 5\boldsymbol{P}_m) = \frac{1}{6}\boldsymbol{P}_{m-1} + \left(1 - \frac{1}{6}\right)\boldsymbol{P}_m$$

Thus the curve begins at a point $\boldsymbol{P_s}$ that is one-sixth of the way from $\boldsymbol{P}_0$ to $\boldsymbol{P}_1$ and ends at a point $\boldsymbol{P_e}$ that is one-sixth of the way from $\boldsymbol{P}_m$ to $\boldsymbol{P}_{m-1}$. Differentiating (Eq. 1.10) and (Eq. 1.11) and then evaluating at $\bar{u} = 0$ and $\bar{u} = 1$, we find the first derivative vectors at $\boldsymbol{P_s}$ and $\boldsymbol{P_e}$ to be

$$\boldsymbol{C}_2^{(1)}(0) = \frac{1}{2}(\boldsymbol{P}_1 - \boldsymbol{P}_0)$$

$$\boldsymbol{C}_{m+1}^{(1)}(1) = \frac{1}{2}(\boldsymbol{P}_m - \boldsymbol{P}_{m-1})$$

Thus the curve is tangent at its endpoints to the first and last line segments of the control graph, as shown in Figure 1.16. If we compute the second derivative vectors at $\boldsymbol{P_s}$ and $\boldsymbol{P_e}$, we find that they are collinear with the tangent vectors, so that the curvature at $\boldsymbol{P_s}$ and $\boldsymbol{P_e}$ is 0.

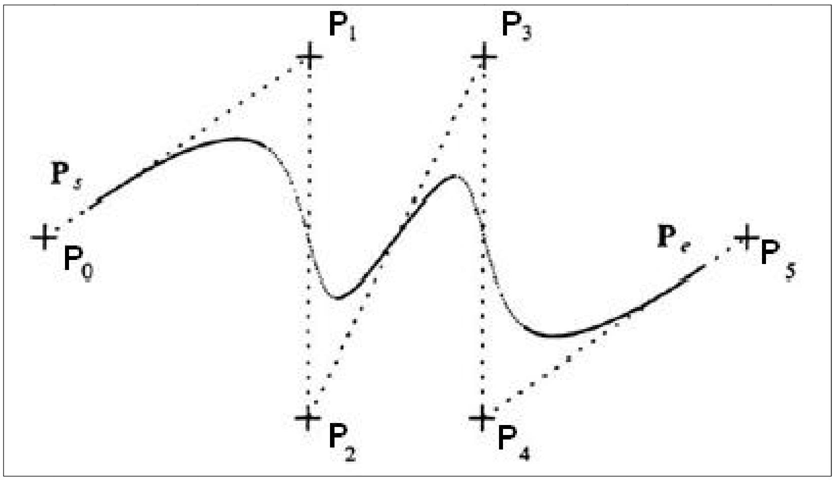


Figure 1.16 P_0 and P_5 are double vertices

1.7 TRIPLE VERTICES

Suppose instead that we now triple the first and last vertices, so that the curve is computed from the *m* +5 vertices $\boldsymbol{P}_0, \boldsymbol{P}_0, \boldsymbol{P}_0, \boldsymbol{P}_1, \dots, \boldsymbol{P}_{m-1}, \boldsymbol{P}_m, \boldsymbol{P}_m, \boldsymbol{P}_m$. This adds two additional segments

Eq. 1.12

$$\boldsymbol{C_1}(\bar{u}) = \boldsymbol{P}_0[b_{-3}(\bar{u}) + b_{-2}(\bar{u}) + b_{-1}(\bar{u})] + \boldsymbol{P}_1 b_{-0}(\bar{u})$$

$$\boldsymbol{C_2}(\bar{u}) = \boldsymbol{P}_0[b_{-3}(\bar{u}) + b_{-2}(\bar{u})] + \boldsymbol{P}_1 b_{-1}(\bar{u}) + \boldsymbol{P}_2 b_{-0}(\bar{u})$$

to the beginning of the curve and two additional segments

Eq. 1.13

$$\boldsymbol{C}_{\boldsymbol{m}+\mathbf{1}}(\bar{u}) = \boldsymbol{P}_{m-2}b_{-3}(\bar{u}) + \boldsymbol{P}_{m-1}b_{-2}(\bar{u}) + \boldsymbol{P}_{m}[b_{-1}(\bar{u}) + b_{-0}(\bar{u})]$$

$$\boldsymbol{C}_{\boldsymbol{m}+\mathbf{2}}(\bar{u}) = \boldsymbol{P}_{m-1}b_{-3}(\bar{u}) + \boldsymbol{P}_{m}[b_{-2}(\bar{u}) + b_{-1}(\bar{u}) + b_{-0}(\bar{u})]$$

to the end of the curve. If we now substitute in Eq. 1.4 and evaluate $\boldsymbol{C}_{\mathbf{2}}(0)$ and $\boldsymbol{C}_{\boldsymbol{m}+\mathbf{3}}(1)$ we find that

$$\boldsymbol{P}_{\boldsymbol{s}} = \boldsymbol{C}_{\mathbf{1}}(0) = \boldsymbol{P}_{\mathbf{0}}$$

and $$\boldsymbol{P}_{\boldsymbol{e}} = \boldsymbol{C}_{\boldsymbol{m}+\mathbf{2}}(1) = \boldsymbol{P}_{\boldsymbol{m}}$$

That is, the curve interpolates the first and last control vertices, as shown in Figure 1.17. The first and last segments of the curve are now straight line segments.

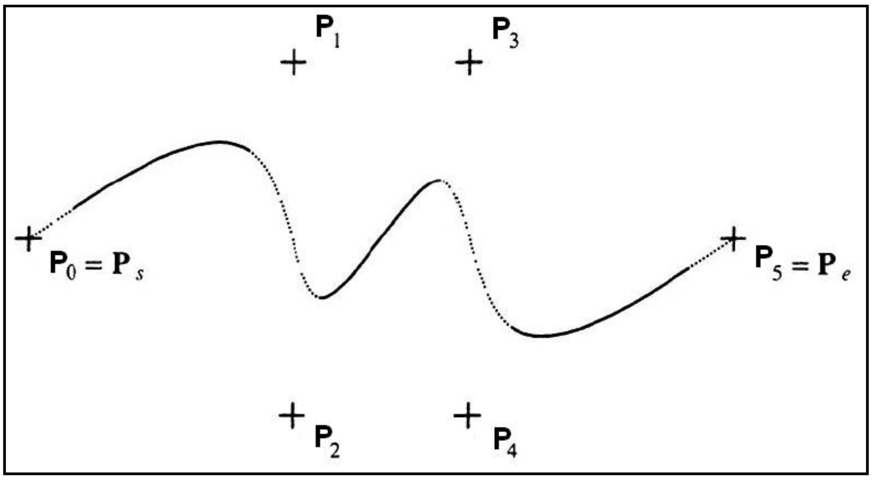


Figure 1.17 $\boldsymbol{P}_{\mathbf{0}}$ **and** $\boldsymbol{P}_{\boldsymbol{m}}$ **are triple vertices, and are interpolated. The control graph has been omitted so that the curve can be seen to reach** $\boldsymbol{P}_{\mathbf{0}}$ and $\boldsymbol{P}_{\boldsymbol{m}}$.

We can verify this easily by simplifying (Eq. 1.12) and (Eq. 1.13). The equation that results for the first segment is

$$\boldsymbol{C}_{\mathbf{1}}(\bar{u}) = \left[1 - \frac{\bar{u}^3}{6}\right]\boldsymbol{P}_0 + \left[\frac{\bar{u}^3}{6}\right]\boldsymbol{P}_1$$

or

$$\boldsymbol{C}_{\mathbf{1}}(s) = (1 - s)\boldsymbol{P}_0 + s\boldsymbol{P}_1$$

for $s = \frac{1}{6}\bar{u}^3$, which we recognize as the equation of a line. The last segment of the curve is, analogously,

$$\boldsymbol{C}_{\boldsymbol{m}+\mathbf{2}}(\bar{u}) = \left[1 - \frac{\bar{u}^3}{6}\right]\boldsymbol{P}_{m-1} + \left[1 - \frac{1 - \bar{u}^3}{6}\right]\boldsymbol{P}_{m}$$

or

$$\boldsymbol{C}_{m+2}(t) = t\boldsymbol{P}_{m-1} + (1-t)\boldsymbol{P}_m$$

for $t = \frac{1}{6}(1-\bar{u}^3)$.

The second segment $\boldsymbol{C}_2(\bar{u})$ and the penultimate segment $\boldsymbol{C}_{m+1}(\bar{u})$ begin and end, respectively, with a double vertex, and so exhibit the behaviour described for double vertices. Thus $\boldsymbol{C}_2(0)$ lies on the line segment from $\boldsymbol{P}_0$ to $\boldsymbol{P}_1$ and the curvature of $\boldsymbol{C}_2(\bar{u})$ is zero at that point, since it has the same first and second derivatives there as $\boldsymbol{C}_1(\bar{u})$, which is a straight line. By the same argument the curvature at $\boldsymbol{C}_{m+1}(1)$ is zero.

1.8 MULTIPLE INTERIOR VERTICES

The analysis of double vertices is equally applicable on the interior of a B-spline curve. Triple interior vertices are particularly interesting. So long as the triple vertex and the vertices immediately preceding and succeeding it fail to be collinear, the left and right derivative vectors at the triple vertex also fail to be collinear; the curve is said to be have a corner.

At first sight this may seem to contradict the fact that the curve is C^2 continuous. The first derivative vector is **0** at the joint and is continuous there. A corner results because the derivative vectors just to the left and right of the joint point in different directions.

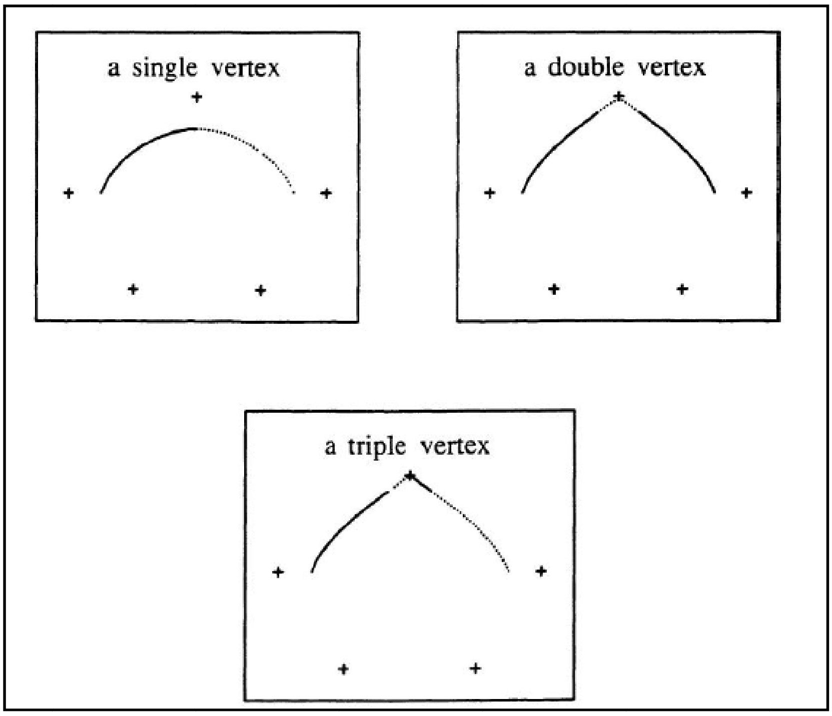


Figure 1.18 Forming a corner by tripling a vertex. The double vertex is not interpolated, while the triple is.

Now turning to B-spline surfaces, if we rewrite Eq. 7.10 in terms of basis segments instead of basis functions we have

Eq. 1.14

$$\boldsymbol{C}_{\boldsymbol{i},\boldsymbol{j}}(\bar{u},\bar{v}) = \sum_{r=-3}^{0}\sum_{s=-3}^{0} \boldsymbol{P}_{i+r,j+s}\, b_r(\bar{u}) b_s(\bar{v})$$

so that $\boldsymbol{C}_{\boldsymbol{i},\boldsymbol{j}}(\bar{u},\bar{v})$, the i,jth patch, is completely determined by sixteen control vertices.

Now let us see how two patches are assembled into a composite surface. Return to Figure 7.14. Adding an additional knot u_8 will create the four additional B-splines shown in Figure 1.19.

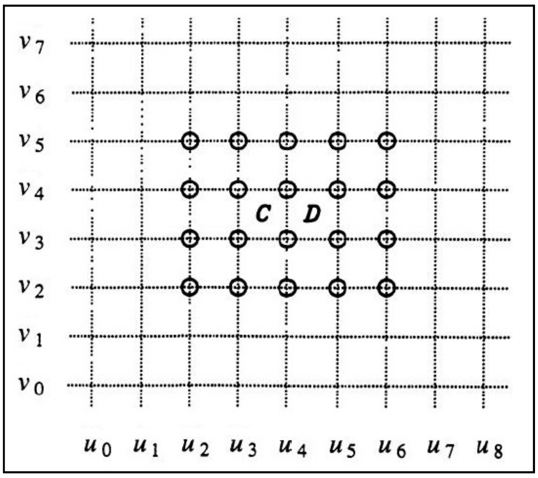


Figure 1.19 The parameter grid which results from adding an additional knot u_8 to the grid of Figure 7.14. There are now four more B-splines, for a total of twenty. The leftmost sixteen B-splines are used to build a patch over parameter region C, while the rightmost sixteen are used to build a patch over parameter region D. (The middle twelve B-splines contribute to both.)

By factoring once the $b_r(\bar{u})$ out of Eq. 1.14 and once the $b_s(\bar{v})$, we easily conclude that the uniform cubic B-spline surface we have defined is C^2 continuous along lines of constant $\bar{u}$ and $\bar{v}$. It follows from elementary calculus that the uniform cubic B-spline surfaces are therefore C^2 continuous in every direction.

PART II

ISOGEOMETRIC ANALYSIS

1 INTRODUCTION TO THE ISOGEOMETRIC METHOD

The need of isogeometric analysis came of the realization that the widely used isoparametric method combined with the conventional basis (Lagrange and Hermite piecewise continuous polynomials) presents a number of inefficiencies during the analysis. Firstly, the FEA method based on conventional basis lacks in numerical stability in some cases (e.g. discontinuous data to interpolate see Figure 1.2). Moreover, due to the advances in the CAGD (NURBS and splines have been used since a long ago) a conversion from the CAD model to an analysis model is required, and at the end the results must be sent back to a computer graphics engine based on CAGD to plot them. This procedure results in usually a non-exact model for analysis and a new approximation procedure to plot the results; especially when conic sections are required which are poorly represented using the conventional basis. Furthermore, any required mesh refinements are not applied locally as easily as it is done while using a geometric basis such as B-Splines or NURBS functions. Lastly, contrary to the Lagrange – Hermite basis, using the isogeometric analysis enables the analyst to apply numerical methods of higher order (e.g. least-squares method), since the continuity of the basis is higher.

The isoparametric method is the outcome of the effort made by Taig and Irons to construct non-rectangular elements and elements with curved edges. According to the isoparametric analysis, we set an appropriate basis to approximate the unknown solution fields and then use that basis to approximate the unknown geometry as well. That enables us to easily use accurate numerical integration methods using a so called parent integration domain and generally form more efficient algorithms. But the main importance of the isoparametric concept is that the three basic convergence conditions (smoothness, continuity, completeness) are virtually automatic.

Now turning to the newborn isogeometric concept or inverse isoparametric concept, one selects a basis capable of exactly representing the geometry and uses it as a basis for the approximation of the solution field with numerical methods. A number of different geometric bases can be used for the isogeometric analysis. The entities described in the first part (Bézier, B-Splines and NURBS) are also included to these bases. The common way in which the bases of these geometric entities are constructed, sets them suitable for the isogeometric analysis.

Some of the fundamental properties of the functions of these geometric bases that are mentioned in the first part of this work and are crucial for the analysis needs are:

- they may constitute the weights of a barycentric combination, $\sum_{i=0}^{n} N_a = 1$, like the combination of points that generates a new point.
- they form convex combinations, $N_i(\xi) \geq 0 \; \forall \xi \in \hat{\Omega}$, where $\hat{\Omega}$ is the parameter domain. The result of such a combination lies in the convex hull of the control variables (convex hull property).
- the derivatives of them sum to zero $\sum_{i=0}^{n} N_i^{(k)} = 0$; due to this property the derivative basis can be used as weights in a combination that generates a vector from a point set (basis for calculating vector fields).
- In particular, as far as the B-Spline basis functions are concerned, they are piecewise functions with local support; thus the approximated solution fields can be locally handled when they are strongly variant.
- The rational basis are suitable for complex shapes representing exactly all the conic sections.

The goal of our analysis is to compute an unknown solution field. This field is usually described as affine mappings of the domain. For example, the theory of linear elasticity problems outlines the displacement field through a sequence of affine transformations (stretches, shears, rotations, translations, see Part I, Section 5) of infinitesimal volumes of the domain. In computational mechanics, once linear basis functions are selected to interpolate the solution field, the kinematics of each element are described by merely using single affine transformations (linear kinematics). In this case each infinitesimal volume considered in linear elasticity theory is simulated by a single finite element. This technique of analysis using linear functions is brief when the element size is large and not small enough to adapt the behaviour of an infinitesimal one. With a higher degree basis functions, a sequence of overlapped affine transformations can be applied to each element, which results a fine analysis procedure. Essentially the overlapped affine mappings are done by more control handles that have under an overlapped control a wider span of the domain. This seems to result a more smooth distribution of the error. In computational geometry, the tools in use (Bezier, BSpline, NURBS entities) are all suitably designed in order to easily and accurately control the commonly used affine mappings concerning the geometry (transition, scaling, rotation, etc), or the overlapped affine transformations that occur when a single control point is displaced and the body must take a shape locally in a more smooth distribution expanding it to a defined region. The fact that the basis has the barycentric property makes it affine invariant and lets us apply the desired affine transformations of the body on the control handles directly. There are only two types of affine transformations of the points, the transition of the points (displacement) and the identity (see Part I, Section 5). The barycentric combination property of the basis is also a property of the conventional (Lagrange, Hermite) basis of the FEA; this lets any affine transformations implemented to the body to be done through the control handles. It seems to generate the isoparametric concept where by an affine transformation of the physical domain to the parametric one anything is done to the first can be mapped to the second and vice versa.

Turning to the second property of the basis functions, the convex combination of the control variables ensures the numerical stability of the analysis procedure. In other words, the solution field lies exclusively in the convex hull that is formed by the control variables. The convex hull property combined with the extended support of the basis functions virtually yields the variation diminishing property which deprives the solution field from suffering from oscillations.

The derivatives of the basis functions that sum to zero yield a combination that constructs vectors using point sets (see Part I, Section 4); this fact that sets them suitable for the computation of a vector field like that of gradients.

2 THE FINITE ELEMENT METHOD THEORY IMPLEMENTED IN ISOGEOMETRIC ANALYSIS METHOD

The analysis problem

Given: the geometry of the body, the8.4 applied loads $f^{S_f}, f^B, R_C^i, i = 1,2, \dots$, the support conditions on S_u , the material stress-strain law, and the initial stresses in the body

Calculate: the displacements $\boldsymbol{U}$ of the body and the corresponding strains ε and stresses σ

This problem could be also described by a differential equation (Laplace's equation). The formulation of this kind is called the **strong form** of this ***boundary value problem***. We are looking for a function $u: \bar{\Omega} \to \mathbb{R}$ such that

Eq. 2.1

$$\Delta u + f = 0 \qquad in\ \Omega$$

Eq. 2.2

$$\begin{aligned} &u = g && on\ \Gamma_D \\ &\nabla u \cdot \boldsymbol{n} = h && on\ \Gamma_N \\ &\beta u + \nabla u \cdot \boldsymbol{n} = r && on\ \Gamma_R \end{aligned}$$

where Δ is the Laplacian differential operator, $\overline{\Gamma_D \cup \Gamma_N \cup \Gamma_R} = \Gamma \equiv \partial\Omega, \Gamma_D \cap \Gamma_N \cap \Gamma_R = \emptyset$, and $\boldsymbol{n}$ is the unit outward normal vector on $\partial\Omega$. The functions $f: \Omega \to \mathbb{R}, g: \Gamma_D \to \mathbb{R}, h: \Gamma_N \to \mathbb{R}, and\ r: \Gamma_R \to \mathbb{R}$, and β are all given. The boundary conditions given in Eq. 2.2 represent the three major types of boundary conditions that could be encountered. These are Dirichlet conditions, Neumann conditions and Robin conditions, respectively. In 2D domains it is

Eq. 2.3

$$\nabla u(x,y) = \begin{bmatrix} \dfrac{\partial u(x,y)}{\partial x} \\ \dfrac{\partial u(x,y)}{\partial y} \end{bmatrix}$$

And the directional derivative

Eq. 2.4

$$\nabla u(x,y) \cdot n(x,y) = \frac{\partial u(x,y)}{\partial n}$$

The governing equations Eq. 2.1 in 2D elasticity problems are

Eq. 2.5

$$\nabla \cdot \sigma + f = 0 \quad in\ \Omega$$

$$\sigma = D\varepsilon$$

$$\varepsilon = \frac{1}{2}(\nabla u + \nabla u^T)$$

ε is the linearized strain tensor and ∇u is the gradient of the displacement field

Eq. 2.6

$$\nabla u(x, y) = \begin{bmatrix} \frac{\partial u_1(x, y)}{\partial x} & \frac{\partial u_1(x, y)}{\partial y} \\ \frac{\partial u_2(x, y)}{\partial x} & \frac{\partial u_2(x, y)}{\partial y} \end{bmatrix}$$

σ is the stress tensor and the divergence of that tensor is a vector whose components are the divergences of the rows of the tensor

Eq. 2.7

$$\nabla \cdot \sigma = \begin{bmatrix} \frac{\partial \sigma_{11}}{\partial x} + \frac{\partial \sigma_{12}}{\partial y} \\ \frac{\partial \sigma_{21}}{\partial x} + \frac{\partial \sigma_{22}}{\partial y} \end{bmatrix}$$

For a sufficiently smooth domain, and under certain restrictions on *g*, *h*, and *r*, a unique solution *u* satisfying Eq. 2.1 and Eq. 2.2 is known to exist, but an analytical expression will usually be impossible to obtain. However, we may seek an approximate solution.

The techniques that we use in order to compute an approximate solution are called ***numerical methods***. Different numerical methods are simply different techniques for finding the control variables d_A which define the solution field, such that $\hat{u}^h \approx u$, where $\hat{u}^h$is the approximate solution function.

Approximate solutions of differential equations satisfy only part of the conditions of the problem: for example the differential equation may be satisfied only at a few positions, rather than at each point. The approximate solution is expanded in a set of known functions with arbitrary parameters. In engineering problems there are two most used methods to determine the parameters: the Method of Weighted Residual and the Variational Method. In the method of weighted residual one works directly with the differential equation and boundary conditions whereas in the variational method one uses a functional related to the differential equation and boundary conditions. In both methods there are two strategies. (1) A first approximation may be sufficient; its validity is assessed using our intuition and experience. Furthermore, insight is often gained from the analytical solution. (2) A sequence of approximations can be calculated to converge to the solution. In the second strategy the calculations must be amenable to a computer; successive approximations must be calculated without any reformulation or intervention by the analyst. The method of weighted residual (often abbreviated MWR) actually encompasses several methods (collocation, Galerkin, integral, etc.) and provides a framework to compare, contrast, and elucidate the features of individual methods. Variational methods that are usually derived from a stationary principle of energy, are not applicable to all problems, and thus suffer a lack of generality. Sometimes they provide powerful results, such as upper and lower bounds on quantities of interest. MWR is easy to apply, whereas variational methods require more mathematical manipulation. Nevertheless, when variational principles are applicable they are equivalent to the weighted residual methods. (Finlayson, 1972)

2.1 WEIGHTED RESIDUAL METHODS – GALERKIN'S METHOD

We present the Galerkin's numerical method for the approximation of the solution to the boundary value problem. This method belongs to the more general class of weighted residual methods. The method of weighted residual is a tool for finding approximate solutions to the equations of change of distributed systems. It is also applicable to nonlinear and non-self-adjoint problems, which is one of its most attractive features. The weighted residual method is applied among other problems, to the boundary value problems, initial value problems and eigenvalue problems.

The more general time dependent problem is: Given a system of differential or integro-differential equations of change and constitutive relations, the boundary conditions representing the interactions between the system and its surroundings, and the initial conditions (in time dependent problems) representing some base state of interest, the general approach is to assume a trial solution whose functional dependence on position is chosen, but which includes undetermined functions of time $\alpha_i(t)$. The latter are found by requiring that the trial solution satisfy the differential equation in some specified approximate sense.

2.1.1 DETAILS IN THE WEIGHTED RESIDUAL METHODS

In these methods, as a first step it is assumed a trial function which is a combination of n basis (or shape or approximating) functions using the unknown coefficients α_i (dependent or independent of time) to be optimized in a later step for a better approximation of the solution. Expressed by

Eq. 2.8

$$\bar{u}(x, a_i) = \varphi_{\Gamma_D}(x) + \sum_{i=1}^{n} a_i \varphi_i(x)$$

In elasticity, shape functions $\varphi_i(x)$ indicate specific shapes that the domain can be formed to; obeying the boundary conditions, but with the request that they should form a basis (independent shapes).

The basis functions are chosen such that the essential boundary conditions are satisfied.

Eq. 2.9

$$\bar{u}(x, a_i) = g, \qquad \varphi_{\Gamma_D}(x) = g, \qquad \varphi_i(x) = 0, \qquad x \in \Gamma_D$$

Such that the trial function could satisfy the boundary conditions for all the coefficients α_i. It is not necessary that the trial solution be linear in the α_i, but such a choice is usually made for simplicity.

In order to improve the approximate solution, we can add more terms to the selected trial function, such that the basis to be enriched with more independent potential shape functions. We need the same number of basis functions as that of unknown coefficients α_i so that the coefficients can be determined properly. In general, accuracy of an approximated solution is dependent upon proper selection of the trial function.

Once a trial function is selected, the residual can be formed by substituting the trial function into the differential equation. The residual is a function that measures the extent to which the trial function does not satisfy the differential equation at each point. Thus, the residual R becomes

Eq. 2.10

$$R(x, a_i) = \Delta\bar{u}(x, a_i) + f(x)$$

because $\bar{u}$ is different from the exact solution, the residual does not vanish for all values of $x \in \Omega$, see Figure 2.1. However, we state at this point that a vanishing total summed residual over the entire domain is a decent case (we require in Figure 2.1 the total hatched area above the curve of the exact solution equals to the total hatched area below the curve of the exact solution). The technics that can accomplish this request are the weighted residual methods. Each of them has its own way to vanish the total summed residual as it is presented in following context. Nevertheless, all these technics are based on one single general concept, which is at each point of the domain the residual is weighted, in order to vanish the total sum of it. This distribution of weights inside the domain is expressed by a weighting function (or test function or variation function or perturbation function). Mathematically, in a continuous domain, these methods are expressed by

Eq. 2.11

$$I = \langle w(x), R(x, a_i) \rangle = \int_0^1 w(x) \cdot R(x) dx = 0$$

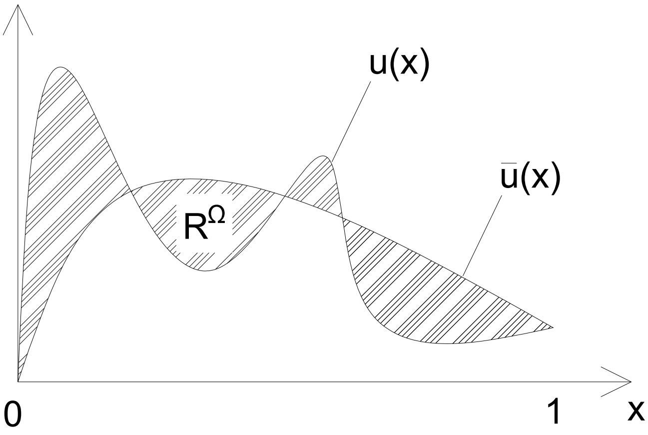


Figure 2.1 A graphical representation of the residual at each point of the domain (the distance of the two curves) and the total residual that is the hatched area

If we assume that each point of the domain can take an arbitrary value of the field that we are seeking, then each single point can be considered as a single independent variable. The amount of these independent variables in a continuous domain is infinite. This case can be expressed by Eq. 2.8, which should be a series of infinite terms (shape functions) of dirac δ functions (see Figure 2.2).

Eq. 2.12

$$\bar{u}(x, a_i) = \varphi_{\Gamma_D}(x) + \sum_{i=1}^{\infty} a_i \delta_i(x)$$

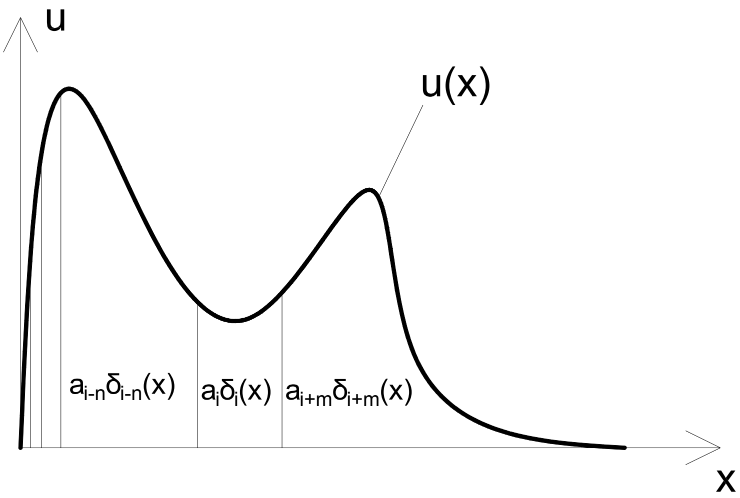


Figure 2.2 The exact solution is formed by an infinite number of dirac shape functions

Thus, as independent variables they can form a basis of an infinite vector space. A linear vector space that contains every possible solution (or equilibrium state) and can be formed by the linear combination of its basis, is called a complete space. This means that there is no need for any extra dimension (additional independent variables, thus a new space) to express any equilibrium state. The corresponding weighting functions should be components of an equal sized space as is the space of the shape functions because every single point of the domain is assumed to act independently. In that interpretation of the trial and weighting functions, in terms of linear algebra, Eq. 2.11 can be directly expressed as an internal product of two vectors with infinite components in an inner product space such as $\mathbb{R}^n \ or \ L^2(\Omega)$.

As mentioned before, the requirement of such an inner product in weighted residual methods is the inner product to vanish; consequently based on the notion of orthogonality, they should become orthogonal. This requirement is accomplished by two cases: a) the inner product of every non-zero weighting function with the residual have a projection that is zero in the current inner product space and non-zero in a new space with more dimensions b) the residual function vanishes at each point of the entire domain. If the trial function is expressed by linear combination of an infinite number of dirac δ_i functions then it is the exact solution function. Such functions as mentioned before are members of a complete vector space. Thus, the requirement of the vanishing inner product in these spaces is satisfied by the case b) stated above.

We return now to the trial functions in the approximation technics that the number of the shape functions is finite and each of them relates strongly all the points in the domain. An example of such shape functions can be seen in figure below

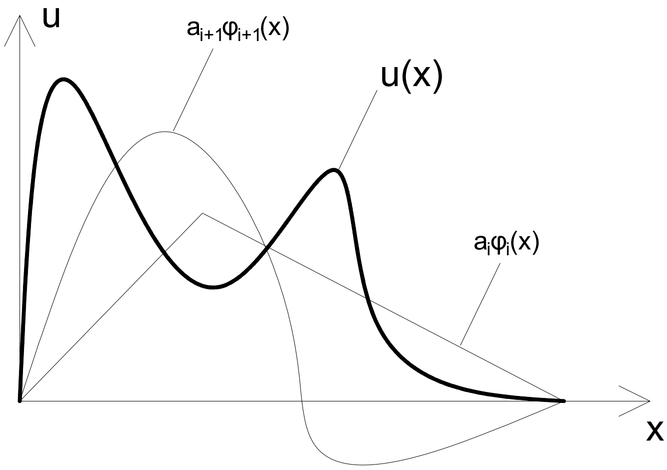


Figure 2.3 Arbitrary independent shape functions linearly combined by coefficients a_i. Each of the shape function relates strongly all the points in the domain

This case where is considered a finite number of independent shape functions is only a finite subspace of the complete space; each shape function is an independent variable component of the subspace basis. The same consideration is made for the weighting functions in order to solve a finite number of vanishing equations for the coefficients a_i. This is the reason that this finite set of shape functions cannot represent exactly the solution function; except the cases where the solution is simple enough (e.g. linear type) then the solution can be expressed by a finite set of shape functions. This means that the vanished inner product of the two subspaces in the current inner product subspace cannot be assured that it vanishes in all other dimensions of the complete inner product space that can form the exact solution.

The next step is to compute the unknown coefficients α_i, so that the chosen trial function best approximates the exact solution. For this process we need as many weighting functions as the unknown coefficients; each of them will render a vanished total summed residual for specific values of the unknown coefficients. The more the independent occasions that the total summed residual is vanished the better the approximate solution. Since the number of the unknowns expresses usually the potential of variability in the domain, more alternative weighting distributions are required for large variations. These weighting functions will assume a distribution of weights each of them in an independent way. In other words a distribution of perturbations of the trial function is defined by each weighting function. Once the system of these equations is solved the trial function obtains a form for which each equation of the set of the total weighted summed residuals vanishes. Obviously, the quality of the approximation is dependent not only on the shape functions but also on the way that the weighting function is perturbing the residual and implicitly how it perturbs the approximate solution. In elasticity problems, the weighting function is perturbing the structure in different ways to make a sense how it works (reacts), the more testing that can be made by the weighting functions the best approximation rendered for the actual behaviour of the structure locally. In other words using only a subspace of the weighting functions and a subspace of the shape functions the global equilibrium is satisfied but not the equilibrium of each point (same as in variational methods).

Assuming a finite subspace of weighting functions V^n and a finite subspace of shape functions S^n there can be written n vanishing equations

Eq. 2.13

$$w_i(x) \cdot R(x, a_i) = 0$$

In the complete inner product space S the weighted residual of the approximate solution is

Eq. 2.14

$$R^{w}(x) = u_{exc}(x) - \bar{u}(x)$$

For the sake of geometrical interpretation we assume a three dimensional complete space (xyz) and a two dimensional subspace (xy) of the approximate solutions. Then one can observe in Figure 2.4 that the approximate solution weighted by every weighting function is only an approximation to the exact solution. It is also noticeable that there is one optimal approximate solution (the left pictured one) which possesses the minimum possible residual (the projection of the exact solution to the subspace). This optimum approximation is the Galerkin's method type of the weighted residuals methods. The characteristic of that method, as one can observe in the figure, is that each component of the residual is orthogonal to each component of the solution subspace and consequently orthogonal to the components of the approximate solution (cannot be further optimized).

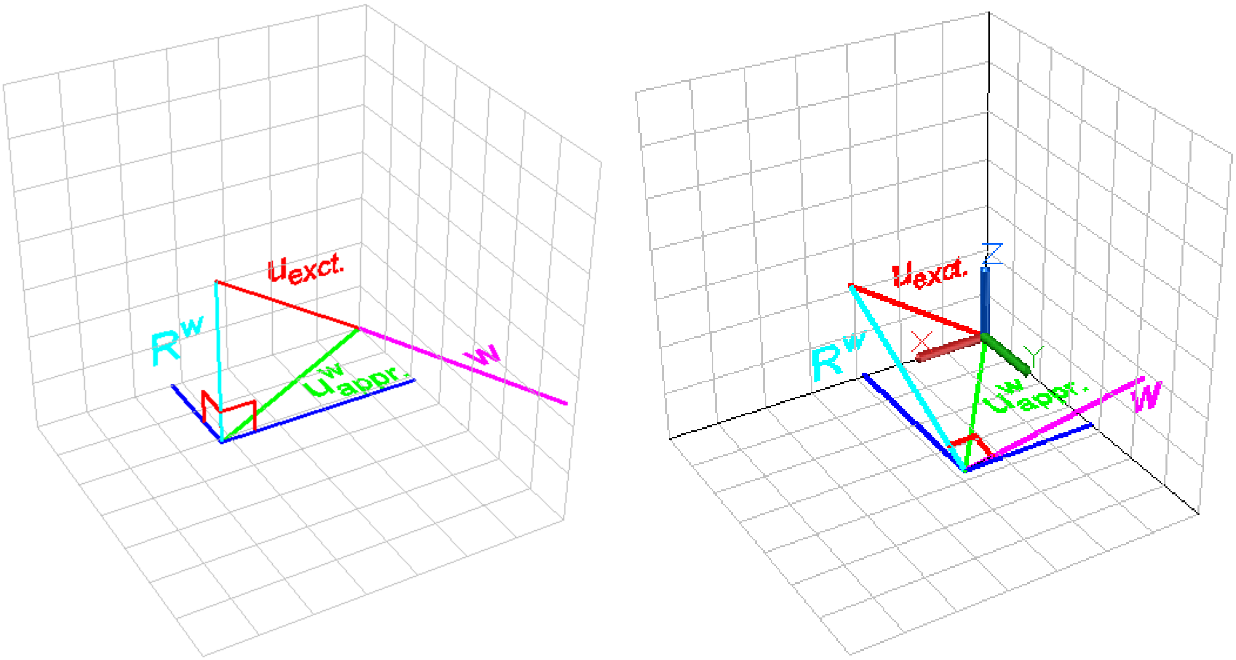


Figure 2.4 A simplified geometrical representation of the weighted residual methods

$$w_1 {R_1}^{w} + w_2 {R_2}^{w} = 0$$

It is clear that the resultant approximate solution differs depending on the weighting function. The weighting functions can be chosen in many ways and each choice corresponds to a different criterion of methods of weighted residual. These methods can be classified based on how the weighting function is determined. Some of the methods of weighted residual are explained below.

1. Collocation Method. The displaced Dirac delta function, $\delta(x - x_i)$, is used as the weighting function, where the sampling point x_i must be within the domain, $0 < x_i < 1$. In other words,

Eq. 2.15

$$w_i = \delta(x - x_i)$$

This method is just limited to vanish the residual in the sampling points without any care for the rest of the domain.

Eq. 2.16

$$\int_V w_j R \, dV = R|_{x_j}$$

As the number of sampling points increases the residual is zero at more and more points and presumably approaches zero everywhere.

2. Subdomains Method. The domain Ω is divided into N smaller sybdomains Ω_j and as weighting functions are chosen

Eq. 2.17

$$w_j = \begin{cases} 1, & x \in \Omega_j \\ 0, & x \neg\in \Omega_j \end{cases}$$

The weighed differential equation is integrated over a subdomain and set to zero. As N increases, the differential equation is satisfied on the average in smaller and smaller subdomains and presumably approaches zero everywhere. Lanczos expanded the solution in terms of Chebyshev polynomials, and used the roots to a Chebyshev polynomial as the collocation points. Lanczos method has been revived as the orthogonal collocation method which has proved to be very suitable for nonlinear problems.

3. Least Squares Method. The least squares method was originated by Gauss in 1795 for least squares estimation. The same ideas are published by Legendre in 1806. The weighting function is determined from the residual such that

Eq. 2.18

$$w_i = \frac{dR}{d(\alpha_i)} \quad thus \quad I(a_i) = \langle w(x), R(x, a_i) \rangle = \int_V R^2(a_i, x) dV$$

This form vanishes the total residual and minimizes by variations the square of the distance between the trial solution and the exact. Essentially the residual is projected to itself and the magnitude of the error field is set to zero. The mean square residual has theoretical significance since error bounds can be derived in terms of it. Thus, minimization of this integral gives the best possible bounds for the error.

4. Galerkin's Method. For Galerkin's method, the weighting function comes from the chosen trial function. That is,

Eq. 2.19

$$w_i = \frac{d\bar{u}}{d(\alpha_i)}$$

In this method the weighting functions are chosen to be the shape functions $\varphi_i(x)$. This method developed by the Russian mathematician Galerkin in 1915 (contribution by Bubnov) as the first criterion of what is known as the method of weighted residuals. ***The shape functions must be chosen as members of a complete set of functions***. The Galerkin method then can be interpreted as making the residual orthogonal to members of the complete set. In the approximation scheme outlined above, the residual is continuous (depending upon the differential operator and the choice of shape functions), and hence the residual can vanish only if it is orthogonal to each member of a complete system of functions. Of course in practice the residual is made orthogonal to no more than a modest, finite number of the members of a complete set (see Figure 2.4). Many similar to the original Galerkin technique are often referred to as the Galerkin or generalized Galerkin method that is:

- The one in which the coefficients are given as functions of time $a_i = a_i(t)$ for time dependent problems.
- The one in which trial solutions are of the more general form $\bar{u} = f(\boldsymbol{x}, \{a_i\})$
- The one in which weighting functions are of the form $K(\varphi_i)$, rather than the shape functions u_i, where K is a specified differential operator.

The least squares method produces a symmetric matrix regardless of a chosen trial function. However, Galerkin's method may produce a symmetric matrix under certain conditions, as will be explained next (weak formulations).

The formulation described is called the strong formulation of the weighted residual method. This includes the highest order of derivative term in the differential equation. The integral must have a non-zero finite value to yield a meaningful approximate solution to the differential equation. This means a trial function should be differentiable twice and its second derivative should not vanish. In order to reduce the requirement for a trial function in terms of order of differentiability, integration by parts is applied to the strong formulation. With that intent we form the weak statement of the problem which is derived from Eq. 2.11 and Eq. 2.1 integrating by parts.

2.1.2 THE WEAK FORMULATION OF THE ANALYSIS PROBLEM

Input : $f, g, h \; and \; r$

Output : $u \in S$ such that $\forall w \in V$

Eq. 2.20

$$\int_{\Omega} \nabla w \cdot \nabla u \, d\Omega + \beta \int_{\Gamma_R} wu \, d\Gamma = \int_{\Omega} wf \, d\Omega + \int_{\Gamma_N} wh \, d\Gamma + \int_{\Gamma_R} wr \, d\Gamma$$

Despite the fact that the strong form of the equation Eq. 2.1 required u to have well defined second derivatives, the weak form from which the numerical method is built only requires that first derivatives be square-integrable. Under appropriate regularity assumptions it can be shown that the weak solution and the strong solution are equivalent. Weak formulation has an advantage for Galerkin's method where weighting functions are obtained directly from the selected trial function, as is mentioned already introducing Eq. 2.19. If a governing differential equation is the self-adjoint, that is when the weak form can be written by the differential operators $D^a u$, $D^a w$, Galerkin's method along with the weak formulation, results in the variational form of the problem and a symmetric matrix in terms of unknown coefficients of the trial function. In this frame some useful definitions from functional analysis are presented below.

- The space of square integrable functions on Ω

Eq. 2.21

$$L^2(\Omega) = \{u | u: \Omega \to \mathbb{R}, \int_{\Omega} u^2 \, d\Omega < +\infty\}$$

- Sobolev space

Eq. 2.22

$$H^1(\Omega) = \{u | D^{\boldsymbol{a}} u \in L^2(\Omega), |\boldsymbol{a}| \leq 1\}, \boldsymbol{a} = \{a_1, \dots, a_d\}, |\boldsymbol{a}| = \sum_{i=1}^{d} a_i$$

$$D^{\boldsymbol{a}} = D_1^{a_1} D_2^{a_2} \dots D_d^{a_d}, \qquad D_i^j = \frac{\partial^j}{\partial x_j^i}$$

We can now define a collection of ***trial solutions*** to our problem as

Eq. 2.23

$$S = \{u | u \in H^1(\Omega), u|_{\Gamma_D} = g\}$$

and a collection of ***weighting functions*** (space of variations)

Eq. 2.24

$$V = \{w | w \in H^1(\Omega), w|_{\Gamma_D} = 0\}$$

The weighting (or test or variations) functions have to be zero on the essential boundaries as the values of the solution cannot be perturbed there, basis (or shape) functions and the trial function itself must be well defined and satisfy $u(x) = \varphi(x) = g,\ x \in \Gamma_D$.

2.1.3 PIECEWISE BASIS FUNCTIONS AND DISCRETIZATION OF THE DOMAIN IN CLASSICAL FEA

Regardless of the weak or strong formulation, the accuracy of an approximate solution depends greatly on the chosen trial function. However, assuming a proper trial function for the unknown exact solution is not an easy task. This is especially true when the unknown exact solution is expected to have a large variation over the problem domain, the domain has a complex shape, and/or the problem has complicated boundary conditions. In order to overcome these problems, a trial function can be described using piecewise continuous shape functions (functions with local support). The selection of trial functions that are linear combinations of piecewise continuous basis functions comprise an advantage of the weak formulation over the strong formulation in Galerkin's method. There can be used a basis of piecewise shape functions that are smooth in the interior C^∞ and weakly continuous at the ends satisfying the existence of the integral. Furthermore, the residual is optimum in each subdomain for a given shape functions basis; this is achieved by controlling the total trial function locally rather than in global sense and simultaneously vanishing the residual in each node. In the figure below it is shown the occasion of a trial function combining two linear hat functions.

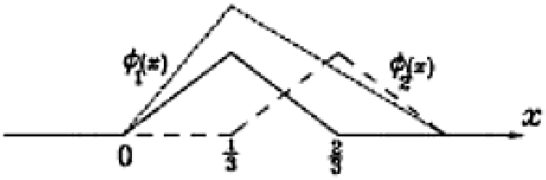


Figure 2.5 A trial function as a combination of two linear piecewise continuous test functions

To build a trial function with piecewise functions, the domain should be divided into subdomains by some nodes as is shown in figure above. A piecewise polynomial is a function that is defined by a polynomial on each subdomain and is derived usually using interpolation theory. These piecewise shape functions are derived systematically; starting with an assumed degree of algebraic polynomial:

Eq. 2.25

$$\bar{u}(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots = \sum_{i=0}^{n} a_i x^i$$

The assumption that the trial function is interpolatory at some nodes (a desire to handle the trial solution to some nodal values), results in a systematic and discretised formulation consisted of piecewise continuous intepolatory basis functions in terms of the nodal variables d_i; these variables possess values of the exact solution field and are in substance the unknown control variables of the trial function. The discretized form of the trial function is now expressed as

Eq. 2.26

$$u^h(x) = N_1(x)d_1 + N_2(x)d_2 + \cdots + N_n(x)d_n \;\; or \;\; u^h(x) = \sum_{i=1}^{n} N_i d_i$$

The superscript h denotes that the solution is defined on a discretized domain defined by a step length h. The interpolatory piecewise continuous shape functions that results this method are of the so called type of Lagrange polynomials. Now the problem is discretised and the solution field can be approximated by solving a system of equations for the values of the control variables d_i.

Discretizing the solution field by giving the control to a finite number of control variables, a space of potential trial functions is formed by a finite basis(the corresponding to the control variables shape functions N_i). Essentially this space is an approximate subspace of the more general complete solution space S that consists of an infinite number of shape functions. These subspaces consists of basis functions that are also members of the complete solution space; they hold on to the linear space property (any solution can be formed as a linear combination of the basis), and the essential boundary conditions can be easily implied for all the linear combinations, just by constraining the corresponding boundary basis functions. Such a solution space can be denoted as S^h, and as an approximate subspace (for a specific mesh) it is $u^h \in S^h \subset S$ of the more general complete solution space S. Using the weighted residual method of Galerkin type, the corresponding approximate subspace (for a specific mesh) of the weighting functions is $w^h \in V^h \subset V$. In fact the basis of the space of variations V^h is a part of the basis of the trial solutions space S^h, excluding the basis functions $g^h \in S^h$ that correspond to the control variables of the essential boundaries. Consequently, there are only few shape functions that are non-zero at the essential boundaries and must satisfy the corresponding boundary conditions $N_{g_h}(x) = g,\ x \in \Gamma_D$. All the other shape functions are zero on the essential boundaries and can be members of the basis of the weighting functions space, thus $w^h(x) = 0$, $x \in \Gamma_D$. The basis functions $g^h \in S^h$ (lifting functions) are set to be $g^h|_{\Gamma_D} = g,\ g^h|_{\neg\Gamma_D} = 0$.

Thus,

Eq. 2.27

$$S^h = V^h \cup g^h$$

According to the above definitions we can decompose the trial function to a variable part and to a constrained part; the variable part of basis functions is in the weighting space also according to the Galerkin's method , $v^h \in V^h$, so that

Eq. 2.28

$$u^h = v^h + g^h$$

Now we can write the weak form in the way that the variational form would be written considering that the constrained part of the domain is excluded. Here entire the domain is expressed, including the constrained part g^h:

Eq. 2.29

$$\int_{\Omega} \nabla w \cdot \nabla(v^h + g^h)\, d\Omega + \beta \int_{\Gamma_R} w(v^h + g^h)\, d\Gamma = \int_{\Omega} wf\, d\Omega + \int_{\Gamma_N} wh\, d\Gamma + \int_{\Gamma_R} wr\, d\Gamma \rightarrow$$

Eq. 2.30

$$\begin{aligned} &\int_{\Omega} \nabla w \cdot \nabla v^h\, d\Omega + \beta \int_{\Gamma_R} w \cdot v^h\, d\Gamma \\ &\qquad = \int_{\Omega} wf\, d\Omega + \int_{\Gamma_N} wh\, d\Gamma + \int_{\Gamma_R} wr\, d\Gamma - \int_{\Omega} \nabla w \cdot \nabla g^h\, d\Omega - \beta \int_{\Gamma_R} w \cdot g^h\, d\Gamma \end{aligned}$$

2.1.4 THE WEAK FORMULATION OF THE GALERKIN'S METHOD COMBINED WITH PIECEWISE BASIS FUNCTIONS IN S^h

Given: boundary conditions g^h, h, r

Find: approximate solution $u^h = v^h + g^h$, where $v^h \in V^h$

Such that: for arbitrary $w^h \in V^h$

Eq. 2.31

$$a(w^h, u^h) = L(w^h) \iff a(w^h, v^h) = L(w^h) - a(w^h, g^h)$$

where the functional $a(\cdot,\cdot)$ expressed by the total solution basis

Eq. 2.32

$$a(w^h, u^h) = \int_{\Omega} \nabla w \cdot \nabla(v^h + g^h)\, d\Omega + \beta \int_{\Gamma_R} w(v^h + g^h)\, d\Gamma$$

and the functional $a(\cdot,\cdot)$ expressed by the variational solution basis (variational form)

Eq. 2.33

$$a(w^h, v^h) = \int_{\Omega} \nabla w \cdot \nabla v^h\, d\Omega + \beta \int_{\Gamma_R} w \cdot v^h\, d\Gamma$$

Also the known functional consists of the natural conditions

Eq. 2.34

$$L(w^h) = \int_{\Omega} wf\, d\Omega + \int_{\Gamma_N} wh\, d\Gamma + \int_{\Gamma_R} wr\, d\Gamma$$

and the constrained part of the $a(\cdot,\cdot)$ functional

Eq. 2.35

$$a(w^h, g^h) = \int_{\Omega} \nabla w \cdot \nabla g^h\, d\Omega + \beta \int_{\Gamma_R} w \cdot g^h\, d\Gamma$$

In the form of $a(w^h, v^h) = L(w^h) - a(w^h, g^h)$ the unknown information is on the left-hand side, while everything on the right-hand side is given.

Some properties of the linear functionals $a(\cdot,\cdot)$ and $L(\cdot)$ are:

The symmetry of $a(\cdot,\cdot)$ that follows directly from its definition that $a(w, u) = a(u, w)$.

Also $a(\cdot,\cdot)$ is bilinear and $L(\cdot)$ is linear. This is, for all constants C_1 and C_2,

Eq. 2.36

$$a(w, C_1 v + C_2 g) = C_1 a(w, v) + C_2 a(w, g)$$

Eq. 2.37

$$L(C_1 v + C_2 g) = C_1 L(v) + C_2 L(g)$$

2.1.5 MATRIX FORM OF THE WEAK GALERKIN FORMULATION IN S^h

Let the solution space consist of all linear combinations of a given basis functions $N_A : \hat{\Omega} \to \mathbb{R}$, where $A = 1, \dots, n_{np}$. Recall that very few functions are non-zero on the boundary of the domain. Without loss of generality, we may assume a numbering for these functions such that for $n_{eq} < n_{np}$:

Eq. 2.38

$$N_A|_{\Gamma_D} = 0 \; \forall \; A = 1, \dots, n_{eq}$$

which are the unconstrained basis functions and form the variational space V^h. Such that there exist d_A for the linear combination

Eq. 2.39

$$v^h = \sum_{A=1}^{n_{eq}} N_A \, d_A$$

Furthermore, the function g^h is given similarly by coefficients $g_A, A = 1, \dots, n_{np}$. In practice, we will always choose g^h such that $g_1 = \cdots = g_{n_{eq}} = 0$ as they have no effect on its value on Γ_D, and so

Eq. 2.40

$$g^h = \sum_{A=n_{eq}+1}^{n_{np}} N_A \, d_A$$

Recall Eq. 2.28

Eq. 2.41

$$u^h = v^h + g^h$$

Which form the solution space S^h, that for any $u^h \in S^h$

Eq. 2.42

$$u^h = v^h + g^h = \sum_{A=1}^{n_{eq}} N_A \, d_A + \sum_{C=n_{eq}+1}^{n_{np}} N_C \, d_C = \sum_{A=1}^{n_{eq}} N_A \, d_A + g^h$$

Also in this Galerkin weak formulation it is for all $w^h \in V^h$ and for the arbitrary I_A

Eq. 2.43

$$w^h = \sum_{B=1}^{n_{eq}} N_B\, I_B$$

Inserting the Eq. 2.42 and Eq. 2.43 into Eq. 2.31 and take advantage of linearity to obtain the expression

Eq. 2.44

$$a\left(\sum_{B=1}^{n_{eq}} N_B\, I_A, \sum_{A=1}^{n_{eq}} N_A\, d_A\right) = L\left(\sum_{B=1}^{n_{eq}} N_B\, I_B\right) - a\left(\sum_{B=1}^{n_{eq}} N_B\, I_B, g^h\right) \longrightarrow$$

$$\sum_{B=1}^{n_{eq}} I_B \left(\sum_{A=1}^{n_{eq}} a(N_B, N_A) d_A\right) = \sum_{B=1}^{n_{eq}} I_B L(N_B) - \sum_{B=1}^{n_{eq}} I_B a(N_B, g^h) \longrightarrow$$

$$\sum_{B=1}^{n_{eq}} I_A \left(\sum_{A=1}^{n_{eq}} a(N_B, N_A) d_A\right) = \sum_{B=1}^{n_{eq}} I_B L(N_B) - \sum_{B=1}^{n_{eq}} I_B a(N_B, g^h)$$

As the I_A are arbitrary (to hold for all $w^h \in V^h$) we can set them as identity matrices or factorize the expression to a vanishing parenthesis and it follows that

Eq. 2.45

$$\sum_{A=1}^{n_{eq}} a(N_B, N_A) d_A = L(N_A) - a(N_A, g^h)$$

Which yields the single elements

Eq. 2.46

$$K_{BA} = a(N_B, N_A)$$

$$F_A = L(N_A) - a(N_B, g^h)$$

And the matrix form

Eq. 2.47

$$\boldsymbol{K} = [K_{BA}]$$

$$\boldsymbol{F} = [F_A]$$

$$\boldsymbol{d} = \{d_A\}$$

$$\boldsymbol{Kd} = \boldsymbol{F}$$

Solving for the d_A $A = 1, \ldots, n_{np}$ as

Eq. 2.48

$$\boldsymbol{d} = \boldsymbol{K}^{-1}\boldsymbol{F}$$

Then inserting the d_A back into Eq. 2.42 the final solution u^h is derived

$$u^h = \sum_{A=1}^{n_{eq}} N_A\, d_A + \sum_{C=n_{eq}+1}^{n_{np}} N_C\, d_C$$

2.1.6 THE VARIATIONAL FORM OF THE BVP

We could also reach the same form of the analysis problem of Eq. 2.29 by applying the principal of virtual work, which is derived by the variational formulation of the principle of minimum potential energy of a system. "The equilibrium of a body requires that for any compatible small virtual displacements imposed to the body in its state of equilibrium, the total internal virtual work is equal to the total external virtual work."

Eq. 2.49

$$W_{int} = W_{ext} \Rightarrow \int_V \{\delta\varepsilon\}^T\{\sigma\}\, dV = \int_V \{\delta u\}^T\{f\}^B\, dV + \int_{S_f} \left\{\delta u_{S_f}\right\}^T \{f^{S_f}\}\, dS + \sum_i \{\delta d_i\}^T \left\{R_i^C\right\}$$

where δu are the virtual displacements (perturbations or variations), that is the corresponding entity of the weighting functions w in the weighted residual methods, and $\delta\varepsilon$ are the corresponding virtual strains, which is in substance the ∇w term. The adjective "virtual" refers to those entities that are used by the analyst in a thought experiment to establish the integral equilibrium equation.

$\delta u_{S_f}^T$ is a vector of the virtual displacements of a random point on the surface S_f, where forces f^{S_f} are applied. $\{\delta d_i\}$ are the nodal virtual displacements combined with the nodal concentrated forces $\left\{R_i^C\right\}$. And $\{f\}^B$ are the body forces defined in the domain.

Evidently we can use the same piecewise continuous shape functions that are derived before considering a discretised domain and an interpolatory polynomial to approximate the displacement field, that is Eq. 2.26 again

Eq. 2.50

$$u^h(x) = \sum_{i=1}^{n} N_i d_i$$

The kinematic equations of the body, which are derivatives of the equations expressed in Eq. 2.50 are

Eq. 2.51

$$\{\varepsilon\} = [B]\{d_i\}$$

Substituting Eq. 2.51 and the constitutive relation expressed below into Eq. 2.49

Eq. 2.52

$$\{\sigma\} = [D]\{\varepsilon\}$$

we get

Eq. 2.53

$$\{\delta d_i\}^T \int_V [B]^T[D][B]\{d_i\}\,dV = \delta\{d_i\}^T \int_V [N]^T f^B\,dV + \delta\{d_i\}^T \int_{S_f} [N]^T f^{S_f}\,dS + \sum_i \delta\{d_i\}^T R_i^C$$

Which is exactly the same as is the formulation of Eq. 2.29, where

Eq. 2.54

$$a(w^h, v^h) = \int_V [B]^T[D][B]\,dV\{d_i\}$$

Eq. 2.55

$$L(w^h) = \int_V [N]^T f^B\,dV + \delta\{d_i\}^T \int_{S_f} [N]^T f^{S_f}\,dS + \sum_i \delta\{d_i\}^T R_i^C$$

2.2 CONVERGENCE CRITERIA AND ISOPARAMETRIC METHOD

2.2.1 CONVERGENCE CRITERIA (HUGHES, 1987)

We wish to define the shape functions in such a way that, as the finite element mesh is refined, the approximate Galerkin solution converges to the exact solution. The following question arises: What conditions must the shape functions satisfy so that this property is guaranteed? We shall be content, for the time being, to state sufficient conditions for convergence. These conditions are possessed by the most prevalent and important finite element shape functions. However, we note that convergent elements can be constructed from shape functions which do not satisfy all these requirements. Nevertheless these conditions may be considered basic in that they provide the simplest criteria to ensure convergence for a wide class of problems.

The basic convergence requirements are that the shape functions be:

1. Smooth (at least C^1) on each element interior Ω^e
2. Continuous across each element boundary Γ^e
3. Complete (can represent constant and linear states)

Conditions 1 and 2 guarantee that first derivatives of the shape functions have, at worst, finite jumps across the element interfaces; see Figure 2.6. This ensures that all integrals necessary for the computation of element arrays are well defined, since at most first derivatives appear in the integrand. If we permit finite discontinuities in the shape functions on element boundaries, the derivatives possess delta functions and we are unable to make sense out of the squares of these quantities that would appear in the stiffness integrands see Figure 2.6. In elasticity these two conditions imply the compatibility requirement, which is the displacements within the elements and across the element boundaries must be continuous and no gaps occur between elements. Elements with the latter property are called non-conforming or incompatible and the former one that satisfy the convergence criteria are conforming or compatible. There are some convergent non-conforming elements in the literature but there are no incomplete.

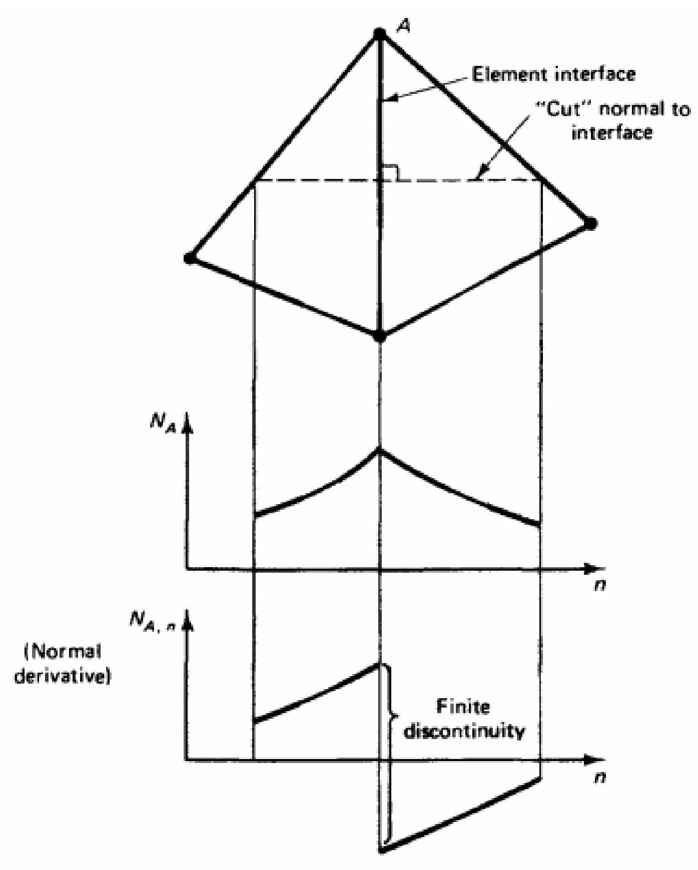


Figure 2.6 Example of shape function that satisfies conditions C1 and C2

Completeness requires that the element interpolation function is capable of exactly representing an arbitrary linear polynomial when the nodal degrees of freedom are assigned values in accordance with it. This requirement is vital because as the finite element mesh is further and further refined, the exact solution and its derivatives approach constant values over each element domain. To ensure that these constant values are representable, the shape functions must contain all constant and linear monomials. That is,

$$u^h = a_0 + a_1 x + a_2 y + a_3 z \ \ | \, {d_i}^e = a_0 + a_1 dx_i^e + a_2 dy_i^e + a_3 dz_i^e$$

This argument has been proved to be the key mathematical idea for proving convergence theorems for finite element approximations. Thus in elasticity, the presence of all monomials through linear terms means that an element may exactly represent all rigid motions and constant strain states.

For theories involving m$^{\text{th}}$ derivatives in the stiffness integrands, all the convergence criteria must be adapted (a basis of Hermite polynomials is used in such occasions):

1. Smooth, at least C^m on each element interior, Ω^e
2. Continuous across each element boundary at least C^{m-1}
3. Complete in monomials up to an order of m

2.2.2 ISOPARAMETRIC METHOD (HUGHES, 1987)

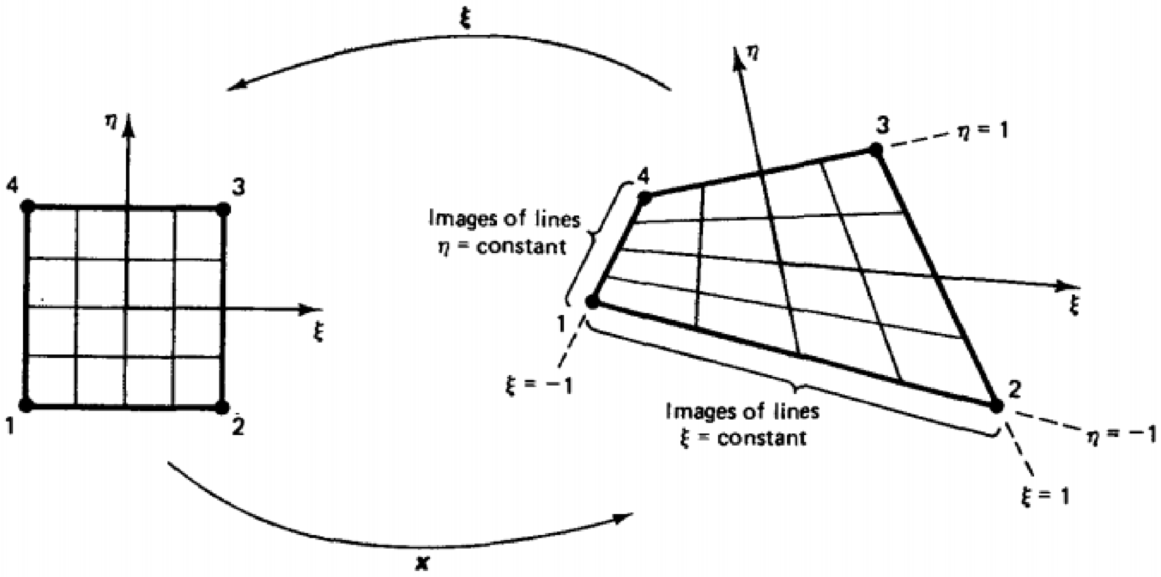


Figure 2.7 The linear mapping in $\mathbb{E}^2$ to it self

The main advantage of the isoparametric methods are that it is often easier to establish the convergence conditions directly rather than prove it for individual shape functions. So the importance of the isoparametric concept is that the three basic convergence conditions are virtually automatic. In addition, isoparametric elements may be designed to take on convenient shapes for practical analysis, including curved boundaries, and lend themselves to concise computer implementation.

Let : $\hat{\Omega} \to \Omega^e$, where $\hat{\Omega}$ is a parametric domain and Ω^e is the physical domain of an element, be of the form

Eq. 2.56

$$x(\xi) = \sum_{i=1}^{n_{en}} N_i(\xi) x_i^e$$

The element is said to be isoparametric if u^h can be written as

Eq. 2.57

$$u^h(\xi) = \sum_{i=1}^{n_{en}} N_i(\xi) d_i^e$$

The key point to observe in the definition is that the shape functions which define

Eq. 2.56 also serve to define Eq. 2.57.

In general, the degree of approximation used to describe the coordinate transformation is not equal to the degree of approximation used to represent a dependent variable $\widehat{N}_i^e(\xi) \neq N_i^e(x)$. In other words, two independent elements can be used in the finite element analysis: one for the approximation of the geometry x and the other for the interpolation of the dependent variable u of the solution space. Depending on the relationship between the degree of approximation used for the coordinate transformation and that used for the dependent variable, the finite element formulations are classified into three categories:

Subparametric formulations: m<n

Isoparametric formulations: m=n

Superparametric formulations: m>n

In ***subparametric*** formulations, the geometry is represented by lower order elements than those used to approximate the dependent variables. An example of this category is provided by the Euler-Bernoulli beam element, where the Hermite cubic element is used to approximate the transverse deflection, while the geometry is approximated, when straight beams are analyzed, with linear interpolation functions. In ***isoparametric*** formulations (the most common in practice and the one that is applied in Isogeometric analysis), the same element is used to approximate the geometry as well as the dependent unknowns: $\widehat{N}_i^e(\xi) = N_i^e(x)$. In ***superparametric*** formulations, the geometry is represented with higher-order elements than those used to approximate the dependent variables. This formulation is seldom used in practice. (Reddy, An introduction to the finite element method, Second Ed., 1993)

We shall now discuss some fundamental mathematical properties of mappings. Based on this, we shall argue that convergence condition of the smoothness of the function at the interior is generally achieved for isoparametric elements.

CONVERGENCE CONDITION 1

It is a consequence of the inverse function theorem that if x is

a) One-to-one
b) Onto
c) C^k, k≥1
d) $J(\xi) > 0 \ for \ all \ \xi \in \hat{\Omega}$

Then the inverse mapping $\xi = x^{-1} : \ \Omega^e \rightarrow \hat{\Omega}$ exists and is C^k

Let the mapping defined by Eq. 2.56 satisfy a) through d) then the smoothness requirement is satisfied also

Proof: By virtue of the hypotheses, $N_a = N_a(\xi)$ is also a C^1 function. Also considering the inverse mappin, $\xi = \xi(x)$ is also C^1. Thus $N_\alpha(x) = N_\alpha(\xi(x))$ is a C^1 function of x (this fact may be proved with the aid of the chain rule).

Definition 1: A mapping $x: \hat{\Omega} \to \Omega^e \subset R^n$ is said to be ***one-to-one*** if for each pair of points $\xi^{(1)}, \xi^{(2)} \in \hat{\Omega}$ such that $\xi^{(1)} \neq \xi^{(2)}$, then $x(\xi^{(1)}) \neq x(\xi^{(2)})$ in words, this statement means that two different points of $\hat{\Omega}$ do not get mapped into the same point in Ω^e.

Definition 2: $x: \hat{\Omega} \to \Omega^e$ is said to be ***onto*** if $\Omega^e = x(\hat{\Omega})$ (i.e. each point in Ω^e is the image of a point in $\hat{\Omega}$ under the mapping x)

Definition 3: Let $x: \hat{\Omega} \to \Omega^e$ be a ***differentiable mapping***. The determinant of the derivative, denoted by $J = \det\left(\frac{\partial x}{\partial \xi}\right)$, is called the ***Jacobian determinant***.

The Jacobian determinants in two and three dimensions, respectively, are given explicitly by

$$J = det\begin{bmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{bmatrix} \quad , \qquad J = det\begin{bmatrix} x_{,\xi} & x_{,\eta} & x_{,\zeta} \\ y_{,\xi} & y_{,\eta} & y_{,\zeta} \\ z_{,\xi} & z_{,\eta} & z_{,\zeta} \end{bmatrix}$$

CONVERGENCE CONDITION 2

If $\sum_{a=1}^{n_{en}} N_a = 1$, then completeness condition is satisfied for isoparametric elements

Proof: (for the three dimensional case)

$$\begin{aligned} u^h = \sum_{a=1}^{n_{en}} N_a\, d_a^e &= \sum_{a=1}^{n_{en}} N_a\,(c_0 + c_1 x_a^e + c_2 y_a^e + c_3 z_a^e) \\ &= c_0 \sum_{a=1}^{n_{en}} N_a + c_1 \sum_{a=1}^{n_{en}} N_a x_a^e + c_2 \sum_{a=1}^{n_{en}} N_a y_a^e + c_3 \sum_{a=1}^{n_{en}} N_a z_a^e \\ &= c_0 \sum_{a=1}^{n_{en}} N_a + c_1 x + c_2 y + c_3 z \end{aligned}$$

The condition that the shape functions sum to one is easily checked on a case by case basis.

The only remaining convergence condition is C2, the continuity requirement on Γ^e. This condition can be verified once the construction of the global shape functions from the element shape functions is explicated. It happens that if this procedure is done in the "obvious" way, continuity is achieved.

Summary. The importance of the isoparametric concept is that the three basic convergence conditions are virtually automatic. In addition, isoparametric elements may be designed to take on convenient shapes for practical analysis, including curved boundaries, and lend themselves to concise computer implementation.

2.3 THE ISOGEOMETRIC ANALYSIS

2.3.1 THE ESSENTIALS IN ISOGEOMETRIC ANALYSIS

Any affine transformation of space can be accomplished by means of a certain orthogonal transformation and successive "compressions" on some three mutually perpendicular lines. In an affine transformation, parallel lines and planes are transformed into parallel lines and planes. In mechanics, it is used in the study of small deformations of continuous media; in such deformations, small elements of the medium in the first approximation undergo affine transformations.

By the above definition of the affine mapping we observe that this transformation is not irregular to consider a big number of independent variables (degrees of freedom). We can handle the behaviour of a certain domain just on a few control variables. We need an expression of the type $\mathbf{x} = \sum \alpha_j \mathbf{x_j}$ which specifies how we have to weight the points $\mathbf{x_j}$ such that their weighted average is $\mathbf{x}$. This relation should be still valid if we apply an affine map to all points $\mathbf{x_j}$ and to $\mathbf{x}$. As an example, the midpoint of a straight line segment will be mapped to the midpoint of the affine image of that straight line segment. Also, the centroid of a number of points will be mapped to the centroid of the image points. The property of the weights that keeps them unchanged under affine transformations is called barycentric; and the linear combination using that type of weights is called barycentric combination. In the notion of barycentric combinations it is:

A map $\boldsymbol{\Phi}$ *that maps* $\mathbb{E}^3$ *into itself is called an affine map if it leaves barycentric combinations invariant.* So if

Eq. 2.58

$$\boldsymbol{x} = \sum \alpha_j \boldsymbol{x_j}\,; \quad \sum \alpha_j = 1; \quad \boldsymbol{x}, \boldsymbol{x_j} \in \mathbb{E}^3$$

and $\boldsymbol{\Phi}$ is an affine map, then also

Eq. 2.59

$$\boldsymbol{\Phi x} = \sum \alpha_j \boldsymbol{\Phi x_j} \;\; ; \;\; \boldsymbol{\Phi x}, \boldsymbol{\Phi x_j} \in \mathbb{E}^3$$

Generally affine maps are expressed as

$$\boldsymbol{\Phi x} = A\boldsymbol{x} + \boldsymbol{v}$$

This very important property of the geometric bases makes them suitable not only in computational geometry but in analysis too. The geometric bases are constructed especially for geometric designs accompanied with all the necessary properties that will make the creation and the modification of a locus accurate and easy handled. The smooth and stable control of the locus is one of the main issues in creating accurate geometries numerically. These characteristics are desirable in analysis also. The analysis, in terms of the kinematics in continuum mechanics, is not that far from handling a locus in a smooth way. One can observe in all the governing equations that a degree of smoothness is required in all the phenomena. Another primal admission of continuum mechanics is that each particle (point) of the domain at every state should occupy exclusively its own position (one-to-one correspondence, see convergence condition 1). This requirement is perfectly satisfied by the geometric bases that are structured by simple affine mappings conserving the barycentric property.

Thus, due to the barycentric property of the basis the implementation of it on a field lends

- Integrity (onto correspondence of the points)
- C^{∞} Continuity in the interior
- and one-to-one correspondence of each point

to characterize entire the domain of the field, which additionally is being conserved under affine transformations.

Considering these properties we have the ability to impose affine transformations that are considered in an analysis process

- form affine mappings that are helpful in analysis procedures using simplified domains (isoparametric concept)
- application of any affine transformation on the domain through the control handles (control points or variables) that are being weighted by the basis

Specifically the geometric bases can be characterized as isoparametric as they possess

- The barycentric combination property $\sum_{A=0}^{n} N_A = 1$
- C^{∞} Smoothness in the interior of the element as every basis function is constructed by an affine transformation of a lower degree one till the root; the root is an affine transformation by itself that is the identity function (step function) and the linear interpolation functions.

- C^k Smoothness at the element boundaries C^k where $k = p - 1 - m$, p the degree of the basis and m the multiplicity of the knots that form the element boundaries
- One to one and onto mappings are formed between the physical and the parametric domain
- The mappings are differentiable and the determinant of them is non singular (except certain conditions overlapped control points and zero knot spans)

Furthermore the geometric bases are compatible with the weighted residual methods as they possess all the requirements to form a convergent solution space. The required properties of a basis are

- Members of a complete Sobolev vector space (continuous enough, can form linear combinations etc.)
- The boundary conditions are easily implemented by boundary shape functions; thus the trial function satisfies the boundary conditions for any solution.

It is noticeable that compared to the classical FEA the isoparametric analysis that is based on geometric bases (isogeometric), can represent the geometry exactly and approximate the undergoing kinematics. **That is, the resulted displacement field is not the exact one, but the error is distributed smoothly in entire the domain.**

In isoparametric concept as explained in section 2.2.2 the same shape functions are used for the geometric mapping as for the approximation of the solution field. Which are

Eq. 2.60

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}(\xi) = N_A(\xi) \cdot \boldsymbol{P_A}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \;\; , \quad \xi \in \hat{\Omega} \,, \boldsymbol{x}\colon \hat{\Omega} \rightarrow \Omega$$

Eq. 2.61

$$\hat{u}^h(\xi) = \sum_{A=0}^{n} N_A(\xi) \boldsymbol{d_A} \,, \quad \hat{u}^h\colon \widehat{\Omega} \rightarrow \mathbb{R}$$

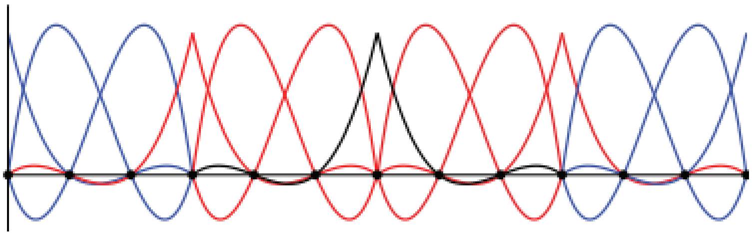
Where $N_A(\xi)$ are functions of a geometric basis in parametric domain

The solution field of the physical domain Ω is given by the parametric coordinate $\xi \in \hat{\Omega}$, in which the geometric basis functions are defined. Equivalently, Eq. 2.61, could be expressed with regard to the physical coordinates by considering a composition with the inverse of the geometrical mapping $\{x\} = \sum_{A=0}^{n} N_A(\xi) \boldsymbol{P}_A$, such that $u^h\colon \Omega \rightarrow \mathbb{R}$

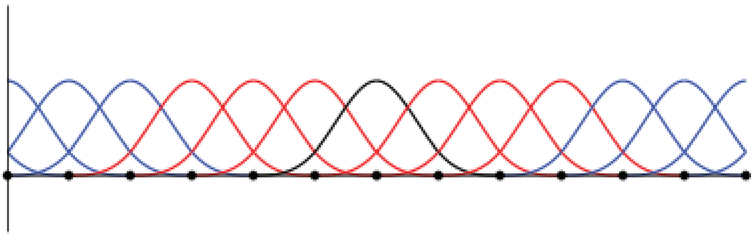
Eq. 2.62

$$u^h = \hat{u}^h \circ \boldsymbol{x}^{-1}$$

The notion of the parametric domain is the same as in CAGD (computer aided geometric design). The mapping from the parametric domain to the physical is also the same. In that notion we take advantage of the multi-spanned patches, consisted of non-zero knot spans to realize them as finite elements and the corresponding approximatory (non-interpolatory) with local support (like piecewise) basis functions. The crucial difference of the local support functions and the piecewise continuous functions in classical FEA is that the geometric basis functions are smooth and non zero in all the elements that they are supported and at their boundaries too. In figure below it is shown that the black colored function in FEA is only C^0 continuous at the boundaries.



(a) Standard cubic finite element basis functions with equally spaced nodes



(b) Cubic B-spline basis functions with equally spaced knots

Figure 2.8

This renders in the FEA the first derivative at that boundary to be only weakly satisfied by a finite jump that denotes the difference in slopes see Figure 2.9. In Isogeometric analysis there are no such jumps of the solution field. This gives rise for deriving smoothly distributed displacement fields as well as the derivative fields (stress-strain fields, curvature fields etc.) see Figure 2.10.

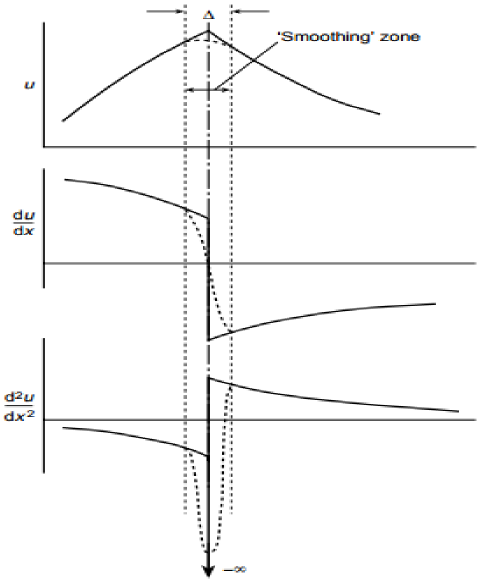


Figure 2.9 Differentiation of function with slope discontinuity C^0 continuity

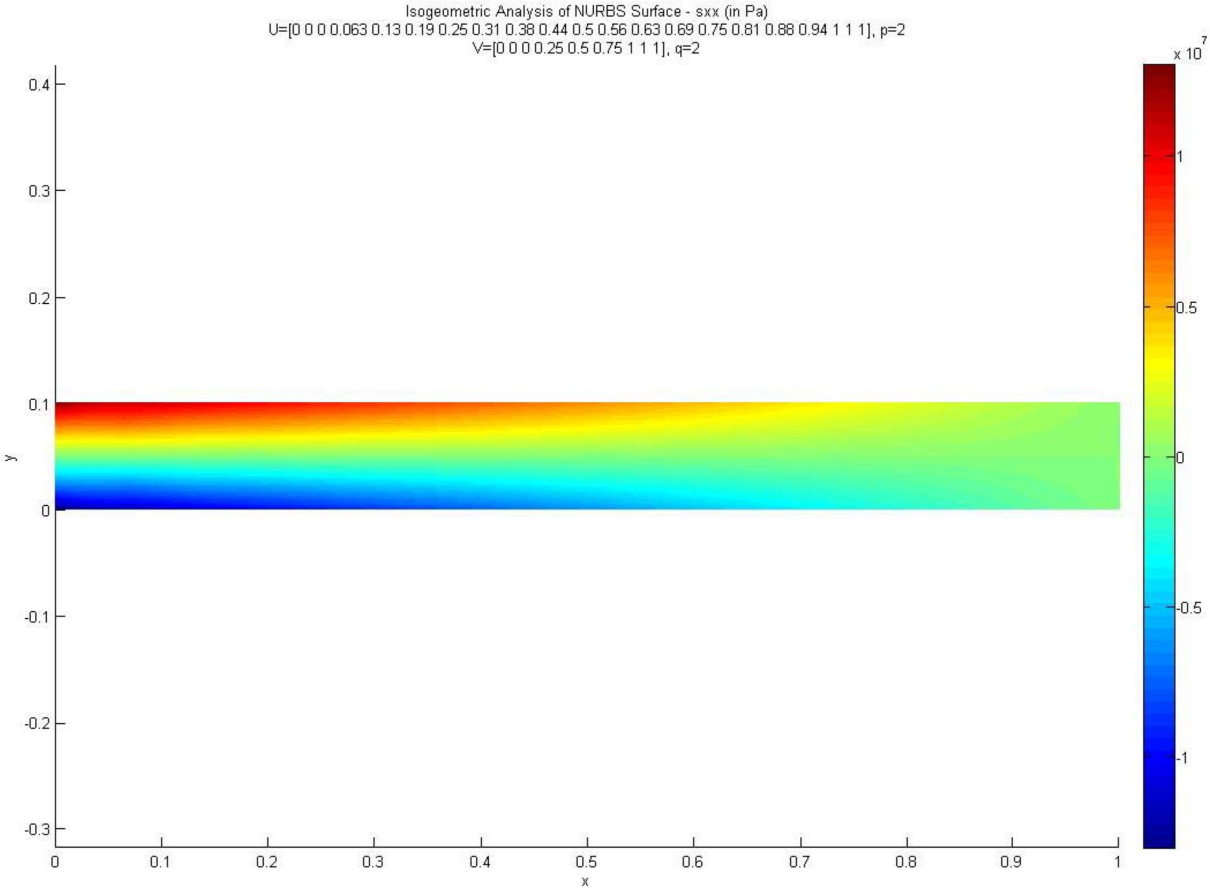


Figure 2.10 Example of smooth distribution of the normal stress in a plane stress problem of a fixed bending beam. The basis functions are of 2nd degree and the field is discretized by a 4x16 mesh

Since a basis is formed by such a geometric approximatory and locally supported functions and a well-defined problem is considered, then the implementation of the Galerkin's method will intuitively yield a very smooth distribution of the error along the domain. As is explained in section 2.1.1 the Galerkin's method is able to give the optimum solution of the current selected solution subspace S^n by orthogonalizing the residual to the solution subspace itself. This yields a projection of the exact solution to the solution subspace, thus the minimum residual (distance) of the two vectors. Due to the generally smooth behaviour of the basis (it is imported from CAGD) one would expect that the error distribution should be smoothly distributed too, especially compared to the classical basis of the FEA that are more steep and nervous. This smoothness is justified by the variation diminishing property and the convex hull property that possess these mappings; by these properties the solution field is guaranteed to be bounded by the convex hull of the control polygon made by the values on the control variables; and finally the solution field should be characterized by an absence of any oscillations.

2.3.2 CONSIDERED DOMAINS AND MAPPINGS

As is defined in the computational geometry part I, there are two domains that the geometric basis maps the points from the one to the other. Essentially the basis maps by a single or multiple sequential affine transformations the parametric domain to the physical one. The basis functions are used as weights to combine convexly the entities in the physical domain that are called control points; this renders a locus in the physical domain mapped from a simple one that is defined in the parametric domain. In Figure 2.11, an index space is illustrated, which actually does not being mapped ever; it is only for indexing needs of the basis functions and the knot spans (zero and non-zero). Thus, in the index space all the knot spans are shown in equal length, insensibly of what is the real length of them in the parametric domain and of course the physical. In Figure 2.11, it is shown a parametric domain with its bivariate basis defined.

The same bivariate basis is plotted in Figure 2.13 in a 3D mode; it is illustrated that each bivariate basis function is a surface in the parametric domain. The corresponding one parameter basis functions are plotted in both directions to indicate that each bivariate basis function is a tensor product (all the combinations of products between the two coordinates is implied) of two single parameter.

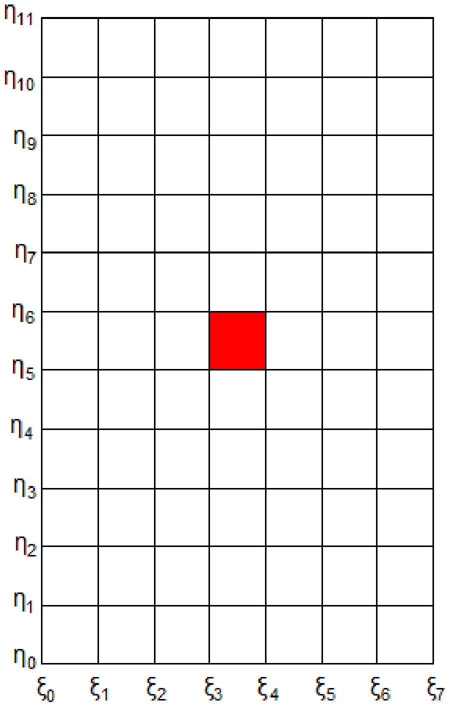


Figure 2.11 The index space of a bivariate parametric domain. This domain is not mapped! (red region indicates the same group of particles in different domains)

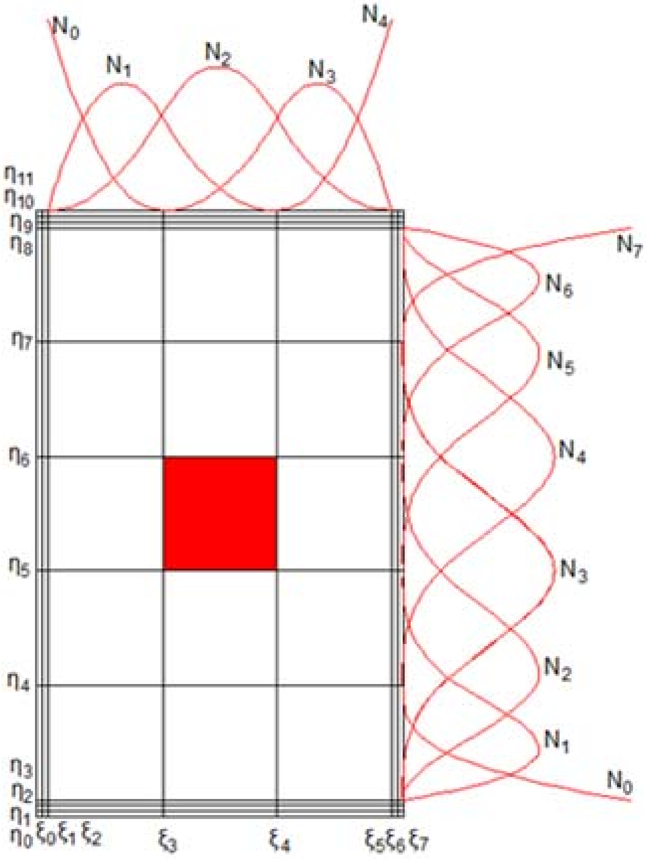


Figure 2.12 A parametric domain of a bivariate basis of degrees p=2 and q=3 (red region indicates the same group of particles in different domains)

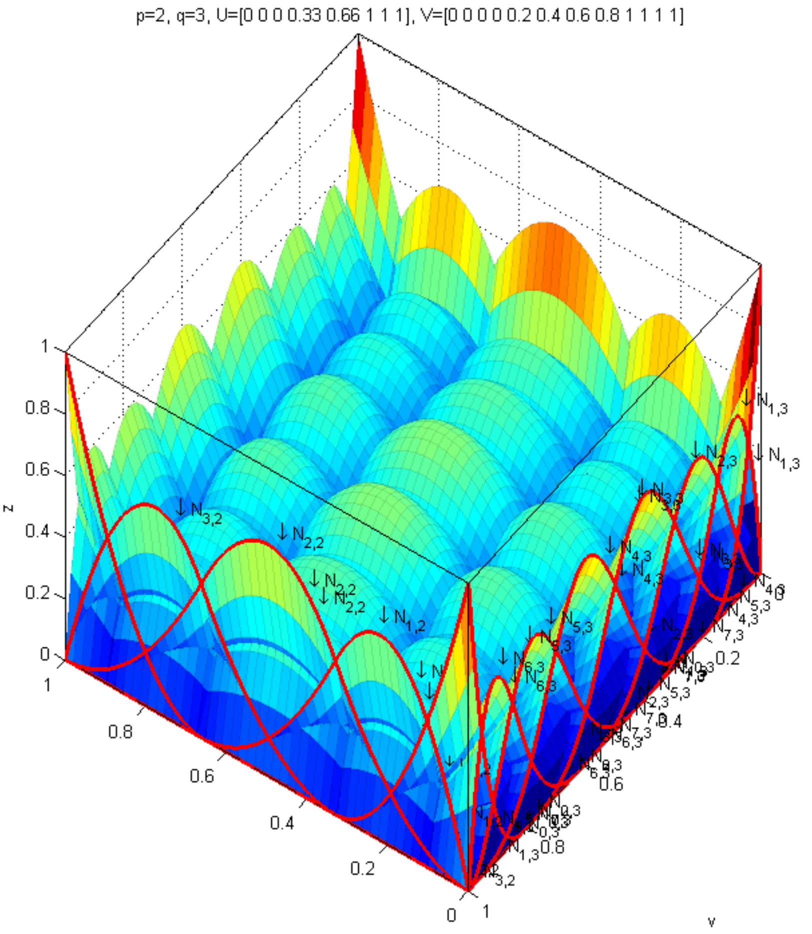


Figure 2.13 Bivariate basis shown in 3D mode

This parametric domain is mapped by sequential (two in number for a 2nd degree basis) affine transformations to the physical domain see Figure 2.14; in this figure it is shown two geometries that can be considered to be mapped directly from the same parametric domain, or alternatively the second geometry can be considered to be mapped from the first one after a deformation. This deformation, the map between the two geometries, is an affine map also; it is implemented by affinely transforming the control points only. This ability of affinely transforming a geometry by the affine transformation (displacements) of its control points, is due to the barycentric property of the basis which makes it unaffected to such transformations.

Contrary to the classical FEA where the nodal variables have physical meaning, since they are values of the solution field, in isogeometric analysis there is no physical meaning of the displacements of the control points. Due to the non-intepolatory character of the geometric basis the displacements of the control points obtain values out of the displacement field, yet are utilized as coefficients in convex combinations of geometric basis functions so as to approximate the solution field.

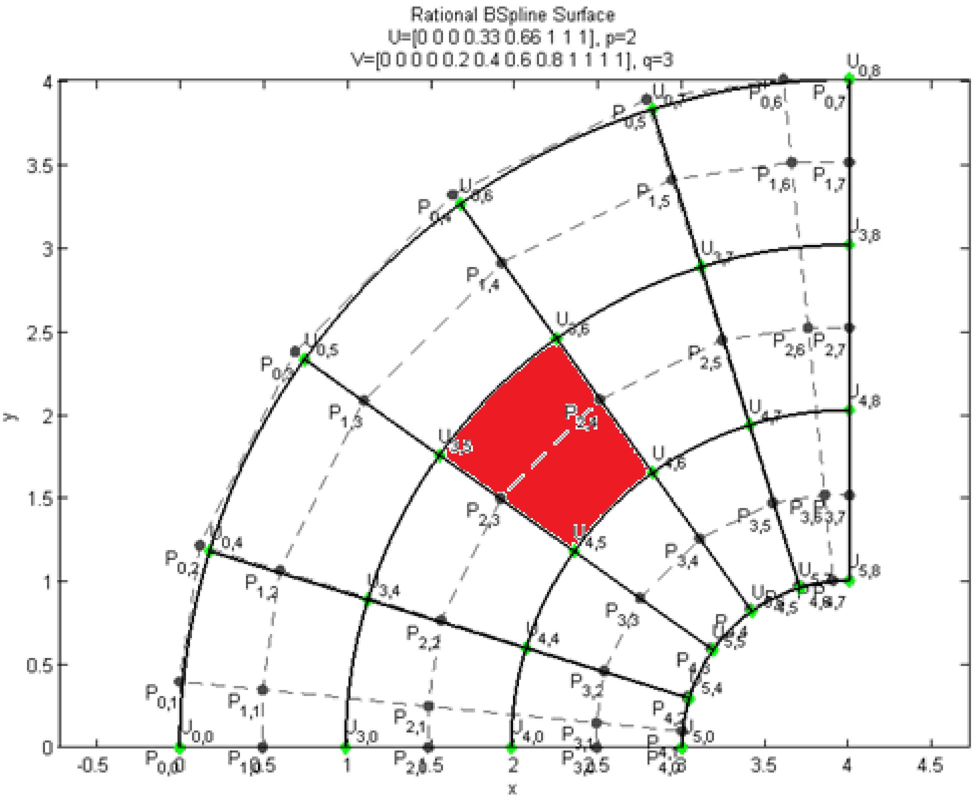

Rational BSpline Surface
U=[0 0 0 0.33 0.66 1 1 1], p=2
V=[0 0 0 0 0.2 0.4 0.6 0.8 1 1 1 1], q=3

Figure 2.14 Domain in physical space mapped affinely from the parametric (the red region indicates the same group of particles in different domains).

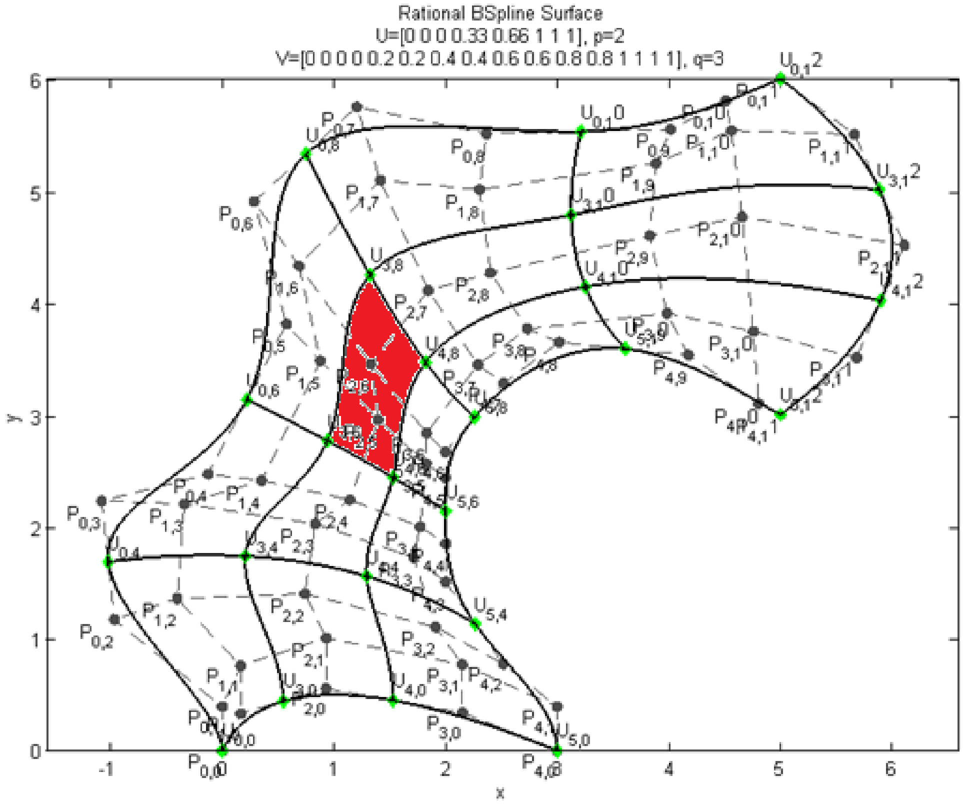

Rational BSpline Surface
U=[0 0 0 0.33 0.66 1 1 1], p=2
V=[0 0 0 0 0.2 0.2 0.4 0.4 0.6 0.6 0.8 0.8 1 1 1 1], q=3

Figure 2.15 Deformed domain in physical space; mapped affinely from the above one that is already in physical domain or directly from the parametric domain (the red region indicates the same group of particles in different domains)

To get a feeling about the influence of the displacement of a single control point it is illustrated below a geometry that is mapped from the parametric domain of Figure 2.12 as is (the jacobian matrix is an identity matrix). Then the control point that corresponds to the bivariate basis function $N_{4,4}$ is displaced and the effect of that displacement is shown in Figure 2.16. The blue dashed lines is the control polygon and the thick black lines are the knot lines of the parametric domain mapped to the present physical domain. We see the effect of the displacement that extends only on the local support of the corresponding bivariate shape function, which is the 4th knot span in ξ direction and the 4th to 8th knot spans in η direction (the knot spans are indexed according to the index space see Figure 2.11 and the left knot of that span). In Figure 2.17 it is shown the result of the displacement of the control point that corresponds to the bivariate basis function $N_{2,4}$; the area that this bivariate shape function is supported is distorted analogous to the control point's displacement. The resulted displacement and strain fields are shown in subsequent figures.

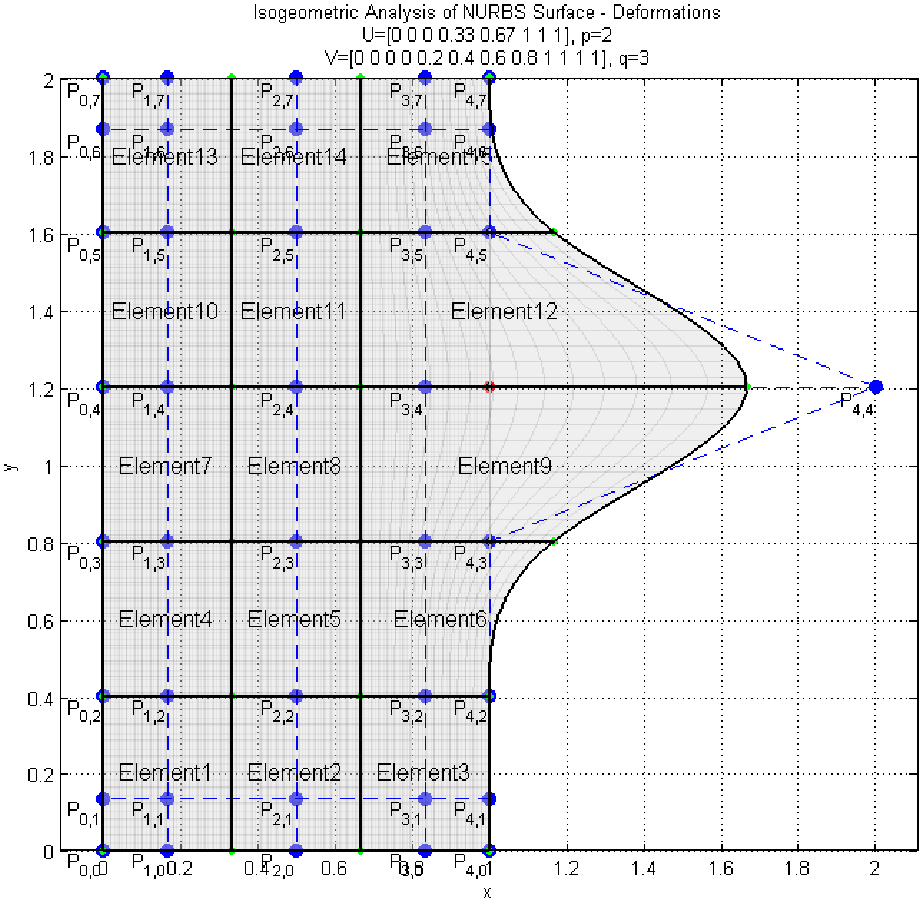


Figure 2.16 Distortion of the corresponding group of particles that the bivariate shape function is supported in (knot spans 4thx(4th-8th))

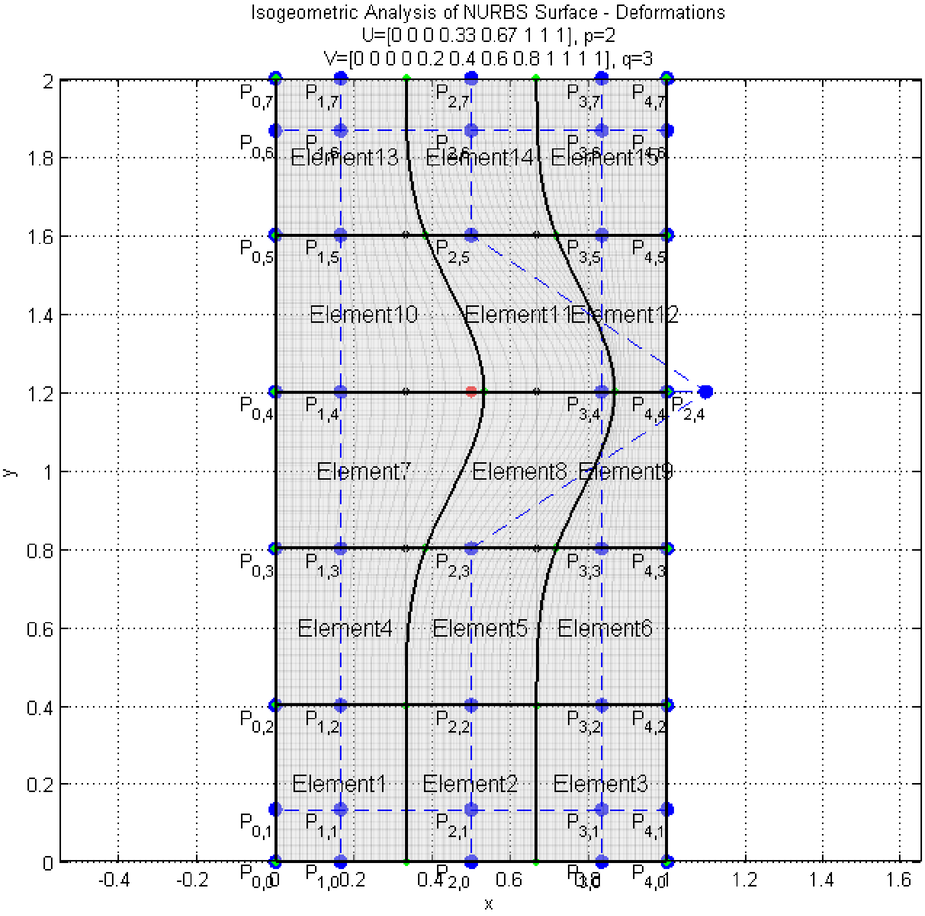


Figure 2.17 Distortion of the corresponding group of particles that the bivariate shape function is supported in (knot spans (2nd-4th)x(4th-8th))

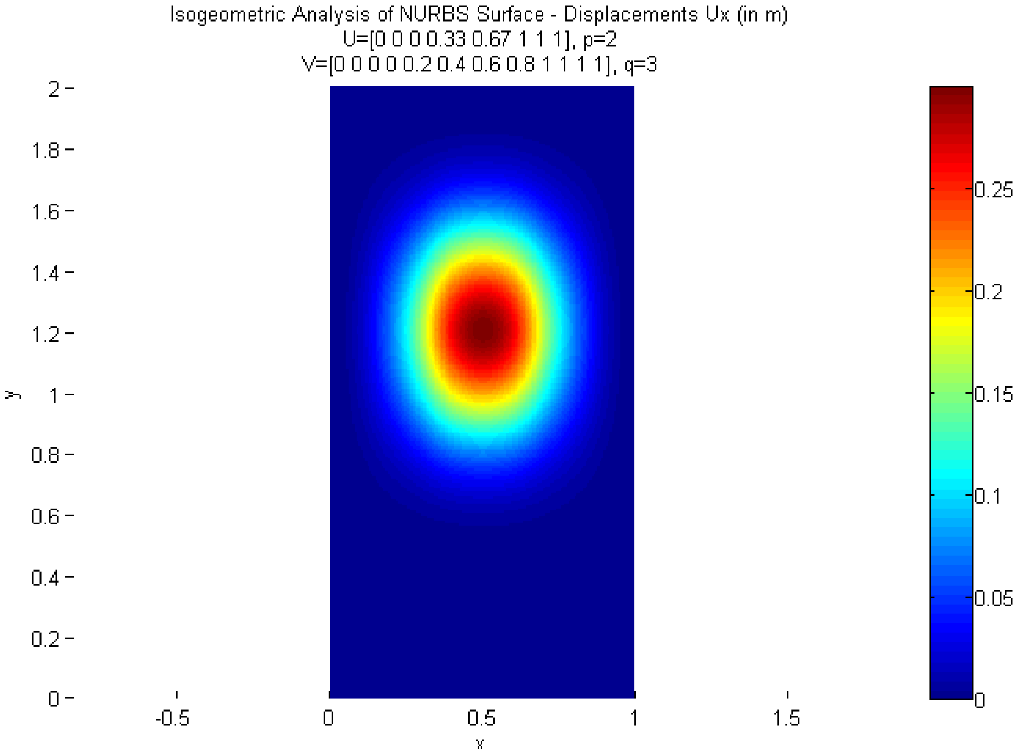


Figure 2.18 U_x displacement field due to the distortion of Figure 2.17

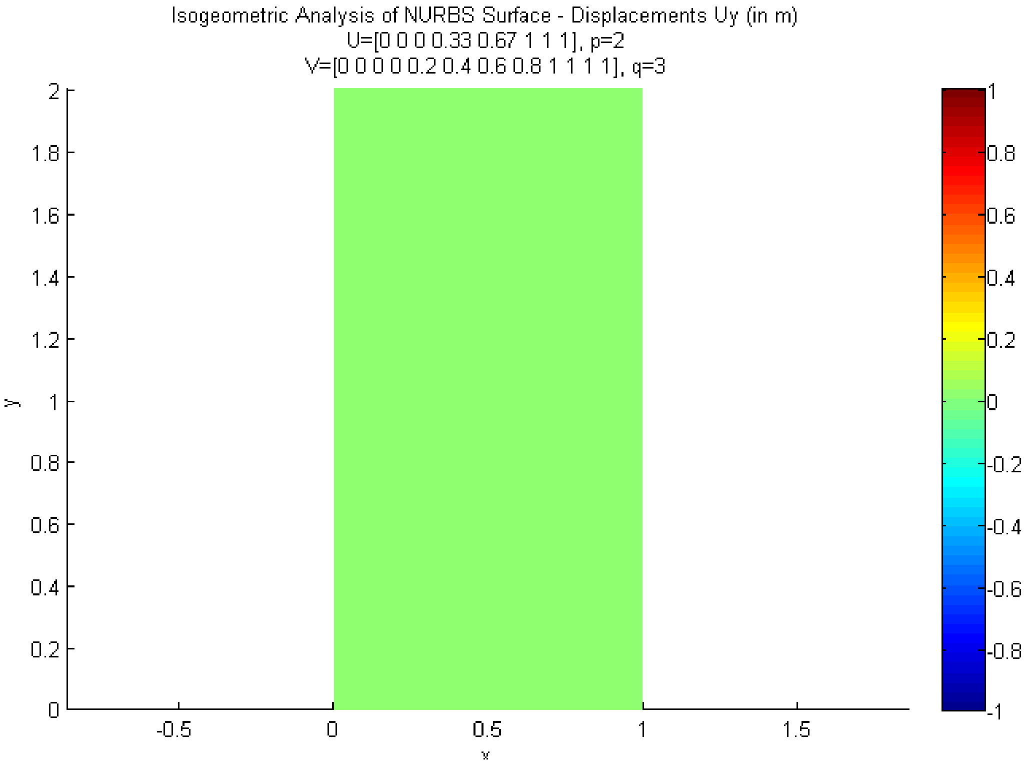


Figure 2.19 U_y displacement field due to the distortion of Figure 2.17

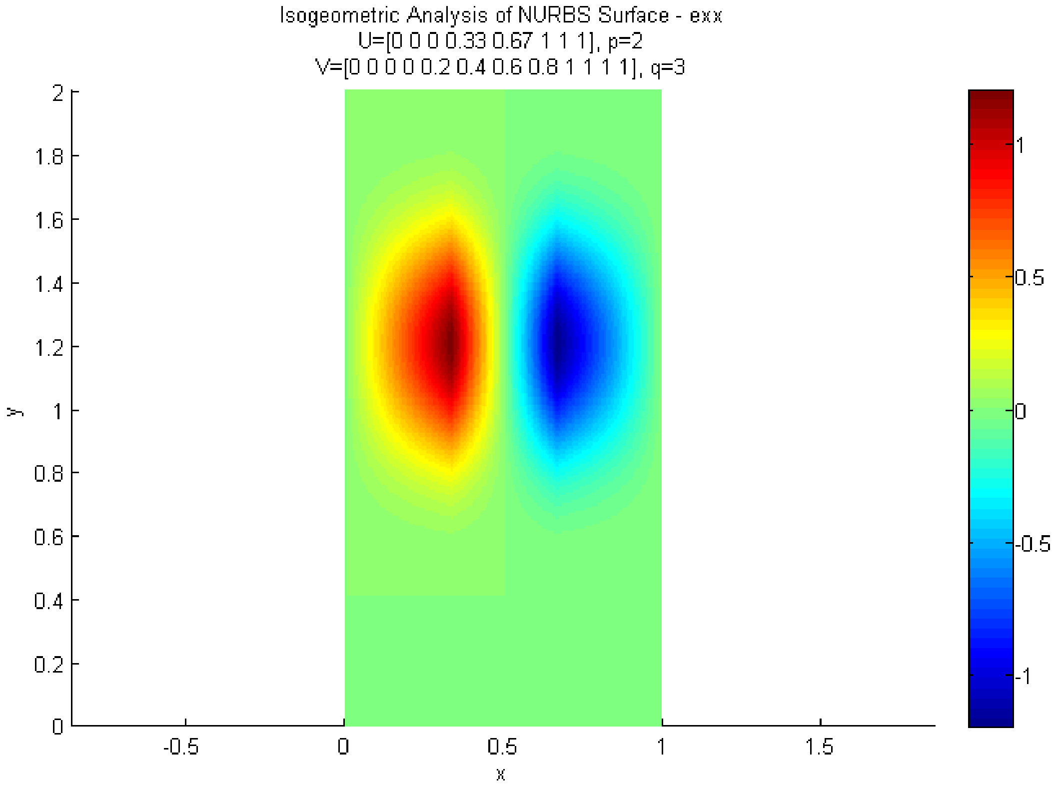


Figure 2.20 e_{xx} strain field due to the distortion of Figure 2.17

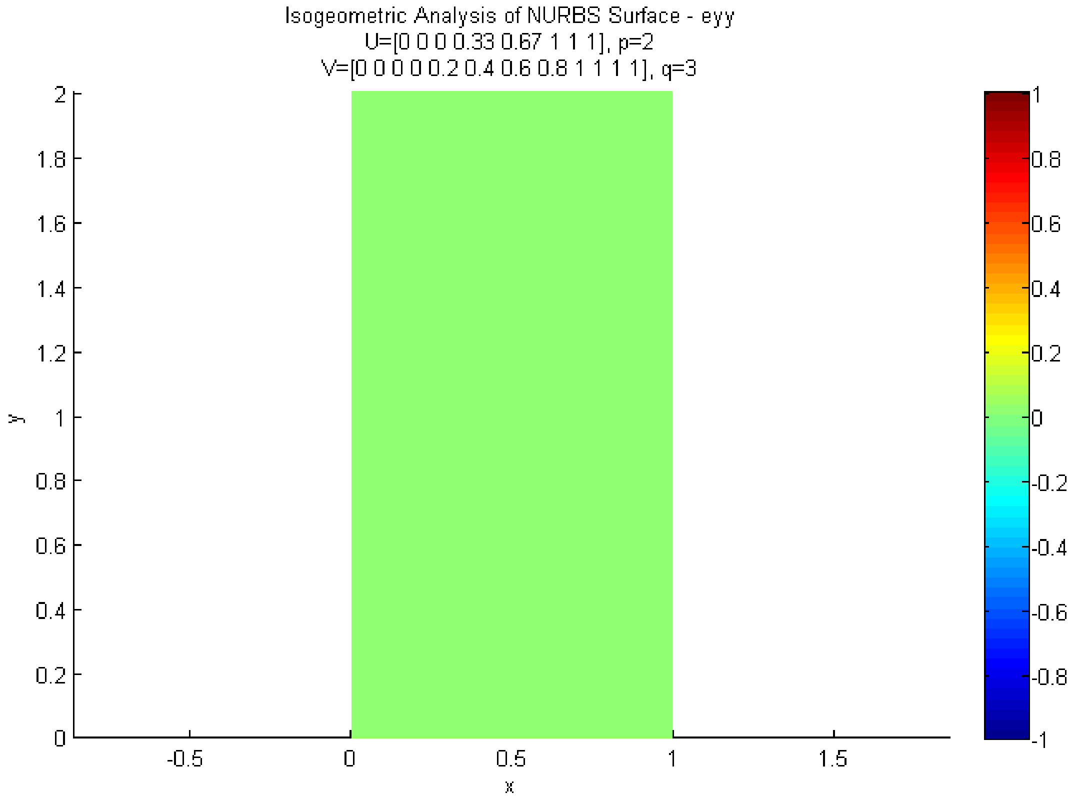


Figure 2.21 e_{yy} strain field due to the distortion of Figure 2.17

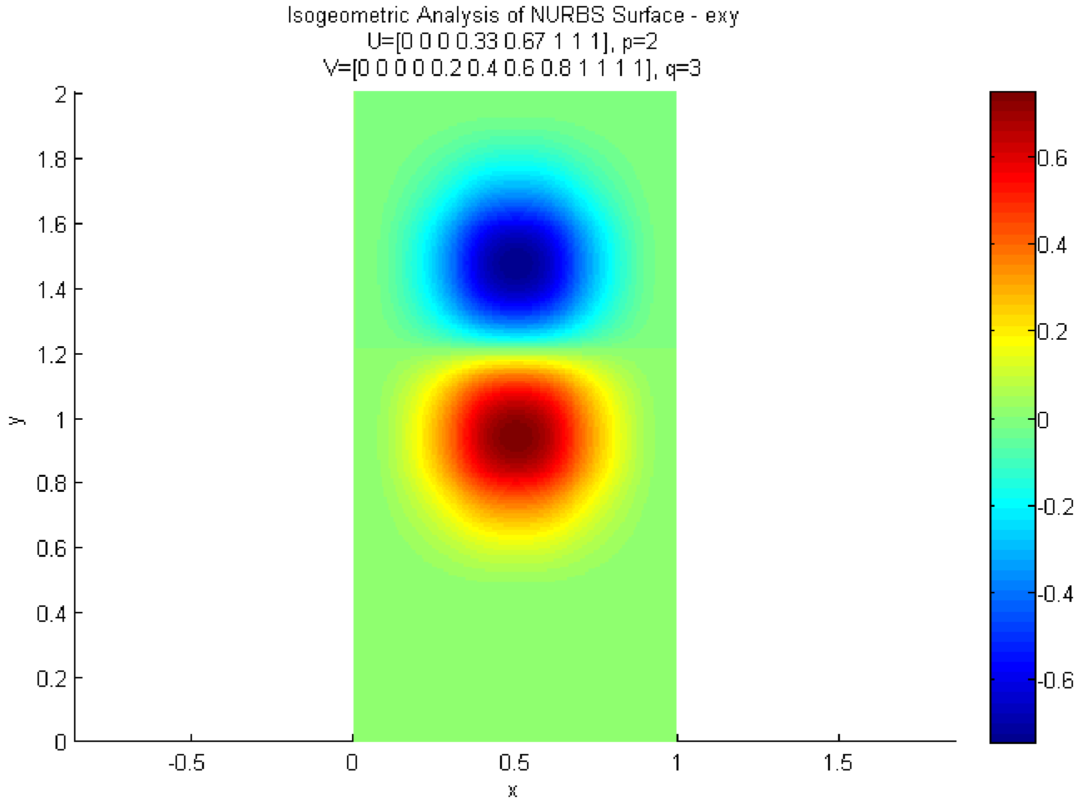


Figure 2.22 e_{xy} strain field due to the distortion of Figure 2.17

2.3.3 THE BASIS FUNCTIONS AND THE REFINEMENTS

The subdomains of a locally supported function are defined by knots and the role of the control variables is undertaken by the displacements of the control points.

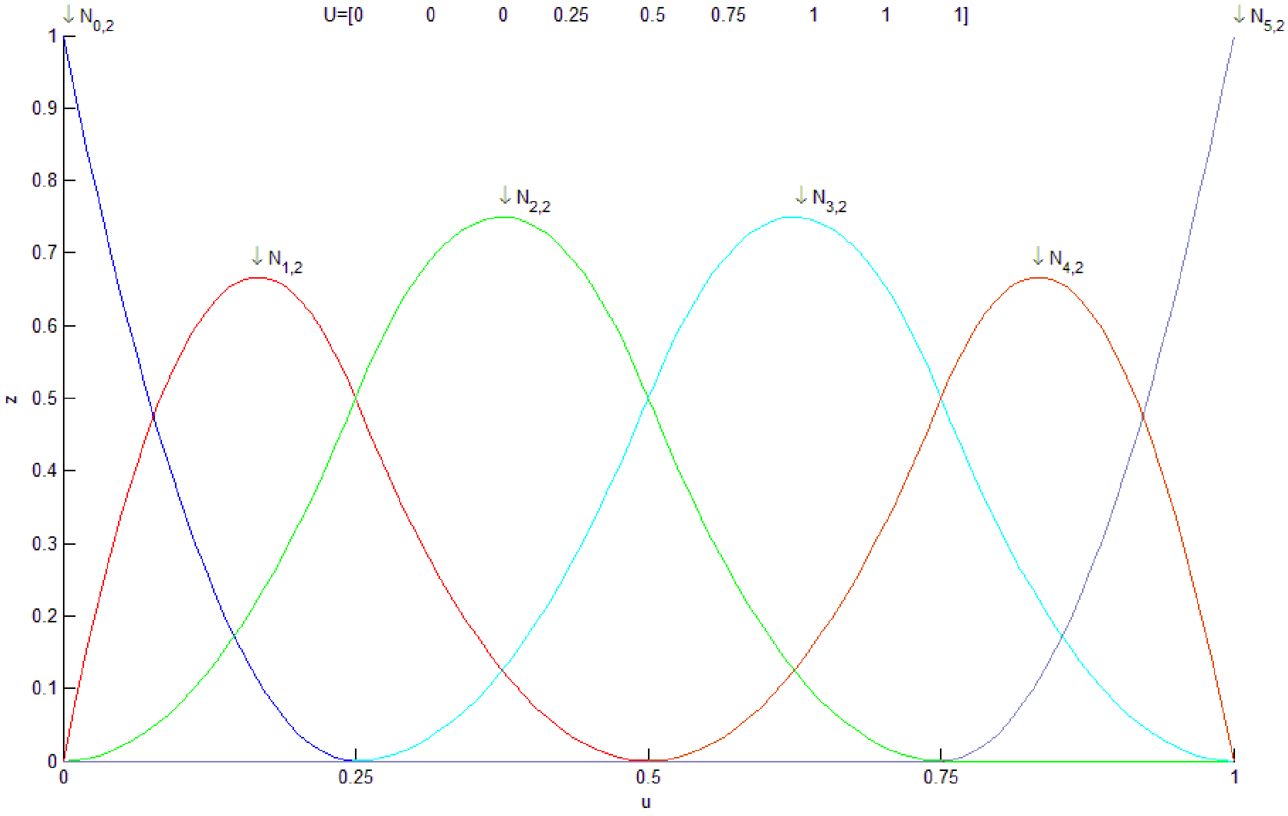


Figure 2.23 Piecewise functions of B-Spline type of 2nd degree defined in four subdomains (elements). Subdomains are defined by the knot vector U=[0 0 0 0.25 0.5 0.75 1 1 1]

More piecewise functions can be used along with more subdomains to improve the accuracy of the approximate solution (h-refinement). As we increase the number of subdomains for the locally supported functions, we can represent a strongly variant function by using the linear combination of more basis functions as in Eq. 2.61. The accuracy is also improved by setting zero knot spans which results in additional test functions, as it is shown in the figure below. By introducing zero knot spans, the continuity at those knots is degraded by the number of zero knot spans C^{p-1-k} (k-refinement), as it is assumed that these functions pass in the zero knot span; where k is the number of zero knot spans in that point and p is the polynomial degree.

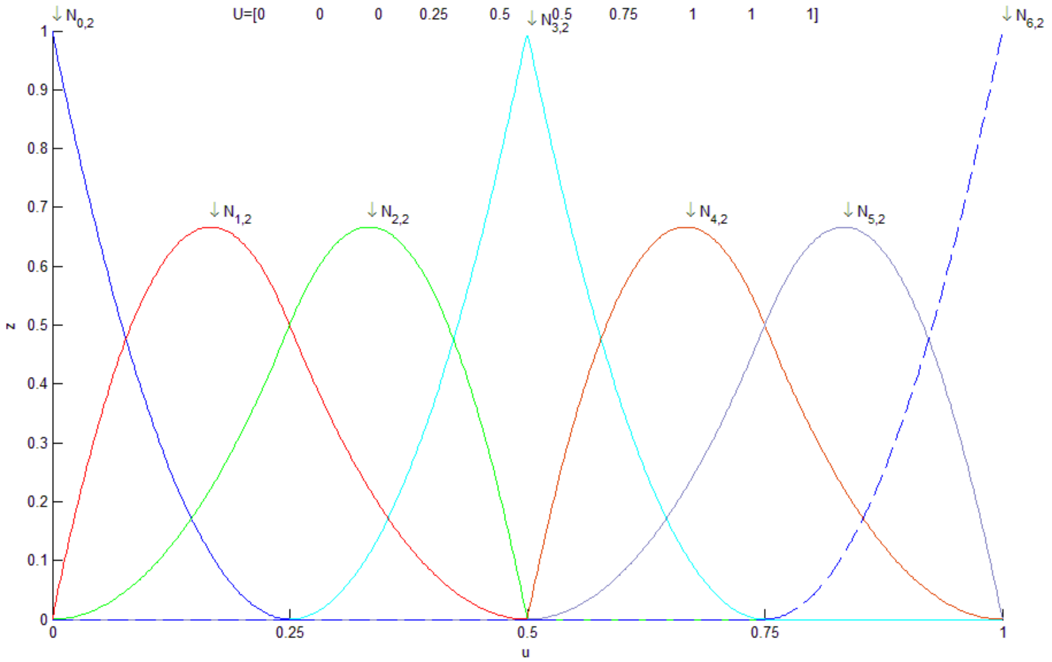


Figure 2.24 k-Refined piecewise functions of B-Spline type of 2nd degree defined in four nonzero and one zero length subdomain. Subdomains are defined by the knot vector U=[0 0 0 0.25 0.5 0.5 0.75 1 1 1]

The accuracy is also improved and the basis is enriched with more shape functions by degree elevation (p-refinement). In the figure below a basis is shown that stems from the one in Figure 2.18 applying an hpk-refined.

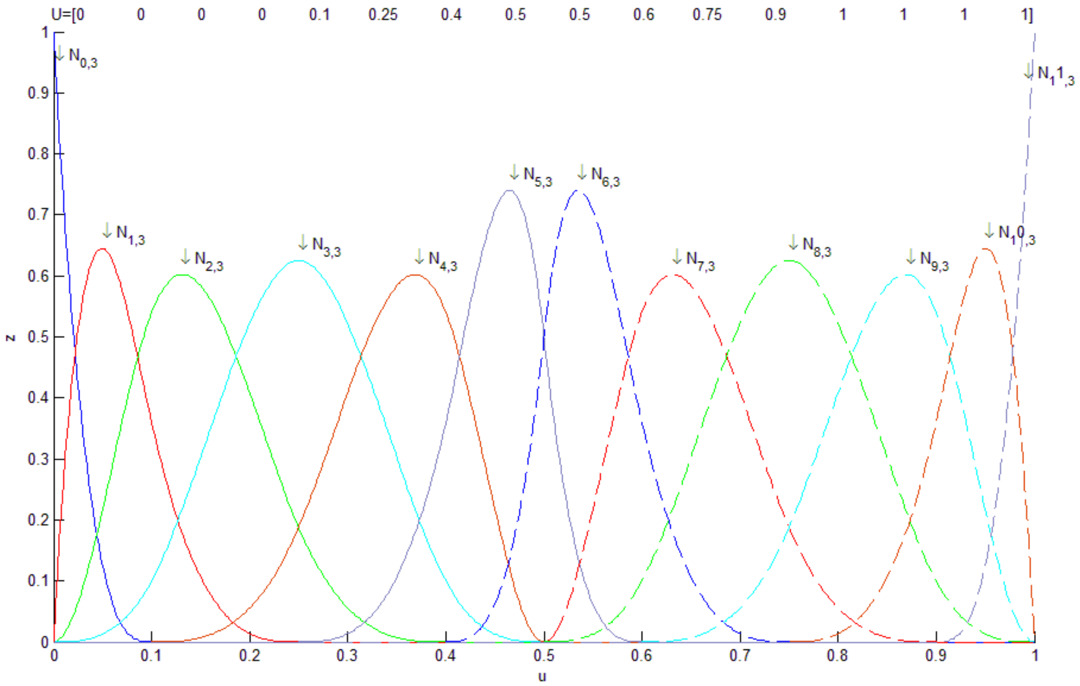


Figure 2.25 hpk-Refined piecewise functions of B-Spline type of 3rd degree defined in eight nonzero and one zero length subdomain. Subdomains are defined by the knot vector U=[0 0 0 0 0.1 0.25 0.4 0.5 0.5 0.6 0.75 0.9 1 1 1 1]

2.3.4 THE NUMERICAL INTEGRATION

Implementing the finite element method one should develop the corresponding routines with the less possible calculations. Considering the patch of knot spans that is represented in the parametric domain, one can assume that entire the patch is one single element; which element is mapped afterwards to the physical space. This is feasible; as it can isoparametrically analysed by a single mapping. Though, we can observe in the parametric domain that the shape functions are zero everywhere except of a few knot spans that they are locally supported; it will be a waste of computing time to calculate vanishing shape functions. Thus, the best tactic is to consider indeed a discretization of the parametric domain and assume that every non-zero knot-span is a single element. With the help of the index space see Figure 2.11, one can indicate easily for each element which shape functions are non-zero. These elements can be isoparametrically analysed considering a simpler one domain by a mapping. The simpler domain of course is the one suitable for gauss quadratures method of integration that is sized $[-1,1] \times [-1,1]$. Such a domain is called parent domain and is illustrated in Figure 2.21.

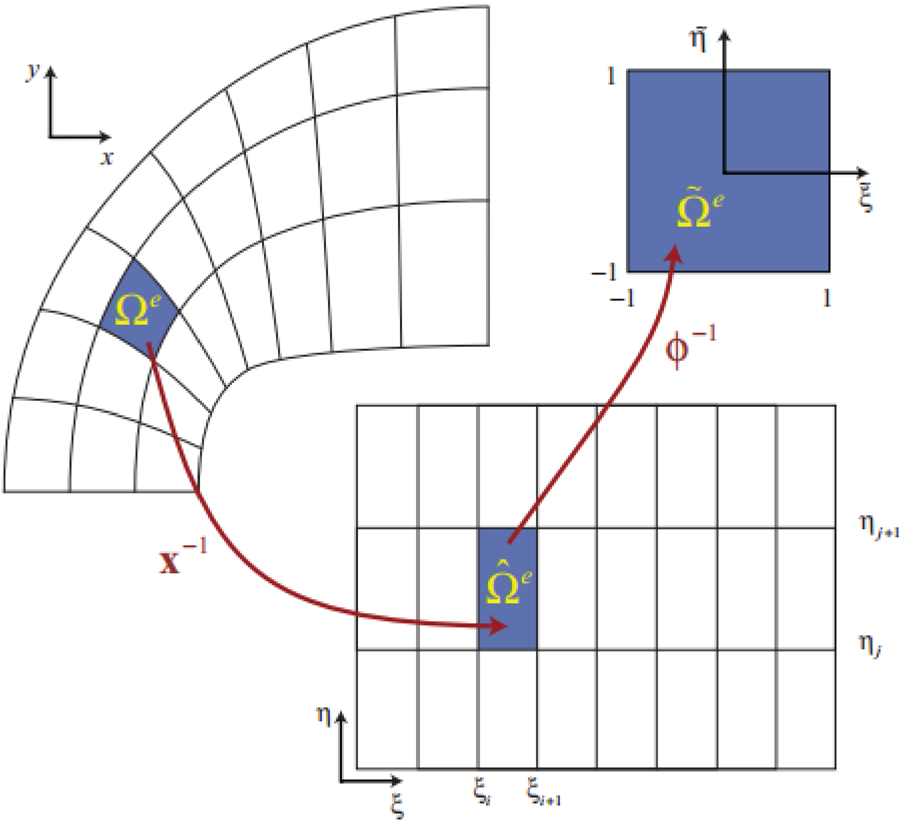


Figure 2.26 The most efficient implementation of the numerical integration is done at the knot spans individually. A third domain is defined, known also in classical FEA as the parent domain, to implement the gauss quadratures type of integration

3 STEP BY STEP IMPLEMENTATION OF THE MATRICES OF THE PLANE STRESS-STRAIN PROBLEM

3.1 DERIVATION OF STIFFNESS MATRIX

In order to compute the stiffness matrix, according to the Eq. 2.54, we first need to form the deformation matrix $\boldsymbol{B_2}$**,** which is with respect to the parametric domain; afterwards it is derived the one that is referred to the physical domain $\boldsymbol{B} = \boldsymbol{B_1} \cdot \boldsymbol{B_2}$ after mapping the $\boldsymbol{B_2}$ with the help of the $\boldsymbol{B_1}$ matrix that consist of elements from the inverse jacobian matrix. Thus, we compute the strain tensor at a point in the physical domain by mapping it from the parametric.

We derive the matrix $\boldsymbol{B}$ from an expression that relates the strains in the physical domain to the coordinates in the parametric domain.

Eq. 3.1

$$\boldsymbol{\varepsilon}(\xi,\eta) = \begin{bmatrix} \dfrac{\partial \mathrm{u}(\xi,\eta)}{\partial \mathrm{x}} \\ \dfrac{\partial \mathrm{v}(\xi,\eta)}{\partial \mathrm{y}} \\ \dfrac{\partial \mathrm{u}(\xi,\eta)}{\partial \mathrm{y}} + \dfrac{\partial \mathrm{v}(\xi,\eta)}{\partial \mathrm{x}} \end{bmatrix}$$

where $\xi, \eta \in \hat{\Omega}$, $\mathrm{x}, \mathrm{y} \in \Omega$ and $\mathrm{u}(\xi,\eta)$, $\mathrm{v}(\xi,\eta)$ are the displacements in the physical domain being searched from the parametric domain. These gradients can be derived in an implicit way using the chain rule

Eq. 3.2

$$\frac{\partial \mathrm{u}(\xi,\eta)}{\partial \mathrm{x}} = \frac{\partial \mathrm{u}(\xi,\eta)}{\partial \xi}\frac{\partial \xi}{\partial \mathrm{x}} + \frac{\partial \mathrm{u}(\xi,\eta)}{\partial \eta}\frac{\partial \eta}{\partial \mathrm{x}} = \begin{bmatrix} \dfrac{\partial \xi}{\partial \mathrm{x}} & \dfrac{\partial \eta}{\partial \mathrm{x}} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathrm{u}}{\partial \xi} \\ \dfrac{\partial \mathrm{u}}{\partial \eta} \end{bmatrix}$$

$$\frac{\partial \mathrm{v}(\xi,\eta)}{\partial \mathrm{y}} = \frac{\partial \mathrm{v}(\xi,\eta)}{\partial \xi}\frac{\partial \xi}{\partial \mathrm{y}} + \frac{\partial \mathrm{v}(\xi,\eta)}{\partial \eta}\frac{\partial \eta}{\partial \mathrm{y}} = \begin{bmatrix} \dfrac{\partial \xi}{\partial \mathrm{y}} & \dfrac{\partial \eta}{\partial \mathrm{y}} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \mathrm{v}}{\partial \xi} \\ \dfrac{\partial \mathrm{v}}{\partial \eta} \end{bmatrix}$$

$$\frac{\partial u(\xi,\eta)}{\partial y}+\frac{\partial v(\xi,\eta)}{\partial x}=\frac{\partial u(\xi,\eta)}{\partial \xi}\frac{\partial \xi}{\partial y}+\frac{\partial u(\xi,\eta)}{\partial \eta}\frac{\partial \eta}{\partial y}+\frac{\partial v(\xi,\eta)}{\partial \xi}\frac{\partial \xi}{\partial x}+\frac{\partial v(\xi,\eta)}{\partial \eta}\frac{\partial \eta}{\partial x}$$

$$=\begin{bmatrix}\frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x}\end{bmatrix}\begin{bmatrix}\frac{\partial u}{\partial \xi}\\ \frac{\partial u}{\partial \eta}\\ \frac{\partial v}{\partial \xi}\\ \frac{\partial v}{\partial \eta}\end{bmatrix}$$

Hence,

Eq. 3.3

$$\boldsymbol{\varepsilon}(\xi,\eta)=\begin{bmatrix}\frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & 0 & 0\\ 0 & 0 & \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y}\\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x}\end{bmatrix}\begin{bmatrix}\frac{\partial u}{\partial \xi}\\ \frac{\partial u}{\partial \eta}\\ \frac{\partial v}{\partial \xi}\\ \frac{\partial v}{\partial \eta}\end{bmatrix}$$

where

Eq. 3.4

$$\boldsymbol{B_1}(\xi,\eta)=\begin{bmatrix}\frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & 0 & 0\\ 0 & 0 & \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y}\\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x}\end{bmatrix}$$

This is the Jacobian matrix of physical to parametric domain mapping.

The vector of gradients of the displacements with respect to the parametric coordinates in Eq. 3.3 can be computed considering the inverse isoparametric concept (isogeometric). This is

Eq. 3.5

$$\begin{Bmatrix}u\\ v\end{Bmatrix}(\xi,\eta)=R_{i,j}(\xi,\eta)\cdot \boldsymbol{d_{i,j}}$$

Where $\boldsymbol{d_{i,j}}$ are the displacements of the control points.

And the derivatives

Eq. 3.6

$$\begin{bmatrix} \dfrac{\partial}{\partial \xi}\begin{Bmatrix} u \\ v \end{Bmatrix}(\xi,\eta) \\ \dfrac{\partial}{\partial \eta}\begin{Bmatrix} u \\ v \end{Bmatrix}(\xi,\eta) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial \xi}R_{i,j}(\xi,\eta) \\ \dfrac{\partial}{\partial \eta}R_{i,j}(\xi,\eta) \end{bmatrix} \cdot \boldsymbol{d_{i,j}}$$

Where the basis derivatives are the corresponding to the type of geometric basis (see Part I).

Entire the vector of gradients is

Eq. 3.7

$$\begin{bmatrix} \dfrac{\partial u(\xi,\eta)}{\partial \xi} \\ \dfrac{\partial u(\xi,\eta)}{\partial \eta} \\ \dfrac{\partial v(\xi,\eta)}{\partial \xi} \\ \dfrac{\partial v(\xi,\eta)}{\partial \eta} \end{bmatrix} = \begin{bmatrix} R_{1,u}(\xi,\eta) & 0 & R_{2,u}(\xi,\eta) & 0 & \cdots \\ R_{1,v}(\xi,\eta) & 0 & R_{2,v}(\xi,\eta) & 0 & \cdots \\ 0 & R_{1,u}(\xi,\eta) & 0 & R_{2,u}(\xi,\eta) & \cdots \\ 0 & R_{1,v}(\xi,\eta) & 0 & R_{2,v}(\xi,\eta) & \cdots \end{bmatrix} \begin{bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \\ d_{3x} \\ d_{3y} \\ \vdots \end{bmatrix}$$

where

Eq. 3.8

$$\boldsymbol{B_2}(\xi,\eta) = \begin{bmatrix} R_{1,u}(\xi,\eta) & 0 & R_{2,u}(\xi,\eta) & 0 & \cdots \\ R_{1,v}(\xi,\eta) & 0 & R_{2,v}(\xi,\eta) & 0 & \cdots \\ 0 & R_{1,u}(\xi,\eta) & 0 & R_{2,u}(\xi,\eta) & \cdots \\ 0 & R_{1,v}(\xi,\eta) & 0 & R_{2,v}(\xi,\eta) & \cdots \end{bmatrix}$$

Substituting in Eq. 3.3 it is

Eq. 3.9

$$\boldsymbol{\varepsilon}(\xi,\eta) = \begin{bmatrix} \dfrac{\partial u(\xi,\eta)}{\partial x} \\ \dfrac{\partial v(\xi,\eta)}{\partial y} \\ \dfrac{\partial u(\xi,\eta)}{\partial y} + \dfrac{\partial v(\xi,\eta)}{\partial x} \end{bmatrix} = \boldsymbol{B_1}(\xi,\eta) \cdot \ \boldsymbol{B_2}(\xi,\eta) \cdot \boldsymbol{d_{i,j}}$$

$$= \begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \eta}{\partial x} & 0 & 0 \\ 0 & 0 & \dfrac{\partial \xi}{\partial y} & \dfrac{\partial \eta}{\partial y} \\ \dfrac{\partial \xi}{\partial y} & \dfrac{\partial \eta}{\partial y} & \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \eta}{\partial x} \end{bmatrix} \begin{bmatrix} R_{1,u}(\xi,\eta) & 0 & R_{2,u}(\xi,\eta) & 0 & \cdots \\ R_{1,v}(\xi,\eta) & 0 & R_{2,v}(\xi,\eta) & 0 & \cdots \\ 0 & R_{1,u}(\xi,\eta) & 0 & R_{2,u}(\xi,\eta) & \cdots \\ 0 & R_{1,v}(\xi,\eta) & 0 & R_{2,v}(\xi,\eta) & \cdots \end{bmatrix} \begin{bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \\ d_{3x} \\ d_{3y} \\ \vdots \end{bmatrix}$$

To compute the $\boldsymbol{B_1}$ matrix we need to compute the Jacobian matrix of the physical to parametric domain mapping. This can be done by inverting the parametric to the physical domain mapping that is expressed by the Jacobian matrix J.

Eq. 3.10

$$J(\xi,\eta)=\begin{bmatrix}\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi}\\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta}\end{bmatrix}$$

This mapping is done by a geometric basis as is presented in Part I

Eq. 3.11

$$\begin{Bmatrix}x\\y\end{Bmatrix}(\xi,\eta)=R_{i,j}(\xi,\eta)\cdot\boldsymbol{P_{i,j}}$$

For this mapping the gradients in physical domain are being calculated easily with respect to the coordinates of the parametric domain. That is

$$\begin{bmatrix}\frac{\partial}{\partial u}\begin{Bmatrix}x\\y\end{Bmatrix}(\xi,\eta)\\ \frac{\partial}{\partial v}\begin{Bmatrix}x\\y\end{Bmatrix}(\xi,\eta)\end{bmatrix}=\begin{bmatrix}\frac{\partial}{\partial \xi}R_{i,j}(\xi,\eta)\\ \frac{\partial}{\partial \eta}R_{i,j}(\xi,\eta)\end{bmatrix}\cdot\boldsymbol{P_{i,j}}$$

The components of the J^{-1} are

$$\frac{\partial \xi}{\partial x}=\frac{1}{detJ}\frac{\partial y}{\partial \eta}$$

$$\frac{\partial \eta}{\partial x}=-\frac{1}{detJ}\frac{\partial y}{\partial \xi}$$

$$\frac{\partial \xi}{\partial y}=-\frac{1}{detJ}\frac{\partial x}{\partial \eta}$$

$$\frac{\partial \eta}{\partial y}=\frac{1}{detJ}\frac{\partial x}{\partial \xi}$$

Therefore,

$$\boldsymbol{B_1}(\xi,\eta)=\begin{bmatrix}\frac{1}{detJ}\frac{\partial y}{\partial \eta} & -\frac{1}{detJ}\frac{\partial y}{\partial \xi} & 0 & 0\\ 0 & 0 & -\frac{1}{detJ}\frac{\partial x}{\partial \eta} & \frac{1}{detJ}\frac{\partial x}{\partial \xi}\\ -\frac{1}{detJ}\frac{\partial x}{\partial \eta} & \frac{1}{detJ}\frac{\partial x}{\partial \xi} & \frac{1}{detJ}\frac{\partial y}{\partial \eta} & -\frac{1}{detJ}\frac{\partial y}{\partial \xi}\end{bmatrix}$$

We have computed the deformation matrix $\boldsymbol{B}(\xi,\eta) = \boldsymbol{B_1}(\xi,\eta) \cdot \boldsymbol{B_2}(\xi,\eta)$, so we can now proceed with the computation of the stiffness matrix using Eq. 2.54

$$K = \sum_{m} \int_{V^{(m)}} \boldsymbol{B}^{(\boldsymbol{m})^T} \boldsymbol{D}^{(\boldsymbol{m})} \boldsymbol{B}^{(\boldsymbol{m})} \, dV^{(m)}$$

$$= \sum_{m} \int_{-1}^{1}\int_{-1}^{1} \left[\boldsymbol{B}(\xi,\eta)^{(\boldsymbol{m})}\right]^T \boldsymbol{D}^{(\boldsymbol{m})} \boldsymbol{B}(\xi,\eta)^{(\boldsymbol{m})} \det[J_1]\det[J_2] \, t d\xi d\eta$$

$$= \sum_{m}^{n_{el}} \sum_{(\xi,\eta)\in P_g}^{n_{g.p.}} \left[\boldsymbol{B}(\xi,\eta)^{(\boldsymbol{m})}\right]^T \boldsymbol{D}^{(\boldsymbol{m})} \boldsymbol{B}(\xi,\eta)^{(\boldsymbol{m})} w_{\tilde{\xi}} w_{\tilde{\eta}} \det[J_1]\det[J_2] \, t d\xi d\eta$$

Where $\det[J_1]$ is the determinant of the Jacobian matrix of parent to parametric domain mapping (to convert the weights $w_{\tilde{\xi}} w_{\tilde{\eta}}$ used in integration that form the area of the parent domain) and $\det[J_2]$ is the determinant of the Jacobian matrix of parametric to physical domain mapping (to convert the stiffness to the physical domain); t denotes the constant width of the element. If the width varies in the element, then we use the formulation

$$dV = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{vmatrix} d\xi d\eta d\zeta = \det[J_1]\det[J_2] d\tilde{\xi} d\tilde{\eta} d\tilde{\zeta}$$

The inner summation is over the Gauss points P_g selected in order to compute the integral.

3.2 DERIVATION OF THE BODY & TRACTION FORCES VECTOR

$$F_B = \sum_{m} \int_{V^{(m)}} \boldsymbol{R}^{(m)^T} \boldsymbol{f}^{\boldsymbol{B}(\boldsymbol{m})}\, dV^{(m)} =$$

$$= \sum_{m} \int_{V^{(m)}} \begin{bmatrix} f_x^{B(m)}(\xi,\eta) \\ f_y^{B(m)}(\xi,\eta) \end{bmatrix} [R_{11}(\xi,\eta) \quad R_{12}(\xi,\eta) \quad \dots \quad R_{nm}(\xi,\eta)]\, dV^{(m)}$$

$$= \sum_{m} \sum_{(\xi,\eta)\in P_g} \begin{bmatrix} f_x^{B(m)}(\xi,\eta) R_{\mathbf{11}}(\xi,\eta) \\ f_y^{B(m)}(\xi,\eta) R_{\mathbf{11}}(\xi,\eta) \\ f_x^{B(m)}(\xi,\eta) R_{12}(\xi,\eta) \\ f_y^{B(m)}(\xi,\eta) R_{12}(\xi,\eta) \\ \vdots \end{bmatrix} w_{\tilde{\xi}} w_{\tilde{\eta}} \det[J_1] \det[J_2]\, d\xi d\eta$$

The traction forces are computed in the same way by integrating in one dimension that is the corresponding boundaries of the referred elements.

3.3 DERIVATION OF THE MASS MATRIX

Eq. 3.12

$$M = \sum_{m} \int_{V^{(m)}} \rho\, \boldsymbol{R}^{(m)} \boldsymbol{R}^{(m)^T}\, dV^{(m)} =$$

$$= \sum_{m} \int_{V^{(m)}} \rho \begin{bmatrix} R_{11}(\xi,\eta) \\ R_{12}(\xi,\eta) \\ R_{13}(\xi,\eta) \\ R_{14}(\xi,\eta) \\ \vdots \end{bmatrix} [R_{11}(\xi,\eta) \quad R_{12}(\xi,\eta) \quad \dots \quad R_{nm}(\xi,\eta)]\, dV^{(m)}$$

$$= \sum_{m} \sum_{(\xi,\eta)\in P_g} \rho \begin{bmatrix} R_{11}(\xi,\eta) \\ R_{12}(\xi,\eta) \\ R_{13}(\xi,\eta) \\ R_{14}(\xi,\eta) \\ \vdots \end{bmatrix} [R_{11}(\xi,\eta) \quad R_{12}(\xi,\eta) \quad \dots \quad R_{nm}(\xi,\eta)] w_{\tilde{\xi}} w_{\tilde{\eta}} \det[J_1] \det[J_2]\, d\xi d\eta$$

4 APPLICATIONS ON LINEAR ISOGEOMETRIC ANALYSIS

For the solution of an Isogeometric problem we have to specify the domain of the problem by its exact geometry and any other geometry refinements necessary for a good analysis model without degrading the exactness of the geometry.

We first decide on how to represent the geometry of the problem in the simplest and exact way. We must specify the degree of the basis and the knot vectors according to the complexity of the geometry. Specifically, curved edges need a higher order of basis, whereas for those complex shapes that require local control (basis functions with local support) internal knots are helpful (once no local control is requisite, a Bezier geometry is constructed without internal knots). Furthermore, the minimum number of required control points and their positions have to be specified, to exactly represent the desired geometry. Finally, when the NURBS or the rational Bezier method is used, the weights of the control points have to be specified to represent a specific conic section (see Part I section 8.4).

Further refinement of the Isogeometric problem is made for the sake of analysis. Any refinement of the geometry can be implemented as needed, h-refinement, p-refinement or k-refinement or a combination of them without the change of the geometry.

4.1 LINEAR ELASTIC ANALYSIS OF A CANTILEVER BEAM

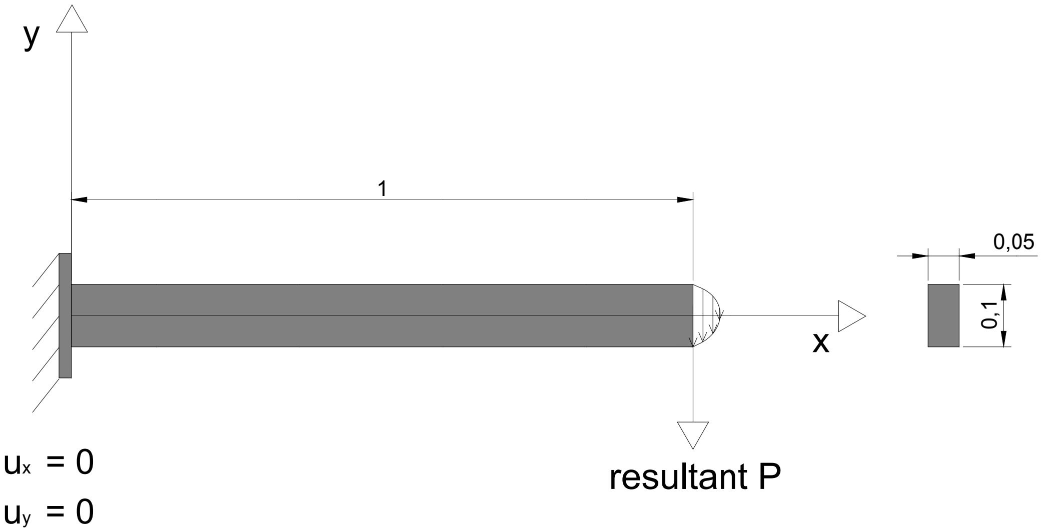


Figure 4.1

We assume a rigidly fixed cantilever beam consisted of plane – stress elements, carrying a vertical tip load. The elastic properties of the material are **E** = 210GPa, v = 0.3. The dimensions of the beam are $L \times H \times W = 1\,m \times 0.1\,m \times 0.05\,m$, whereas the applied tip force of 1 kN is applied as a parabolic distribution on the section. In all occasions the fixed end of the beam is achieved by constraining all the degrees of freedom of the control points that are on that boundary line.

We perform a linear analysis of the cantilever beam based on B-Splines and compare the stress, strain and displacement fields with the analytical data computed according to the classical theory of elasticity as:

$$\sigma_{xx} = -\frac{P}{I}xy \quad \sigma_{yy} = 0 \quad \sigma_{xy} = \frac{P}{2I}(y^2 - \frac{h^2}{4})$$

where $I = \frac{bh^3}{12}$, b denotes the width and h the height of the beam

$$u = \frac{3Px^2y}{4E\left(\frac{h}{2}\right)^3} + \frac{3P(1+\nu)y}{2E\frac{h}{2}} - \frac{P(2+\nu)y^3}{4E\left(\frac{h}{2}\right)^3} - \frac{3Pl^2}{4E\left(\frac{h}{2}\right)^3}y$$

$$v = -\frac{3P\nu xy^2}{4E\left(\frac{h}{2}\right)^3} - \frac{P(1+\nu)x^3}{4E\left(\frac{h}{2}\right)^3} - \frac{Pl^3}{2E\left(\frac{h}{2}\right)^3} + \frac{3Pl^2}{4E\left(\frac{h}{2}\right)^3}x$$

where l denotes the length of the beam.

Now turning to the implementation of the isogeometric method to this problem, we first decide that a first degree basis in both parametric directions is adequate to describe this simple linear geometry. Moreover, no internal knots are needed to control locally the geometry. Having in hand the degree of the basis and the knot vectors we can place the control points to shape the rectangular beam. For this first degree basis and the globally controlled geometry, four control points placed at the corners are adequate see Figure 4.2. All of the weights corresponding to the NURBS basis are set to unit as there are no conic sections to be represented. Due to this fact, the NURBS basis degenerates into a B-Spline basis. Furthermore, the basis degenerates further into a Bezier geometry basis (Bernstein polynomials) since only one knot span has been defined in each direction.

For the sake of analysis the previous Bezier geometry is refined. Using a single generator of basis functions as we refine the geometry by knot insertion it can produce a corresponding B-Spline basis and the necessary knot spans. Various parametric analysis are made to examine the efficiency and stability of the method by hp refinement in addition to the examination of the numerical integration. All these parametric analysis are presented in subsequent sections by graphs of $L^2(\Omega)$ and energy norms. In this section some results of the plane stress analysis are presented for a qualitative assessment. These results are for 4th degree (in both directions) bivariate basis functions see Figure 4.3 through Figure 4.13.

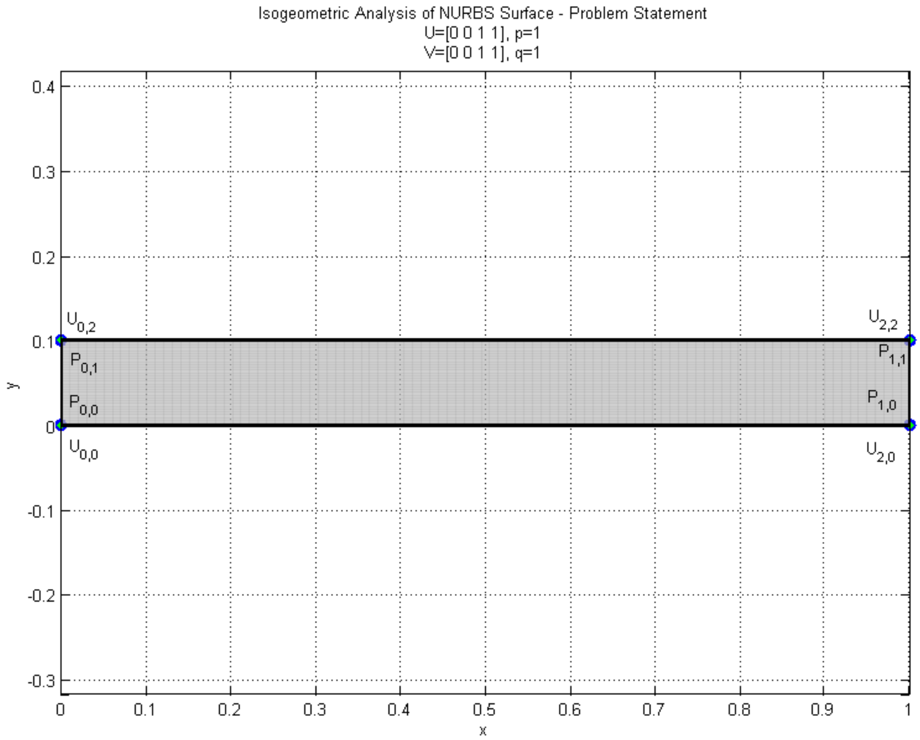


Figure 4.2 The geometry of the beam in its most simple structure. One non-zero knot span in each direction (one element totally) and one control point at each corner for linear interpolation.

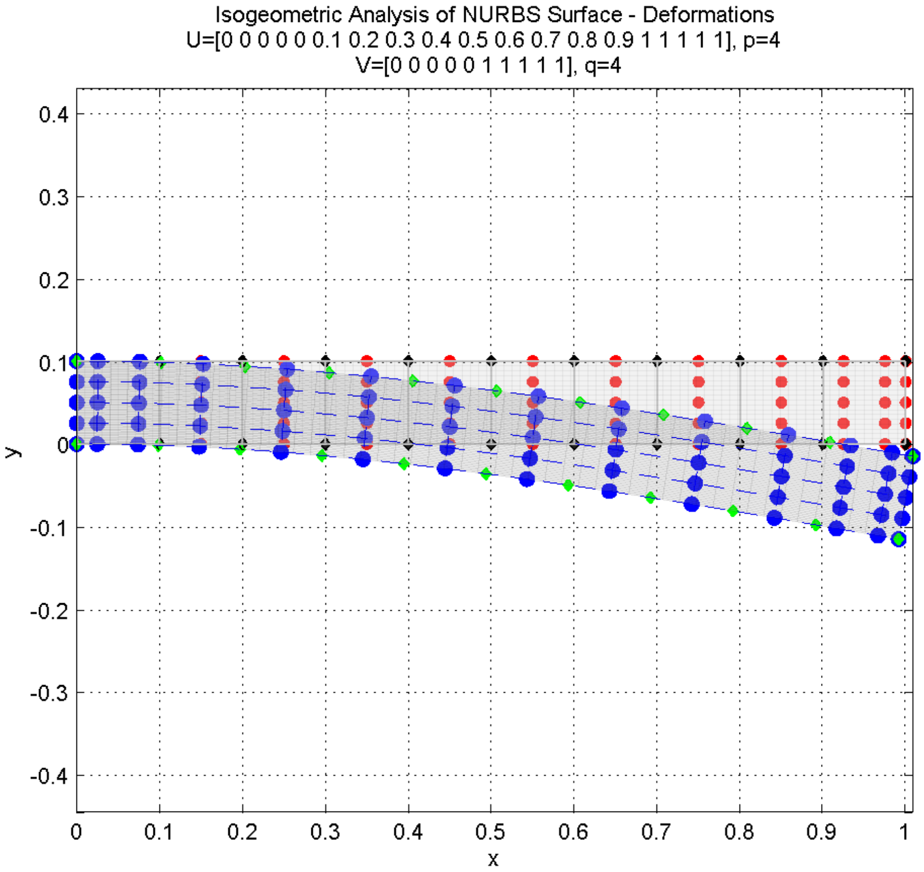


Figure 4.3 The deformed shape of the beam

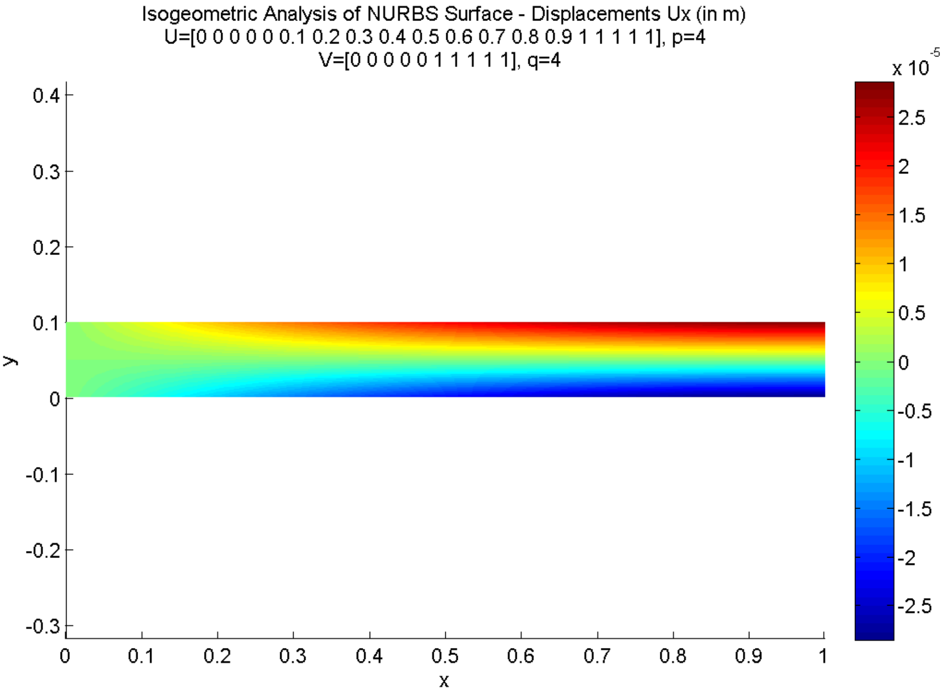


Figure 4.4

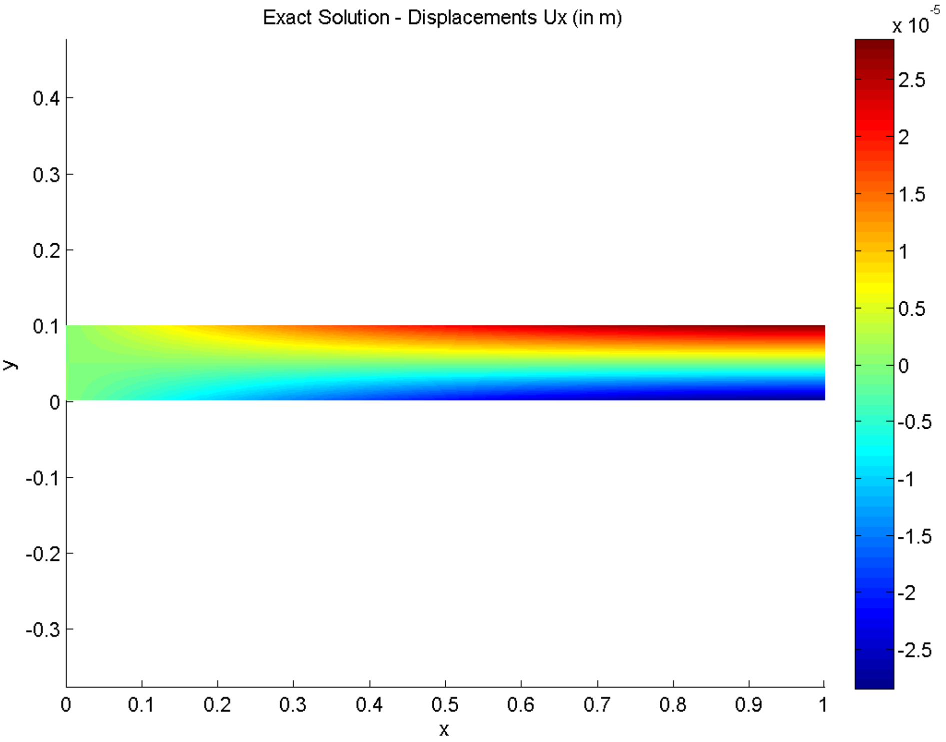


Figure 4.5

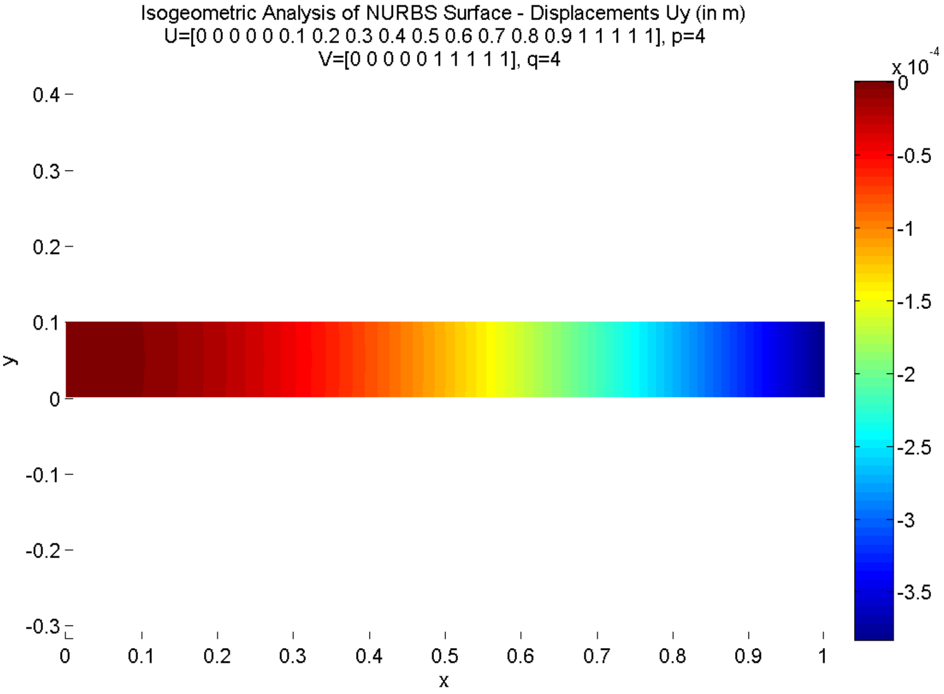


Figure 4.6

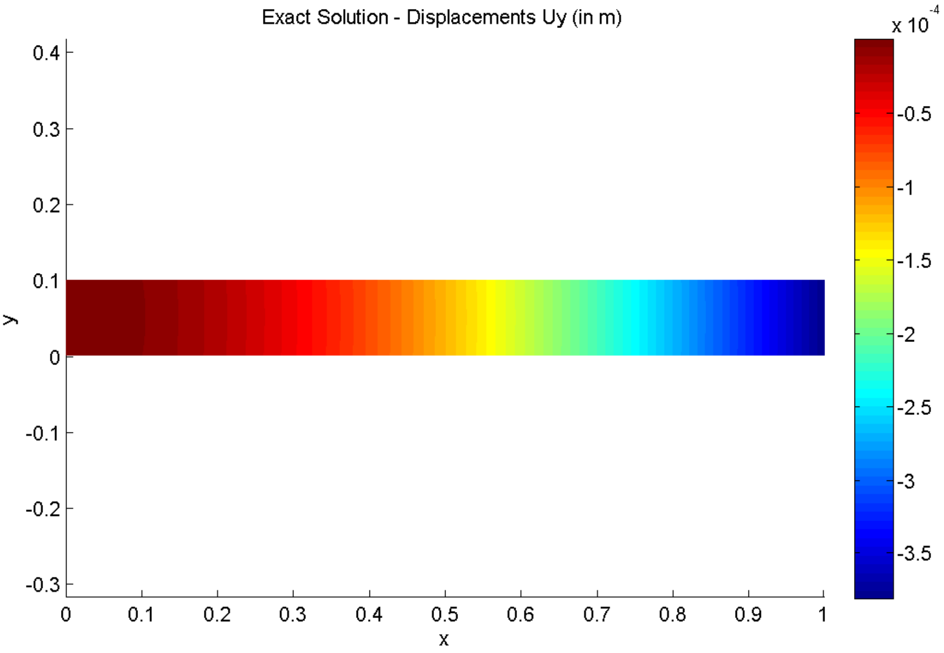


Figure 4.7

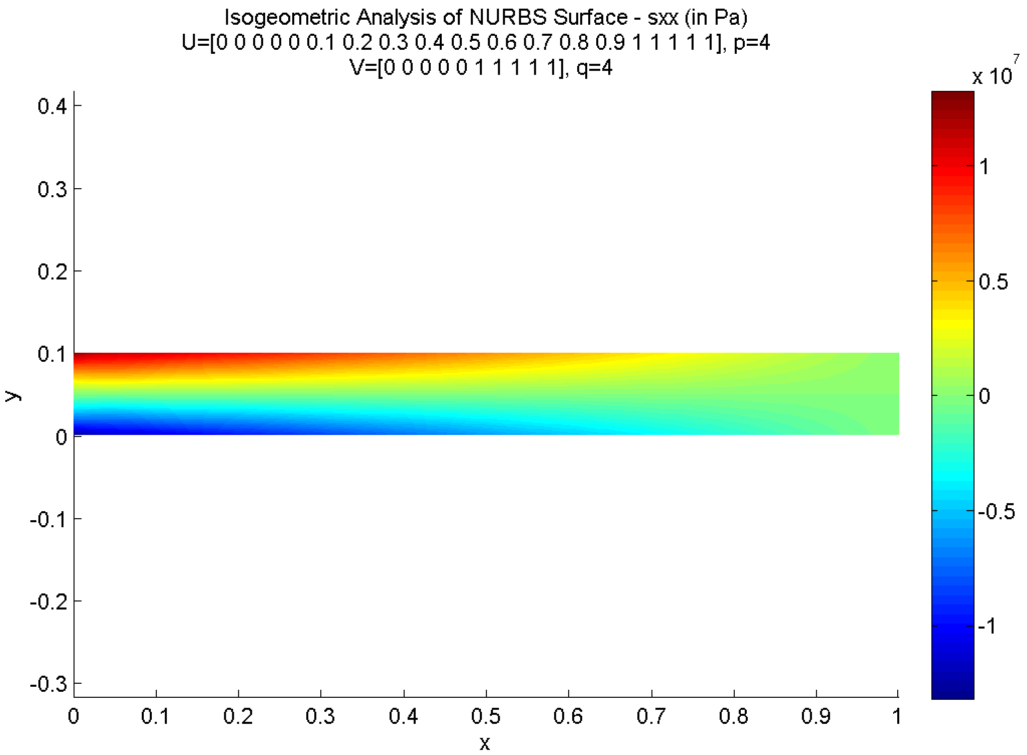


Figure 4.8

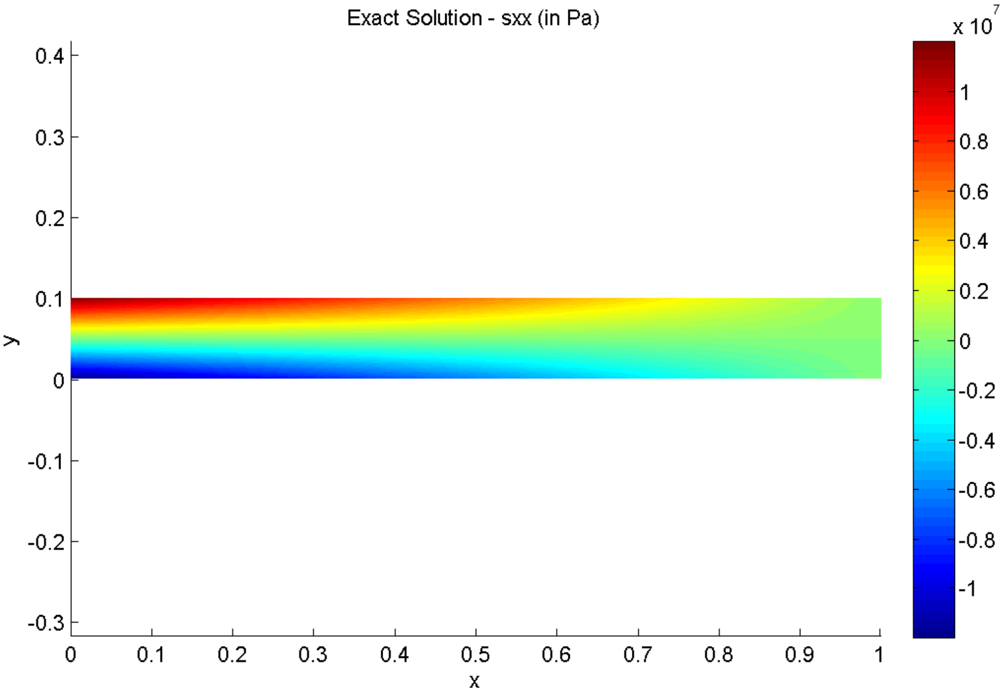


Figure 4.9

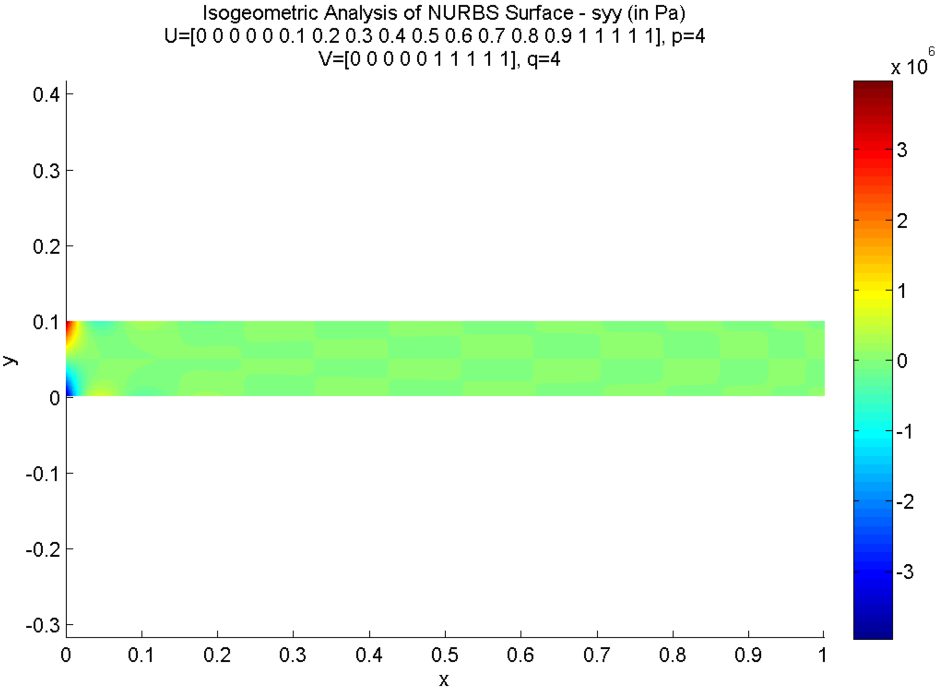


Figure 4.10

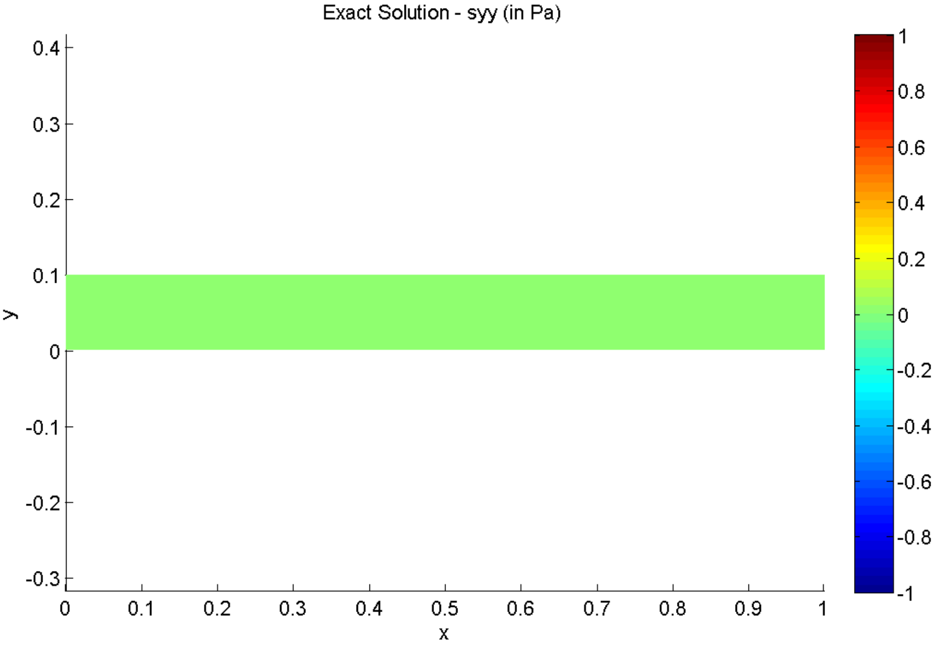


Figure 4.11

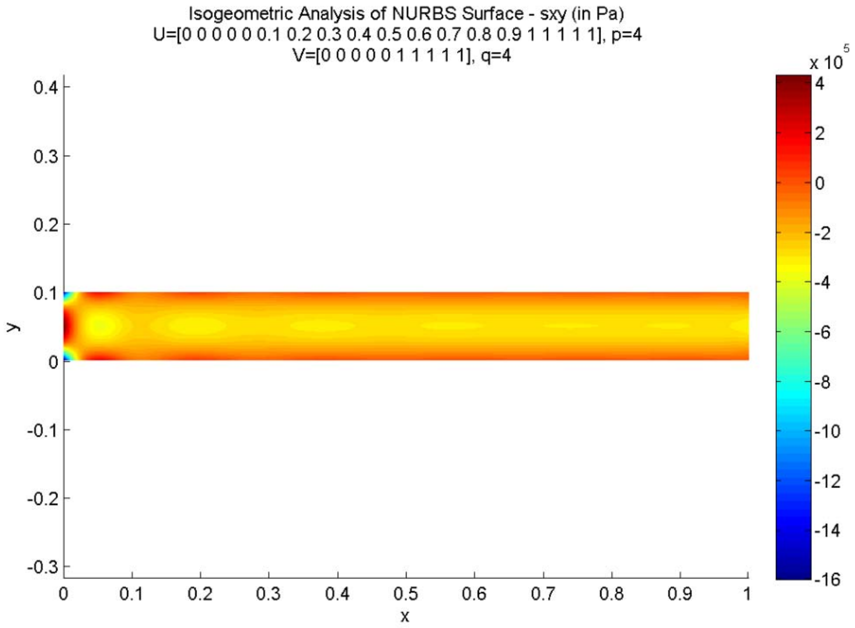


Figure 4.12

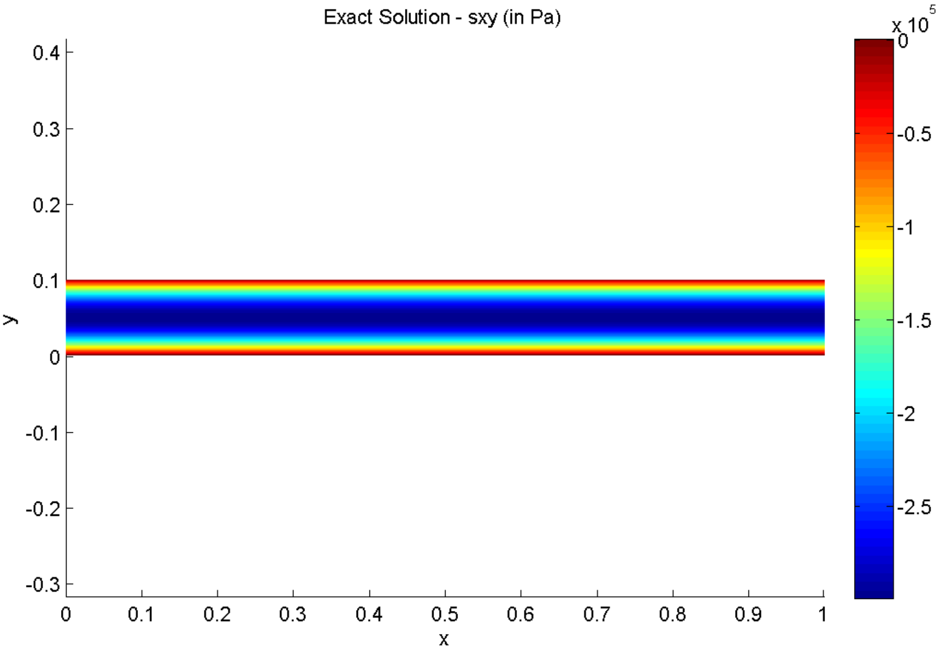


Figure 4.13

We realize that the results are quite accurate for this very coarse mesh (1x10 elements). Even the shear stresses sxy and the vertical syy are very close to being accurate excluding the region near the support that indicates the difference in considered boundary conditions.

4.2 EXAMINATION OF THE NUMERICAL INTEGRATION ACCURACY BY THE GAUSS – LEGENDRE QUADRATURE RULE

This examination starts from the 4th degree basis functions and ends at the 2nd degree basis. The discretization (knot spans) is the same in all the cases, 1x10 elements. The $L^2(\Omega)$ (relative and absolute) norms, for the displacement error field, the stress error field, the normal stress error field and the energy norms are plotted in each degree of the basis functions.

The considered parameter in these analyses is the number of the gauss quadrature points in each element that will try to approximate the field by Legendre polynomials. The lower number of gauss points is derived by the known rule that the exact integration of the basis functions is achieved by the rule n=p+1; where n is the number of gauss points in the referred direction and p is the degree.

4.2.1 INTEGRATION OF 4TH DEGREE BASIS FUNCTIONS

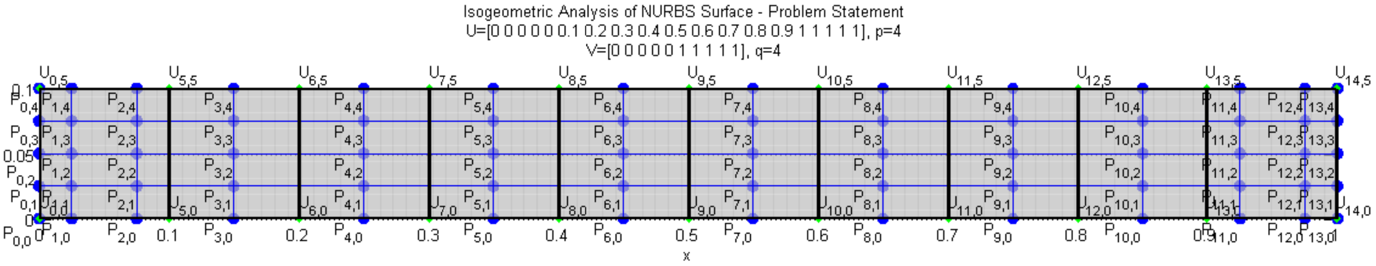


Figure 4.14 The problem that is considered for examination of numerical integration accuracy (blue dots represents the control points, the thick black lines the element borders a.k.a. knot lines, and the green diamonds are the cross-points of the knot lines)

Our first result is that an inadequate number of gauss points results in a highly distorted geometry that fails to represent the real deformed geometry see Figure 4.17; thus the approximation of the displacement field is too bad. After trying a number of combinations of gauss points along u and v direction, we observed that the deformed geometry is a good approximation of the solution field was reached for a 3 × 4 net of gauss points per element. These qualitative results are confirmed by graphs indicating the various norms that mentioned above at each case. In each graph one can observe that the exact integration is achieved when the corresponding norm remains unchanged while increasing the number of integration points.

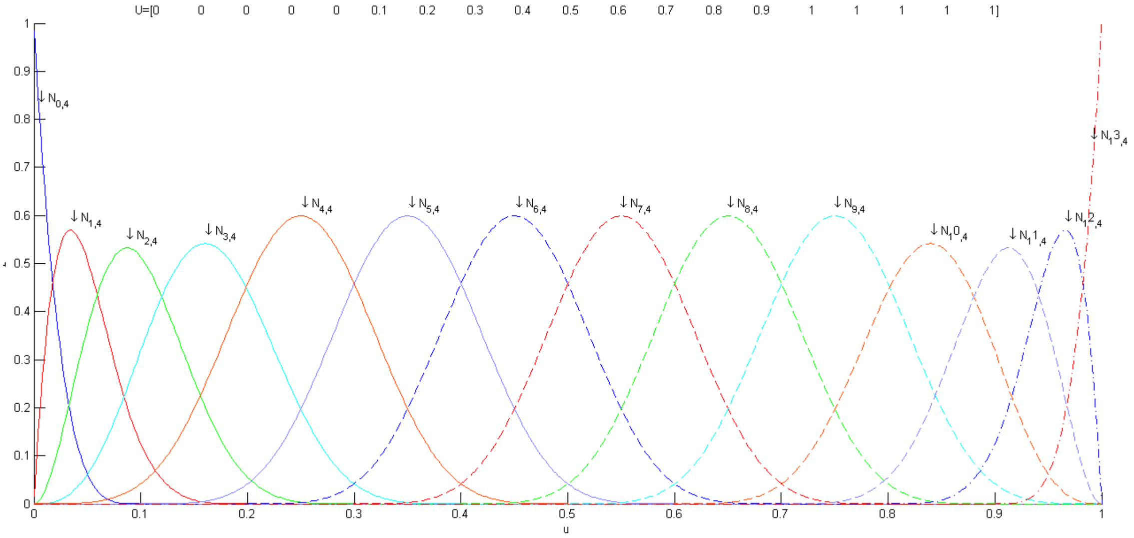


Figure 4.15 The basis functions in u direction of 4th degree

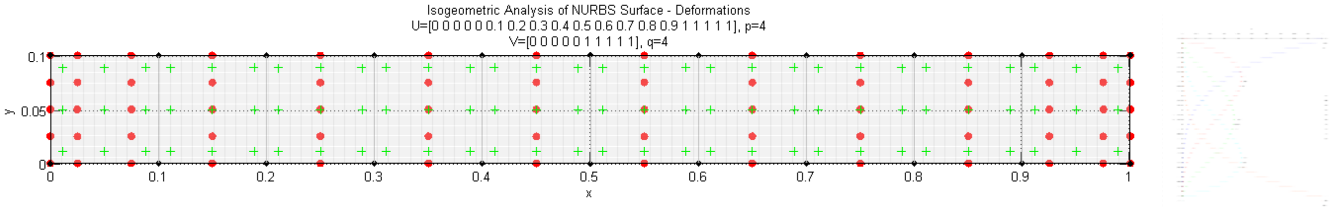
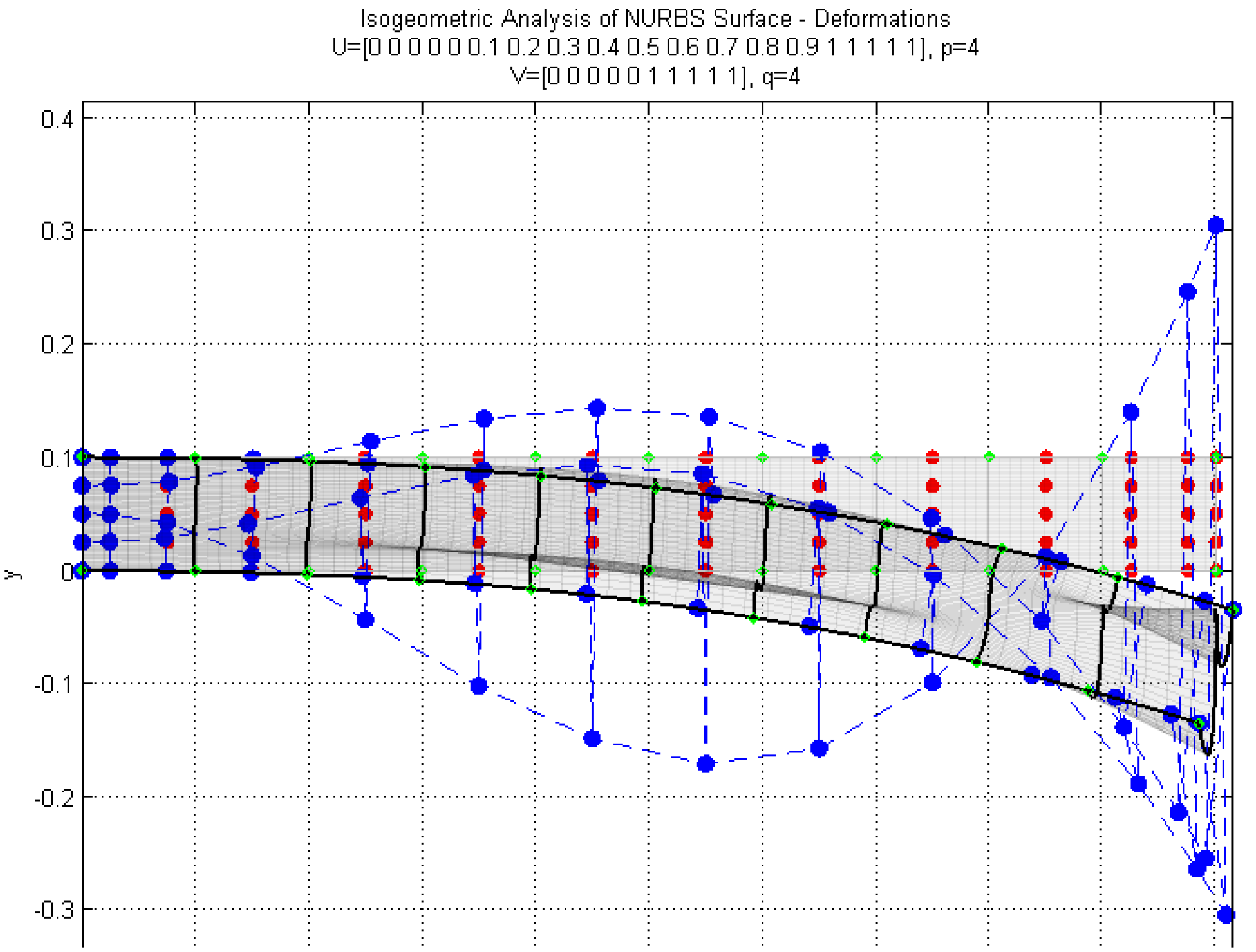
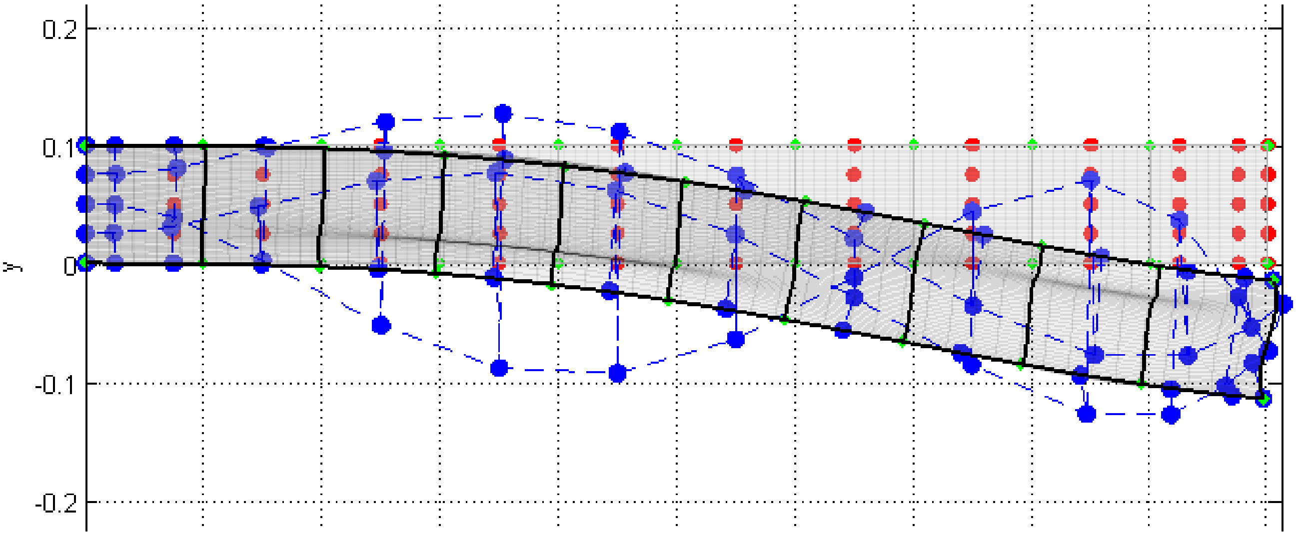


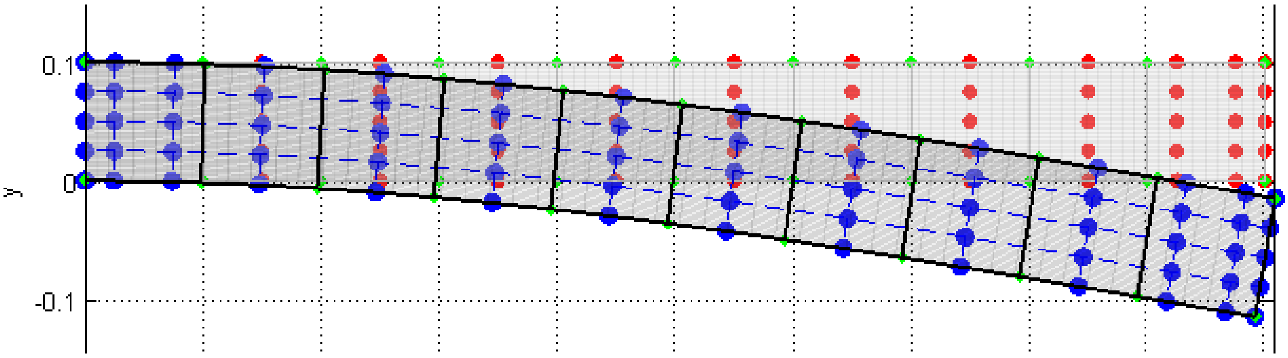
Figure 4.16 The mapped positions of the 3x3 g.q.p. (the green crosses are the gauss points, the red dots are the control points)


Isogeometric Analysis of NURBS Surface - Deformations
U=[0 0 0 0 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1 1 1 1], p=4
V=[0 0 0 0 0 1 1 1 1 1], q=4
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
y

a)


0.2
0.1
0
-0.1
-0.2
y

b)



c)

Figure 4.17 Deformations(x300 scaled) under 4^{th}x4^{th} degree basis and a) 3x3 g.q.p. per element b) 4x3 g.q.p. per element b) 3x4 g.q.p.

On the graphs of the results one can decode the information in the title of each figure according to this legend:

eldof: degrees of freedom of element (total in x and y directions)

sdof: total degrees of freedom of the system (total in x and y directions)

h: maximum size of element

g.q.p.: gauss quadrature points in two directions of the plane element (ξ-dir x η-dir)

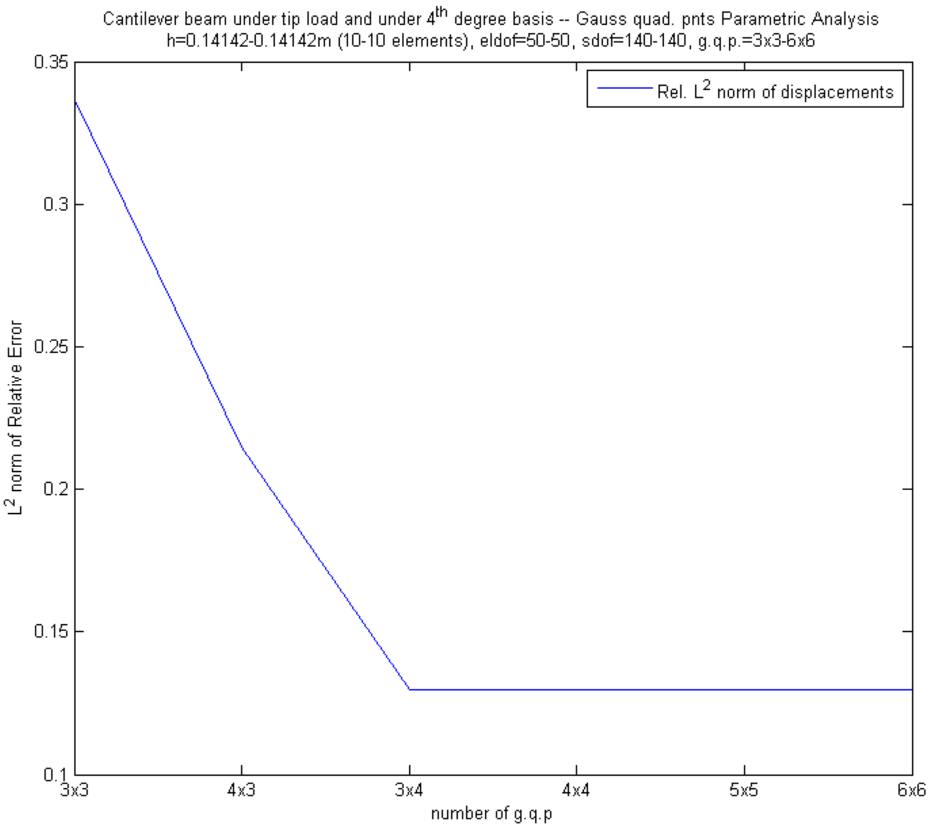


Figure 4.18 Relative norm of displacement field under the variation of the number of g.q.p.

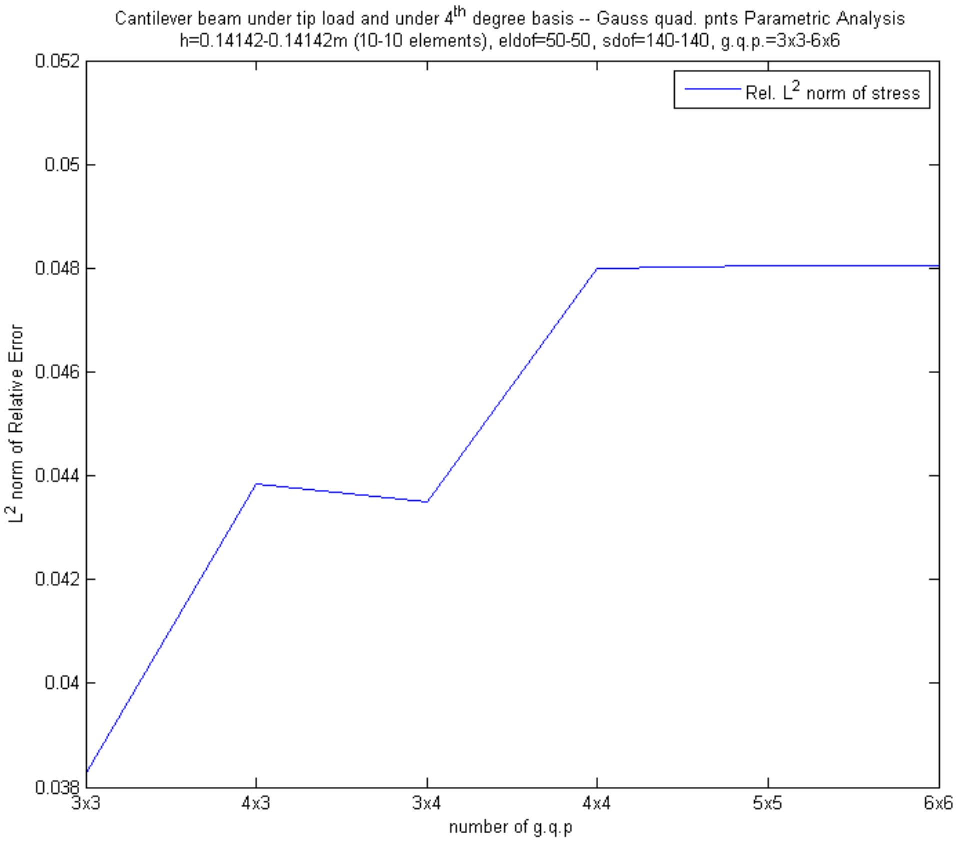


Figure 4.19 Relative norm of stress field under the variation of the number of g.q.p.

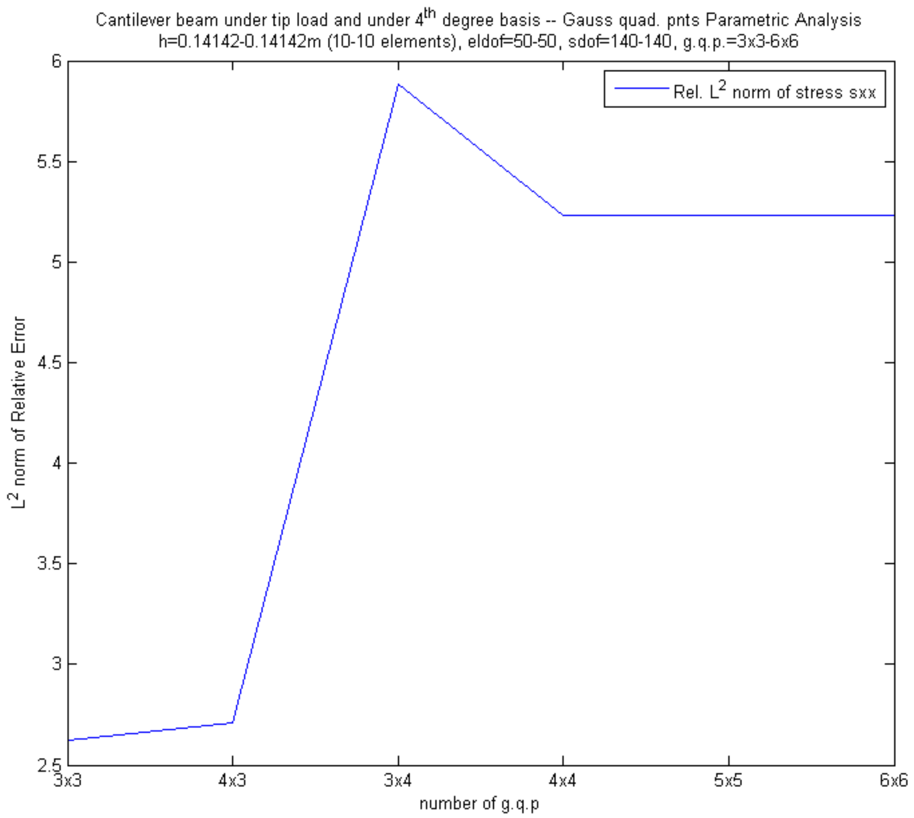


Figure 4.20 Relative norm of sxx stress field under the variation of the number of g.q.p.

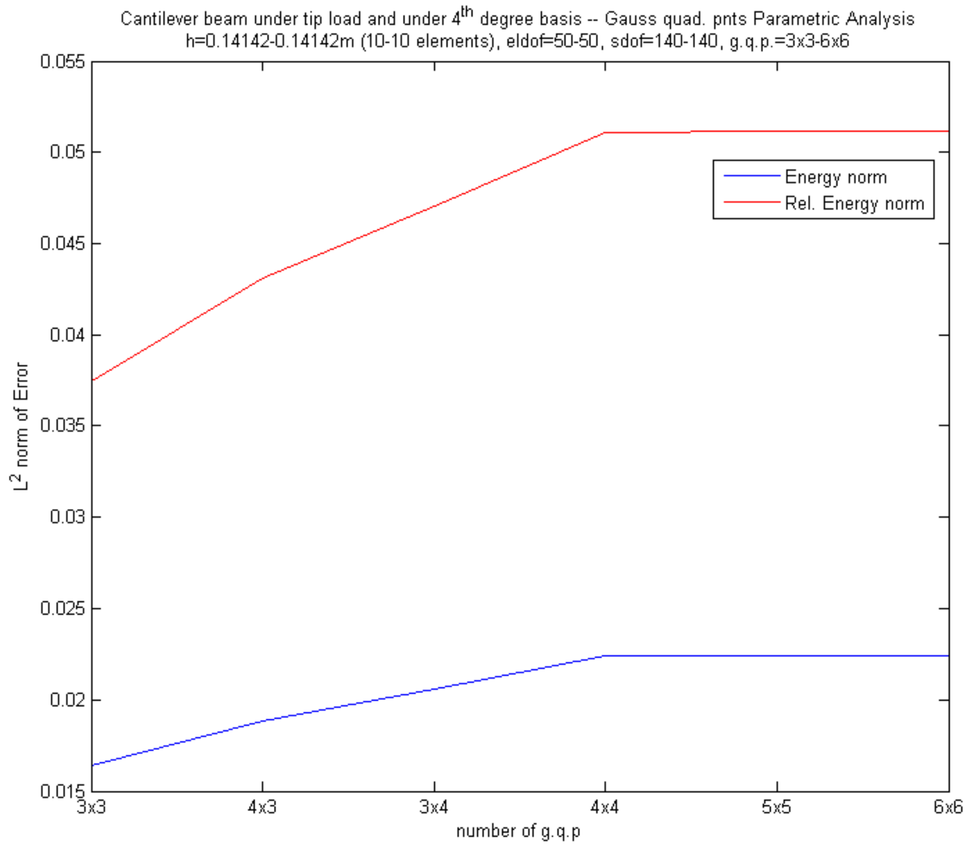


Figure 4.21 Energy norms under the variation of the number of g.q.p.

4.2.2 INTEGRATION OF 3RD DEGREE BASIS FUNCTIONS

Now, let's examine the case of assuming 3rd degree basis functions in both directions Figure 4.22 depicts the deformation of the beam scaled-up 300 times. The net of gauss points per element is of size 2×3. The outcome of our research on gauss points was analogous to that obtained in the case of the 4^{th} degree basis functions; for a 2×2 or 3×2 net, the geometry of the deformed beam appears to be highly distorted.

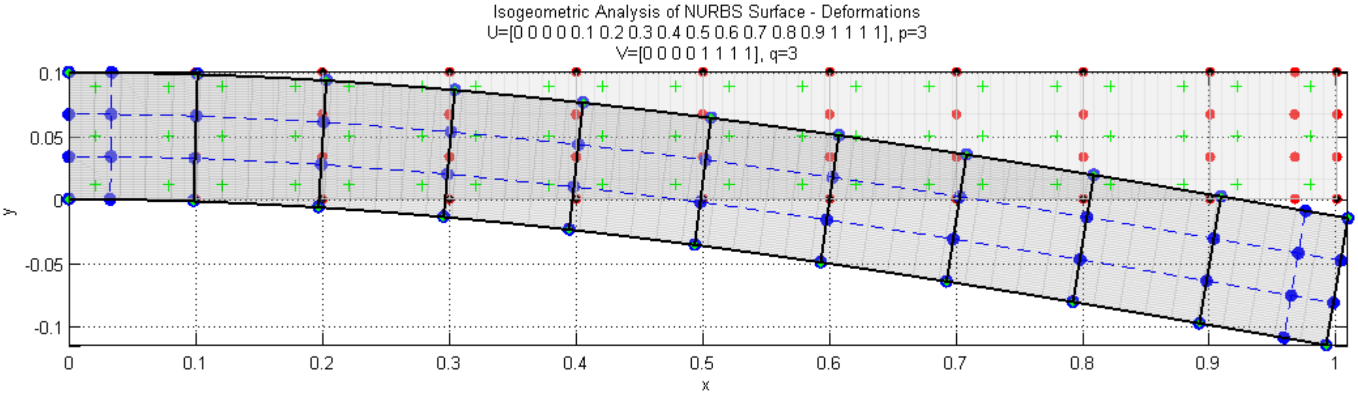


Figure 4.22 Deformations(x300 scaled) under 3^{rd}x3^{rd} degree basis and 2x3 g.q.p. All the lower occasions of number of gauss points (2x2 & 3x2) yield a very high distorted geometry that does not correspond at all to the prospective one.

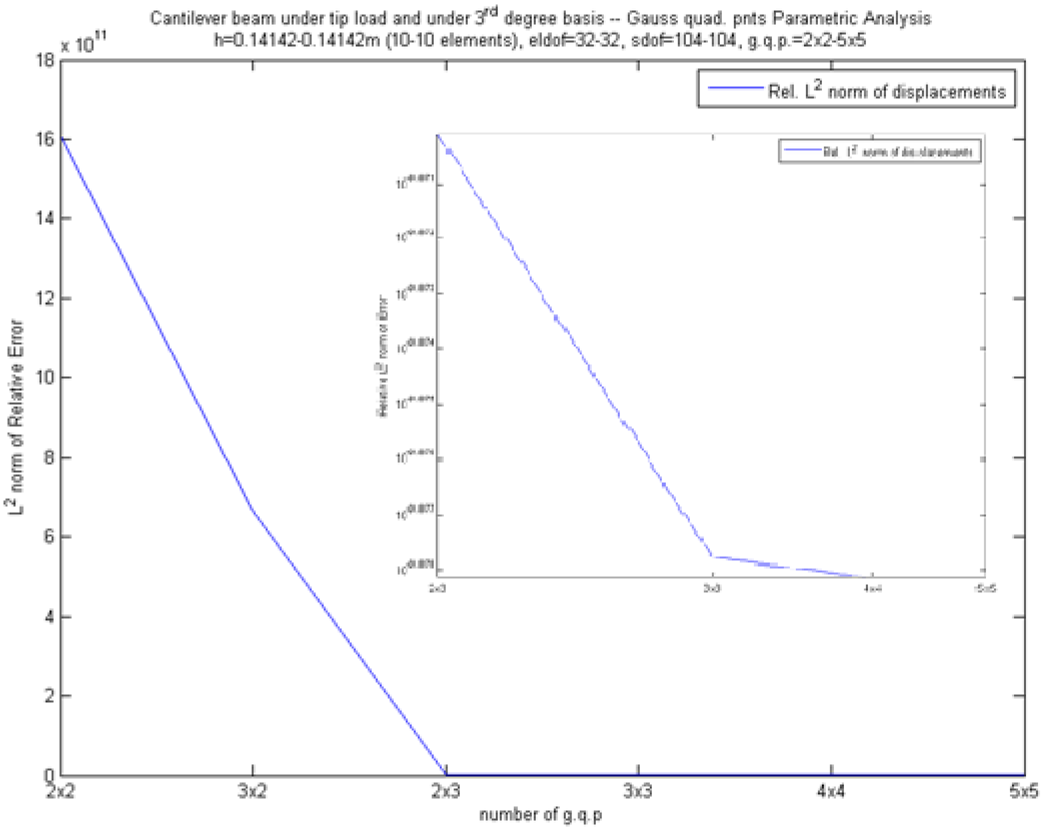


Figure 4.23

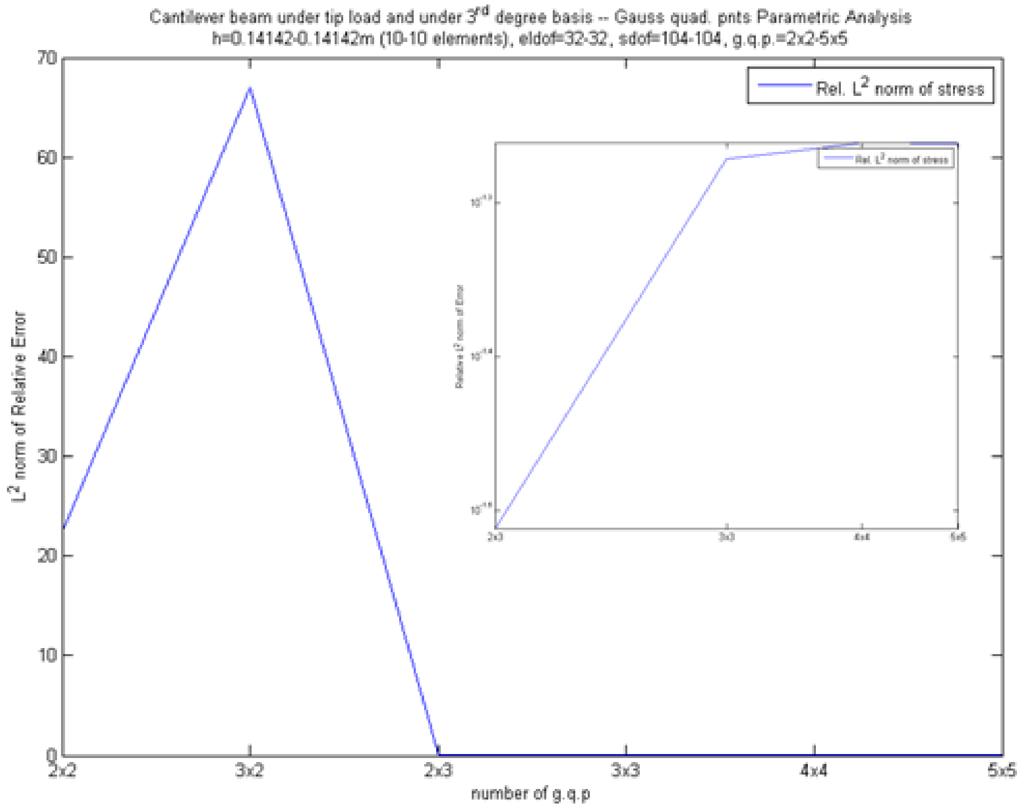


Figure 4.24

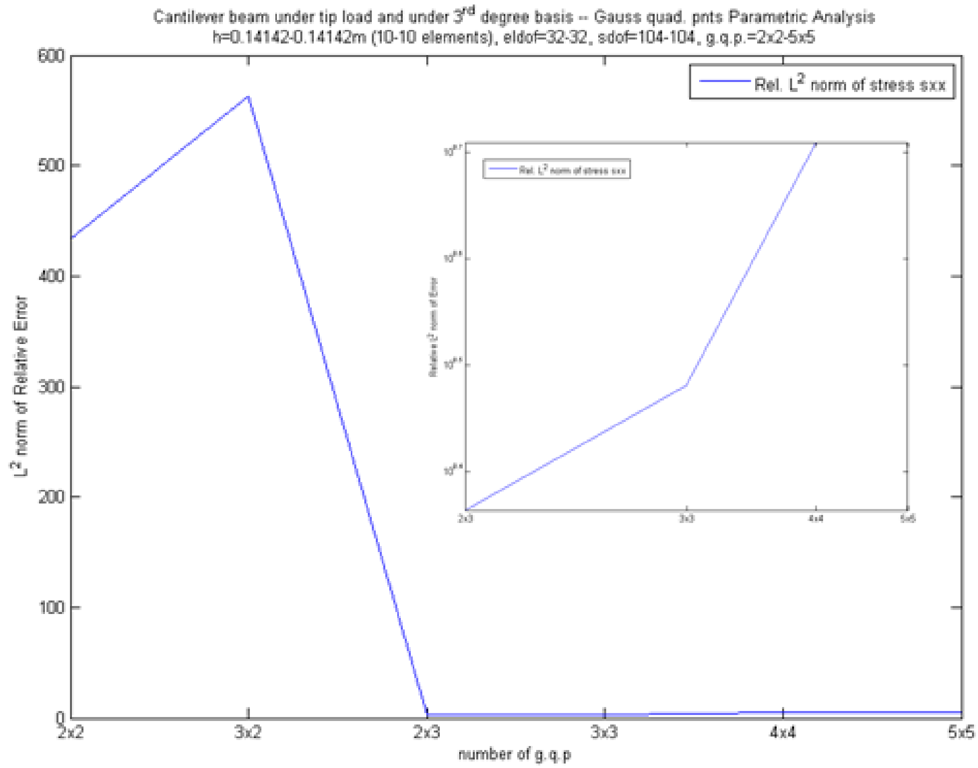


Figure 4.25

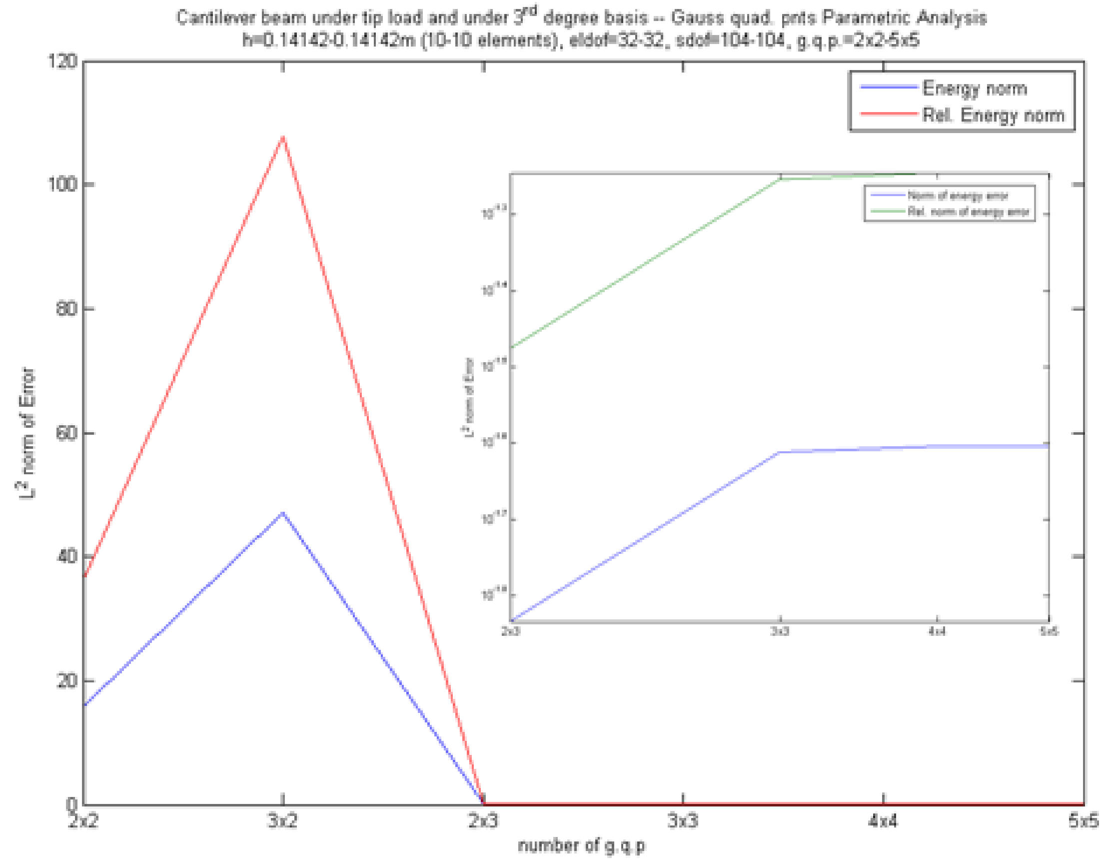


Figure 4.26

4.2.3 INTEGRATION OF 3^{RD} DEGREE BASIS FUNCTIONS OF A TENSION PROBLEM

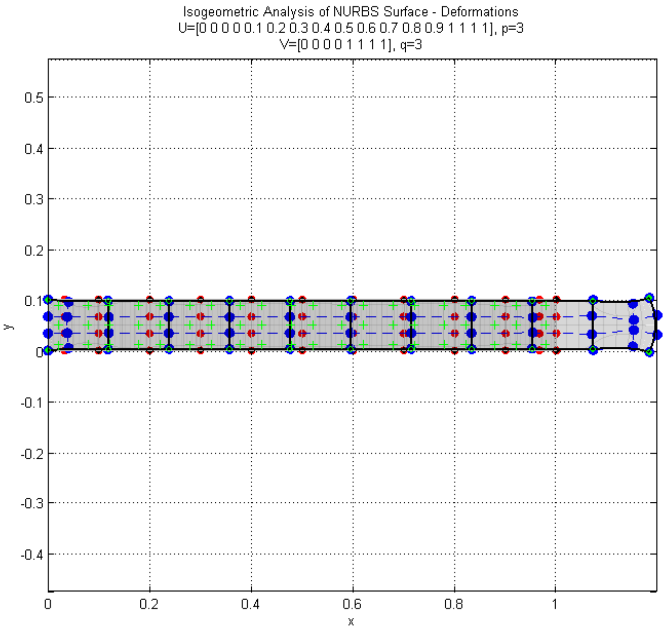


Figure 4.27 A bar with the same material properties and geometry as the fixed beam, under parabolic distribution of tension at the tip

To make sure that these observations of numerical integration efficiency is not problem dependent we render a problem of a bar under tension. Using the same material properties and the same geometry as that of the beam analysed we realize that the observations are not problem dependent. The corresponding results are plotted below.

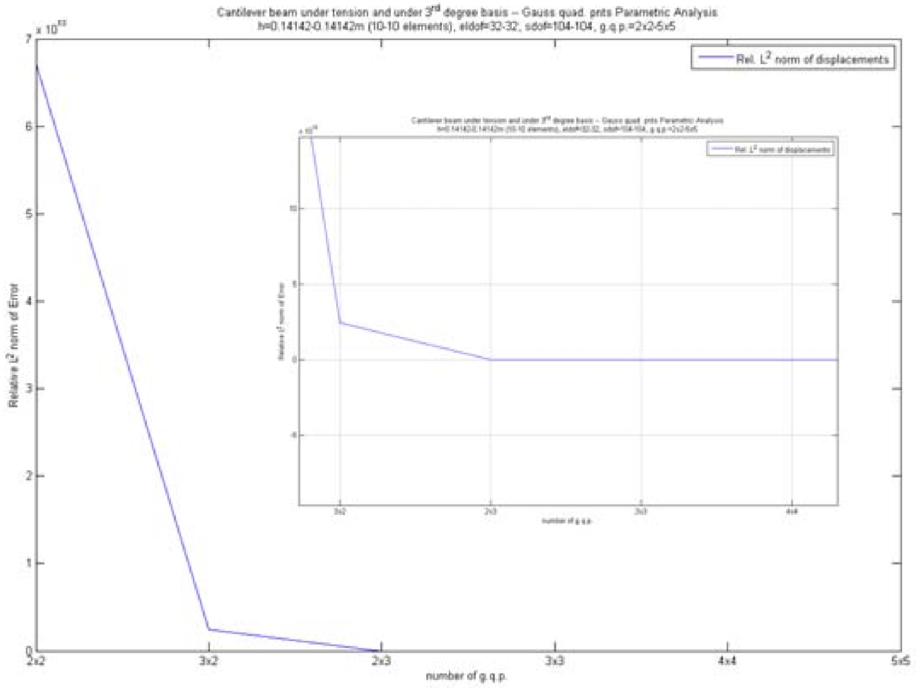


Figure 4.28

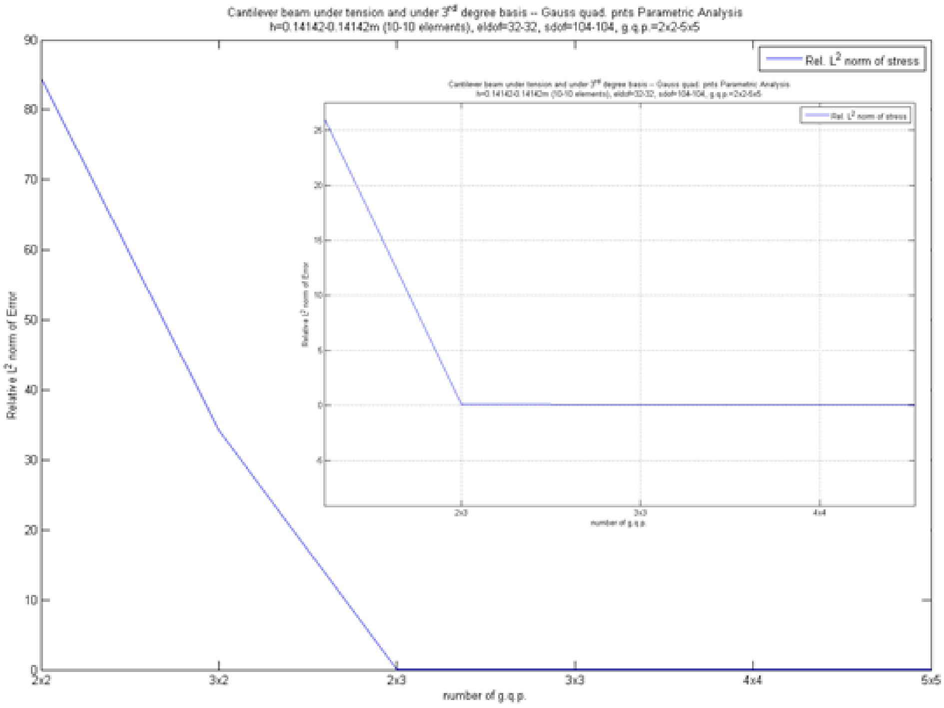


Figure 4.29

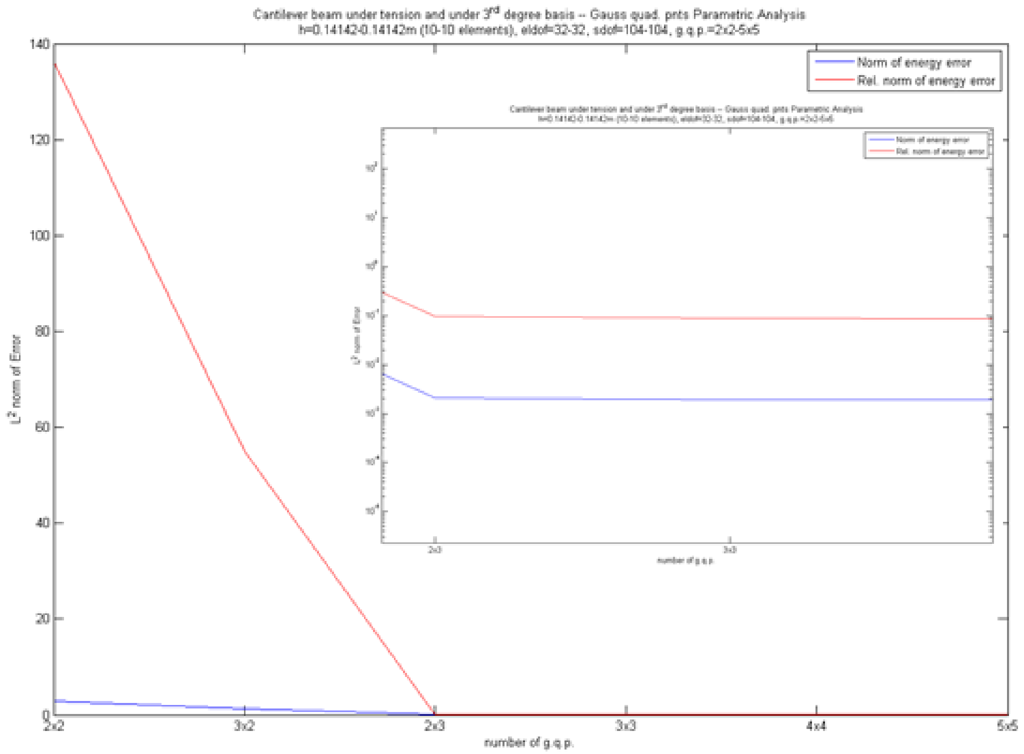
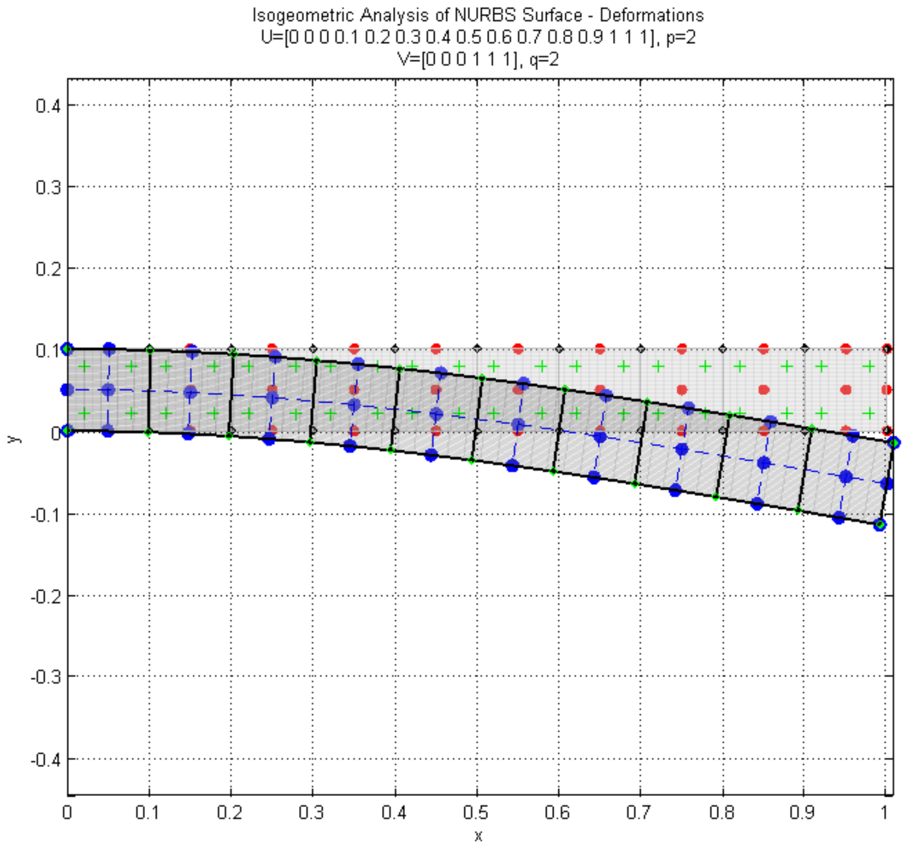


Figure 4.30

4.2.4 INTEGRATION OF 2ND DEGREE BASIS FUNCTIONS OF A TENSION PROBLEM



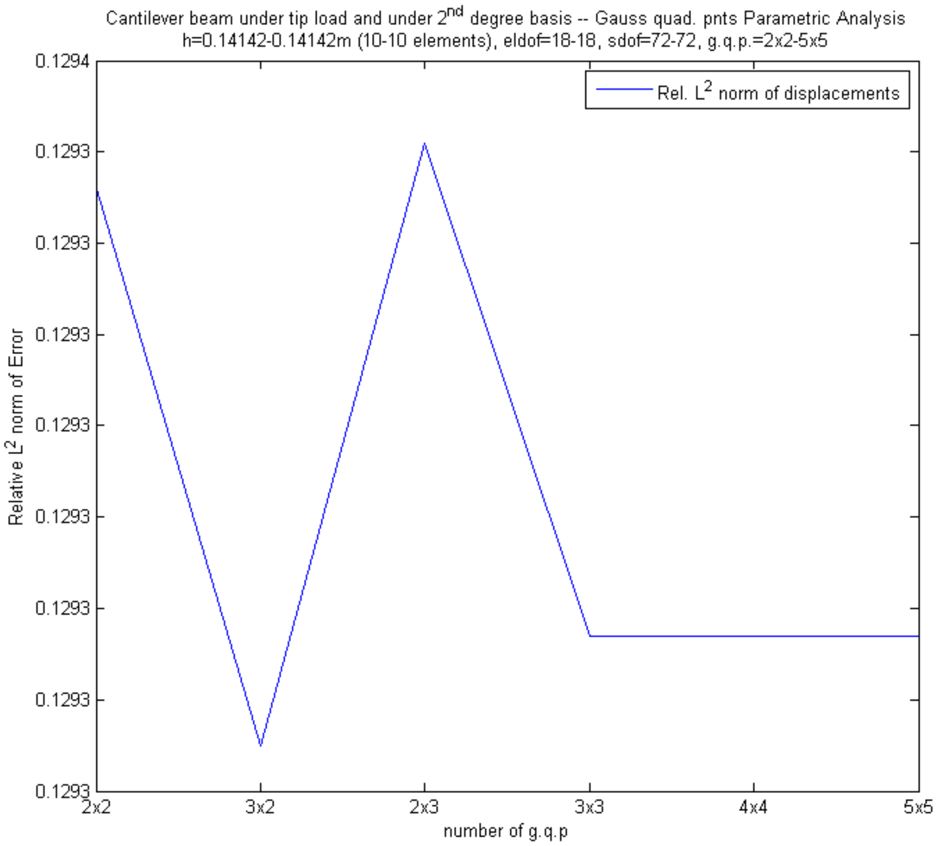


Figure 4.31

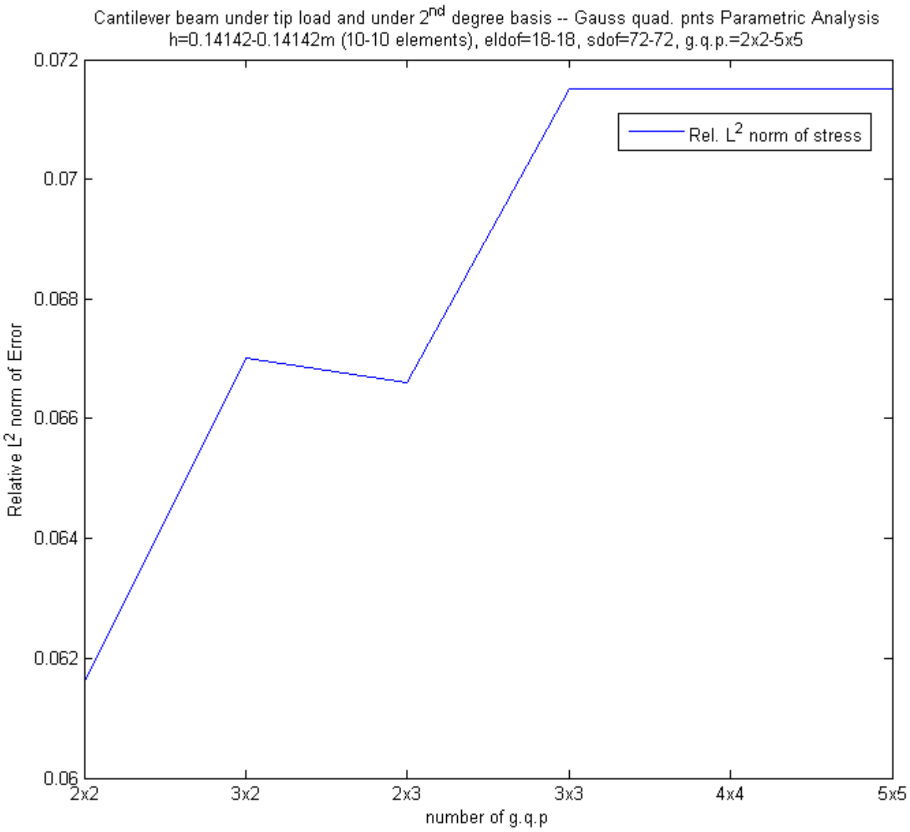


Figure 4.32

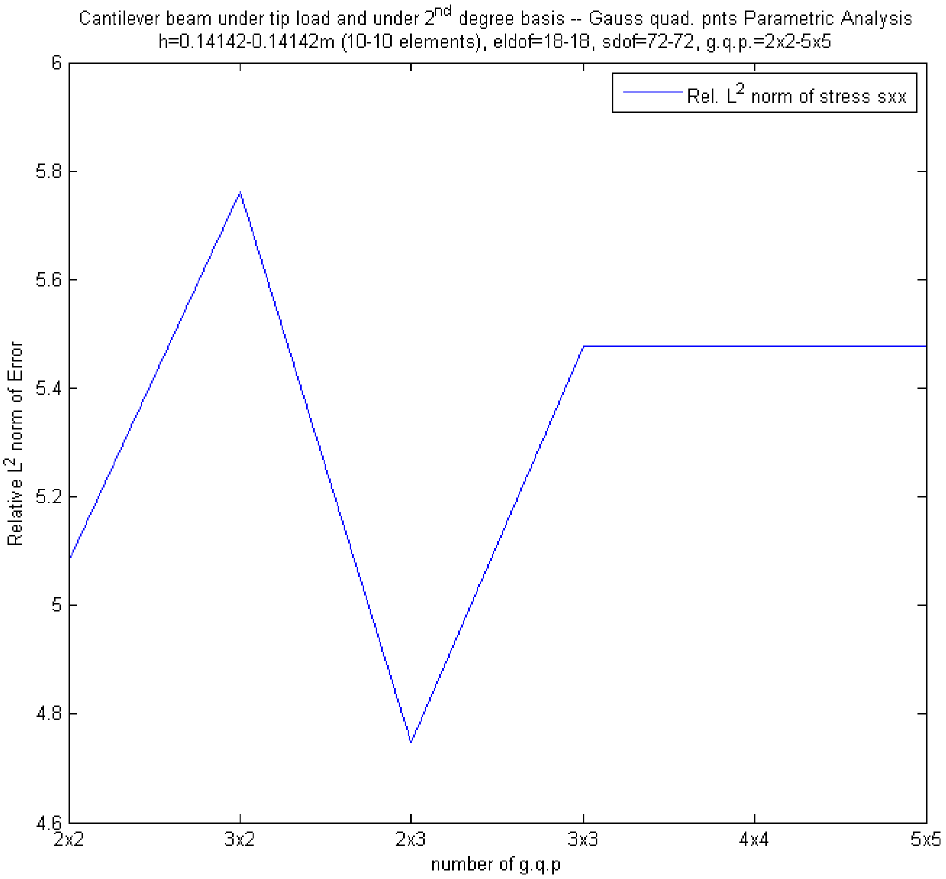


Figure 4.33

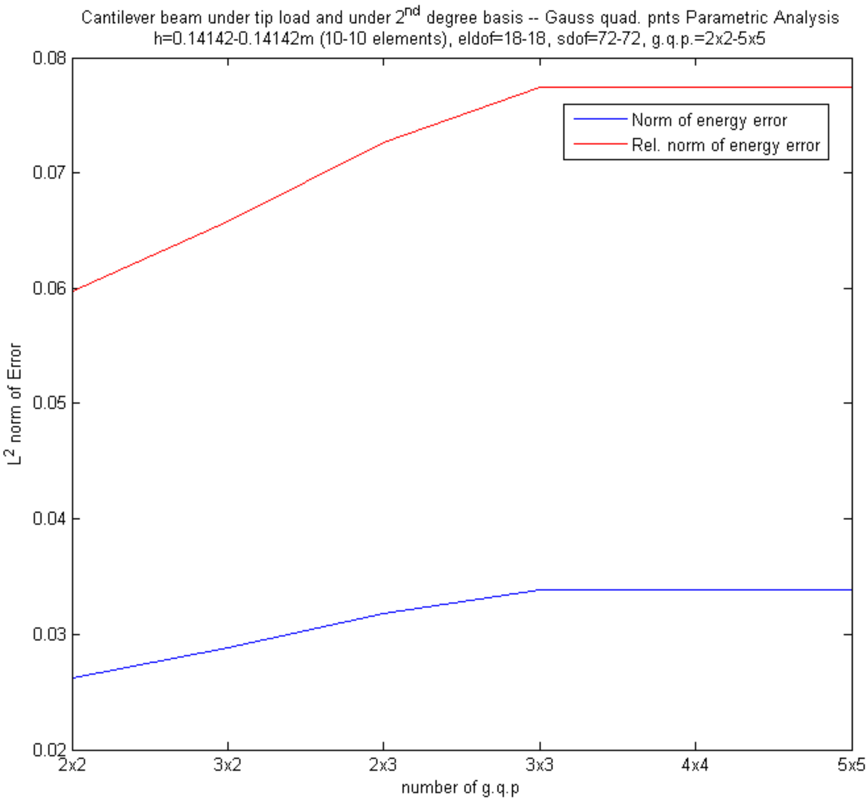


Figure 4.34

4.3 H-REFINEMENT & P-REFINEMENT ANALYSES

In this section parametric analysis are made on h and p refinement. These analyses made to get a better insight into the capabilities of the Isogeometric analysis method and to ensure that the codes are properly developed (CAGD and Analysis codes). The figure below illustrates the change of the total d.o.f of the system as we refine the analyzed model. The analyzed problem is the same that is presented previously in geometry and material properties. The discretization and the degree of the basis functions is refined.

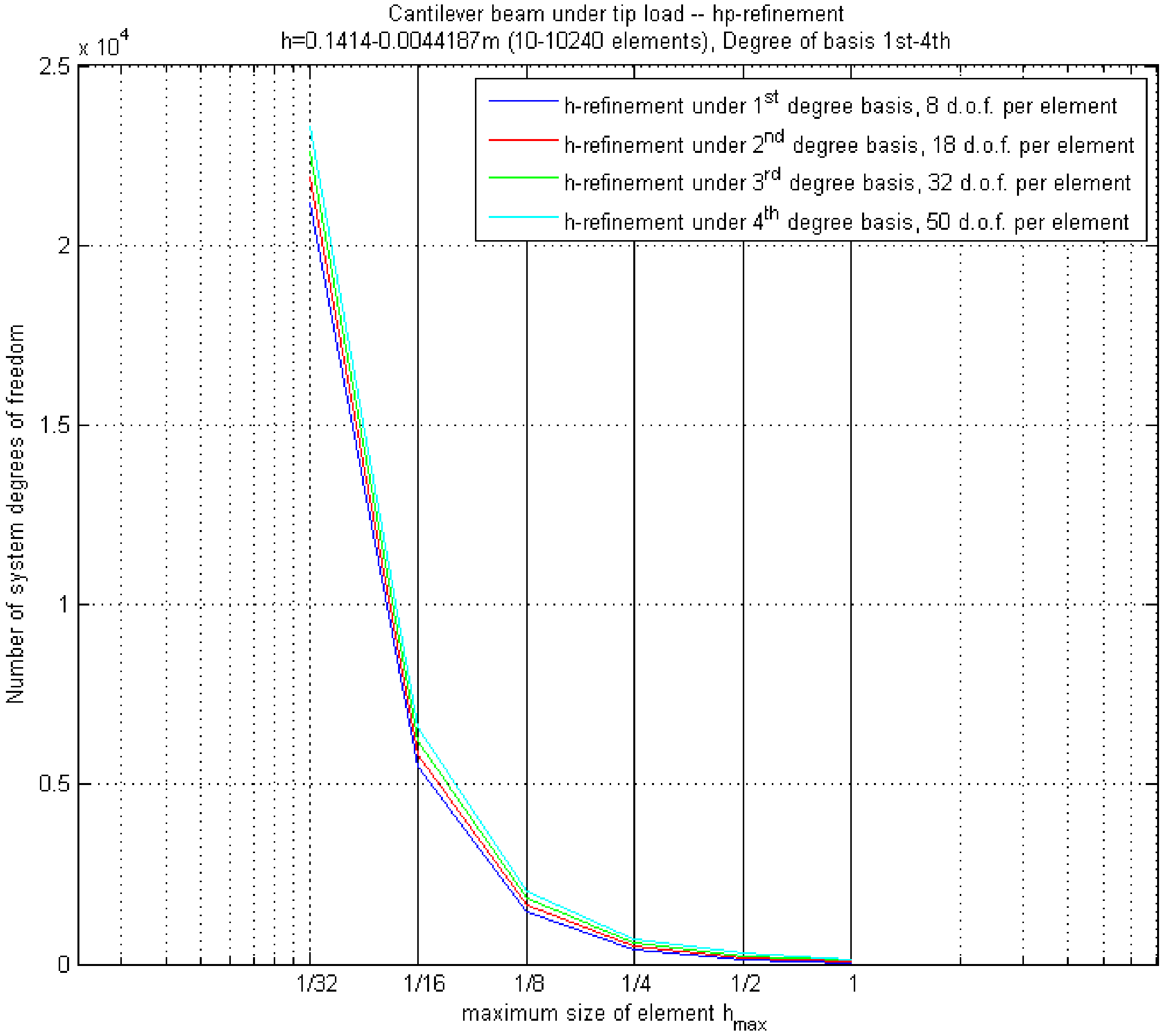


Figure 4.35 Change of the number of the system's d.o.f. as a function to h-refinement. Each curve is a specific degree of basis functions as is shown in legend

We observe that the degrees of freedom change strongly only by the h-refinement.

In Figure 4.36 it is shown an example of the produced total stiffness matrix of the same problem for a 4th degree basis functions and a discretization of 1x10 elements. (the d.o.f. are numbered in the most short direction)

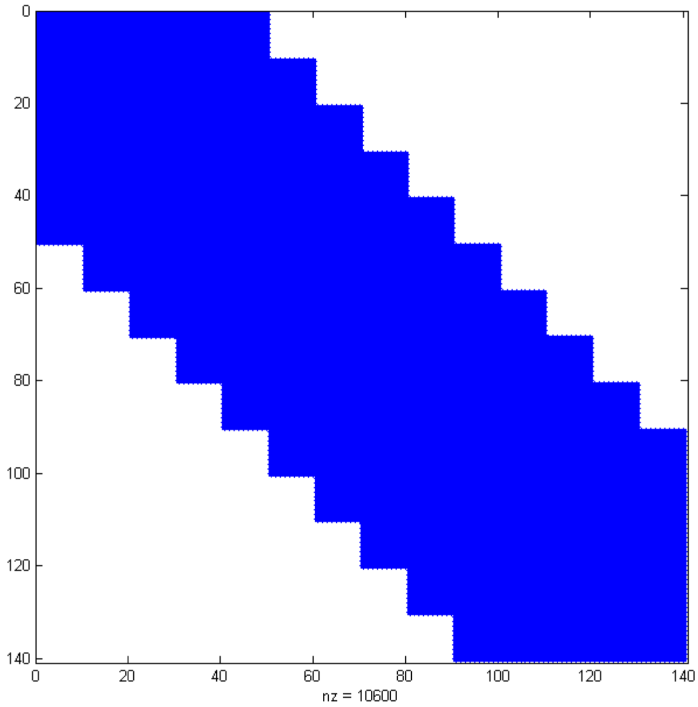


Figure 4.36 The illustration of a total stiffness matrix

The rundown of all the analyses done in hp-refiement is illustrated in Figure 4.37 through. As expected, the results of a higher degree basis functions are by far closely to the convergent solution. As higher is the degree of shape functions more closer is the solution to the convergent by a coarse mesh.

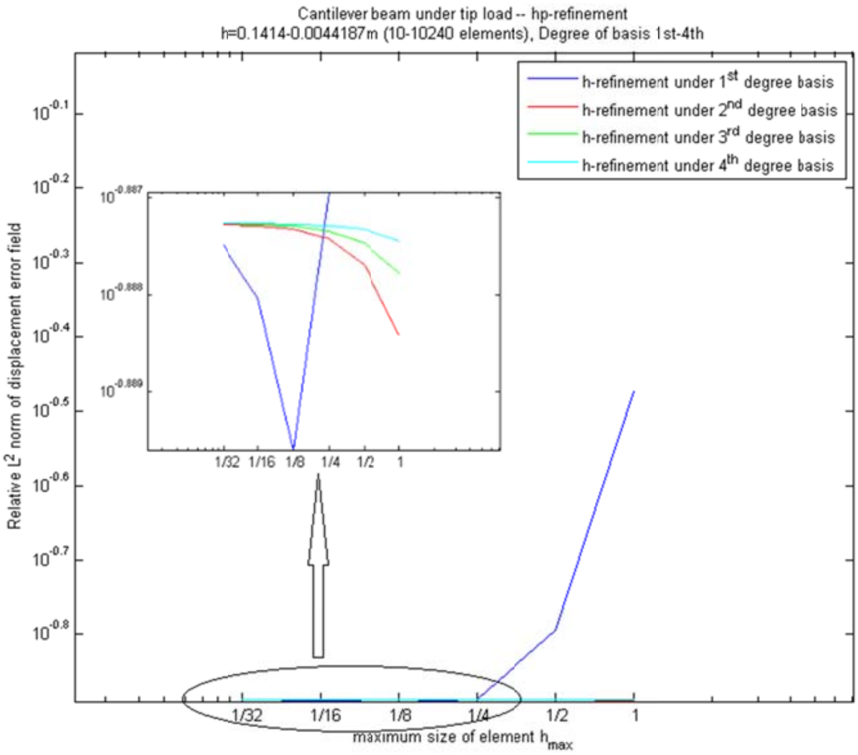


Figure 4.37 Relative norm of the displacement error field under the hp-refinement analyses

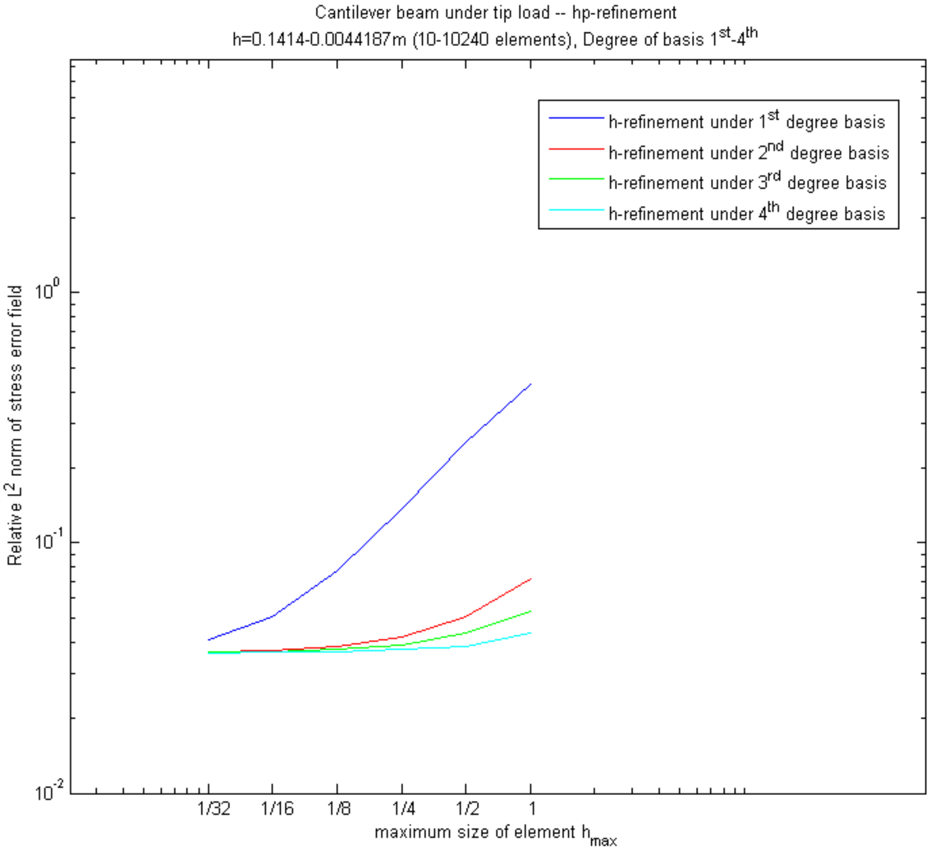


Figure 4.38 Relative norm of the stress error field under the hp-refinement analyses

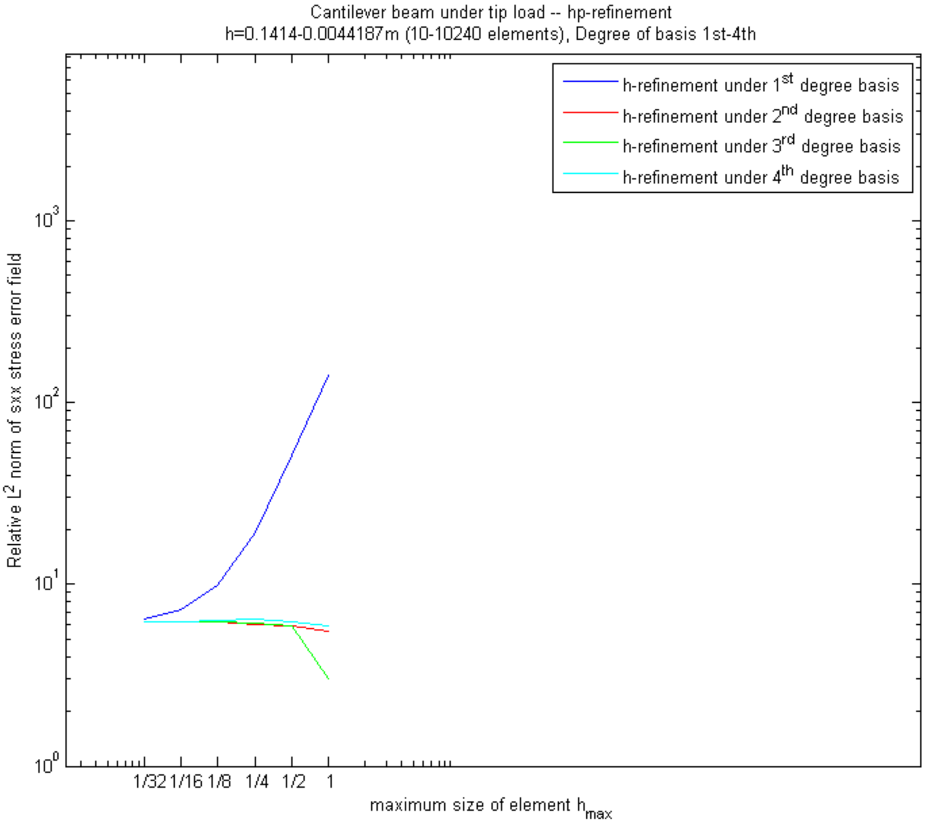


Figure 4.39 Relative norm of the normal stress error field under the hp-refinement analyses

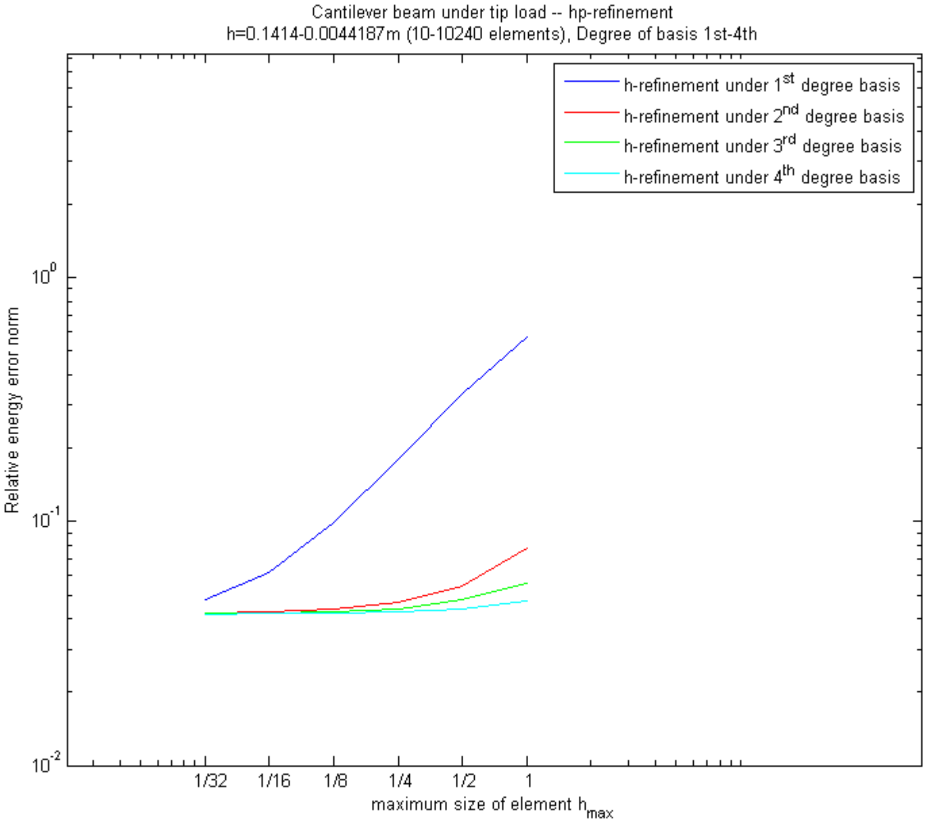


Figure 4.40 Relative energy norms under the hp-refinement analyses

The details of these analyses are shown in subsections.

4.3.1 H-REFINEMENT OF THE PROBLEM UNDER 1ST DEGREE BIVARIATE BASIS

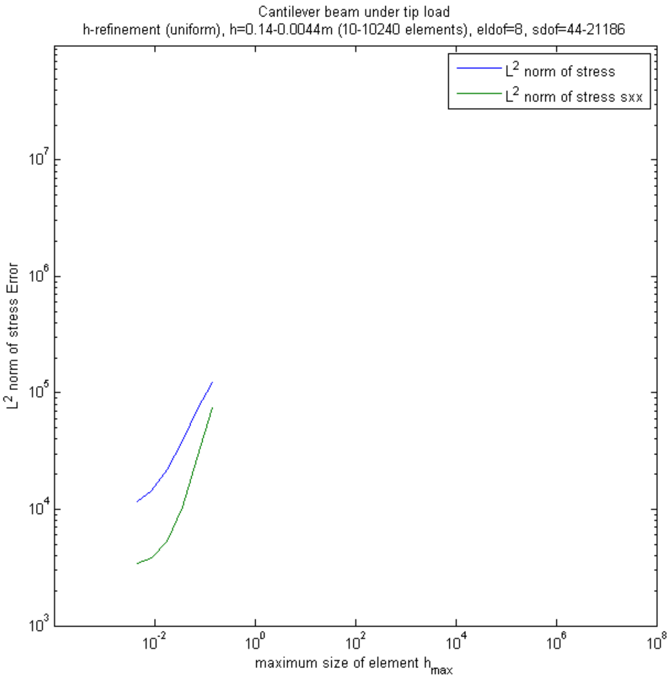


Figure 4.41

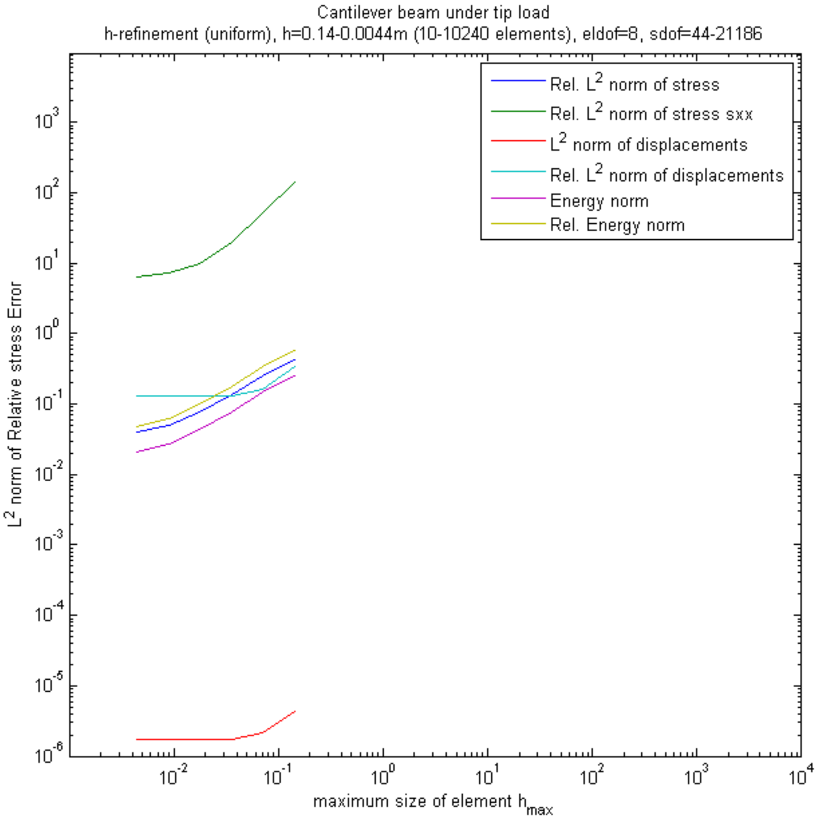


Figure 4.42

4.3.2 H-REFINEMENT OF THE PROBLEM UNDER 2ND DEGREE BIVARIATE BASIS

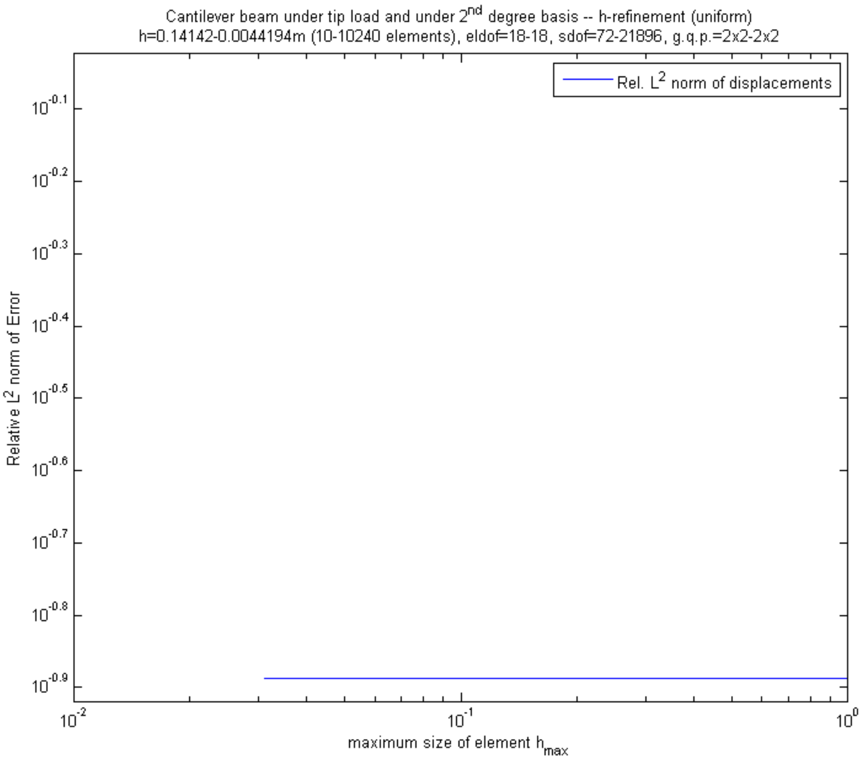


Figure 4.43

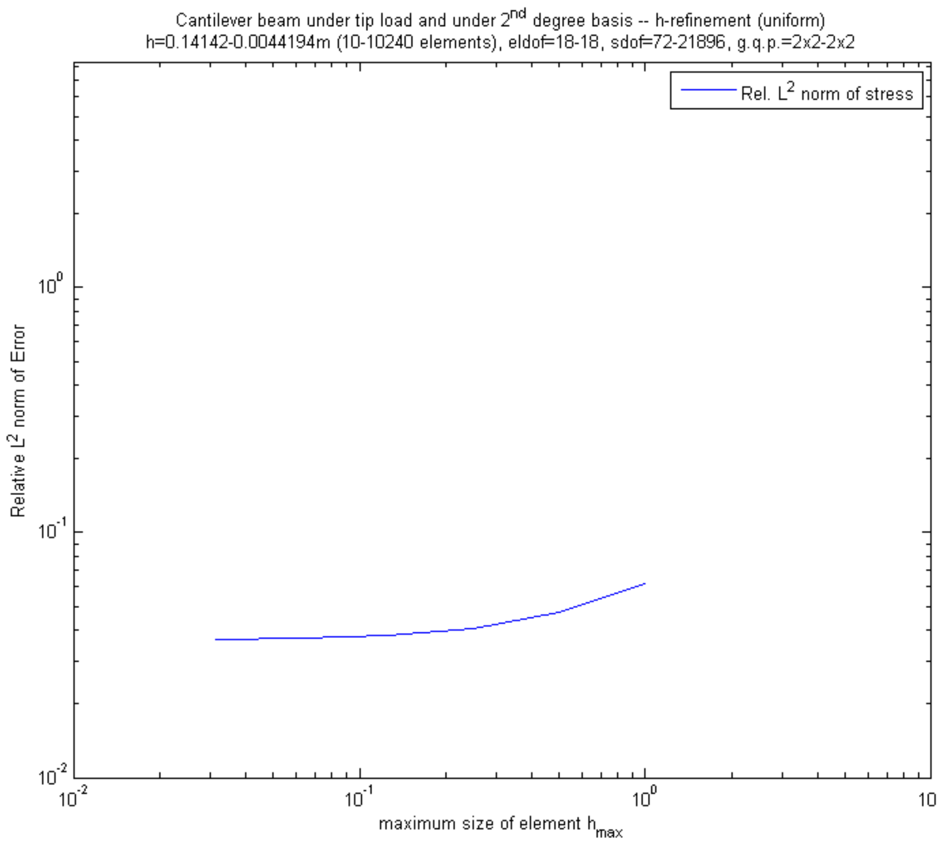


Figure 4.44

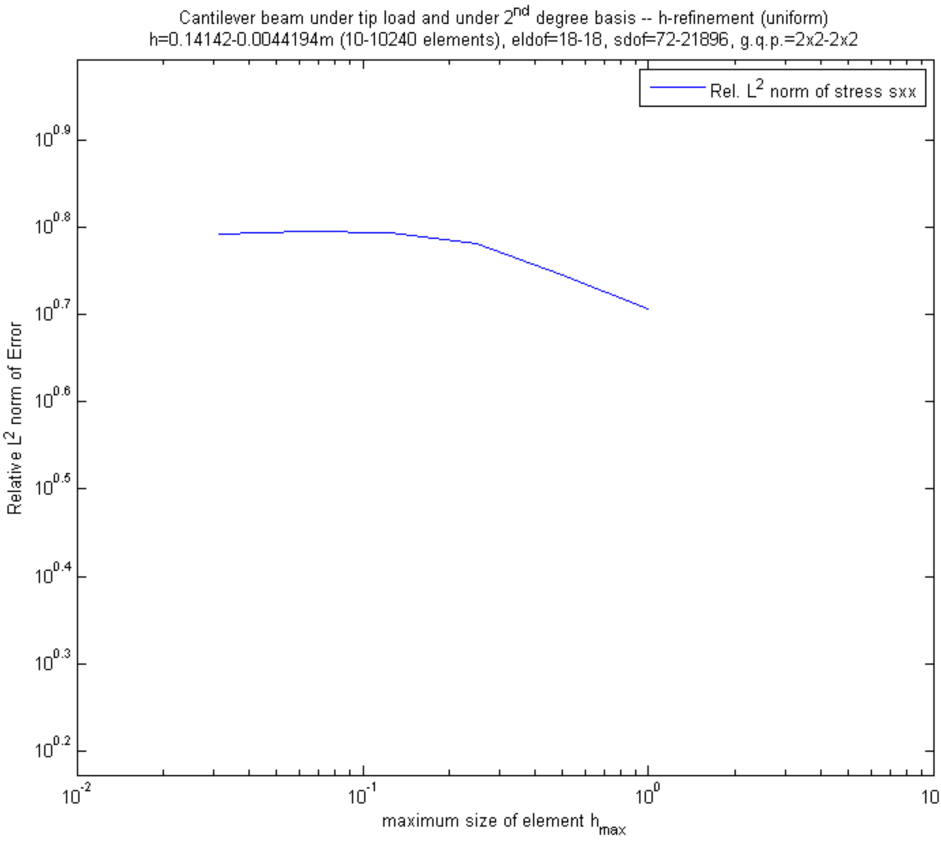


Figure 4.45

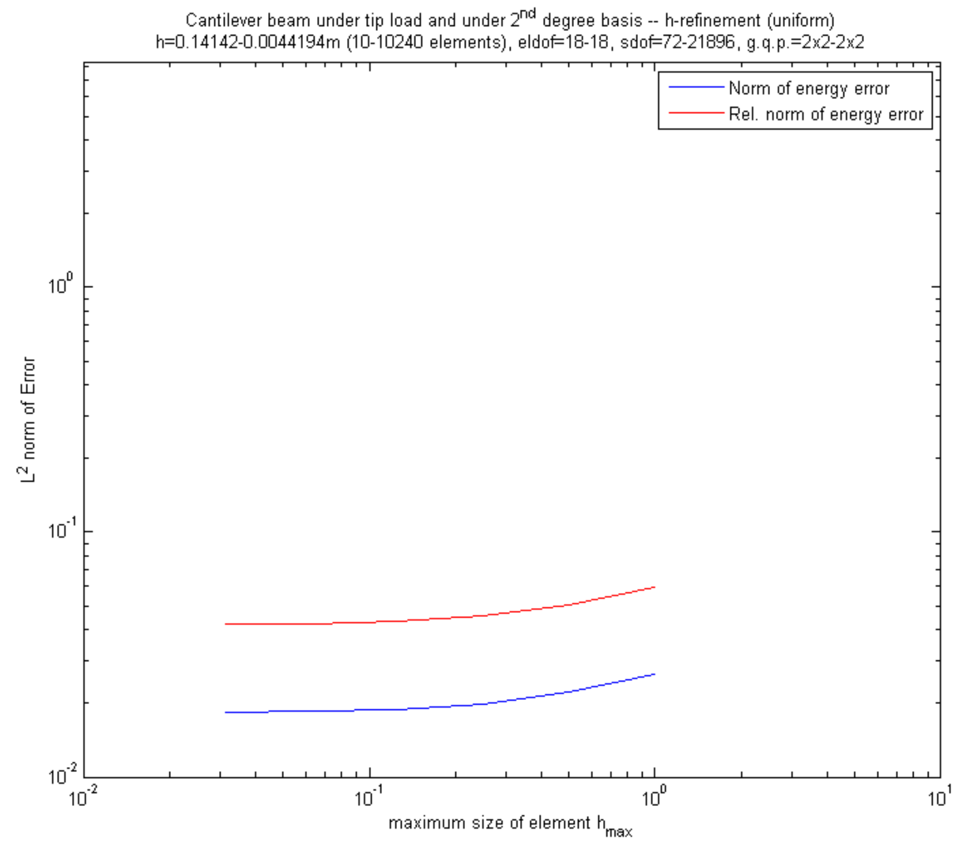


Figure 4.46

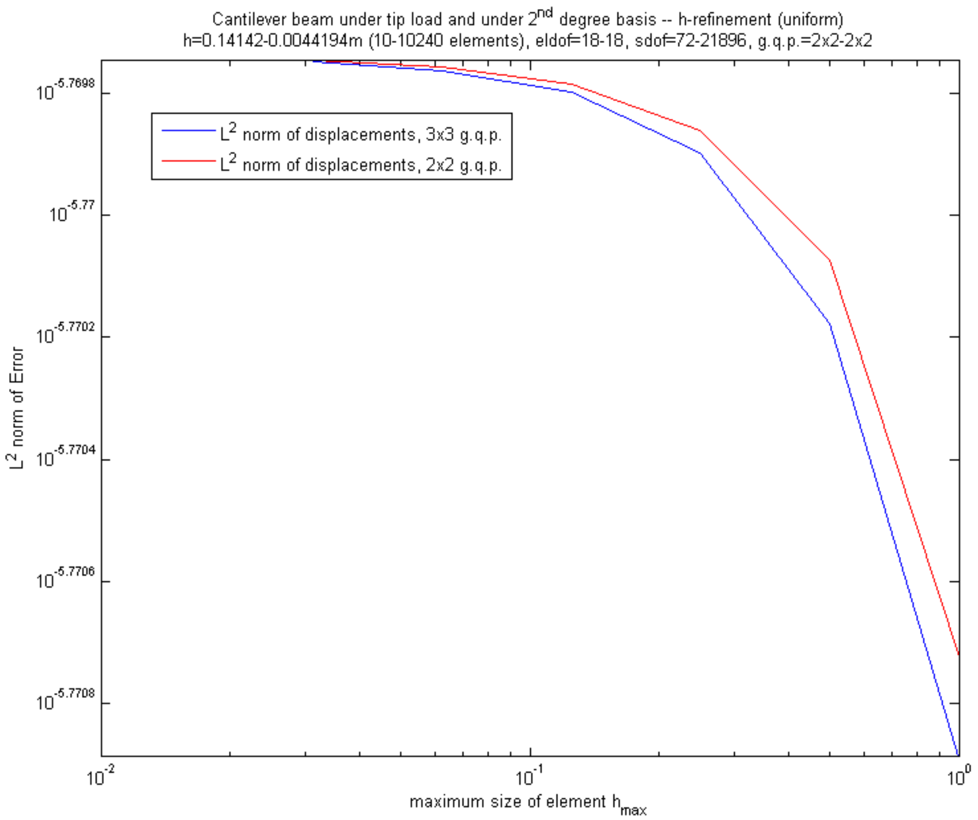


Figure 4.47

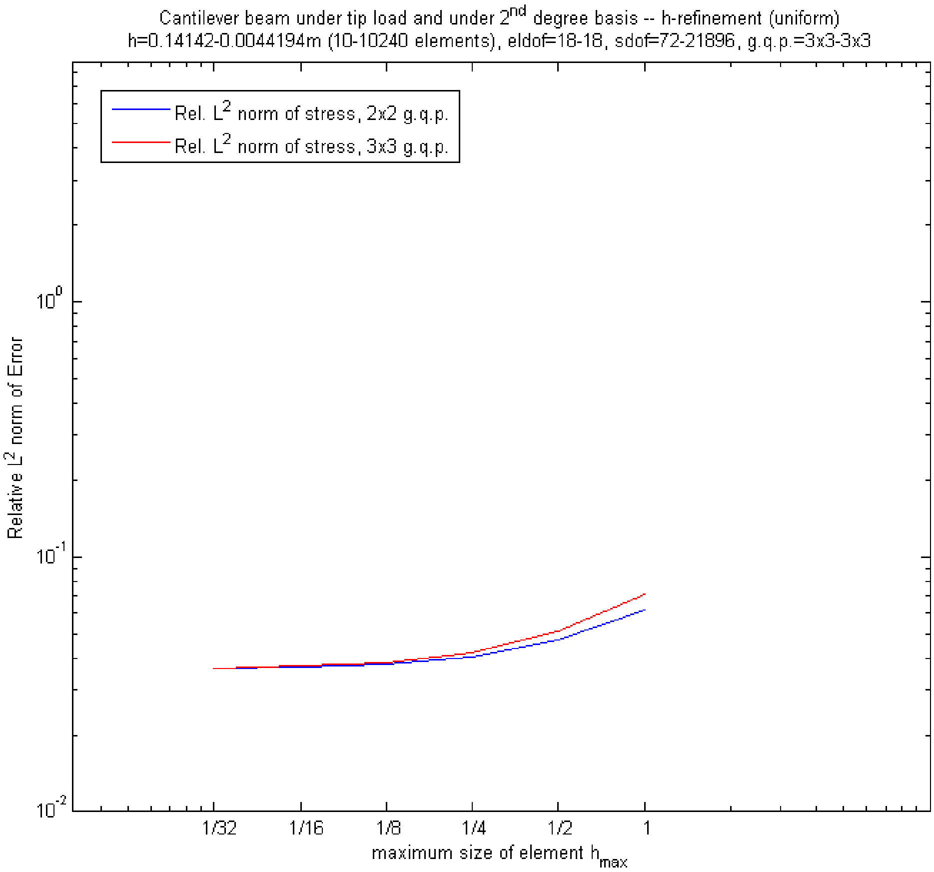


Figure 4.48

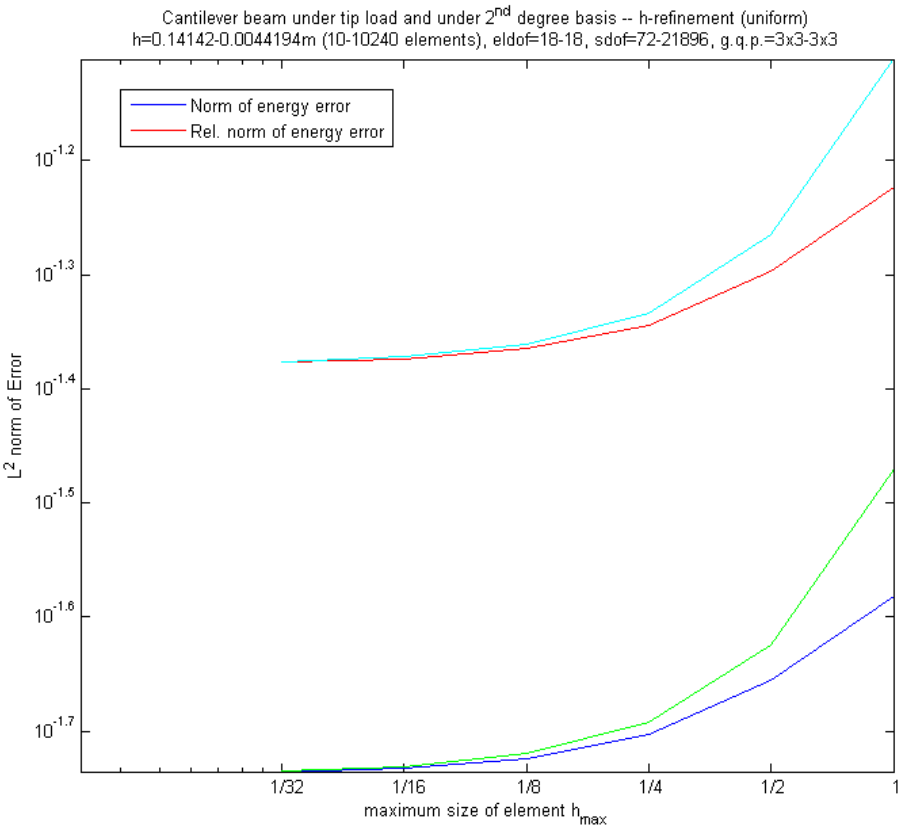


Figure 4.49

4.3.3 H-REFINEMENT OF THE PROBLEM UNDER 3RD DEGREE BIVARIATE BASIS

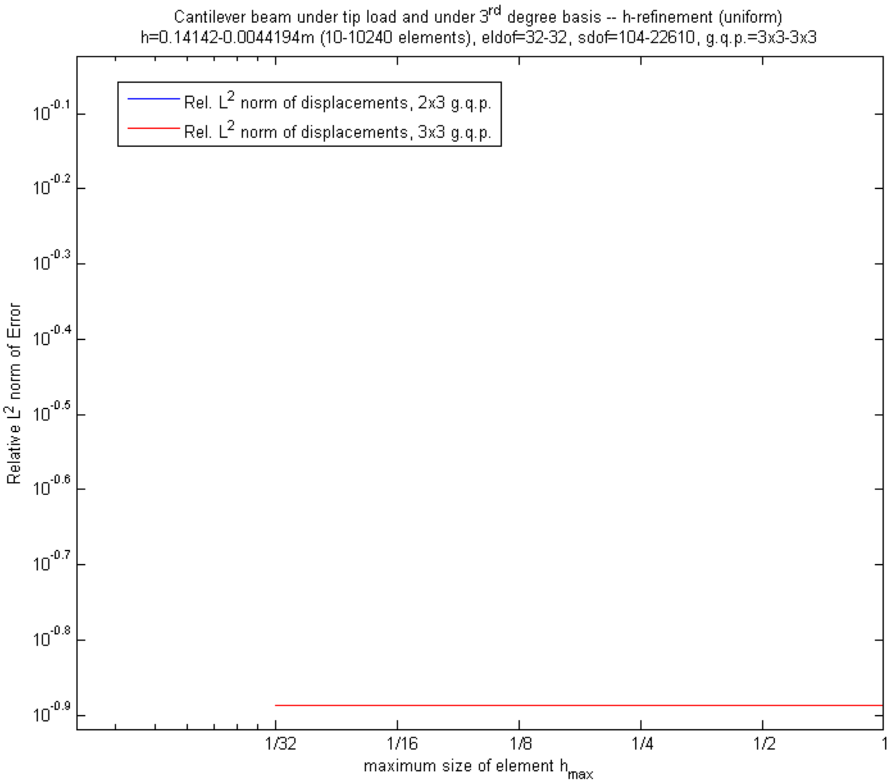


Figure 4.50

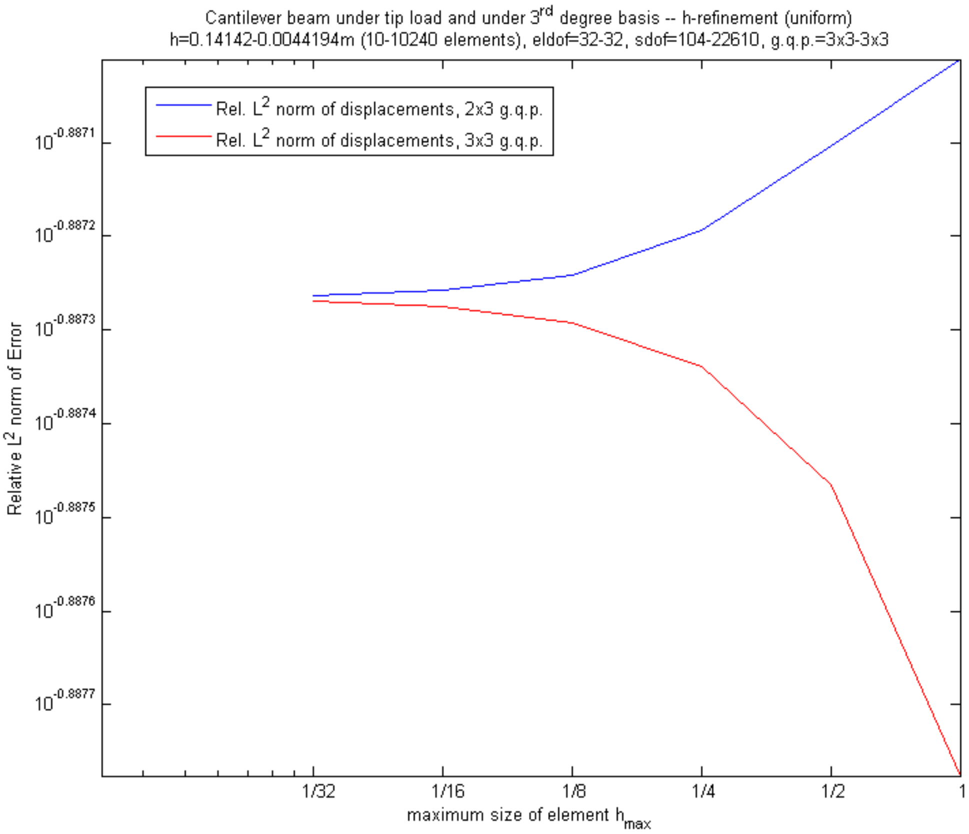


Figure 4.51

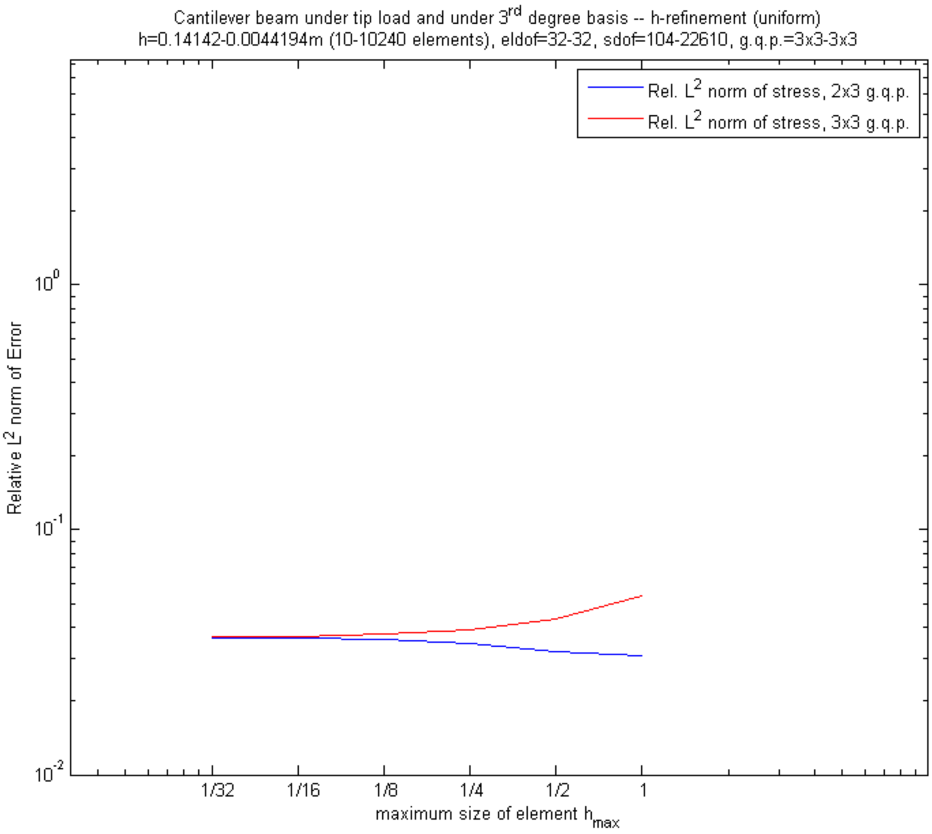


Figure 4.52

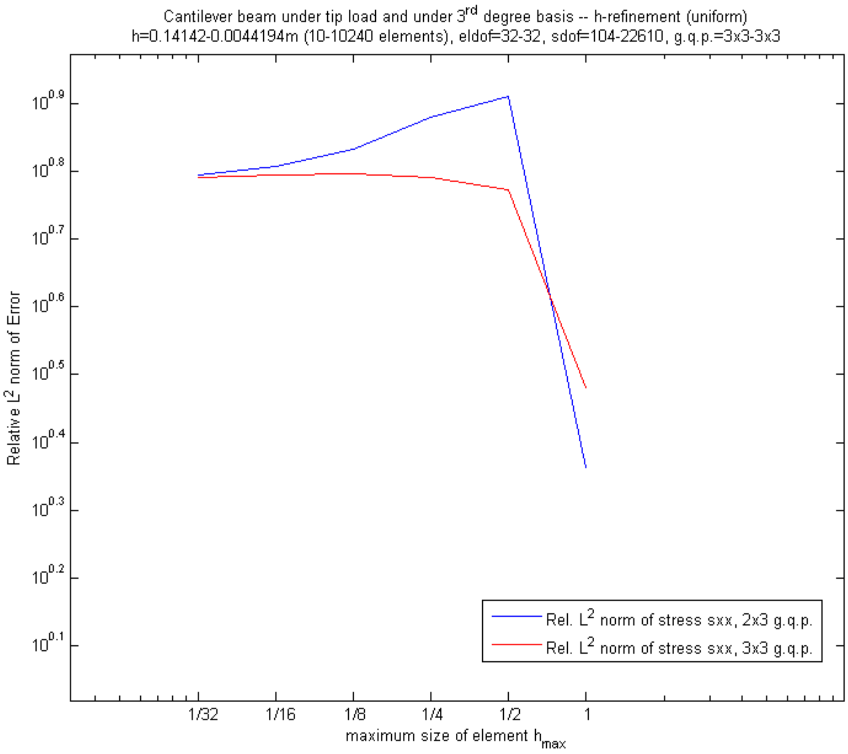


Figure 4.53

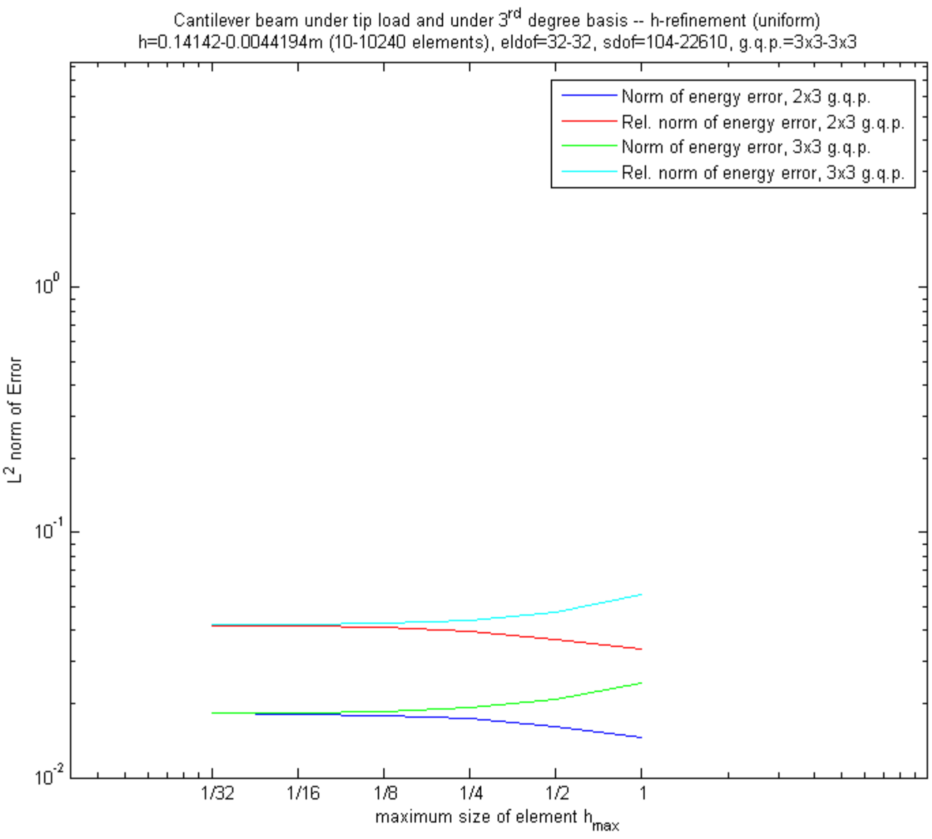
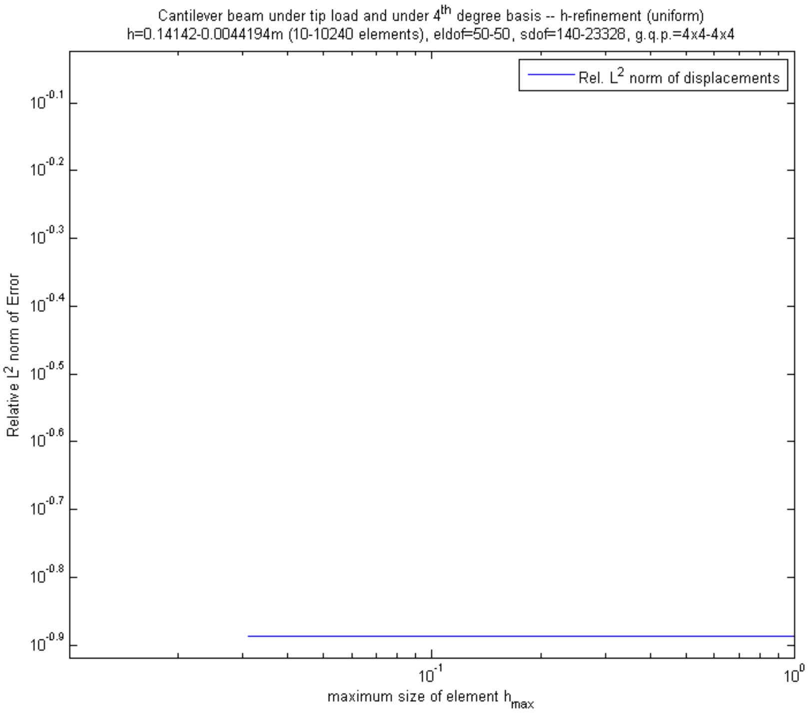
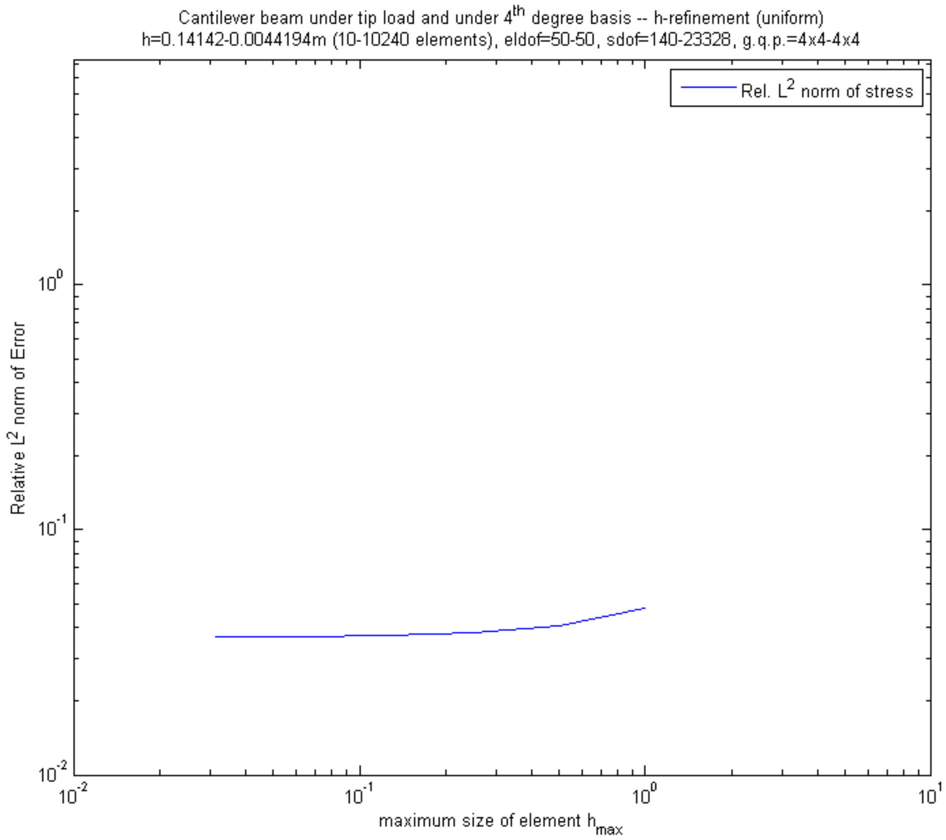
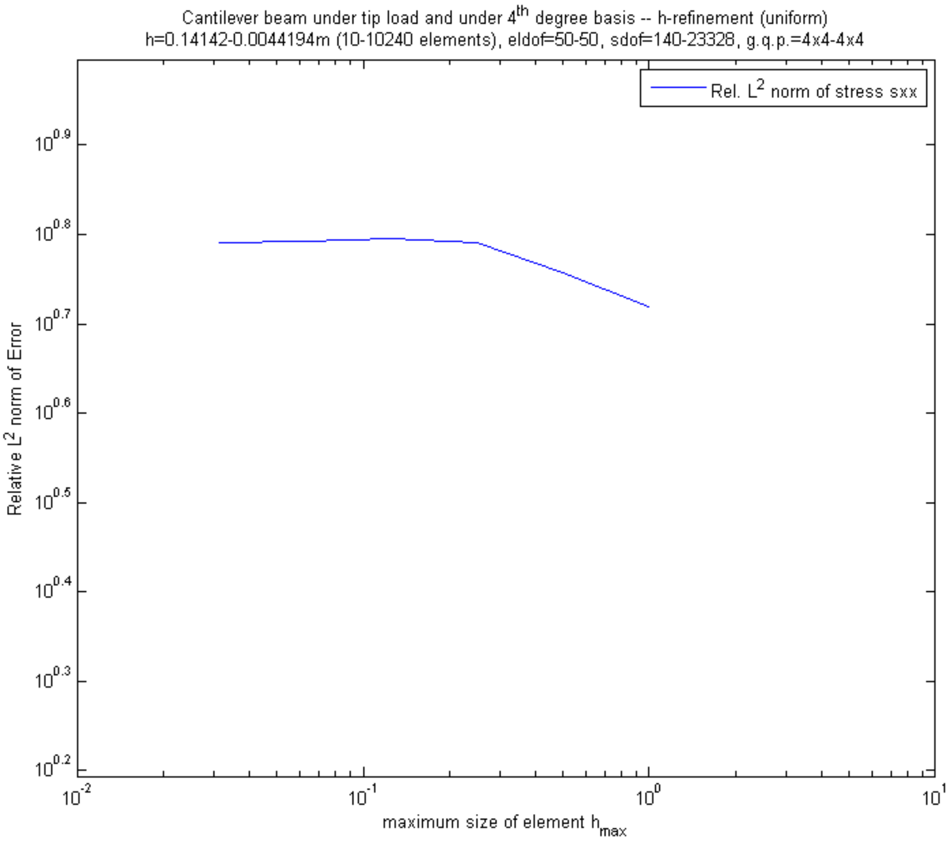


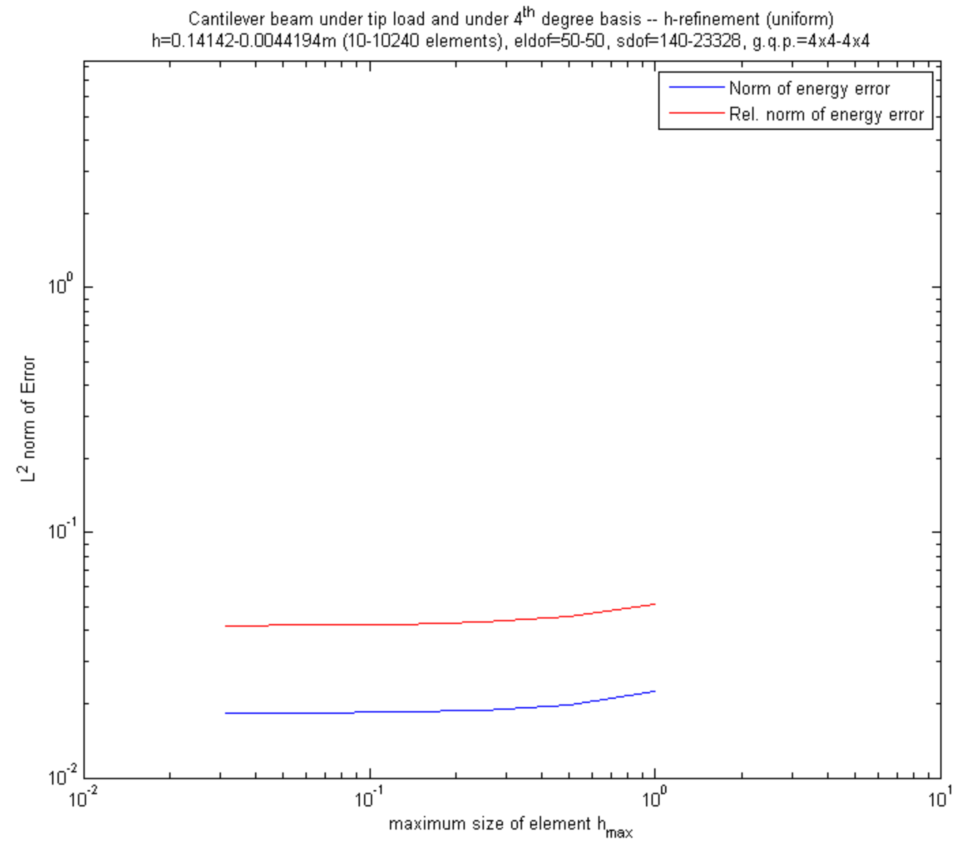
Figure 4.54

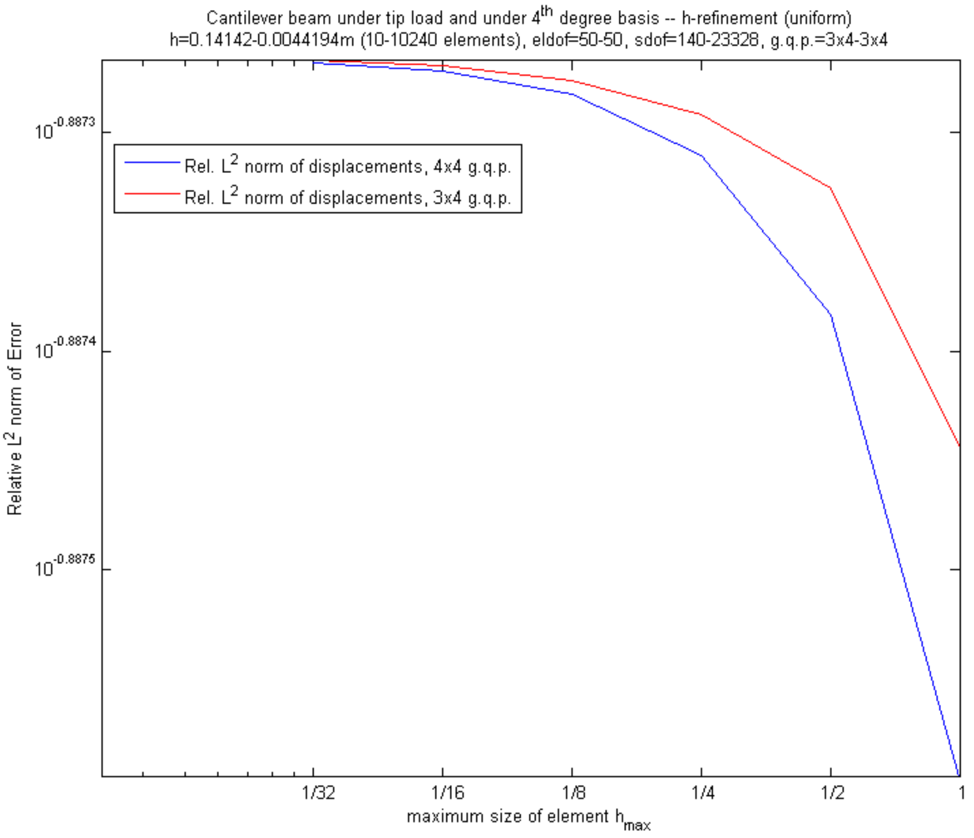
4.3.4 H-REFINEMENT OF THE PROBLEM UNDER 4TH DEGREE BIVARIATE BASIS

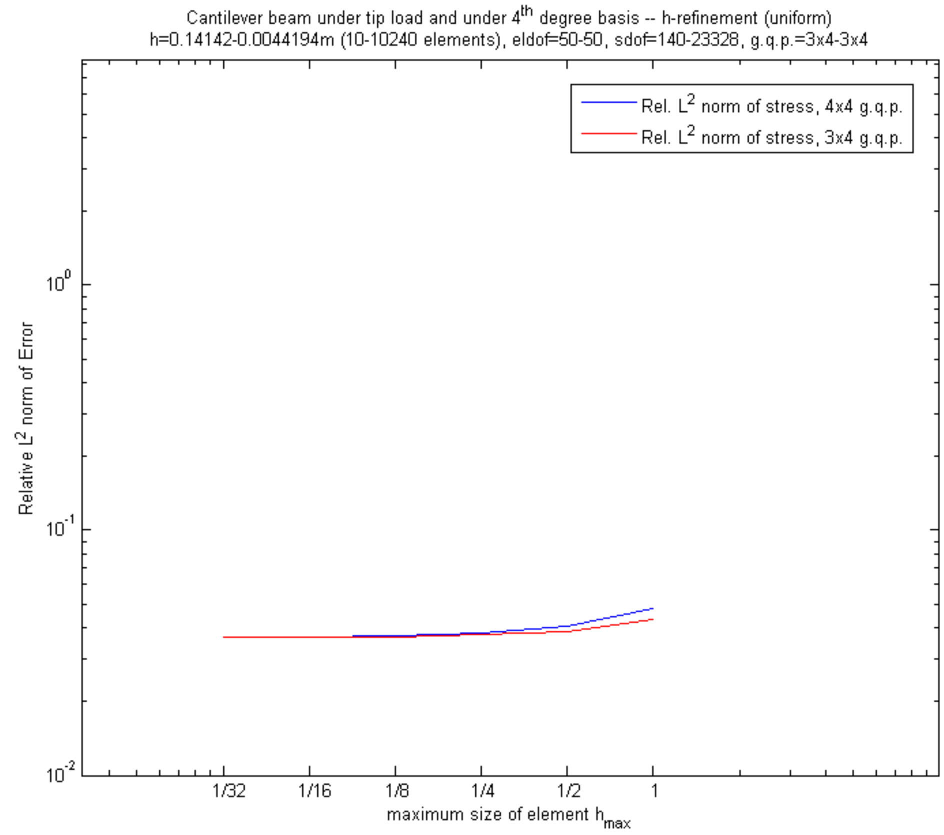


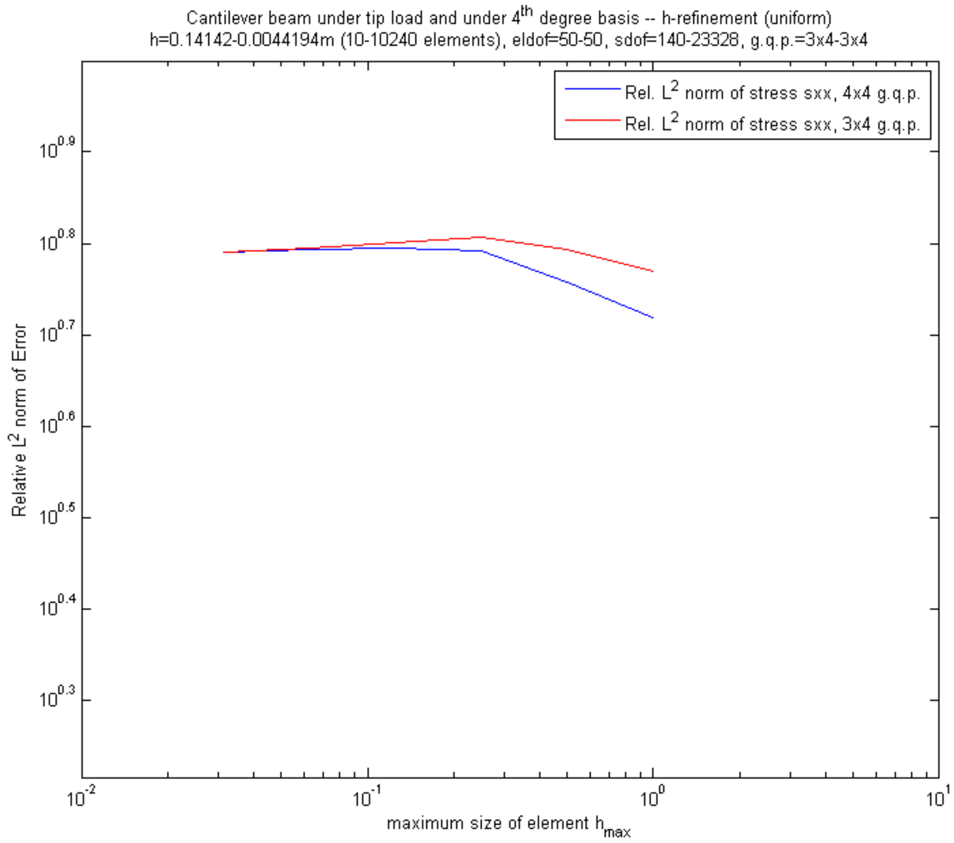

Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=4x4-4x4
Rel. L^2 norm of stress
Relative L^2 norm of Error
10^{0}
10^{-1}
10^{-2}
10^{-2}
10^{-1}
10^{0}
10^{1}
maximum size of element h_{max}

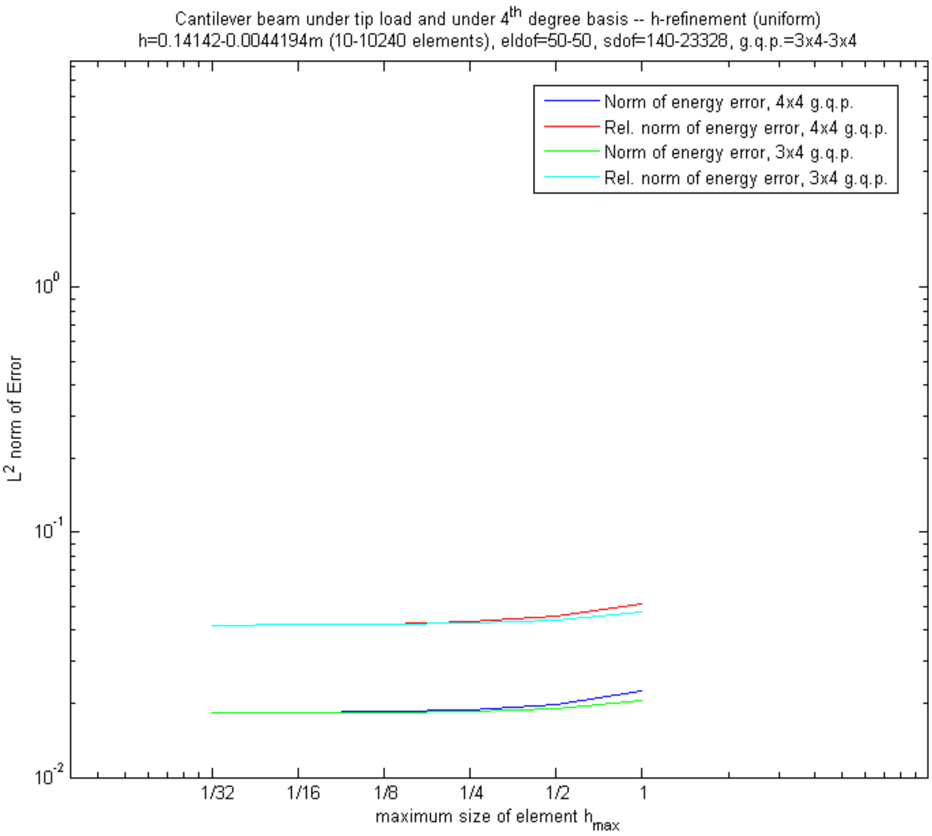

Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=4x4-4x4
Rel. L^2 norm of stress sxx
Relative L^2 norm of Error
$10^{0.9}$
$10^{0.8}$
$10^{0.7}$
$10^{0.6}$
$10^{0.5}$
$10^{0.4}$
$10^{0.3}$
$10^{0.2}$
10^{-2}
10^{-1}
10^{0}
10^{1}
maximum size of element h_{max}


Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=4x4-4x4
Norm of energy error
Rel. norm of energy error
L2 norm of Error
10^0
10^-1
10^-2
10^-2
10^-1
10^0
10^1
maximum size of element h_max


Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=3x4-3x4
Rel. L2 norm of displacements, 4x4 g.q.p.
Rel. L2 norm of displacements, 3x4 g.q.p.
Relative L2 norm of Error
10^-0.8873
10^-0.8874
10^-0.8875
1/32
1/16
1/8
1/4
1/2
1
maximum size of element h_max


Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=3x4-3x4
Rel. L2 norm of stress, 4x4 g.q.p.
Rel. L2 norm of stress, 3x4 g.q.p.
Relative L2 norm of Error
10^0
10^-1
10^-2
1/32
1/16
1/8
1/4
1/2
1
maximum size of element h_max


Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=3x4-3x4
Rel. L2 norm of stress sxx, 4x4 g.q.p.
Rel. L2 norm of stress sxx, 3x4 g.q.p.
Relative L2 norm of Error
10^0.9
10^0.8
10^0.7
10^0.6
10^0.5
10^0.4
10^0.3
10^-2
10^-1
10^0
10^1
maximum size of element h_max


Cantilever beam under tip load and under 4th degree basis -- h-refinement (uniform)
h=0.14142-0.0044194m (10-10240 elements), eldof=50-50, sdof=140-23328, g.q.p.=3x4-3x4
Norm of energy error, 4x4 g.q.p.
Rel. norm of energy error, 4x4 g.q.p.
Norm of energy error, 3x4 g.q.p.
Rel. norm of energy error, 3x4 g.q.p.
L2 norm of Error
10^0
10^-1
10^-2
1/32
1/16
1/8
1/4
1/2
1
maximum size of element h_max

5 INFINITE PLATE WITH CIRCULAR HOLE UNDER CONSTANT IN-PLANE TENSION

NURBS are particularly well suited to linear elasticity. It is obvious that representing geometry accurately at all levels of discretization should lead to improved accuracy across all meshes as compared with less geometrically accurate methods. Furthermore, as we have seen, the standard formulation for linear elasticity uses the displacements as the unknown degrees-of-freedom. In practice, however, it is often the case that the quantity of interest is not the displacement but the stress. The stress is a function of the gradient of the displacement, and so any approach using elements that are only C^0 across element boundaries results in stress values being undefined at these element boundaries see Figure 2.9. Alternatively, a C^1 NURBS basis results in unambiguous, continuous stresses across element boundaries.

In this example, an infinite plate with a circular hole under constant in-plane tension at infinity is undergone isogeometric analysis based on NURBS. The infinite plate is modelled by a finite quarter plate, due to the symmetry as far as the geometry and tension are concerned.

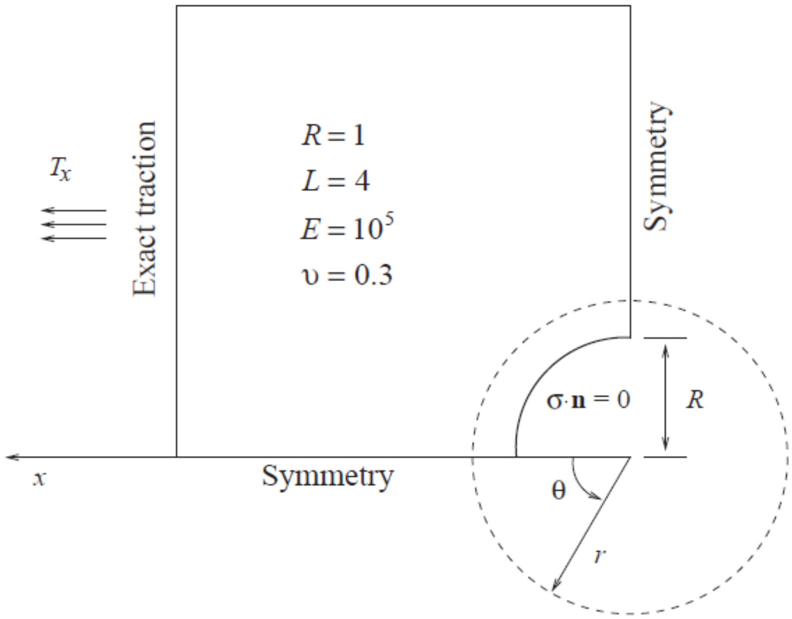


Figure 5.1 Definition of the problem

The problem is analysed by two elements and by 4th degree basis functions in both directions.

We observe that a discontinuity exist at the left corner which is due to the technic used to construct the corner from a smooth curve by overlapping p control points, where p the degree of the basis in that direction. The continuity at the knot line between the two elements is C^1 as it is p-1-m where p the degree and m the knot multiplicity. For better results the technic of construction of the corner must be optimized. Also a k-refinement on the elements border may help.

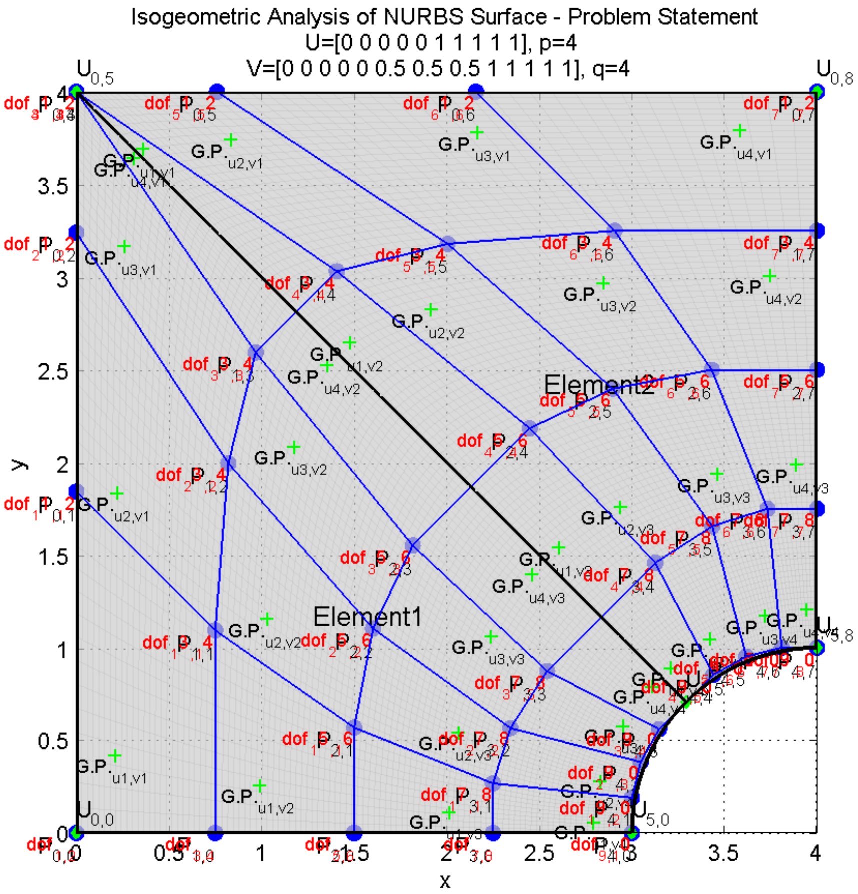


Figure 5.2

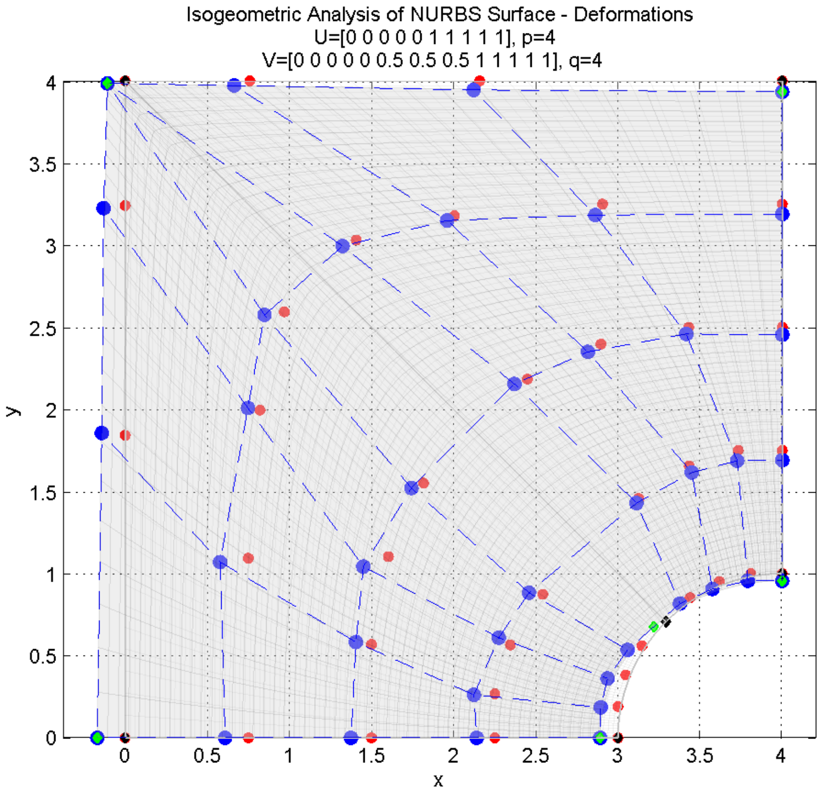


Figure 5.3

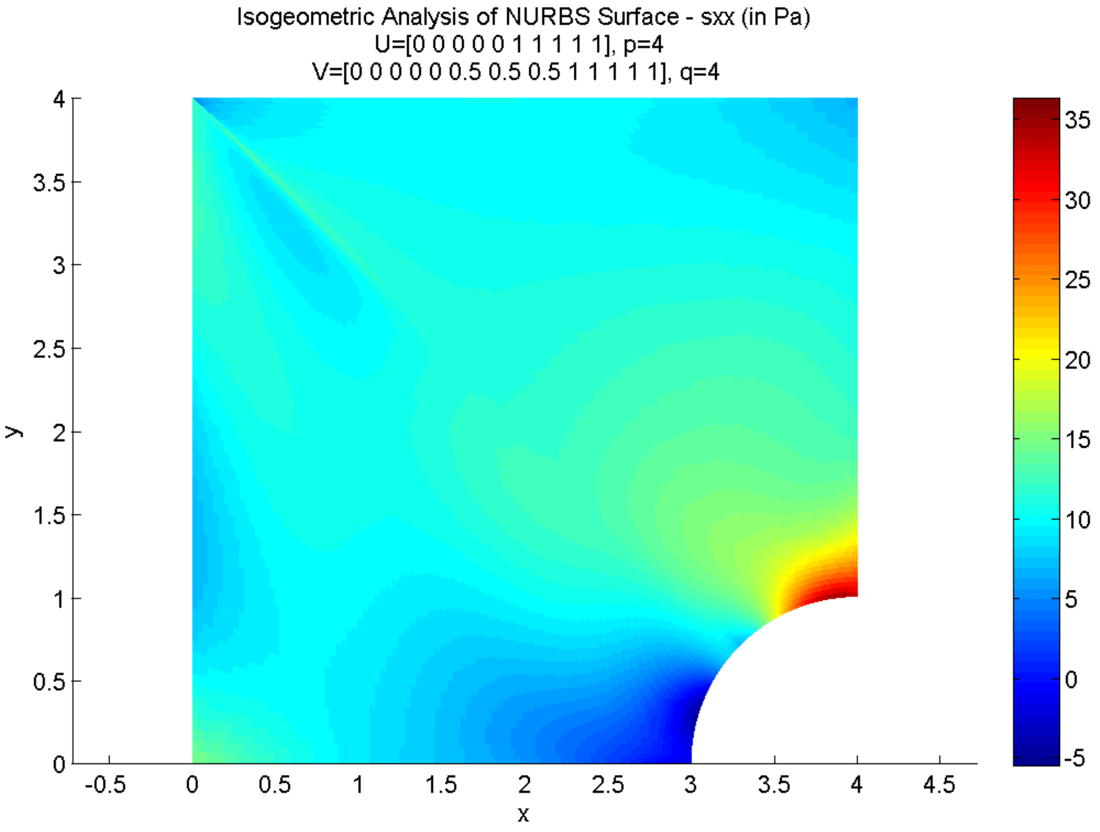


Figure 5.4

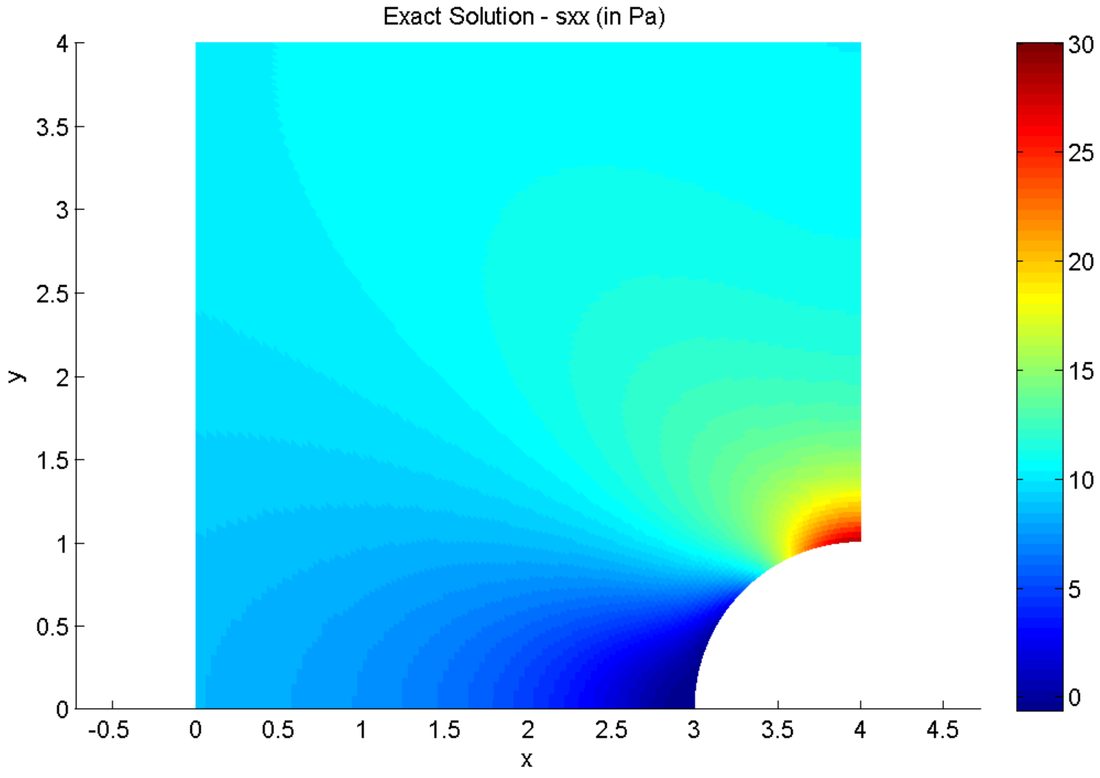


Figure 5.5

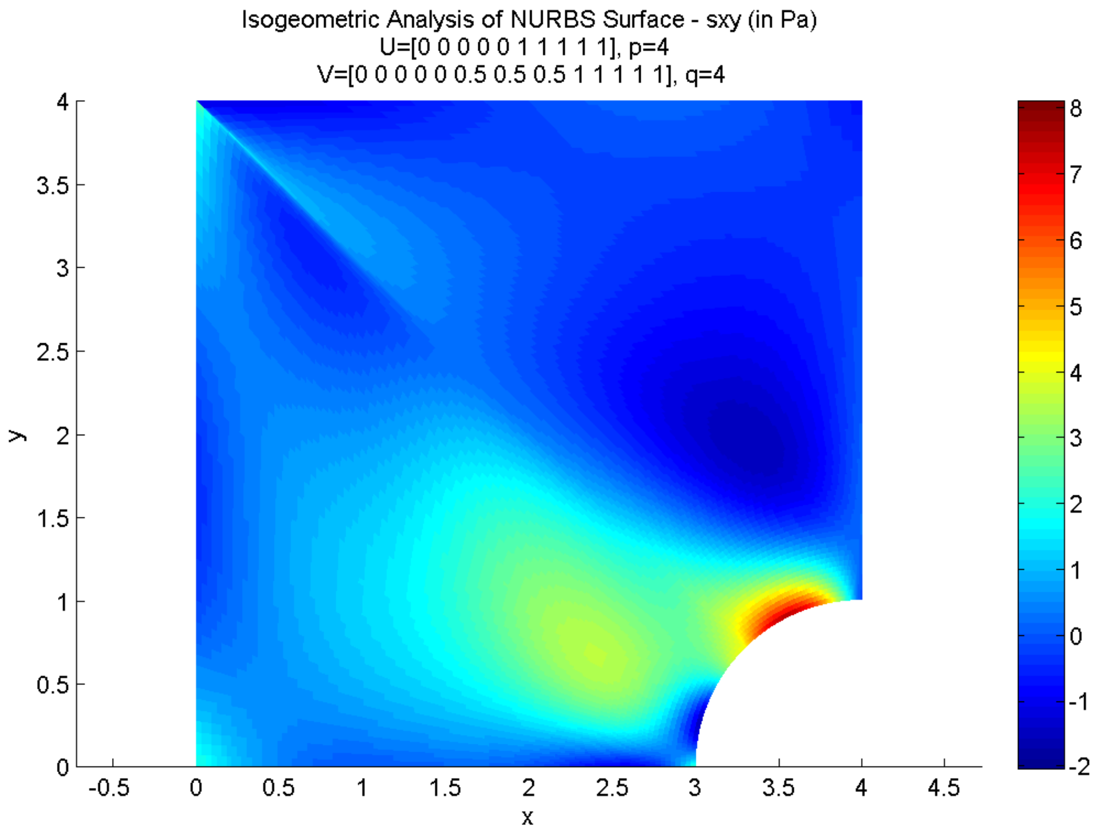


Figure 5.6

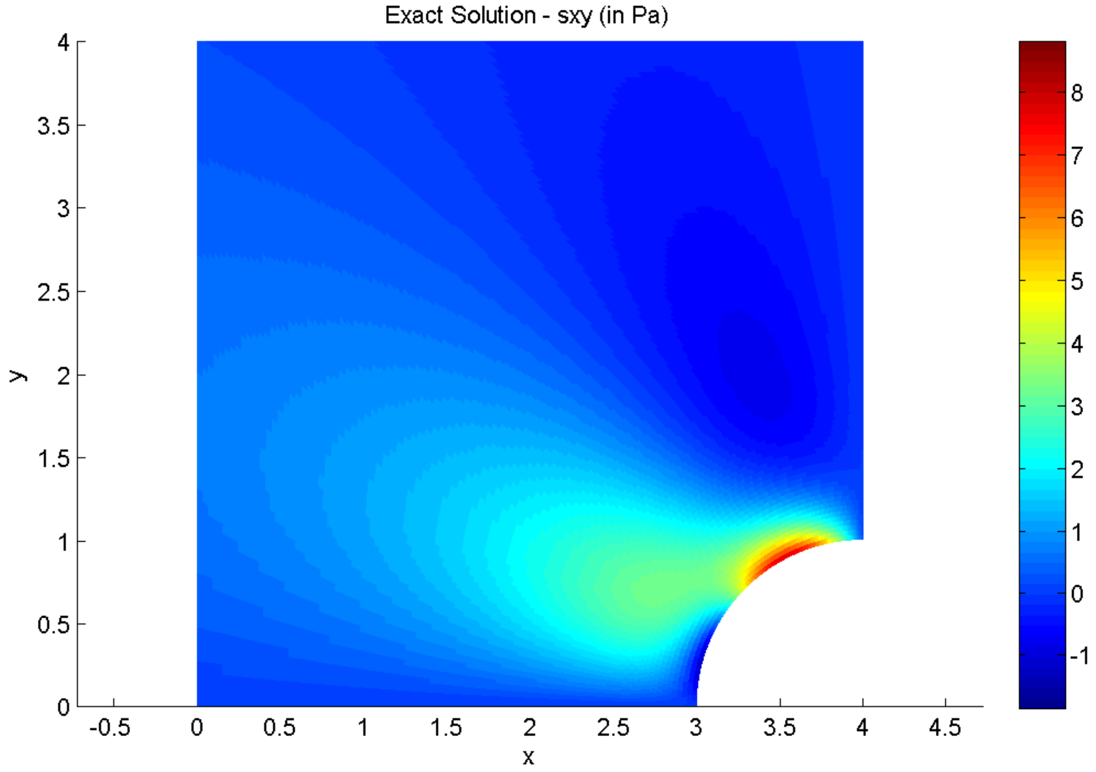


Figure 5.7

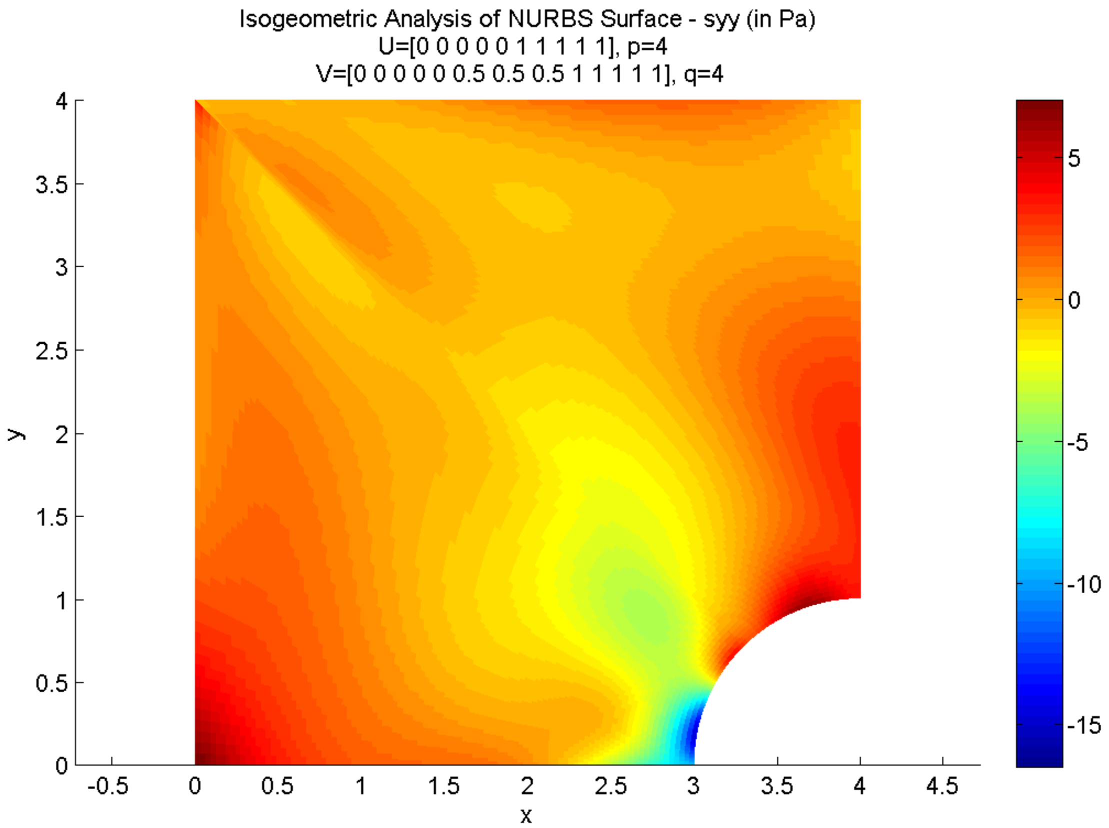
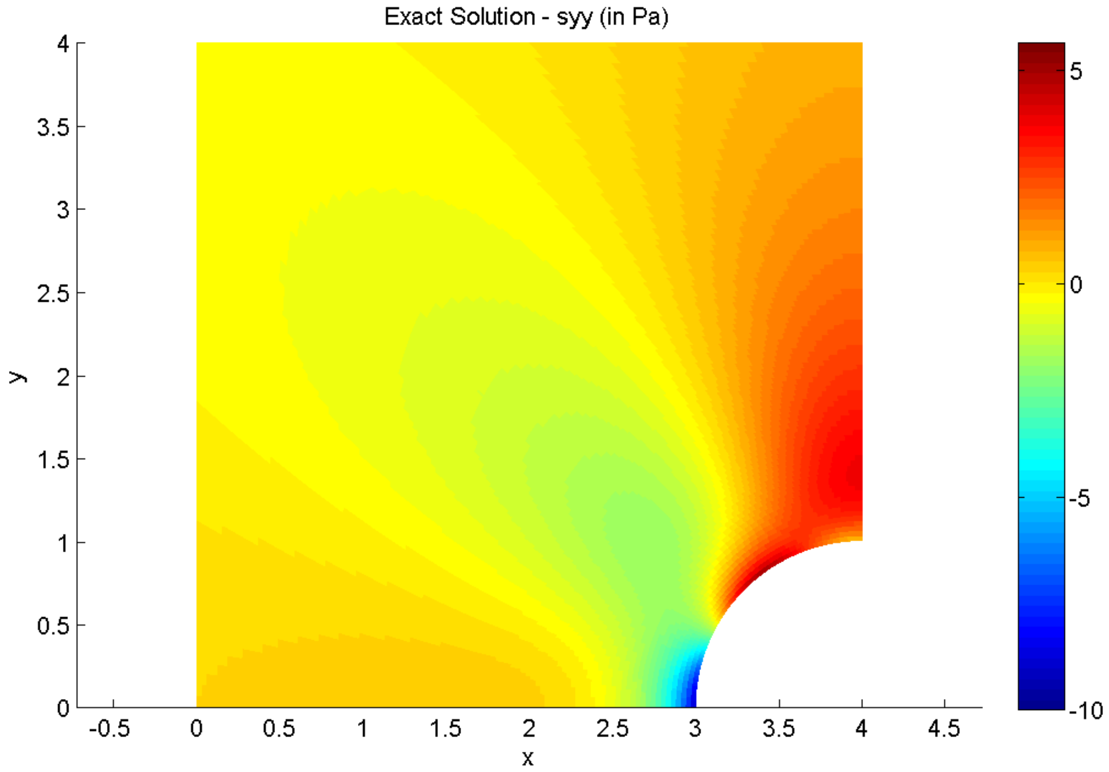

Isogeometric Analysis of NURBS Surface - syy (in Pa)
U=[0 0 0 0 0 1 1 1 1 1], p=4
V=[0 0 0 0 0 0.5 0.5 0.5 1 1 1 1 1], q=4
y
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
x
5
0
-5
-10
-15

Figure 5.8


Exact Solution - syy (in Pa)
y
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
x
5
0
-5
-10

PART III

NON-LINEAR ISOGEOMETRIC ANALYSIS WITH BOUC-WEN HYSTERESIS MODEL

1 HYSTERESIS AS A GENERAL PHENOMENON

Hysteresis is the dependence of a system not only on its current environment but also on its past environment. This dependence arises because the system can be in more than one internal state. To predict its future development, either its internal state (‘’local memory hysteresis’’) or its history (“non-local memory hysteresis”) must be known. If a given input alternately increases and decreases, the output tends to form a loop. However, loops may also occur because of a dynamic lag between input and output. Often, this effect is also referred to as hysteresis, or *rate-dependent hysteresis*. This effect disappears as the input changes more slowly; so many experts do not regard it as true hysteresis. More generally, hysteresis arises in phase transitions.

The term hysteresis comes from the Greek 'to be late, or come behind'. ***The physicist Ewing (1881) first coined the term to refer to effects (in terms of magnetization) that remain after the initial cause (the application of a magnetizing force) is removed.*** Such effects have subsequently been discovered or invoked in relation to a wide array of physical, biological and social phenomena. A general account of hysteresis as a system property has been provided in (M.A. Krasnosel'skii and A.V. Pokrovskii, Systems with Hysteresis, Springer-Verlag, Berlin (1989)). The key elements required to produce hysteresis are some form of nonlinearity in the way the elements in a system respond to some action; and heterogeneity in the elements and therefore in their responses to actions. The key implications of hysteresis are remanence, in that the application and reversal of an action will not be followed by a return to the *status quo ante*; and a selective memory, in which only the nondominated extremum values of actions remain in the memory bank, dominated extremum values being wiped out.

This nonlinear behaviour is encountered in many different areas of science in which the input–output relations between variables involve memory effects. Examples include magnetic hysteresis, ferromagnetic hysteresis, ferroelectric hysteresis, optical hysteresis, electron beam hysteresis, hysteresis in continuum mechanics, plasticity, friction, in the recently studied materials with shape memory, superconducting hysteresis, in filtration through porous media, adsorption and desorption hysteresis, economic hysteresis, hysteresis in biology, etc.

In natural systems hysteresis is often associated with irreversible thermodynamic change. ***In mechanical systems, hysteresis refers to the memory nature of inelastic behaviour where the restoring force depends not only on the instantaneous deformation but also on the history of the deformation***. Many artificial systems are designed to have hysteresis: for example, in thermostats and Schmitt triggers, hysteresis is produced by positive feedback to avoid unwanted rapid switching, the frictional sliders in damping devices, etc.

The detailed modelling of these systems using the laws of physics is an arduous task, and the obtained models are often too complex to be used in practical applications involving characterization of systems, identification or control. For this reason, alternative models of these complex systems have been proposed. A fundamental theory allowing a general mathematical framework for modelling hysteresis has not been developed up to now. For specific problems, models describing hysteretic systems can be derived from an understanding of physical laws. They combine some physical understanding of the hysteretic system along with some kind of black-box modelling. For this reason, some authors have called these models 'semi-physical' or phenomenological. In general, engineering practice seeks for alternative more simple models which, although not giving the 'best' description of the physical behaviour of the system, do keep relevant input–output features and are useful for characterization, design and control purposes.

In this context, several mathematical models have been proposed to describe the behaviour of hysteretic processes. The Duhem model uses the property that a hysteretic system's output changes its character when the input changes direction (frictional slider model); the Prandtl-Ishlinskii hysteresis operators has been proposed as a model for plasticity– elasticity and the Preisach model has been used for modelling electromagnetic hysteresis. In the areas of smart structures, civil engineering and the plasticity of materials, another model has been used extensively to describe the hysteresis phenomenon: the so-called Bouc–Wen model. It consists of a first-order nonlinear differential equation that relates the input displacement to the output restoring force in a rate-independent hysteretic way. The parameters that appear in the differential equation can be tuned to match the hysteresis loop of the system under study.

1.1 INTRODUCTION TO MATHEMATICAL MODELING OF HYSTERESIS (MAYERGOYZ, 2003)

The distinct feature of mathematical models of hysteresis nonlinearities with "nonlocal memories'' is that their future states depend on past histories of input variations. It turns out that memories of rate-independent hysteresis nonlinearities are quite selective. Indeed, only some past input extrema (not the entire input variations) leave their marks upon the future states of rate-independent hysteresis nonlinearities. Thus, special mathematical tools are needed to describe nonlocal selective memories of such hysteresis nonlinearities. The origin of such tools can be traced back to the landmark paper of Preisach. Preisach type models of hysteresis are constructed as superpositions of the simplest hysteresis nonlinearities-rectangular loops.

Hysteresis operators are naturally defined on sets of piecewise monotonic functions that do not form complete function spaces. This leads to the problem of continuous extension of hysteresis operators from the above sets to some complete function spaces. The reader interested in this type of mathematical problems is referred to the study by the mathematicians M. Krasnoselskii and A. Pokrovskii as well as to the books of A. Visintin and M. Brokate and J. Sprekels.

1.1.1 SCALAR HYSTERESIS IN THE CONTROL THEORY LANGUAGE

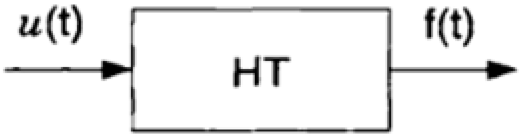


Figure 1.1

Consider a transducer of Figure 1.1, that can be characterized by an input *u(t)* and an output *f(t)*. This transducer is called a ***hysteresis transducer (HT) if its input-output relationship is a multibranch nonlinearity for which branch-to-branch transitions occur after input extrema.*** This multibranch nonlinearity is shown in Figure 1.2.

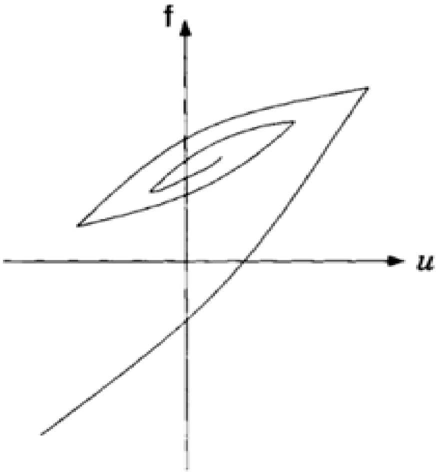


Figure 1.2

The term ***"rate-independent" means that branches of such hysteresis nonlinearities are determined only by the past extremum values (selective character) of input, while the speed (or particular manner) of input variations between extremum points has no influence on branching***. This statement is illustrated by Figure 1.3, where it is shown two different inputs $u_1(t)$ and $u_2(t)$ that successively assume the ***same extremum values but vary differently between these values***. Then, ***for a rate-***

independent HT, these two inputs will result in the same f-u diagram, provided that the initial state of the transducer is the same for both inputs.

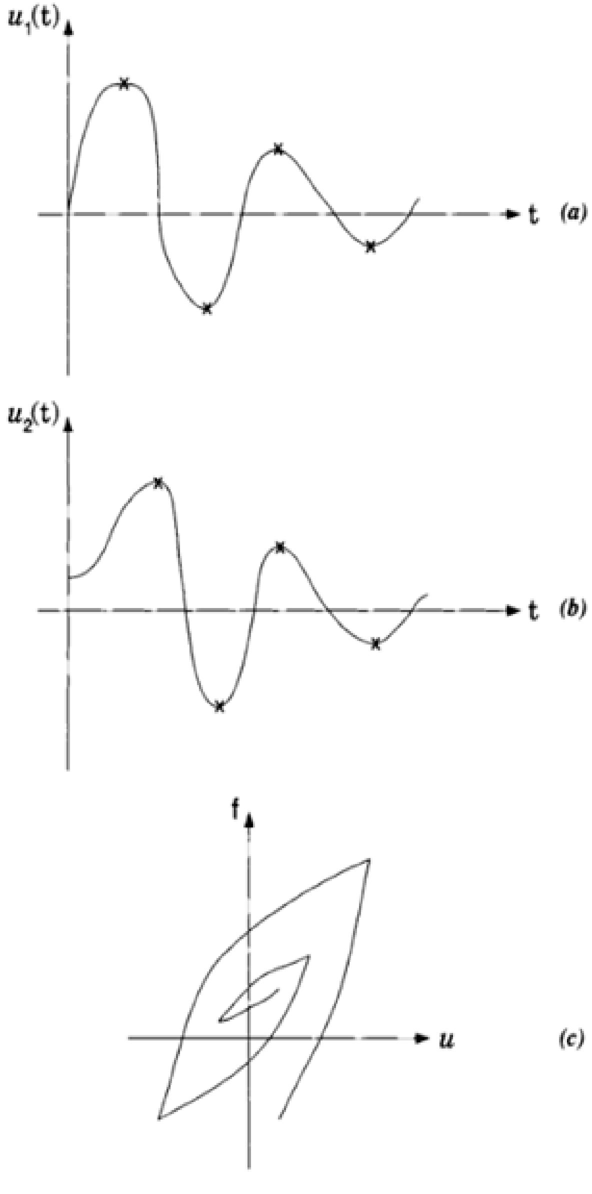


Figure 1.3

The given definition of rate-independent hysteresis is consistent with existing experimental facts. Indeed, it is known in the area of magnetic hysteresis that a shape of major (or minor) loop (see Figure 1.4) can be specified without referring to how fast magnetic field *H* varies between two extremum values $+H_m$ and $-H_m$. This indicates that time effects are negligible and the given definition of a rate-independent hysteresis transducer is an adequate one. It is worthwhile to keep in mind that, ***for very fast input variations, time effects become important and the given definition of rate-independent hysteresis fails***. In other words, this definition (as any other definition) has its limits of applicability to real life problems.

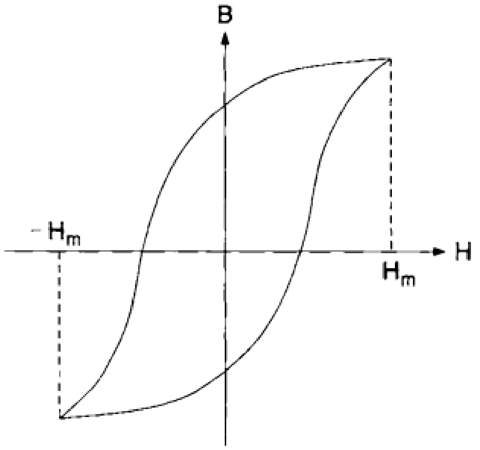


Figure 1.4

It is also important to stress that the notion of ***rate-independent hysteresis implies three distinct time scales***. The first is the ***time scale of fast internal dynamics of the transducer (in control theory framework)***. The second is the ***time scale on which observations (measurements) are performed***. This time scale is much larger than the time scale of internal transducer dynamics so that every observation can be identified with a specific output value of the transducer. The third is the ***time scale of input variations***. This time scale is much larger than the observation time scale so that every measurement can be associated with a specific value of input.

In the existing literature, the ***hysteresis phenomenon is by and large linked with the formation of hysteresis loops (looping)***. ***This may be misleading and create the impression that looping is the essence of hysteresis***. In this respect, the given definition of hysteresis emphasizes the fact that ***history dependent branching constitutes the essence of hysteresis, while looping is a particular case of branching***. Indeed, ***looping occurs when the input varies back and forth between two consecutive extremum values, while branching takes place for arbitrary input variations***. From the given definition, it can also be concluded that ***scalar hysteresis can be interpreted as a nonlinearity with a memory which reveals itself through branching***.

In the given definition of hysteresis, the physical meanings of the input $u(t)$ and the output $f(t)$ were left unspecified. It was done deliberately, for the sake of mathematical generality. However, it is not difficult to specify the meanings of $u(t)$ and $f(t)$ in particular applications. For instance, in magnetism $u(t)$ is the magnetic field and $f(t)$ is the magnetization, in mechanics $u(t)$ is the displacement and $f(t)$ the force, in adsorption $u(t)$ is the gas pressure and $f(t)$ is the amount of material adsorbed. The notion of hysteresis transducer may have different interpretations as well.

All rate-independent hysteresis nonlinearities fall into two general classifications: (a) hysteresis nonlinearities with local memories, and (b) hysteresis nonlinearities with nonlocal memories. The ***hysteresis nonlinearities with local memories are characterized by*** the following property. ***The value of output $f(t_0)$ at some instant of time t_0 and the values of input $u(t)$ at all subsequent instants of time $t \geq t_0$ uniquely predetermine the value of output $f(t)$ for all $t > t_0$. In other words***, for hysteresis transducers with local memories ***the past exerts its influence upon the future through the current value of output***. This is not the case for ***hysteresis transducers with nonlocal memories***. For such transducers, ***future values of output $f(t)$ ($t \geq t_0$) depend not only on the current value of output $f(t_0)$ but on past extremum values of input as well***.

Typical examples of hysteresis nonlinearities with local memories are shown in figures below. Figure 1.5 shows the simplest hysteresis nonlinearity with local memory. It is specified by a major loop which is formed by ascending and descending branches. These branches are only partially reversible (their vertical sections are not reversible). This type of hysteresis nonlinearity is characteristic, for instance, of single Stoner-Wolhfarth magnetic particles. For this type of hysteresis, branching occurs if extremum values of input exceed $+u_m$ or $-u_m$.

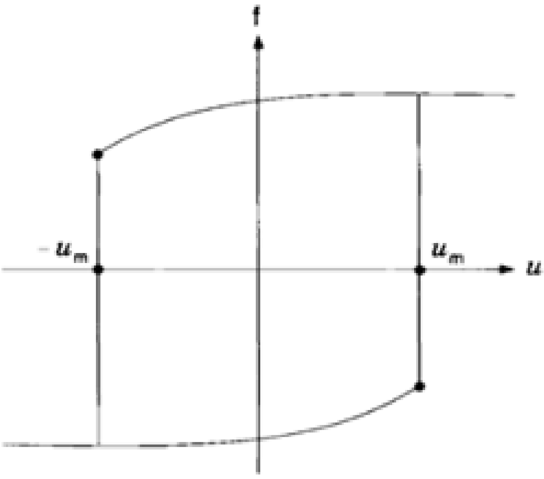


Figure 1.5

A more complicated type of hysteresis nonlinearities with local memories is illustrated by Figure 1.6. There is a set of inner curves within the major loop and only one curve passes through each point in the *f-u* diagram. These curves are fully reversible and can be traversed in both directions, for a monotonically increasing and decreasing input *u(t)*. For this type of hysteresis, branching may occur only when ascending or descending branches of major loops are reached.

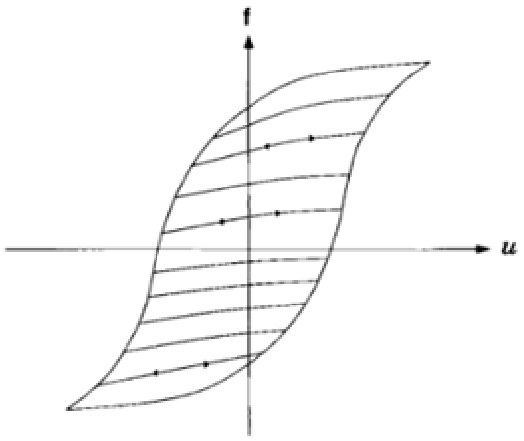


Figure 1.6

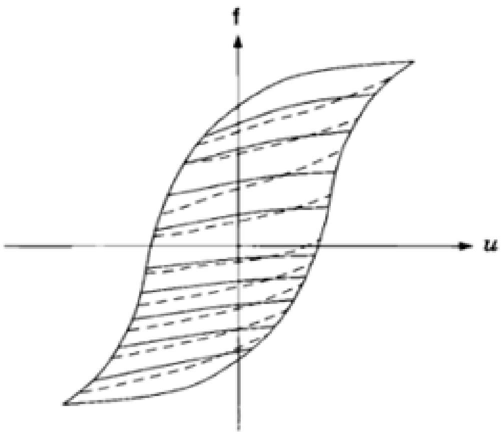


Figure 1.7

A hysteresis nonlinearity with local memory that has two sets of inner curves (the ascending and descending curves) is shown in Figure 1.7. This type of hysteresis was probably first described by Madelung [6] in the beginning of the century, and afterwards it was independently invented by many authors time and time again [7], [8]. For this hysteresis nonlinearity, only one curve of each set passes through each point in the *f-u* diagram. If the input *u(t)* is increased, the ascending curve is followed. If it is decreased, the descending curve is traced. Thus, branching occurs for any input extremum. However, in general, minor loops are not formed; if *u(t)* varies back and forth between the same two values, the output usually exhibits a continued upward drift.

It is clear from the above examples that ***all hysteresis nonlinearities with local memories have the following common feature: every reachable point in the f-u diagram corresponds to a uniquely defined state***. ***This state predetermines the behavior of HT in exactly one way for increasing u(t) and exactly one way for decreasing u(t)***. ***In other words, at any point in the f-u diagram there are only one or two curves that may represent the future behavior of HT with local memory*** Figure 1.8.

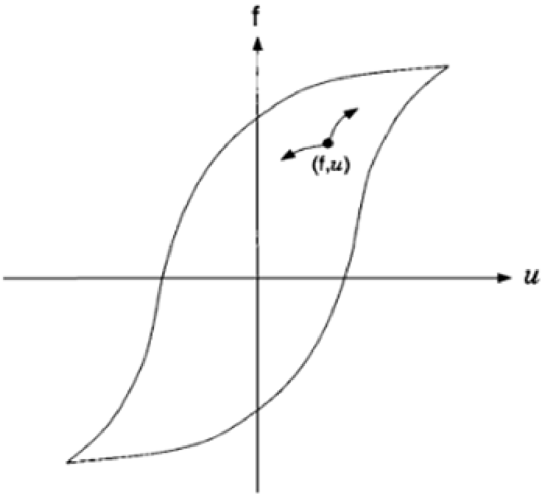


Figure 1.8

This is not true ***for hysteresis transducers with nonlocal memories***. In the latter case, ***at any reachable point in the f-u diagram there is an infinity of curves that may represent the future behavior of the transducer Figure 1.9***. ***Each of these curves depends on a particular past history, namely, on a particular sequence of past extremum values of input. By analogy with the random process theory***, ***hysteresis nonlinearities with local memories can be called Markovian hysteresis nonlinearities, while hysteresis nonlinearities with nonlocal memories are non-Markovian. It is clear that hysteresis nonlinearities with nonlocal memories are much more complicated than those with local memories.***

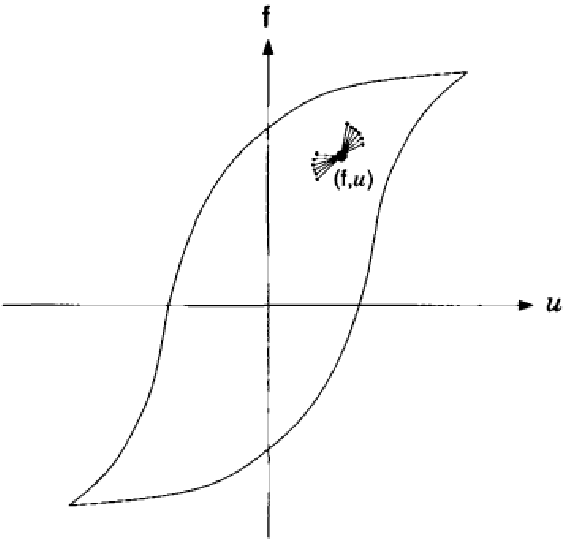


Figure 1.9

Mathematical models of hysteresis nonlinearities with local memories have been extensively studied by using differential and algebraic equations. These models have achieved high level of sophistication that is reflected, for instance, in publications [9-

12]. However, ***the notion of hysteresis nonlinearities with local memories is not consistent with experimental facts***. For instance, it is reported in [13] that ***crossing and partially coincident minor loops have been experimentally observed***. These loops are ***schematically shown in Figure 1.10 & Figure 1.11***, respectively. The existence of crossing minor loops attached to a major loop is more or less obvious, while the presence of partially coincident minor loops is a more subtle phenomenon. ***The existence of crossing and partially coincident minor loops clearly suggests that the states of the corresponding hysteresis transducers are not uniquely specified by their inputs and outputs***. ***Thus, hysteresis of this transducer does not have a local memory.***

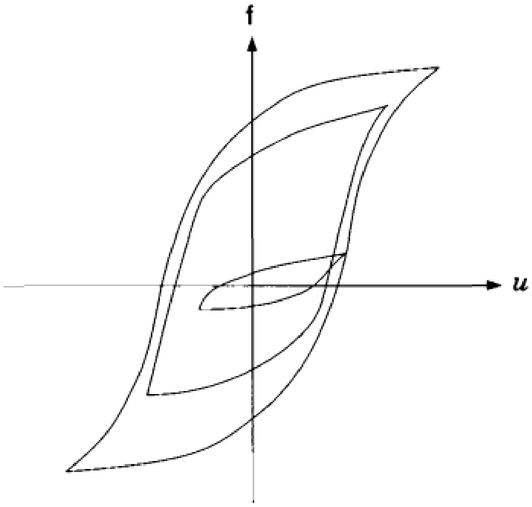


Figure 1.10

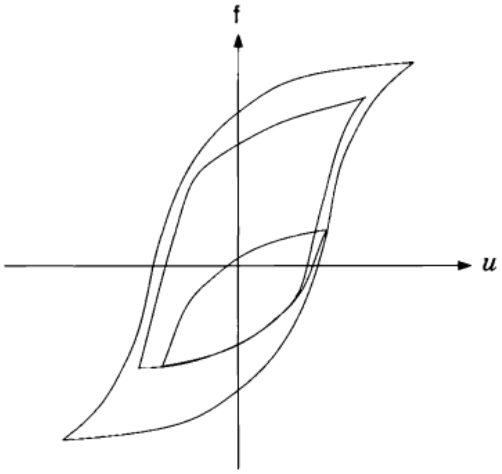


Figure 1.11

A question arises, ***why these models are need***. The answer is that ***the hysteresis transducer is usually a part of a system***. As a result, ***its input is not known beforehand, but is determined by the interaction of the transducer with the rest of the system***. ***Since the input of HT is not predictable a priori, it is impossible to***

specify ahead of time the branches of hysteresis nonlinearity which will be followed in a particular regime of the system. This is the main impediment as far as self-consistent mathematical descriptions of systems with hysteresis are concerned. To overcome the difficulty mentioned above, ***mathematical models of hysteresis are needed***. ***These models represent new mathematical tools that by themselves (due to their structure) will detect and accumulate input extrema and will choose appropriate branches of the hysteresis nonlinearity according to the accumulated histories. Coupled together with mathematical description of the rest of the system, these models will constitute complete and self-consistent mathematical descriptions of systems with hysteresis. Without such models, the self-consistent mathematical descriptions of systems with hysteresis are virtually impossible.***

1.1.2 VECTOR HYSTERESIS

We next turn to the discussion of vector hysteresis. This hysteresis can be characterized by a vector input $\vec{u}(t)$ and vector output $\vec{f}(t)$, Figure 1.12. Two- and three-dimensional vector inputs and vector outputs are most relevant to practical applications. However, the formal mathematical generalization of these models to n dimensions (n > 3) is straightforward. It is believed that such a generalization will be performed by the reader if it is needed.

Figure 1.12

The most immediate problem we face is how to define vector hysteresis in a mathematically rigorous as well as physically meaningful way. To do this, it is important to understand what constitutes in the case of vector hysteresis the essential part of past input history that affects the future variations of output. ***In the case of scalar rate-independent hysteresis, experiments show that only past input extrema (not the entire input variations) leave their mark upon future states of hysteresis nonlinearities. In other words, the memories of scalar hysteresis nonlinearities are quite selective***. ***There is no experimental evidence that this is the case for vector hysteresis***. As a result, ***we must resign ourselves to the fact that all past vector input values may affect future output variations***. ***The past input variations can be characterized by an oriented curve L traced by the tip of the vector input*** $\vec{u}(t)$, Figure 1.13.

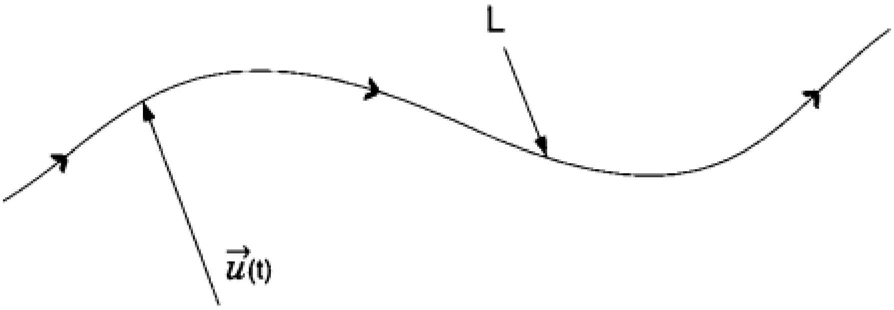


Figure 1.13

Such a curve can be called an input "hodograph". Vector rate-independent hysteresis can be defined as a vector nonlinearity with the property that the shape of curve L and the direction of its tracing (orientation) may affect future output variations, while the speed of input hodograph tracing has no influence on future output variations. Next, we demonstrate that ***scalar rate-independent hysteresis can be construed as a particular case of vector rate-independent hysteresis***. This case is realized ***when the vector input is restricted to vary along only one direction*** (one line). In fact, it can be successfully argued (at least in the area of magnetics) that there is no such a thing as scalar hysteresis. Whenever we talk about scalar hysteresis, we are actually dealing with some specific properties of vector hysteresis that have been observed for vector input variations restricted to some fixed directions. It is apparent that, for unidirectional input variations $\vec{u}(t) = \vec{a} \cdot u(t)$, input hodographs (see Figure 1.14) are uniquely determined by current values of u(t) as well as by past extrema of *u(t)*. In this sense, vector rate-independent hysteresis is reduced to scalar rate-independent hysteresis with the input *u(t)*.

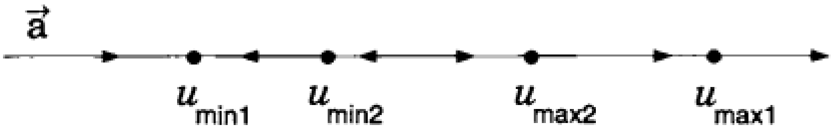


Figure 1.14

Next, we shall give another ***equivalent definition of rate-independent vector hysteresis in terms of input projections. This definition will be convenient in the design of mathematical models of vector hysteresis***. Consider input projection along some arbitrary chosen direction. As the vector $\vec{u}(t)$ traces the input hodograph, the input projection along the chosen direction may achieve extremum values at some points of this hodograph. In this sense, the extrema of input projection along the chosen direction samples certain points of the input hodograph. If the projection direction is continuously changed, then the extrema of input projections along the continuously changing direction will continuously sample all points on the input hodograph. In this way, the past extrema of input projections along all possible directions reflect the shape of input hodograph and, consequently, the past history of input variations. Thus, ***we arrive at the definition of vector rate-independent hysteresis as a vector nonlinearity with the property that past extrema of input projections along all possible directions may affect future output values***. It is clear that mathematical models of vector hysteresis are imperative for self-consistent descriptions of systems with vector hysteresis. ***These models should be able to detect and store past extrema of input projections along all possible directions and choose the appropriate value of vector output according to the accumulated history.***

We are deal exclusively with the mathematical models of hysteresis that are purely phenomenological in nature. Essentially, these models represent the attempt to describe and generalize experimental facts. They provide no insights into specific physical causes of hysteresis. Nevertheless, they have been and may well continue to be powerful tools for device design. There are, however, fundamental models of hysteresis which attempt to explain experimental facts from first principles as we will see in next section.

Summarizing the above discussion, it can be concluded that the phenomenological approach is more directly connected with macroscopic experimental data. For this reason, it is of a great value to device designers. *The fundamental approach, on the other hand, is intimately related to material structure and, therefore, it can be useful in the design of new materials.*

2 THE BOUC-WEN HYSTERESIS MODEL (TRIANTAFYLLOU SAVVAS, 2011)

The Bouc–Wen model is a smooth endochronic model that is often used to describe hysteretic phenomena. It was introduced by Bouc and extended by Wen. The versatility of the Bouc–Wen model has been demonstrated in numerous cases. It has been used extensively for the modeling of MR dampers, wood joints, welded steel joints and isolation devices, to name a few. A survey on the implementation of the Bouc–Wen hysteretic model can be found in the work of Ismail et al. Bouc presented his formulation (1967) of the single degree degrading hysteresis model with pinching. Subsequently, many modifications have been introduced, such as the Bouc-Wen model (Wen, 1976, 1980), the Baber-Noori model (Baber and Wen 1980, Baber et al. 1986) and the Reinhorn model (Sivaselvan and Reinhorn, 2000). These hysteresis models –also known as smooth hysteretic models- are capable of simulating different types of hysteretic behavior using a single smooth hysteretic function affected by a set of user-defined parameters.

The last decades Bouc-Wen hysteretic model is proven very versatile in expressing a wide range of hysteretic response including stiffness degradation, strength deterioration as well as pinching phenomena in reinforced concrete, steel members and connections, wood etc., (Foliente G. C, 1995). In addition, considerable effort has been devoted to alleviate Bouc-Wen model from inconsistencies regarding thermodynamic admissibility, (Erlicher and Point 2004, Erlicher and Bursi, 2009) and violation of plasticity postulates, (Charalampakis and Koumousis, 2009). The rate form of evolution equations, derived also on the basis of endochronic theories of plasticity (Valanis, 1971), is capable of expressing in an integrated way the phenomenological hysteretic behavior at the component level. This facilitates direct incorporation of identified model parameters for various members and/or connections leading to a more effective and controllable analysis, as compared to the pointwise stress-strain relations required in standard Finite Element Analysis. These features are revealed at the cost of extending the elastic finite elements by introducing additional stiffness matrices that account for inelastic behavior and the inherent interaction of different components of stress.

During the last decade, Bouc-Wen model has been adopted by many researchers, (Pires, 1993, Choi and Lee, 2001) as a robust and accurate tool, to simulate the hysteretic behavior of various materials. At the same time, techniques were developed for the identification of the Bouc-Wen model parameters utilizing among others, advanced analytical techniques, as in Chatzi and Smyth (2008), evolutionary identification approaches, (Charalampakis and Koumousis, 2008a) and more recently in Chang et al. (2010) using wavelet analysis. Rigorous mathematical proofs on the convergence of the identified model parameters to their true counterparts can be found in (Faycal Ikhouane, Jose Rodellar, 2007).

There are also implementations in control of mechanical systems and structures with Bouc–Wen hysteretic behavior. In this sense, it may be useful to distinguish between active and semi-active control. A control law is said to be active when the control signal directly feeds an actuator that applies the desired feedback control force. With an active control scheme, energy is injected into the closed-loop system. A control law is semi-active when the corresponding actuator does not pour energy into the closed loop. Instead, the control signal is generated by the controller to modify the characteristics of an adaptive passive-like actuator. Examples of semi-active actuators are the devices based on smart materials, in particular the magnetorheological dampers. (Faycal Ikhouane, Jose Rodellar, 2007)

2.1 THE CONCEPT OF HYSTERESIS IN STRUCTURAL MECHANICS

Consider the single degree of freedom (s.d.o.f.) oscillator presented in Figure 2.1. The oscillator exhibits an elastic-perfectly plastic material behavior with a yield stress σ_y.

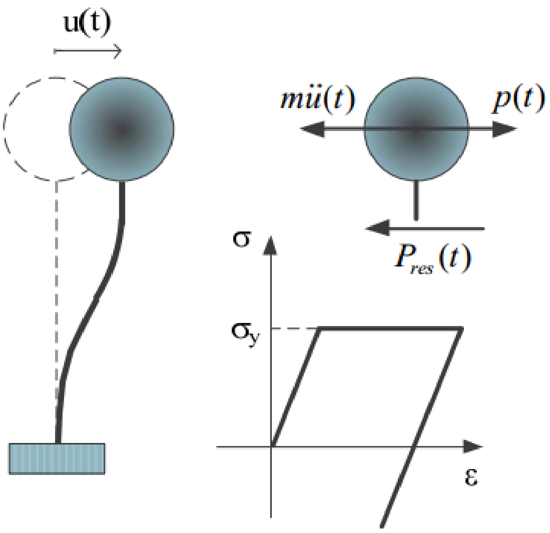


Figure 2.1 Single degree of freedom oscillator under cyclic excitation

The response of the nonlinear oscillator is depicted in more detail in Figure 2.2. For stresses smaller than the yield stress, material behavior is defined by Hooke's law, so that the elastic range of the response is evaluated as:

Eq. 2.1

$$\sigma(\varepsilon) = E\varepsilon, \qquad |\sigma| \leq \sigma_y$$

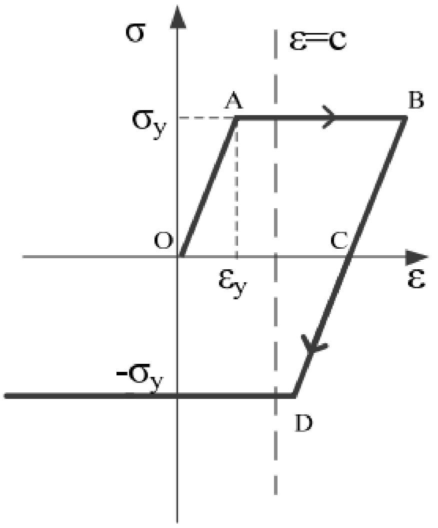


Figure 2.2 Hysteretic loop

and there is 1:1 correspondence between the input and the output.

However, there are at least two possible stress states $\sigma \in [-\sigma_y, \sigma_y]$ that correspond to an arbitrary strain level $\varepsilon = c, c \in [\varepsilon_y, +\infty)$, larger than the yield strain ε_y. Thus, there does not exist a function $\sigma(\varepsilon)$ that can uniquely map the current level of strain to the current level of stress even for the trivial case of an elastic-perfectly plastic material. The mathematical theory of hysteresis tries to define a proper output function $\sigma = \sigma(t) : [0,\ T] \to \mathbb{R}$ given an input function $\varepsilon(t) : [0,\ T] \to \mathbb{R}$, such that the derived vector phase space $(\sigma(t),\ \varepsilon(t))$ coincides with the curve presented in Figure 2.2.

Thus, the mechanical problem of hysteresis is translated into the mathematical problem of defining an operator, denoted herein as the hysteresis operator $\mathrm{B}[\mathrm{In}](t)$ where $\mathrm{In}: [0, \mathrm{T}] \to \mathbb{R}$, is an arbitrary time history input such as displacement, strain e.t.c. From physical point of view, the functional has to be rate independent since the hysteretic energy accumulated over consequent loading and unloading cycles does not depend on the rate of the input or output functions. Moreover, the hysteretic operator must be piecewise monotone, as the shape of the hysteretic loops implies (e.g. with respect to Figure 2.2, monotonically increasing in path OAB and monotonically decreasing in path BCD). Finally, the operator must have some property of memory which in mathematical terms is covered by the notion of causality (Logemann and Mawby, 2003).

It is evident from the approach presented in this paragraph that the notion of hysteresis is coped with mathematical tools that are indifferent to the input and output functions describing the hysteretic loop. For this reason, the theory presented herein constitutes a phenomenological approach. Nevertheless, there are aspects that are directly connected to the mechanical properties of hysteresis such as energy dissipation mechanisms and hardening effects that will be addressed through this theory in a consistent way.

2.2 THE INITIAL DERIVATION OF THE BOUC-WEN MODEL

Bouc (1967) studied the response of a single degree of freedom oscillator with mass m and a hysteretic restoring force $P_{res}(t)$. According to the notions described in paragraph 2.1, ***the hysteretic restoring force is considered to be the result of a hysteretic operator B over the displacement*** $\boldsymbol{u(t)}$***:***

Eq. 2.2

$$P_{res}(t) = B[u](t)$$

The equation of motion of the s.d.o.f oscillator is then expressed as:

Eq. 2.3

$$m\frac{d^2u}{dt^2} + B[u](t) = p(t)$$

where $p(t)$ is the external force. Based on the initial work of Volterra (1928) for an internal restoring force with hysteretic properties, ***Bouc defined operator B as an integral scheme***:

Eq. 2.4

$$B[u](t) = \int_{t_0}^{t} \mu(t,t')\, du\,(t')$$

Where ***due to μ which is an intrinsic kernel with t*** $> t'$***, the operator accumulates the history of the inputs from each sub-instant t' till the current time t and being able to evaluate the restoring force with history effects***. Furthermore, ***the following assumption is adopted for the kernel:***

Eq. 2.5

$$\mu(t,t') = \mu(t-t')$$

that is, ***the evolution of μ is irrelevant to the velocity of the oscillator***. ***The property of piecewise monotony is met by requiring that the kernel is a bounded, continuous and decreasing function of the time increment*** $\Delta t = t - t'$. Thus the following expressions hold:

Eq. 2.6

$$0 \leq \mu(\Delta t) < \infty, \quad \frac{d\mu(\Delta t)}{d\Delta t} \leq 0$$

The condition of causality is met since the upper limit of the integral in Eq. 2.4 is the current time t and the current value of the operator is the cumulative sum of the kernel over the displacement.

Since the kernel ***depends on the time-step Δt, the derived hysteretic restoring force depends on the rate of the imposed load***, yielding a formulation ***not eligible for a rate-independent plasticity formulation. To overcome this deficiency, Bouc introduced the following transformation***:

Eq. 2.7

$$\varDelta t \rightarrow \varDelta u(t,t') \Longrightarrow \mu(\varDelta t) \rightarrow \mu\big(\varDelta u(t,t')\big)$$

mapping the time increment Δt to the corresponding displacement increment Δu. Thus, the hysteretic force is expressed as:

Eq. 2.8

$$B[u](t) \equiv P_{res}(t) = \int_{t_0}^{t} \mu\big(\varDelta u(t,t')\big) du(t')$$

where ***the kernel μ is now a bounded, positive and decreasing function of Δu***:

Eq. 2.9

$$0 \leq \mu(\varDelta u) < \infty, \quad \frac{d\mu(\varDelta u)}{d\varDelta u} \leq 0$$

Similarly, ***the intrinsic time step Δu is a positive, increasing function, since time t is larger than t'***. ***Different definitions of the intrinsic time step lead to different hysteresis formulations***, given that they all comply to Eq. 2.9. A typical example that is consistent with the above remarks is the following:

Eq. 2.10

$$d\varDelta u = |du| = \left|\frac{du}{dt}\right| dt = \left|\frac{du}{d\tilde{t}}\right| d\tilde{t} := d\theta, \tilde{t} = \varphi(t)$$

The mathematical expression of the restoring force introduced in ***Eq. 2.8 though rigorous, fails to clarify the key parts of the restoring force in terms of mechanics***. Trying ***to clarify the physical properties*** of the hysteretic operator *B*, ***Bouc introduced two arbitrary continuous scalar functions f, Φ with the following properties***:

Eq. 2.11

$$f: \mathbb{R} \rightarrow \mathbb{R}\,;\ f(0) = 0\,;\ |f(u_1) - f(u_2)| \leq K_1(A)|u_1 - u_2|$$

$$\varPhi: \mathbb{R} \rightarrow \mathbb{R}\,;\ \varPhi(0) = 0\,;\ |\varPhi(u_1) - \varPhi(u_2)| \leq K_2(A)|u_1 - u_2|$$

where K_1, K_2 constants, for every A, u_1, u_2. These functions are defined with the Lipschitz condition to gain some of the required properties of a hysteresis operator (piecewise monotone, decreasing functions). A generalization of the Volterra expression Eq. 2.8 is then established, such that:

Eq. 2.12

$$P_{res}(t) = f\big(u(t)\big) + z(t)$$

$$z(t) = \Phi\big(u(t')\big) \int_{t_0}^{t} \mu\big(\Delta_u(t, t')\big) du$$

Since Eq. 2.12 hold for every function f, Φ, they also hold for:

Eq. 2.13

$$f\big(u(t)\big) = ku(t) + \tilde{f}\big(u(t)\big)$$

where $\tilde{f}(.)$ is also a continuous scalar function.

Thus, substituting Eq. 2.13 into Eq. 2.12 ***a clear distinction is made between the linear elastic component of the restoring force*** $ku(t)$***, the nonlinear elastic term*** $\tilde{f}\big(u(t)\big)$ ***and the nonlinear, history dependent, component*** $z(t)$. ***Operator Φ depends on the displacement time history u(t)***, ***so that*** the expression of the ***nonlinear component is irrelevant to the displacement rate***. The integral of the ***second of expressions of Eq. 2.12*** is a Lebesgue – Stieltjes integral (Halmos, 1974) that can be ***cast in the following Riemannian form:***

Eq. 2.14

$$z(t) = z\big(\theta(t)\big) = \int_{0}^{\theta(t)} \mu(\theta(t) - \theta') \frac{d\Phi}{du} \frac{du}{d\theta'} d\theta'$$

where $\theta(t)$ ***is an intrinsic time*** complying to Eq. 2.10. The integral of ***Eq. 2.14 is the "memory" of the dynamical system, since*** $z(t)$ ***is an integral over the time period*** $t - t_0$***.*** As such, it adheres, by definition, to the Volterra property. Furthermore, ***since the kernel of the integral does not explicitly depend on t, the hysteretic parameter*** $z(t)$ ***is by definition rate-independent.*** Thus, the formulation proposed by Bouc is a ***formal, continuous and stable hysteretic operator*** (Brokate et al., 1993).

Thus, the single degree of freedom equation of motion is evaluated as:

Eq. 2.15

$$\begin{cases} m\dfrac{d^2w}{dt^2} + P_{res}(t) = p(\quad) \\ P_{res}(t) = ku(t) + \tilde{f}\big(u(t)\big) + z(t) \\ z(t) = \displaystyle\int_0^{\theta(t)} \mu(\theta(t) - \theta')\frac{d\Phi}{du}\frac{du}{d\theta'}d\theta' \end{cases}$$

Bouc imposed the following ***expression on the variation of u***:

Eq. 2.16

$$\Delta u(t,t') = \theta(t) - \theta(t') = \theta - \theta' = \int_{t'}^{t} d\theta(\tau) = \int_{t'}^{t} \left|\frac{du}{d\tau}\right| d\tau := V_{t'}^{t}u$$

where $\boldsymbol{V_{t'}^{t}u}$ ***is defined as the total variation of u on*** $[\boldsymbol{t'},\boldsymbol{t}]$. The following relation proven to hold:

Eq. 2.17

$$d\theta(\tau) = |du(\tau)|$$

Any type of function can serve as a kernel. However, it can be proved that the ***differential equation of the nonlinear component can be derived always for an exponential kernel.***

2.2.1 THE EXPONENTIAL KERNEL CASE

Consider the following case where:

Eq. 2.18

$$f(t) = 0, \qquad \Phi(u) = u$$

Eq. 2.18 fulfills the properties set on expression Eq. 2.11.

Substituting into the second of expressions of Eq. 2.15, the following expression derived for the restoring force:

Eq. 2.19

$$P_{res} = z(t) = \int_0^{\theta(t)} \mu(\theta(t) - \theta')\frac{du}{d\theta'}d\theta'$$

The kernel in the integral of Eq. 2.19 is considered as an exponent of the following form:

Eq. 2.20

$$\mu(\theta) = Ae^{-\beta\theta}, \;\; A,\beta > 0$$

that complies with Eq. 2.9. Differentiating Eq. 2.19:

Eq. 2.21

$$dP_{res} = dz(\theta)$$

Substituting the integral form of $z(t)$ into Eq. 2.21, the following expression is derived:

Eq. 2.22

$$\begin{aligned} dP_{res} &= \left[Ae^{-\beta\theta}du(\theta')\right]_{\theta'=\theta} - \left[\int_{\theta_0}^{\theta} \frac{\partial\left(Ae^{-\beta(\theta-\theta')}\right)}{\partial\theta}\frac{du(\theta')}{d\theta'}d\theta'\right]d\theta \\ &= Adu(\theta) - \beta\left[\int_{\theta_0}^{\theta} A\, e^{-\beta(\theta-\theta')}\frac{du(\theta')}{d\theta'}d\theta'\right]d\theta \end{aligned}$$

Finally, taking into account the definition of the kernel introduced into Eq. 2.20 the following equation is derived:

Eq. 2.23

$$dP_{res} = dz = Adu - \beta z d\theta$$

Or equivalently in rate form:

Eq. 2.24

$$dP_{res} = \frac{dz}{dt} = A\frac{du}{dt} - \beta z\frac{d\theta}{dt}$$

Combining the first and second of Eq. 2.24, the following rate form is derived for the hysteretic parameter $z(t)$:

Eq. 2.25

$$\frac{dz}{dt} = A\frac{du}{dt} - \beta z\frac{d\theta}{dt}$$

Finally, substituting Eq. 2.17 into Eq. 2.25, the following, trivial equation of the Bouc model is derived:

Eq. 2.26

$$\frac{dz}{dt} = [A - \beta z \cdot sign(du)]\frac{du}{dt} \qquad A, \beta > 0$$

and relation Eq. 2.15 is rewritten as:

Eq. 2.27

$$\begin{cases} \dfrac{d^2u}{dt^2} + P_{res}^{B}(t) = p(t) \\ \dfrac{dP_{res}^{B}}{dt} = \dfrac{dz}{dt} = [A - \beta z \cdot sign(du)]\dfrac{du}{dt} \end{cases}$$

Eq. 2.27 correspond to the simple case of a perfectly nonlinear s.d.o.f oscillator. The second of Eq. 2.27 can be solved by quadratures and the restoring force is established as a function of the displacement u:

Eq. 2.28

$$P_{res}(u) = z(u) = \frac{A}{\beta \cdot sign(du)}\left(1 - e^{-\beta \cdot sign(du)u}\right)$$
$$= \frac{A}{\beta} sign(du)\left(1 - e^{-\beta \cdot sign(du)u}\right)$$

Referring to Eq. 2.20 one can assume without loss of generality that A=Cβ. Thus, Eq. 2.28 is rewritten as:

Eq. 2.29

$$P_{res}(u) = C \cdot sign(du)\left(1 - e^{-\beta \cdot sign(du)u}\right)$$

Different values of *C* and *β* give rise to different hysteretic loops with the rigid plastic body being an upper limit. In the limit case where $\beta \to \infty$ the restoring force coincides with the expression of the perfect slider with unit threshold (Duhem hysteresis model).

Eq. 2.30

$$\lim_{\beta \to \infty} P_{res} = C \cdot sign(du)$$
$$F_s = \mu \cdot N \cdot sign(du)$$

where F_s is the friction force, μ the coefficient of friction and *N* the normal force. Thus, ***the trivial case of Bouc-Wen hysteresis smoothens the standard expression of the friction force*** by merely relying on the mathematical expression of hysteresis as established by the pioneering work of Volterra (1928).

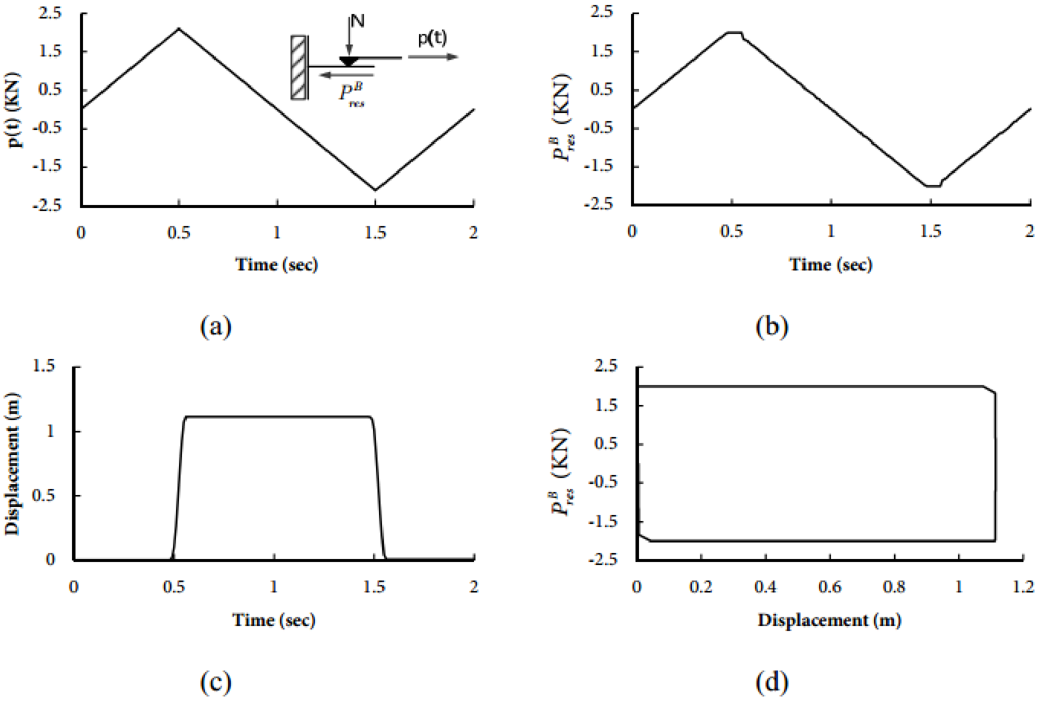


Figure 2.3 (a) External Force (b) Friction Force (c) Displacement on slider (d) Friction Force-Displacement hysteretic loop

As an example, the response of the dynamic system presented in Eq. 2.27 is examined with constants $C=2KN$ and $\beta=10000$ under cyclic loading. The excitation is presented in Figure 2.3 (a). In Figure 2.3 (b) and (c) the time-history of the friction force and the displacement are presented respectively. As predicted by Eq. 2.30, the system evolves as a perfect slider, with zero displacement until the external force reaches the sliding threshold defined by the constant C. The corresponding hysteresis loop is presented in Figure 2.3 (d). As expected, a permanent displacement is observed after full unloading due to the dissipative nature of the friction force.

2.3 FROM CLASSICAL PLASTICITY TO BOUC-WEN HYSTERESIS

2.3.1 DECOMPOSING THE BOUC-WEN HYSTERETIC MODEL

In this work, the Bouc-Wen model as introduced in Wen, (1980) and later modified is used as the basis for every subsequent step of analysis:

Eq. 2.31

$$\begin{cases} \ddot{u} + c\dot{u} + P_{res}^{BW} = p(t) \\ P_{res}^{BW} = \alpha K u + z \\ \dot{z} = (1-\alpha)K\left[A - |z|^n\left(\beta + \gamma \cdot sign(z\dot{u})\right)\right]\dot{u} \end{cases}$$

Where:

c: is the viscous damping coefficient,

P_{res}^{BW}: is the Bouc-Wen restoring force,

z: is the hysteretic part of the restoring forces,

α: is the post-elastic to elastic stiffness ratio,

K: is the elastic stiffness of the oscillator,

A, *β*, *γ:* are Bouc-Wen model parameters.

Parameter *A* has been proven to be redundant in subsequent works (Ma et al, 2004) and will be considered to be equal to unity throughout this work.

As implied by the first of Eq. 2.31, the restoring force is split into two parts. The first part is linear with an effective stiffness equal to the plastic stiffness of the material and a hysteretic one with *z* being the restoring force that bares the memory of the nonlinear system.

The evolutionary equation of the hysteretic parameter is in fact the evolution of the hysteretic restoring forces. However, the stiffness of the hysteretic restoring forces can be considered as invariant and can be excluded from that form. Therefore, the hysteretic parameter *z* can be considered to be the hysteretic displacements of the system and thus:

Eq. 2.32

$$\begin{cases} \ddot{u} + c\dot{u} + P_{res}^{BW} = p(t) \\ P_{res}^{BW} = \alpha K u + (1-\alpha)Kz \\ \dot{z} = \left[1 - \left|\dfrac{z}{z_y}\right|^n \left(\beta + \gamma \cdot sign(z\dot{u})\right)\right]\dot{u} \end{cases}$$

where z_y is the maximum value of the hysteretic parameter.

The formulations presented in Eq. 2.31 and Eq. 2.32 are based on mechanical insight rather than the mathematical theory of hysteresis. Thus, the derivation of Eq. 2.31 from the mathematical background established in section 2.2 is not straightforward. However, a mechanical representation of the model can be established that allows for the decomposition of Eq. 2.31.

The simplest decomposition that can be represented mechanically is the one of elastic-fully plastic system. It is illustrated in Figure 2.4 and essentially consists of a friction slider and an elastic spring. As the displacement u is such low that the spring is tensioned under a force that is lower than the friction force, the system is elastic; as the displacement overrides the crucial displacement that makes the spring to yield the friction force, the system transforms to a mechanism and the tension force stops to increase (this is the horizontal branch illustrated in figure). If we decide to stop trailing the system, a displacement x is tracked by the slider while being trailed. If further the system will start moving in the opposite direction, the spring should be firstly unloaded until the spring start to take load of the opposite direction. When it is fully unloaded, one will realize that the system is no more in the same position as it was before (as it was trailed as mechanism); this difference between the initial position and the new trailed position of the system in rest state, is called persistent displacement or plastic displacement. After the rest state the system can undergo an opposite direction load until the load is reached again the friction force. Such systems rendering the described behaviour, can be represented also simply in the stress space of the yield locus. This locus is in 1-D space and consists of a border point, which is set at a distance from the zero stress equal to the magnitude of the friction force (we use the force variable in this space to be compatible with the described system).

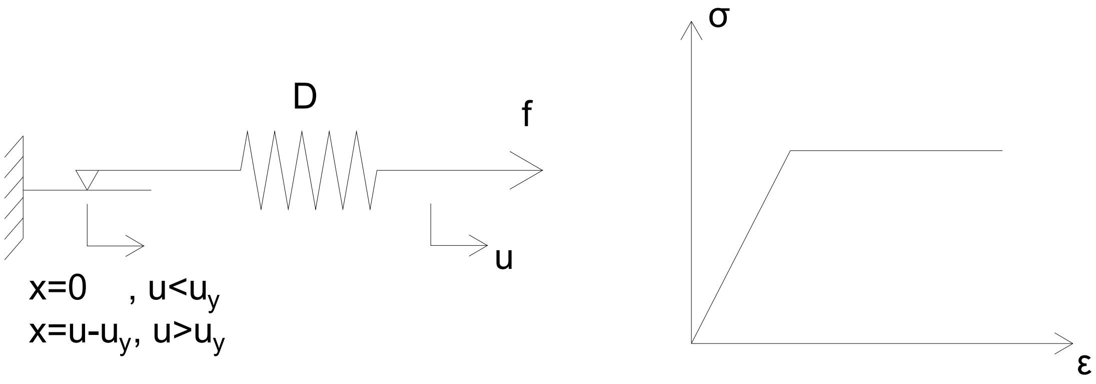
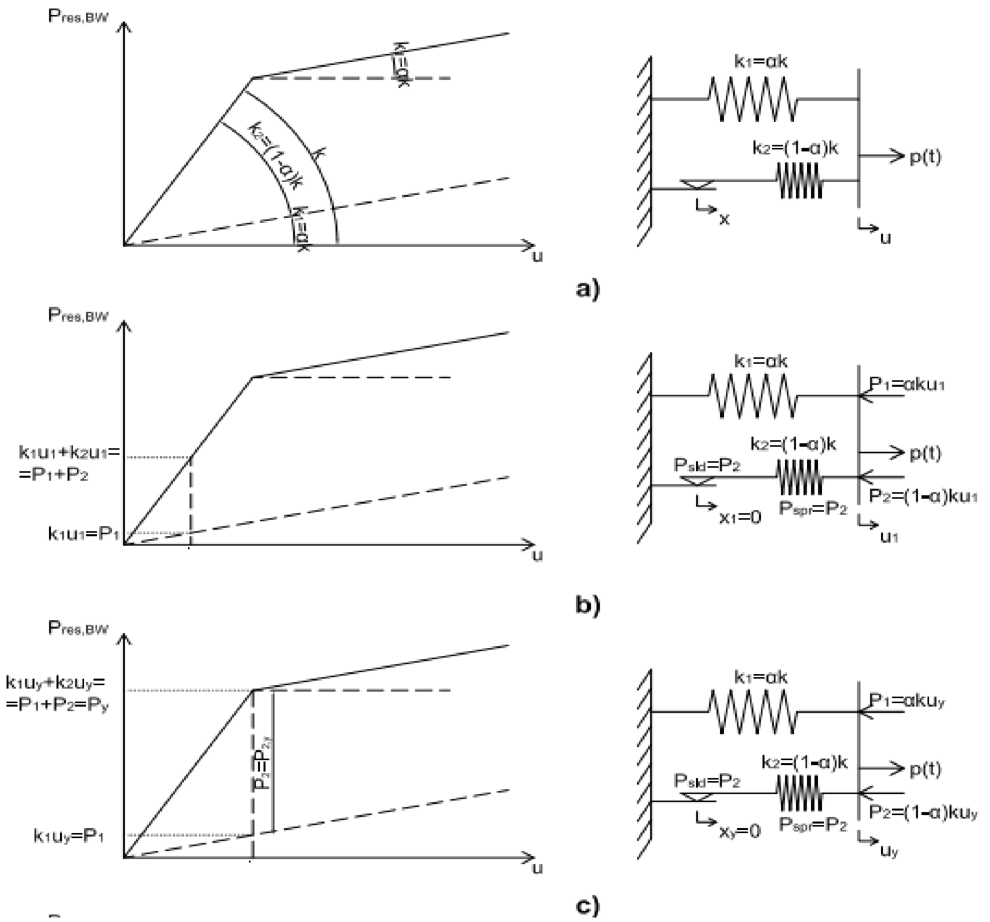


Figure 2.4 Mechanical representation of the elastic-fully plastic system

The described decomposition is a special case of the Bouc-Wen model; that is for α=0. The general model of the Bouc-Wen model is subsequently described with an extra component in parallel connection.

This decomposition is schematically represented in Figure 2.5 considering damping coefficient $c = 0$ for the sake of presentation; the model can be visualized as a parallel combination of a linear spring (Spring #1) and a nonlinear element, as the one described before (elastic-perfectly plastic element). The nonlinear element consists of a linear spring (Spring #2) and a friction slider connected in series. Thus, a two degree of freedom system is introduced, ***u, being the total displacement*** and ***z, being the relative displacement of Spring 2***. From compatibility considerations, the friction sliding displacement, if any, is determined by the difference $(x = u - z)$.



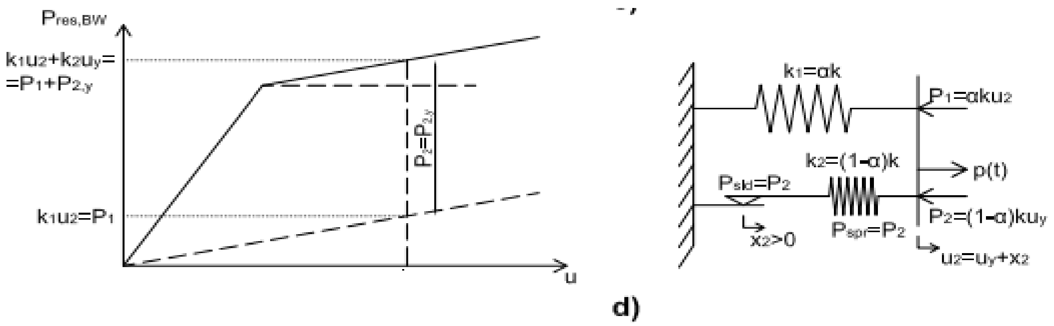


Figure 2.5 Bouc-Wen model statement by decomposed components a) System description b) Response to an elastic displacement c) Response to a yielding displacement d) Response to a post yielding displacement

As long as the force acting on the friction slider is smaller than a threshold (F_{sl}), sliding does not occur, thus $x = 0$ and the relative displacement on Spring #2 is equal to the total imposed displacement. In such a case, the system behaves elastically with combined stiffness k, since springs #1 and #2 are given an elastic stiffness of ak and $(1 - a)k$ respectively, α being the inelastic to elastic stiffness ratio.

When the friction slider threshold is overcome, sliding occurs and the relative displacement in spring #2 remains constant, denoted herein as z_y. All these phases are summarized in the following force-displacement relationship:

Eq. 2.33

$$P_{res}^{BW} = P_1 + P_2 = \alpha ku + (1 - a)kz$$

where *z* is:

Eq. 2.34

$$z = \begin{cases} u, & x \leq u_y \\ z_y, & x > u_y \end{cases}$$

As in engineering applications, the internal variable x is neither easy to measure, nor derive theoretically, the total displacement at which sliding occurs is used instead. This can be easily derived (from a uniaxial tension experiment or implementing a specific yield criterion) and thus Eq. 2.34 is treated equivalently as:

Eq. 2.35

$$z = \begin{cases} u, & u \leq u_y \\ u_y, & u > u_y \end{cases}$$

Wen (1980) proposed the following relation in order to smooth the transition from the elastic (no sliding) to the inelastic response (sliding) of the system:

Eq. 2.36

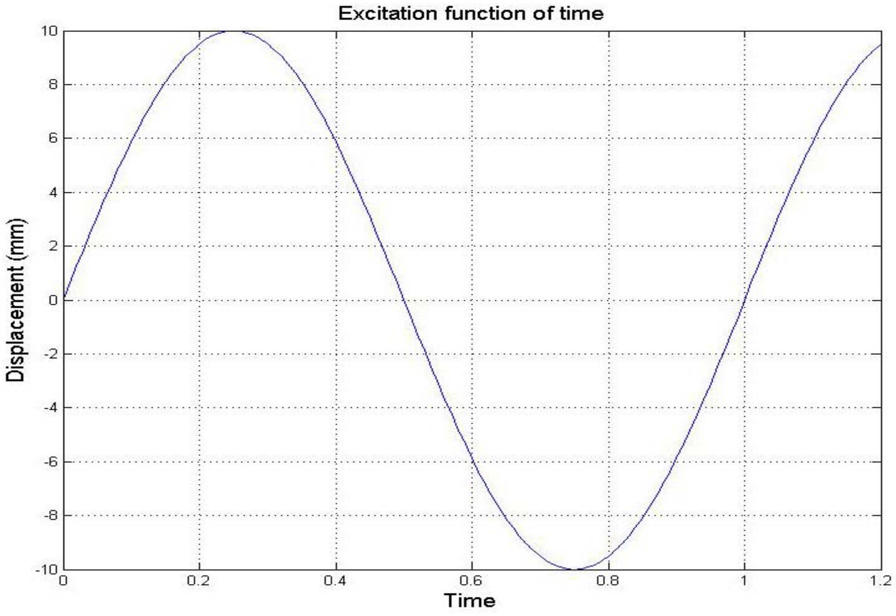
$$\dot{z}(t) = f\big(\dot{u}(t),\ z(t)\big) = \big[A - h_1\big(z(t)\big) \cdot h_2\big(\dot{u}(t), z(t)\big)\big] \cdot \dot{u}(t)$$

where:

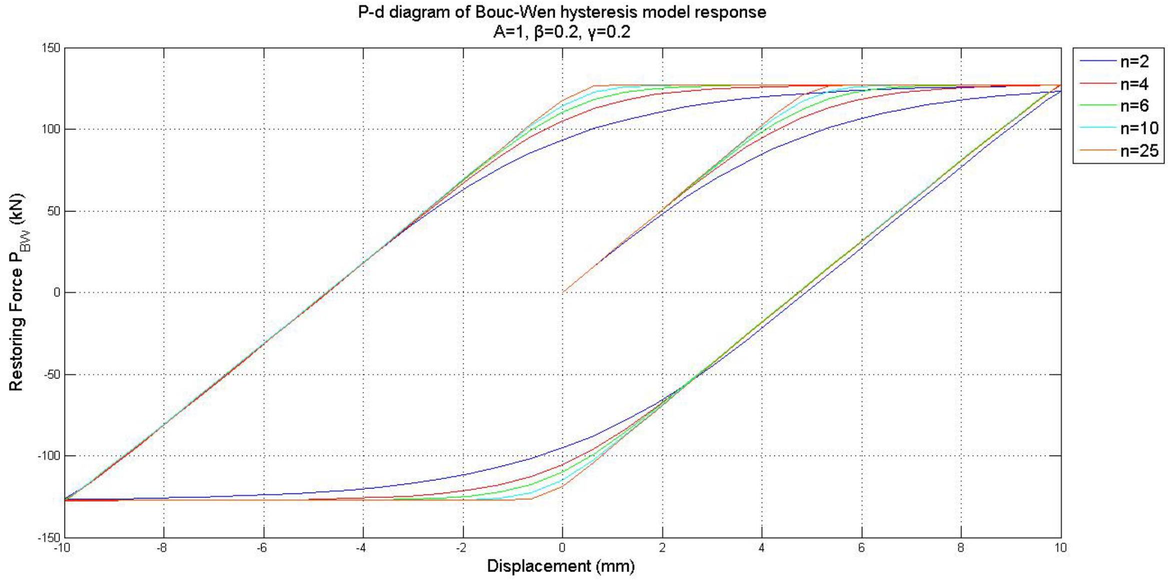
Eq. 2.37

$$h_1\big(z(t)\big) = \left|\frac{z(t)}{z_y}\right|^n, \quad h_2\big(\dot{u}(t), z(t)\big) = \beta + \gamma \cdot sign\big(z(t) \cdot \dot{u}(t)\big)$$

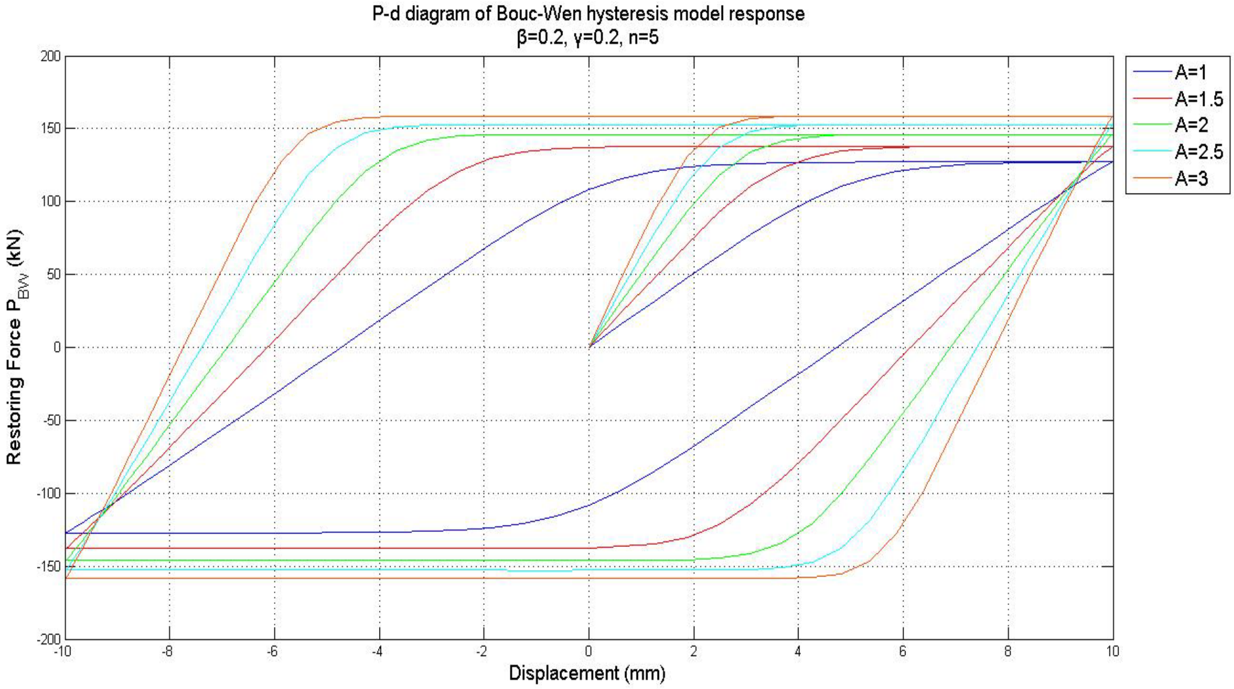
h_1 can be regarded as a uniaxial flow rule and h_2 as the corresponding cyclic loading rate, while in the above relation, $dot(.)$ denotes differentiation with respect to time (not necessarily considering viscous and dynamic phenomena; in the most simple case it just considers steps to catch the frames of sequential states). Parameter *n* controls the smoothness of the transition from the elastic to the inelastic regime, while the terms *β* and *γ* introduced in Eq. 2.37 are shape factors that affect the shape of the hysteresis loop (Sivaselvan & Reinhorn, 2000). In Figure 2.6 the results from a strain controlled numerical experiment on a D18 rebar are presented for different values of the model parameters *n*, *β* and *γ*. Material parameters are S500 and E=200 GPa, while the length of the bar is considered to be 2m.



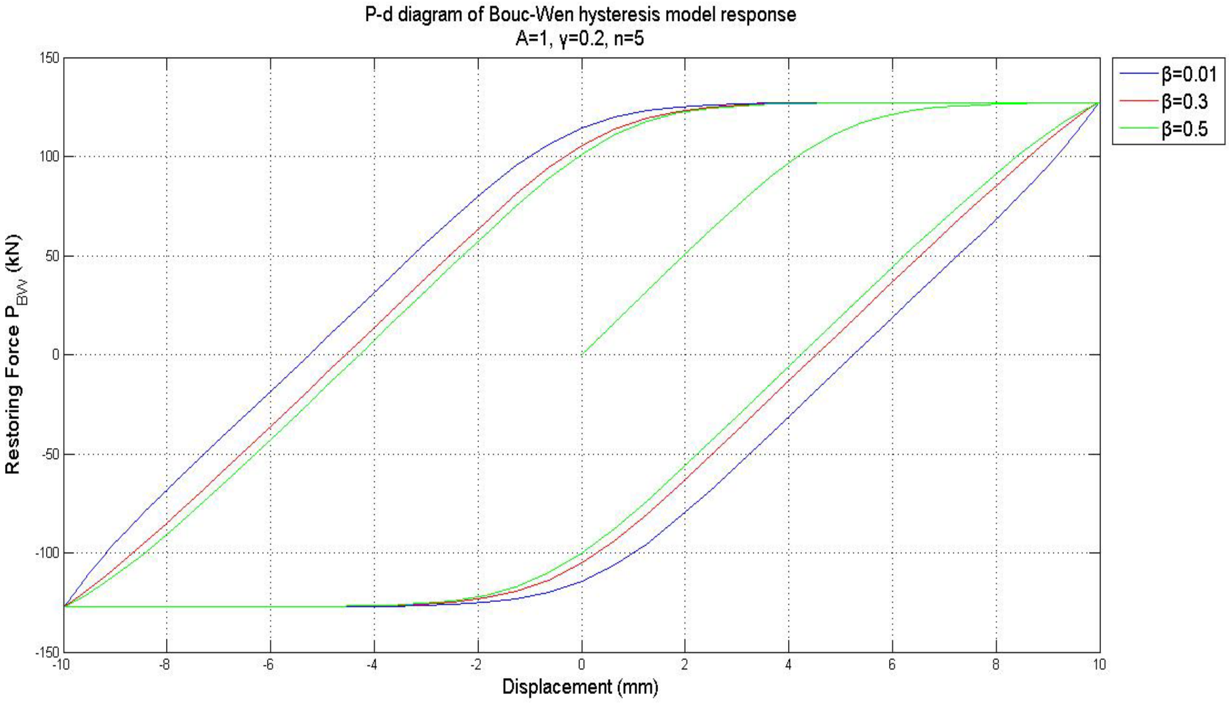
a)


P-d diagram of Bouc-Wen hysteresis model response
A=1, β=0.2, γ=0.2
Restoring Force P_{BW} (kN)
Displacement (mm)
150
100
50
0
-50
-100
-150
-10
-8
-6
-4
-2
0
2
4
6
8
10
n=2
n=4
n=6
n=10
n=25

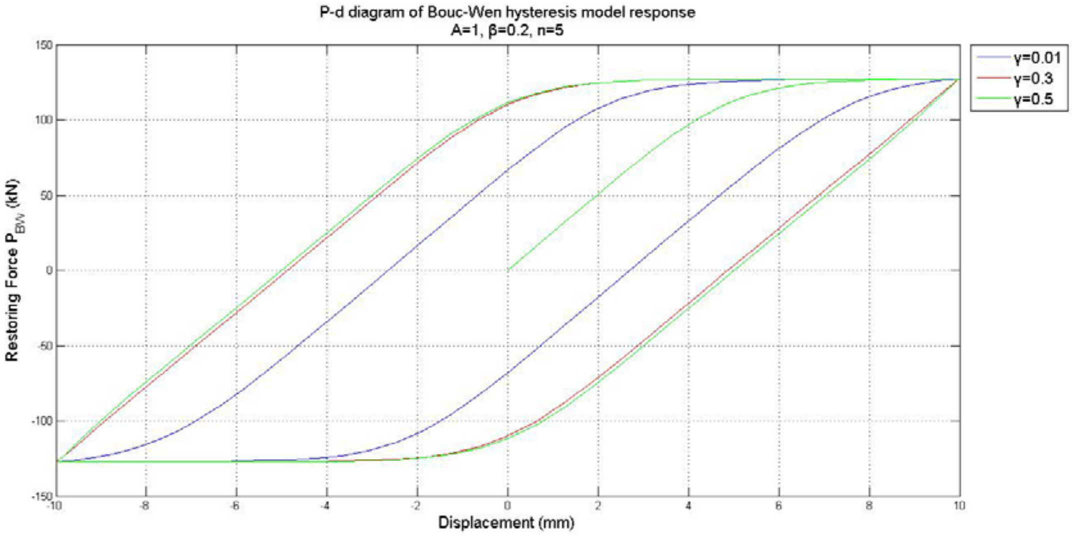
b)


P-d diagram of Bouc-Wen hysteresis model response
β=0.2, γ=0.2, n=5
Restoring Force P_{BW} (kN)
Displacement (mm)
200
150
100
50
0
-50
-100
-150
-200
-10
-8
-6
-4
-2
0
2
4
6
8
10
A=1
A=1.5
A=2
A=2.5
A=3

c)



d)



e)

Figure 2.6 Strain controlled numerical experiment Variation in hysteretic loop with respect to (b) n (c) A (d) β (γ=0.2, n=5) (e) γ (b=0.2, n=5)

Up to this point, the presentation of the Bouc-Wen model is based on the grounds of force-displacement relations. Though versatile, this formulation limits the applicability of these relations where the Finite Element Method is concerned. In the next paragraphs, a general formulation is presented, within the framework of classical plasticity, that allows for the ***implementation of these smooth-hysteretic operators, thus avoiding the need for piece-wise linear hysteretic models.***

2.4 THE GENERALIZED BOUC-WEN MODEL IN THE COMPLETE STRESS SPACE

Though the derivation of the Bouc-Wen model presented in section 2.2 was based on mathematical grounds, it can be proven that ***the same relations can be deducted considering the physics of classical plasticity***. ***The advantage of this approach is the fact that smoothed plasticity relations are deducted in terms of tensorial stress-strain relations***. ***This leads to a versatile material model both from computational and experimental perspective***. Classical plasticity is based on a set of governing equations, namely the flow rule, the yield condition, the consistency condition and the hardening rule. In the work presented herein, ***the case of associative plasticity is addressed where the plastic potential coincides with the flow rule***. Denoting the flow rule as Φ, the rate of plastic deformation is defined as:

Eq. 2.38

$$\{\dot{\varepsilon}^p\} = \dot{\lambda}\frac{\partial \Phi}{\partial\{\sigma\}}$$

where $\{\dot{\varepsilon}^p\}$ is the plastic strain tensor, $\dot{\lambda}$ the plastic multiplier, $\{\sigma\}$ the stress tensor and $(\dot{\ })$ denotes differentiation with respect to time. The plastic multiplier and the yield function are found to comply with the Kuhn-Tucker optimality conditions:

Eq. 2.39

$$\dot{\lambda} \geq 0, \qquad \Phi \leq 0, \qquad \dot{\lambda} \cdot \Phi = 0$$

The consistency condition is an immediate consequence of relation Eq. 2.39 stating that when at yield:

Eq. 2.40

$$\dot{\lambda} \cdot \Phi = 0$$

A typical isotropic yield criterion (or plasticity model for brevity) is the von-Mises yield criterion defined as:

Eq. 2.41

$$\Phi = \|\{\sigma\} - \{\eta\}\| - \sigma_M \leq 0$$

where $\{\sigma\}$ is the deviatoric stress tensor and $\{\eta\}$ the deviatoric back-stress tensor. The evolution of the back-stress, determines the type of hardening introduced in the material model during subsequent cycles of loading and unloading. A commonly used type of hardening is the linear kinematic hardening assumption which dictates a constant plastic modulus during plastic loading. This is accomplished by demanding:

Eq. 2.42

$$\{\dot{\eta}\} = C\{\dot{\varepsilon}^{p}\}$$

A key concept of classical plasticity is the additive decomposition of the strain into reversible elastic and irreversible plastic components. Consequently, the additive decomposition of the strain rate is established as:

Eq. 2.43

$$\{\dot{\varepsilon}\} = \{\dot{\varepsilon}^{el}\} + \{\dot{\varepsilon}^{p}\} \rightarrow \{\dot{\varepsilon}^{el}\} = \{\dot{\varepsilon}\} - \{\dot{\varepsilon}^{p}\}$$

Where $\{\dot{\varepsilon}\}$ is the rate of the total deformation tensor, while $\{\dot{\varepsilon}^{el}\}$ is the rate of the elastic part of the total deformation vector. Based on observations, the unloading stiffness of a plastified material is considered equal to the elastic and thus the following relation holds between the total stress tensor and the elastic part of the strain rate:

Eq. 2.44

$$\{\dot{\sigma}\} = [D]\{\dot{\varepsilon}\}$$

where $[D]$ is the elastic constitutive matrix. Substituting equation Eq. 2.38 into relation Eq. 2.43 and using relation Eq. 2.44 the following equation is derived:

Eq. 2.45

$$\{\dot{\sigma}\} = [D]\left(\{\dot{\varepsilon}\} - \dot{\lambda}\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}\right)$$

By means of the consistency condition (equation Eq. 2.40) and relation Eq. 2.45 the value of the plastic multiplier $\dot{\lambda}$ is evaluated as:

Eq. 2.46

$$\dot{\lambda}\dot{\Phi} = 0 \rightarrow \dot{\lambda}\left(\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^{T}\{\dot{\sigma}\} + \left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^{T}\{\dot{\eta}\}\right) = 0$$

When at yield, Φ=0 and $\dot{\lambda} > 0$ the relation Eq. 2.46 can be written as:

Eq. 2.47

$$\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^{T}\{\dot{\sigma}\} + \left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^{T}\{\dot{\eta}\} = 0 \rightarrow \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^{T}\{\dot{\sigma}\} = -\left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^{T}\{\dot{\eta}\}$$

Premultiplying relation Eq. 2.45 with $\frac{\partial \Phi(\{\sigma\})}{\partial\{\sigma\}}$ the following equation is derived:

Eq. 2.48

$$\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^{T}\{\dot{\sigma}\} = \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^{T}[D]\left(\{\dot{\varepsilon}\} - \dot{\lambda}\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}\right)$$

Substituting equation Eq. 2.47 into equation Eq. 2.48 the following relation is established:

Eq. 2.49

$$-\left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^T \{\dot{\eta}\} = \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^T [D]\left(\{\dot{\varepsilon}\} - \dot{\lambda}\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}\right)$$

In classical plasticity the hardening law is defined as a relation between the back-stress tensor and the plastic strain tensor. This relation can be either rate dependent or rate independent. In any case, the back-stress is finally derived as a function of the plastic multiplier $\dot{\lambda}$ and one can write:

Eq. 2.50

$$\{\dot{\eta}\} = \dot{\lambda} G(\{\eta\}, \Phi)$$

Substituting relation Eq. 2.50 into equation Eq. 2.49 the following relation is derived:

Eq. 2.51

$$-\left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^T \dot{\lambda} G(\{\eta\}, \Phi) = \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^T [D]\left(\{\dot{\varepsilon}\} - \dot{\lambda}\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}\right)$$

Rearranging and solving for the plastic multiplier the following expression is derived:

Eq. 2.52

$$\dot{\lambda} = \left(-\left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^T G(\{\eta\}, \Phi) + \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^T [D]\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}\right)^{-1} \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^T [D]\{\dot{\varepsilon}\}$$

In the case of the elastic perfectly plastic material G=0, and relation Eq. 2.52 coincides with the one proposed by Casciati, 2006. Equations Eq. 2.47 to Eq. 2.52 hold when yielding has occurred, either in the positive or in the negative semi-plane and thus by introducing the following Heaviside functions:

Eq. 2.53

$$H_1(\Phi) = \begin{cases} 1, & \Phi = 0 \\ 0, & \Phi < 0 \end{cases} \qquad H_2(\dot{\Phi}) = \begin{cases} 1, & \dot{\Phi} > 0 \\ -1, & \dot{\Phi} < 0 \end{cases}$$

a single relation is established for the plastic multiplier, in the whole domain of the strain tensor:

Eq. 2.54

$$\dot{\lambda} = H_1(\Phi) H_2(\dot{\Phi}) \left(-\left\{\frac{\partial \Phi}{\partial\{\eta\}}\right\}^T G(\{\eta\}, \Phi) + \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^T [D]\left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}\right)^{-1} \left\{\frac{\partial \Phi}{\partial\{\sigma\}}\right\}^T [D]\{\dot{\varepsilon}\}$$

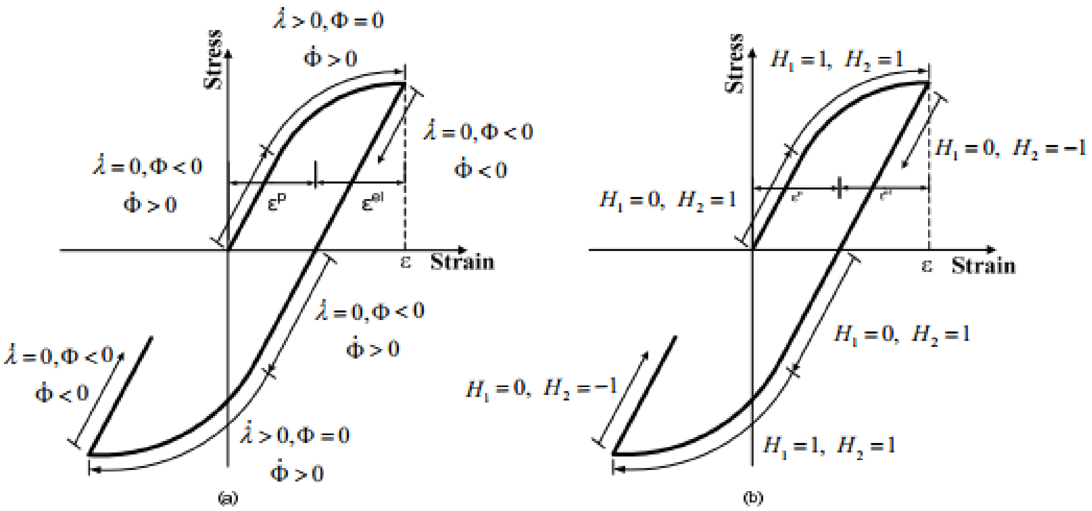


Figure 2.7

Instead of describing the cyclic behaviour of a material in a step-wise approach considering the domains of the Kuhn-Tucker conditions (Figure 2.7(a)) or of the correspondent Heaviside functions (Figure 2.7(b)), Casciati, proposed the smoothening of the latter, introducing additional material parameters.

According to this approach, the two Heaviside functions are smoothened using the following expressions:

Eq. 2.55

$$H_1(\Phi) = \left|\frac{\Phi}{\Phi_0}\right|^N, \qquad N \geq 2$$

and by the definition of the Heaviside function

Eq. 2.56

$$H_2(\dot{\Phi}) = \left(\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}^T \{\dot{\sigma}\}\right) = 0.5 + 0.5sign\left(\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}^T \{\dot{\sigma}\}\right) = \beta + \gamma sign(\{\varepsilon\}^T\{\dot{\sigma}\})$$

where N, band g are model parameters and F0is the maximum value of the yield function or yield point. In the special case where β=γ=0.5, the unloading stiffness is equal to the elastic one. The model proposed by Baber-Noori is thermodynamically admissible as long as relation $-\beta \leq \gamma \leq \beta$ is satisfied. An immediate consequence of equation Eq. 2.55 is that the material is allowed to yield even before the theoretical yield point is reached (Φ_0). Rearranging equation Eq. 2.45 and substituting the definition of the plastic multiplier, the following Bouc-Wen model is derived:

Eq. 2.57

$$\{\dot{\sigma}\} = [D]\left([I] - \left|\frac{\Phi}{\Phi_0}\right|^N \left(\beta + \gamma sign(\{\varepsilon\}^T\{\dot{\sigma}\})\right)[R(\Phi)]\right)\{\dot{\varepsilon}\}$$
$$= [D]\big([I] - H_1(\Phi)H_2(\dot{\Phi})[R(\Phi)]\big)\{\dot{\varepsilon}\}$$

where the matrix [R] is evaluated as:

Eq. 2.58

$$[R(\Phi)] = \left(-\left\{\frac{\partial\Phi}{\partial\{\eta\}}\right\}^T G(\{\eta\},\Phi) + \left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}^T [D]\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}\right)^{-1} \left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}^T [D]$$

and defines the interaction relation between the components of the stress tensor at yield. Thus, the step-wise plasticity equations of relation Eq. 2.39 are replaced by a continuous stress-strain relation. In the uniaxial case, the von Mises yield criterion is reduced to the following form:

Eq. 2.59

$$\Phi_{VM} = \frac{(\sigma_{11} - \eta_{11})^2}{(\sigma_y)^2} - 1$$

And the interaction relation

Eq. 2.60

$$[R] = \frac{E}{c + E}$$

and accordingly, relation Eq. 2.57 becomes:

Eq. 2.61

$$\dot{\sigma}_{11} = E\left(1 - \left|\frac{(\sigma_{11} - \eta_{11})^2}{(\sigma_y)^2}\right|^N \left(\beta + \gamma sign(\varepsilon_{11}\dot{\sigma}_{11})\right)\frac{E}{c + E}\right)\dot{\varepsilon}_{11}$$

The similarities between equation Eq. 2.61 and Bouc's derivation of the hysteretic parameter z in equation Eq. 2.32 are evident.

2.5 REFORMULATION TO THE PARALLEL MODEL OF HYSTERESIS IN THE COMPLETE STRESS SPACE

The generalized parallel model of Bouc-Wen introduced by Karray and Bouc (Wen, 1980, Casciati, 2006) is a subcase of the formulation presented in the previous Section. Generalizing the parallel spring concept introduced in Figure 2.5(a), the stress tensor is decomposed into an elastic and hysteretic part as follows:

Eq. 2.62

$$\{\sigma\} = [\alpha]\{\sigma^e\} + ([I] - [\alpha])\{\sigma^h\}$$

where $[\alpha]$ denotes square diagonal matrix with post yield to elastic stiffness ratios, which for an isotropic material is considered constant in every direction, $[I]$ is the identity matrix, while the elastic part $\{\sigma^e\} = [{\sigma^e}_{11} \quad {\sigma^e}_{22} \quad {\sigma^e}_{12}]^T$ is expressed by the following relation:

Eq. 2.63

$$\{\sigma^e\} = [D]\{\varepsilon\}$$

$[D]$ is the elastic constitutive matrix. The hysteretic part $\{\sigma^h\} = [{\sigma^h}_{11} \quad {\sigma^h}_{22} \quad {\sigma^h}_{12}]^T$ evolves according to the following Bouc-Wen hysteretic rule (Sivaselvan and Reinhorn, 2003):

Eq. 2.64

$$\begin{bmatrix} \dot{\sigma}_{11}^h \\ \dot{\sigma}_{22}^h \\ \dot{\sigma}_{12}^h \end{bmatrix} = [D]\big([I] - H_1(\Phi)H_2\big(\dot{\Phi}\big)[\tilde{R}(\Phi)]\big)\begin{bmatrix} \dot{\varepsilon}_{11} \\ \dot{\varepsilon}_{22} \\ \dot{\varepsilon}_{12} \end{bmatrix}$$

where $H_1(\Phi)\ and\ H_2\big(\dot{\Phi}\big)$ are smoothed Heaviside functions defined in equations Eq. 2.55 and Eq. 2.56 respectively while $[\tilde{R}(\Phi)]$ is the interaction matrix defined in equation Eq. 2.58 setting $G(\{\eta\}, \Phi) \equiv 0$.

However,* equations Eq. 2.62 to Eq. 2.64 *are capable of simulating hysteretic systems with linear kinematic hardening. Clearly this limits the applicability of the model.

Writing equation Eq. 2.62 in rate form and substituting relation Eq. 2.64 the following equation is derived:

Eq. 2.65

$$\{\dot{\sigma}\} = [\alpha][D]\{\dot{\varepsilon}\} + ([I] - [\alpha])[D]\big([I] - H_1(\Phi)H_2\big(\dot{\Phi}\big)[\tilde{R}(\Phi)]\big)\{\dot{\varepsilon}\}$$

Matrix $[\alpha]$ diagonal, thus relation Eq. 2.65 can be cast on the following form:

Eq. 2.66

$$\{\dot{\sigma}\} = [D]^T\left([\alpha]^T + ([I] - [\alpha])^T\big([I] - H_1(\Phi)H_2\big(\dot{\Phi}\big)[\tilde{R}(\Phi)]\big)\right)\{\dot{\varepsilon}\}$$

Where,

Eq. 2.67

$$[\tilde{R}(\Phi)] = \left(\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}^T [D]\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}\right)^{-1}\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}\left\{\frac{\partial\Phi}{\partial\{\sigma\}}\right\}^T [D]$$

2.6 THE VON MISES YIELD SURFACE

The von Mises yield criterion suggests that the yielding of materials begins when the second deviatoric stress invariant J_2reaches a critical value. For this reason, it is sometimes called the J_2-plasticity or J_2 flow theory. Because the von Mises yield criterion is independent of the first stress invariant, J_1, it is applicable for the analysis of plastic deformation for ductile materials such as metals, as the onset of yield for these materials does not depend on the hydrostatic component of the stress tensor. Prior to yield, material response is assumed to be elastic.

In materials science and engineering the von Mises yield criterion can be also formulated in terms of the von Mises stress or equivalent tensile stress, σ_y, a scalar stress value that can be computed from the stress tensor. In this case, a material is said to start yielding when its von Mises stress reaches a critical value known as the yield strength, σ_y. The von Mises stress is used to predict yielding of materials under any loading condition from results of simple uniaxial tensile tests. The von Mises stress satisfies the property that two stress states with equal distortion energy have equal von Mises stress.

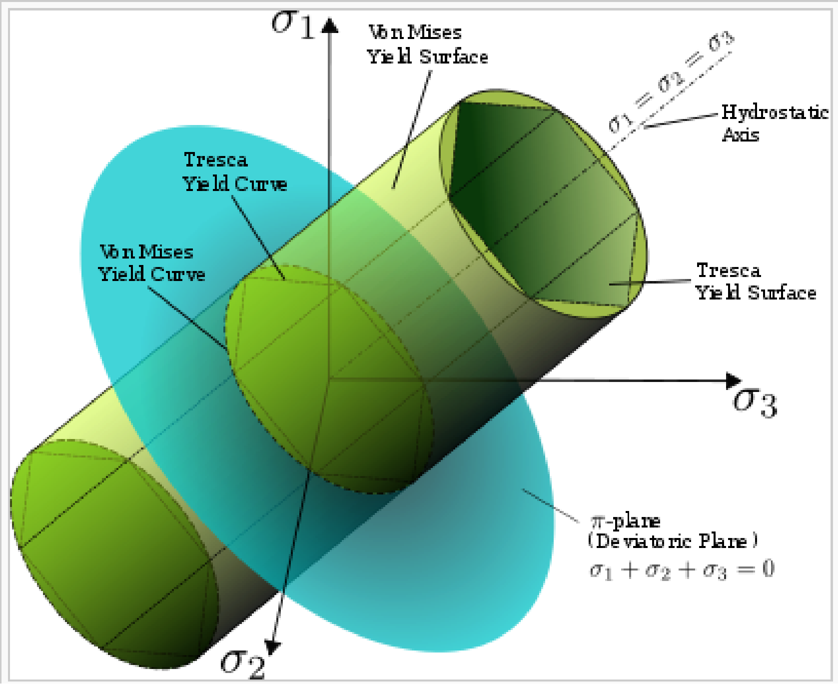


Figure 2.8 The von Mises yield surfaces in principal stress coordinates circumscribes a cylinder with radius $\sqrt{2/3}\,\sigma_y$ around the hydrostatic axis. Also is shown Tresca's hexagonal yield surface

The yield surface is given by the function of a cylinder locus in principle stress space with radius κ and a relation between the stresses that is given by the second invariant of the stress deviator

Eq. 2.68

$$\Phi = \frac{J_2}{\kappa^2}$$

The second invariant of stress deviator

Eq. 2.69

$$J_2 = \frac{1}{6}((\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2) + {\sigma_{12}}^2 + {\sigma_{23}}^2 + {\sigma_{13}}^2$$

and κ is the yield stress of the material in pure shear. The magnitude of the shear yield stress in pure shear is $\sqrt{3}$ times lower than the tensile yield stress σ_y in the case of simple tension. Thus, we have

$$\kappa = \frac{\sigma_y}{\sqrt{3}}$$

Then,

Eq. 2.70

$$\begin{aligned}\Phi &= \frac{\frac{1}{6}((\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2) + {\sigma_{12}}^2 + {\sigma_{23}}^2 + {\sigma_{13}}^2}{\left(\frac{s_y}{\sqrt{3}}\right)^2} \\ &= \frac{\frac{1}{6}((\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2) + {\sigma_{12}}^2 + {\sigma_{23}}^2 + {\sigma_{13}}^2}{\left(\frac{s_y}{\sqrt{3}}\right)^2} \\ &= \frac{(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 + 6({\sigma_{12}}^2 + {\sigma_{23}}^2 + {\sigma_{13}}^2)}{2 \cdot {s_y}^2}\end{aligned}$$

The yield gradient with respect to the von-Mises yield surface is:

Eq. 2.71

$$\frac{\partial \Phi}{\partial \{\sigma\}} = \begin{bmatrix} \frac{(2\sigma_{11} - \sigma_{22} - \sigma_{33})}{{s_y}^2} \\ \frac{(2\sigma_{22} - \sigma_{11} - \sigma_{33})}{{s_y}^2} \\ \frac{(2\sigma_{33} - \sigma_{11} - \sigma_{22})}{{s_y}^2} \\ 6\frac{\sigma_{12}}{{s_y}^2} \\ 6\frac{\sigma_{23}}{{s_y}^2} \\ 6\frac{\sigma_{13}}{{s_y}^2} \end{bmatrix}$$

For the case of the parallel model the processed stresses are hysteretic, that is only the second component of the total stress decomposition; thus the yielding stress must adapt to this part by this analogous

Eq. 2.72

$$\sigma_y^h = (1-a)\sigma_y$$

2.7 SUMMARY

In this section, the properties of hysteretic systems are presented and the expression of the Bouc-Wen model is derived accordingly, based on concepts of the mathematical theory of hysteresis. Next, a general form of the Bouc-Wen model is derived in stress-strain form, based on the phenomenological concepts of the classical theory of plasticity. A rate form of the stress tensor is derived that accounts for the full cyclic behavior of the continuum. This rate form is general in the sense that it accounts for every combination of yield criteria and hardening laws whereas existing formulations only describe hysteretic behavior with linear kinematic hardening.

The formulation derived depends on total stress components rather than their deviatoric parts, thus yielding a formulation that is easily incorporated in the Finite Element scheme, as will be presented in subsequent chapters. A general formulation is presented, within the framework of classical plasticity, that allows for the implementation of these smooth-hysteretic operators, thus avoiding the need for piece-wise linear hysteretic models.

3 FORMULATION OF THE BOUC-WEN MODEL IN FEM AND IGA

The generalized Bouc-Wen model expressed in the complete stress space can be easily formulated in a variational form of the stationary principle of energy. Such a variational form is the virtual work principle and is described in part II section 2.1.6.

Eq. 3.1

$$W_{int} = W_{ext} \Rightarrow \int_{V} \{\delta\varepsilon\}^T \{\sigma\}\, dV = \int_{V} \{\delta u\}^T \{f\}^B\, dV + \int_{S_f} \left\{\delta u_{S_f}\right\}^T \{f^{S_f}\}\, dS + \sum_{i} \{\delta d_i\}^T \{R_i^C\}$$

As it is shown this formulation is equivalent with the weak formulation of the weighted residual Bubnov-Galerkin method. It is also shown, by their properties in part II section 2.3.1, that the geometric basis functions can be members of a complete Sobolev $H^1(\Omega)$ solution space S, such that

Eq. 3.2

$$S = \{u | u \in H^1(\Omega), u|_{\Gamma_D} = g\}$$

thus, such a basis can form a subspace $S^h \subset S$ to solve this problem.

3.1 IMPLEMENTING THE PARALLEL BOUC-WEN MODEL

Introducing the stress-strain expression of the parallel case of the Bouc-Wen model into the virtual work principle expressed in rate form, it yields

Eq. 3.3

$$\int_{V} \{\delta\varepsilon\}^T \Big([\alpha][D] + ([I] - [\alpha])[D]\big([I] - H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big)\Big) \{\dot{\varepsilon}\}\, dV$$

$$= \int_{V} \{\delta u\}^T \{\dot{f}\}^B\, dV + \int_{S_f} \left\{\delta u_{S_f}\right\}^T \{\dot{f}^{S_f}\}\, dS + \sum_{i} \{\delta d_i\}^T \{\dot{R}_i^C\}$$

This formulation is geometrically linear (in large displacements notion and large deformations notion too), thus the virtual displacement field will be independent on time. A formulation with geometrical non-linearities (in large displacements notion) should have the virtual displacement field in rate form too. Furthermore, implementing the formulation with large strains the non-linear part of the deformation gradient tensor should be introduced too.

Eq. 3.3 can be decomposed to two parts, the linear one

Eq. 3.4

$$\int_V \{\delta\varepsilon\}^T[\alpha][D]\{\dot{\varepsilon}\}\,dV = \int_V \{\delta u\}^T\{\dot{f}\}^B\,dV + \int_{S_f}\left\{\delta u_{S_f}\right\}^T\{\dot{f}^{S_f}\}\,dS + \sum_i \{\delta d_i\}^T\{\dot{R}_i^C\}$$

And the hysteretic one

Eq. 3.5

$$\int_V \{\delta\varepsilon\}^T[D]([I]-[\alpha])\big([I]-H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big)\{\dot{\varepsilon}\}\,dV$$

$$= \int_V \{\delta u\}^T\{\dot{f}\}^B\,dV + \int_{S_f}\left\{\delta u_{S_f}\right\}^T\{\dot{f}^{S_f}\}\,dS + \sum_i \{\delta d_i\}^T\{\dot{R}_i^C\}$$

In Eq. 3.4 the linear stiffness matrix is easily formed as in a fully linear problem.

Eq. 3.6

$$\int_V \{\delta\varepsilon\}^T[\alpha][D]\{\dot{\varepsilon}\}\,dV = \int_V [B]^T[\alpha][D][B]\,dV\{\dot{d}_i\} = [K^L]\{\dot{u}\}$$

For isogeometric implementation see part II section 3; in addition to that only one matrix more exist, the matrix $[\alpha]$, that is smeared with the elastic constitutive matrix $[D]$.

In Eq. 3.4 there are some entities expressed in rate form; thus this stiffness matrix is a tangent one and a step method should be considered to calculate it in each step.

Eq. 3.7

$$\int_V \{\delta\varepsilon\}^T[D]([I]-[\alpha])\big([I]-H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big)\{\dot{\varepsilon}\}\,dV$$

$$= \int_V [B]^T[D]([I]-[\alpha])\big([I]-H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big)[B]\,dV\{\dot{d}_i\}$$

$$= [K^H]\{\dot{u}\}$$

We observe that this tangent hysteretic matrix is in the same form as the linear one, but the constitutive matrix is refreshed in each step. Thus this tangent hysteretic constitutive matrix is

Eq. 3.8

$$[D^H] = [D]([I]-[\alpha])\big([I] - H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big)$$

Finally considering the composition again it is

Eq. 3.9

$$([K^L]+[K^H])\{\dot{u}\} = \{\dot{F}\} \rightarrow [K^T]\{\dot{u}\} = \{\dot{F}\}$$

3.2 Implementation of the Euler step method

We can rewrite Eq. 3.8 as

Eq. 3.10

$$[D^H] = [D]([I]-[\alpha])\big([I] - H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big) = [D]([I]-[\alpha])G^H$$

Thus, the only part that should be computed in each step is G^H which represents the hysteresis of the system.

Eq. 3.11

$$G^H = \big([I] - H_1(\Phi)H_2(\dot{\Phi})[\tilde{R}(\Phi)]\big)$$

The yield condition term is

Eq. 3.12

$$H_1\big(\Phi(\{\sigma\})\big) = \left|\frac{\Phi(\{\sigma\})}{\Phi_0}\right|^N$$

that is computed considering the stress state at each step

The cyclic condition term is

Eq. 3.13

$$H_2\left(\dot{\Phi}(\{\sigma\},\{\dot{\varepsilon}\})\right) = \beta + \gamma sign(\{\dot{\varepsilon}\}^T\{\sigma\})$$

That is computed considering the stress state and a strain step

And the stress interaction matrix, that is computed considering the stress state,

Eq. 3.14

$$[\tilde{R}(\Phi(\{\sigma\}))] = \left(\left\{\frac{\partial\Phi(\{\sigma\})}{\partial\{\sigma\}}\right\}^T [D] \left\{\frac{\partial\Phi(\{\sigma\})}{\partial\{\sigma\}}\right\}\right)^{-1} \left\{\frac{\partial\Phi(\{\sigma\})}{\partial\{\sigma\}}\right\}\left\{\frac{\partial\Phi(\{\sigma\})}{\partial\{\sigma\}}\right\}^T [D]$$

To initiate the analysis we consider an initial stress state in the system and a total force vector. The total excitation vector is discretized to steps by an arbitrary function and a step length dt. If displacement control analysis is implemented the step displacements are converted to step forces by any method of constraints (explicitly, Lagrange multipliers, penalty method, etc.)

In each step the total system is solved considering a step force vector $\{dF\}$

Eq. 3.15

$$\{du\} = [K^T]^{-1}\{dF\}$$

now $\{d\varepsilon\}$ strain vector can be computed by the kinematic equations; this is the input to Eq. 2.66 to compute $\{d\sigma\}$. In the next step the stress state of the system is refreshed and the same steps are implemented again.

3.3 THE DATA FLOW OF PROGRAMMING THE ISOGEOMETRIC METHOD WITH BOUC-WEN HYSTERESIS MODEL

Input

- Model data in CAGD form
  - Control Points (Pi)
  - Knot Vectros (U, V)
  - Weights for rational basis (w)
  - hpk-refinement
- Analysis Data
  - Number of integration gauss quadrature points in two directions
  - Elastic Modulus, E
  - Poisson ratio, ν
  - Material Density, r (for dynamic analysis)
  - Thickness of the domain, thick
  - Constitutive Matrix
  - Boundary conditions on the control variables
  - Traction forces
  - Body forces
- Bouc Wen Data
  - Loop shape parameter, β
  - Loop shape parameter, γ
  - Smoothness of the transition to the yield branch parameter, n
  - Yield Surface (Type, yield stress s_y)
  - Hardening rule (stiffness ratio α, isotropic hardening κ)

Analysis

- Preliminaries
  - Form of knot vectors containing only the non-zero knot spans, that will constitute the elements in analysis
  - Assignment of boundaries (knots) for each element (meshing the analysis domain)
- Construction of the total force vector
  - Total force vector concerning body forces
  - Total force vector concerning traction forces
  - Total force vector superpositioning the total force vectors of body forces, traction forces and the forces on the control variables
- Discretizing the force vector or the displacement vector, to a number of steps of the non-linear hysteretic analysis
- Construction of the total linear stiffness matrix, which is computed only once in the analysis procedure
  - Looping through the elements
    - Indicating the effective basis functions, control points and weights of the current element
    - Constructing the index vector, which maps the d.o.f. of the current element to the global numbering of d.o.f., used for the total matrices and vectors assemblage
    - Construction of the element's linear stiffness matrix
      - Looping through the integration points
        - Map of the integration point from the parent integration domain $\widetilde{\Omega} : \tilde{\xi} \in [-1, \quad 1]$ to the parametric domain $\widehat{\Omega} : \hat{\xi} \in [\hat{\xi}_0, \quad \hat{\xi}_n]$ (domain where the CAGD basis is defined) and calculation of the mapping jacobian determinant to convert the integration weights (the size of the parent domain) to the parametric domain

Analysis continued...

  - Calculation of the derivatives of the basis functions with respect to the parametric variables $(\hat{\xi}, \hat{\eta})$, $\frac{\partial R_{i,j}(\hat{\xi},\hat{\eta})}{\partial \hat{\xi}}$
  - Formulation of the deformation matrix B1 with respect to the parametric domain
  - Calculation at the mapped integration point the jacobian matrix of the mapping $\widehat{\Omega} \rightarrow \Omega$, $\frac{\partial x}{\partial \hat{\xi}} = \frac{\partial R_{i,j}(\hat{\xi},\hat{\eta})}{\partial \hat{\xi}} P_{i,j}$, and the mapping jacobian determinant to convert the volume integrated in parametric domain to the physical domain
  - Calculation of the inverse jacobian matrix $\Omega \rightarrow \widehat{\Omega}$, $\frac{\partial \hat{\xi}}{\partial x}$
  - Calculation of the deformation matrix B with respect to the physical domain $\mathrm{B} = \mathrm{B1} * \frac{\partial \hat{\xi}}{\partial x} = \frac{\partial R_{i,j}(\hat{\xi},\hat{\eta})}{\partial \hat{\xi}} \frac{\partial \hat{\xi}}{\partial x} = \frac{\partial R_{i,j}(\hat{\xi},\hat{\eta})}{\partial x}$
  - Calculation of the element's linear stiffness matrix at current integration point. If the stiffness ratio α is used in the hysteretic formulation, it is applied in each integration point
- Assemble the element linear stiffness matrix to the total linear stiffness matrix using the d.o.f. mapping vector
- In the same way the body forces, the traction forces (integrating on the boundaries) and the mass matrix are calculated

Analysis continued...

- Non-Linear Hysteretic Analysis by the Euler step method. Calculation of the tangent hysteretic stiffness matrix in each step, considering a step total force vector. The first step is done considering an initial stress condition at the integration points (hysteretic or total stresses, depends on the Bouc wen formulation type).
  - Calculation of the total tangent hysteretic stiffness matrix
    - Looping through the elements
      - Indicating the effective basis functions, control points and weights of the corresponding element (as in linear matrix)
      - Constructing the index vector, which maps the d.o.f. of the current element to the global numbering of d.o.f., used for the total matrices and vectors assemblage (as in linear matrix)
      - Construction of the element's tangent hysteretic stiffness matrix (considering a step force vector)
        - Looping through the integration points (considering a current stress condition)
          - The same procedure is implemented as in the calculation of the elastic stiffness matrix to derive the deformation matrix B or can be stored in a database if no geometric non-linearity are considered
          - Calculation of the step deformations (de) in current integration point de=B*du, where du are the current displacements of the control variables related to the current element
          - Calculation of the (hysteretic or total) stress point in the corresponding yield surface space
          - Calculation of the gradients at the (hysteretic or total) stress point in the yield surface space

Analysis continued...

        - Calculation of the H1, H2 and the interaction matrix R
        - Calculation of the tangent hysteretic stiffness matrix at the integration point using the tangent hysteretic constitutive matrix G and the corresponding deformation matrix B, and the two jacobian determinants of the double mapping from the parent domain (numerical integration domain) to the physical domain
        - Calculation of the step (hysteretic or total) stresses ds=G*de and addition of them to the current stress condition to refresh it for the next analysis step
    - Assemble the element tangent hysteretic stiffness matrix to the total stiffness matrix using the dof mapping vector
- Add the tangent hysteretic stiffness matrix to the linear stiffness matrix at the current step and solve for the step displacements of the control variables
- loop again to derive a new tangent hysteretic stiffness matrix, new step stresses and new step displacements of the control variables.

4 APPLICATIONS ON PLASTICITY PROBLEMS

The problem of the rigidly fixed cantilever beam that is analyzed as an elastic problem in part II, is reconsidered in this section applying the Bouc-Wen hysteresis model of plasticity.

The material properties are

Modulus of elasticity $E = 210GPa$

Poisson ratio $v = 0.3$

Yield stress $s_y = 245MPa$

The parameters of the Bouc-Wen model are set for a smooth elastic-perfectly plastic behavior, the parameters of which are

α=0, β=0.5, γ=0.5, n=2

The implemented excitation is a sinusoidal displacement of the tip as a function of the time; the maximum imposed displacement is $u = 0.05m$.

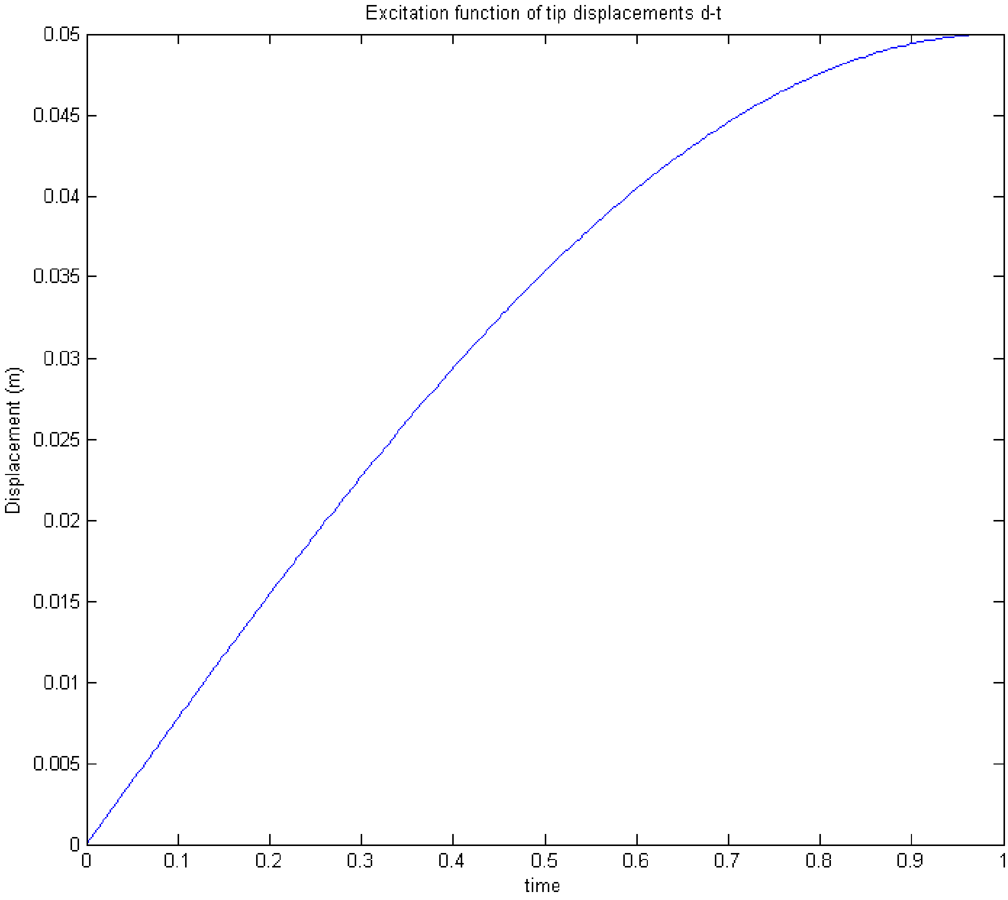


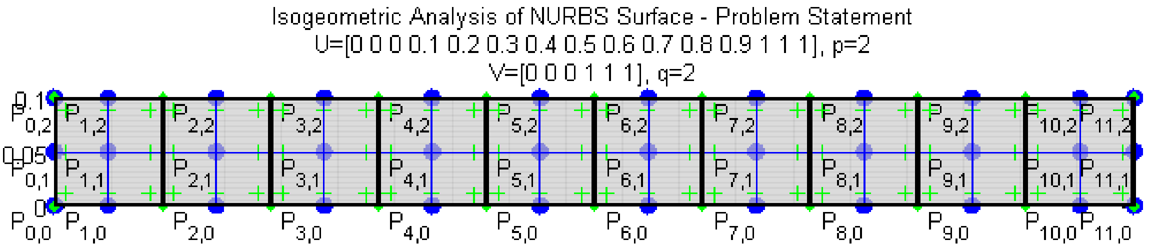
Figure 4.1 Displacement of tip - time diagram

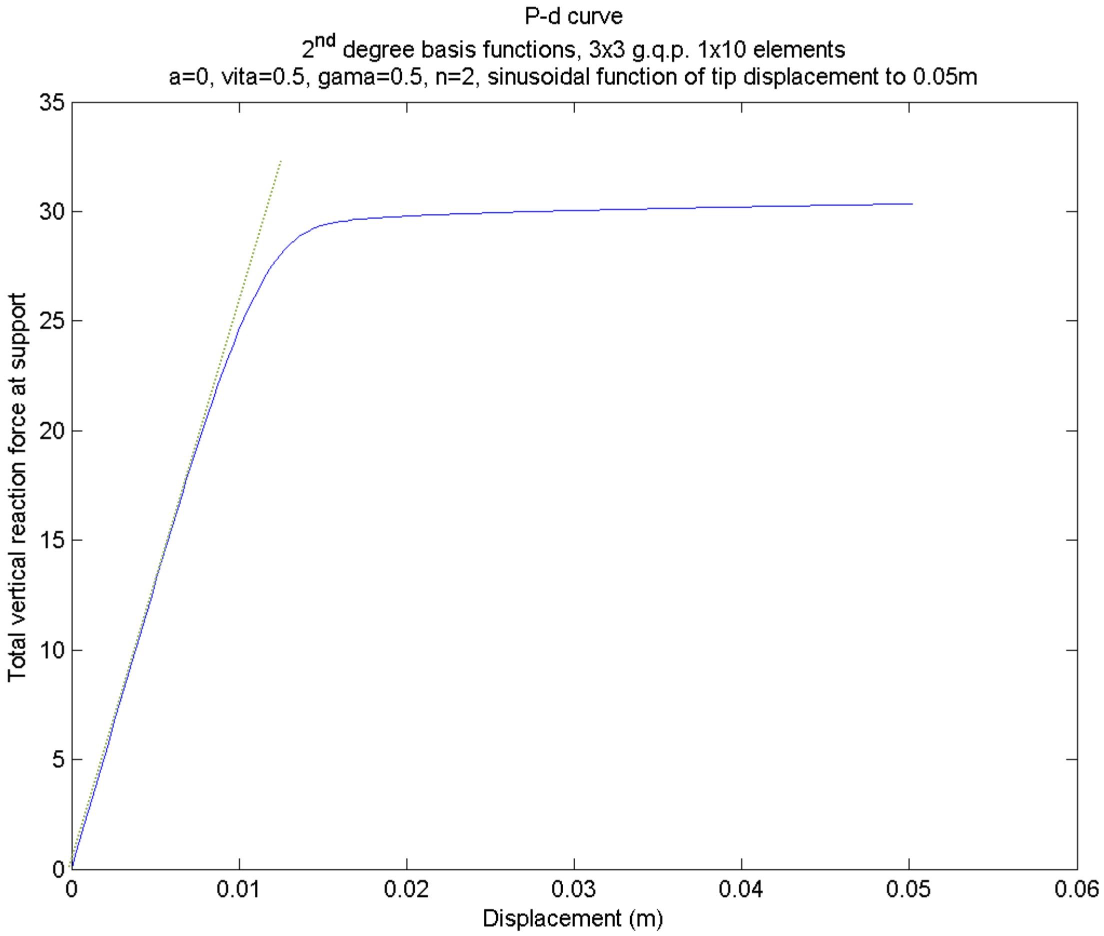
The initial yield load and the ultimate load can be analytically evaluated as:

Eq. 4.1

$$P_y = \frac{\sigma_y b h^2}{6L} = 20kN, \qquad P_u = \frac{\sigma_y b h^2}{4L} = 30kN$$

The geometry that is considered is shown below





We see that both the first yield force and the ultimate force are achieved in a good approximation by that coarse analysis of 10 elements only.

The same fixed beam problem with the same model parameters is analyzed under a cyclic displacement excitation. The function of the excitation is shown below.

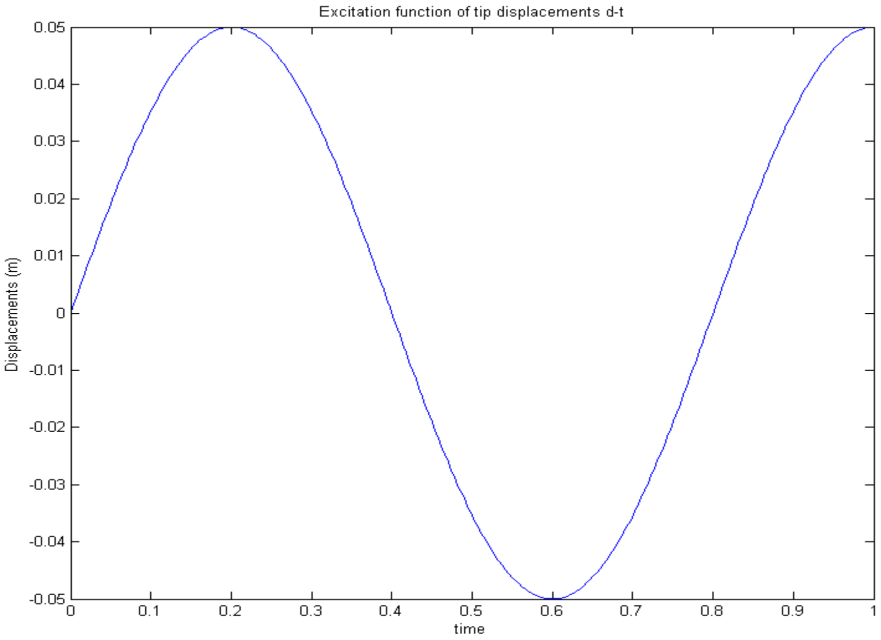


Figure 4.2

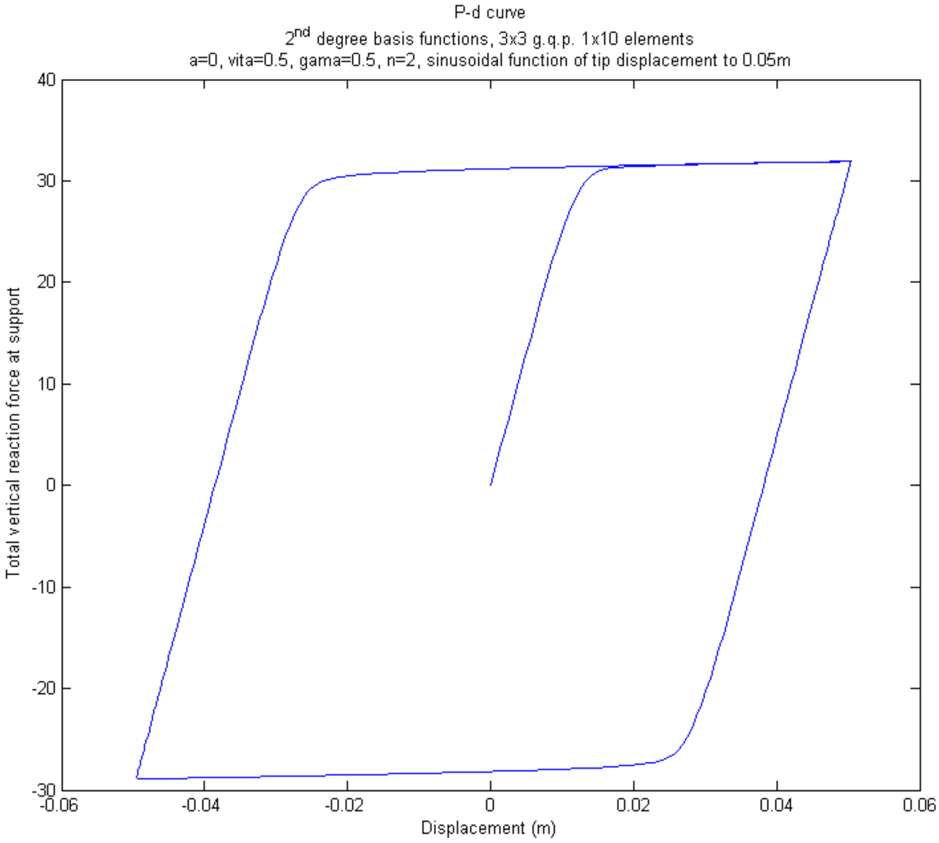


Figure 4.3

The accuracy once again is fair enoupgh and the full cycle response is well achieved.

5 EXAMINATION OF THE METHOD BY PARAMETRIC ANALYSES

The monotonic problem stated above in applications section is analyzed under some parameters. The time step dt, the degree of the basis functions and the maximum size of element.

- The variation in time step that is considered corresponds to:
  250 steps, 500 steps and 1000 steps.
- The variation in degree of the basis functions is:

  1st degree (FEM case), 2nd degree, 3rd degree and 4th degree

- The variation in maximum length size corresponds to the number of elements

  1x10 elements (0.141m), 2x20 elements, 4x40 elements, 8x80 elements

The results are shown below for each variant of time step in 2D plots and the total three-variate parametric analyses in surface plots.

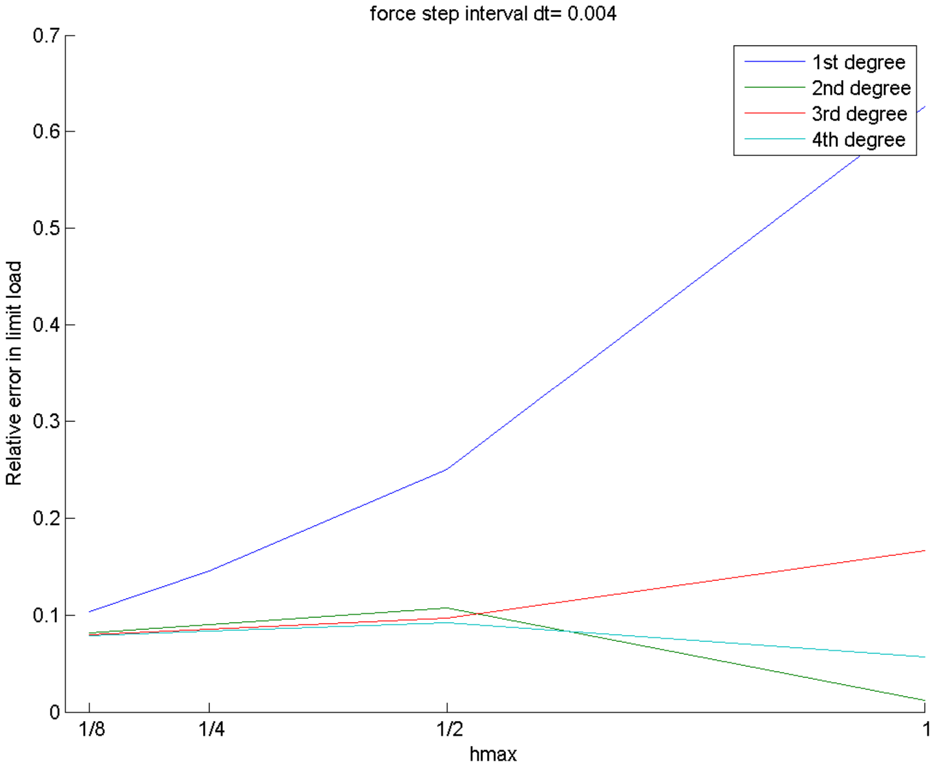


Figure 5.1 *The convergence rate by h-refinement for each degree of basis functions and 250 steps analyses*

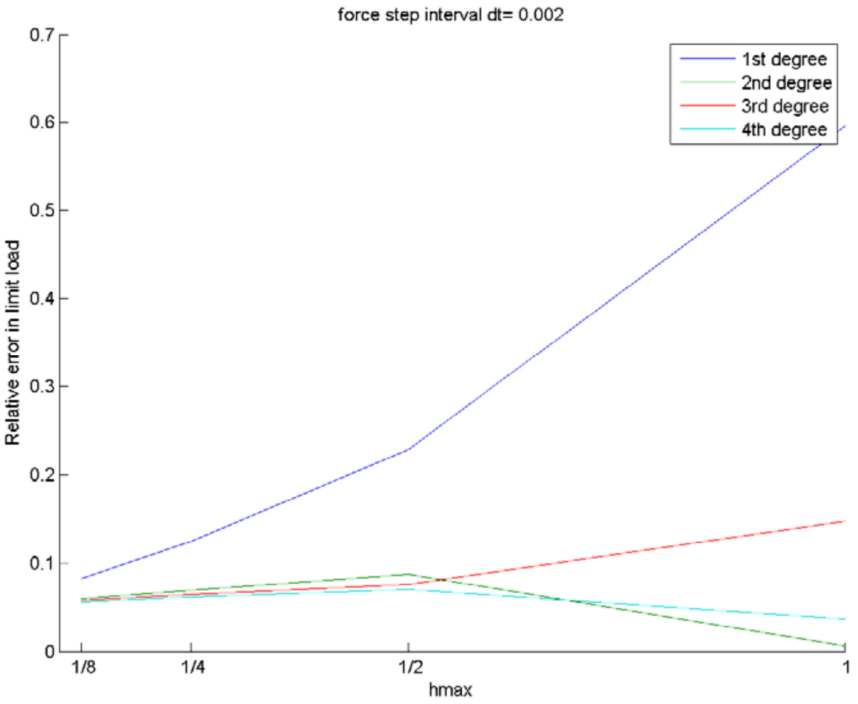


Figure 5.2 The convergence rate by h-refinement for each degree of basis functions and 500 steps analyses

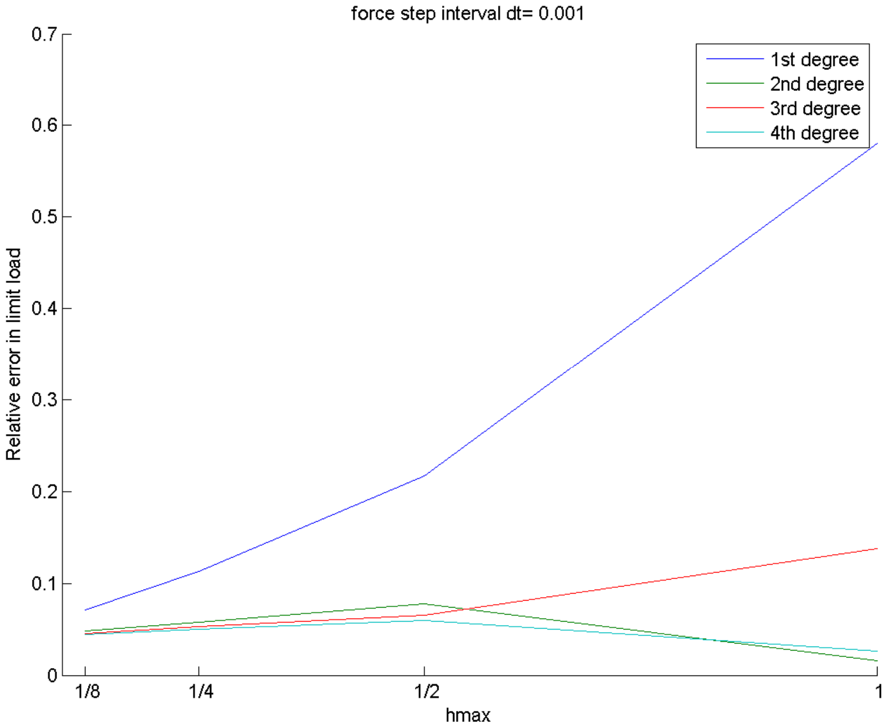


Figure 5.3 The convergence rate by h-refinement for each degree of basis functions and 1000 steps analyses

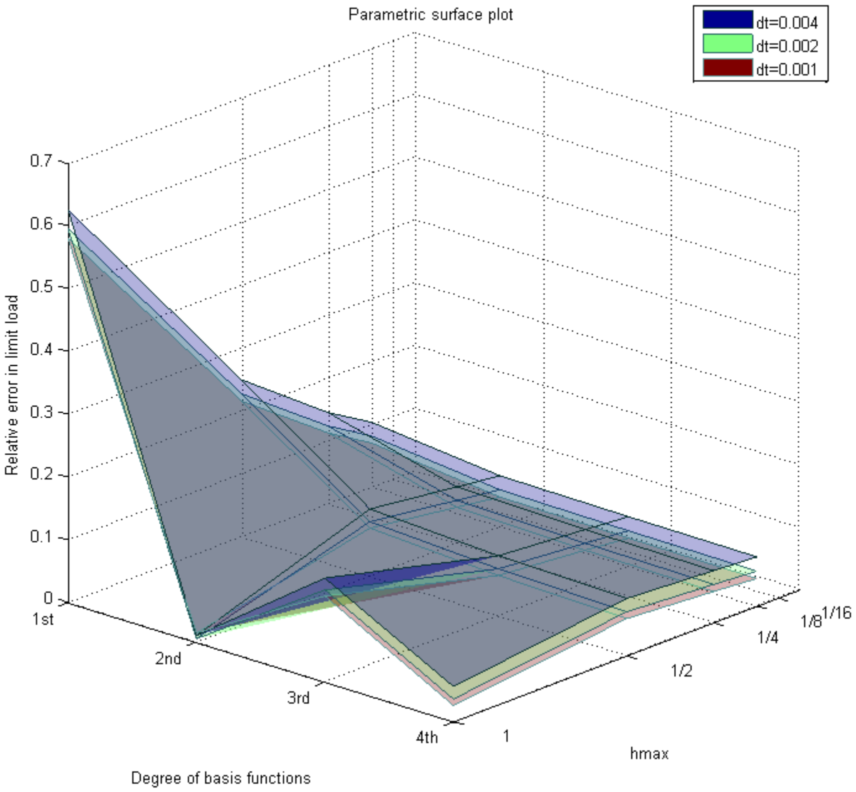


Figure 5.4 The three-variate parametric analyses in surface plots.

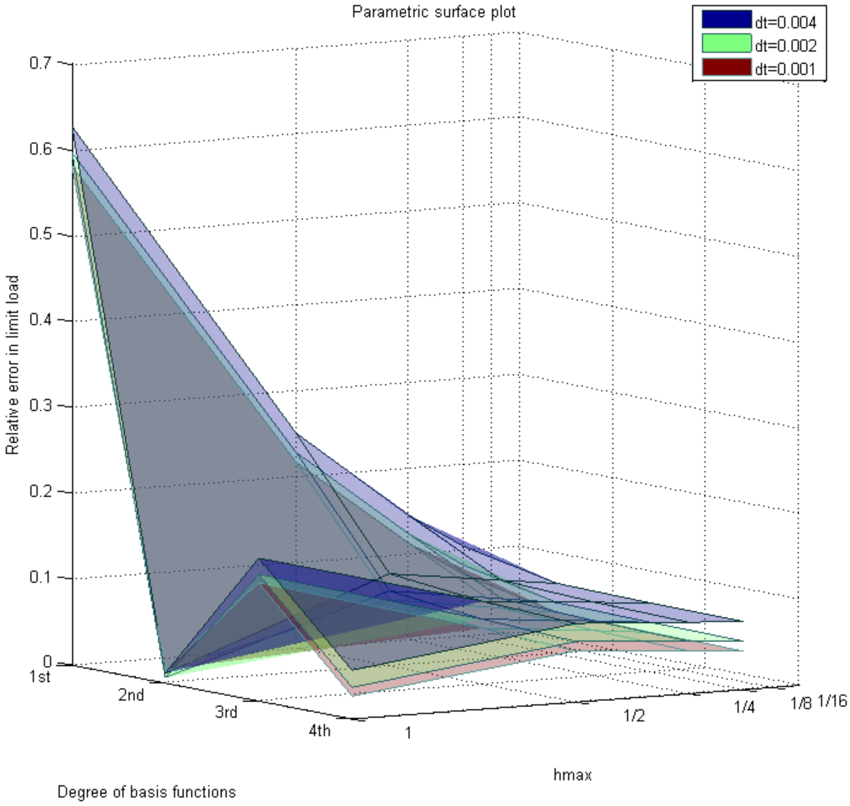


Figure 5.5 The three-variate parametric analyses in surface plots (second angle of the same plots).

The results of the parametric analyses illustrate these observations

- The basis functions of linear interpolation (classical FEM) fail to give a good result under a coarse mesh.
- The higher order bases functions give a low error from the very coarse mesh
- Some abnormalities are observed while the higher order bases functions are analysed under remeshing but seems to get stabilized for finer meshes.
- The step time is independent from the other variables (h, p) that is rendered by the parallel surfaces.

Generally, the accuracy of the results in the elastoplastic analyses performed in this work depends mainly on the degree of the basis functions in relation to the mesh size and consequently to the control points density. From our results, though, it becomes evident that for the particular problem solved the 2nd degree basis functions and a coarse meshing gave the most fit results. Apart from the suspicion that the density of the control points in relation to the degree of the basis affects the approximation of the solution field, the strange results may be attributed to the features of the problem, loading and boundary conditions. Larger parametric analyses in more mesh variants may clear this ambiguity, but due to lack of time and only these results in hand we can conclude with a reservation that the analyses converge at the last meshes as the gradient of the surface there vanishes.

CONCLUSION

The main advantage of isogeometric FEM is the isoparametric concept that is the same basis functions that represent exactly the geometry are used to approximate the unknown field of solution. The C^{∞} continuity at the interior of the knot spans and the considerable continuity at the knots gives also the ability to approximate well and smooth the derivatives of the solution field, i.e. the strain field, stress field, curvature field etc. In addition, in cases of exclusively discrete results the same type of basis (the degree of which is according to the number of the data to be interpolated e.g. integration points) can be used for accurate graphical representations of the solution.

Isogeometric analysis formulates stiffness matrices with the same bandwidth size as in the case of classical isoparametric elements, depending on the order of the basis. The benefit of using geometric bases is the ability to generically modify the mesh size directly from the parametric domain that **globally** affect the accuracy of the solution.

For the isogeometric formulation elastoplastic analysis is performed in a smoother way as the constitutive relations are built in the inelastic element. Moreover, utilizing the Bouc – Wen modeling of rate independent plasticity is straightforward in the isogeometric formulation. The generalized Bouc – Wen model expressed in the complete stress space is formulated in a variational form of the stationary principle of energy, which is equivalent to the weak formulation of the Galerkin method.

As presented in part II section 4.3 it turns out that with the same coarse discretization by elevating the degree of the basis the convergence rate increases considerably for the stress and energy fields, which is more pronounced for the displacement field.

The higher order geometric bases are powerful in non-linear stiff problems too. This is evident by this application of the Bouc-Wen hysteresis plasticity model. The very fine approximation of these geometric bases on the kinematics of the solid continuum gives rise to highly non-linear stiff problems to give accurate results by very coarse meshes. Combined with the power of hysteresis models that achieve directly by one step the non-linear response of a system, we can have accurately and ultra-fast the unknown solution field in time domain.

LIST OF FIGURES

BIBLIOGRAPHY

Akin, J. (2005). *Finite Element Analysis with error estimators.* Elsevier.

B.A. Finlayson, L.E. Scriven. (1966). The Method of Weighted Residuals-A Review. *Applied Mechanics Reviews*.

Barber, J. (2004). *Solid Mechanics and its Applications - Elasticity, Second Ed.* Kluwer Academic Publishers.

Bathe, K.-J. (1996). *Finite Element Procedures.* Prentice Hall.

Cook, Robert D., David S. Malkus, Michael E. Plesha, Robert J. Witt. (2002). *Concepts and Applications of Finite Element Analysis, Fourth Edition.* John Wiley & Sons, Inc.

Farin, G. (2002). *Curves and Surfaces for CAGD.* Morgan Kaufmann Publishers, Academic Press.

Faycal Ikhouane, Jose Rodellar. (2007). *Systems with Hysteresis.* Barcelona, Spain: Wiley.

Finlayson, B. A. (1972). *The Method of Weighted Residuals and Variational Principles.* Academic Press.

G. Thomas Mase, George E. Mase. (1999). *Continuum mechanics for engineers, Second Ed.* CRC Press.

Giorgio Bertotti, Isaak Mayergoyz. (2005). *The Science of Hysteresis, 3 Volume Set.* Elsevier.

Gockenbach, M. S. (2006). *Understanding and Implementing the Finite Element Method.* Houghton, Michigan: Society for Industrial and Applied Mathematics, Philadelphia.

Hughes, T. J. (1987). *The Finite Element Method, Linear Static and Dynamic FEA.* Prentice Hall.

J. Austin Cottrell, Thomas J.R. Hughes, Yuri Bazilevs. (2009). *Isogeometric Analysis Toward Integration of CAD and FEA.* John Willey & Sons, Ltd.

Les Piegel, Wayne Tiller. (1997). *The NURBS Book.* Berlin, Heidelberg: Springer.

Lubliner, J. (2008). *Plasticity Theory.* University of California at Berkley: Dover Publications.

Mayergoyz, I. (2003). *Mathematical Models of Hysteresis and their applications, Second Edition.* U.S.A.: Elsevier Science Inc.

O.C. Zienkiewicz & R.L. Taylor. (2000). *The Finite Element Method, Volume I, The basis, 5th Ed.* Butterworth Heinemann.

Reddy, J. N. (1993). *An introduction to the finite element method, Second Ed.* Texas: McGraw-Hill Inc.

Reddy, J. N. (2008). *An introduction to continuum mechanics.* Texas: Cambridge university press.

Richard H. Bartels, John C. Beatty, Brian A. Barsky. (1987). *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* LOS ALTOS, CALIFORNIA: MORGAN KAUFMANN PUBLISHERS.

Rogers, D. F. (2001). *An Introduction to NURBS.*

Solin, P. (2006). *Partial Differential Equations and the Finite Element Method.* Texas, El Paso: Wiley Interscience.

T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs. (2004). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering, Elsevier*.

Triantafyllou Savvas. (2011). Hysteretic Finite Elements and Macro-elements for Non-Linear Dynamic Analysis of Structures. *Doctoral Thesis*. Athens, Greece: National Technical University of Athens.

Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli. (2006). Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*.

Y.C.Fung, Pin Tong. (2001). *Classical And Computational Solid Mechanics, Advanced Series in Engineering Science Vol.1.* World Scientific.

Young W. Kwon, Hychoong Bang. (2000). *The Finite Element Method using Matlab, Second Ed.* CRC Press.

Γ.Ι. Τσαμασφύρος, Ε.Ε. Θεοτόκογλου. (2005). *Μέθοδος των πεπερασμένων στοιχείων Ι.* Εκδόσεις Συμμετρία.

Γ.Ι. Τσαμασφύρος, Ε.Ε. Θεοτόκογλου. (2005). *Μέθοδος των πεπερασμένων στοιχείων ΙΙ.* Εκδόσεις Συμμετρία.

Γδούτος, Ε. Ε. (1999). *Θεωρία Ελαστικότητας.* Εκδόσεις Αδελφών Κυριακίδη Α.Ε.

Παπαδρακάκης, Μ. (2001). *Ανάλυση Φορέων με τη Μέθοδο των Πεπερασμένων Στοιχείων*. Παπασωτηρίου.