REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DU GENIE DE LA CONSTRUCTION

DEPARTEMENT DE GENIE MECANIQUE

MÉMOIRE DE FIN D'ÉTUDES

En vue de l'obtention du diplôme de Master 2

En Génie Mécanique

Option : Construction Mécanique

THÈME

RECALAGE DE MODÈLE ÉLÉMENTS FINIS ISOGÉOMÉTRIQUE

Proposé et dirigé par :

M^{r} : F.ASMA

Réalisé par :

M^{r} : BACHA Ferhat

«2011»

Remerciements

Je remercie tout d'abord Dieu de m'avoir donné la force, la volonté et le courage pour l'élaboration de ce travail.

Je remercie mon promoteur Mr F.ASMA pour avoir bien voulu m'encadrer et pour ses précieux conseils et orientations.

Je remercie mon cousin BACHA Mouhemed pour son aide

Je remercie mon camarade BELGAID Hocine pour son aide

Mes remerciements vont également a tous mes enseignants, les responsables et personnel du département de Génie mécanique.

J'adresse mes plus vifs remerciements à tous ceux qui ont contribué de prés ou de loin à la réalisation de ce modeste travail.

Je dédie ce modeste travail

A mes très chers parents

A mes frères et sœur

A la mémoire de mon grand père

A la mémoire de ma grande mère

A toute ma famille

A tous mes amis

Sommaire

CHAPITRE IV APPLICATION ET DESCUTION

Liste des figures

Liste des figures

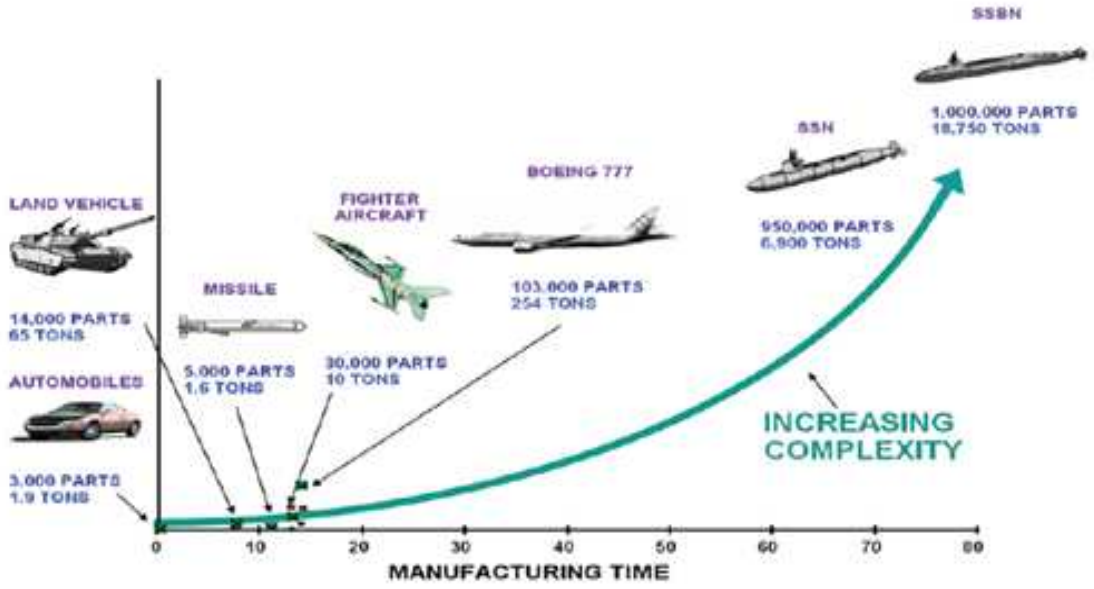
Liste des figures

Introduction générale

La géométrie est la base de l'analyse, pourtant des méthodes modernes pour le calcul géométrique jusqu'à récemment ont eu un très petit impact sur le calcul mécanique. Peut être la raison est que La méthode des éléments finis (MEF) n'a été développé que dans les années 50 et 60, avant l'apparition de la conception assistée par ordinateur (CAO) et sa généralisation dans les années 70 et 80.ceci peut expliquer pourquoi les représentations géométriques dans l'analyse d'élément finis et DAO sont si différents, les principaux programmes d'éléments finis étaient techniquement murs longtemps avants que le DAO ait été largement adopté. Actuellement, le DAO est plus industrialisé que l'analyse, l'analyse désignée sous le nom de l'ingénierie assistée par ordinateur (IAO) dans la recherche. Il est très difficile de mesurer avec précision la taille des industries de IAO et DAO .la situation typique dans la pratique en matière de technologie et que les conceptions sont renfermée dans le system DAO et le maillage est généré par des information de DAO. Alors adopter différentes descriptions géométriques pour l'analyse et une qui est seulement approximative.

Parfois le maillage peut être fait automatiquement mais dans la plupart des cas il peut être fait au mieux semi automatique. Il y a toujours des situations gouvernantes dans l'industrie dans lesquelles les schémas sont faits à partir du DAO et le maillage généré à partir des données de DAO.

Il est estimé qu'environ 80% de temps de l'analyse globale est consacré au maillage comme par exemple dans l'industrie automobile, aérospatiale et navale. Dans l'industrie automobile, un maillage entier d'un véhicule peut prendre environ quatre mois pour le construire. Alors la conception en ingénierie devient de plus en plus complexe comme le montre la figure suivante (voir nombre de pièces constituants l'objet)



Classement de quelques conceptions par rapport aux nombre de pièces, temps de mise au point et poids.

La conception et l'analyse sont deux domaines qui ne peuvent jamais être séparés, la conception des systèmes sophistiqués dans l'ingénierie est basée sur un éventail d'analyse numérique et des méthodes de simulations, comme la mécanique des structures, dynamique des fluides .etc…

Les tendances récentes prennent place dans l'analyse d'ingénierie et de calcul haute performance et demande également une très grande précision et une meilleure intégration de l'ensemble modélisation et analyse des processus. Nous notons qu'un maillage aux éléments finis n'est qu'une approximation de la géométrie DAO que nous considérons comme « exacte ». Ce rapprochement peut dans de nombreuses situations créer des erreurs dans les résultats d'analyse. Ceci exige le changement d'éléments finis classiques par un procédé d'analyse basé sur les représentations exactes de DAO. Ce concept désigné sous le nom de l'analyse Isogéométrique, a été proposé la première fois par Thomas J.R. Hughes, Cottrell, et Bazilevs. Elle permet de faire le lien entre le D.A.O. (dessin Assistée par Ordinateur) et la simulation numérique, elle utilise les mêmes concepts pour les calculs par éléments finis que pour la CAO: les fonctions NURBS (Non Uniform Rational B-Splines) par exemple. Ces éléments finis de nouvelle génération se nomment éléments finis isogéométrique. Cette technique est extrêmement intéressante car elle supprime définitivement la

notion de maillage éléments finis ainsi que la pauvreté de convergence des éléments finis habituels. L'idée principale repose sur l'utilisation de fonctions de base servant à définir la géométrie. Plusieurs avantages rendent l'approche très attractive comparée à la méthode des éléments finis classiques, elle permet une représentation précise voire exacte des géométries complexes.

Il y a actuellement une avancée vers une précision plus élevée et une meilleure simulation de la réalité. De nouvelles technologies sont présentées et adoptées rapidement dans les logiciels de conception, et des nouvelles et meilleures peuvent être établies avec ces nouvelles technologies de DAO et le nouveau concept d'analyse (l'analyse isogéométrique).

Cette nouvelle approche d'analyse isogéométrique et ces applications dans le domaine d'ingénierie et vraiment très signifiant. Dans ce travail on propose une application de ce nouveau concept dans le recalage du modèle. Le recalage de modèle éléments finis a déjà fait l'objet de nombreux travaux et publications, mais reste un sujet de recherche qui mobilise scientifiques et industriels, en particulier dans les domaines de l'aéronautique et de l'automobile. Dans ces domaines de pointe, l'optimisation du design pour des raisons de réduction des coûts, de gain de poids et d'accroissement des performances, nécessite une modélisation de plus en plus fine du comportement réel des structures.

Ce travail est organisé en quatre chapitres.

Le premier chapitre décrit brièvement l'évolution de la modélisation géométrique des courbes et surface, voir les courbes de Bézier et les polynômes de Bernstein en 1962 avec les travaux de PIERRE Bézier et les courbes B-spline puis NURBS.

La méthode de l'analyse isogéométrique et sa comparaison avec l'analyse aux éléments finis classiques est illustré dans le chapitre II

Le chapitre III est destiné aux recalages et les différentes méthodes utilisées.

Est en fin en termine par une application de recalage. Et on termine par une conclusion générale.

I. MODELISATION GEOMETRIQUE ET RAFFINEMENT

I.1 MODELISATION GEOMETRIQUE DES COURBES ET SURFACES

I.1.1 Modèle de Bézier polynomial

I.1.1.1 Introduction

Le modèle de Bézier (proposé par l'ingénieur français Pierre Bézier en 1962) a été développé dans les bureaux d'études de Renault, il a étudié le problème de conception de surfaces 3D (carrosseries d'automobile,…) ; pour les premiers programmes de la Conception Assistée par Ordinateur CAO ; le but était de trouver un moyen pour définir une courbe de manière précise et simple.

Il s'agit d'un modèle permettant de créer des formes du plan et de l'espace à partir de points de contrôle [1].

I.1.1.2.POLYNOMES DE BERNSTEIN

I.1.1.2.1. Introduction

Pierre Bézier a utilisé des polynômes d'approximation de fonctions pour la description de ces courbes et de ces surfaces appelés polynômes de Bernstein.

I.1.1.2.2.Définition

On définit pour i ∈ {0…..n} les polynômes de Bernstein par la formule :

$$\boldsymbol{B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}} \quad \text{Avec} \quad t \in [0\ 1] \ \textbf{…… (I-1.1)}$$

C_n^i: Coefficients du binôme de Newton définis comme suit :

$$\boldsymbol{C_n^i = \frac{n!}{(n-i)!\,i!}} \ \textbf{…… (I-1.2)}$$

Exemple : pour n=3 et i=1, 2, 3 on trouve les polynômes suivant, et les courbes associées sur MATLAB :

$$B_{03}(t) = (1-t)^3; B_{13}(t) = 3t(1-t)^2;\ B_{23}(t) = 3t^2(1-t); B_{33}(t) = t^3.$$

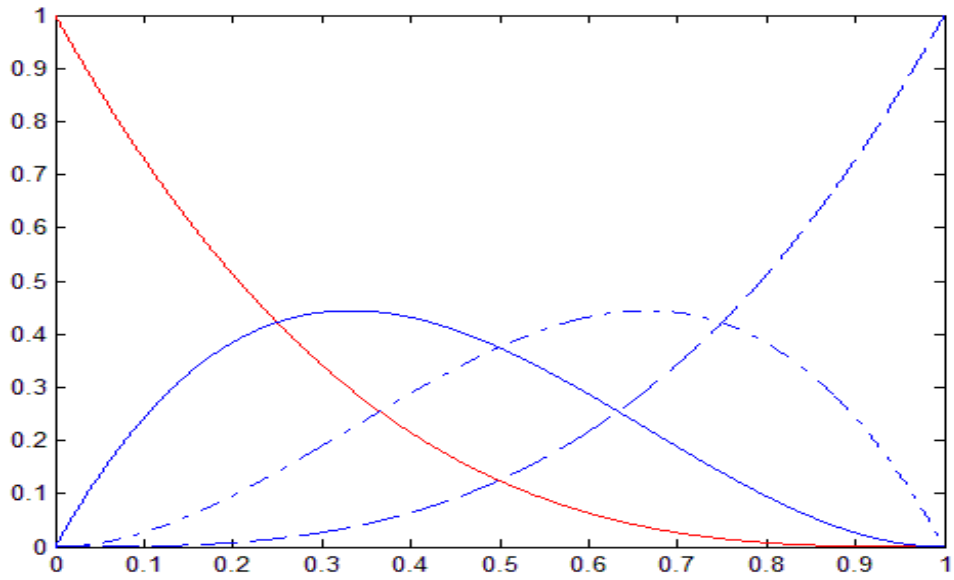


Figure (I-01) *: courbes des fonctions de base de Bernstein de degré 3*

I.1.1.2.3.Propriétés des polynômes de Bernstein

- Partition unité

$$\sum_{i=0}^{n} B_{i,n}(t) = \big(t + (1-t)\big)^n = 1 \qquad \forall\, t \in [0\ 1] \qquad B_{i,n}(t) \geq 0$$

Pour t=0 $B_{0,n}(0) = 1$ et $B_{i,n}(0) = 0$

Pour t=1 $B_{i,n}(1) = 1$ et $B_{n,n}(1) = 1$

- Relation de récurrence

$$B_{i,n}(t) = t.B_{i-1,n-1}(t) + (1-t)B_{i,n-1}(t) \qquad \forall\ t\ \in [0\ 1]$$

$$B_{0,n}(t) = t.B_{0,n-1}(t) \qquad \forall\ i\ \in \{1 \ldots \ldots n-1\}$$

- Maximum

  Max $B_{i,n}(t) = B_{i,n}\left(\frac{i}{n}\right)$ donc $a \;\; t = \frac{i}{n}$ $B_{i,n}(t)$ atteint son maximum

- Symétrie

Les polynômes de Bernstein sont symétriques par apport à t = 1/2

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

- Intégration

$\int_0^1 B_{0,3}(t)\,dt = \int_0^1 (1-t)^3 dt = \frac{1}{4}$

$\int_0^1 B_{1,3}(t)\,dt = \int_0^1 3t\,(1-t)^2 dt = \frac{1}{4}$

$\int_0^1 B_{2,3}(t)\,dt = \int_0^1 (3t^2 - 3t^3) dt = \frac{1}{4}$

$\int_0^1 B_{3,3}(t)\,dt = \int_0^1 t^3 dt = \frac{1}{4}$

On remarque que toutes les intégrales sont égales à $\frac{1}{4}$

Donc on peut généraliser pour tout *n* différent de 0 et $\forall$ i ϵ {0…..n}

Que $\int_0^1 B_{i,n}(t)\,dt = \frac{1}{n+1}$

I.1.1.3.Courbes de Bézier non rationnelles

Une courbe de Bézier de degré 3 est caractérisée généralement par 4 points, appelés points de contrôle qui définissent le polygone caractéristique associé à cette courbe. Le premier point et le dernier sont des nœuds. Les deux autres points de contrôle permettent de définir la forme de l'arc (courbe), la courbe ne passant pas en général par ces deux points [1].

Pour définir de manière mathématique une courbe de Bézier de degré *n* associée au polygone caractéristique (PG) défini par les points de contrôles (P_0, P_1……, P_n) dans le repère R (O, X, Y), on utilise les polynômes de Bernstein à l'aide de l'équation suivante :

$$\boldsymbol{C_n(t) = \sum_{i=0}^{n} B_{i,n}(t).P_i} \quad \text{..... (I-1.3)}$$

Avec :

$\boldsymbol{B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}}$ **:** Polynômes de Bernstein où n est le degré de la courbe

i = {0 1 2 ……n}

tϵ [0 1]

Pi= $\begin{Bmatrix} X \\ Y \end{Bmatrix}$ cordonnées des points de contrôle qui définissent le polygone caractéristique dans le repère R (O, X, Y)

Pour n=3 par exemple, à l'aide de MATLAB on retrouve la courbe représentée sur la Figure I-02

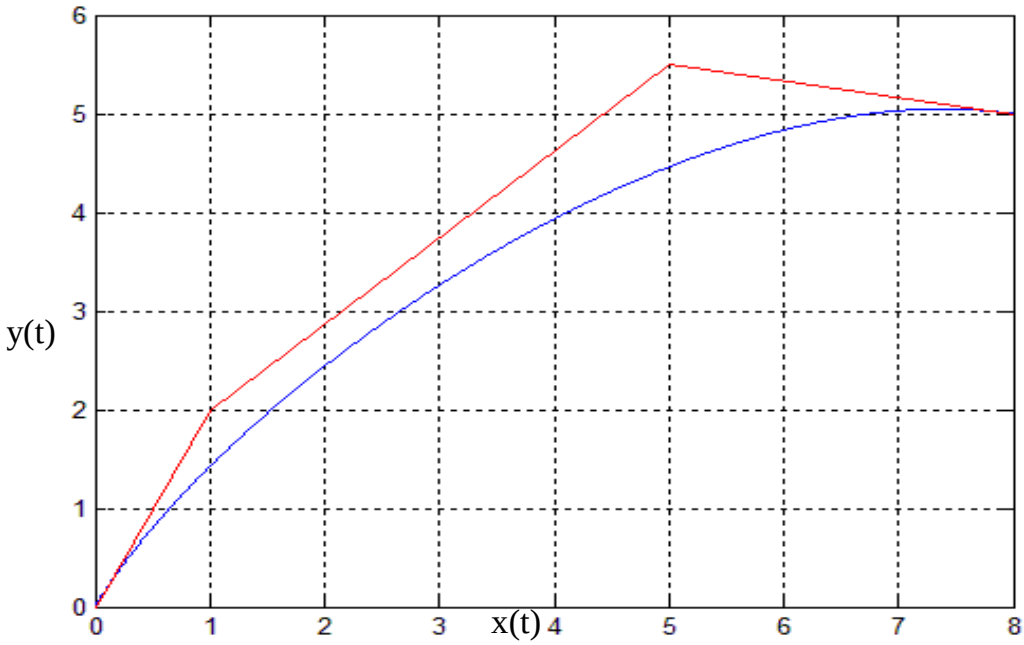


Figure I-02 : *courbe de Bézier non rationnelle et son polygone caractéristique*

I.1.1.3.1. Méthode de De Casteljau (algorithme de De Casteljau)

Les points de la courbe de Bézier $\boldsymbol{C_n(t)}$ peuvent être évalués différemment à l'aide de l'équation (I-1.4) en utilisant l'algorithme de De Casteljau.

Etant donnés les points de contrôle P_0,…... P_1 de la courbe de Bézier et t ϵ [0,1], l'algorithme de De Casteljau est défini comme suit :

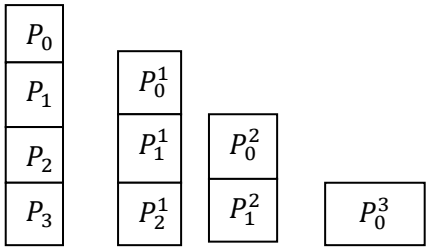
$$\boldsymbol{P_i^r(t) = (1-\tau)P_i^{r-1}(t) + \tau P_{i+1}^{r-1}(t)} \text{ Avec } \begin{cases} r = 1, \dots \dots . n \\ i = 0, \dots \dots \dots n-1 \end{cases} \cdots \textbf{(I-1.4)}$$

$$\text{Et} \quad \boldsymbol{P_i^0(t) = P_i} \ \textbf{…….. (I-1.5)}$$

Et le polygone formé par les points P_0, P_1, …… , P_n, est appelé polygone de Bézier ou le polygone de contrôle. Le coefficient$\boldsymbol{P_i^r(t)}$ peut s'écrire sous la forme triangulaire.

Exemple : $\boldsymbol{P_0^3(t)}$

Sa forme triangulaire peut être représentée par :



La figure suivante montre une construction d'une courbe de Bézier par la méthode de De castiljau

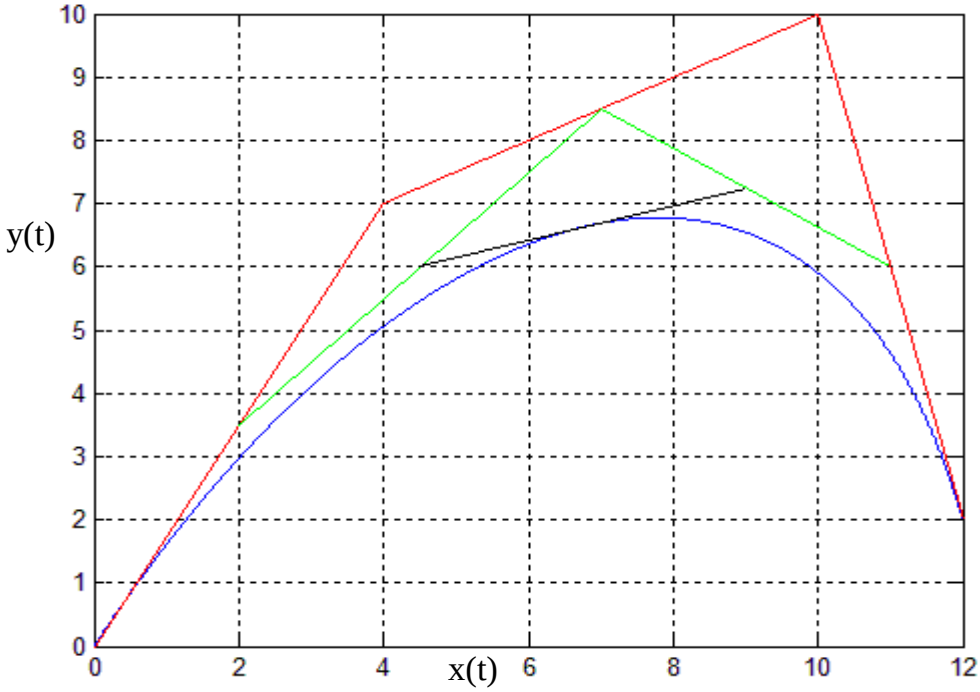


Figure (I-3) : *Construction d'une courbe par la méthode de De Casteljau*

I.1.1.3.2. Propriétés des courbes de Bézier non rationnelles :

- une courbe de Bézier associée aux points P_0, P_1….. ,P_n passe par les points P_0, P_n ; mais en général elle ne passe pas par les autres points.

- Une courbe de Bézier se trouve à l'intérieur de l'enveloppe convexe (polygone caractéristique) de ses points de contrôle.

- Une courbe de Bézier est infiniment dérivable (de classe C^n).
- Les polynômes de Bernstein sont strictement positifs sur [0,1].par conséquent, si on modifie un des points de contrôle, toute la courbe de Bézier sera modifiée.

- la courbe de Bézier est tangente en P_0 au segment [P_0, P_1] et en P_n au segment [P_{n-1}, P_n]

I.1.1.4. Surface de Bézier non rationnelle

I.1.1.4.1. définition

La surface de Bézier est une extension directe de la courbe de Bézier non rationnelle. Toute extension dans deux direction u, v des courbes de Bézier non rationnelles est la représentation des surfaces de Bézier non rationnelle. On définie la surface de Bézier par :

$$\boldsymbol{S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v).P_{i,j}} \text{ … … … … … … …. } \mathbf{(I-1.6)}$$

$P_{i,j}$: Les pôles du réseau caractéristique (les points de contrôles) avec :

- $0 \leq i \leq n$
- $0 \leq j \leq m$

$B_{i,n}(u)$, $B_{j,m}(v)$: Les polynômes de Bernstein avec (u, v)ϵ $[0,1]^2$

Les points de contrôle associés à la surface de Bézier sont aussi les points de contrôle correspondant à un ensemble de courbes de Bézier.

L'expression précédente (I-1.6) peut s'écrire sous forme matricielle comme suit :

$$\boldsymbol{S(u,v) = [B_{0,n}(u), B_{1,n}(u), \dots\dots, N_{n,n}(u)] \begin{bmatrix} P_{0,0} \dots\dots\dots & P_{0,m} \\ \vdots & \vdots \\ P_{n,0} \dots\dots\dots & P_{n,m} \end{bmatrix} \begin{bmatrix} B_{0,m}(v) \\ \cdot \\ \cdot \\ \cdot \\ B_{m,m}(v) \end{bmatrix} \dots \textbf{(I-1.7)}}$$

Avec : *u ϵ [0,1]et v ϵ [0,1]*

La figure suivante montre une surface de Bézier non rationnelle

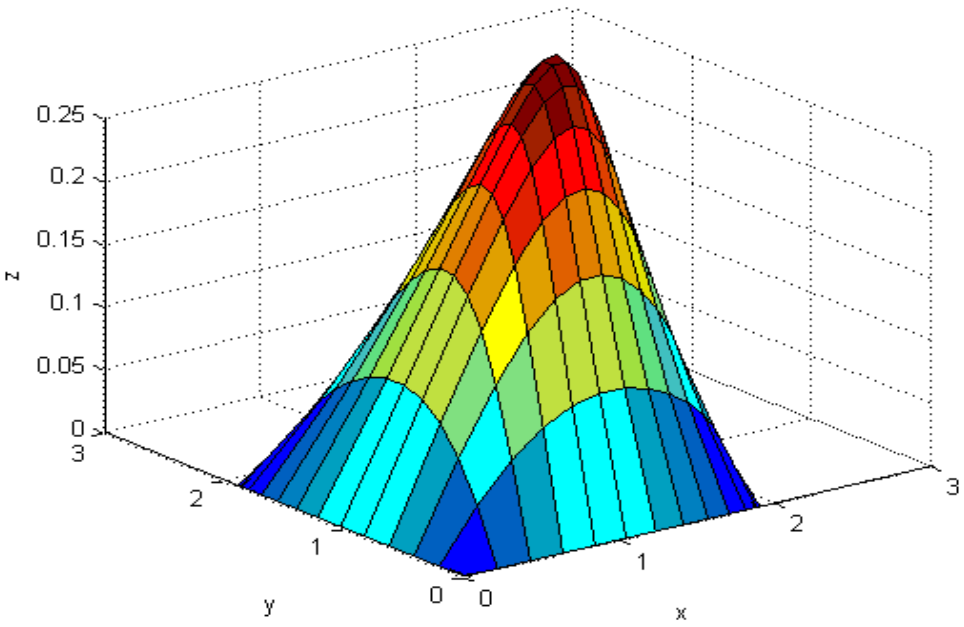


Figure (I-4) : *Surface de Bézier non rationnelle*

I.1.1.4.2. Propriétés des surfaces de Bézier

- non négativité : $B_{i,n}(u)B_{j,m}(v) \geq 0 \ \ \forall\, i,j,u,v$
- partition unité : $\boldsymbol{\sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,n}(u)B_{j,m}(v) = 1} \ \ \forall\, u\ et\ v$
- enveloppe convexe : la surface $\boldsymbol{S(u,v)}$ est contenue dans l'enveloppe convexe des points de contrôle
- la surface passe par les quatre points extrêmes (interpolation)

I.1.2 Modèle de Bézier rationnel

Pour décrire très exactement des courbes comme les cercles, il faut des degrés de liberté supplémentaires.

L'idée est d'ajouter des poids aux points de contrôle (les w_i).le dénominateur n'est là que pour normaliser la somme des poids supplémentaires, afin que la courbe soit correctement définie. Les modèles géométriques rationnels font appel aux concepts des cordonnées homogènes.

I.1. 2.1. Cordonnées homogènes

Le concept général des coordonnées homogènes est d'exprimer des points de l'espace R^N (dimension N) dans un espace R^{N-1} (dimension N-1)

Soit P un point de coordonnées *(x, y, w)* de R^3, sa projection est obtenue comme suit:

$$F : R^3 \longrightarrow R^2$$

$$P\begin{pmatrix} x \\ y \\ w \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \end{pmatrix}$$

L'emploi des coordonnées homogènes en CAO offre plusieurs avantages comme la possibilité de combiner toutes les transformations géométriques courantes (translation, symétrie, rotation,…)

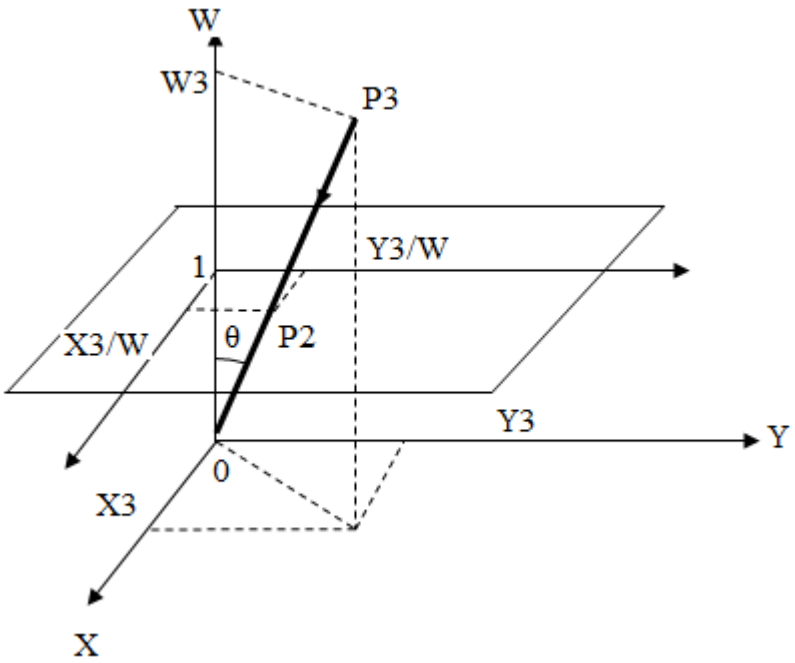


Figure (I- 5) : *illustration de la transformation homogène entre R^3 et R^2*

I.1. 2. 2. Courbes de Bézier rationnelle

Une courbe de Bézier rationnelle est définie par l'équation suivante :

$$\boldsymbol{C(t) = \frac{\sum_{i=0}^{n} B_{i,n}(t) W_i P_i}{\sum_{i=0}^{n} B_{i,n} W_i}} \ldots\ldots\ldots\ldots\ldots\ldots (\boldsymbol{I - 1.8})$$

$$\boldsymbol{C(t) = \sum_{i=0}^{n} S_{i,n}(t) P_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\boldsymbol{I - 1.9})$$

$$\boldsymbol{S_{i,n}(t) = \frac{B_{i,n}(t) W_i}{\sum_{i=0}^{n} B_{i,n} W_i}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\boldsymbol{I - 1.10})$$

$\boldsymbol{S_{i,n}(t)}$: Polynômes de Bernstein rationnels

$\boldsymbol{B_{i,n}(t)}$: Polynômes de Bernstein

$\boldsymbol{W_i}$: Les poids associés aux pôles de polygone caractéristique avec W_i toujours positifs

$$W_i > 0 \;\; et \; t\epsilon \; [0,1]$$

$\boldsymbol{P_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}}$: Pôles du polygone caractéristique.

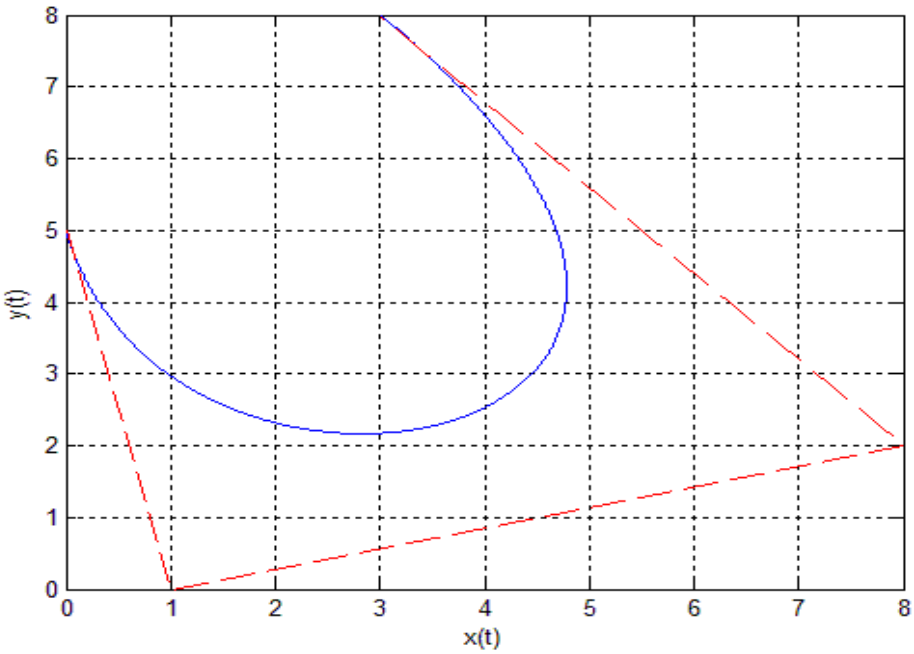


Figure (I-6) : *courbe Bézier rationnelle avec w=[1 1 1 1]*

I.1. 2.2.1 Propriétés des courbes de Bézier rationnelles

- Toute courbe de Bézier passe par les points de contrôles extrêmes
- Invariance affine : la transformation affine d'une courbe de Bézier est la courbe passant par la transformée des points
- Enveloppe convexe : une courbe de Bézier appartient à l'enveloppe convexe des points qui la contrôlent si les poids $\boldsymbol{W_i}$ sont tous positifs.
- La symétrie : la courbe de Bézier rationnelle associée au polygone de contrôle ne changera pas de forme lorsqu'on fait inverser les pôles du ce polygone et les vecteurs poids associés

La courbe de Bézier rationnelle présente deux autres propriétés par rapport au modèle de Bézier non rationnel [2]:

- La première propriété c'est l'invariance projective : si on veut transformer une courbe de Bézier rationnelle par une transformation projective, on a qu'à agir sur le polygone de contrôle en écrivant les poids (w_i) en coordonné homogène
- La deuxième est la précision linéaire

I.1. 2. 2.2 Influence des poids (w_i) sur la courbe de Bézier rationnelle

Les poids W_i sont des paramètres de contrôle supplémentaires de la forme de la courbe

- Si on augmente un poids W_i la courbe se rapproche du point (P_i) associé comme le montre la figure (I-7)
- Si W_i =0 le point P_i associé perd toute influence sur la forme de la courbe figure(I-8)
- Si on change le signe d'un poids W_i, la courbe s'éloigne du point (P_i) associé comme le montre la figure (I-8)

Remarque : nous remarquons que la modification des poids W_i a une grande influence sur la forme de la courbe même si on peut obtenir cette modification par une modification des points de contrôle (P_i) mais la modification des poids est plus souple et simple à mettre en œuvre dans la pratique.

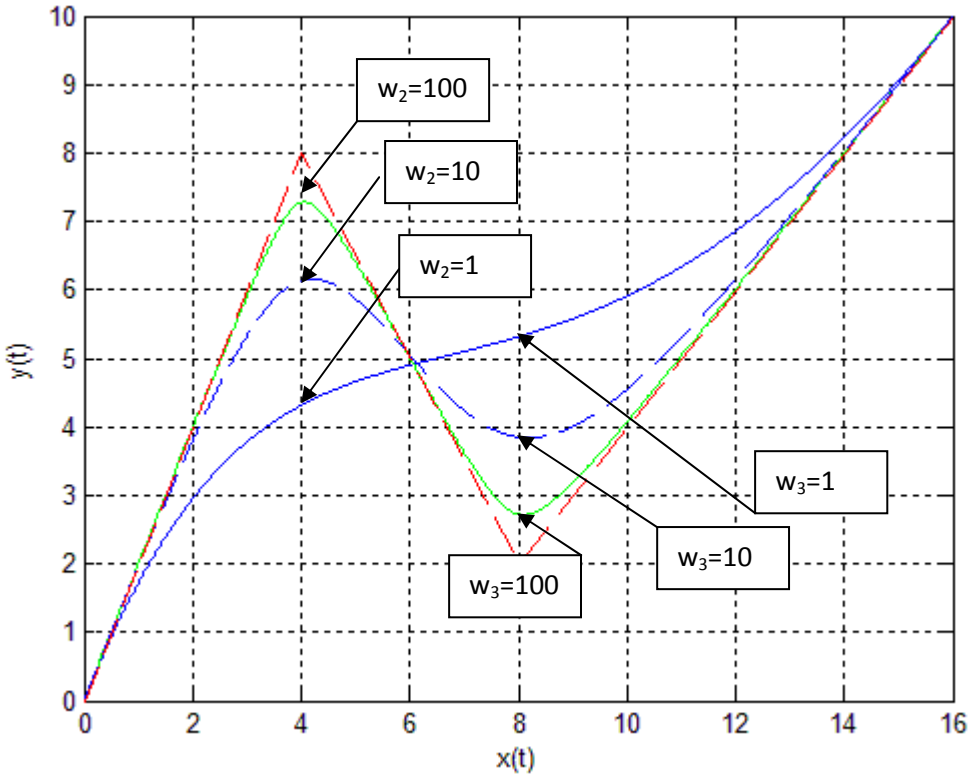


Figure (I-7) : *Influence d'une augmentation d'un poids sur la forme de la courbe*

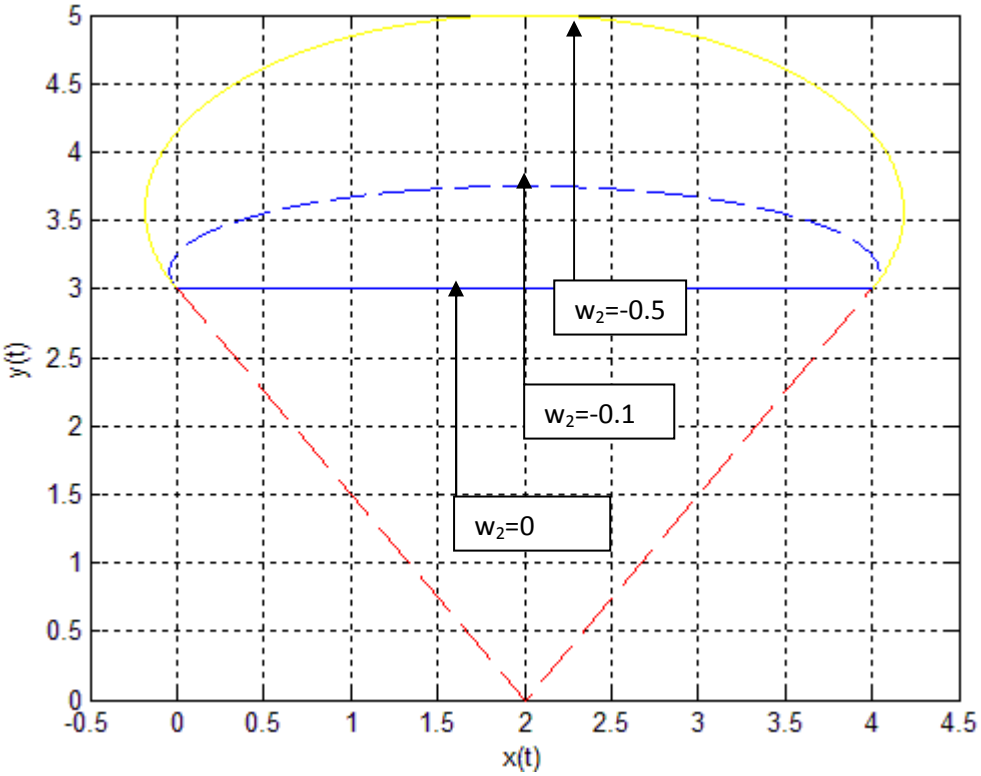


Figure (I-8) : *Influence du changement de signe d'un poids sur la forme de la courbe*

I.1. 2.3. Surfaces de Bézier rationnelles

La surface de Bézier rationnelle se construit d'une manière analogue à celle de Bézier non rationnelle (polynômiale) en utilisant la formulation suivante :

$$\mathbf{s(u,v)} = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{W_{i,j}\, B_{i,n}(u) B_{j,m}(v) P_{i,j}})}{\sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{W_{i,j}\, B_{i,n}(u) B_{j,m}(v)}} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.11})$$

Avec :uϵ [0, 1], vϵ [0, 1] , $W_{i,j} > 0$

$$\mathbf{s(u,v)} = \sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{G_{i,n}(u) G_{j,m}(v) P_{i,j}} \ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.12})$$

$$\mathbf{G_{i,n}(u) G_{j,m}(v)} = \frac{\mathbf{W_{i,j} B_{i,n}(u) B_{j,m}(v) P_{i,j}}}{\sum_{i=0}^{n}\sum_{j=0}^{m} \mathbf{W_{i,j}\, B_{i,n}(u) B_{j,m}(v)}} \ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.13})$$

$P_{i,j}$: Les pôles du réseau caractéristique (les points de contrôles) avec :

- $0 \leq i \leq n$
- $0 \leq j \leq m$

$B_{i,n}(u)$, $B_{j,m}(v)$: Les polynômes de Bernstein avec (u, v)ϵ $[0,1]^2$

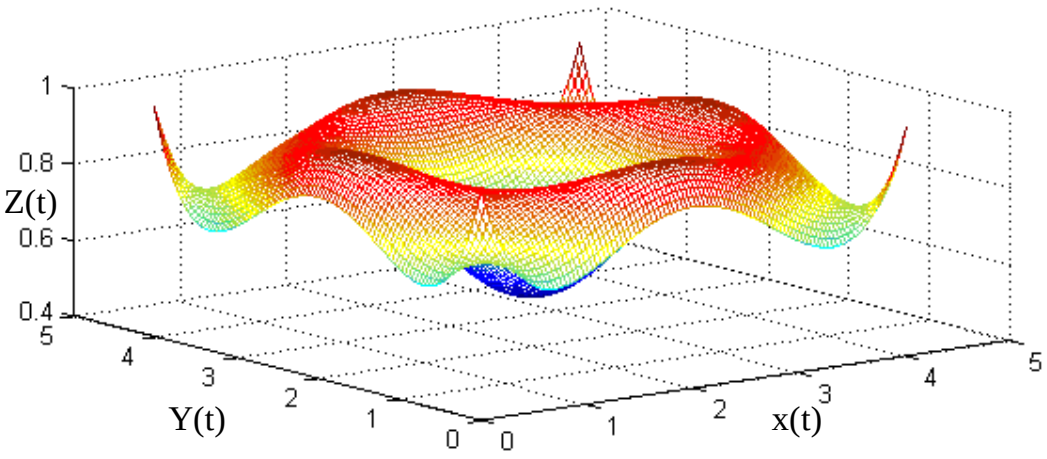


Figure (I-9) *: surfaces Bézier rationnelle*

I.1.3. Modèle B-Spline non rationnel

I.1.3.1. Introduction

L'Inconvénient des courbes de Bézier ; définies précédemment ; est que les formes de l'objet sont définies par le biais de points de contrôle, la modification locale d'un de ces points perturbe l'allure globale de la courbe.

La définition de courbes polynomiales par morceau (B-spline) permet de répondre aux insuffisances du modèle de Bézier, car elle offre une grande flexibilité et une grande précision. Les B-spline (basis spline) ont été proposées par De Boor (1972) et utilisées pour la première fois par Riesenfeld dans des applications CAO en 1973 [2].

I.1.3.2 Définition

Les courbes B-splines sont définies à partir d'une combinaison linéaire des fonctions polynomiales (fonctions de base) d'une manière analogue à celle de Bézier qui sont définies à base des polynômes de Bernstein.

I.1.3. 3. Nœud et vecteur nodal

Soit (ξ_0, ξ_1, ξ_2……. ξ_m) une suite de m+1 entiers naturels, telle que cette suite soit croissante ($\xi_{i-1} \leq \xi_i$). Ces valeurs sont appelées **nœuds** et l'élément (ξ_0, ξ_1, ξ_2,…… ξ_m) est appelé un **vecteur nœud** ou **vecteur nodal**. Il existe deux familles de vecteurs nodaux :

- ✓ **Vecteur uniforme**
  Un vecteur nodal est dit uniforme si l'espacement entre ses nœuds est uniforme Ξ = [0 1 2 3 4 5 6]
- ✓ **Vecteur non uniforme**
  Un vecteur est dit non uniforme si l'espacement entre ses nœuds n'est pas régulier. Ξ = [0 0 0 1 2 3 3 3]

I.1.3. 3.1. Multiplicité

La multiplicité d'un nœud est le nombre de fois qu'il apparait dans une séquence nodale Ξ =[0 0 1 2 2 2]

La multiplicité de 0 est égale à 2, La multiplicité de 1 est égale à 1et la multiplicité de 2 est égale à 3

I.1.3.4. Fonctions de base

Soit un vecteur nodal $\boldsymbol{\Xi}$ composé de valeurs nodales croissantes

$\boldsymbol{\Xi} = [\xi_0 , \xi_1 , \xi_2 , \ldots\ldots \xi_m]$

La définition récursive des fonctions B-splines donnée par De Boor est comme suit :

$$N_{i,m}(\xi) = \frac{\xi - \xi_i}{\xi_{i+m} - \xi_i} N_{i,m-1}(\xi) + \frac{\xi_{i+m+1} - \xi}{\xi_{i+m+1} - \xi_{i+1}} N_{i+1,m-1}(\xi) \ \ldots\ldots\ldots\ldots . (\mathbf{I - 1.14})$$

Avec $\xi \epsilon [\xi_{i+1}, \xi_{i+m+1}]$

$$N_{i,0}(\xi) = \begin{cases} 1 & si\ \xi_i \leq \xi < \xi_{i+1} \\ 0 & ailleurs \end{cases} \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(I-1.15)}$$

On convient, dans cette définition, que si le numérateur et le dénominateur sont nuls ensembles ou le dénominateur est nul, alors l'expression $N_{i,m}(\xi)$ est nulle.

Donc $\frac{0}{0} = 0$

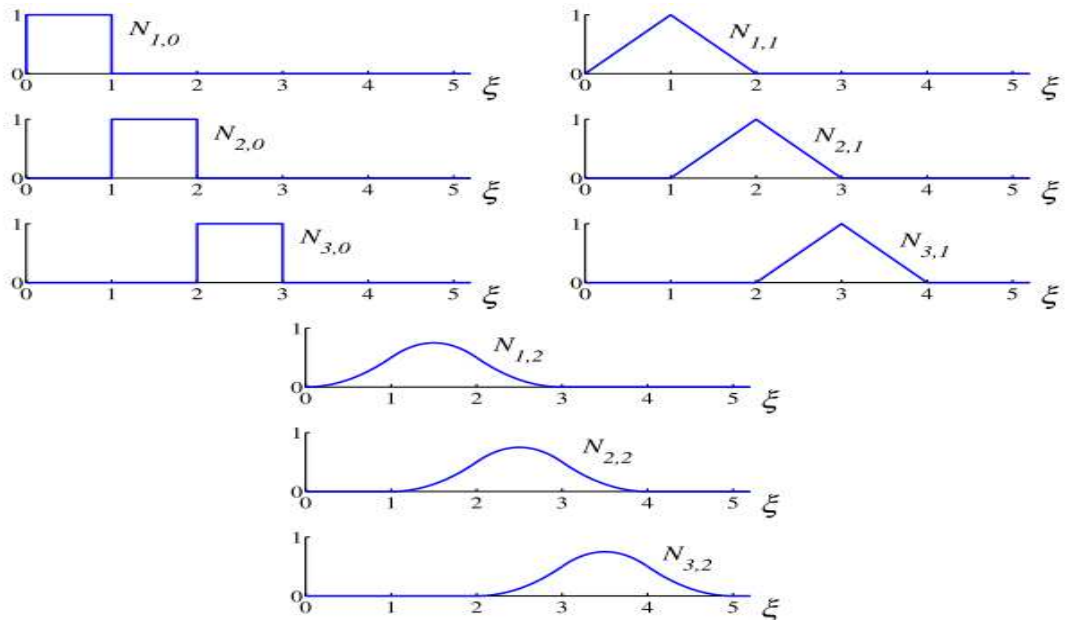


Figure (I-11) : *fonctions de base d'ordre 0, 1, 2pour le vecteur uniforme* $\boldsymbol{\Xi}$*= [0 1 2 3 4 5]*

I.1.3.5. Différents types de fonctions de B-Splines

I.1.3.5. 1. B-Splines uniformes

On appelle B-spline uniforme les fonctions de base et leur courbe construite à l'aide d'un vecteur nodal où l'intervalle entre deux nœuds est constant, qu'on peut représenter sous forme d'une suite arithmétique :
$\Xi_{j+1} = \Xi_j + \Delta\Xi$ *avec generalement* $\Delta\Xi = 1$ *et* $\Xi_0 = 0$.

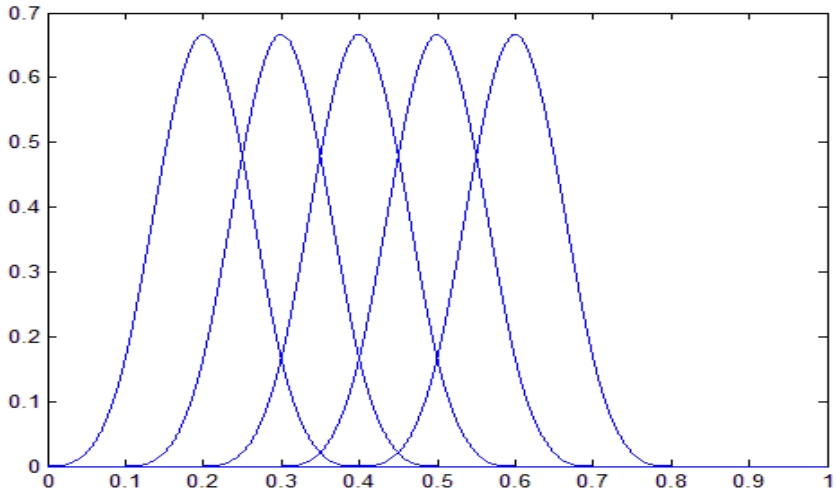


Figure (I-10): *courbes des fonctions de base B-spline uniformes*

I.1.3.5. 2. B-Splines non uniformes

On appelle B-spline non uniforme les fonctions de base et leurs courbes construites à l'aide d'un vecteur nodal où l'intervalle entre deux nœuds successifs n'est pas constant. exemple : le vecteur nodal $\boldsymbol{\Xi}$=[0 0 0 1 2 3 3 3]

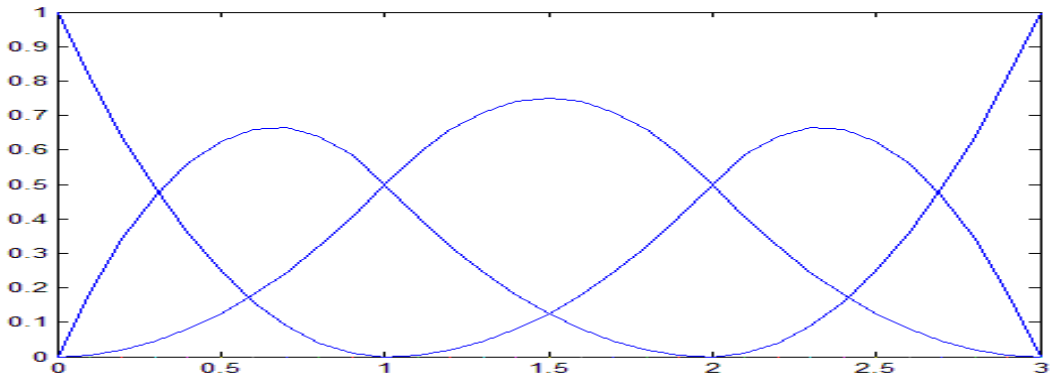


Figure (I-11) : *courbes des fonctions de base B-spline non uniformes* $\boldsymbol{\Xi}$ *= [0 0 0 1 2 33 3]*

I.1.3.6. Propriétés des fonctions de base

- chaque intervalle $[\xi_i ; \xi_{i+1}]$ définit un arc de courbe délimité par les points $P(\xi_i)$ et $P(\xi_{i+1})$
- la fonction $N_{i,m}(\xi)$ s'annule en dehors de l'intervalle $[\xi_i ; \xi_{i+1}]$ ce qu'on appelle support local

- pour tout ξ appartenant à $[\xi_i ; \xi_{i+1}]$ la somme des fonctions B-splines, pour *m* fixé est égale à 1 :

$$\sum_{i=0}^{n} N_{i,n}(\xi) = 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (I - 1.16)$$

- le non négativité : la fonction de base $N_{i,m}(\xi)$ est toujours positive

$$N_{i,m}(\xi) \geq 0 \quad i=1, 2, 3, \ldots\ldots\ldots\ldots\ldots n$$

- dérivabilité et continuité

si dans l'intervalle $[\xi_i ; \xi_{i+m+1}]$ tous les nœuds sont de multiplicité **1**,la fonction $N_{i,m}(\xi)$ est continûment dérivable jusqu'a l'ordre m-1 .par contre s'il existe un nœud ξ_i de multiplicité **p** dans cet intervalle , $N_{i,m}(\xi)$ est **m-p** fois continûment dérivable en ce nœud ξ_i

- le vecteur des nœuds influe principalement sur les propriétés de continuité et de dérivabilité.

I.1.3.7. Courbes B-spline non rationnelles uniformes

Les fonctions B-spline définies précédemment vont maintenant servir à définir les courbes B-spline. Le choix du vecteur nœud va permettre d'introduire des éléments de discontinuité agissant sur la forme de la courbe.

On définit une courbe B-spline à partir de *n+1* points $\boldsymbol{P_0, P_1, P_2, \ldots\ldots\ldots P_n}$ par :

$$\boldsymbol{C(\xi) = \sum_{i=0}^{n} N_{i,m}(\xi) P_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{(I - 1.17)}$$

Avec $\boldsymbol{P_i}$ **:** les pôles du polygone de contrôle

$\boldsymbol{N_{i,m}}$: Fonctions de base de degré m

pour m=2 la courbe B-spline est de degré 2

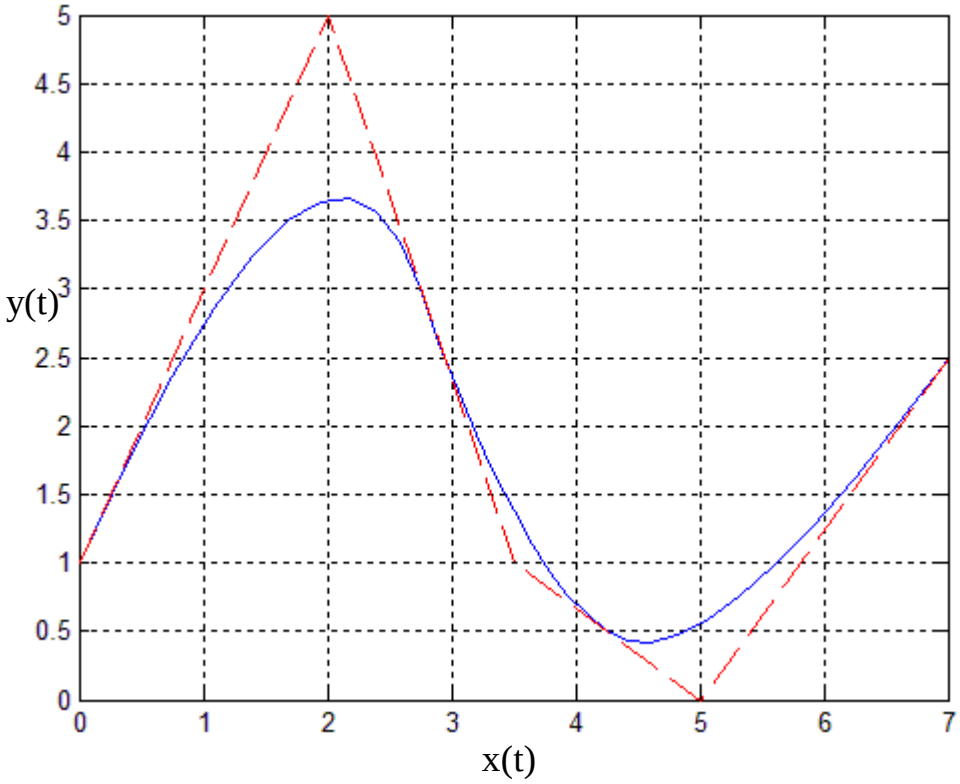


Figure (I.12): *courbe B-spline non rationnelle uniforme*

I.1.3.8. Propriétés des courbes B-spline non rationnelles

Les courbes B-Spline non rationnelles présentent les propriétés suivantes :

- enveloppe convexe : la courbe est contenue dans une enveloppe convexe des pôles
- contrôle local : le déplacement d'un pole engendre une déformation locale de la courbe
- invariance affine : une transformation affine est appliquée à la courbe si elle est appliquée au polygone caractéristique
- la courbe B-spline non rationnelle coïncide avec son polygone caractéristique au premier et au dernier point de contrôle
- la courbe B-spline et *p fois* continument dérivable *(p=m-k)*

En plus de ces propriétés on trouve une propriété de diminution de variation, cette propriété est particulière en comparaison avec le comportement d'une courbe représenté par les polynômes standard de Lagrange. On prend l'exemple [3] illustré sur la figure (I-13). On note qu'en mesure que l'ordre augmente l'amplitude des oscillations augmente également, par contre les B-Spline se comportent très différemment. Cette propriété mène les B-spline à avoir une très grande utilité dans l'analyse.

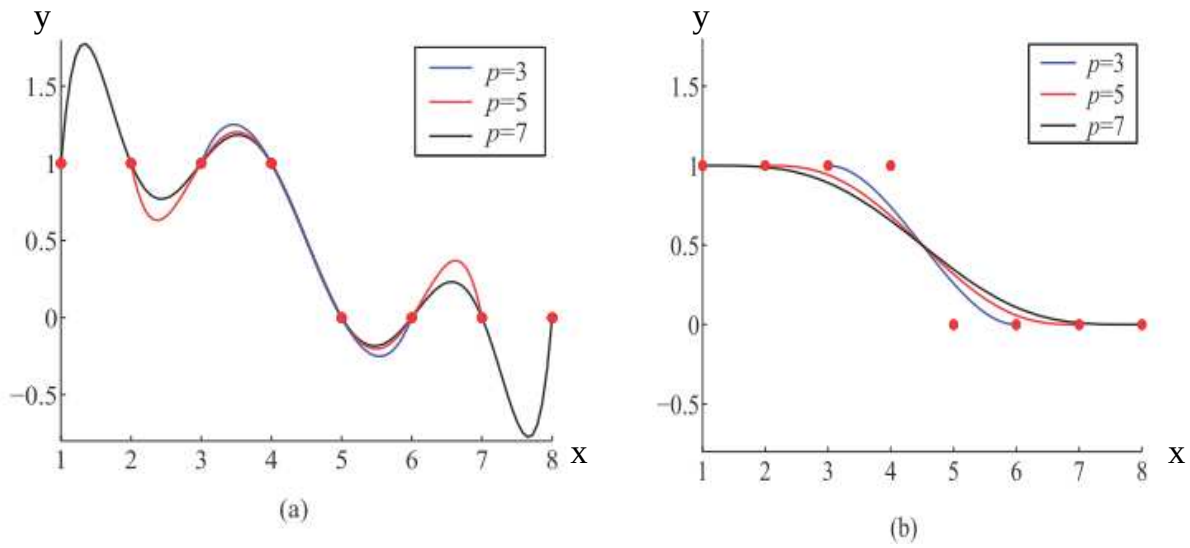


Figure (I-13) : *comparaison entre l'interpolation de Lagrange et B-Spline*
a)-interpolation de Lagrange b)- interpolation B-spline

I.1.3.9. Surfaces B-spline non rationnelles

La surface B-spline est une extension directe de la courbe B-spline non rationnelle. Toute extension dans deux directions ***u, v*** simultanément des courbes B-spline non rationnelles est la représentation des surfaces B-spline non rationnelle. une surface B-Spline de degré *(p,q)*est définie comme suit :

$$S(\xi,\mathfrak{y}) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(\xi)N_{j,q}(\mathfrak{y})P_{i,j} \quad \dots\dots\dots\dots\dots\dots (\mathrm{I}-1.18)$$

Avec : $u \times v \in [\xi_0, \xi_{n+p+1}] \times [\eta_0, \eta_{m+q+1}]$

Sous forme matricielle l'expression devient :

$$S(\xi,\mathfrak{y}) = \left[N_{0,p}(\xi), N_{1,p}(\xi), \dots\dots, N_{0,p}(\xi)\right] \begin{bmatrix} P_{0,0} \dots\dots\dots & P_{0,m} \\ \vdots & \vdots \\ P_{n,0} \dots\dots\dots & P_{n,m} \end{bmatrix} \begin{bmatrix} N_{0,q}(\mathfrak{y}) \\ . \\ . \\ . \\ N_{m,q}(\mathfrak{y}) \end{bmatrix} \dots (I-1.19)$$

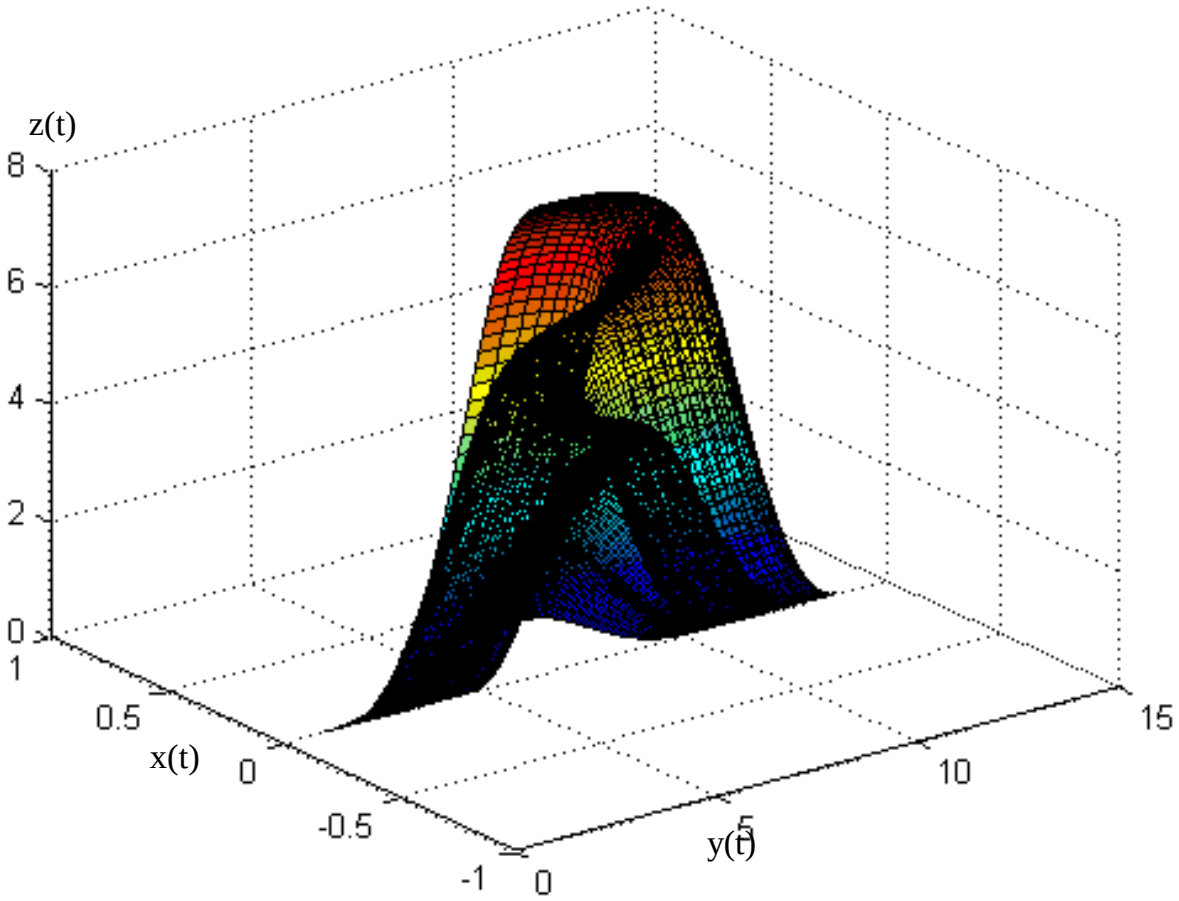


Figure (I- 14) : *surface B-Spline non rationnelle*

I.1.3.10. Propriétés de la surface B-Spline

- non négativité : $N_{i,n}(\xi)N_{j,m}(\eta) \geq 0 \ \ \forall\, i, j, \xi, \eta$
- partition unité : $\sum_{i=0}^{n}\sum_{j=0}^{m} \boldsymbol{N_{i,n}(\xi)N_{j,m}(\eta) = 1} \ \ \forall\, \xi \ et \ \eta$
- enveloppe convexe : la surface $\boldsymbol{S(\xi, \eta)}$ est contenue dans l'enveloppe convexe des points de contrôle.
- Si n>0 et m>0, alors $N_{i,n}(\xi)N_{j,m}(\eta)$ atteint exactement une seule valeur maximale.
- Invariance affine
- Modification locale : si on change les cordonnées d'un point $P_{i,j}$ seulement le rectangle $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+q+1}]$ qui sera modifié.

I.1.4. modèle B-Spline rationnel NURBS

Les NURBS (Non Uniform Rational B-Spline) correspondent à une généralisation des B-Splines car ces fonctions sont définies avec des points en cordonnées homogènes. Le principal intérêt de ces courbes NURBS est qu'elles parviennent même à ajuster des courbes (cercle, conique,…). Une courbe B-spline définie dans R^d (d-dimension) n'est qu'une projection d'une courbe B-Spline définie dans R^{d+1} (d+1 dimension).

I.1.4.1 fonctions de base NURBS

Les fonctions de base B-Spline rationnelle (NURBS) sont définies par la formule mathématique suivante :

$$\boldsymbol{R_{i,m}(\xi) = \frac{w_i N_{i,m}(\xi)}{\sum_{i=1}^{n} w_i N_{i,m}(\xi)}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.20})$$

Avec $\boldsymbol{N_{i,m}(\xi)}$ **:** fonctions de base B-Spline et $\boldsymbol{w_i}$: les poids

I.1.4.2 Propriété des fonctions de bases NURBS

- Non négativité : $\boldsymbol{R_{i,m}(\xi) \geq 0}$ pour toutes valeurs de ***i, m*** et $\boldsymbol{\xi}$
- Partition unité : $\boldsymbol{\sum_{i=1}^{n} R_{i,m}(\xi) = 1}\ \forall\ \xi\epsilon\ [\xi_{min}, \xi_{max}]$
- Si tous les poids $\boldsymbol{w_i}$ sont égaux, les NURBS deviennent des B-Splines.
- Invariance affine.

I.1.4.3 courbe B-Spline Rationnelle (NURBS)

Pour obtenir une courbe dans R^d avec les NURBS, on a les points de contrôles P_i^w(i=1, 2, …n) pour la courbe définie par B-Splines dans R^{d+1} avec le vecteur nœud $\boldsymbol{\Xi}$, les points de contrôles pour la courbe définie par les NURBS sont $(P_i)_j = \frac{(P_i^w)_j}{w_i}, j = 1, \ldots d$ avec $(P_i)_j$ est la j-iéme composante du vecteur $\boldsymbol{P_i}$ et $w_i = (P_i^w)_{d+1}$ sont les poids.

La formule mathématique qui permet de les représenter est définie comme suit : Soit $\boldsymbol{\Xi}$ un vecteur nœud, soit $\boldsymbol{P}$ un polygone de contrôle dans $\mathbf{R}^n$ et $\boldsymbol{w}$ des poids attachés à chaque point de contrôle $\boldsymbol{P_i}$. On suppose que les poids ne sont pas tous nuls. La courbe B-Spline rationnelle (NURBS) de degré m associée est :

$$\boldsymbol{C(\xi) = \sum_{i=1}^{n} P_i R_{i,m}(\xi)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.21})$$

Avec : $\boldsymbol{R_{i,m}(\xi) = \frac{w_i N_{i,m}(\xi)}{\sum_{i=1}^{n} w_i N_{i,m}(\xi)}}$ représente les fonctions NURBS

$\boldsymbol{m}$: est le degré des fonctions $\boldsymbol{N_{i,m}(\xi)}$ Les fonctions de bases B-Spline

$\boldsymbol{P_i}$: Les points de contrôle $\boldsymbol{w_i}$ les poids associés aux pôles.

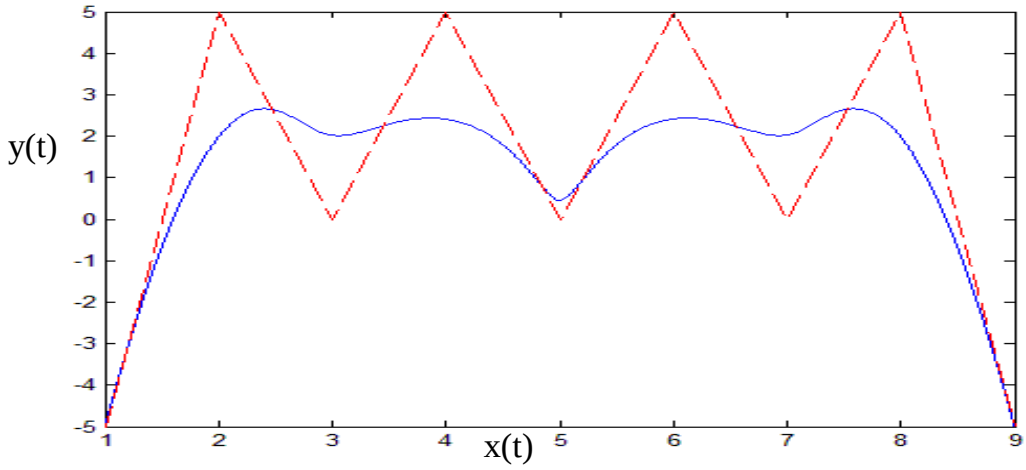


Figure (I.14) : *Courbe Non Uniform rational -Spline (NURBS)*

I.1.4.3.1 Propriétés géométriques des courbes NURBS

- Représentation des coniques : en choisissant correctement les points de contrôles toute forme conique peut être représentée exactement par une NURBS.
- Invariance affine : la transformée affine d'une courbe NURBS est la courbe passant par la transformée des points de contrôle.
- Invariance projective : contrairement aux courbes B-Spline l'image d'une courbe NURBS par une projection est la courbe NURBS passant par la projection des points.

  (Les poids doivent être recalculés en fonction de la matrice de projection)
- Dérivation et continuité : pour tout $\boldsymbol{\xi}$ qui n'est pas une valeur nodale, $\boldsymbol{C(\xi)}$ est infiniment dérivable, si $\boldsymbol{\xi}$ est égale à une valeur nodale de multiplicité *k* alors $\boldsymbol{C(\xi)}$ est *(m-k)* fois dérivable ($\boldsymbol{C(\xi)}$ n'est pas nécessairement continue).
- Les courbes NURBS peuvent représenter exactement toutes les formes coniques usuelles de conception
- L'influence de la variation des poids $\boldsymbol{W_i}$ sur les pôles $\boldsymbol{P_i}$ est analogue à celle de Bézier rationnelle sauf que la variation dans le cas de NURBS est locale.

I.1.4.3.2 Construction d'un cercle par le modèle NURBS (B-Spline rationnel)

Les courbes NURBS dans R^d sont obtenues par une transformation projective des courbes B-Spline représentée dans R^{d+1}, particulièrement les sections coniques comme les cercles et les ellipses peuvent être construites d'une manière exacte ; par une transformation projective de la courbe quadratique B-Spline, la figure (I-15) représente un cercle dans R^2 construit à partir de la courbe B-Spline quadratique dans R^3. La transformation est faite par la projection de chaque point de la courbe sur le plan (z=1) [3]

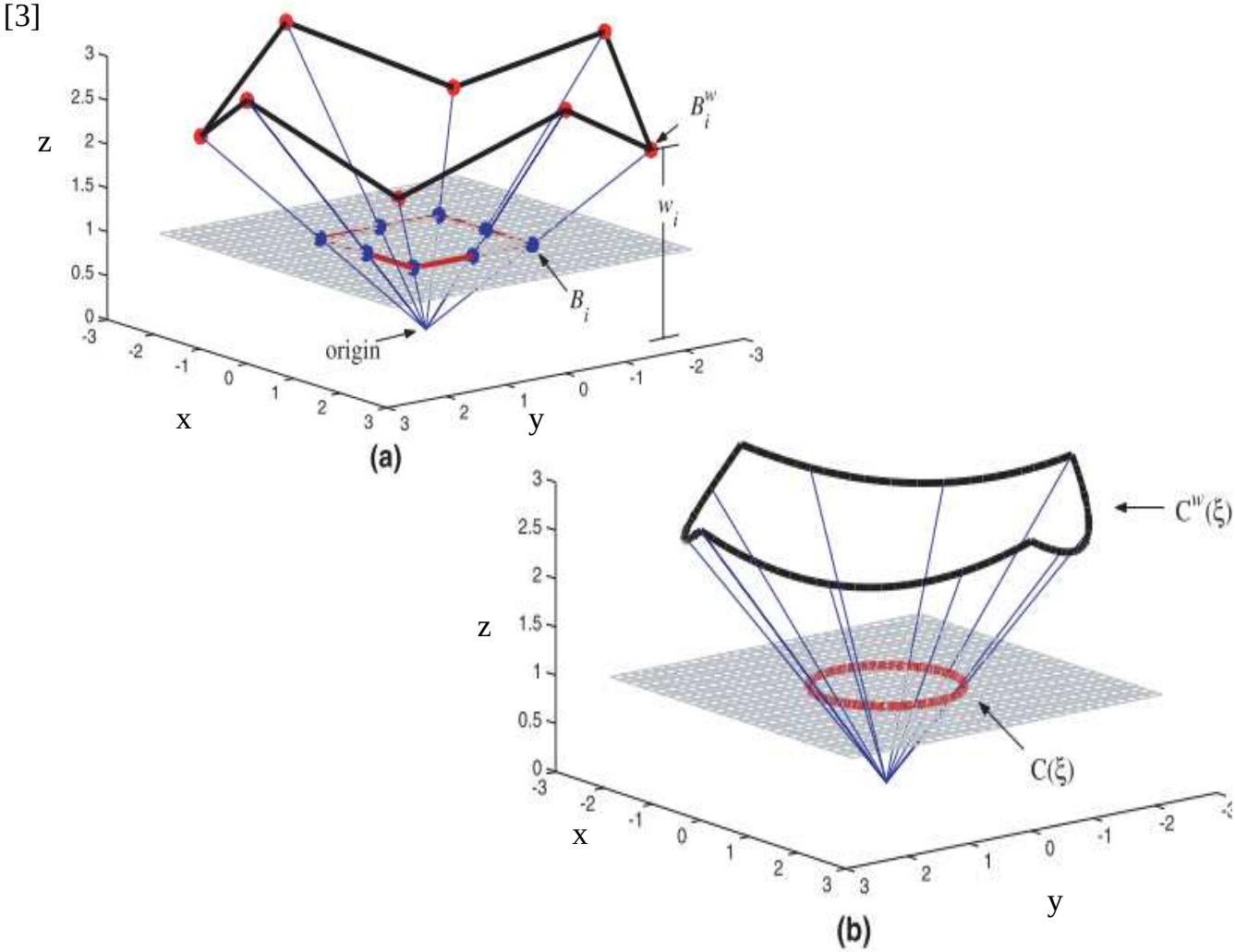


Figure (I-15) : *cercle construit dans R2 par une transformation projective des fonctions de base B-Spline. (a)transformation projective des points de contrôle, les poids w_i sont les z-composantes de B_i^w . (b) transformation projective de la courbe B-Spline $C^w(\xi)$à la courbe NURBS $C(\xi)$*

I.1.4.5 Dérivées des fonctions de bases NURBS

$$\frac{d}{dt}\boldsymbol{R_{i,m}(\xi) = w_i}\ \frac{\boldsymbol{W(\xi)N'_{i,m}(\xi) - W'(\xi)N_{i,m}(\xi)}}{\boldsymbol{(W(t))^2}}\ \textbf{.....}\ \textbf{(I-1.22)}$$

Avec : $W(\xi) = \sum_{i=1}^{n} N_{i,m}(\xi)w(i)$ et $W'(\xi) = \sum_{i=1}^{n} N'_{i,m}(\xi)w(i)$

$$N'_{i,m}(\xi) = \frac{d}{dt}N_{i,m}(\xi) = \frac{m}{t_{i+m}-t_i}N_{i,m-1}(\xi) - \frac{m}{t_{i+m+1}-t_{i+1}}N_{i+1,m-1}(\xi)$$

D'une manière générale pour *k-éme* dérivée on a la formule générale suivante :

$$\frac{d^k}{dt^k}R_{i,m}(\xi) = \frac{A_i^{(k)}(\xi) - \sum_{j=1}^{k}\binom{k}{j}W^j(\xi)\frac{d^{k-j}}{dt^{k-j}}R_{i,m}(\xi)}{W(\xi)}\ \text{.....}\ \textbf{(I-1.23)}$$

Avec : $W^j(\xi) = \frac{d^k}{dt^k}W(\xi)$ $A_i^{(k)}(\xi) = w_i\frac{d^k}{dt^k}N_{i,m}(\xi)$

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

Dérivée *k-éme* des fonctions de base B-spline est donnée par :

$$\frac{d^k}{dt^k}N_{i,m}(\xi) = \frac{m}{t_{i+m}-t_i}\left(\frac{d^{k-1}}{dt^{k-1}}N_{i,m-1}(\xi)\right) - \frac{m}{t_{i+m+1}-t_{i+1}}\left(\frac{d^{k-1}}{dt^{k-1}}N_{i+1,m-1}(\xi)\right)\ \text{.....}\ \textbf{(I-1.24)}$$

Les fonctions de base NURBS sont des fonctions liées aux fonctions de base B-Spline, donc leurs drivées dépendent des dérivées des fonctions de base B-Spline. La figure qui suit montre les dérivées NURBS et celle des B-Spline. Sur la figure on trouve.

a)- fonctions de base B-Spline, b)- leurs 1^{ére} dérivées, c)-leurs 2^{iéme} dérivées, e)- fonctions de base NURBS, f)- leurs 1^{ére} dérivées, g)-leurs 2^{iéme} dérivées.

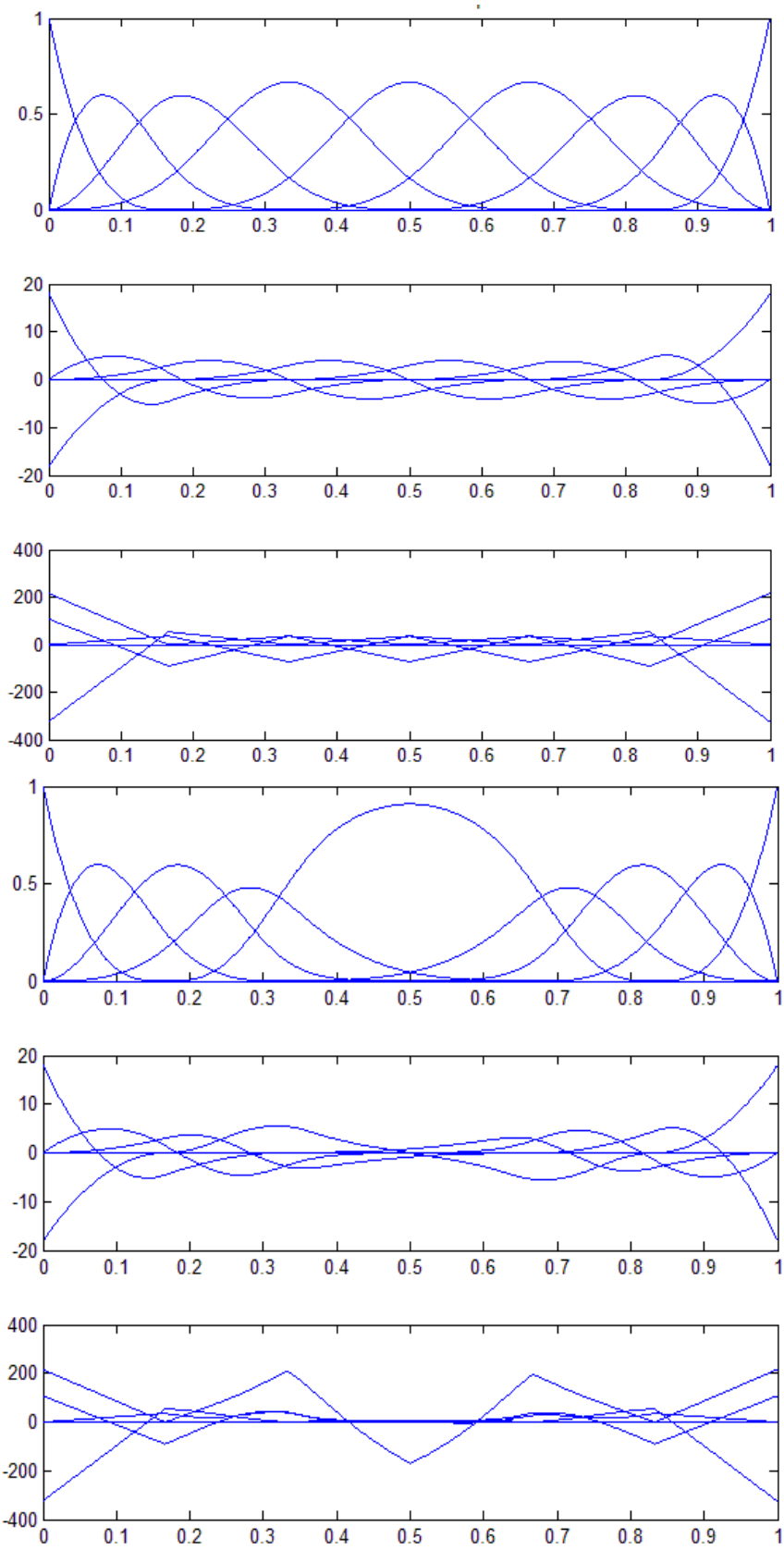


Figure (III-16) : 1ére et 2éme *dérivées fonctions de bases B-Spline et NURBS*

I.1.4.6 Surfaces NURBS

une surface NURBS de degré (*p,q)* est définie par :

$$\boldsymbol{S(\xi,\eta)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(\xi)N_{j,q}(\eta)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(\xi)N_{j,q}(\eta)w_{i,j}}} \qquad \boldsymbol{0\le\xi,\eta\le 1 \ldots.(I-1.25)}$$

$\{P_{i,j}\}$:points de contrôles bidirectionnels ,$\{w_{i,j}\}$: les poids et $\{N_{i,p}(\xi)\}\ et\{N_{j,q}(\eta)\}$ sont les fonctions de base B-Spline

On pose $\boldsymbol{R_{i,j}(\xi,\eta)=\frac{N_{i,p}(\xi)N_{j,q}(\eta)w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(\xi)N_{j,q}(\eta)w_{i,j}}}$: fonctions de bases NURBS

Alors l'équation précédente devient

$$\boldsymbol{S(\xi,\eta)=\sum_{i=0}^{n}\sum_{j=0}^{m}R_{i,j}(\xi,\eta)P_{i,j}}$$

La figure (I-17) montre une surface NURBS

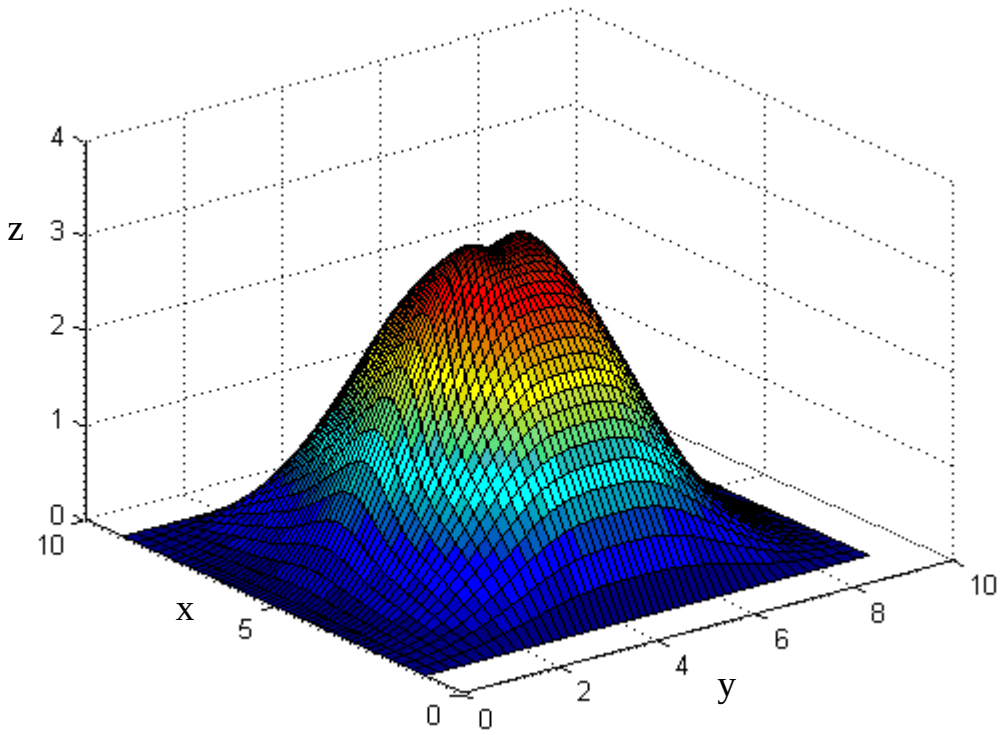


Figure (I-17) : *surface NURBS de degré 2*

I.1.4.7 Propriété des surfaces NURBS

- Non négativité : $\boldsymbol{R_{i,j}(\xi, \eta) \geq 0 \; \forall \; i, j, \xi \; et \; \eta}$
- Partition unité : $\sum_{i=0}^{n} \sum_{j=0}^{m} \boldsymbol{R_{i,j}(\xi, \eta) = 1 \; \forall (\xi, \eta) \; \epsilon \; [0.1] \times [0.1]}$
- Si $p > 0 \; et \; q > 0$ alors $\boldsymbol{R_{i,j}(\xi, \eta)}$ atteint exactement une seul valeur maximal
- $\boldsymbol{R_{0,0}(0,0) = R_{n,0}(1,0) = R_{0,m}(0,1) = R_{n,m}(1,1) = 1}$
- Invariance affine
- Enveloppe convexe
- Modification locale : si on déplace le point $P_{i,j}$, ou on change la valeur de $w_{i,j}$ alors seulement le rectangle $\boldsymbol{\left[\xi_i, \xi_{i+p+1}\right] \times \left[\eta_j, \eta_{j+q+1}\right]}$ qui sera modifié

I.1.4.8 Modélisation et construction des surfaces particulières par les NURBS [2]

I.1.4.8.1 Surfaces bilinéaire

Soit $P_{0.0}, P_{1.0}, P_{0.1}, P_{1.1}$ quatre points de l'espace, la surface NURBS bilinéaire représentée par les quatre points est obtenue par une interpolation bilinéaire entre les quatre segments $P_{0.0}\, P_{1.0}, P_{0.1} P_{1.1}, P_{0.0}\; P_{0.1}, et\; P_{1.0}\, P_{1.1}$. La surface NURBS bilinéaire est obtenue par l'équation suivante :

$$\boldsymbol{S(\xi, \eta) = \sum_{i=0}^{1} \sum_{j=0}^{1} R_{i,1}(\xi) R_{j,1}(\eta) P_{i,j}} \quad \textbf{..... (I-1.26)}$$ avec $\boldsymbol{\Xi}$=$\boldsymbol{\mathcal{H}}$=[0 0 1 1]

L'équation (I-1.26) présente une simple interpolation linéaire entre les segments opposés dans les deux directions (ξ, η). Par contre on peut construire une surface bilinéaire par une interpolation linéaire entre les segments non parallèles (diagonaux) d'un cube .La figure suivante montre les deux cas :

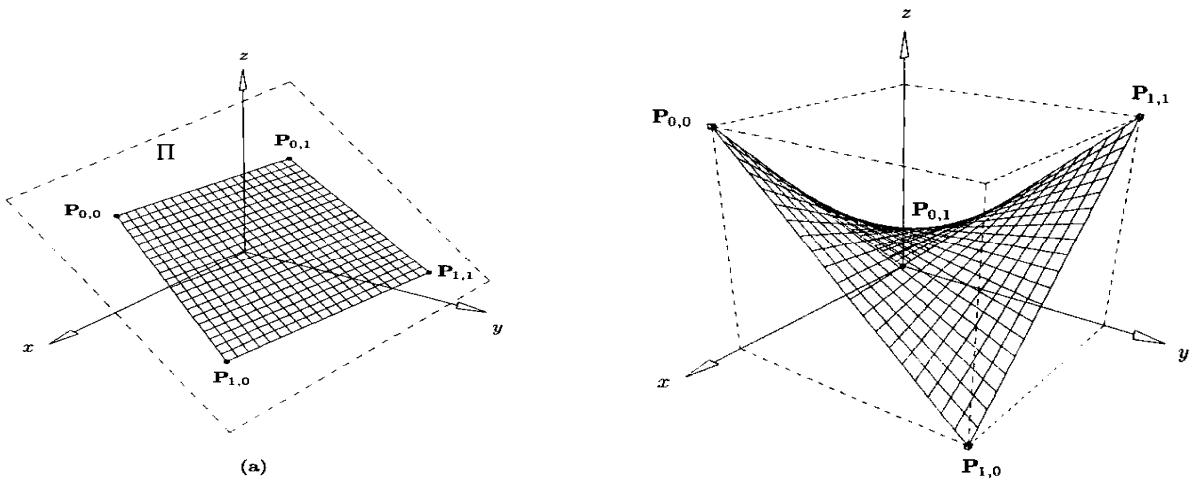


Figure (I-18) : *a)-surface bilinéaire définie par des segments opposés sur un plan (π) b)-surface bilinéaire définie par des segments non parallèles d'un cube.*

I.1.4.8.2 Construction d'une surface réglée

Supposons que nous avons deux courbes NURBS

$$C_k(\xi) = \sum_{i=0}^{n_k} R_{i,p_k}(\xi) P_i^k \quad avec\ k = 1,2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.27})$$

On définit le vecteur nœud $\Xi^{\,k} = \{\xi_0^k, \xi_1^k, \ldots\ldots\ldots, \xi_{m_k}^k\}$. On veut construire une surface réglée dans la direction ŋ (une interpolation linéaire entre $C_1(\xi)\ et\ C_2(\xi)$).

En outre, nous avons besoin que l'interpolation soit entre les points dont les paramètres sont égaux, i.e. pour une valeur fixe de $\bar{\xi}$ la surface $\boldsymbol{S}(\bar{\xi}, \eta)$ et la droite liant les points $C_1(\bar{\xi})$et $C_2(\bar{\xi})$ .la surface réglée avec les NURBS est donnée par la formule suivante :

$$\boldsymbol{S(\xi,\eta) = \sum_{i=0}^{n}\sum_{j=0}^{1} R_{i,p,j,1}(\xi,\eta) P_{i,j}} \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{I-1.28})$$

Avec $\boldsymbol{R_{i,p,j,1}(\xi,\eta) = R_{i,p}(\xi) R_{j,1}(\eta)}$ **:** fonction de base NURBS

Et $\boldsymbol{P_{i,j}}$: pôles du réseau caractéristique.

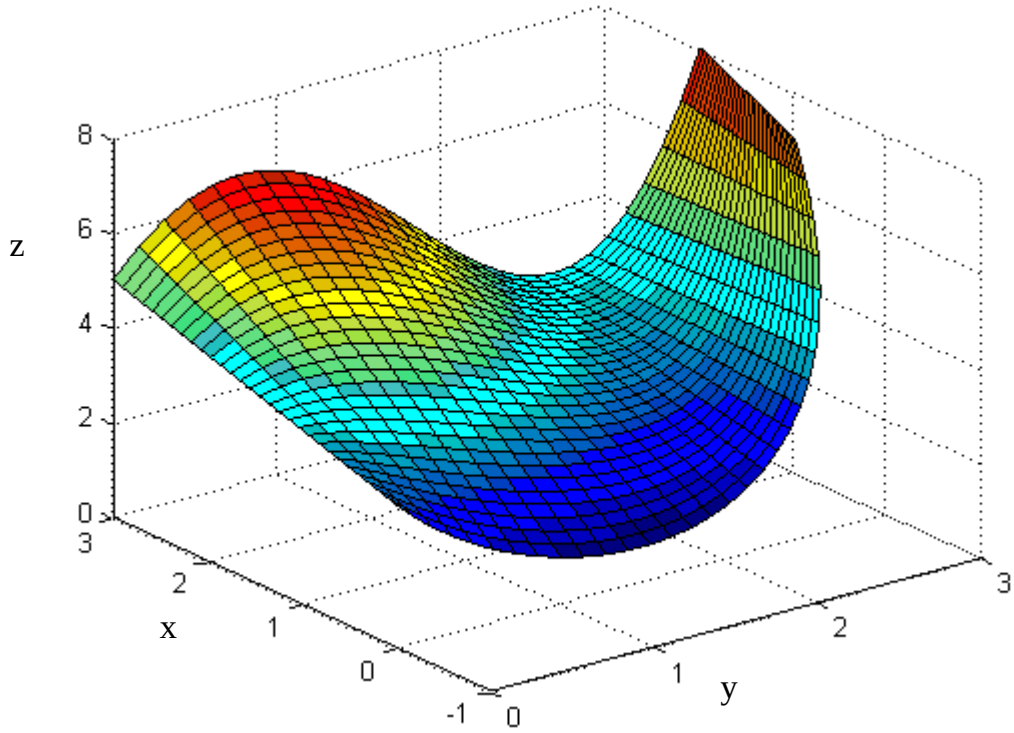


Figure (I-19) : *surface réglées NURBS*

I.1.4.8.3 Construction d'une surface d'extrusion

Une surface d'extrusion est générée par une courbe généralement plane glissant le long d'une trajectoire rectiligne ou non. Etant donné un vecteur **B** de longueur unité et $C(\xi) = \sum_{i=0}^{n} R_{i,p}\,(\xi) P_i$ une courbe NURBS de degré p avec le vecteur nœud $\boldsymbol{\Xi}$ et les poids w_i. La surface d'extrusion S (ξ,ŋ) est obtenue par translation de la courbe $C(\xi)$ le long du vecteur **B.**

La surface d'extrusion est obtenue par la formule suivante :

$$\boldsymbol{S(\xi,\eta) = \sum_{i=0}^{n}\sum_{j=0}^{1} R_{i,p,j,1}(\xi,\eta) P_{i,j}} \ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{(I-1.29)}$$

Avec $\boldsymbol{R_{i,p,j,1}(\xi,\eta) = R_{i,p}(\xi) R_{j,1}(\eta)}$ **:** fonction de base NURBS

Et $\boldsymbol{P_{i,j}}$: pôles de réseau caractéristique

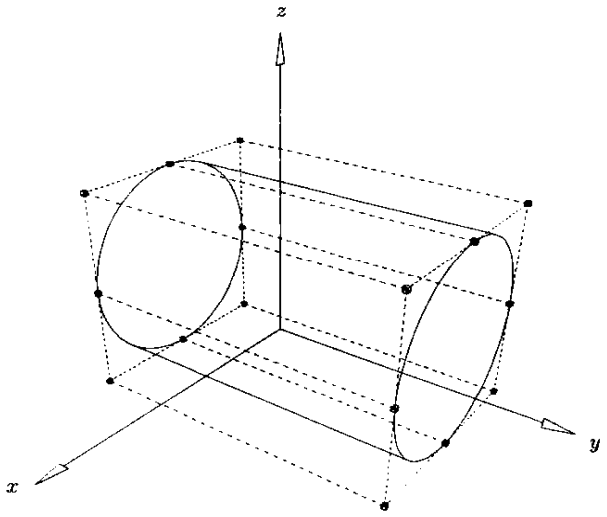


Figure (I-19) : *principe de création d'une surface d'extrusion*

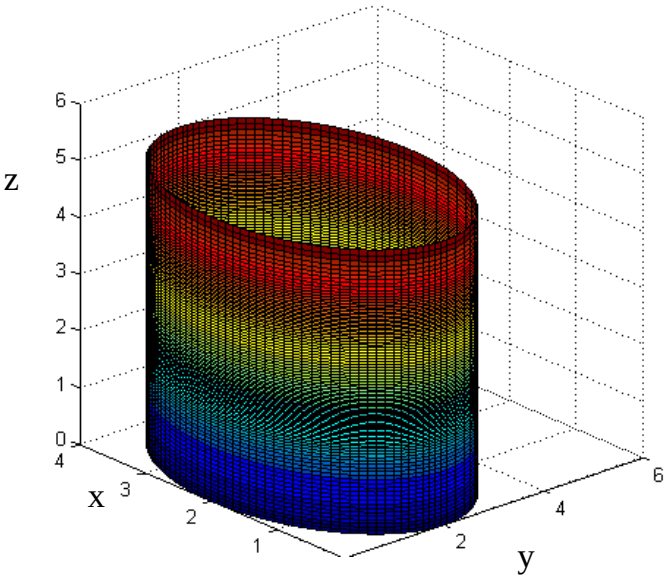


Figure (I-20) : *surface d'extrusion NURBS*

I.1.4.8.4 Construction d'une surface de révolution [2]

Une surface de révolution est une surface possédant un axe de symétrie, Pour modéliser ce genre d'objets, il suffit de construire un profil $\boldsymbol{C(\xi)}$ puis de faire tourner ce profil autour d'un axe. La figure suivante montre le principe.

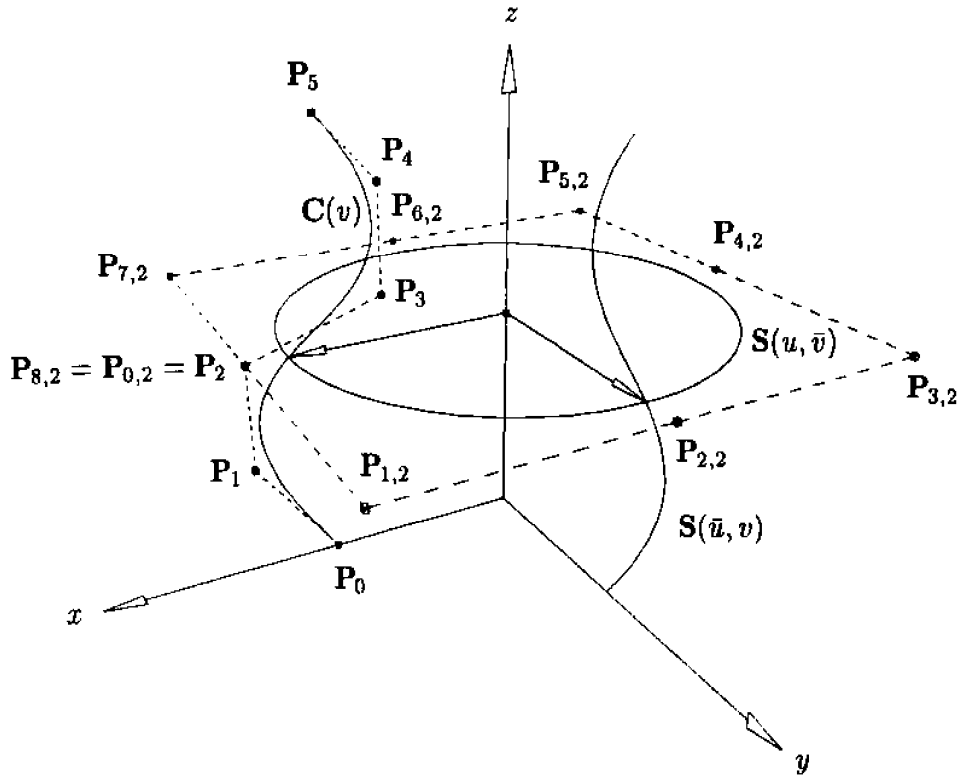


Figure (I-21) : *définition d'une surface de révolution*

$\boldsymbol{C(\xi) = \sum_{i=1}^{n} P_i R_{i,m}(\xi)}$, est la courbe NURBS de degré *m* $\boldsymbol{C(\xi)}$ est appelée génératrice. Cependant, la courbe $\boldsymbol{C(\xi)}$ se trouve dans le plan XZ et que nous tournons cette courbe de 360° autour de l'axe Z, la surface $\boldsymbol{S(\xi, \eta)}$ et telle que représentée sur la figure (I-22)

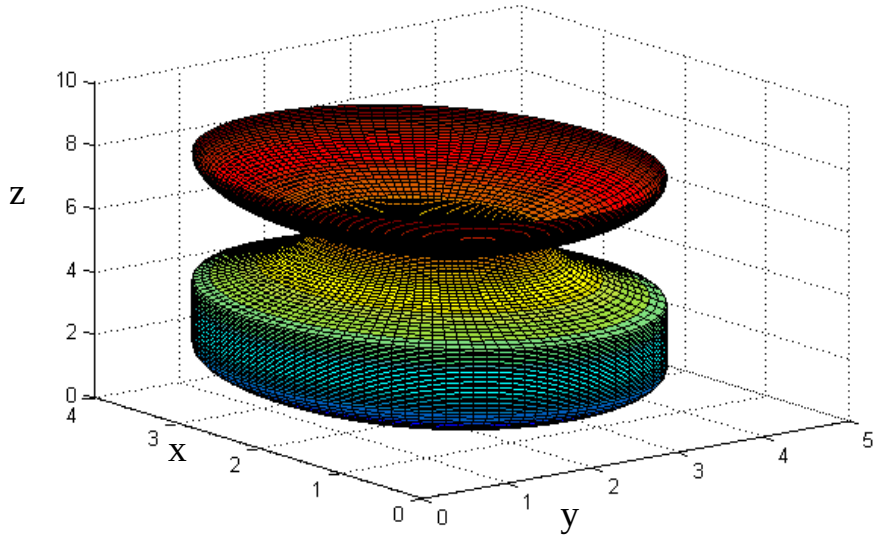


Figure (I-22) : *surface de révolution NURBS*

I.2. Raffinement

L'application du processus de raffinement dans un contexte isogeometrique a suscité beaucoup d'attention [4] où on constate que le raffinement peut être effectué dans hpk-espace - une caractéristique unique des méthodes isogéométriques. Le h et p-raffinement sont analogues au h-raffinement et p-raffinement traditionnels dans MEF, mais le k-raffinement est spécial à l'analyse isogéométrique [4]. On se base sur le fait que les processus de h-raffinement et p-raffinement dans des méthodes isogéométriques ne permutent pas, par contre, dans le cas de k-raffinement, d'abord l'ordre des fonctions de base est augmenté (p-raffinement) et puis les valeurs des nœuds sont ajoutés pour créer de nouveaux éléments (h-raffinement).

I.2.1. h-raffinement

Le h-raffinement dans l'analyse isogéométrique est basé sur l'insertion de nœud. Etant donnée le vecteur nœud $\Xi = \{\xi_1, \xi_2, \dots \xi_{n+p+1}\}$, on appelle $\bar{\xi}$ le nouveau nœud tel que $\bar{\xi} \in [\xi_i, \xi_{i+1}]$.les (n+1) nouvelles fonctions de base sont obtenues en utilisant les équations (**I-1.14)** et **(I-1.15)** avec le nouveau vecteur nœud $\bar{\Xi} = \{\xi_1, \xi_2, \dots, \xi_k, \bar{\xi}, \xi_{k+1}, \dots \xi_{n+p+}\}$ [4]

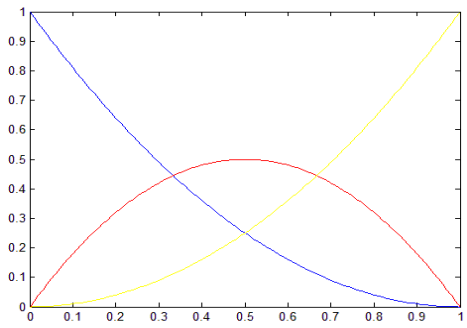
Définissant les points de contrôle de la courbe originale $\boldsymbol{P} = \{\boldsymbol{p_1}, \boldsymbol{p_2} \dots \dots, \boldsymbol{p_n}\}$

Le nouvel ensemble de points de contrôle $\bar{\boldsymbol{P}} = \{\bar{\boldsymbol{p}}_1, \bar{\boldsymbol{p}}_2, \dots \dots \bar{\boldsymbol{p}}_n\}$ sont obtenus avec

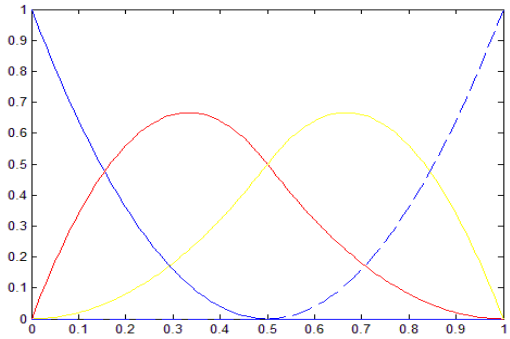
$$\bar{\boldsymbol{p}}_i = \boldsymbol{\alpha}_i \boldsymbol{p}_i + (\mathbf{1} - \boldsymbol{\alpha}_i)\boldsymbol{p}_{i-1} \quad \text{..... (I-2.1)}$$

$$\text{Où} \begin{cases} \mathbf{1} & \boldsymbol{si} \quad \mathbf{1} \le \boldsymbol{i} \le \boldsymbol{k} - \boldsymbol{p} \\ (\boldsymbol{\tau} - \boldsymbol{t_i})/(\boldsymbol{t_{i+p}} - \boldsymbol{t_i} & \boldsymbol{si} \quad \boldsymbol{k} - \boldsymbol{p} + \mathbf{1} \le \boldsymbol{i} \le \boldsymbol{k} \\ \mathbf{0} & \boldsymbol{si} \quad \boldsymbol{k} + \mathbf{1} \le \boldsymbol{i} \le \boldsymbol{n} + \boldsymbol{p} + \mathbf{2} \end{cases} \quad \text{.... (I-2.2)}$$

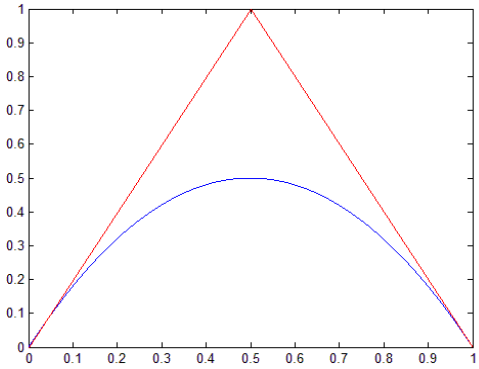
Des valeurs de nœud déjà actuelles dans le vecteur nœud peuvent être répétées. Cependant, ceci réduit la continuité de la base au nœud correspondant. La continuité de la courbe est préservée en choisissant les points de contrôle en utilisant les équations (I-5 et I-6). La figure (I-4) illustre une courbe raffinée avec l'insertion de nœud



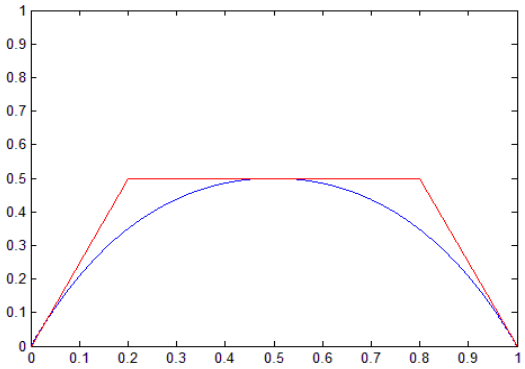
a)- fonctions de base originales



b)- nouvelles fonctions de base

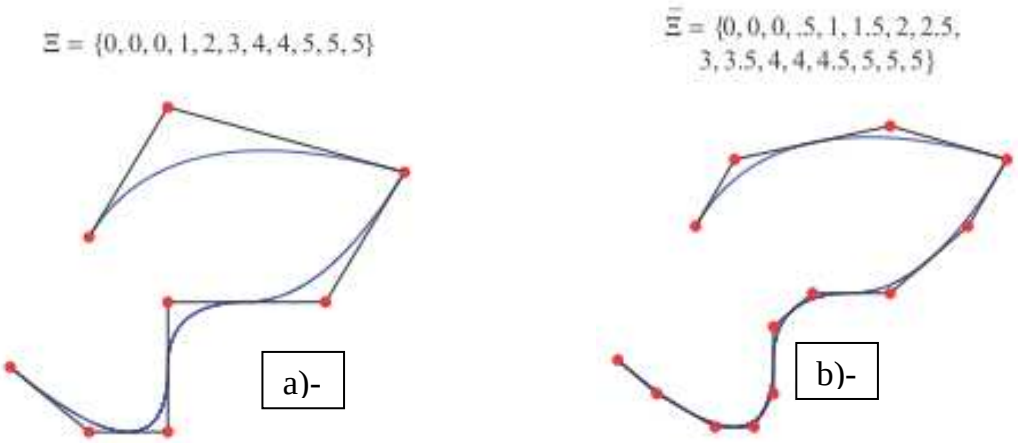


c-) courbe originale avec Ξ*= [0 0 0 1 1 1]*



d)- courbe raffinée avec Ξ*= [0 0 0 0.5 1 1 1]*

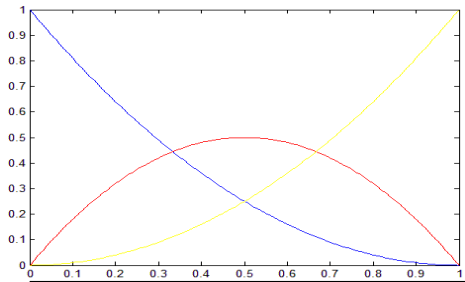
Figure(I-23) : *le concept de raffinement par insertion de nœud (h-raffinement)*



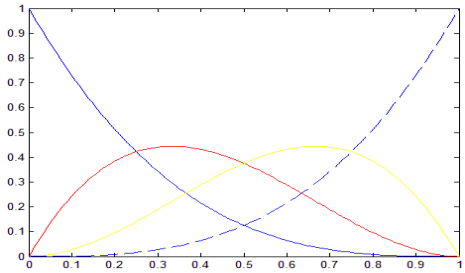
Figure(I-24) :*un exemple illustrant le concept de h-raffinement a)- courbe originale b-) courbe raffinée avec h-raffinement*

I.2.2. p-raffinement

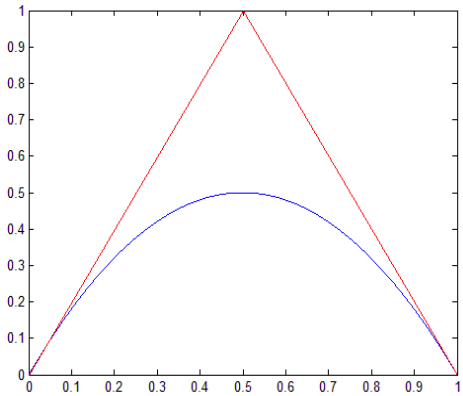
Le mécanisme pour mettre en application le p-raffinement *est l'élévation d'ordre*. L'ordre du polynôme des fonctions de base peut être augmenté sans changer la courbe paramétriquement ou géométriquement. Noter que chaque valeur unique de nœud dans Ξ doit être répétée afin de préserver des discontinuités dans la *n-éme* dérivée de la courbe. Le nombre de nouveaux points de contrôle dépend des multiplicités des nœuds existants. Comme l'insertion de nœud, l'espace parcouru par la base élevée contient l'espace parcouru par la base originale, ainsi on peut augmenter l'ordre sans changer la géométrie de la courbe B-spline [3].La figure suivante montre un exemple de p-raffinement



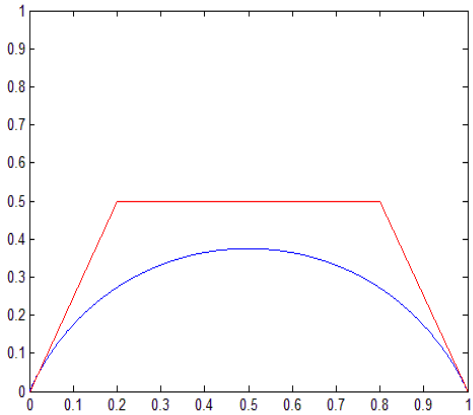
a)- fonctions de base originales



b)- nouvelles fonctions de base



c-) courbe originale avec Ξ*= [0 0 0 1 1 1]*



d)- courbe raffinée avec Ξ*= [0 0 0 0 1 1 1 1]*

Figure(I-25) : *le concept de d'élévation p-raffinement*

Figure (I-26) *un exemple illustrant le concept de p-raffinement a)- courbe originale b-) courbe raffinée avec p-raffinement*

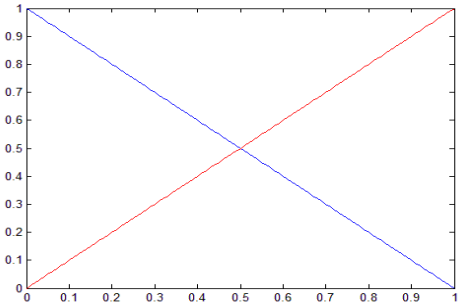
I.2.3. k-raffinement

k-raffinement : *élévation d'ordre et une continuité élevée*

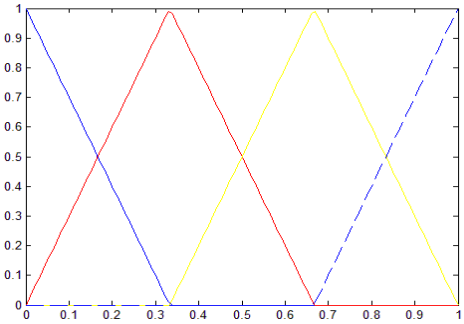
Si on insert une valeur de nœud $\bar{\xi}$ entre deux nœuds dans la courbe d'ordre *m*, les fonctions de base auront un nombre de *m-1* dérivées continues au nœud $\bar{\xi}$, et si nous élevons plus tard l'ordre des fonctions de base à *q* la multiplicité de chaque nœud augmente copris celui inséré et le nombre de dérivées continues ne change pas (discontinuité à la $n^{ième}$ dérivée). Mais si on commence par une élévation d'ordre des fonctions de base à *q*, ensuite on insert la valeur du nœud $\bar{\xi}$, cette fois-ci les fonctions de base auront *q-1* dérivées continues au nœud $\bar{\xi}$, ce procédé est appelé *k-raffinement.* [3][4]

Le concept du k-raffinement est très important parce que l'analyse isogéométrique est fondamentalement une approche évoluée. Dans le p-raffinement traditionnel il y a une non homogénéité dans la structure dûe à la différence des fonctions de base liées à la surface, au sommet, et aux nœuds intérieurs. En outre, la continuité C^0 maintenue pendant le processus de raffinement implique une multiplication dans le nombre de nœuds. Dans le k-raffinement, il y a une homogénéité de la structure et la croissance du nombre de variables de contrôles est limitée [*3*].

.

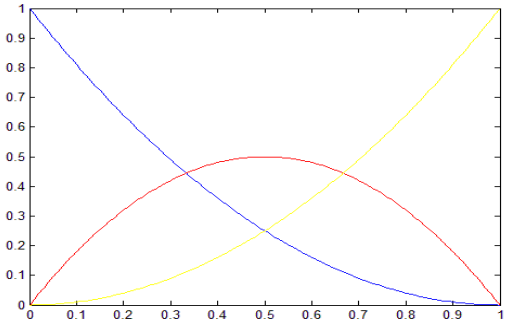


Deux fonctions de bases de C^0 *avec T= [0 0 1 1]*

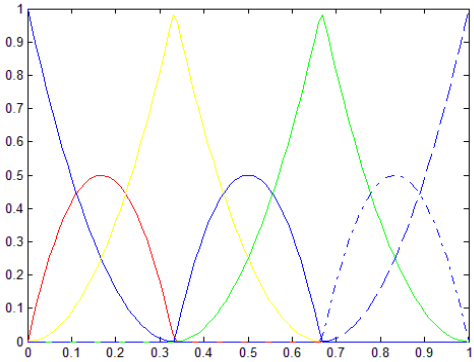


Insertion de nœud $\bar{\xi}$ ***=1/3,2/3***
T= [0 0 1/3 2/3 1 1] et m=1

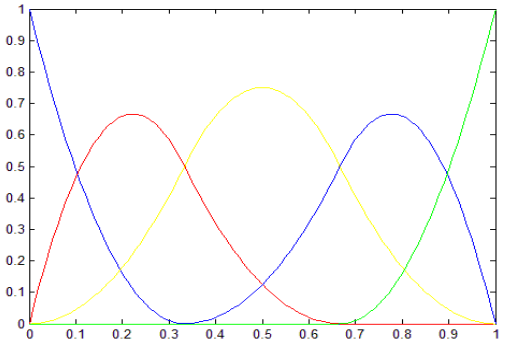
a-) **b-)**



Elévation d'ordre m=1 m=2
T= [0 0 0 1 1 1] m=2



Elévation d'ordre m=1 m=2
T=[0 0 01/3 1/3 2/3 2/3 1 1 1]
m=2 sept fonctions de base quadratique de C^0



Insertion de nœud $\bar{\xi}$ ***= 1/3, 2/3***
T=[0 0 1/3 2/3 1 1] et m=2
cinq fonctions de base quadratique de C^1

Figure (I-27) : *le concept de k-raffinement.*

Un exemple de k-raffinement et une comparaison avec le traditionnel p-raffinement est donné dans la figure (I.27), cet exemple montre que le k-raffinement produit peu de fonctions et d'une continuité plus élevée, ainsi peu de variables de contrôle ou degrés de liberté. En commençant avec ***p+1*** fonctions de base, insérant ***n-(p+1)*** nœuds (cela afin d'obtenir ***n*** fonctions de base), suivi d'une élévation d'ordre à ***r***, nous obtenons ***(r+1)n-rp*** fonctions de base de continuité C^{p-1}. En utilisant le k-raffinage et commençant avec ***p+1***fonctions, appliquant un ordre d'élévation ***r***, suivi d'insertion de ***n-(p+1)*** nœuds, nous obtenons **n+r** fonctions de base de continuité C^{r+p-1}. Il est clair que le k-raffinement produit moins d'inconnues que le p-raffinement, et cela pour un même maillage et un même ordre d'approximation.

Dans la méthode du k-raffinement pour un maillage fixe, des nœuds sont ajoutés aux valeurs limites, croissant leur multiplicité, mais pas de nœuds intérieurs ajoutés.

II.1 Rappel sur la méthode des éléments finis (MEF) appliquée en élasticité linéaire

II.1.1 Principe

La méthode des éléments finis (MEF) ou l'analyse aux éléments finis (AEF) est une méthode numérique de résolution généralement pour les problèmes dont la géométrie est complexe où les solutions analytiques ne peuvent pas êtres obtenues. Le concept est de construire des objets de forme complexe en utilisant des éléments simples. Obtenir une solution approximative aux problèmes à valeur finie ; en outre, discrétiser un objet de forme complexe en éléments plus petits. Les deux figures suivantes (figure (II.1.1) et la figure (II.1.2) récapitulent le principe [5].

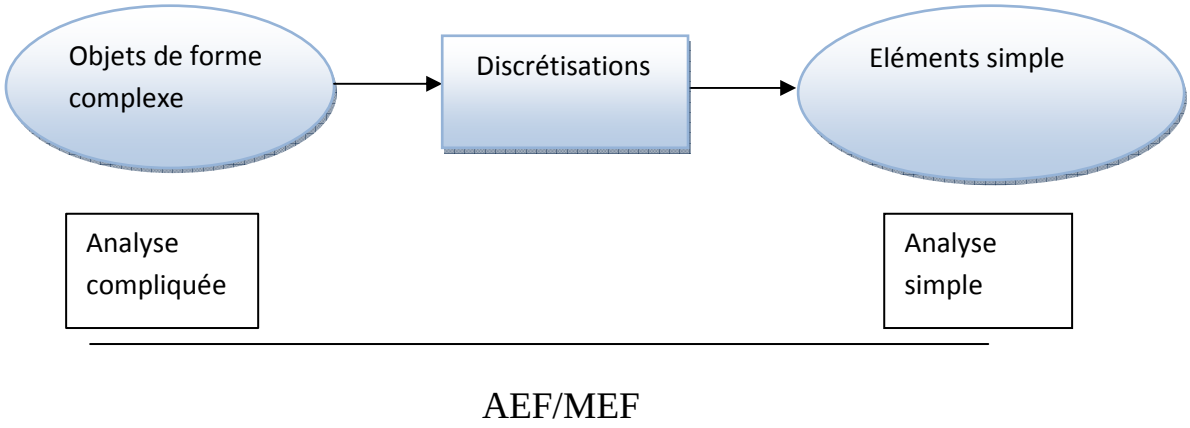


Figure (2.1.1) : *concept de la méthode des éléments finis*

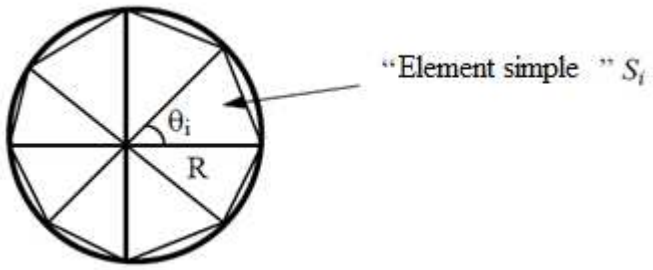


Figure (2.1.2) : *subdivision d'un cercle en triangles élémentaires afin d'approximer sa surface.*

- ✓ Surface d'un triangle : $s_i = \frac{1}{2}R^2 \sin\theta_i$
- ✓ Surface du cercle : $S_N = \sum_1^N s_i = \frac{1}{2}R^2 N \sin(\frac{2\pi}{N}) \rightarrow S = \pi R^2 \quad car\ N \rightarrow \infty.$
  Où N est le nombre total de triangles.

II.1.2 Procédure générale pour AEF

- Discrétiser la structure en éléments délimités par des nœuds
- Choix des fonctions d'interpolation
- Décrire la forme et le comportement de chaque quantité physique en chaque élément.
- Assembler les éléments pour avoir un système d'équation global.
- Appliquer les conditions aux limites.
- Résoudre le système d'équation
- Calculer les quantités désirées
- Interprétation des résultats.

Elément barre

Considérons l'élément barre sur la figure (2.1.3) :

- ***L*** : longueur de l'élément.
- ***A*** : surface de la section droite.
- ***E*** : module d'élasticité de Young.
- ***u=u(x)*** : déplacement.
- ***ε=ε(x)*** : déformation.
- ***σ=σ(x)*** : contrainte.

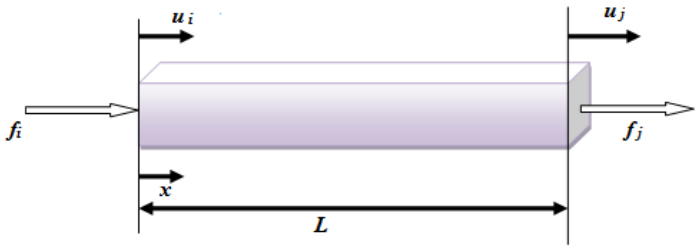


Figure (2.1.3) : *élément barre en traction*

Relation déplacements-déformations :

$$\varepsilon = \frac{du}{dx} \qquad (II.1.1)$$

Relation contraintes-déformations :

$$\sigma = E\varepsilon \qquad (II.1.2)$$

II.1.2.1 Matrice de rigidité (méthode directe)

considérant la barre comme étant un seul élément $dx = L$, l'équation (II.1.1) devient :

$$\varepsilon = \frac{u_i - u_j}{L} \qquad (II.1.3)$$

$$\sigma = E\varepsilon = \frac{E(u_i - u_j)}{L} \qquad (II.1.4)$$

Nous avons aussi $\sigma = \frac{F}{A}$ $(II.1.5)$

Alors : $\frac{F}{A} = \frac{E(u_i - u_j)}{L}$ $(II.1.6)$

Finalement on peut écrire le système d'équations suivant :

$$\begin{cases} \frac{AE}{L}(u_i - u_j) = f_i \\ \frac{AE}{L}(-u_i + u_j) = f_j \end{cases} \qquad (II.1.7)$$

Ce système peut s'écrire sous forme matricielle comme suit :

$$\begin{pmatrix} f_i \\ f_j \end{pmatrix} = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{pmatrix} u_i \\ u_j \end{pmatrix} \qquad (II.1.8)$$

Ou bien:

$$\boldsymbol{F = KU} \qquad (II.1.9)$$

Où :

$\boldsymbol{K}$= $\frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ est la matrice de rigidité élémentaire.

$\boldsymbol{U}$=$\begin{pmatrix} u_i \\ u_j \end{pmatrix}$ est le vecteur des déplacements nodaux.

$\boldsymbol{F}$= $\begin{pmatrix} f_i \\ f_j \end{pmatrix}$ est le vecteur des forces nodales.

Cette forme nous permet de combiner facilement les équations de tous les éléments d'une structure.

Degrés de liberté *(ddl)*

Le nombre de degrés de liberté *(ddl)* : c'est le nombre de composantes du vecteur des déplacements pour chaque nœud. Pour un élément barre sollicité en traction, le nombre de *ddl* par nœud est égal à un.

II.1.2.2 Matrice de rigidité (une approche plus formelle)

Nous avons montré comment obtenir les équations élémentaires de rigidité pour un élément barre en utilisant la méthode directe. Nous pouvons également obtenir ces équations par un procédé plus général et plus formel. Qui peut être appliqué à beaucoup d'autres situations plus compliquées, aussi, cela nous rapprochera plus du concept isoparamétrique.

On a les fonctions de forme suivantes :

$$N_i(\xi) = 1 - \xi \qquad N_j(\xi) = \xi \qquad (II.1.10)$$

Où :

$$\xi = \frac{x}{L} \qquad 0 \leq \xi \leq 1 \qquad (II.1.11)$$

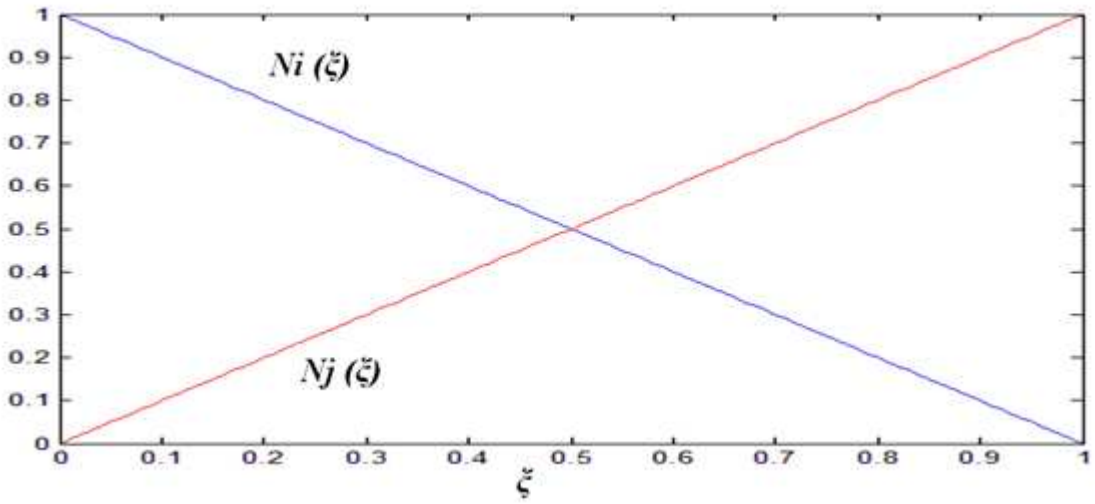


Figure (2.1.4) : *Allure des fonctions de forme linéaire en MEF*

Le champ des déplacements pour un élément barre de longueur L. assumant que le déplacement ***u*** varie linéairement suivant l'axe de la barre

$$u(x) = u(\xi) = N_i(\xi)u_i + N_j(\xi)u_j \qquad (II.1.12)$$

$$u = \begin{bmatrix} \frac{L-x}{L} & \frac{x}{L} \end{bmatrix} \begin{pmatrix} u_i \\ u_j \end{pmatrix} = \mathrm{Nd} \qquad (II.1.13)$$

N est appelé matrice de fonction de forme

La contrainte axiale est donnée par :

$$\varepsilon = \frac{du}{dx} = \left[\frac{d}{dx} N\right] \mathbf{d} = \mathbf{Bd} \qquad (II.1.14)$$

Donc $\mathbf{B} = \left[\frac{d}{dx} N\right] =$ **B**= [*-1/L 1/L*] $(II.1.15)$

Pour la plupart des éléments une formule générale est employée pour calculer la matrice de rigidité K.

$$\mathbf{K} = \left[\int_V (\mathbf{B}^{\mathbf{T}} E \mathbf{B}) dV\right] \qquad (II.1.16)$$

Avec $dV = Adx$ et **B**= [*-1/L 1/L*] (II.1.17)

Cette expression peut être déduite en utilisant des différentes méthodes telles que le principe de l'énergie potentielle minimale ou la méthode de Galerkin.

A partir de (II.1.16) en peut écrire : $\mathbf{K} = \int_0^L \begin{pmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{pmatrix} E \begin{pmatrix} -\frac{1}{L} & \frac{1}{L} \end{pmatrix} A\, dx \qquad (II.1.18)$

D'où $\mathbf{K} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (II.1.19)$

Le même résultat en utilisant la méthode directe

II.1.3 Analyse linéaire en statique

La plupart des problèmes d'analyse de structure, peuvent être traités comme étant des problèmes linéaires statiques, se basant sur les hypothèses suivantes :

- Petites déformations (la structure chargée ne change pas de forme).
- Matériau élastique (pas de plasticité ou de défaut).
- Chargement statique (le chargement et appliqué à la structure lentement et régulièrement).

L'analyse linéaire peut fournir la plupart des informations à propos du comportement de la structure et peut être une bonne approximation pour plusieurs

autres analyses. C'est aussi une base pour l'analyse non linéaire dans la plupart des cas [*6*].

II.1.3 Fonctions de forme

En MEF classique, les champs de déplacements dans chaque élément sont approchés par des fonctions de forme polynomiales multipliées par les déplacements des nœuds. Les fonctions d'interpolations utilisées sont habituellement de type Lagrange ou Hermite d'ordre particulier. Pour chacun de ces éléments, le nombre d'équations de base est le même nombre de degrés de liberté. Ces équations sont ainsi assemblées à l'aide des nœuds partagés par les éléments voisins, les conditions aux limites sont appliquées par la suite afin de résoudre l'équation pour trouver les déplacements nodaux [7].

$$B_j(x) = \prod_{\substack{i=0 \\ i\neq j}}^{n} \frac{x-x_i}{x_j-x_i} = \frac{x-x_0}{x_j-x_0}\ \frac{x-x_1}{x_j-x_1}\ \ldots.\ \frac{x-x_n}{x_j-x_n} \qquad (II.1.20)$$

Propriétés des fonctions d'interpolation de Lagrange

a. $\sum_{j=1}^{n} B_j(x) = 1$

b. $B_j(x) = \begin{matrix} 1\ pour\ x = x_j \\ 0 \qquad ailleur \end{matrix}$

c. L'ensemble de *n* points de contrôle peut être interpolé par un seul polynôme de Lagrange de degré *n-1*

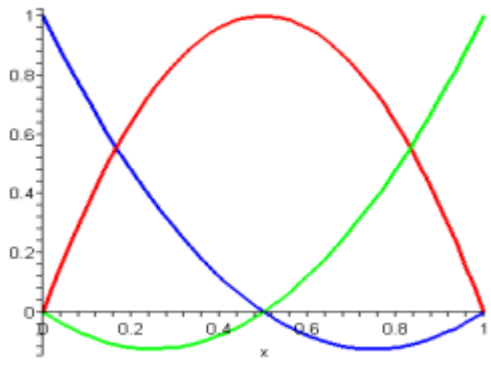


Figure (2-1.5) : *fonctions d'interpolation de Lagrange*

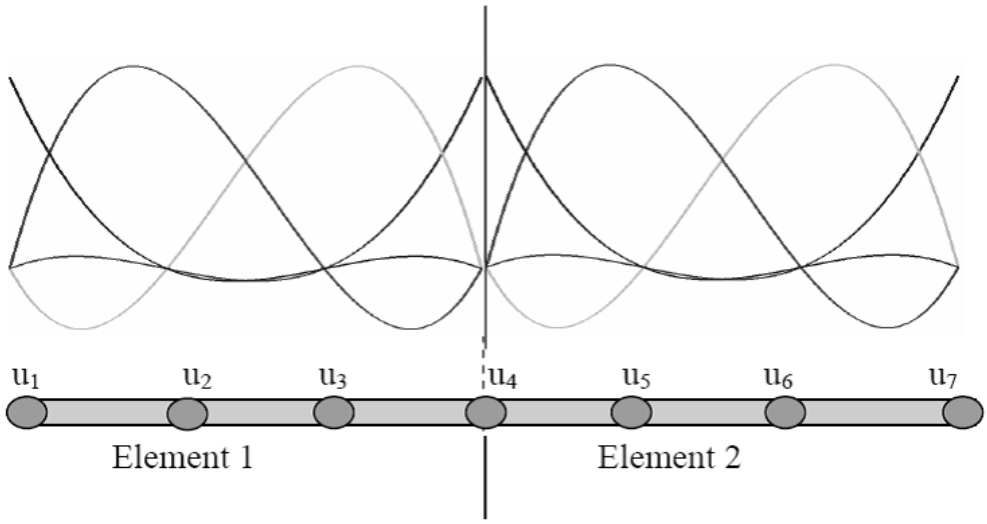


Figure (2.1.6): *deux éléments de Lagrange cubiques*

La figure (2.1.6) montre deux éléments cubiques de Lagrange en cas d'éléments finis réguliers, avec quatre degrés de liberté par élément. Comme il peut être vu dans la figure, il n'y a aucun chevauchement dans les définitions de fonction entre les deux éléments. Par conséquent, les degrés de liberté associés dans un élément ne sont pas partagés par les éléments contigus, excepté à la limite de l'élément adjacent. Pour calculer les matrices raideur et masse (K, M) de l'élément, il y a un besoin d'évaluer des intégrales. En raison de la complexité algébrique, il n'est toujours pas possible de calculer les intégrales exactement, alors on fait appel à des méthodes d'intégration, La méthode la plus employée est la méthode de gauss-Legendre dans laquelle l'intégrale peut être calculée pour un intervalle [- 1, 1].

II.2 Les NURBS comme base pour l'analyse

II.2.1 Introduction

L'analyse aux éléments finis (AEF) utilise des fonctions de forme et des nœuds, alors que le dessin assisté par ordinateur (DAO) utilise des fonctions de base et des points de contrôle. La situation typique dans la pratique est que les conceptions sont faites à l'aide des systèmes DAO et le maillage généré à partir des données de DAO. Ce qui mène à la complexité de l'analyse et à des résultats moins approchés. La figure suivante nous donne une idée sur cette approche [8].

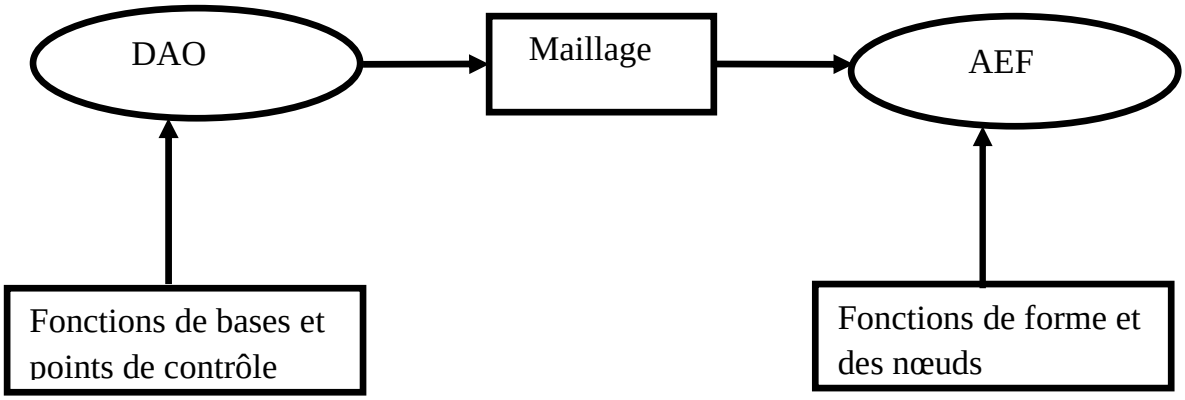


Figure (II.2.1) : *relation entre DAO et AEF dans l'analyse aux éléments finis classique*

L'analyse isogéométrique, introduite récemment par T.J.R.Hughes, permet de faire le lien entre le D.A.O. (Dessin Assistée par Ordinateur) et AEF. Le nom de l'analyse isogéométrique signifie que les mêmes fonctions de base peuvent être employées dans le DAO et le AEF. La figure (II.2.2) montre la relation entre DAO et AEF dans l'analyse isogéométrique [8].

II.2.2 Définition

L'analyse isogéométrique est une méthode utilisant les fonctions de base NURBS, qui servaient habituellement comme modèle de représentation géométrique en dessin assisté par ordinateur (DAO). Le principal avantage des NURBS par rapport aux fonctions de forme polynomiales utilisées en MEF est que ces dernières ne font

qu'approcher la géométrie d'origine, alors que les NURBS décrivent la géométrie exacte du modèle ce qui donne des résultats beaucoup plus précis.

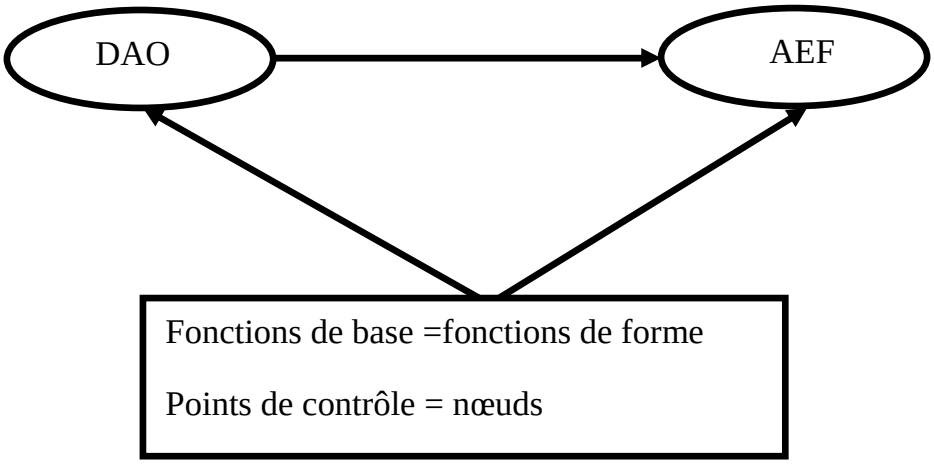


Figure (II.2.1) : *relation entre DAO et AEF dans l'analyse isogéométrique*

Dans l'analyse isogéométrique, la géométrie du modèle ***C(ξ)*** est exprimée par une combinaison linéaire de fonctions de base NURBS $\boldsymbol{R_i}$ et de points de contrôle $\boldsymbol{B_i}$, tel que :

$$C(\xi) = \sum_{i=1}^{n} R_i(\xi) B_i \qquad (II.2.1)$$

Avec les $fonctions\ de\ base\ NURBS\ \boldsymbol{R_{i,m}(\xi)} = \frac{\boldsymbol{w_i N_{i,m}(\xi)}}{\sum_{i=1}^{n} \boldsymbol{w_i N_{i,m}(\xi)}}$ **(II. 2. 2)**

$\boldsymbol{N_{i,m}(\xi)}$ **:** Les fonctions de bases B-Spline.

$\boldsymbol{w_i}$ **:** Les poids associés aux pôles

m : est le degré de la fonction.

II.2.3 Analyse isogéométrique en utilisant les NURBS (AIG)

La méthode d'analyse isogéométrique comme la méthode des éléments finis, utilise le concept isoparamétrique. Dans la méthode des éléments finis chaque élément est généralement transformé en coordonnées paramétriques (figure (II.2.2)**).**

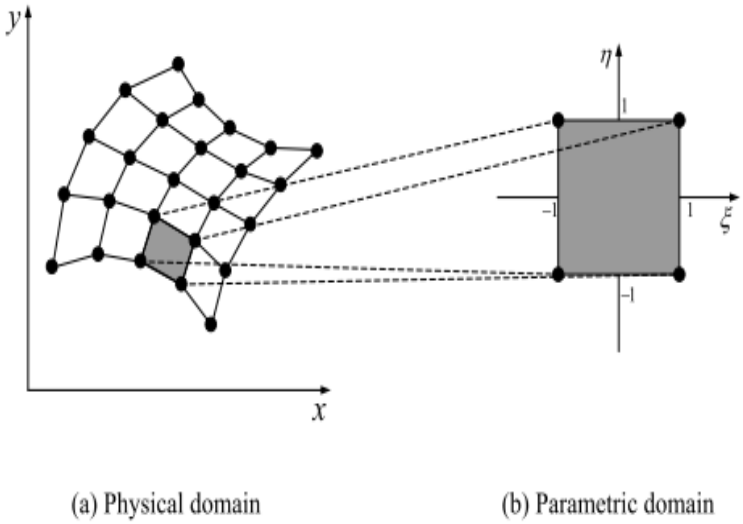


Figure (II.2.2): *Ttransformation du domaine physique au domaine paramétrique MEF*

Dans l'analyse isogéométrique, c'est tous le domaine physique qui est transformé en un domaine paramétrique (figure (II.2.3)**)** [9].

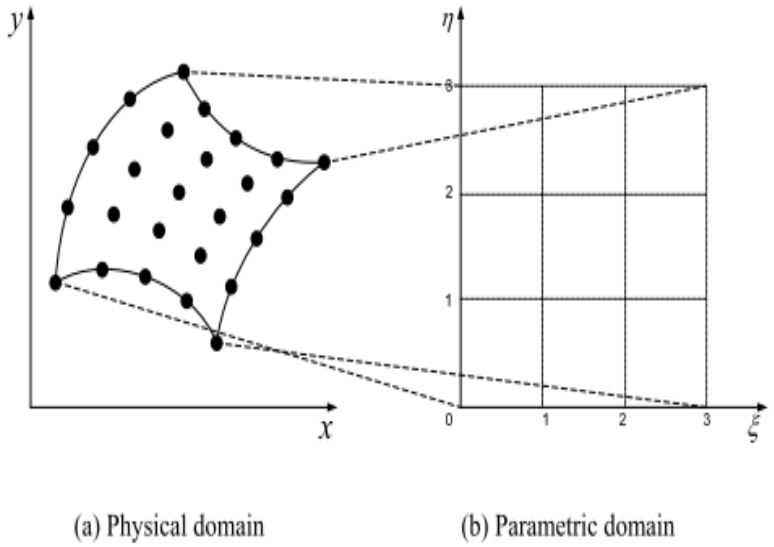


Figure (II.2.3) : *Transformation de domaine physique au domaine paramétrique AIG*

Soit un solide occupant un domaine Ω et son champ de déplacement $\boldsymbol{u(x) \epsilon R^2}$, $\boldsymbol{x \epsilon \Omega}$. Dans le cadre des petites déformations, la réponse du solide est décrite en termes du tenseur de déformations linéaire [10] :

$$\varepsilon(u) = \nabla^s u = \frac{1}{2}(\nabla u + \nabla u^T) \quad \text{……..(II.2.3)}$$

L'équation gouvernant les déplacements $\boldsymbol{u}$, avec comme conditions aux limites imposées $u = \bar{u}\ en\ \delta_u\Omega \subseteq \delta\Omega$, s'écrit :

$$\int_\Omega \delta\varepsilon : \sigma d\Omega = \int_\Omega \delta u . \rho b d\Omega + \int_\Omega \delta u . \bar{t} d\Gamma \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots (II.2.4)$$

Où :

- σ : Le tenseur de contraintes dépendant de $\boldsymbol{u}$.
- $\delta\varepsilon$: la première variation de déformation.
- ρb : les forces volumiques du corps.
- $\bar{t}$: la traction imposée sur une limite $\boldsymbol{\delta_t\Omega \subset \delta\Omega}$ du solide**.**

La méthode consiste alors à trouver les solutions $\boldsymbol{u}$ que l'équation (II.2.4) admet pour toute variation des déplacements admissibles, avec $\boldsymbol{\delta u = 0\ en\ \delta_u\Omega}$.

Dans la méthode des éléments finis la géométrie est définie comme suit:

$$x(\xi, \eta) = \sum_{i=1}^{n} N_i(\xi, \eta)\, x_i \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (II.2.5)$$

Avec : x_i la position du nœud correspondant.

Dans l'analyse isogéométrique la géométrie est définie comme suit :

$$x(\xi, \eta) = \sum_{i=1}^{n} R_i(\xi, \eta)\, P_i = \boldsymbol{RP} \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (II.2.6)$$

Où $R_i(\xi, \eta)$ sont les fonctions de base NURBS et **R** la matrice des fonctions de base :

$$\boldsymbol{R} = \begin{bmatrix} R_{1.1} & 0 & R_{2.1} & 0 & \dots & R_{n.m} & 0 \\ 0 & R_{1.1} & 0 & R_{2.1} & \dots & 0 & R_{n.m} \end{bmatrix} \dots\dots\dots\dots\dots\dots\dots (II.2.7)$$

$$\boldsymbol{P} = [P_{1.1}^x, P_{1.1}^y, P_{2.1}^x, \dots, P_{n.m}^y]^T = [x_1, y_1, x_2, y_2 \dots\dots\dots\dots\dots\dots x_n, y_n]^T \dots (II.2.8)$$

Le champ des solutions ***u(ξ)*** est représenté par une combinaison linéaire des mêmes fonctions de base $\boldsymbol{R_i}$ avec les coefficients de réponses $\boldsymbol{d_i}$ (des déplacements nodaux, dans le cas d'élasticité linéaire), tel que :

$$u(\xi, \eta) = \sum_{i=1}^{n} R_i(\xi, \eta) d_i = \boldsymbol{Rd} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (II.2.9)$$

$$\boldsymbol{d} = [\boldsymbol{u_1}, \boldsymbol{v_1}, \boldsymbol{u_2}, \boldsymbol{v_2}, \dots\dots\dots\dots\dots \boldsymbol{u_n}, \boldsymbol{v_n}]^{\boldsymbol{T}} \dots\dots\dots\dots\dots\dots\dots\dots (II.2.10)$$

II.2.3.1 Matrice raideur

Dans ce qui suit, nous allons déduire la matrice de rigidité pour un "patch". A partir des équations (II.2.3) et (II.2.4), le vecteur des déformations correspondant à la condition des déformations planes est le suivant :

$$\varepsilon^h = \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{pmatrix} = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{d} \qquad (II.2.11)$$

Avec ***B(x)*** définissant l'operateur des déformations linéaires dont la notation matricielle est donnée par :

$$\boldsymbol{B} = \begin{bmatrix} \dfrac{\delta}{\delta x} & 0 \\ 0 & \dfrac{\delta}{\delta y} \\ \dfrac{\delta}{\delta x} & \dfrac{\delta}{\delta y} \end{bmatrix} \boldsymbol{R}. \qquad (II.2.12)$$

La transformation entre le système de coordonnées globales et les coordonnées paramétriques des NURBS est donnée par le Jacobien qui est défini comme suit :

$$\boldsymbol{J} = \begin{bmatrix} \frac{\delta x}{\delta \xi} & \frac{\delta y}{\delta \xi} \\ \frac{\delta x}{\delta \eta} & \frac{\delta y}{\delta \eta} \end{bmatrix} \qquad (II.2.13)$$

Ce qui mène a :

$$\begin{pmatrix} \frac{\delta \boldsymbol{R}}{\delta x} \\ \frac{\delta \boldsymbol{R}}{\delta y} \end{pmatrix} = \boldsymbol{J^{-1}} \begin{pmatrix} \frac{\delta \boldsymbol{R}}{\delta \xi} \\ \frac{\delta \boldsymbol{R}}{\delta \eta} \end{pmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (II.2.14)$$

Les $\frac{\delta R}{\delta \xi}$ et $\frac{\delta R}{\delta \eta}$sont les dérivées partielles des fonctions de bases NURBS vues dans le chapitre 1. A partir de là, la matrice de rigidité élémentaire est donnée par :

$$K = \iint_{\Omega_{\text{patch}}} \boldsymbol{B^T}(\xi,\eta)\mathbb{C}\,\boldsymbol{B}(\xi,\eta)\,\det(\boldsymbol{J})\,d\xi d\eta. \qquad (II.2.15)$$

Avec$\mathbb{C}$, la matrice caractéristique au comportement élastique du matériau, $\mathbb{C} = c_{i,j}$, coefficients élastiques.

II.2.3.2 Matrice de masses

Dans les problèmes de dynamique des structures, l'équation des mouvements est donnée pour un système conservatif libre, dans la forme matricielle :

$$\boldsymbol{M\ddot{U} + KU = F} \qquad (II.2.15)$$

Tel que :

- M : La matrice de masse.
- $\ddot{U}$: La deuxième dérivée par rapport au temps du vecteur des déplacements, qui représente les accélérations nodales.

La matrice de masse est donnée par :

$$M = \delta \iint_{\Omega_{\text{patch}}} \boldsymbol{B^T}(\xi,\eta)\,\rho\boldsymbol{B}(\xi,\eta)\,\det(\boldsymbol{J})\,d\xi d\eta. \qquad (II.2.16)$$

Tel que :

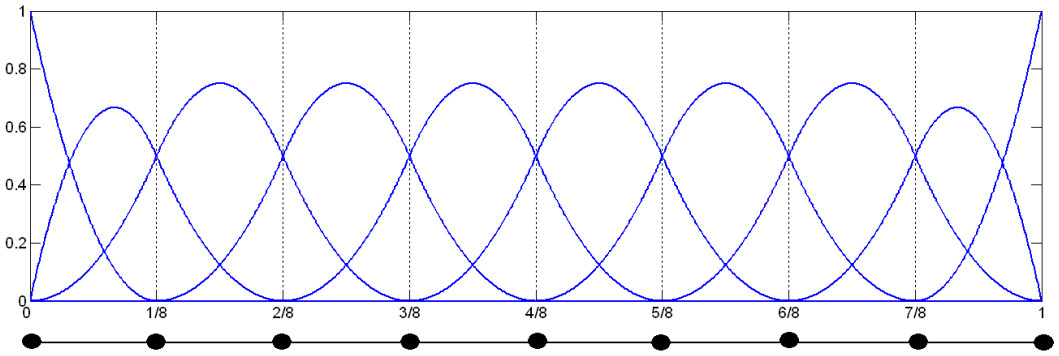
- δ : Indice de Cronneker.
- ρ : La masse volumique du matériau.

Remarque

- **Intégrale numérique** : Vu la complexité analytique des fonctions de base NURBS, une intégration numérique (quadrature) s'impose, pour cela plusieurs méthodes peuvent être employées, telles que la quadrature de Gauss, de Simpson, la méthode du point milieu, la méthode des trapèzes, cette dernière sera employée dans l'application accompagnant ce présent mémoire.

II.2.4 résumé des principales notions de l'analyse isogéométrique [11]

- Le maillage par les NURBS est défini par le produit des vecteurs nœuds. pour une structure unidimensionnelle par le vecteur nodal $\boldsymbol{\Xi}$ et par le produit vectoriel $\boldsymbol{\Xi} \times \mathbf{H}$ pour une structure bidimensionnelle.
- L'espace inter-nodal subdivise de domaine en éléments.
- Le support local de chaque fonction de base s'étend suivant un nombre restreint d'éléments, figure (II.2.4).
- Les points de contrôle définissent la géométrie.
- Le concept isoparamétrique est invoqué ; les champs (déplacement, vitesse, température, …) sont représentés avec les mêmes fonctions de base que celles de la géométrie. Les coefficients des fonctions de base sont des degrés de liberté, ou des variables de contrôle.
- Trois stratégies différentes de raffinement de maillage sont possibles:
  - h-raffinement : par des insertions de nœud.
  - p-raffinement : par l'élévation de l'ordre des fonctions de base.
  - et une nouvelle possibilité appelée k-raffinement.



Figure(II.2.4) : *Chaque fonction de base est définie sur un intervalle de l'espace paramétrique comprenant un nombre restreint d'éléments.*

III.1 Recalage de modèle

La méthode de recalage est basée sur la perturbation du système, par une perturbation dans le système matriciel (matrice raideur, masse et amortissement) de telle façon que la réponse de ce nouveau modèle ressemble aux données mesurées aussi près que possible. Avec la technique de recalage de modèle, les défauts peuvent être identifiés par la comparaison entre le modèle recalé et le modèle original. Le recalage de modèle n'est pas seulement une technique de détection des défauts dans une structure, mais aussi mesurer la gravité des défauts [12].

III.1.1 Modélisation éléments finis et grandeurs analytiques

Dans l'analyse des structures il est essentiel de déterminer les valeurs propres et les vecteurs propres.

Pour un système linéaire, les caractéristiques dynamiques (fréquences et formes propres) peuvent être décrites par un ensemble d'équations du deuxième ordre:

$$([K_a] - \Omega^2{}_{ai}[M_a])\{\emptyset_a\} = 0 \qquad pour \quad i = 1, 2, \dots, n \qquad (2.4.1.1)$$

Où :

- $[K_a]$ et $[M_a]$ sont respectivement, les matrices globales de rigidité et de masse.
- $[\Omega_{ai}^2] = \omega_{a1}{}^2, \dots, \omega_{ai}\ et\ \{\emptyset_a\}_i$ sont la i^{eme} fréquence naturelle et la i^{eme} forme propre de la structure.

Supposons que le modèle analytique ($[K_a]$ et $[M_a]$) doit être recalé en utilisant les données du modèle expérimental ($\Omega_x{}^2 = \omega_{x1}{}^2, \dots, \omega_{xi}\ et\ \{\emptyset_x\}_i$ de sorte qu'il représente plus exactement les caractéristiques dynamiques de la structure modelée.

Dans ce qui suit, on fait un récapitulatif pour quelques méthodes de recalage.

III.1.2 Méthode de recalage

Le recalage a pour but de corriger les paramètres du modèle pour que celui-ci vérifie au mieux le comportement réel de la structure ou de détecter d'éventuels endommagements d'une structure en service. Certains paramètres, tels que les conditions aux limites, sont estimés avec une incertitude importante. D'autre part, le procédé de fabrication n'est que réalisation imparfaite de modèle.

Dans le recalage on trouve deux grandes familles de méthodes; les méthodes globales et les méthodes locales.

III.1.2.1 Méthodes globales

Appelées aussi méthodes directes Basées sur l'établissement des grandeurs modales expérimentales.

III.1.2.1.1 Utilisation de la matrice de masse pour référence

La matrice de masse est considérée comme exacte, la matrice des vecteurs propres expérimentaux est corrigée pour vérifier les propriétés d'orthogonalité. Les corrections de la matrice de raideur sont ensuite calculées pour vérifier les mesures modales tout en minimisant l'écart avec la matrice initiale. Le choix de la matrice de masse est justifié par le fait que dans le cas contraire une analyse statique pourrait souvent être plus précise qu'une analyse dynamique. Le problème peut être formulé [13] [14]:

- Trouver $\emptyset_c$ minimisant : $\left|\left|(M_x]^{-\frac{1}{2}}([\emptyset_c] - [\emptyset_x])\right|\right|_F^2$ ……..(III.1)
- Sous la contrainte : $[\emptyset_c]^t[M_x]\,[\emptyset_c] = [I]$…… (III.2)
- Trouver $[\Delta K]$ symétrique minimisant : $\left|\left|[(M_x]^{-\frac{1}{2}}[\Delta K](M_x]^{-\frac{1}{2}}\right|\right|_F^2$ …..(III.3)
- Sous la contrainte :
  $([K_a] + [\Delta K])\,[\emptyset_c] = [M_x]\;\,[\emptyset_c]\,[\Omega_a^2]$…….…..…… (III.4)

La norme $\left||A|\right|_F^2$ de la matrice A d'ordre *n* est une norme de FROBENIUS définie par :

$$\left||[A]|\right|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^2 = |tr([A]^t[A])|^2 \quad \text{.....(III.5)}$$

La résolution par la méthode des multiplicateurs de Lagrange fournit les solutions :

$$[\emptyset_c] = [\emptyset_a]([\emptyset_a]^t[M_x][\emptyset_a])^{-1/2} \quad \text{.....(III.6)}$$

$$\Delta K = -[K_a][\emptyset_c][\emptyset_c]^t[M_x] - [M_x][\emptyset_c][\emptyset_c]^t[K_a] + [M_x][\emptyset_c[\emptyset_c]^t[K_a][\emptyset_c][\emptyset_c]^t[M_x] + [M_x][\emptyset_c][\Omega_a^2][M_x] \text{.....} \quad \text{(III.7)}$$

III.1.2.1.2 Utilisation des modes expérimentaux pour référence

Méthode De Minimisation De Contrainte (CMM)

Dans ce cas on suppose que les valeurs propres et les vecteurs propres sont connus ($\emptyset_x et \Omega_x^2$) ensuite on calcule les défauts de masse et de raideur ($\Delta M, \Delta K$), aussi à partir de ces vecteurs et valeur propres ($\emptyset_x et \Omega_x^2$) , on calcule aussi la matrice de masse (M_x), La matrice de masse est cette fois corrigée pour assurer l'orthogonalité des mesures. Le problème s'exprime alors [13]:

- Trouver $[\Delta M]$ symétrique, minimisant : $\left|\left|[(M_x]^{-\frac{1}{2}}[\Delta M](M_x]^{-\frac{1}{2}}\right|\right|_F^2$ .....(III.8)
- Sous la contrainte : $[\emptyset_x]^t([M_x] + [\Delta M])[\emptyset_x] = [I]$..... (III.9)
- Trouver$[\Delta K]$ symétrique, minimisant : $\left|\left|[(M_x]^{-\frac{1}{2}}[\Delta K](M_x]^{-\frac{1}{2}}\right|\right|_F^2$ ……. (III.10)
- Sous la contrainte : $([K_a] + [\Delta K])\,[\emptyset_x] = ([M_x] + [\Delta M])\,[\emptyset_x]\,[\Omega_a^2]$ .....(III.11)

Les matrices de correction sont données par:

- $\Delta M = [M_x][\emptyset_x]\overline{[M]}^{-1}\left([I] - \overline{[M]}\right)\overline{[M]}^{-1}[\emptyset_x]^t[M_x]$.....(III.12)

Avec $[M_x] = [\emptyset_x][M_a][\emptyset_x]^t$ ……..…...……(III.13)

- $\Delta K =$

$-[K_a][\emptyset_x][\emptyset_x]^t[M_x] - [M_x][\emptyset_x][\emptyset_x]^t[K_a] +$

$[M_x][\emptyset_x][\emptyset_x]^t[K_a][\emptyset_x][\emptyset_x]^t[M_x] +$

$[M_x][\emptyset_x][\Omega_a^2][M_x]$ ……...…(III.14)

III.1.2.1.3 EMM (error matrix method)

Cette méthode établit les matrices d'écart [ΔM] et [ΔK] et en déduit les indicateurs d'erreur caractérisant les zones du modèle mal modélisées [15].

Cette méthode suppose que la matrice expérimentale de rigidité [K_x] est disponible, on définit alors la différence entre les deux matrices de rigidité ([Ka] et [Kx]) comme "la matrice d'erreur de rigidité" :

$[\Delta K] = [K_x] - [K_a]$ … (III. 15)

Après développement l'équation (III.16) devient :

$[K_x]^{-1} = [K_a]^{-1} - (\sum_{i=1}^{\infty}(-1)^i ([K_a]^{-1}[\Delta K])^i)[K_a]^{-1}$ … … … … … … … … … … (III. 16)

Cette équation peut être approximée par :

$[K_x]^{-1} \equiv [K_a]^{-1} - [K_a]^{-1}[\Delta K][K_a]^{-1}$ … … … … … … … … … … … … … … … … … … (III. 17)

De sorte que : $[\Delta K] \equiv [K_a]([K_a]^{-1} - [K_x]^{-1})[K_a]$ … … … … … … … … … … … (III. 18)

Par conséquent, une matrice d'erreur de rigidité peut être estimée en utilisant les données modales expérimentales et les données modales analytiques correspondantes de la manière suivante :

$[\Delta K] \equiv [K_a]([\emptyset_a][\Omega_a^2]^{-1}[\emptyset_a]^t - [\emptyset_x][\Omega_x^2]^{-1}[\emptyset_x]^t)[K_a]$ … … … … … … … (III. 19)

La matrice d'erreur déduite ainsi peut être employée pour identifier et pour localiser la différence entre la matrice expérimentale de rigidité et l'analytique.

De la même manière on définie l'erreur dans la matrice masse et on trouve :

$[\Delta M] \equiv [M_a](\ [\emptyset_a]\ [\emptyset_a]^t - \ [\emptyset_x]\ [\emptyset_x]^t)[M_a]$ … … … … … … … … … … … … … . (III. 20)

Cette classe de méthodes (Utilisation de la matrice de masse pour référence et Méthode De Minimisation De Contrainte (CMM)) est basée sur l'hypothèse que l'une des deux grandeurs: matrices des modes expérimentaux, matrice de masse, est exacte ; appelée aussi méthode de référence de base "Reference Basis Methods". Les grandeurs restantes sont corrigées par minimisation d'une fonction objective avec des contraintes imposées par des multiplicateurs de Lagrange [13].

En plus de ces méthodes on trouve d'autres méthodes globales de recalage comme [16] :

- **Méthode FBM** (Force Balance Method) : elle déduit de l'équation d'équilibre dynamique les zones du modèle mal modélisées par analyse des forces résiduelles.
- **Méthode BEE** (Balancing the Eigenvalue Equation) elle exploite les matrices d'écart [ΔM]et [ΔK] qu'elle met en œuvre au sein de l'équation d'équilibre dynamique pour en déduire un indicateur d'erreur qui, centré sur les degrés de liberté, identifie conséquemment les zones de modèle mal modélisées.

III.1.2.2 Méthodes locales

Les méthodes locales se basent sur un paramétrage initial du problème donné par les relations (III.22) et (III.24). Ces paramètres peuvent être de deux natures distinctes : les paramètres structuraux et les paramètres physiques [15].

Paramétrage structurel : le paramétrage structurel consiste à utiliser directement la discrétisation éléments finis du modèle en lui associant des paramètres décrivant des variations proportionnelles dans les matrices élémentaires correspondantes.

Si l'on convient de représenter par $[\theta]$ toute matrice structurelle associée au modèle ([M] ou [K]), la procédure est basée à produire, à chaque pas, la valeur $[\theta]$:

$$[\theta] = [\theta]^{0} + [\Delta\theta] \quad \text{……..(III.21)}$$

Si l'on représente par $[\theta^e]_i$ toute sous-matrice issue du modèle participant à l'assemblage général de$[\theta]$, le paramétrage structurel {p} de la correction peut être exprimé :

$$[\Delta\theta] = \sum_{\mathrm{i}=1}^{\mathrm{N_p}} \Delta\mathrm{p_i} \, . \, [\theta^e]_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . \, (\mathrm{III}.22)$$

Le paramétrage {p} comprend N_p paramètres P_i

- **Paramétrage physique**

La formulation suivante consiste à définir des paramètres rendant clairement compte de la physique du système (géométrie, propriétés physiques, ….).

$$[\theta] = [\theta]^0 + \sum_{i=1}^{n} \delta p_i \, . \frac{\partial[\theta]}{\partial p_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots ... (\mathrm{III}.23)$$

L'avantage principal du paramétrage physique est que les résultats fournis sont facilement interprétables en termes de modifications pertinentes du système

III.1.2.2.1 méthode SEF

La méthode SEF (Substructure Energy Functions) établit des fonctions énergétiques basées sur les matrices élémentaires $[M^e]$ et $[K^e]$ participant à l'assemblage global des matrices de masse et de raideur [M] et [K], Pour ***m*** modes propres identifiés dans l'espace fréquentiel d'intérêt, on peut ainsi constituer, pour chaque structure élémentaire définie par les matrices élémentaires, deux indicateurs d'énergie résiduelle cinétique et potentielle.

On a :

$$\Delta E_i^c = \sum_{r=1}^{m} ([\emptyset_{A,a}] - [\emptyset_{E,a}])^T \, . \, [M^e] . \left([\emptyset_{A,a}] - [\emptyset_{E,a}]\right) . \, \omega_{E,a}^2 \ldots\ldots\ldots\ldots\ldots\ldots . \, (\mathrm{III}.24)$$

$$\Delta E_i^p = \sum_{r=1}^{m} ([\emptyset_{A,a}] - [\emptyset_{E,a}])^T \, . \, [K^e] . \left([\emptyset_{A,a}] - [\emptyset_{E,a}]\right) . \, \omega_{E,a}^2 \ldots\ldots\ldots\ldots\ldots\ldots . \, (\mathrm{III}.25)$$

Les énergies résiduelles importantes indiquent des structures élémentaires erronées. Les énergies résiduelles peu significatives indiquent, soit une structure élémentaire bien modélisée, soit une insensibilité des grandeurs expérimentales aux modifications structurelles correspondantes.

III.1.2.2.2 minimisation de l'erreur sur les réponses

L'objectif est de minimiser l'écart entre les réponses mesurées et calculées. Les sensibilités des pulsations et des modes propres sont obtenues par différentiation de l'équation d'équilibre (III.27) et des conditions d'orthogonalité (III.28) :

$$([K] - w_i^2[M])\ \frac{\partial \emptyset_i}{\partial p} + \left(\frac{\partial [K]}{\partial p} - w_i^2 \frac{\partial [M]}{\partial p} - \frac{\partial w_i^2}{\partial p}[M]\right)\emptyset_i = 0 \ldots\ldots\ldots\ldots\ldots \text{(III. 26)}$$

$$\emptyset_i^t \frac{\partial [M]}{\partial p}\emptyset_i + 2\emptyset_i^t [M]\frac{\partial \emptyset_i}{\partial p} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(III. 27)}$$

D'autres techniques telles que la méthode du Simplex ou les algorithmes génétiques permettent d'éviter le calcul des sensibilités, mais impliquent généralement des temps de calcul exagérés. Ces techniques sont basées sur une perturbation aléatoire ou systématique des paramètres et pour sélectionner les perturbations les mieux adaptées [13].

Les méthodes locales, aussi appelées itératives, nécessitent un investissement plus important de l'utilisateur, aussi bien au niveau du paramétrage de la structure que de 1'exploitation des mesures, de l'évaluation de la validité du modèle, de la compréhension des causes des erreurs et du choix des paramètres à corriger. Ces méthodes dépassent la simple correction d'un modèle, pour s'intégrer dans un processus de compréhension du comportement réel de la structure. En effet, même si souvent le gain en productivité est limité, elles permettent de déterminer les limites de la modélisation et donnent une vision plus critique des résultats de calculs. Leur principal avantage est de conserver le sens physique du modèle en corrigeant soit des

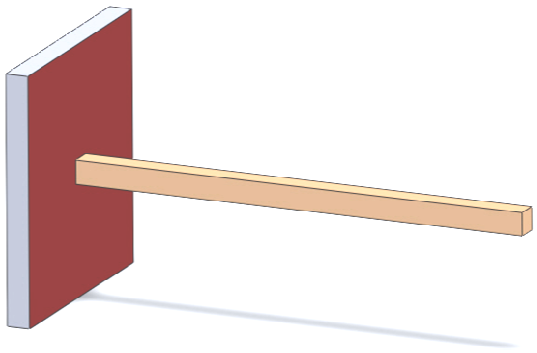
paramètres globaux tels que les dimensions géométriques, soit des paramètres locaux tels que les modules de Young ou la densité d'éléments ou de groupes d'éléments [13].

IV. Application du recalage de modèle en utilisant les fonctions de base NURBS (AIG)

Dans le chapitre précédent on à illustré quelques méthodes de recalage, Ce chapitre fera l'objet d'une application d'une de ces méthodes de recalage de modèle, soit une méthode globale (directe).

En appliquant la méthode de CMM (constraint minimisation method) pour le recalage de modèle et cela en utilisant un modèle isogéométrique au lieu du modèle aux éléments finis classiques.

Dans cette application on utilise une simple poutre encastrée-libre. La figure (IV.1) illustre les propriétés du matériau et les dimensions de la poutre.



Longueur L=50 mm

Section carrée de coté a=5mm

Module d'élasticité E=210 000 Mpa

Masse volumique ρ=2800 Kg/m^3

Figure (IV-1) : *modèle à recaler avec ces caractéristiques*

IV.1 Création du modèle

En commençant par la modélisation de la géométrie du modèle par une courbe NURBS. Le modèle sera représenté par une simple droite, reliant les deux points de contrôles $x_1=0$ *et* $x_9=50$ associés a deux fonctions de base NURBS d'ordre $p=1$ et de vecteur nœud $\Xi=\{0, 0, 1, 1\}$ et les poids $w=\{1,1\}$, vue la simplicité du modèle. Les fonctions de base NURBS utilisées sont illustrées dans la figure (IV-2)

IV.2 discrétisation

Après discrétisation du modèle en 9 éléments. Les courbes NURBS seront raffinées par insertion de nœuds (h-raffinement), le nouveau vecteur nodal sera alors :

Ξ= {0, 0, 1/9, 2/9, 3/9, 4/9, 5/9, 6/9, 7/9, 8/9, 1,1} et la figure (IV-3) nous montre les fonctions de base après raffinement.

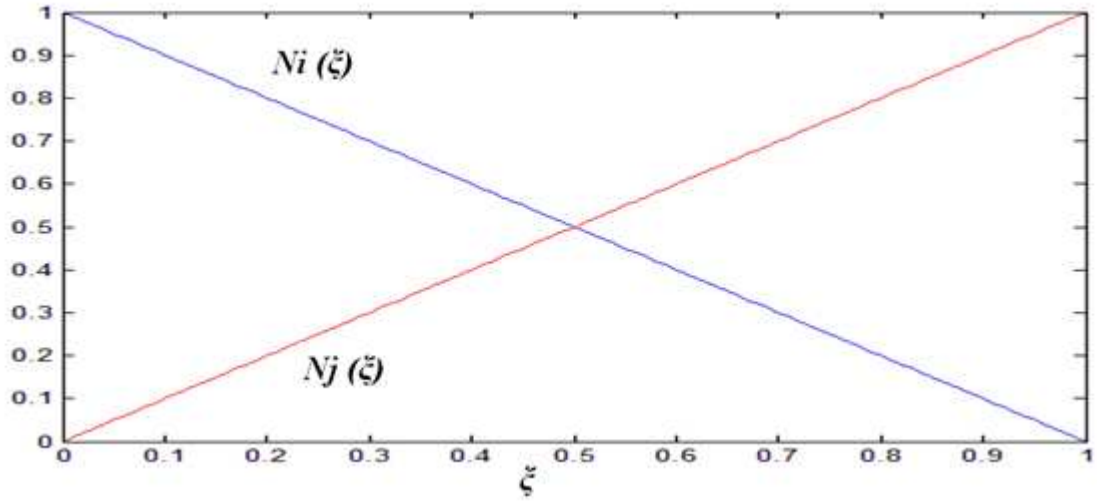


Figure (IV-2) : *Allure des fonctions de base NURBS d'ordre p=1 définissant la poutre.*

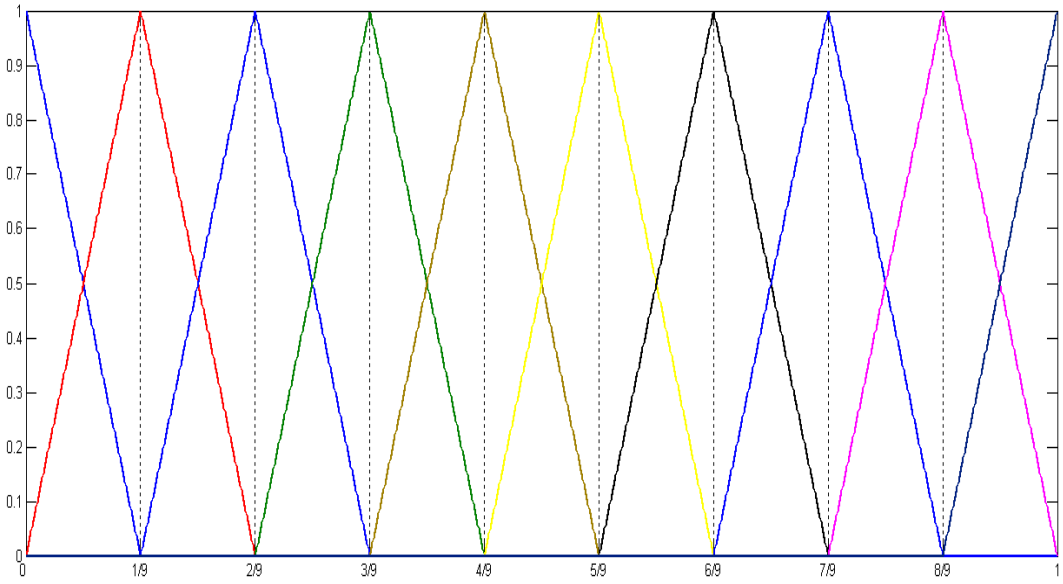


Figure (IV-3) : *Raffinement de la courbe NURBS avec insertion de nœuds. La courbe est discrétisée en 9 éléments.*

IV.3 Evaluation des matrices Masse et Rigidité du modèle

- **Matrice de raideur**

En appliquant la formule $(II.2.15)$ vue en chapitre (II) pour évaluer la matrice de rigidité pour chaque élément :

$$Me_{ij} = \rho \int_0^{1/9} R_i(\xi) R_j(\xi)\ J\ d\xi \ldots\ldots\ldots\ldots\ldots (IV.1)$$

Avec :

$$J = \sum_{i=1}^{9} R_i(\xi)\ x_i \ldots\ldots\ldots (IV.2)$$ Le Jacobien

Après la résolution d'équation avec intégration par la méthode des trapèzes on trouve :

$$Ke1 = 10^7 \begin{bmatrix} 1.0490 & -1.0490 \\ -1.0490 & 1.0490 \end{bmatrix} \ldots\ldots\ldots (IV.3)$$

Vue la nature de modèle, les résultats pour le reste d'éléments sera le même alors :

$$Ke1 = Ke2 = \cdots \ldots\ldots\ldots\ldots\ldots = Ke9$$

Pour avoir la matrice de rigidité globale on assemble les matrices de rigidité élémentaires.

- **Matrice masse**

De même pour la matrice de masse globale. En calcul la matrice masse pour chaque élément à partir de l'équation $(II.2.16)$ vue en chapitre (II) et on fait l'assemblage.

La condition à la limite encastrée est introduite en supprimant la première ligne est la première colonne des matrices de masse et de rigidité.

IV.4 Evaluation des valeurs et formes propres

Les valeurs et vecteurs propres du modèle sont obtenus par la résolution de l'équation :

$$\left(K_a - {\Omega_a}^2 M\right).\emptyset_a = 0 \ldots\ldots\ldots\ldots\ldots (IV.4)$$

Les valeurs propres de notre modèle sont représentées sur la figure (IV-4)

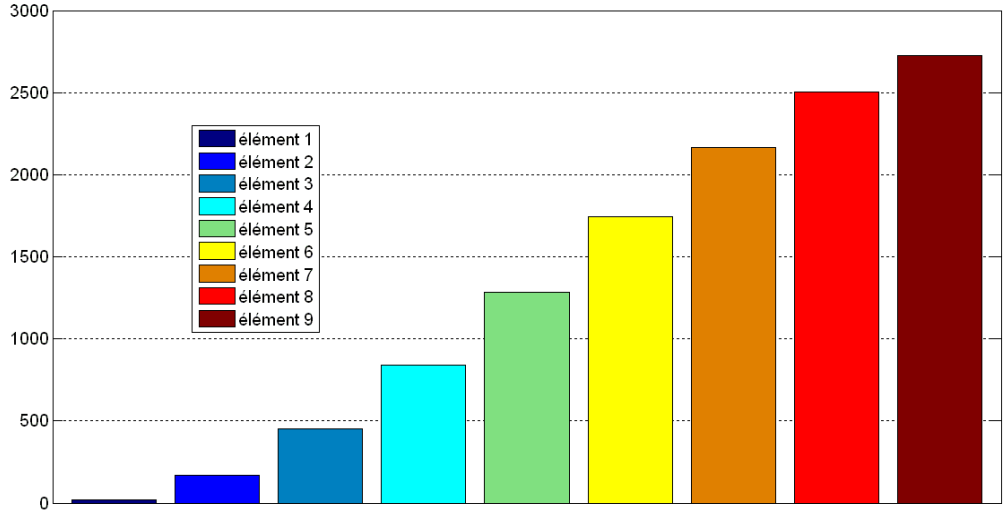


Figure (IV.4) : *Les valeurs propres aux nœuds du modèle analysé*

IV.4 Simulation numérique d'un cas de recalage

Après le calcul des deux matrices du système (masse et raideur), on procède au recalage en utilisant la méthode de CMM pour notre application, dans cette méthode on considère que les valeurs propres et les vecteurs propres du système perturbé ($\emptyset_x et \Omega_x^2$) sont connues et à travers ces données on évaluera les matrices de masse et de raideur $K_p et\ M_p$ du système perturbé, ainsi que les matrices de correction(ΔK et ΔM).

Les valeurs propres $\emptyset_x$ du modèle perturbé sont représentées sur la figure (IV-5)

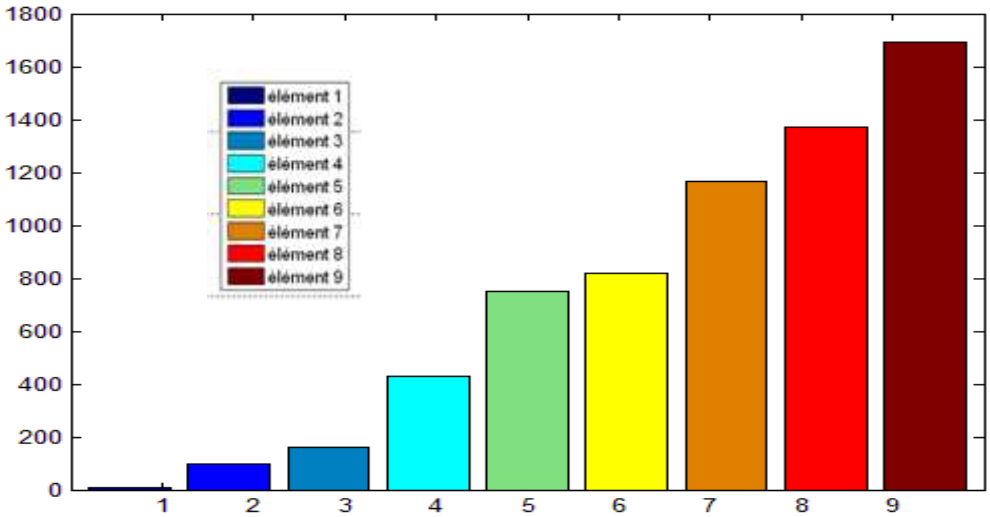
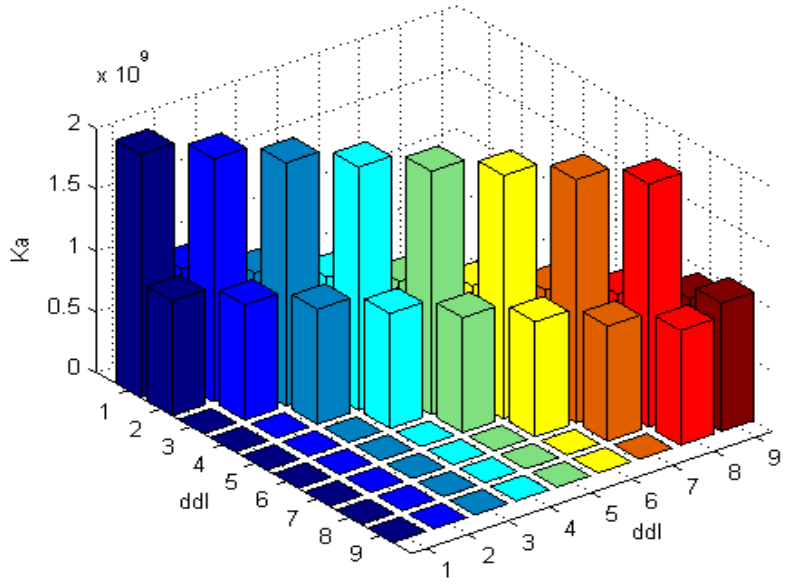
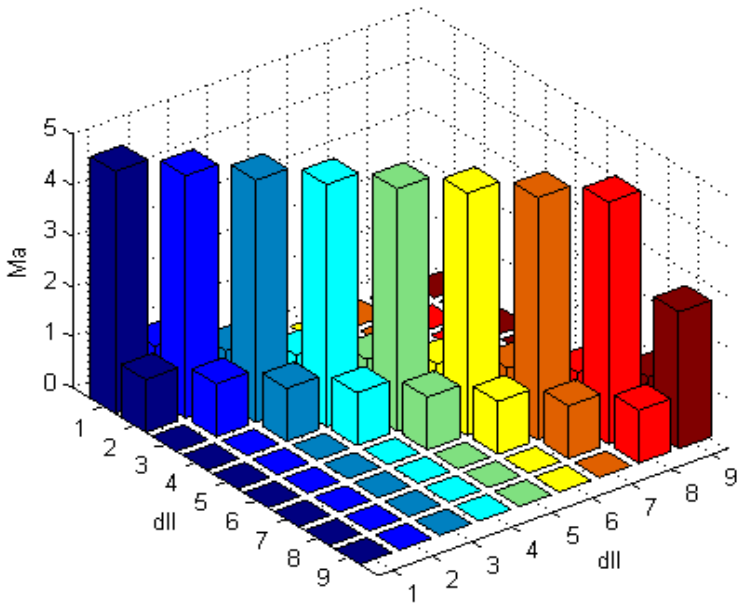


Figure (IV.5) : *Les valeurs propres aux nœuds du modèle perturbé*

Les figures (IV.6) et (IV.7) montrent l'état des matrices masse et raideur avant la perturbation du système.



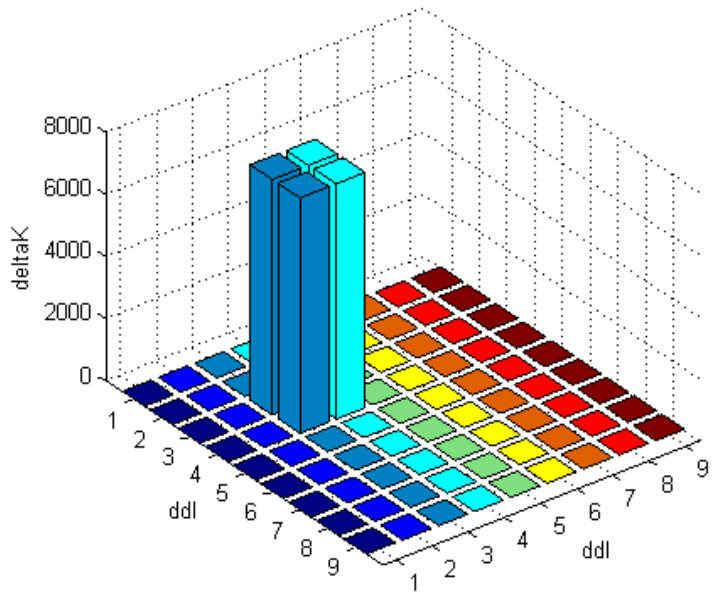
Figure(IV.6) : *Matrice raideur avant perturbation*



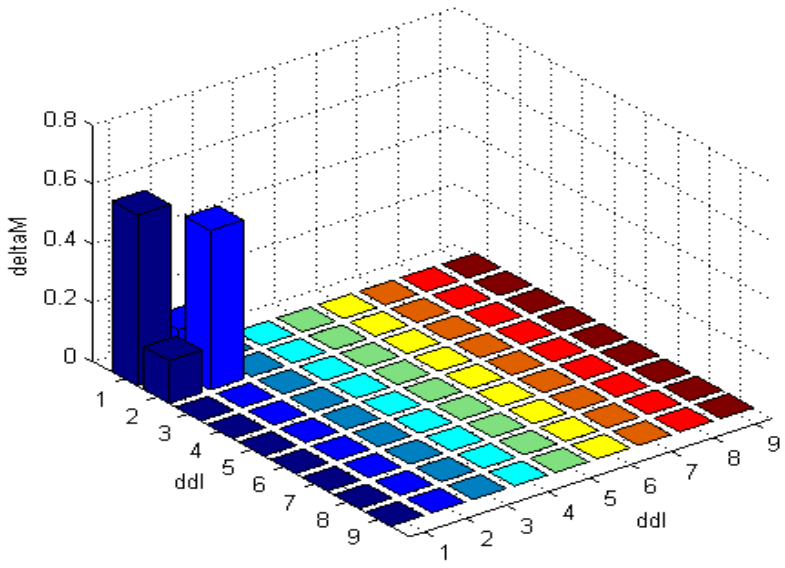
Figure(IV.7) : *Matrice masse avant perturbation*

En utilisant les équations ((III.13) et (III.15)) vues dans le chapitre (III) de la méthode de CMM pour calculer les matrices de corrections(ΔK et ΔM).

Les deux figures (IV.8) et (IV.9) présentent les défauts (ΔK et ΔM) sur les matrices raideur et masse respectivement.



Figure(IV.8) : *Localisation de* ΔK



Figure(IV.9) : *Localisation de* ΔM

Discussion des résultats

À travers les résultats obtenus, représentés sur les figures (IV.8) et (IV.9).

- Pour la matrice de raideur, le défaut (ΔK) influe sur les degrés de liberté 3 et 4 d'où les nœuds (4 et 5), ce qui veut dire que l'élément 4 est endommagé (son module de Young est perturbé).
- Pour la matrice masse, le défaut (ΔM) influe sur les degrés de liberté 1 et 2 d'où les nœuds (2 et 3), ce qui veut dire que l'élément 2 est endommagé (sa masse volumique est perturbé).

Dans cette application on à utiliser la méthode CMM (Constraint minimisation method) qu'est une méthode globale directe vue la simplicité du modèle à recaler. Dans un cas plus délicat où la structure a une forme plus compliquée, on à intérêt à utiliser des méthodes locales itératives.

Les méthodes de recalage, en général sont utilisées quant il y a un manque de données expérimentales pour tous les degrés de libertés, et la méthode utilisée est la plus facile dans la manipulation mathématique.

Conclusion générale

Dans ce travail on a vue tout d'abord la modélisation géométrique en utilisant les différents modèles de modélisation en DAO, comme le modèle de Bézier qui utilise les fonctions de base de Bernstein ensuite le modèle B-spline défini avec les fonctions de base par morceaux.

On a aussi récapitulé le modèle aux éléments finis classique et le nouveau modèle d'analyse basé sur les NURBS intitulé (*analyse isogéométrique*).

L'analyse isogéométrique utilise les mêmes fonctions de forme utilisées dans les systèmes (DAO) ce qui donne une très grande précision comparant à l'analyse aux éléments finis classique. Le tableau suivant illustre quelques différences entre l'analyse isogéométrique en utilisant les NURBS et l'analyse aux éléments finis classiques :

Analyse isogéométrique utilisant les NURBS	Analyse aux éléments finis classiques
-description exacte de la géométrie -utilise les points de contrôles -utilise les variables de contrôles -h-p-k raffinement - Continuité élevée et facilement contrôlée	- description approximative de la géométrie -utilise les points nodaux -utilise les variables nodales -h-p raffinement -continuité C^0, fixée

Le but de ce travail a été le recalage d'un modèle élément fini en utilisant un modèle isogéométrique basé sur les fonctions de base NURBS, il y a d'autre fonctions de base développées récemment pour ce nouveau concept d'analyse et qu'en appelle les fonctions de base T-Spline. Le recalage de modèle peut être fait en utilisant différentes méthode, Dans ce travail on a opté pour une méthode globale qui est CMM ; une méthode classique de recalage ; cette méthode utilise comme référence les vecteurs et les valeurs propres, pour définir les matrices de système perturbé (M_x et K_x) et aussi déterminer les défauts de raideur et de masse($\Delta K \; et \; \Delta M$), les localiser dans le système matriciel du modèle.

[1] **Mrs. BOUARIF.K et HAMOUR.M**, conception et usinage des formes complexes, mémoire d'ingénieur (UMMTO), 2005.

[2] **Les piegl Wayne Tiller,** The NURBS book 2nd edition, 1996.

[3] **THOMAS.J.R.Hughes, J.A.cottrell,Y.Bazilevs.** Isogeometric analysis toward integration of CAD and FEA, 2010.

[4] **T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs.** Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. 2004

[5] **Prof. Dr. Eleni Chatzi,** The Finite Element Method for the Analysis of, Non-Linear and Dynamic Systems, 24 September, 2010.

[6] **Yijun Liu**, Introduction to Finite Element Method, université de Cincinnati, 1999.

[7] **BHAVYA AGGARWAL.** B-spline finite element for plane elasticity problem, MASTER OF SCIENCE, Texas A&M University, December 2006.

[8] **Jingang *Li*,** Isogeometric finite element analysis using T-Spline, THESIS, Brigham Young University, December 2009

[9] **Seung-Hyun H**, Isogeometric Shape Design Optimization Using NURBS Basis Functions **,** Thesis SEOUL National University, February 2010.

[10] A. Wall, Moritz A. Frenzel, Christian Cyron Isogeometric structural shape optimization Comput. Methods Appl. Mech. Engrg. 197 (2008) 2976–2988 Wolfgang

[11] **ALESSANDRO REALI,** An Isogeometric Analysis Approach for the Study of Structural Vibrations, Pavia, October 2004.

[12] **Jhabindra Prasad Ghimire,** An investigation of model validation and updating of existing structure for vibration-based structural change identification, Master of Engineering Saitama University Japan, February 2004.

[13] **Louis HUMBERT,** Recalage des modèles éléments finis à partir de mesures vibratoires, thèse l'école centrale de Lyon, 1999.

[14] **Jimin He**, Identification of structural dynamic characteristics, Imperial College, LONDON SW, October 1998.

[15]**Hervé Algrain,** recalage de modèle dynamique des structures dissipatrices, thèse université de Mons, janvier 2000.

[16] **Ney Roitman, Carlos Magluta, Luiz Augusto . Aragão Filho**
Structural Failure Localization Using Direct Update Methods, Graduate Institute of Federal University of Rio de Janeiro (COPPE/UFRJ), 1999.

Courbe de Bezier

```
clc
clear all
close all
x=[4 2 2 6 6 4];
y=[0 0.001 16 8 0.001 0]
n=length(x)-1;
t=0:0.0001:1;
bb=zeros(n,length(t));
cxx=zeros(1,length(t));
cyy=zeros(1,length(t));
for i=0:n;
    b=(factorial(n)/(factorial(i).*factorial(n-i))).*(t.^i).*((1-t).^(n-i));
    bb(i+1,1:length(t))=b;
    eval(['b' num2str(i) '=b']);
    cx=x(i+1).*b;
    cxx=cxx+cx;
    cy=y(i+1).*b;
    cyy=cyy+cy;
end;

plot(t,bb(1,1:length(t)));
for j=2:n+1;
    hold on;
    plot(t,bb(j,1:length(t)));
end;

%cx=b0*0+b1*2+b2*5+b.*8
%cy=b0*0+b1*4+b2*6+b3*5
%figure(2)
%plot(cx,cy)
%hold on
%plot(x,y,'-r')
figure(2)
plot(cxx,cyy)
hold on
plot(x,y,'--r')
grid
```

Fonctions de base NURBS, B-Spline et leurs dérivées

```
clear all;
close all;
clc
m=3;
n=8;
t=[0:0.01:1];
u=[0 0 0 0 1/6 1/3 1/2 2/3 5/6 1 1 1 1];
N=zeros(n+1,length(t));
for j=1:length(t)
    xi=t(j);
    for i=0:n
        if ((xi>=u(i+1))&(xi<u(i+2)))
            N(i+1,j)=1;
        else
```

```
            N(i+1,j)=0;
        end
    end;
    N(n+1,length(t))=1;
end;

for k=1:3;
    for j=1:length(t)
        xi=t(j);
        for i=0:n
            if (u(i+k+1)==u(i+1))
                a=0;
            else

                a=(((xi-u(i+1))/(u(i+k+1)-u(i+1)))*N(i+1,j));
            end;
            if k==m+1
                b=0;
            else
                if (u(i+k+2)==u(i+2))
                    b=0;
                else

                    b=(((u(i+k+2)-xi)/(u(i+k+2)-u(i+2)))*N(i+2,j));
                end;
            end;
               N(i+1,j)=a+b;
        end;
    end;
end;

for i=1:n;
    plot(t,N(i,:))
    hold on
end
title('fonctions de bases B-spline')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%derivé Bsplines
for k=1:3;
    for j=1:length(t)
        xi=t(j);
        for i=0:n;
            if (u(i+k+1)==u(i+1))
                c=0;
            else
                c=(k/(u(i+k+1)-u(i+1)))*N(i+1,j);
            end
            if k==m+1
                d=0;
            else
                if (u(i+k+2)==u(i+2))
                    d=0;
                else
                    d=(k/(u(i+k+2)-u(i+2)))*N(i+2,j);
                end
            end
```

```
            N1(i+1,j)=c+d;
        end
    end
end
figure(2)
for i=1:n;
    plot(t,N1(i,:))
    hold on
end
title('derivé fonctions de bases B-spline')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:3;
    for j=1:length(t)
        xi=t(j);
        for i=0:n;
            if (u(i+k+1)==u(i+1))
                c=0;
            else
                c=(k/(u(i+k+1)-u(i+1)))*N1(i+1,j);
            end
            if k==m+1
                d=0;
            else
                if (u(i+k+2)==u(i+2))
                    d=0;
                else
                    d=(k/(u(i+k+2)-u(i+2)))*N1(i+2,j);
                end
            end
            N2(i+1,j)=c+d;
        end
    end
end
figure(5)
for i=1:n;
    plot(t,N2(i,:))
    hold on
end
title('2eme derivé fonctions de bases B-spline')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%les poids les fonctions de bases NURBS (R(i,p))

w=[1 1 1 1 10 1 1 1 1];
DD=zeros(1,length(t));
DD1=zeros(1,length(t));
DD2=zeros(1,length(t));
for i=1:length(w)
    D=w(i)*N(i,:);
    D1=w(i)*N1(i,:);
    D2=w(i)*N2(i,:);
    DD=DD+D;
    DD1=DD1+D1;
     DD2=DD2+D2;
end
for i=1:length(w)
    C=w(i)*N(i,:);
```

```
    R(i,:)=C./DD;

end
figure(3)
for i=1:n;
    plot(t,R(i,:))
    hold on
end
title('fonctions de bases NURBS')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1 ere derivé des fonctions de bases R'

for i=0:n
    for j=1:length(t)
    R1(i+1,j)=w(i+1)*((DD*N1(i+1,j)-DD1*N(i+1,j))/DD.^2);
    end



end
figure(4)
for i=1:n;
    plot(t,R1(i,:))
    hold on
end
title('derivé fonctions de bases NURBS')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2 ere derivé des fonctions de bases R'
for i=0:n;
    C=w(i+1)*N(i+1,:);
    C1=w(i+1)*N1(i+1,:);
    C2=w(i+1)*N2(i+1,:);

R2(i+1,:)=((C2./DD(1,(1:length(t))))+((2.*C1(1,(1:length(t))).*(DD1(1,(1:lengt
h(t))).^2))./(DD(1,(1:length(t))).^3)))-
(((2.*C1(1,(1:length(t))).*DD1(1,(1:length(t))))+(C(1,(1:length(t))).*DD2))./(
DD(1,(1:length(t))).^2));

end
figure(6)
for i=1:n;
    plot(t,R2(i,:))
    hold on
end
title('2eme derivé fonctions de bases NURBS')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Algorithme de De Casteljau

```
clear all
close all
clc
B=zeros(4);
b=[0 4 10 12];
B(1:4,1)=b;
t=0:0.0001:1;
tt=zeros(1,length(t));
```

```
for k=1:length(t);
for i=1:3;
    for j=2:4;
        B(i,j)=(t(k).*B(i+1,j-1))+((1-t(k))*B(i,j-1));

    end;
end;
tt(k)=B(1,4);
end;
```

```
b2=[0 7 10 2];
B2(1:4,1)=b2;
t2=0:0.0001:1;
tt2=zeros(1,length(t));
for k=1:length(t);
for i=1:3;
    for j=2:4;
        B2(i,j)=(t2(k).*B2(i+1,j-1))+((1-t(k))*B2(i,j-1));

    end;
end;
tt2(k)=B2(1,4);
end;

plot(tt,tt2)

hold on
plot(b,b2,'r')

bb1=[(b(2)+b(1))/2 (b(3)+b(2))/2 (b(3)+b(4))/2];
bb2=[(b2(2)+b2(1))/2 (b2(3)+b2(2))/2 (b2(4)+b2(3))/2];
hold on
plot(bb1,bb2,'g')

bbb1=[(bb1(2)+bb1(1))/2 (bb1(3)+bb1(2))/2];
bbb2=[(bb2(2)+bb2(1))/2 (bb2(3)+bb2(2))/2];
hold on
plot(bbb1,bbb2,'k')
grid
```

programme de recalage avec la méthode CMM

```
clear all
close all
clc
p=1
n=10
t=0:0.001:1;
u=[0 0 1/9 2/9 3/9 4/9 5/9 6/9 7/9 8/9 1 1];
%u=[0 0 1/10 2/10 3/10 4/10 5/10 6/10 7/10 8/10 9/10 1 1]
for j=1:length(t)
```

```
    for i=1:n
        if t(j)>=u(i) & t(j)<u(i+1)
            N0(i,j)=1;
        else
            N0(i,j)=0;
        end
    end
    N0(n,length(t))=1;
end

N=zeros(n,length(t),p+1);
N(:,:,1)=N0;

for k=2:p+1
    for j=1:length(t)
        for i=1:n
            if u(i+k-1)-u(i)==0
                a=0;
            else
                a=((t(j)-u(i))/(u(i+k-1)-u(i)))*N(i,j,k-1);
            end
            if u(i+k+1-1)-u(i+1)==0
                b=0;
            else
                b=((u(i+k+1-1)-t(j))/(u(i+k+1-1)-u(i+1)))*N(i+1,j,k-1);
            end
            N(i,j,k)=a+b;
        end
    end
end
subplot(3,1,1)
for i=1:n
    plot(t,N(i,:,p+1))
    hold on
end
title('fonctions de base b-spline')

%**********Calcul de la premiere derivée DN*************

for j=1:length(t)
    for i=1:n
        if u(i+p)-u(i)==0
            a=0;
        else
            a=(p/(u(i+p)-u(i)))*N(i,j,p);
        end
        if u(i+p+1)-u(i+1)==0
            b=0;
        else
            b=(p/(u(i+p+1)-u(i+1)))*N(i+1,j,p);
        end
        DN(i,j)=a-b;
    end
end
subplot(3,1,2)
for i=1:n
```

```
    plot(t,DN(i,:))
    hold on
end

%**********Calcul de la deuxieme derivée D2N************
%A)Calcul de la premiere derivée au niveau p-1 (DNP)
for j=1:length(t)
    for i=1:n
        if u(i+p-1)-u(i)==0
            a=0;
        else
            a=((p-1)/(u(i+p-1)-u(i)))*N(i,j,p-1);
        end
        if u(i+p+1-1)-u(i+1)==0
            b=0;
        else
            b=((p-1)/(u(i+p+1-1)-u(i+1)))*N(i+1,j,p-1);
        end
        DNP(i,j)=a-b;
    end
end

%B)Calcul de la deuxieme derivée au niveau p

for j=1:length(t)
    for i=1:n
        if u(i+p)-u(i)==0
            a=0;
        else
            a=(p/(u(i+p)-u(i)))*DNP(i,j);
        end
        if u(i+p+1)-u(i+1)==0
            b=0;
        else
            b=(p/(u(i+p+1)-u(i+1)))*DNP(i+1,j);
        end
        D2N(i,j)=a-b;
    end
end

subplot(3,1,3)
for i=1:n
    plot(t,D2N(i,:))
    hold on
end



%***********Calcul des fonctions de base NURBS R***********
%w: vecteur des poids
w=[1 1 1 1 1 1 1 1 1 1]
for j=1:length(t)
    denom=0;
    for i=1:n
        a=N(i,j,p+1)*w(i);
```

```
        denom=denom+a;
    end
    for i=1:n
        R(i,j)=(N(i,j,p+1)*w(i))/denom;
    end
end
figure(2)
subplot(3,1,1)
for i=1:n
    plot(t,R(i,:))
    hold on
end
title('fonctions de base NURBS')

%***********Calcul de la premiere dirivée des NURBS DR**********
for j=1:length(t)
    W1=0;
    W2=0;
    for i=1:n
        V1=N(i,j,p+1)*w(i);
        V2=DN(i,j)*w(i);
        W1=W1+V1;
        W2=W2+V2;
    end
    for i=1:n
        DR(i,j)=w(i)*(((W1*DN(i,j))-(W2*N(i,j,p+1)))/(W1^2));
    end
end
subplot(3,1,2)
for i=1:n
    plot(t,DR(i,:))
    hold on
end

%***********Calcul de la deuxieme dirivée des NURBS D2R**********
for j=1:length(t)
    W1=0;
    W2=0;
    W3=0;
    for i=1:n
        V1=N(i,j,p+1)*w(i);
        V2=DN(i,j)*w(i);
        V3=D2N(i,j)*w(i);

        W1=W1+V1;
        W2=W2+V2;
        W3=W3+V3;
    end
    for i=1:n

D2R(i,j)=((D2N(i,j)*w(i))/W1)+((2*N(i,j,p+1)*w(i)*(W2^2))/(W1^3))+(((2*N(i,j,p
+1)*w(i)*W2)+(N(i,j,p+1)*w(i)*W3))/(W1^2));
    end
end
subplot(3,1,3)
for i=1:n
```

```
    plot(t,D2R(i,:))
    hold on
end

%************************************************
%************************ke1*********************
%*************calcul du jacobien jac**************
%************************************************
%x:le polygone de control
L=50;
x=[0;(1/9)*L;(2/9)*L;(3/9)*L;(4/9)*L;(5/9)*L;(6/9)*L;(7/9)*L;(8/9)*L;L];
t1=0:0.001:1/9;

jac=zeros(1,length(t1));
for i=1:2
    jacobian=x(i)*DR(i,:);
    jac=jac+jacobian(i);
end

%***********Construction de la matrice de rigidité**************
%integration par lamethode des trapèses%%%%%%%%%%%%%%%%%%%%%%%%%
hh=0.001;
E=21e2;
EA=E*25;
for j=1:length(t1)
    for i=1:2
        for k=1:2
            F(i,j,k)=DR(i,j)*DR(k,j)*(jac(j)^-1);
        end
    end
end
B=zeros(2,2);
for j=2:length(t1)-1
    A=F(:,j,:);
    B(:,:)=B(:,:)+A(:,:);
end

for i=1:2
    for j=1:2
        G(i,j)=F(i,1,j);
        H(i,j)=F(i,length(t1),j);
    end
end
ke1=EA*(hh/2)*(G+(2*B)+H);

%***************KG raideur global%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KG=zeros(10);
for i=1:9
    K=zeros(10);
    K(i:i+1,i:i+1)=ke1;
    KG=KG+K;
end
KG
%force
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%masse%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KG(1,:)=zeros(1,10);
hh=0.001;
S=4e-4;
rho=2800;
for j=1:length(t1)
    for i=1:2
        for k=1:2
            F(i,j,k)=(R(i,j)*R(k,j))*(jac(j));
        end
    end
end
B=zeros(2,2);
for j=2:length(t1)-1
    A=F(:,j,:);
    B(:,:)=B(:,:)+A(:,:);
end

for i=1:2
    for j=1:2
        G(i,j)=F(i,1,j);
        H(i,j)=F(i,length(t1),j);
    end
end
Me1=rho*S*(hh/2)*(G+(2*B)+H);
%%%%%%%%%%%%%%%%%masse global%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KG(1,:)=zeros(1,10);

KG(:,1)=zeros(10,1);

MG=zeros(10);
for i=1:9
    M=zeros(10);
    M(i:i+1,i:i+1)=Me1;
    MG=MG+M;
end
MG
%%%%%%%%%%%%%%%%%%C.L %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

KG(1,:)=[];
KG(:,1)=[];
MG(1,:)=[];
MG(:,1)=[];
[u,v]=eig(KG,MG);
%%%%%%%%%%%%%%%%%%%%%%normalisation de vecteur propre%%%%%%%%%%%%%%%%%%%%%%
Vec_Norm=[];
den=length(u);
for ij=1:den;
     alR=u(:,ij)'*MG*u(:,ij);
     V_R=u(:,ij)/sqrt(alR);
     Vec_Norm=[Vec_Norm V_R];
```

```
end
%%%%%%%%%%%%%%mode%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
  Ms=Vec_Norm'*MG*Vec_Norm;
  Ks=Vec_Norm'*KG*Vec_Norm;
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%masse perturbé%%%%%%%%%%ùGGFF
M1=zeros(10,10);
M1(1:2,1:2)=Me1;
M2=zeros(10,10);
M2(2:3,2:3)=0.3*Me1;
M3=zeros(10,10);
M3(3:4,3:4)=Me1;
M4=zeros(10,10);
M4(4:5,4:5)=Me1;
M5=zeros(10,10);
M5(5:6,5:6)=Me1;
M6=zeros(10,10);
M6(6:7,6:7)=Me1;
M7=zeros(10,10);
M7(7:8,7:8)=Me1;
M8=zeros(10,10);
M8(8:9,8:9)=Me1;
M9=zeros(10,10);
M9(9:10,9:10)=Me1;

MG1=M1+M2+M3+M4+M5+M6+M7+M8+M9;
%%%%%%%%%%%%%%%%%%%%%%%%%%raideur perturbé%%%%%%%%%%%%%%%%%%
K1=zeros(10,10);
K1(1:2,1:2)=ke1;
K2=zeros(10,10);
K2(2:3,2:3)=ke1;
K3=zeros(10,10);
K3(3:4,3:4)=ke1;
K4=zeros(10,10);
K4(4:5,4:5)=0.2*ke1;
K5=zeros(10,10);
K5(5:6,5:6)=ke1;
K6=zeros(10,10);
K6(6:7,6:7)=ke1;
K7=zeros(10,10);
K7(7:8,7:8)=ke1;
K8=zeros(10,10);
K8(8:9,8:9)=ke1;
K9=zeros(10,10);
K9(9:10,9:10)=ke1;

KG1=K1+K2+K3+K4+K5+K6+K7+K8+K9;

%%%%%%%%%%vecteur valeur propre systeme perturbé%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%C.L %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KG1(1,:)=[];
KG1(:,1)=[];
```

```
MG1(1,:)=[];
MG1(:,1)=[];
[u1,v1]=eig(KG1,MG1);
%%%%%%%%%%%%%%%%%%%%%%%%calcul delta M et deltaK %%%%%%%%%%%%%
%delM=MG*u*(2*eye(10)-u'*MG*u-u1'*MG*u)*u'*MG;
%delK=KG*(u*inv(v)-u1*inv(v1)*u1')*KG
%-u1'*KG*u)*u'*MG;
%%%%%%%%%%%%%%%%%%%%%%%%calcule de delta K%%%%%%%%%%%%%%%%%%%%
%*************************calcul *************************
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mg=u1'*MG*u1;
MX=MG+MG*u1*inv(mg)*(eye(9)-mg)*inv(mg)*u1'*MG;
KX=KG-KG*u1*u1'*MX-MX*u1*u1'*KG+MX*u1*v1*u1'*MX+MX*u1*u1'*KG*u1*u1'*MX;
Mbar=u1'*MG*u1;
deltaM=MX*u1*inv(Mbar)*(eye(9)-Mbar)*inv(Mbar)*u1'*MX
deltaK=-KG*u1*u1'*MX-MX*u1*u1'*KG+MX*u1*v1*u1'*MX+MX*u1*u1'*KG*u1*u1'*MX
%Kprime=MG*u1*v1/u1
%***********resultats pour la raideur*****************
figure(3)
for i=1:9
    bar3(abs(KG))
end
title('matrice raideur original')
figure(4)
for i=1:9
    bar3(abs(KX))
end
title('matrice raideur perturbé')
figure(5)
for i=1:9
    bar3(abs(deltaK))
end

title('defaut de raideur')
%***********resultats pour la masse*****************
figure(6)
for i=1:9
    bar3(abs(MG))
end
title('matrice masse original')
figure(7)
for i=1:9
    bar3(abs(MX))
end
title('matrice masse perturbé')
figure(8)
for i=1:9
    bar3(abs(deltaM))
end

title('defaut de masse')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psetaerreurK=(abs(deltaK)./abs(KG))*100
figure(9)
```

```
bar3(psetaerreurK)

psetaerreurM=(deltaM./MG)*100
figure(10)
bar3(abs(psetaerreurM))

ff=zeros(10,10);
for i=1:10
    f=((u1(:,i)-u(:,i))/u(:,i))*100;
    ff=ff+f;
end
```