



HAL
open science

Interpolation et approximation de données à l'aide de courbes et surfaces paramétriques de type B-splines

Alexandra Valdescault, Alain Batailly, Simon Jones

► **To cite this version:**

Alexandra Valdescault, Alain Batailly, Simon Jones. Interpolation et approximation de données à l'aide de courbes et surfaces paramétriques de type B-splines. 2012. hal-00724596

HAL Id: hal-00724596

<https://hal.science/hal-00724596>

Submitted on 21 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Note technique

Interpolation et approximation de données à l'aide de courbes et surfaces paramétriques de type **B-splines**

A. Valdescault¹, A. Batailly² & S. Jones³



Université McGill

Laboratoire de dynamique des structures et vibrations

août 2012

1. alexandra.valdescault@mail.mcgill.ca
2. alain.batailly@mcgill.ca
3. jones5@rose-hulman.edu

Table des matières

1	Courbes paramétriques	5
1.1	Généralités	5
1.1.1	Définitions	5
1.2	B-Splines	6
1.2.1	Approximation	7
1.2.2	Interpolation	8
1.2.3	Conditions limites	9
2	Surfaces paramétriques	13
2.1	Généralités	13
2.2	Convention adoptée	15
2.3	Tracé	15
2.4	Approximation	17
2.5	Interpolation	18
2.5.1	Surface ouverte	19
2.5.2	Surface fermée	22
3	Algorithme de détection	24
3.1	Procédure itérative de Newton-Raphson	24
3.2	Cas des surfaces fermées	25
3.3	Validation	26
4	Code Matlab	28
5	Conclusion	37
6	Annexe	38
6.1	Courbe paramétrique cubique	38
6.2	B-Splines de degré 1	38
6.2.1	Conditions limites	39
6.3	B-Spline de degré 2	40
6.3.1	Conditions limites	40
6.4	B-Splines de degré 4	42
6.4.1	Conditions limites	43
6.5	NURBS : <i>Non-Rational Uniform B-Splines</i>	47

Introduction

Cette note technique a pour objectif de fournir une description détaillée des opérations élémentaires à suivre dans la construction de courbes et surfaces splines paramétriques pour l'approximation ou l'interpolation de données dans le plan ou l'espace. Dans cette note technique, le terme d'*approximation* de points de données par une courbe paramétrique est utilisé lorsque la courbe peut ne pas passer par les points de données. Inversement, le terme d'*interpolation* de points de données par une courbe paramétrique est utilisé lorsque la courbe est construite de façon à passer par tous les points de données.

Une présentation générale des B-Splines en 2D et 3D est aussi donnée. La plupart des codes Matlab nécessaires à la réalisation des exemples présentés sont fournis dans la section 4. Ces codes sont généraux et permettent de créer une courbe ou une surface paramétrique en utilisant une B-spline (bi)cubique pour tout ensemble de points \mathbf{P} du plan tel que celui représenté sur la figure 1.

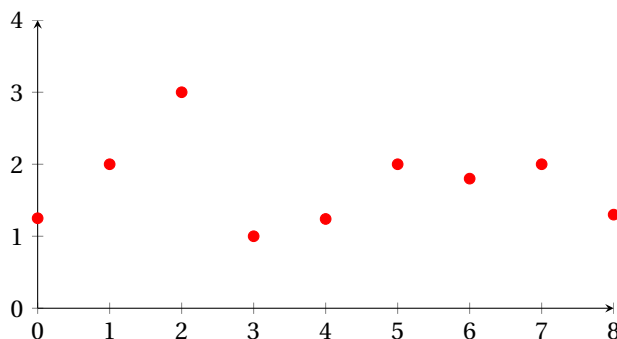


Figure 1 – Ensemble de coordonnées (x, y)

Dans le cadre de l'étude de problèmes mécaniques, plusieurs méthodes telles que la méthode des éléments finis sont utilisées pour la modélisation de structures et de surfaces. Dans le cas particulier des simulations de contact, lorsqu'un maillage éléments finis linéaire est utilisé, la discontinuité de la normale à la surface de contact peut engendrer des instabilités numériques ou même la divergence de la simulation. L'utilisation d'éléments quadratiques ou d'éléments isogéométriques peut s'avérer très coûteux en temps de calculs, c'est pourquoi l'utilisation de surfaces paramétriques – construites par interpolation des nœuds du maillage présents sur la surface de contact – pour la représentation de la surface de contact est souvent intéressante.

Dans le cas général de la résolution de problèmes mécaniques dans le cadre des grandes déformations, la représentation de la surface de contact peut s'avérer délicate. Souvent, les courbes ou surfaces paramétriques sont construites en supposant un cadre particulier (déplacements de certains nœuds limités à une direction, petites déformations...) ce qui permet de réduire les temps de calculs.

L'objectif de cette note est de présenter en détails la construction de courbes et surfaces paramétriques (essentiellement des splines uniformes) à partir d'un ensemble de points de l'espace ou du plan. La seule hypothèse faite concerne la structure des points de données $\mathbf{P} = (\mathbf{P}_x; \mathbf{P}_y; \mathbf{P}_z)$ à interpoler (tels que les nœuds d'un maillage éléments finis). Il est supposé qu'il existe une transformation ϕ de l'espace telle que l'ensemble des points $(\hat{\mathbf{P}}_x; \hat{\mathbf{P}}_y; \hat{\mathbf{P}}_z) = (\phi\mathbf{P}_x; \phi\mathbf{P}_y; \phi\mathbf{P}_z)$ appartiennent au plan $(O; \vec{x}; \vec{y})$ et définissent un domaine rectangulaire au sein duquel ils sont régulièrement espacés comme schématisé sur la figure 2.

Concrètement, l'existence de la transformation ϕ est garantie dès qu'il est possible d'écrire les coordonnées des points de données \mathbf{P} sous forme matricielle : \mathbf{P}_x , \mathbf{P}_y et \mathbf{P}_z , chacune de ces matrices possédant $(n - 2)$ lignes et $(m - 2)$ colonnes (les paramètres n et m sont explicités dans la suite de cette note).

Par ailleurs, comme mentionné précédemment, les développements présentés dans cette note sont faits en envisageant des simulations de contact en grandes déformations. Outre la modélisation de la surface de contact, un aspect fondamental des simulations de contact concerne la détection de la distance entre tout point de l'espace (par exemple un nœud du maillage de la structure impactante) et la surface de contact. C'est pour cette raison que la section 3 présente un algorithme de détection du point

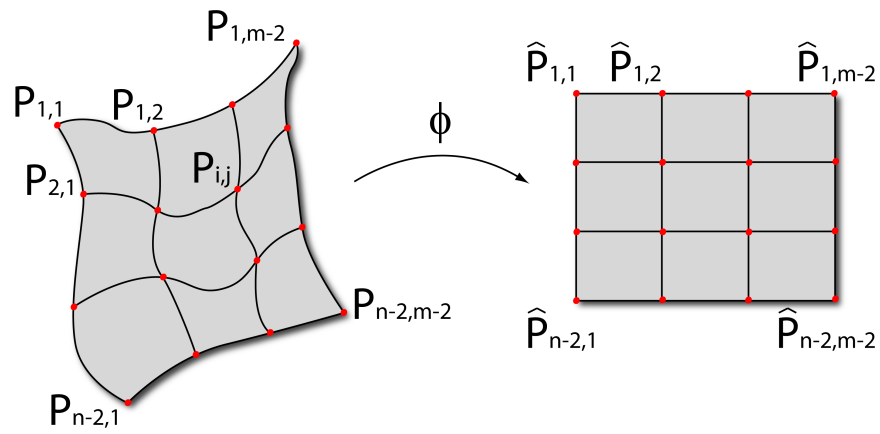


Figure 2 – Hypothèse d'organisation des points de données

le plus proche sur la spline pour tout point de l'espace. Cet algorithme repose sur une procédure itérative de type Newton-Raphson qui est confrontée, dans la section 3.3, à une stratégie d'optimisation afin de valider les résultats obtenus.

1 Courbes paramétriques

1.1 Généralités

Les courbes paramétriques peuvent être définies de façon assez générale par l'équation suivante :

$$c(\kappa) = (x(\kappa); y(\kappa)) = \sum_{j=1}^n \mathbf{Q}_j B_{j,d}(\kappa) \quad (1)$$

où

- \mathbf{Q}_j sont les points de contrôle ;
- $B_{j,d}$ sont des fonctions polynômiales de degré d dont l'influence est localisée à un certain nombre de secteurs élémentaires, appelés patches ;
- d est le degré de la courbe paramétrique ;
- n est le nombre total de points de contrôle.

La courbe c est définie sur un intervalle $\kappa \in [a; b]$.

1.1.1 Définitions

1. **Paramètre intrinsèque** : nom donné au paramètre κ dont la valeur évolue dans l'intervalle $[a; b]$ lorsqu'on parcourt la courbe paramétrique ;

2. **Vecteur de nœuds** :

- Il est représenté par la variable $\mathbf{t} = [t_1 \ t_2 \ t_3 \ \dots \ t_{n+d-1} \ t_{n+d} \ t_{n+d+1}]$ où t_i est appelé *nœud* et $t_{h-1} < t_h < t_{h+1} \dots$, avec : $((d+1) \leq h \leq n)$ et inclut une partition de l'intervalle $[a; b]$, c'est-à-dire $t_{d+2} = a$ et $t_{n+d-1} = b$;
- Chaque intervalle de nœuds $[t_h, t_{h+1}]$ détermine le début et l'arrêt d'un patch (secteur élémentaire de la courbe paramétrique) ;

Il existe plusieurs types de vecteurs de nœuds : uniforme, non-uniforme, périodique et non-périodique. Les vecteurs de nœuds de type uniforme et périodique sont caractérisés par les valeurs de nœuds qui sont uniformément espacées. Les vecteurs de nœuds de type non-uniforme sont surtout utilisés dans les B-Splines non-uniformes et rationnels (NURBS : *Non-Uniform and Rational B-Splines*).

3. **Patch** : Un patch est l'entité élémentaire de la courbe paramétrique correspondant à $\kappa \in [t_i; t_{i+1}]$, $i = d+2, \dots, n+d-1$;

4. **Points de contrôle** :

- ils sont représentés par la variable \mathbf{Q}_j , $j = 1, \dots, n$;
- leurs positions influencent la forme de la courbe paramétrique c comme le montre l'équation (1) ;
- le nombre total de points de contrôle est lié au degré de la courbe paramétrique : $n = d+1$.

Le tableau suivant présente le nombre requis de points de contrôle pour tracer un patch d'une courbe paramétrique selon l'ordre de la spline.

Degré	Ordre	Nombre de points de contrôle
2	3	$n = 3$
3	4	$n = 4$
4	5	$n = 5$

5. **Points de données** :

- ils sont représentés par la variable \mathbf{P} ;
- pour un vecteur de nœuds donné, $\mathbf{t} = [t_1 \ t_2 \ t_3 \ \dots \ t_{n+d-1} \ t_{n+d} \ t_{n+d+1}]$, les $n-d+1$ points de données sont définis comme :

$$\mathbf{P}_i = c(t_{i+d+1}), \quad i = 1, \dots, n-d+1 \quad (2)$$

- le nombre de points de données est lié au nombre de points de contrôle n et au degré de la courbe d et vaut : $n+1-d$.

6. **Domaine effectif** : ce domaine correspond à l'intervalle $[a; b] = [t_{d+2}; t_{n+d-1}]$ sur lequel l'équation (1) est définie.

1.2 B-Splines

Les fonctions polynômiales $B_{j,d}$ permettant de créer une courbe spline sont définies par la formule réursive Cox-de-Boor :

$$B_{j,0}(\kappa) = \begin{cases} 1 & \text{si } \kappa \in [t_h, t_{h+1}] \\ 0 & \text{si } \kappa \notin [t_h, t_{h+1}] \end{cases} \quad (3)$$

$$B_{j,d}(\kappa) = \begin{cases} \frac{\kappa - t_j}{t_{j+d} - t_j} B_{j,d-1}(\kappa) + \frac{t_{j+d+1} - \kappa}{t_{j+d+1} - t_{j+1}} B_{j+1,d-1}(\kappa) & \text{si } \kappa \in [t_h, t_{h+1}] \\ 0 & \text{si } \kappa \notin [t_h, t_{h+1}] \end{cases} \quad (4)$$

Si la valeur de κ appartient à l'intervalle de nœuds $[t_h, t_{h+1}]$, alors la fonction spline $B_{j,0}$ est non-nulle. Dans le cas contraire, elle est nulle. Dans cette note technique, nous utiliserons un vecteur de nœuds de type uniforme.

$$\begin{aligned} \mathbf{t} &= [t_1 \ t_2 \ t_3 \ \dots \ t_{n+d-1} \ t_{n+d} \ t_{n+d+1}] \\ &= [1 \ 2 \ 3 \ \dots \ n+d-1 \ n+d \ n+d+1] \end{aligned} \quad (5)$$

À partir de de l'équation (4), il est possible d'établir la formule générale pour une B-Spline cubique ($d = 3$).

$$\begin{aligned} B_{j,3}(\kappa) &= \frac{\kappa - t_j}{t_{j+3} - t_j} \left[\frac{\kappa - t_j}{t_{j+2} - t_j} \left(\frac{\kappa - t_j}{t_{j+1} - t_j} B_{j,0}(\kappa) + \frac{t_{j+2} - \kappa}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) \right) \right. \\ &\quad \left. + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+1}} \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) \right) \right] \\ &\quad + \frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+1}} \left[\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) \right) \right. \\ &\quad \left. + \frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) + \frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} B_{j+3,0}(\kappa) \right) \right] \end{aligned} \quad (6)$$

En tenant compte du domaine de définition des fonctions $B_{j,0}$ donné dans l'équation (3) et pour $\kappa \in [t_{h+1}, t_{h+2}]$ inclus dans le domaine effectif, l'équation (1) se réduit aux indices :

$$j = \begin{cases} h-3 \\ h-2 \\ h-1 \\ h \end{cases} \quad (7)$$

et il reste

$$\boxed{\kappa \in [t_{h+1}, t_{h+2}] \text{ où } 4 \leq h \leq n, \quad \mathbf{c}(\kappa) = B_{h-3,3}(\kappa)\mathbf{Q}_{h-3} + B_{h-2,3}(\kappa)\mathbf{Q}_{h-2} + B_{h-1,3}(\kappa)\mathbf{Q}_{h-1} + B_{h,3}(\kappa)\mathbf{Q}_h} \quad (8)$$

L'expression développée de l'équation (8) pour une courbe spline de degré 3 et pour l'intervalle $[t_{h+1}, t_{h+2}]$ peut être trouvée en annexe, dans la section 6.1.

En substituant κ par $t_h + \kappa$ et en supposant que le vecteur de nœuds est uniforme, il est possible d'exprimer les différentes fonctions splines sous la forme polynômiale suivante

$$\begin{aligned} \mathbf{c}(\kappa) &= \sum_{j=1}^n \mathbf{Q}_j B_{j,3}(\kappa) \\ &= B_{h-3,3}(\kappa)\mathbf{Q}_{h-3} + B_{h-2,3}(\kappa)\mathbf{Q}_{h-2} + B_{h-1,3}(\kappa)\mathbf{Q}_{h-1} + B_{h,3}(\kappa)\mathbf{Q}_h \\ &= \frac{1}{6} \left[(1 - \kappa)^3 \mathbf{Q}_{h-3} + (3\kappa^3 - 6\kappa^2 + 4) \mathbf{Q}_{h-2} + (-3\kappa^3 + 3\kappa^2 + 3\kappa + 1) \mathbf{Q}_{h-1} + \kappa^3 \mathbf{Q}_h \right] \end{aligned} \quad (9)$$

Ce qui nous ramène à la forme matricielle suivante

$$\mathbf{c}(\kappa) = \frac{1}{6} \begin{bmatrix} \kappa^3 & \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (10)$$

où $\kappa \in [t_{h+1}, t_{h+2}]$ pour $4 \leq h \leq n$.

Les points de contrôle \mathbf{Q} peuvent être vus comme des points de l'espace \mathbf{P} à approcher et alors l'équation (10) permet de calculer une approximation de ces points par la courbe \mathbf{c} . L'approximation (voir figure 3) d'une série de points du plan ne pose donc aucune difficulté particulière et est présentée dans la section 1.2.1.

En revanche, si la courbe \mathbf{c} doit être construite pour réaliser une interpolation (voir figure 4) d'un ensemble de points de données \mathbf{P} , il faut choisir prudemment les points de contrôle \mathbf{Q} et la démarche à adopter est présentée dans la section 1.2.2.

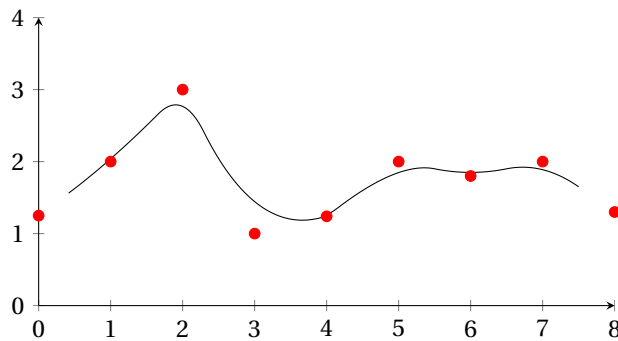


Figure 3 – Approximation des données avec courbes splines

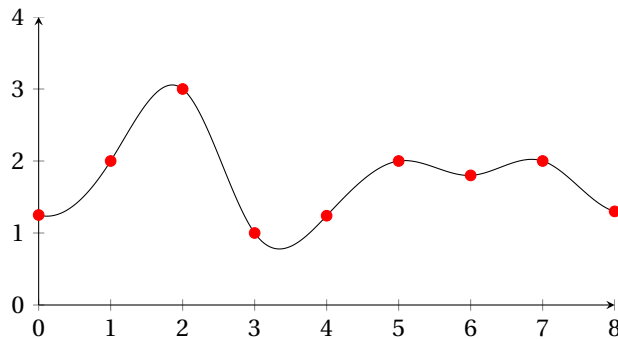


Figure 4 – Interpolation des données avec courbes splines

1.2.1 Approximation

Pour une approximation d'un ensemble de points de données \mathbf{P} , il suffit de choisir les points de contrôle $\mathbf{Q} = \mathbf{P}$ et de tracer la courbe paramétrique en utilisant les équations données dans les sections précédentes. Par exemple, pour une spline de degré 3, la courbe est donnée par

$$\mathbf{c}(\kappa) = \frac{1}{6} \begin{bmatrix} \kappa^3 & \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{h-3} \\ \mathbf{P}_{h-2} \\ \mathbf{P}_{h-1} \\ \mathbf{P}_h \end{bmatrix} \quad (11)$$

où $\kappa \in [t_{h+1}, t_{h+2}]$ pour $4 \leq h \leq n$.

1.2.2 Interpolation

Pour une interpolation d'un ensemble de points de données \mathbf{P} , les points de contrôle \mathbf{Q} doivent être calculés de façon particulière.

Par construction, il y a nécessairement plus de points de contrôle que de points de données (sauf si on considère une interpolation de degré 1 qui ne présente pas d'intérêt pour les applications envisagées). Dans le cas d'une spline cubique, s'il y a $n - 2$ points de données, il faut trouver n points de contrôle.

Les n équations sont obtenues grâce aux $n - 2$ équations traduisant le passage de la courbe par les points de données auxquelles s'ajoutent deux équations associées aux conditions limites à *gauche* et à *droite* de la courbe.

Les dérivées première et seconde de $\mathbf{c}(\kappa)$ sont obtenues grâce à l'équation (10). Elles nous seront utiles dans la détermination des conditions limites et se présentent sous la forme matricielle suivante

$$\begin{aligned}\dot{\mathbf{c}}(\kappa) &= \frac{1}{6} \begin{bmatrix} \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} -3 & 9 & -9 & 3 \\ 6 & -12 & 6 & 0 \\ -3 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \\ \ddot{\mathbf{c}}(\kappa) &= \frac{1}{6} \begin{bmatrix} \kappa & 1 \end{bmatrix} \begin{bmatrix} -6 & 18 & -18 & 6 \\ 6 & -12 & 6 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix}\end{aligned}\quad (12)$$

où $\kappa \in [t_{h+1}, t_{h+2}]$ pour $4 \leq h \leq n$ et où le \cdot traduit la dérivée de \mathbf{c} par rapport au paramètre κ . Pour chaque point de données

$$\begin{aligned}\mathbf{P}_1 &= \mathbf{c}(t_5) = \frac{1}{6}(\mathbf{Q}_1 + 4\mathbf{Q}_2 + \mathbf{Q}_3) \\ \mathbf{P}_2 &= \mathbf{c}(t_6) = \frac{1}{6}(\mathbf{Q}_2 + 4\mathbf{Q}_3 + \mathbf{Q}_4)\end{aligned}\quad (13)$$

Ce qui peut s'écrire sous la forme

$$\begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-3} \\ \mathbf{P}_{n-2} \end{pmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 & \dots & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 & 0 \\ 0 & \dots & & 0 & 1 & 4 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{pmatrix}\quad (14)$$

Les deux équations manquantes sont associées aux conditions limites et on suppose que

$$\begin{aligned}\dot{\mathbf{c}}(t_5) &= \dot{\mathbf{c}}(t_{n+2}) \\ \ddot{\mathbf{c}}(t_5) &= \ddot{\mathbf{c}}(t_{n+2})\end{aligned}\quad (15)$$

où la courbe paramétrique débute à t_5 et prend fin à t_{n+2} .

L'équation (15) est équivalente à

$$\begin{aligned}-\mathbf{Q}_1 + \mathbf{Q}_3 &= -\mathbf{Q}_{n-2} + \mathbf{Q}_n \\ \mathbf{Q}_1 - 2\mathbf{Q}_2 + \mathbf{Q}_3 &= \mathbf{Q}_{n-2} - 2\mathbf{Q}_{n-1} + \mathbf{Q}_n\end{aligned}\quad (16)$$

De cette façon, il est possible de compléter l'équation (14) et d'obtenir la relation entre les points de

données \mathbf{P} et points de contrôle \mathbf{Q}

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-3} \\ \mathbf{P}_{n-2} \\ \mathbf{0} \end{pmatrix} = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 & 0 & \dots & 1 & 0 & -1 \\ 1 & 4 & 1 & 0 & \dots & & & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \dots & 0 & 1 & 4 & 1 & & 0 \\ 0 & \dots & & 0 & 1 & 4 & 1 & \\ 1 & -2 & 1 & 0 & \dots & -1 & 2 & -1 \end{bmatrix} \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \vdots \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{pmatrix} \quad (17)$$

L'équation (17) s'écrit plus simplement

$$\mathbf{P} = \frac{1}{6} \Phi \mathbf{Q} \quad (18)$$

et les points de contrôle \mathbf{Q} sont simplement obtenus en calculant

$$\mathbf{Q} = 6\Phi^{-1}\mathbf{P} \quad (19)$$

Les points de contrôle correspondant aux points de passage donnés étant désormais déterminés, il est possible de tracer la courbe paramétrique résultante en utilisant soit l'équation (10) ou l'équation (8).

1.2.3 Conditions limites

Dans la section 1.2.2, nous avons démontré qu'il était nécessaire d'établir des conditions limites pour trouver les points de contrôle correspondant aux points de passage. Toutefois, qu'il s'agisse d'une interpolation ou d'une approximation, si souhaitée, des conditions sur la position, la pente de première et deuxième dérivées peuvent être imposées aux extrémités d'une courbe spline cubique. La procédure qui suit présente plus en détails les étapes à suivre pour appliquer ces conditions. Il est à noter que les conditions limites pour les B-Splines de degré 1, 2 et 4 peuvent être retrouvées dans les sections (6.2), (6.3) et (6.4) en annexe de cette note.

Méthodologie

Après avoir substitué κ par $\kappa + t_n$ dans l'équation (8), permettant ainsi d'obtenir l'équation (9), la courbe paramétrique cubique débutera à $t_5 = 0$ et prendra fin à $t_{n+2} = 1$.

1. Conditions limites pour la position

$$\mathbf{c}(t_5) = \frac{1}{6} \begin{bmatrix} t_5^3 & t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \end{bmatrix} \quad (20)$$

$$\boxed{\mathbf{c}(t_5) = \frac{1}{6} (\mathbf{Q}_1 + 4\mathbf{Q}_2 + \mathbf{Q}_3)} \quad (21)$$

$$\mathbf{c}(t_{n+2}) = \frac{1}{6} \begin{bmatrix} t_{n+2}^3 & t_{n+2}^2 & t_{n+2} & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (22)$$

$$\boxed{\mathbf{c}(t_{n+2}) = \frac{1}{6} (\mathbf{Q}_{n-2} + 4\mathbf{Q}_{n-1} + \mathbf{Q}_n)} \quad (23)$$

Des deux équations obtenues, six inconnus doivent être déterminés : $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \mathbf{Q}_{n-2}, \mathbf{Q}_{n-1}$ et \mathbf{Q}_n . Pour ce faire, nous devons aussi considérer la dérivée première et seconde de $\mathbf{c}(\kappa)$.

2. Conditions limites pour la dérivée première

$$\dot{\mathbf{c}}(\kappa) = \frac{1}{6} \begin{bmatrix} 3\kappa^2 & 2\kappa & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (24)$$

$$\dot{\mathbf{c}}(t_5) = \frac{1}{6} \begin{bmatrix} t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} -3 & 9 & -9 & 3 \\ 6 & -12 & 6 & 0 \\ -3 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \end{bmatrix} \quad (25)$$

$$\boxed{\dot{\mathbf{c}}(t_5) = \frac{1}{2} (-\mathbf{Q}_1 + \mathbf{Q}_3)} \quad (26)$$

$$\dot{\mathbf{c}}(t_{n+2}) = \frac{1}{6} \begin{bmatrix} t_{n+2}^2 & t_{n+2} & 1 \end{bmatrix} \begin{bmatrix} -3 & 9 & -9 & 3 \\ 6 & -12 & 6 & 0 \\ -3 & 0 & 3 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (27)$$

$$\boxed{\dot{\mathbf{c}}(t_{n+2}) = \frac{1}{2} (-\mathbf{Q}_{n-2} + \mathbf{Q}_n)} \quad (28)$$

3. Conditions limites pour la dérivée seconde

$$\ddot{\mathbf{c}}(\kappa) = \frac{1}{6} \begin{bmatrix} 6\kappa & 2 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (29)$$

$$\ddot{\mathbf{c}}(t_5) = \frac{1}{6} \begin{bmatrix} t_5 & 1 \end{bmatrix} \begin{bmatrix} -6 & 18 & -18 & 6 \\ 6 & -12 & 6 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \end{bmatrix} \quad (30)$$

$$\boxed{\ddot{\mathbf{c}}(t_5) = (\mathbf{Q}_1 - 2\mathbf{Q}_2 + \mathbf{Q}_3)} \quad (31)$$

$$\ddot{\mathbf{c}}(t_{n+2}) = \frac{1}{6} \begin{bmatrix} t_{n+2} & 1 \end{bmatrix} \begin{bmatrix} -6 & 18 & -18 & 6 \\ 6 & -12 & 6 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (32)$$

$$\boxed{\ddot{\mathbf{c}}(t_{n+2}) = (\mathbf{Q}_{n-2} - 2\mathbf{Q}_{n-1} + \mathbf{Q}_n)} \quad (33)$$

Des six équations obtenues, il est maintenant possible de résoudre pour les six inconnus. Ceci requiert toutefois d'assigner des valeurs spécifiques aux variables $\mathbf{c}(t_5)$, $\dot{\mathbf{c}}(t_5)$, $\ddot{\mathbf{c}}(t_5)$, $\mathbf{c}(t_{n+2})$, $\dot{\mathbf{c}}(t_{n+2})$ et $\ddot{\mathbf{c}}(t_{n+2})$. Ces valeurs déterminent les conditions limites à imposer sur la position, la dérivée première et la dérivée seconde. Les systèmes d'équations à résoudre peuvent être présentés sous la forme matricielle suivante.

$$\begin{bmatrix} \mathbf{c}(t_5) \\ \dot{\mathbf{c}}(t_5) \\ \ddot{\mathbf{c}}(t_5) \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 \\ -3 & 0 & 3 \\ 6 & -12 & 6 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \end{bmatrix} \quad (34)$$

$$\begin{bmatrix} \mathbf{c}(t_{n+2}) \\ \dot{\mathbf{c}}(t_{n+2}) \\ \ddot{\mathbf{c}}(t_{n+2}) \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 \\ -3 & 0 & 3 \\ 6 & -12 & 6 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix}$$

À partir de l'équation (26), il est possible d'isoler \mathbf{Q}_3 :

$$\mathbf{Q}_3 = 2\dot{\mathbf{c}}(t_5) + \mathbf{Q}_1$$

En substituant l'équation (26) dans l'équation (21), nous obtenons

$$3\mathbf{c}(t_5) = \mathbf{Q}_1 + 2\mathbf{Q}_2 + \dot{\mathbf{c}}(t_5) \quad (35)$$

En substituant l'équation (26) dans l'équation (31), nous obtenons

$$\ddot{\mathbf{c}}(t_5) = 2\mathbf{Q}_1 - 2\mathbf{Q}_2 + 2\dot{\mathbf{c}}(t_5) \quad (36)$$

En utilisant l'équation (35) ainsi que l'équation (36), nous pouvons résoudre pour \mathbf{Q}_1 et \mathbf{Q}_2 :

$$\mathbf{Q}_1 = \mathbf{c}(t_5) - \dot{\mathbf{c}}(t_5) + \frac{1}{3}\ddot{\mathbf{c}}(t_5) \quad (37)$$

$$\mathbf{Q}_2 = \mathbf{c}(t_5) - \frac{1}{6}\ddot{\mathbf{c}}(t_5) \quad (38)$$

Et qui plus est, en substituant l'équation (37) et l'équation (38) dans l'équation (21), nous pouvons résoudre pour \mathbf{Q}_2 :

$$\mathbf{Q}_3 = \mathbf{c}(t_5) + \dot{\mathbf{c}}(t_5) + \frac{1}{3}\ddot{\mathbf{c}}(t_5) \quad (39)$$

La même procédure peut être utilisée pour résoudre pour \mathbf{Q}_{n-2} , \mathbf{Q}_{n-1} and \mathbf{Q}_n .

$$\mathbf{Q}_{n-2} = \mathbf{c}(t_{n+2}) - \dot{\mathbf{c}}(t_{n+2}) + \frac{1}{3}\ddot{\mathbf{c}}(t_{n+2}) \quad (40)$$

$$\mathbf{Q}_{n-1} = \mathbf{c}(t_{n+2}) - \frac{1}{6}\ddot{\mathbf{c}}(t_{n+2}) \quad (41)$$

$$\mathbf{Q}_n = \mathbf{c}(t_{n+2}) + \dot{\mathbf{c}}(t_{n+2}) + \frac{1}{3}\ddot{\mathbf{c}}(t_{n+2}) \quad (42)$$

Les solutions peuvent être présentées sous la forme matricielle suivante

$$\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 6 & -6 & 2 \\ 6 & 0 & -1 \\ 6 & 6 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_5) \\ \dot{\mathbf{c}}(t_5) \\ \ddot{\mathbf{c}}(t_5) \end{bmatrix} \quad (43)$$

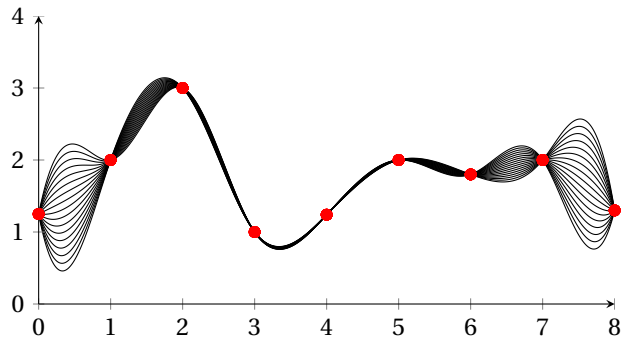


Figure 5 – Évolution des conditions limites en fonction des premier et dernier points de contrôle

$$\begin{bmatrix} \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 6 & -6 & 2 \\ 6 & 0 & -1 \\ 6 & 6 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_{n+2}) \\ \dot{\mathbf{c}}(t_{n+2}) \\ \ddot{\mathbf{c}}(t_{n+2}) \end{bmatrix} \quad (44)$$

La figure suivante présente l'effet qu'ont les conditions limites aux extrémités de la courbe paramétrique.

Les équations présentées dans cette section permettent de construire une spline cubique pour l'interpolation ou l'approximation d'un ensemble de points de données quelconque du plan. La section suivante étend l'ensemble de ces développements au cas des surfaces paramétriques pour des splines bi-cubiques. Les conventions utilisées pour le cas 3D s'inspirent de celles présentées plus haut pour le cas 2D.

2 Surfaces paramétriques

Cette section traite de la procédure à suivre pour créer une surface paramétrique en utilisant des B-splines bicubiques. En accord avec ce qui est mentionné dans l'introduction, il existe une transformation ϕ de l'espace telle que l'ensemble des points de contrôle \mathbf{Q} (ou de données \mathbf{P}) $(\hat{\mathbf{Q}}_x; \hat{\mathbf{Q}}_y; \hat{\mathbf{Q}}_z) = (\phi \mathbf{Q}_x; \phi \mathbf{Q}_y; \phi \mathbf{Q}_z)$ appartiennent au plan $(O; \vec{x}; \vec{y})$ et définissent un domaine rectangulaire au sein duquel ils sont régulièrement espacés. Il est donc possible d'écrire les coordonnées des points de contrôle \mathbf{Q} sous forme matricielle pour chaque coordonnée : \mathbf{Q}_x , \mathbf{Q}_y et \mathbf{Q}_z . Par convention, la dimension de ces matrices est $(n) \times (m)$.

2.1 Généralités

La formule qui permet de définir une surface paramétrique est

$$S(u, v) = \sum_{i=1}^n \sum_{j=1}^m \mathbf{Q}_{i,j} B_{i,p}(u) B_{j,d}(v) \quad (45)$$

où

- n et m correspondent aux nombres de lignes et colonnes des matrices \mathbf{Q}_x , \mathbf{Q}_y et \mathbf{Q}_z ;
- $n - p + 1$ et $m - d + 1$ correspondent respectivement au nombre de points de données par rangée et par colonne dans la matrice des points de données \mathbf{P}_x , \mathbf{P}_y et \mathbf{P}_z ;
- d et p représentent le degré de chacune des fonctions polynômiales $B_{j,d}(v)$ et $B_{i,p}(u)$.

Les paramètres qui sont nécessaires pour la création d'une surface paramétrique sont les suivants :

1. Paramètres intrinsèques

- deux paramètres intrinsèques u et v sont requis pour définir une surface paramétrique ;
- le paramètre intrinsèque qui permet de parcourir la surface paramétrique en passant d'une ligne à l'autre dans les matrices de coordonnées \mathbf{P}_x , \mathbf{P}_y et \mathbf{P}_z est u ;
- le paramètre intrinsèque qui permet de parcourir la surface paramétrique en passant d'une colonne à l'autre dans les matrices de coordonnées \mathbf{P}_x , \mathbf{P}_y et \mathbf{P}_z est v .

La surface paramétrique est divisée en plusieurs patches où chacun des patches est limité par une valeur minimale et maximale de chacun des paramètres intrinsèques.

2. Vecteur de nœuds

- un vecteur de nœuds doit être défini pour chaque paramètre intrinsèque ;
- la taille du vecteur de nœuds équivaut au nombre de colonnes ou de rangées dans la matrice des points de contrôle à l'addition de l'ordre (degré + 1) de la fonction spline correspondante. Par convention, on choisit

$$\begin{aligned} u \in \mu &= [\mu_1 \mu_2 \mu_3 \dots \mu_{n+p-1} \mu_{n+p} \mu_{n+p+1}] \\ v \in \nu &= [\nu_1 \nu_2 \nu_3 \dots \nu_{m+d-1} \nu_{m+d} \nu_{m+d+1}] \end{aligned}$$

Dans cette note technique, nous considérons que les deux fonctions splines, $B_{i,p}(u)$ et $B_{j,d}(v)$, sont de degré 3 et que les vecteurs de nœuds leur correspondant sont uniformes. Ainsi, pour le paramètre intrinsèque u , nous obtenons

$$\begin{aligned} \mu &= [\mu_1 \mu_2 \mu_3 \dots \mu_{n+2} \mu_{n+3} \mu_{n+4}] \\ &= [1 \ 2 \ 3 \ \dots \ n+2 \ n+3 \ n+4] \end{aligned}$$

et pour le paramètre intrinsèque v

$$\begin{aligned} \nu &= [\nu_1 \nu_2 \nu_3 \dots \nu_{m+2} \nu_{m+3} \nu_{m+4}] \\ &= [1 \ 2 \ 3 \ \dots \ m+2 \ m+3 \ m+4] \end{aligned}$$

De même que pour les courbes paramétriques, un domaine effectif est défini pour chaque vecteur de nœuds (cf. section 6)

$$\begin{aligned} [\mu_{p+2}; \mu_{m+p-1}] &\subset [\mu_1; \mu_{n+p+1}] \\ [\nu_{d+2}; \nu_{m+d-1}] &\subset [\nu_1; \nu_{m+d+1}] \end{aligned}$$

3. Points de données

- ils sont représentés par la variable \mathbf{P} ;
- ils se présentent sous forme matricielle dont la taille est $(n - p + 1) \times (m - d + 1)$ et où le numéro de rangée est dénoté par l'indice i et le numéro de colonne, par l'indice j .

Les figures 6 et 7 présentent une nappe de points dans l'espace (x,y,z) .

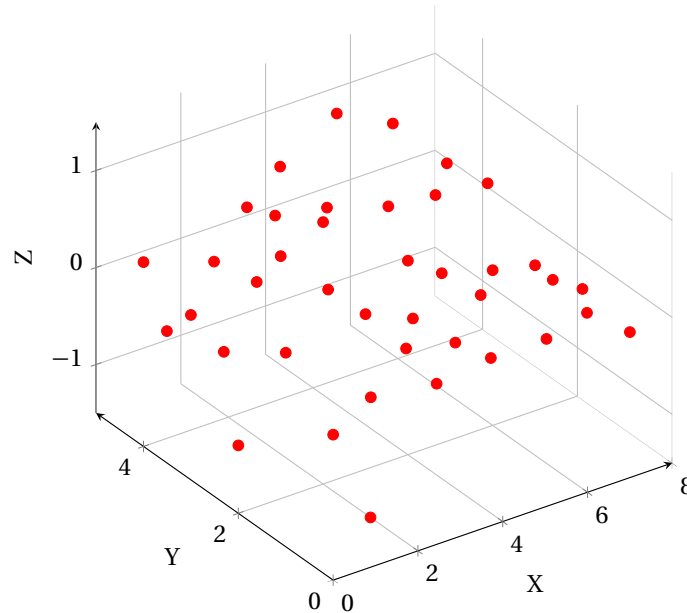


Figure 6 – Ensemble de coordonnées (x, y, z) (angle de vue 1)

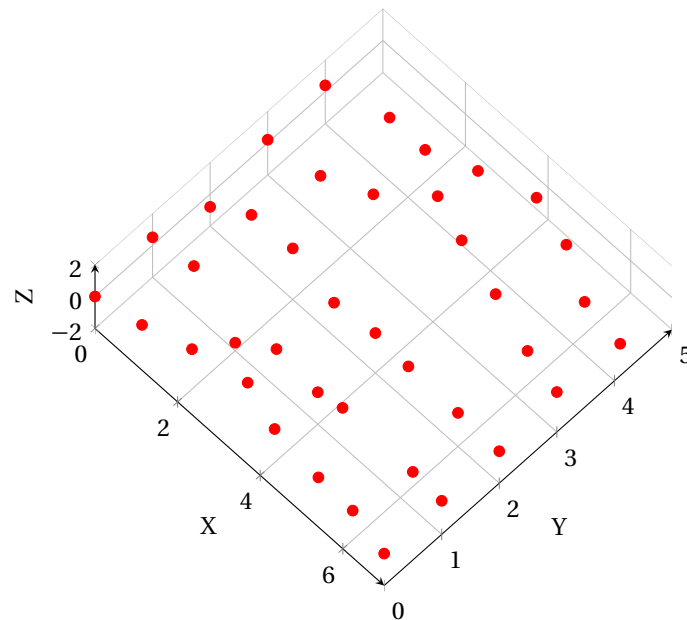


Figure 7 – Ensemble de coordonnées (x, y, z) (angle de vue 2)

4. Fonctions polynômiales

- elles sont notées $B_{i,p}(u)$ et $B_{j,d}(v)$;

- elles sont définies pour chacun des paramètres intrinsèques et ce, à l'aide de la formule Cox-de-Boor présentée à l'équation (4).

5. **Points de contrôle**

- ils sont représentés par la variable $Q_{i,j}$
- le nombre de points de contrôle par rangée est (degré de $B_{i,p}(u) + 1$) et par colonne, est (degré de $B_{j,d}(v) + 1$) et ce, pour un seul patch ;
- il y a, par convention, un total de $n \times m$ points de contrôle.

2.2 **Convention adoptée**

La figure 8 résume la convention adoptée pour la définition de la surface paramétrique (en supposant que $d = p = 3$).

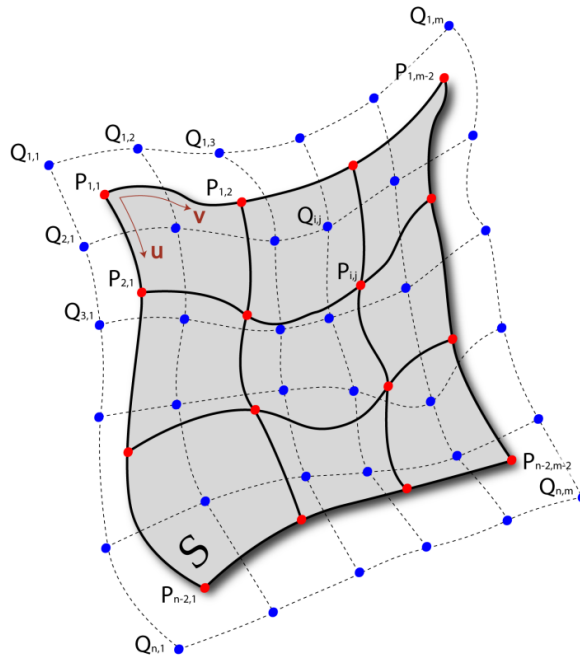


Figure 8 – Convention adoptée pour une surface paramétrique

2.3 **Tracé**

Dans le cas des splines bi-cubiques, les fonctions polynômiales $B_{i,p}(u)$ et $B_{j,d}(v)$ sont définies à l'aide de la formule Cox-de-Boor présentée à l'équation (4), dans la section (1.2). Ces fonctions sont détaillées dans la section (??) de l'annexe.

Les fonctions polynômiales pour des splines bi-cubiques pour chaque paramètre intrinsèque u et v

sont

$$\begin{aligned}
B_{i,3}(u) = & \frac{u - \mu_i}{\mu_{i+3} - \mu_i} \left[\frac{u - \mu_i}{\mu_{i+2} - \mu_i} \left(\frac{u - \mu_i}{\mu_{i+1} - \mu_i} B_{i,0}(u) + \frac{\mu_{i+2} - u}{\mu_{i+2} - \mu_{i+1}} B_{i+1,0}(u) \right) \right. \\
& + \left. \frac{\mu_{i+3} - u}{\mu_{i+3} - \mu_{i+1}} \left(\frac{u - \mu_{i+1}}{\mu_{i+2} - \mu_{i+1}} B_{i+1,0}(u) + \frac{\mu_{i+3} - u}{\mu_{i+3} - \mu_{i+2}} B_{i+2,0}(u) \right) \right] \\
& + \frac{\mu_{i+4} - u}{\mu_{i+4} - \mu_{i+1}} \left[\frac{u - \mu_{i+1}}{\mu_{i+3} - \mu_{i+1}} \left(\frac{u - \mu_{i+1}}{\mu_{i+2} - \mu_{i+1}} B_{i+1,0}(u) + \frac{\mu_{i+3} - u}{\mu_{i+3} - \mu_{i+2}} B_{i+2,0}(u) \right) \right. \\
& + \left. \frac{\mu_{i+4} - u}{\mu_{i+4} - \mu_{i+2}} \left(\frac{u - \mu_{i+2}}{\mu_{i+3} - \mu_{i+2}} B_{i+2,0}(u) + \frac{\mu_{i+4} - u}{\mu_{i+4} - \mu_{i+3}} B_{i+3,0}(u) \right) \right]
\end{aligned} \tag{46}$$

$$\begin{aligned}
B_{j,3}(v) = & \frac{v - \nu_j}{\nu_{j+3} - \nu_j} \left[\frac{v - \nu_j}{\nu_{j+2} - \nu_j} \left(\frac{v - \nu_j}{\nu_{j+1} - \nu_j} B_{j,0}(v) + \frac{\nu_{j+2} - v}{\nu_{j+2} - \nu_{j+1}} B_{j+1,0}(v) \right) \right. \\
& + \left. \frac{\nu_{j+3} - v}{\nu_{j+3} - \nu_{j+1}} \left(\frac{v - \nu_{j+1}}{\nu_{j+2} - \nu_{j+1}} B_{j+1,0}(v) + \frac{\nu_{j+3} - v}{\nu_{j+3} - \nu_{j+2}} B_{j+2,0}(v) \right) \right] \\
& + \frac{\nu_{j+4} - v}{\nu_{j+4} - \nu_{j+1}} \left[\frac{v - \nu_{j+1}}{\nu_{j+3} - \nu_{j+1}} \left(\frac{v - \nu_{j+1}}{\nu_{j+2} - \nu_{j+1}} B_{j+1,0}(v) + \frac{\nu_{j+3} - v}{\nu_{j+3} - \nu_{j+2}} B_{j+2,0}(v) \right) \right. \\
& + \left. \frac{\nu_{j+4} - v}{\nu_{j+4} - \nu_{j+2}} \left(\frac{v - \nu_{j+2}}{\nu_{j+3} - \nu_{j+2}} B_{j+2,0}(v) + \frac{\nu_{j+4} - v}{\nu_{j+4} - \nu_{j+3}} B_{j+3,0}(v) \right) \right]
\end{aligned} \tag{47}$$

De même que pour les courbes paramétriques, l'équation (45) se limite à un nombre restreint de termes selon les valeurs des paramètres intrinsèques u et v . Dans le cas où u appartient à $[\mu_{i+1}, \mu_{i+2}]$ et v appartient à $[\nu_{j+1}, \nu_{j+2}]$, l'équation (45) se réduit aux indices suivants :

$$i = \begin{cases} r-3 \\ r-2 \\ r-1 \\ r \end{cases} \quad j = \begin{cases} h-3 \\ h-2 \\ h-1 \\ h \end{cases}$$

Ainsi, pour

$$u \in [\mu_{r+1}, \mu_{r+2}] \text{ où } 4 \leq r \leq n$$

$$v \in [\nu_{h+1}, \nu_{h+2}] \text{ où } 4 \leq h \leq m$$

l'équation (45) s'écrit plus simplement

$$\begin{aligned}
\mathbf{S}(u, v) = & B_{h-3,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h-3} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h-3} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h-3} + B_{r,3}(u) \mathbf{Q}_{r,h-3} \right] \\
& + B_{h-2,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h-2} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h-2} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h-2} + B_{r,3}(u) \mathbf{Q}_{r,h-2} \right] \\
& + B_{h-1,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h-1} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h-1} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h-1} + B_{r,3}(u) \mathbf{Q}_{r,h-1} \right] \\
& + B_{h,3}(v) \left[B_{r-3,3}(u) \mathbf{Q}_{r-3,h} + B_{r-2,3}(u) \mathbf{Q}_{r-2,h} + B_{r-1,3}(u) \mathbf{Q}_{r-1,h} + B_{r,3}(u) \mathbf{Q}_{r,h} \right]
\end{aligned} \tag{48}$$

Afin d'exprimer l'équation (48) sous forme polynômiale, nous substituons u par $(u + \mu_r)$ et v par $(v + \nu_h)$ et utilisons un vecteur de noeuds uniforme pour chacun des paramètres intrinsèques. Il vient

$$\begin{aligned}
\mathbf{S}(u, v) &= \sum_{i=1}^n \sum_{j=1}^m \mathbf{Q}_{i,j} \mathbf{B}_{i,3}(u) \mathbf{B}_{j,3}(v) \\
&= (1-v)^3 \left[(1-u)^3 \mathbf{Q}_{r-3,h-3} + (3u^3 - 6u^2 + 4) \mathbf{Q}_{r-2,h-3} + (-3u^3 + 3u^2 + 3u + 1) \mathbf{Q}_{r-1,h-3} + u^3 \mathbf{Q}_{r,h-3} \right] \\
&\quad + (3v^3 - 6v^2 + 4) \left[(1-u)^3 \mathbf{Q}_{r-3,h-2} + (3u^3 - 6u^2 + 4) \mathbf{Q}_{r-2,h-2} + (-3u^3 + 3u^2 + 3u + 1) \mathbf{Q}_{r-1,h-2} + u^3 \mathbf{Q}_{r,h-2} \right] \\
&\quad + (-3v^3 + 3v^2 + 3v + 1) \left[(1-u)^3 \mathbf{Q}_{r-3,h-1} + (3u^3 - 6u^2 + 4) \mathbf{Q}_{r-2,h-1} + (-3u^3 + 3u^2 + 3u + 1) \mathbf{Q}_{r-1,h-1} + u^3 \mathbf{Q}_{r,h-1} \right] \\
&\quad + v^3 \left[(1-u)^3 \mathbf{Q}_{r-3,h} + (3u^3 - 6u^2 + 4) \mathbf{Q}_{r-2,h} + (-3u^3 + 3u^2 + 3u + 1) \mathbf{Q}_{r-1,h} + u^3 \mathbf{Q}_{r,h} \right]
\end{aligned} \tag{49}$$

Ce qui peut s'écrire sous forme matricielle

$$\mathbf{S}(u, v) = \frac{1}{36} \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{r-3,h-3} & \mathbf{Q}_{r-3,h-2} & \mathbf{Q}_{r-3,h-1} & \mathbf{Q}_{r-3,h} \\ \mathbf{Q}_{r-2,h-3} & \mathbf{Q}_{r-2,h-2} & \mathbf{Q}_{r-2,h-1} & \mathbf{Q}_{r-2,h} \\ \mathbf{Q}_{r-1,h-3} & \mathbf{Q}_{r-1,h-2} & \mathbf{Q}_{r-1,h-1} & \mathbf{Q}_{r-1,h} \\ \mathbf{Q}_{r,h-3} & \mathbf{Q}_{r,h-2} & \mathbf{Q}_{r,h-1} & \mathbf{Q}_{r,h} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \tag{50}$$

où $u \in [\mu_{r+1}, \mu_{r+2}]$ et $v \in [\nu_{h+1}, \nu_{h+2}]$ pour $4 \leq r \leq n$ et $4 \leq h \leq m$.

L'équation (51) nous sera utile autant dans l'approximation que dans l'interpolation. La section (2.4) qui suit traite de la procédure qui permet, à partir des points de contrôle donnés, de trouver les coordonnées en (x, y, z) de la courbe spline. La section (2.5) traite de l'interpolation des points de données pour une B-Spline bicubique ainsi que de la procédure à suivre pour trouver les points de contrôle qui leur sont associés.

2.4 Approximation

De même que pour une courbe paramétrique, l'approximation d'un ensemble de points de données \mathbf{P} se fait en considérant ces points comme des points de contrôle et l'équation (51) s'écrit

$$\mathbf{S}(u, v) = \frac{1}{36} \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{r-3,h-3} & \mathbf{P}_{r-3,h-2} & \mathbf{P}_{r-3,h-1} & \mathbf{P}_{r-3,h} \\ \mathbf{P}_{r-2,h-3} & \mathbf{P}_{r-2,h-2} & \mathbf{P}_{r-2,h-1} & \mathbf{P}_{r-2,h} \\ \mathbf{P}_{r-1,h-3} & \mathbf{P}_{r-1,h-2} & \mathbf{P}_{r-1,h-1} & \mathbf{P}_{r-1,h} \\ \mathbf{P}_{r,h-3} & \mathbf{P}_{r,h-2} & \mathbf{P}_{r,h-1} & \mathbf{P}_{r,h} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \tag{51}$$

où $u \in [\mu_{r+1}, \mu_{r+2}]$ et $v \in [\nu_{h+1}, \nu_{h+2}]$ pour $4 \leq r \leq n-2$ et $4 \leq h \leq m-2$.

Les points de données tracés sur les figures 6 et 7 sont approchés par une spline bi-cubique sur les figures 9 et 10.

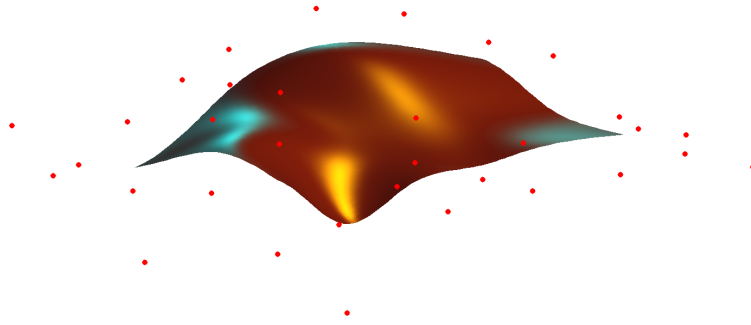


Figure 9 – Approximation des données avec surfaces splines (angle de vue 1)

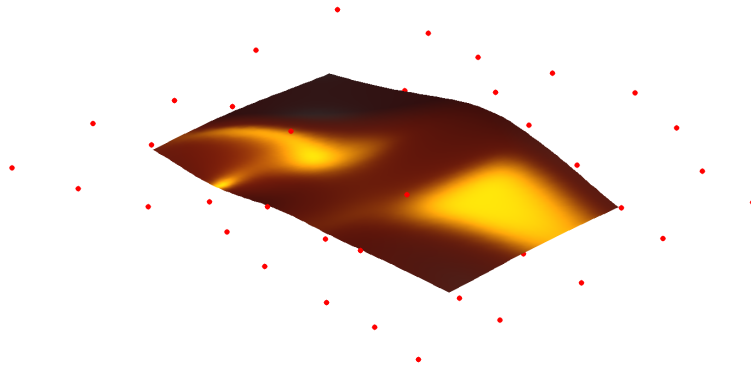


Figure 10 – Approximation des données avec surfaces splines (angle de vue 2)

2.5 Interpolation

Note : Dans cette section afin de faciliter l'écriture des équations, l'ensemble des points de contrôle \mathbf{Q} et de données \mathbf{P} sont réorganisés sous forme matricielle

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{1:n-2,1} \\ \mathbf{P}_{1:n-2,2} \\ \cdot \\ \cdot \\ \mathbf{P}_{1:n-2,m-2} \end{pmatrix} \text{ et } \mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{1:n,1} \\ \mathbf{Q}_{1:n,2} \\ \cdot \\ \cdot \\ \mathbf{Q}_{1:n,m} \end{pmatrix} \quad (52)$$

Étant donné un ensemble de $(n-2) \times (m-2)$ points de données \mathbf{P} , il s'agit de déterminer les $n \times m$ points de contrôle \mathbf{Q} permettant de créer une spline bi-cubique ($d = p = 3$) passant par tous les points de \mathbf{P} .

La traduction mathématique du passage de la spline par les $(n-2) \times (m-2)$ points de données amène $(n-2) \times (m-2)$ équations. Les $n \times m - (n-2) \times (m-2)$ équations obtenues en imposant que la surface passe par les points de données s'écrivent sous forme matricielle

$$\mathbf{P} = \mathbf{A}\mathbf{Q} \quad (53)$$

où \mathbf{A} est une matrice rectangulaire $((n-2) \cdot (m-2)) \times (nm)$. De même que pour les courbes paramétriques, c'est en considérant des conditions limites particulières qu'il est possible d'obtenir les $n \times m - (n-2) \times (m-2)$ équations manquantes.

Le nombre d'équations manquantes correspond au nombre de points de contrôle total moins le nombre de points de contrôle sur le bord du domaine étudié (soit $\mathbf{Q}_{i,j}$, $i = 2, \dots, n-1$ et $j = 2, \dots, m-1$)

comme le montre la figure 8. On définit l'ensemble des points de contrôle *intérieurs* :

$$\mathbf{Q}_{\text{int}} = \mathbf{Q}_{i,j}, \quad i = 2, \dots, n-1 \quad \text{et} \quad j = 2, \dots, m-1 \quad (54)$$

Le choix de conditions limites spécifiques permet ensuite d'obtenir une relation

$$\mathbf{Q} = \mathbf{C}\mathbf{Q}_{\text{int}} \quad (55)$$

où \mathbf{C} est une matrice rectangulaire $(nm) \times ((n-2) \cdot (m-2))$. Il est alors possible de définir un changement de variable entre les points de données et les points de contrôle intérieurs

$$\mathbf{P} = \mathbf{A}\mathbf{C}\mathbf{Q}_{\text{int}} \Rightarrow \mathbf{Q}_{\text{int}} = [\mathbf{A}\mathbf{C}]^{-1} \mathbf{P} \quad (56)$$

La totalité des points de contrôle \mathbf{Q} peut alors être obtenue

$$\boxed{\mathbf{Q} = \mathbf{C}[\mathbf{A}\mathbf{C}]^{-1} \mathbf{P}} \quad (57)$$

Les sections suivantes détaillent l'obtention des matrices \mathbf{A} et \mathbf{C} pour des surfaces ouverte et fermée.

2.5.1 Surface ouverte

Obtention de la matrice \mathbf{A} :

Prenons l'exemple d'un patch défini par quatre points de données ($\mathbf{P}_{r-3,h-3}, \mathbf{P}_{r-3,h-2}, \mathbf{P}_{r-2,h-3}, \mathbf{P}_{r-2,h-2}$). En utilisant l'équation (51), les points de contrôle associés à chaque point de données sont

$$\begin{aligned} \mathbf{P}_{r-3,h-3} &= \frac{1}{36} (\mathbf{Q}_{r-3,h-3} + 4\mathbf{Q}_{r-3,h-2} + \mathbf{Q}_{r-3,h-1} + 4\mathbf{Q}_{r-2,h-3} + 16\mathbf{Q}_{r-2,h-2} + 4\mathbf{Q}_{r-2,h-1} + \mathbf{Q}_{r-1,h-3} + 4\mathbf{Q}_{r-1,h-2} + \mathbf{Q}_{r-1,h-1}) \\ \mathbf{P}_{r-3,h-2} &= \frac{1}{36} (\mathbf{Q}_{r-2,h-3} + 4\mathbf{Q}_{r-2,h-2} + \mathbf{Q}_{r-2,h-1} + 4\mathbf{Q}_{r-1,h-3} + 16\mathbf{Q}_{r-1,h-2} + 4\mathbf{Q}_{r-1,h-1} + \mathbf{Q}_{r,h-3} + 4\mathbf{Q}_{r,h-2} + \mathbf{Q}_{r,h-1}) \\ \mathbf{P}_{r-2,h-3} &= \frac{1}{36} (\mathbf{Q}_{r-3,h-2} + 4\mathbf{Q}_{r-3,h-1} + \mathbf{Q}_{r-3,h} + 4\mathbf{Q}_{r-2,h-2} + 16\mathbf{Q}_{r-2,h-1} + 4\mathbf{Q}_{r-2,h} + \mathbf{Q}_{r-1,h-2} + 4\mathbf{Q}_{r-1,h-1} + \mathbf{Q}_{r-1,h}) \\ \mathbf{P}_{r-2,h-2} &= \frac{1}{36} (\mathbf{Q}_{r-2,h-2} + 4\mathbf{Q}_{r-2,h-1} + \mathbf{Q}_{r-2,h} + 4\mathbf{Q}_{r-1,h-2} + 16\mathbf{Q}_{r-1,h-1} + 4\mathbf{Q}_{r-1,h} + \mathbf{Q}_{r,h-2} + 4\mathbf{Q}_{r,h-1} + \mathbf{Q}_{r,h}) \end{aligned}$$

Ainsi, le vecteur position \mathbf{a} qui permet d'exprimer n'importe quel point de données en fonction des points de contrôle est

$$\mathbf{a} = [1 \ 4 \ 1 \ 0 \ 0 \ \dots \ 4 \ 16 \ 4 \ 0 \ 0 \ \dots \ 1 \ 4 \ 1 \ 0 \ 0 \ \dots] \quad (58)$$

qui contient $m-1$ termes nuls. En calculant le vecteur \mathbf{a} pour chaque point de données, on remplit peu à peu la matrice \mathbf{A} de l'équation (53)

$$\begin{bmatrix} \mathbf{P}_{1,1} \\ \mathbf{P}_{1,2} \\ \mathbf{P}_{1,3} \\ \vdots \\ \mathbf{P}_{n-2,m-4} \\ \mathbf{P}_{n-2,m-3} \\ \mathbf{P}_{n-2,m-2} \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 & \dots & 4 & 16 & 4 & 0 & \dots & 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \dots & 4 & 16 & 4 & 0 & \dots & 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & \dots & 4 & 16 & 4 & 0 & \dots & 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & & & & & \ddots & & & & & & & & & \vdots \\ 0 & \dots & 1 & 4 & 1 & 0 & \dots & 4 & 16 & 4 & 0 & \dots & 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 1 & 4 & 1 & 0 & \dots & 4 & 16 & 4 & 0 & \dots & 1 & 4 & 1 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & 1 & 4 & 1 & 0 & \dots & 4 & 16 & 4 & 0 & \dots & 1 & 4 & 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{1,1} \\ \mathbf{Q}_{1,2} \\ \mathbf{Q}_{1,3} \\ \vdots \\ \mathbf{Q}_{n,m} \\ \mathbf{Q}_{n,m} \\ \mathbf{Q}_{n,m} \end{bmatrix} \quad (59)$$

Obtention de la matrice C :

L'objectif de cette section est d'exprimer tous les points de contrôle sur le contour de la surface paramétrique en fonction des points de contrôle intérieurs et ce, en imposant des conditions limites spécifiques. Ceci permettra d'explicitier la relation (cf. équation (55)) entre les points de contrôle et les points de contrôle intérieurs.

Le choix est fait d'imposer des conditions limites respectant la symétrie des points de contrôles sur le contour par rapport aux points de contrôle intérieurs et on impose

1. Points de contrôle aux quatres extrémités

$$\begin{aligned} Q_{1,1} &= 2Q_{2,2} - Q_{3,3} \\ Q_{1,m} &= 2Q_{2,m-1} - Q_{3,m-2} \\ Q_{n,1} &= 2Q_{n-1,2} - Q_{n-2,3} \\ Q_{n,m} &= 2Q_{n-1,m-1} - Q_{n-2,m-2} \end{aligned} \tag{60}$$

2. Points de contrôle sur le bord du haut

$$\begin{aligned} Q_{1,1} &= 2Q_{2,1} - Q_{3,1} \\ Q_{1,2} &= 2Q_{2,2} - Q_{3,2} \\ &\vdots \\ Q_{1,m-2} &= 2Q_{2,m-2} - Q_{3,m-2} \\ Q_{1,m-1} &= 2Q_{2,m-1} - Q_{3,m-1} \end{aligned} \tag{61}$$

3. Points de contrôle sur le bord de gauche

$$\begin{aligned} Q_{2,1} &= 2Q_{2,2} - Q_{2,3} \\ Q_{3,1} &= 2Q_{3,2} - Q_{3,3} \\ &\vdots \\ Q_{n-2,1} &= 2Q_{n-2,2} - Q_{n-2,3} \\ Q_{n-1,1} &= 2Q_{n-1,2} - Q_{n-1,3} \end{aligned} \tag{62}$$

4. Points de contrôle sur le bord de droite

$$\begin{aligned} Q_{2,m} &= 2Q_{2,m-1} - Q_{2,m-2} \\ Q_{3,m} &= 2Q_{3,m-1} - Q_{3,m-2} \\ &\vdots \\ Q_{n-1,m} &= 2Q_{n-1,m-1} - Q_{n-1,m-2} \\ Q_{n-2,m} &= 2Q_{n-2,m-1} - Q_{n-2,m-2} \end{aligned} \tag{63}$$

5. Points de contrôle sur le bord du bas

$$\begin{aligned} Q_{n,2} &= 2Q_{n-1,2} - Q_{n-2,2} \\ Q_{n,3} &= 2Q_{n-1,3} - Q_{n-2,3} \\ &\vdots \\ Q_{n,m-2} &= 2Q_{n-1,m-2} - Q_{n-2,m-2} \\ Q_{n,m-1} &= 2Q_{n-1,m-1} - Q_{n-2,m-1} \end{aligned} \tag{64}$$

Ainsi, la relation, sous forme matricielle, entre tous les points de contrôle \mathbf{Q} et les points de contrôle intérieurs \mathbf{Q}_{int} est

$$\begin{bmatrix} \mathbf{Q}_{1,1} \\ \mathbf{Q}_{1,2} \\ \mathbf{Q}_{1,3} \\ \vdots \\ \mathbf{Q}_{1,m} \\ \mathbf{Q}_{2,1} \\ \mathbf{Q}_{2,2} \\ \mathbf{Q}_{2,3} \\ \vdots \\ \mathbf{Q}_{2,m} \\ \mathbf{Q}_{3,1} \\ \mathbf{Q}_{3,2} \\ \mathbf{Q}_{3,3} \\ \vdots \\ \mathbf{Q}_{3,m} \\ \mathbf{Q}_{4,1} \\ \mathbf{Q}_{4,2} \\ \mathbf{Q}_{4,3} \\ \vdots \\ \mathbf{Q}_{n-1,m} \\ \mathbf{Q}_{n,1} \\ \mathbf{Q}_{n,2} \\ \mathbf{Q}_{n,3} \\ \vdots \\ \mathbf{Q}_{n,m-2} \\ \mathbf{Q}_{n,m-1} \\ \mathbf{Q}_{n,m} \end{bmatrix} = \begin{bmatrix} 2 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 2 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 2 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & & & & & \ddots & & & & & & \vdots \\ 0 & \dots & 0 & 2 & 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & & & & & \ddots & & & & & & \vdots \\ 0 & \dots & 0 & -1 & 2 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 2 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & & & & & \ddots & & & & & & \vdots \\ 0 & 0 & 0 & \dots & 0 & -1 & 2 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 2 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & & & & \ddots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & -1 & 2 \\ 0 & 0 & \dots & 0 & -1 & 0 & \dots & 0 & 2 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 & 2 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 & 2 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & & & & \ddots & & & & & & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 2 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{2,2} \\ \mathbf{Q}_{2,3} \\ \mathbf{Q}_{2,4} \\ \vdots \\ \mathbf{Q}_{2,m-1} \\ \mathbf{Q}_{3,2} \\ \mathbf{Q}_{3,3} \\ \vdots \\ \mathbf{Q}_{3,m-1} \\ \mathbf{Q}_{4,2} \\ \mathbf{Q}_{4,3} \\ \vdots \\ \mathbf{Q}_{n,m-1} \\ \mathbf{Q}_{n-1,2} \\ \mathbf{Q}_{n-1,3} \\ \vdots \\ \mathbf{Q}_{n-1,m-3} \\ \mathbf{Q}_{n-2,m-2} \\ \mathbf{Q}_{n-2,m-1} \end{bmatrix} \quad (65)$$

À l'aide de la procédure présentée, on peut interpoler l'ensemble des points donnés dans les figures 6 et 7 comme le montrent les figures 11 et 12.

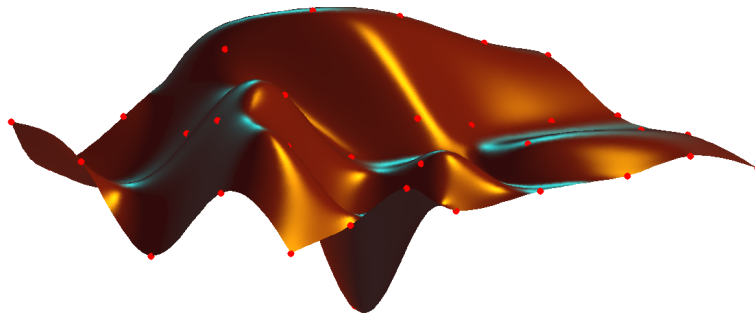


Figure 11 – Interpolation des données avec surfaces splines (angle de vue 1)

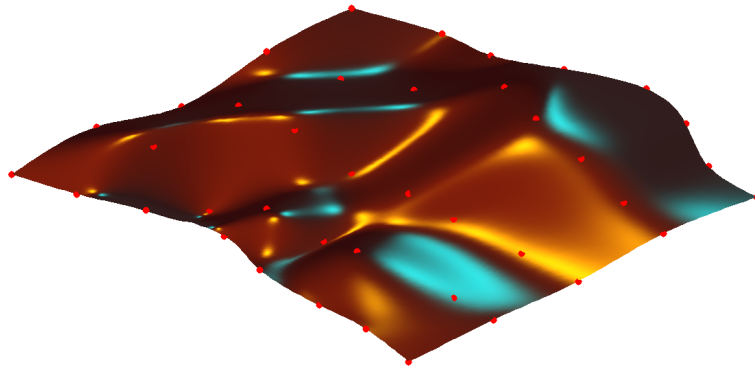


Figure 12 – Interpolation des données avec surfaces splines (angle de vue 2)

2.5.2 Surface fermée

En reprenant les conventions de la section 2.1, une surface est dite *fermée* lorsque

$$\forall v \mathbf{S}(\mu_{p+2}, v) = \mathbf{S}(\mu_{m+p-1}, v) \quad (66)$$

ou bien éventuellement lorsque

$$\forall u \mathbf{S}(u, \nu_{d+2}) = \mathbf{S}(u, \nu_{m+d-1}) \quad (67)$$

Le cas où les conditions (66) et (67) sont satisfaites simultanément n'est pas traité dans cette note.

Une surface fermée peut ainsi être une sphère tronquée, un cylindre, un tore partiel... Si on se limite aux conditions limites de type symétrique définies dans la section précédente, la surface fermée ne sera pas *lisse* au niveau de la zone de recouplement. On cherche donc à imposer des conditions limites particulières pour avoir une continuité de type C^1 sur cette zone.

On se place dans le cas où la condition (67) est satisfaite. On impose

$$\begin{aligned} \mathbf{S}_{,v}(0, 0) &= \mathbf{S}_{,v}(0, v_{max}) \\ \mathbf{S}_{,v}(u_{max}, 0) &= \mathbf{S}_{,v}(u_{max}, v_{max}) \\ \mathbf{S}_{,vv}(0, 0) &= \mathbf{S}_{,vv}(0, v_{max}) \\ \mathbf{S}_{,vv}(u_{max}, 0) &= \mathbf{S}_{,vv}(u_{max}, v_{max}) \end{aligned} \quad (68)$$

où la notation $_{,v}$ signifie dérivée partielle par rapport à v .

En résolvant le système d'équations 68, les inconnues (incluant celles aux extrémités) exprimées en

fonction des points de contrôle intérieurs sont

$$\begin{aligned}
\mathbf{Q}_{1,1} &= 2\mathbf{Q}_{2,2} - \mathbf{Q}_{2,3} + 2\mathbf{Q}_{n-2,2} - \mathbf{Q}_{n-2,3} - 2\mathbf{Q}_{n-1,2} + \mathbf{Q}_{n-1,3} \\
\mathbf{Q}_{1,2} &= \mathbf{Q}_{2,2} + \mathbf{Q}_{n-2,2} - \mathbf{Q}_{n-1,2} \\
\mathbf{Q}_{1,3} &= \mathbf{Q}_{2,3} + \mathbf{Q}_{n-2,3} - \mathbf{Q}_{n-1,3} \\
&\vdots \\
\mathbf{Q}_{1,m-2} &= \mathbf{Q}_{2,m-2} + \mathbf{Q}_{n-2,m-2} - \mathbf{Q}_{n-1,m-2} \\
\mathbf{Q}_{1,m-1} &= \mathbf{Q}_{2,m-1} + \mathbf{Q}_{n-2,m-1} - \mathbf{Q}_{n-1,m-1} \\
\mathbf{Q}_{1,m} &= -\mathbf{Q}_{2,m-2} + 2\mathbf{Q}_{2,m-1} - \mathbf{Q}_{n-2,m-2} + 2\mathbf{Q}_{n-2,m-1} + \mathbf{Q}_{n-1,m-2} - 2\mathbf{Q}_{n-1,m-1} \\
&\vdots \\
\mathbf{Q}_{n,1} &= -2\mathbf{Q}_{2,2} + \mathbf{Q}_{2,3} + 2\mathbf{Q}_{3,3} - \mathbf{Q}_{3,3} + 2\mathbf{Q}_{n-1,2} - \mathbf{Q}_{n-1,3} \\
\mathbf{Q}_{n,2} &= -\mathbf{Q}_{2,2} + \mathbf{Q}_{3,2} + \mathbf{Q}_{n-1,2} \\
\mathbf{Q}_{n,3} &= -\mathbf{Q}_{2,3} + \mathbf{Q}_{3,3} + \mathbf{Q}_{n-1,3} \\
&\vdots \\
\mathbf{Q}_{n,m-2} &= -\mathbf{Q}_{2,m-2} + \mathbf{Q}_{3,m-2} + \mathbf{Q}_{n-1,m-2} \\
\mathbf{Q}_{n,m-1} &= -\mathbf{Q}_{2,m-1} + \mathbf{Q}_{3,m-1} + \mathbf{Q}_{n-1,m-1} \\
\mathbf{Q}_{n,m} &= \mathbf{Q}_{2,m-2} - 2\mathbf{Q}_{2,m-1} - \mathbf{Q}_{3,m-2} + 2\mathbf{Q}_{3,m-1} - \mathbf{Q}_{n-1,m-2} + 2\mathbf{Q}_{n-1,m-1}
\end{aligned} \tag{69}$$

Ainsi, nous pouvons établir sous forme matricielle la relation entre tous les points de contrôle \mathbf{Q} et les points de contrôle intérieurs \mathbf{Q}_{int} .

$$\begin{bmatrix} \mathbf{Q}_{1,1} \\ \mathbf{Q}_{1,2} \\ \mathbf{Q}_{1,3} \\ \vdots \\ \mathbf{Q}_{1,m-2} \\ \mathbf{Q}_{1,m-1} \\ \mathbf{Q}_{1,m} \\ \mathbf{Q}_{2,1} \\ \mathbf{Q}_{2,2} \\ \mathbf{Q}_{2,3} \\ \vdots \\ \mathbf{Q}_{n-1,m-2} \\ \mathbf{Q}_{n-1,m-1} \\ \mathbf{Q}_{n-1,m} \\ \mathbf{Q}_{n,1} \\ \mathbf{Q}_{n,2} \\ \mathbf{Q}_{n,3} \\ \vdots \\ \mathbf{Q}_{n,m-2} \\ \mathbf{Q}_{n,m-1} \\ \mathbf{Q}_{n,m} \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 2 & -1 & 0 & \dots & 0 & -2 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & -1 & 0 & \dots & 0 \\ \vdots & & & & & & & \ddots & & & & & & & \vdots \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & -1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & -1 \\ 0 & \dots & 0 & -1 & 2 & 0 & \dots & 0 & -1 & 2 & 0 & \dots & 0 & 1 & -2 \\ 2 & -1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & \ddots & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & -1 & 2 \\ -2 & 1 & 0 & \dots & 0 & 2 & -1 & 0 & \dots & 0 & 2 & -1 & 0 & \dots & 0 \\ -1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & & & & & \ddots & & & & & & & \vdots \\ 0 & \dots & 0 & -1 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & -1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 1 & -2 & 0 & \dots & -1 & 2 & 0 & \dots & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{2,2} \\ \mathbf{Q}_{2,3} \\ \mathbf{Q}_{2,4} \\ \vdots \\ \mathbf{Q}_{2,m-3} \\ \mathbf{Q}_{2,m-2} \\ \mathbf{Q}_{2,m-1} \\ \mathbf{Q}_{3,2} \\ \mathbf{Q}_{3,3} \\ \mathbf{Q}_{3,4} \\ \vdots \\ \mathbf{Q}_{n-2,m-3} \\ \mathbf{Q}_{n-2,m-2} \\ \mathbf{Q}_{n-2,m-1} \\ \mathbf{Q}_{n-1,2} \\ \mathbf{Q}_{n-1,3} \\ \mathbf{Q}_{n-1,4} \\ \vdots \\ \mathbf{Q}_{n-1,m-3} \\ \mathbf{Q}_{n-1,m-2} \\ \mathbf{Q}_{n-1,m-1} \end{bmatrix} \tag{70}$$

ce qui permet de définir une nouvelle matrice \mathbf{C} et d'obtenir des points de contrôle prenant en compte la condition continuité C^1 sur la zone de recouvrement.

Les figures suivantes présentent l'effet d'appliquer les conditions limites appropriées sur une sphère afin d'assurer une certaine continuité de la surface.

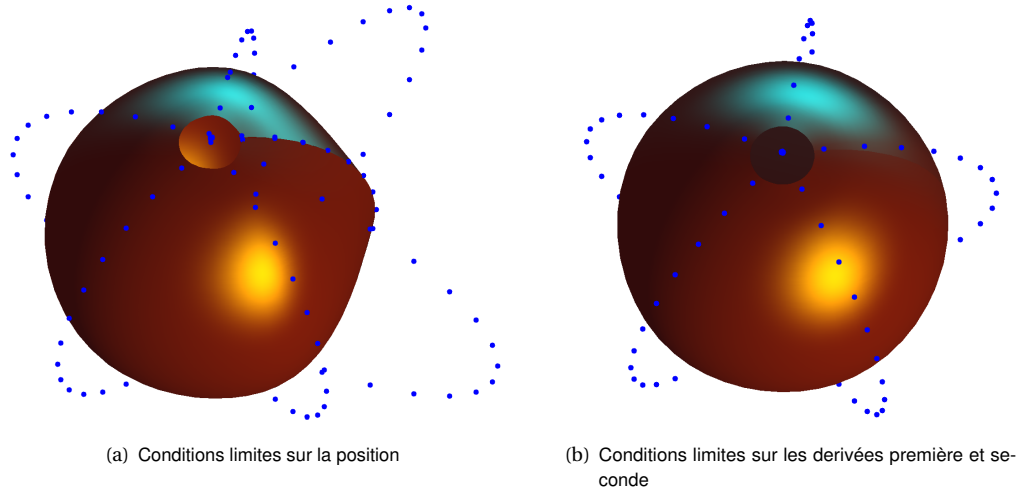


Figure 13 – Interpolation de points appartenant à une sphère

3 Algorithme de détection

L'algorithme de détection suggéré dans cette note technique utilise la procédure Newton-Raphson qui nécessite les dérivées première et seconde de $B_{i,3}(u)$ et $B_{j,3}(v)$. Ces fonctions peuvent être retrouvées plus en détails, dans la section (??) de l'annexe.

3.1 Procédure itérative de Newton-Raphson

Les figures 14 et 15 ont pour objectif d'illustrer la procédure itérative détaillée ci-dessous. Le point impacteur \mathbf{M} est représenté par un point \bullet et n'appartient pas à la surface paramétrique, le choix initial de la procédure itérative est représenté par un point \bullet , le choix initial optimisé est le point \bullet . La solution de la procédure itérative – le point le plus proche de \mathbf{M} sur la spline est le point \bullet .

1. **identification de l'impacteur $\mathbf{M} = (x, y, z)$.** \bullet
2. **détermination du patch $\{S(u, v), u \in [\mu_i; \mu_{i+1}] \text{ et } v \in [\nu_j; \nu_{j+1}]\}$ le plus proche de \mathbf{M}** : la distance entre chaque point de données $\mathbf{P}_{i,j}$ et le point impacteur \mathbf{M} est calculée. En notant les numéros de patch auxquels appartiennent chacun des points les plus proches, il est possible de déterminer le patch le plus proche (ici, le numéro du patch le plus proche est choisi comme celui apparaissant le premier trois fois parmi les numéros de patch autour des points les plus proches pris dans l'ordre croissant de leur distance à l'impacteur) ;
3. **le point $S(\frac{\mu_i + \mu_{i+1}}{2}, \frac{\nu_j + \nu_{j+1}}{2})$ est choisi arbitrairement comme point de départ de la procédure itérative** ; \bullet
4. **un quadrillage de la zone $\{S(u, v), u \in [\mu_i - 1; \mu_{i+1} + 1] \text{ et } v \in [\nu_j - 1; \nu_{j+1} + 1]\}$ permet d'optimiser le point de départ de la procédure itérative, appelé point $\hat{\mathbf{M}}_0$** : Afin d'optimiser la vitesse de convergence de la procédure de Newton-Raphson il est en effet fondamental de s'assurer de la *qualité* du premier point le plus proche détecté. Le choix est fait de scanner (avec un pas $\delta u = 0,25$ et $\delta v = 0,25$) un domaine s'étendant sur la longueur d'un patch autour du patch le plus proche de l'impacteur pour déterminer le point $\hat{\mathbf{M}}_0$. \bullet
5. **procédure itérative de Newton-Raphson :**

$$\hat{\mathbf{M}}_{i+1} = f(\hat{\mathbf{M}}_i), \text{ pour } i \geq 0 \quad (71)$$

Plus précisément, les itérations sont faites dans l'espace des paramètres intrinsèques en partant des coordonnées (u_0, v_0) du point $\hat{\mathbf{M}}_0$. Les coordonnées intrinsèques du point le plus proche déterminé à l'itération i , $\hat{\mathbf{M}}_i$, sont notées (u_i, v_i) . On définit

$$r(u_i, v_i) = S(u_i, v_i) - \mathbf{M}; \quad (72)$$

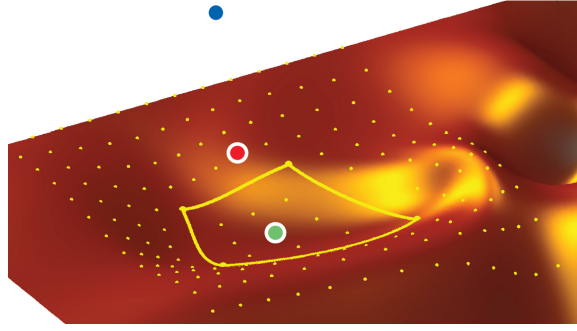


Figure 14 – Détermination du patch le plus proche (délimité par les lignes jaunes) de l'impacteur

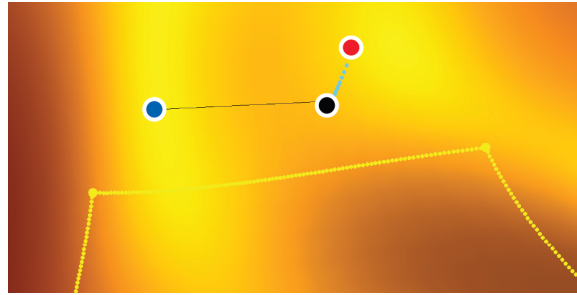


Figure 15 – Procédure itérative

puis la matrice Jacobienne \mathbf{J}_i

$$\mathbf{J}_i = \begin{bmatrix} S_{,u}(u_i, v_i)^2 + r(u_i, v_i)S_{,uu}(u_i, v_i) & S_{,u}(u_i, v_i)S_{,v}(u_i, v_i) + r(u_i, v_i)S_{,uv}(u_i, v_i) \\ S_{,u}(u_i, v_i)S_{,v}(u_i, v_i) + r(u_i, v_i)S_{,uv}(u_i, v_i) & S_{,v}(u_i, v_i)^2 + r(u_i, v_i)S_{,vv}(u_i, v_i) \end{bmatrix} \quad (73)$$

L'incrémentation \mathbf{D}_i des coordonnées intrinsèques est déterminée par

$$\mathbf{D}_i = - \begin{bmatrix} S_{,u}(u_i, v_i)r(u_i, v_i) \\ S_{,v}(u_i, v_i)r(u_i, v_i) \end{bmatrix} \mathbf{J}_i^{-1} \quad (74)$$

et on obtient les coordonnées (u_{i+1}, v_{i+1}) du point $\hat{\mathbf{M}}_{i+1}$

$$(u_{i+1}, v_{i+1}) = (u_i, v_i) + \mathbf{D}_i^T \quad (75)$$

6. à l'itération k , lorsque le critère d'arrêt ϵ_k (choisi ici comme $\|\text{dist}(\mathbf{M}\hat{\mathbf{M}}_{k+1}) - \text{dist}(\mathbf{M}\hat{\mathbf{M}}_k)\| \leq 10^{-8}$) est atteint, le point le plus proche de l'impacteur \mathbf{M} sur la spline est $\hat{\mathbf{M}} = \hat{\mathbf{M}}_k$.

3.2 Cas des surfaces fermées

Afin d'assurer la robustesse du code développé, il est nécessaire de prendre en compte le cas où l'impacteur se situe à proximité de la zone de recoupement d'une surface fermée (voir ligne rouge sur la figure 16). Au niveau de ce recoupement, la discontinuité d'un des deux paramètres intrinsèques (u sur la figure 16) impose de prendre quelques précautions.

À partir des valeurs minimale μ_{\min} et maximale μ_{\max} permises pour le paramètre intrinsèque u , la fonction développée permet de vérifier si la valeur de ce paramètre intrinsèque n'est pas au-delà des limites permises. Si c'est le cas, une rectification de la valeur du paramètre est faite afin d'assurer la

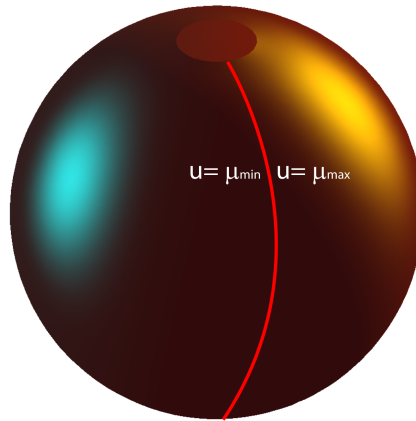


Figure 16 – Exemple de surface fermée (sphère partielle)

convergence de la procédure Newton-Raphson :

$$\begin{aligned}
 &\text{si } u > \mu_{\max} \\
 &\quad u = (u - \mu_{\max}) + \mu_{\min} \\
 &\text{si } u < \mu_{\min} \\
 &\quad u = \mu_{\max} - (\mu_{\min} - u)
 \end{aligned} \tag{76}$$

Cette correction de l'incrémentation n'a lieu que si la surface créée se referme sur elle-même.

3.3 Validation

La validation de l'algorithme de détection est faite en comparant les résultats obtenus à ceux donnés par la fonction d'optimisation *fmincon* de Matlab (avec méthode du point intérieur). À partir d'une nappe de points quelconques au sein d'un domaine rectangulaire dans le plan (O, \vec{x}, \vec{y}) (voir figure 17), la côte de chaque point est calculée aléatoirement. D'un point de donnée à son voisin (dans la direction \vec{x} ou \vec{y}), la variation maximale de la côte est imposée par le paramètre δz .

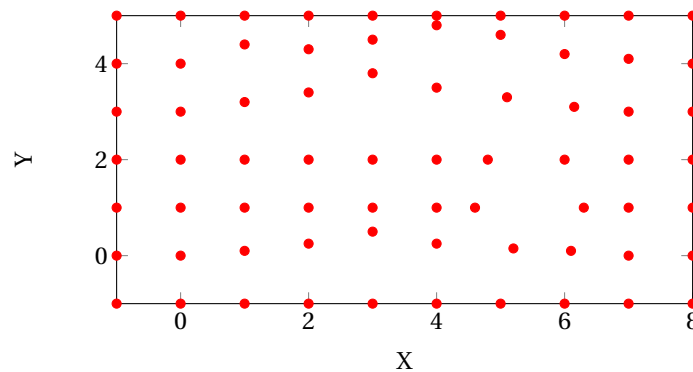


Figure 17 – Points de données utilisés pour la validation représentés dans le plan (O, \vec{x}, \vec{y})

Des simulations de Monte-Carlo sont effectuées pour trois valeurs du paramètre δz : 0,1 ; 0,5 et 1. À titre d'exemple, la figure 18 montre des niveaux d'ondulation obtenus pour $\delta z = 1$. À titre indicatif, dans le cadre d'un algorithme de contact *point/surface*, où la spline représenterait la surface de contact de la structure esclave et l'impacteur un nœud de la structure maître, le niveau d'ondulation est estimé (pour des calculs en grandes déformations) autour de $\delta z = 0,15$.

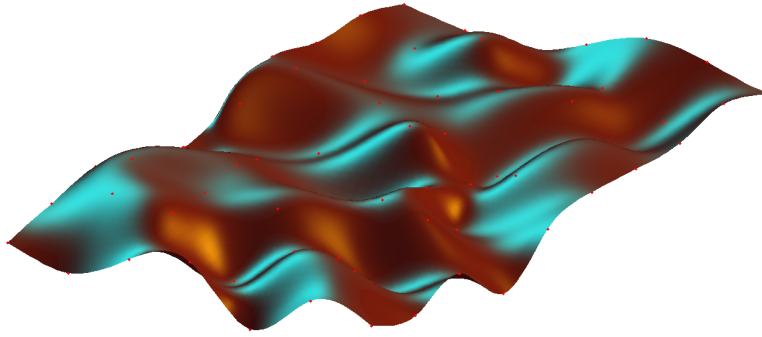


Figure 18 – Points de données de la figure 17 représentés en 3D pour $\delta z = 1$ ($z_{\min} = -0,5$ et $z_{\max} = 0,5$)

Pour chaque valeur de δz , 1000 procédures de détection sont effectuées sur une surface à l'ondulation aléatoire avec un impacteur $\mathbf{M} = (3,5; 2; 3)$, les résultats obtenus sont récapitulés dans le tableau suivant :

δz	Nombre de simulations	Nombre d'erreurs ⁴	taux de réussite
0,1	1000	0	100 %
0,5	1000	1 ⁵	99,9 %
1	1000	2 ⁶	99,8 %

1. Il est considéré que la procédure de détection est fautive lorsque les écarts entre les paramètres intrinsèques obtenus par la procédure proposée et ceux obtenus avec la routine *fmincon* de Matlab dépassent 10^{-6} .

2, 3. La procédure proposée a convergé mais le minimum local détecté est différent du minimum local détecté par *fmincon*.

4 Code Matlab

Le code Matlab fourni permet la création d'une surface paramétrique spline bi-cubique. Le fichier principal est **Spline.m**. Il est possible de créer tout type de surface en respectant le format de saisie des points de données fourni dans les fichiers **quasi_sphere.m** et **jeu_de_points_3.m**.

Une archive contenant tous les fichiers au format Matlab est disponible [ici](#).

1. Spline.m

```
clear;clc;
try;clf(1);clf(2);end;
addpath ./formes
addpath ./fonctions
% PARAMÈTRE (1=oui)           % VALEUR TYPE
%
visualisation=1; % représentation graphique ?           % 1
%
surface_var=1; % surface avec effets de lumière ?           % 1
%
resolution=10; % résolution graphique de la surface (par patch) % 5~10
%
approximation=0; % 0 => interpolation, 1 => approximation % 0
%
PEC_CL=1; % Prise en compte des conditions limites

% 1 - Choix de la surface à représenter (cf. répertoire "formes")
%           jeu_de_points_3
quasi_sphere

% 2 - Sauvegarde éventuelle des points de passage
spline.Px=double(Px);
spline.Py=double(Py);
spline.Pz=double(Pz);
if approximation==1
    save P Px Py Pz
end

% 3 - Calcul de la spline
M=[0 0 0];
nlines=1;
[u,v,sol]=code_principal(spline,M,nlines,visualisation,surface_var,...
...resolution,approximation,PEC_CL);
```

2. code_principal.m

```
function [u,v,sol]=code_principal(spline,M,nlines,visualisation,...
...surface_var,resolution,approximation,PEC_CL);
```

```

% premiere ligne/colonne = derniere ligne/colonne ?
% La fonction "rep_valeurs" determine si la surface a creer est fermee ou
% ouverte
if PEC_CL==1
    [repetition,spline]=rep_valeurs(spline);
else
    repetition=0;
end

% détermination du nombre de points
[spline.n,spline.m]=size(spline.Px);
if approximation==1
    load P
    spline.Qx=spline.Px;spline.Qy=spline.Py;spline.Qz=spline.Pz;
    spline.n=spline.n-2;spline.m=spline.m-2;
    sol=[0 0];
else

    % calcul de la matrice A (lien entre points de maillage et points de
% contrôle intérieurs)
    A=zeros(9*spline.n*spline.m,3);
    for i=[1:1:spline.n];
        for j=[1:1:spline.m];
            plac=((i-1)*spline.m+j);
            A((plac-1)*9+1:plac*9,:)=fct_ligneMAT(i,j,spline.m);
        end
    end
    A=spconvert(A);

    % calcul de la matrice C (lien entre points de contrôle intérieurs et
% tous les autres points de contrôle en supposant des conditions
% limites de type "bords libres").
    if repetition==1
        C(:,:)=fct_MatrixCRep(spline.n,spline.m);
    else
        C(:,:)=fct_MatrixC(spline.n,spline.m);
    end

    % retour à une forme matricielle cohérente avec les matrices Px, Py
% et Pz
    Ai=(A*C)\speye(spline.n*spline.m);
    spline.Qx=fct_ColToMat(Ai,C,spline.Px,spline.n,spline.m);
    spline.Qy=fct_ColToMat(Ai,C,spline.Py,spline.n,spline.m);
    spline.Qz=fct_ColToMat(Ai,C,spline.Pz,spline.n,spline.m);
end

% paramètres intrinsèques, domaine d'étude
spline.t_line(1,:)=[1:1:(spline.n+5)];
spline.t_col(1,:)=[1:1:(spline.m+5)];
interval_v=[spline.t_col(1,3):1/resolution:spline.t_col(1,(end-3))];
interval_u=[spline.t_line(1,3):1/resolution:spline.t_line(1,(end-3))];

```

```

% Table de connectivité
spline.PatchM=zeros((spline.m-1)*(spline.n-1),4);
for k=[0:1:(spline.n-2)]
    for i=[1:1:(spline.m-1)]
        spline.PatchM(i+k*(spline.m-1),1)=i+spline.t_col(2);
        spline.PatchM(i+k*(spline.m-1),2)=i+spline.t_col(3);
        spline.PatchM(i+k*(spline.m-1),3)=k+spline.t_line(3);
        spline.PatchM(i+k*(spline.m-1),4)=k+spline.t_line(4);
    end
end
save spline spline

% visualisation éventuelle
if visualisation==1
    fct_visualisation
end

u=0;
v=0;
sol=0;

```

3. bs3.m

```

function [Newval] = BS3(u,v,t_u,t_v,Qx,Qy,Qz);
a=(min(find(t_u>u)));
if ((t_u(a-1))==u)&&(a>4)
    a=a-1;
end

a=a-1;
tu0=t_u(a-2);%ti-2
tu=t_u(a-1);%ti-1
tu1=t_u(a);%ti
tu2=t_u(a+1);%ti+1
tu3=t_u(a+2);%ti+2
tu4=t_u(a+3);%ti+3

b=(min(find(t_v>v)));
if ((t_v(b-1))==v)&&(b>4)
    b=b-1;
end

b=b-1;
tv0=t_v(b-2);%ti-2
tv=t_v(b-1);%ti-1
tv1=t_v(b);%ti
tv2=t_v(b+1);%ti+1
tv3=t_v(b+2);%ti+2
tv4=t_v(b+3);%ti+3

```

```

V1=(tv2-v)*(tv2-v)*(tv2-v);
V2=((v-tv0)*(tv2-v)*(tv2-v)+(tv3-v)*(v-tv)*(tv2-v)+(tv3-v)*(tv3-v)*(v-tv1));
V3=((v-tv)*(v-tv)*(tv2-v)+(v-tv)*(tv3-v)*(v-tv1)+(tv4-v)*(v-tv1)*(v-tv1));
V4=(v-tv1)*(v-tv1)*(v-tv1);

U1=((tu2-u)*(tu2-u)*(tu2-u));
U2=((u-tu0)*(tu2-u)*(tu2-u)+(tu3-u)*(u-tu)*(tu2-u)+(tu3-u)*(tu3-u)*(u-tu1));
U3=((u-tu)*(u-tu)*(tu2-u)+(u-tu)*(tu3-u)*(u-tu1)+(tu4-u)*(u-tu1)*(u-tu1));
U4=((u-tu1)*(u-tu1)*(u-tu1));

Newvalx=((V1*(Qx(a-2,b-2)*U1+Qx(a-1,b-2)*U2+Qx(a,b-2)*U3+Qx(a+1,b-2)*U4))
+(V2*(Qx(a-2,b-1)*U1+Qx(a-1,b-1)*U2+Qx(a,b-1)*U3+Qx(a+1,b-1)*U4))...
+(V3*(Qx(a-2,b)*U1+Qx(a-1,b)*U2+Qx(a,b)*U3+Qx(a+1,b)*U4))...
+(V4*(Qx(a-2,b+1)*U1+Qx(a-1,b+1)*U2+Qx(a,b+1)*U3+Qx(a+1,b+1)*U4)))/36;

Newvaly=((V1*(Qy(a-2,b-2)*U1+Qy(a-1,b-2)*U2+Qy(a,b-2)*U3+Qy(a+1,b-2)*U4))...
+(V2*(Qy(a-2,b-1)*U1+Qy(a-1,b-1)*U2+Qy(a,b-1)*U3+Qy(a+1,b-1)*U4))...
+(V3*(Qy(a-2,b)*U1+Qy(a-1,b)*U2+Qy(a,b)*U3+Qy(a+1,b)*U4))...
+(V4*(Qy(a-2,b+1)*U1+Qy(a-1,b+1)*U2+Qy(a,b+1)*U3+Qy(a+1,b+1)*U4)))/36;

Newvalz=((V1*(Qz(a-2,b-2)*U1+Qz(a-1,b-2)*U2+Qz(a,b-2)*U3+Qz(a+1,b-2)*U4))...
+(V2*(Qz(a-2,b-1)*U1+Qz(a-1,b-1)*U2+Qz(a,b-1)*U3+Qz(a+1,b-1)*U4))...
+(V3*(Qz(a-2,b)*U1+Qz(a-1,b)*U2+Qz(a,b)*U3+Qz(a+1,b)*U4))...
+(V4*(Qz(a-2,b+1)*U1+Qz(a-1,b+1)*U2+Qz(a,b+1)*U3+Qz(a+1,b+1)*U4)))/36;

Newval=[Newvalx Newvaly Newvalz];

```

4. fct_visualisation.m

```

h=1;
for u=interval_u;
    k=1;
    for v=interval_v;
        % Fonction bs3 qui calcule les coordonnes en x,y,z en fonction des
        % valeurs des parametres intrinseques u et v.
        Newval=bs3(u,v,spline.t_line,spline.t_col,spline.Qx,spline.Qy,...
        ...spline.Qz);
        %
        Pointx(h,k)=Newval(1,1);
        Pointy(h,k)=Newval(1,2);
        Pointz(h,k)=Newval(1,3);
        k=k+1;
    end
    h=h+1;
end
% Tracé et lumière
figure(1);
hold on;
if surface_var==1;
surface(Pointx,Pointy,Pointz, ...

```



```

'EdgeColor','none', ...
'FaceColor',[0.65 0.15 0.15], ...%[0.65 0.15 0.15], ...
'FaceLighting','phong', ...
'AmbientStrength',0.3, ...
'DiffuseStrength',0.6, ...
'Clipping','off',...
'BackFaceLighting','reverselit', ... % 'lit' ou 'reverselit' pour inverser
% intérieur/extérieur
'SpecularStrength',1, ...
'SpecularColorReflectance',1, ...
'SpecularExponent',7, ...
'Tag','TheMathWorksLogo')
l1 = light('Position',[40 100 20], ...%[40 100 20], ... % lumière bleue
'Style','local', ...
'Color',[0 0.8 0.8]); % [0 0.8 0.8]);
l2 = light('Position',[.5 -1 .4], ...%[.5 -1 .4], ... % lumière orange
'Color',[0.8 0.8 0]); % [0.8 0.8 0]);
else
plot3(Pointx,Pointy,Pointz,'+b');
end
plot3(spline.Px(:,:),spline.Py(:,:),spline.Pz(:,:),'r');
xlabel('X');
ylabel('Y');
zlabel('Z');
grid on;
axis equal;

```

5. fct_ColToMat.m

```

function [matrix]=fct_ColToMat(A,C,P,n,m);
% calcul du vecteur contenant tous les points de contrôle en transformant
% d'abord la matrice P en vecteur
try;
    % La fonction "fct_ColumnVect" transforme la matrice des points de
    % données en vecteur colonne
    ColQ=36*C*(A*fct_ColumnVect(P,n,m));
catch;
    'erreur'
end;
% transforme le vecteur des points de contrôle en matrice
k=1;
for i=[1:1:(n+2)];
for j=[1:1:(m+2)];
matrix(i,j)=ColQ(k,1);
k=k+1;
end
end
end

```

6. fct_ColumnVect.m

```

function [colvect] = fct_ColumnVect(P,n,m);
% transforme la matrice P en vecteur
k=1;
for i=[1:1:n];
for j=[1:1:m];
colvect(k,1)=P(i,j);
k=k+1;
end
end

```

7. fct_ligneMAT.m

```

function [vecP] = fct_ligneMAT(i,j,m);
shift=(i-1)*(m+2);
vecP=zeros(9,3);
vecP(:,1)=((i-1)*m+j);
%
vecP(1,2)=shift+j;
vecP(2,2)=shift+j+1;
vecP(3,2)=shift+j+2;
vecP(4,2)=shift+m+2+j;
vecP(5,2)=shift+m+2+j+1;
vecP(6,2)=shift+m+2+j+2;
vecP(7,2)=shift+2*(m+2)+j;
vecP(8,2)=shift+2*(m+2)+j+1;
vecP(9,2)=shift+2*(m+2)+j+2;
%
vecP(1,3)=1;
vecP(2,3)=4;
vecP(3,3)=1;
vecP(4,3)=4;
vecP(5,3)=16;
vecP(6,3)=4;
vecP(7,3)=1;
vecP(8,3)=4;
vecP(9,3)=1;

```

8. fct_MatrixC.m

```

function [C] = fct_MatrixC(n,m);
C=sparse(((n+2)*(m+2)),(m*n));
% Extremités
C(1,1)=2;
C(1,m+2)=-1;
C(m+2,m)=2;
C(m+2,2*m-1)=-1;
C((n+1)*(m+2)+1,end-m+1)=2;
C((n+1)*(m+2)+1,end-2*(m-1))=-1;
C(((n+2)*(m+2)),end)=2;
C(((n+2)*(m+2)),end-(m+1))=-1;

```

```

% Premiere ligne
for i=[2:1:m+1]
    C(i,i-1)=2;
    C(i,i+m-1)=-1;
end
% Bord de gauche
for k=[1:1:n]
    C(k*(m+2)+1,(k-1)*m+1)=2;
    C(k*(m+2)+1,(k-1)*m+2)=-1;
end
% Bord de droite
for k=[2:1:n+1]
    C(k*(m+2),(k-1)*m)=2;
    C(k*(m+2),(k-1)*m-1)=-1;
end
% Derniere ligne
k=(n*m)-m;
for i=[((n+2)*(m+2)-m):1:(((n+2)*(m+2))-1)]
    C(i,k+1)=2;
    C(i,k+1-m)=-1;
    k=k+1;
end
% Points de controle a l'interieur des patches
j=1;
for k=[1:1:n]
    a=(k*(m+2)+2);
    b=(k*(m+2)+1+m);
    for i=[a:1:b]
        C(i,j)=1;
        j=j+1;
    end
end
end

```

9. fct_MatrixCRep.m

```

function [C] = fct_MatrixCRep(n,m);
C=sparse(((n+2)*(m+2)),(m*n));
%Extremities
% Derniers points de controle = aux premiers
% Point de controle Q(1,1)
C(1,1)=2;
C(1,2)=-1;
C(1,end-m+2)=1;
C(1,end-m+1)=-2;
C(1,end-2*m+2)=-1;
C(1,end-2*m+1)=2;
% Point de controle Q(1,m+2)
C(m+2,m-1)=-1;
C(m+2,m)=2;
C(m+2,end)=-2;
C(m+2,end-1)=1;

```

```

C(m+2,end-m)=2;
C(m+2,end-m-1)=-1;
% Point de controle Q(n+2,1)
C((n+1)*(m+2)+1,1)=-2;
C((n+1)*(m+2)+1,2)=1;
C((n+1)*(m+2)+1,m+1)=2;
C((n+1)*(m+2)+1,m+2)=-1;
C((n+1)*(m+2)+1,end-m+2)=-1;
C((n+1)*(m+2)+1,end-m+1)=2;
% Point de controle Q(n+2,m+2)
C((n+2)*(m+2),m-1)=1;
C((n+2)*(m+2),m)=-2;
C((n+2)*(m+2),2*m-1)=-1;
C((n+2)*(m+2),2*m)=2;
C((n+2)*(m+2),end)=2;
C((n+2)*(m+2),end-1)=-1;
% Premiere ligne
for f=1:1:m
    C(f+1,f)=1;
    C(f+1,end-m+f)=-1;
    C(f+1,end-2*m+f)=1;
end
% Bord de gauche
for k=[1:1:n]
    C(k*(m+2)+1,(k-1)*m+1)=2;
    C(k*(m+2)+1,(k-1)*m+2)=-1;
end
% Bord de droite
for k=[2:1:n+1]
    C(k*(m+2),(k-1)*m)=2;
    C(k*(m+2),(k-1)*m-1)=-1;
end
% Derniere ligne
for g=1:1:m
    C((n+1)*(m+2)+g+1,g)=-1;
    C((n+1)*(m+2)+g+1,g+m)=1;
    C((n+1)*(m+2)+g+1,end-m+g)=1;
end
% Points de controle a l'interieur des patches
j=1;
for k=[1:1:n]
    a=(k*(m+2)+2);
    b=(k*(m+2)+1+m);
    for i=[a:1:b]
        C(i,j)=1;
        j=j+1;
    end
end
end

```

10. rep_valeurs.m

```
function [repetition,spline]=rep_valeurs(spline)
repetition=0;
if ((abs(spline.Px(1,:)-spline.Px(end,:))<1e-8) & ((abs(spline.Py(1,:)-...
...spline.Py(end,:))<1e-8) & ((abs(spline.Pz(1,:)-spline.Pz(end,:))<1e-8))
    repetition=1;
elseif (((abs(spline.Px(:,1)-spline.Px(:,end)))<1e-8) & ((abs(spline.Py(:,1)...
...-spline.Py(:,end)))<1e-8) & ((abs(spline.Pz(:,1)-spline.Pz(:,end)))...
...<1e-8) & (repetition==0))
    spline.Px=(spline.Px)';
    spline.Py=(spline.Py)';
    spline.Pz=(spline.Pz)';
    repetition=1;
else
    repetition=0;
end
```

5 Conclusion

La procédure présentée dans cette note permet d'interpoler n'importe quel ensemble de points de l'espace avec des surfaces splines bi-cubiques (la seule hypothèse restrictive concerne l'organisation de ces points de données). L'ajout d'une méthode de détection du point le plus proche sur la spline pour tout point de l'espace permet d'envisager l'intégration de cette procédure au sein d'un code de contact. D'autres applications sont envisageables comme par exemple l'interpolation de données statistiques.

6 Annexe

6.1 Courbe paramétrique cubique

L'expression développée de l'équation (1) pour l'intervalle $[t_{h+1}, t_{h+2}]$ est

$$\begin{aligned}
\mathbf{c}(\kappa) = & \mathbf{Q}_{h-3} \left[\frac{t_{h+4} - \kappa}{t_{h+4} - t_{h+1}} \frac{t_{h+4} - \kappa}{t_{h+4} - t_{h+2}} \frac{t_{h+4} - \kappa}{t_{h+4} - t_{h+3}} \right] \\
& + \mathbf{Q}_{h-2} \left[\frac{\kappa - t_h}{t_{h+3} - t_h} \frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+1}} \frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+2}} + \frac{t_{h+4} - \kappa}{t_{h+4} - t_{h+1}} \frac{\kappa - t_{h+1}}{t_{h+3} - t_{h+1}} \frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+2}} \right. \\
& \left. + \frac{t_{h+4} - \kappa}{t_{h+4} - t_{h+1}} \frac{t_{h+4} - \kappa}{t_{h+4} - t_{h+2}} \frac{\kappa - t_{h+2}}{t_{h+3} - t_{h+2}} \right] \\
& + \mathbf{Q}_{h-1} \left[\frac{\kappa - t_h}{t_{h+3} - t_h} \frac{\kappa - t_h}{t_{h+2} - t_h} \frac{t_{h+2} - \kappa}{t_{h+2} - t_{h+1}} + \frac{\kappa - t_h}{t_{h+3} - t_h} \frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+1}} \frac{\kappa - t_{h+1}}{t_{h+2} - t_{h+1}} \right. \\
& \left. + \frac{t_{h+3} - \kappa}{t_{h+4} - t_{h+1}} \frac{\kappa - t_{h+1}}{t_{h+3} - t_{h+1}} \frac{\kappa - t_{h+1}}{t_{h+2} - t_{h+1}} \right] \\
& + \mathbf{Q}_h \left[\frac{\kappa - t_h}{t_{h+3} - t_h} \frac{\kappa - t_h}{t_{h+2} - t_{h+1}} \frac{\kappa - t_h}{t_{h+1} - t_h} \right]
\end{aligned} \tag{77}$$

6.2 B-Splines de degré 1

À partir de la formule Cox-de-Boor (4), il est possible d'établir la formule générale pour une B-Spline de degré 1.

$$B_{j,1}(\kappa) = \frac{\kappa - t_j}{t_{j+1} - t_j} B_{j,0}(\kappa) + \frac{t_{j+2} - \kappa}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) \tag{78}$$

Ainsi, dans le cas où κ appartient à $[t_{h-1}, t_h]$ inclus dans le domaine effectif, l'équation (1) se réduit aux indices :

$$j = \begin{cases} h-1 \\ h \end{cases} \tag{79}$$

En d'autres termes

$$\boxed{\kappa \in [t_h, t_{h+1}] \text{ où } 2 \leq h \leq n, \quad \mathbf{c}(\kappa) = \mathbf{Q}_{h-1} B_{h-1,1}(\kappa) + \mathbf{Q}_h B_{h,1}(\kappa)} \tag{80}$$

ce qui équivaut aussi à

$$\mathbf{c}(\kappa) = \mathbf{Q}_{h-1} \left(\frac{t_{h+2} - \kappa}{t_{h+2} - t_{h+1}} \right) + \mathbf{Q}_h \left(\frac{\kappa - t_h}{t_{h+1} - t_h} \right) \tag{81}$$

En supposant que nous utilisons un vecteur de nœuds uniforme et en substituant κ par $\kappa + t_h$, nous pouvons exprimer l'équation (81) sous forme polynomiale.

$$\mathbf{c}(\kappa) = (1 - \kappa) \mathbf{Q}_{h-1} + \kappa \mathbf{Q}_h \tag{82}$$

Ce qui nous ramène à la forme matricielle suivante

$$\mathbf{c}(\kappa) = \begin{bmatrix} \kappa & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \tag{83}$$

où $\kappa \in [t_h, t_{h+1}]$ pour $2 \leq h \leq n$.

6.2.1 Conditions limites

Après avoir substitué κ par $\kappa + t_j$ dans l'équation (81), permettant ainsi d'obtenir l'équation (82), la courbe paramétrique de degré 1 débutera à $t_2 = 0$ et prendra fin à $t_{n+1} = 1$.

1. Conditions limites pour la position

$$\mathbf{c}(t_2) = [t_2 \quad 1] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \quad (84)$$

$$\boxed{\mathbf{c}(t_2) = \mathbf{Q}_1} \quad (85)$$

$$\mathbf{c}(t_{n+1}) = [t_{n+1} \quad 1] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (86)$$

$$\boxed{\mathbf{c}(t_{n+1}) = \mathbf{Q}_n} \quad (87)$$

2. Conditions limites pour la dérivée première

$$\dot{\mathbf{c}}(\kappa) = [1 \quad 0] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (88)$$

$$\dot{\mathbf{c}}(t_2) = [1 \quad 0] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \quad (89)$$

$$\boxed{\dot{\mathbf{c}}(t_2) = -\mathbf{Q}_1 + \mathbf{Q}_2} \quad (90)$$

$$\dot{\mathbf{c}}(t_{n+1}) = [1 \quad 0] \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (91)$$

$$\boxed{\dot{\mathbf{c}}(t_{n+1}) = -\mathbf{Q}_{n-1} + \mathbf{Q}_n} \quad (92)$$

Ainsi, les conditions limites que nous obtenons sont

$$\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_2) \\ \dot{\mathbf{c}}(t_2) \end{bmatrix} \quad (93)$$

$$\begin{bmatrix} \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_{n+1}) \\ \dot{\mathbf{c}}(t_{n+1}) \end{bmatrix} \quad (94)$$

6.3 B-Spline de degré 2

À partir de la formule Cox-de-Boor (4), la fonction spline de degré 2 est

$$B_{j,2}(\kappa) = \frac{\kappa - t_j}{t_{j+2} - t_j} \left(\frac{\kappa - t_j}{t_{j+1} - t_j} B_{j,0}(\kappa) + \frac{t_{j+2} - \kappa}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) \right) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+1}} \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) \right) \quad (95)$$

Ainsi, dans le cas où κ appartient à $[t_h, t_{h+1}]$ inclus dans le domaine effectif, l'équation (1) se réduit aux indices

$$j = \begin{cases} h-2 \\ h-1 \\ h \end{cases} \quad (96)$$

En d'autres termes

$$\boxed{\kappa \in [t_h, t_{h+1}] \text{ où } 3 \leq h \leq n, \quad \mathbf{c}(\kappa) = \mathbf{Q}_h B_{h,2}(\kappa) + \mathbf{Q}_{h+1} B_{h+1,2}(\kappa) + \mathbf{Q}_{h+2} B_{h+2,2}(\kappa)} \quad (97)$$

ce qui équivaut aussi à

$$\mathbf{c}(\kappa) = \mathbf{Q}_{h-2} \left[\frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+1}} \frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+2}} \right] + \mathbf{Q}_{h-1} \left[\left(\frac{\kappa - t_h}{t_{h+2} - t_h} \frac{t_{h+2} - \kappa}{t_{h+2} - t_{h+1}} \right) + \left(\frac{t_{h+3} - \kappa}{t_{h+3} - t_{h+1}} \frac{\kappa - t_{h+1}}{t_{h+2} - t_{h+1}} \right) \right] + \mathbf{Q}_h \left[\frac{\kappa - t_h}{t_{h+2} - t_h} \frac{\kappa - t_h}{t_{h+1} - t_h} \right] \quad (98)$$

En supposant que nous utilisons un vecteur de nœuds uniforme et en substituant κ par $\kappa + t_{h-1}$, nous pouvons exprimer l'équation (98) sous forme polynomiale.

$$\mathbf{c}(\kappa) = \frac{1}{2} \left[(\kappa^2 - 2\kappa + 1) \mathbf{Q}_{h-2} + (-2\kappa^2 + 2\kappa + 1) \mathbf{Q}_{h-1} + (\kappa^2) \mathbf{Q}_h \right] \quad (99)$$

ce qui nous ramène à la forme matricielle suivante

$$\mathbf{c}(\kappa) = \frac{1}{2} \begin{bmatrix} \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (100)$$

6.3.1 Conditions limites

Après avoir substitué κ par $\kappa + t_j$ dans l'équation (98), permettant ainsi d'obtenir l'équation (99), la courbe paramétrique de degré 2 débutera à $t_3 = 0$ et prendra fin à $t_{n+1} = 1$.

1. Conditions limites pour la position

$$\mathbf{c}(t_3) = \frac{1}{2} \begin{bmatrix} t_3^2 & t_3 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \end{bmatrix} \quad (101)$$

$$\boxed{\mathbf{c}(t_3) = \frac{1}{2} (\mathbf{Q}_1 + \mathbf{Q}_2)} \quad (102)$$

$$\mathbf{c}(t_{n+1}) = \frac{1}{2} \begin{bmatrix} t_{n+1}^2 & t_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (103)$$

$$\boxed{\mathbf{c}(t_{n+1}) = \frac{1}{2} (\mathbf{Q}_{n-1} + \mathbf{Q}_n)} \quad (104)$$

2. Conditions limites pour la dérivée première

$$\dot{\mathbf{c}}(\kappa) = \frac{1}{2} \begin{bmatrix} 2\kappa & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (105)$$

$$\dot{\mathbf{c}}(t_3) = \frac{1}{2} \begin{bmatrix} t_3 & 1 \end{bmatrix} \begin{bmatrix} 2 & -4 & 2 \\ -2 & 2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \end{bmatrix} \quad (106)$$

$$\boxed{\dot{\mathbf{c}}(t_3) = (-\mathbf{Q}_1 + \mathbf{Q}_2)} \quad (107)$$

$$\dot{\mathbf{c}}(t_{n+1}) = \frac{1}{2} \begin{bmatrix} t_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 2 & -4 & 2 \\ -2 & 2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (108)$$

$$\boxed{\dot{\mathbf{c}}(t_{n+1}) = (-\mathbf{Q}_{n-1} + \mathbf{Q}_n)} \quad (109)$$

Ainsi, les conditions limites que nous obtenons sont les suivantes

$$\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_3) \\ \dot{\mathbf{c}}(t_3) \end{bmatrix} \quad (110)$$

$$\begin{bmatrix} \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_{n+1}) \\ \dot{\mathbf{c}}(t_{n+1}) \end{bmatrix} \quad (111)$$

6.4 B-Splines de degré 4

À partir de la formule Cox-de-Boor (4), la fonction spline de degré 4 est

$$\begin{aligned}
B_{j,4}(\kappa) = & \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left[\left(\frac{\kappa - t_j}{t_{j+3} - t_j} \right) \left[\left(\frac{\kappa - t_j}{t_{j+2} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+1} - t_j} B_{j,0}(\kappa) + \frac{t_{j+2} - \kappa}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) \right) \right. \right. \\
& + \left. \left. \left(\frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) \right) \right] \right. \\
& + \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+1}} \right) \left[\left(\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) \right) \right. \\
& + \left. \left. \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) + \frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} B_{j+3,0}(\kappa) \right) \right] \right] \\
& + \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left[\left(\frac{\kappa - t_{j+1}}{t_{j+4} - t_{j+1}} \right) \left[\left(\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} B_{j+1,0}(\kappa) + \frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) \right) \right. \right. \\
& + \left. \left. \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) + \frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} B_{j+3,0}(\kappa) \right) \right] \right. \\
& + \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+2}} \right) \left[\left(\frac{\kappa - t_{j+2}}{t_{j+4} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} B_{j+2,0}(\kappa) + \frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} B_{j+3,0}(\kappa) \right) \right. \\
& + \left. \left. \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+3}} \right) \left(\frac{\kappa - t_{j+3}}{t_{j+4} - t_{j+3}} B_{j+3,0}(\kappa) + \frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+4}} B_{j+4,0}(\kappa) \right) \right] \right] \right]
\end{aligned} \tag{112}$$

Ainsi, dans le cas où κ appartient à $[t_h, t_{h+1}]$ inclus dans le domaine effectif, l'équation (1) se réduit aux indices

$$j = \begin{cases} h-4 \\ h-3 \\ h-2 \\ h-1 \\ h \end{cases} \tag{113}$$

En d'autres termes

$$\boxed{\kappa \in [t_h, t_{h+1}] \text{ où } 5 \leq h \leq n, \mathbf{c}(\kappa) = \mathbf{Q}_{h-4} \mathbf{B}_{h-4,4}(\kappa) + \mathbf{Q}_{h-3} \mathbf{B}_{h-3,4}(\kappa) + \mathbf{Q}_{h-2} \mathbf{B}_{h-2,4}(\kappa) + \mathbf{Q}_{h-1} \mathbf{B}_{h-1,4}(\kappa) + \mathbf{Q}_h \mathbf{B}_{h,4}(\kappa)} \tag{114}$$

où

$$\begin{aligned}
\mathbf{B}_{h-4,4}(\kappa) &= \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_1} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+2}} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+3}} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+4}} \right) \\
\mathbf{B}_{h-3,4}(\kappa) &= \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+1}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} \right) \\
&+ \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+4} - t_{j+1}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} \right) \\
&+ \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+4} - t_{j+2}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+3}} \right) \\
&+ \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+2}} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+3}} \right) \left(\frac{\kappa - t_{j+3}}{t_{j+4} - t_{j+3}} \right)
\end{aligned} \tag{115}$$

$$\begin{aligned}
B_{h-2,4}(\kappa) = & \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+3} - t_j} \right) \left(\frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+1}} \right) \left(\frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} \right) \\
& + \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+1}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} \right) \\
& + \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+4} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \right) \left(\frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} \right) \\
& + \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+4} - t_{j+1}} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} \right) \\
& + \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+4} - t_{j+2}} \right) \left(\frac{\kappa - t_{j+2}}{t_{j+3} - t_{j+2}} \right) \\
& + \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \right) \left(\frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+2}} \right)
\end{aligned} \tag{116}$$

$$\begin{aligned}
B_{h-1,4}(\kappa) = & \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+3} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+2} - t_j} \right) \left(\frac{t_{j+2} - \kappa}{t_{j+2} - t_{j+1}} \right) \\
& + \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+3} - t_j} \right) \left(\frac{t_{j+3} - \kappa}{t_{j+3} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} \right) \\
& + \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{t_{j+4} - \kappa}{t_{j+4} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} \right) \\
& + \left(\frac{t_{j+5} - \kappa}{t_{j+5} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+4} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+3} - t_{j+1}} \right) \left(\frac{\kappa - t_{j+1}}{t_{j+2} - t_{j+1}} \right)
\end{aligned} \tag{117}$$

$$B_{h,4}(\kappa) = \left(\frac{\kappa - t_j}{t_{j+4} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+3} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+2} - t_j} \right) \left(\frac{\kappa - t_j}{t_{j+1} - t_j} \right) \tag{118}$$

En supposant que nous utilisons un vecteur de nœuds uniforme et en substituant κ par $\kappa + t_{h-1}$, nous pouvons exprimer l'équation (114) sous forme polynomiale.

$$\begin{aligned}
\mathbf{c}(\kappa) = \frac{1}{24} \left[\right. & (\kappa^4 - 4\kappa^3 + 6\kappa^2 - 4\kappa + 1) \mathbf{Q}_{h-4} + (-4\kappa^4 + 12\kappa^3 - 6\kappa^2 - 12\kappa + 11) \mathbf{Q}_{h-3} \\
& + (6\kappa^4 - 12\kappa^3 - 6\kappa^2 + 12\kappa + 11) \mathbf{Q}_{h-2} + (-4\kappa^4 + 4\kappa^3 + 6\kappa^2 + 4\kappa + 1) \mathbf{Q}_{h-1} + (\kappa^4) \mathbf{Q}_h \left. \right] \tag{119}
\end{aligned}$$

Ce qui nous ramène à la forme matricielle suivante

$$\mathbf{c}(\kappa) = \frac{1}{24} \begin{bmatrix} \kappa^4 & \kappa^3 & \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-4} \\ \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \tag{120}$$

6.4.1 Conditions limites

Après avoir substitué κ par $\kappa + t_h$ dans l'équation (114), permettant ainsi d'obtenir l'équation (119), la courbe paramétrique de degré 4 débutera à $t_5 = 0$ et prendra fin à $t_{n+1} = 1$.

1. Conditions limites pour la position

$$\mathbf{c}(t_5) = \frac{1}{24} \begin{bmatrix} t_5^4 & t_5^3 & t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \\ \mathbf{Q}_5 \end{bmatrix} \quad (121)$$

$$\boxed{\mathbf{c}(t_5) = \frac{1}{24} (\mathbf{Q}_1 + 11\mathbf{Q}_2 + 11\mathbf{Q}_3 + \mathbf{Q}_4)} \quad (122)$$

$$\mathbf{c}(t_{n+1}) = \frac{1}{24} \begin{bmatrix} t_{n+1}^4 & t_{n+1}^3 & t_{n+1}^2 & t_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-4} \\ \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (123)$$

$$\boxed{\mathbf{c}(t_{n+1}) = \frac{1}{24} (\mathbf{Q}_{n-3} + 11\mathbf{Q}_{n-2} + 11\mathbf{Q}_{n-1} + \mathbf{Q}_n)} \quad (124)$$

2. Conditions limites pour la dérivée première

$$\dot{\mathbf{c}}(\kappa) = \frac{1}{24} \begin{bmatrix} \kappa^3 & \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} 4 & -16 & 24 & -16 & 4 \\ -12 & 36 & -36 & 12 & 0 \\ 12 & -12 & -12 & 12 & 0 \\ -4 & -12 & 12 & 4 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-4} \\ \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (125)$$

$$\dot{\mathbf{c}}(t_5) = \frac{1}{24} \begin{bmatrix} t_5^3 & t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} 4 & -16 & 24 & -16 & 4 \\ -12 & 36 & -36 & 12 & 0 \\ 12 & -12 & -12 & 12 & 0 \\ -4 & -12 & 12 & 4 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \\ \mathbf{Q}_5 \end{bmatrix} \quad (126)$$

$$\boxed{\dot{\mathbf{c}}(t_5) = \frac{1}{24} (-4\mathbf{Q}_1 - 12\mathbf{Q}_2 + 12\mathbf{Q}_3 + 4\mathbf{Q}_4)} \quad (127)$$

$$\dot{\mathbf{c}}(t_{n+1}) = \frac{1}{24} \begin{bmatrix} t_{n+1}^3 & t_{n+1}^2 & t_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 4 & -16 & 24 & -16 & 4 \\ -12 & 36 & -36 & 12 & 0 \\ 12 & -12 & -12 & 12 & 0 \\ -4 & -12 & 12 & 4 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-4} \\ \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (128)$$

$$\boxed{\dot{\mathbf{c}}(t_{n+1}) = \frac{1}{24} (-4\mathbf{Q}_{n-3} - 12\mathbf{Q}_{n-2} + 12\mathbf{Q}_{n-1} + 4\mathbf{Q}_n)} \quad (129)$$

3. Conditions limites pour la dérivée seconde

$$\ddot{\mathbf{c}}(\kappa) = \frac{1}{24} \begin{bmatrix} \kappa^2 & \kappa & 1 \end{bmatrix} \begin{bmatrix} 12 & -48 & 72 & -48 & 12 \\ -24 & 72 & -72 & 24 & 0 \\ 12 & -12 & -12 & 12 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-4} \\ \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (130)$$

$$\ddot{\mathbf{c}}(t_5) = \frac{1}{24} \begin{bmatrix} t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} 12 & -48 & 72 & -48 & 12 \\ -24 & 72 & -72 & 24 & 0 \\ 12 & -12 & -12 & 12 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \\ \mathbf{Q}_5 \end{bmatrix} \quad (131)$$

$$\boxed{\ddot{\mathbf{c}}(t_5) = \frac{1}{24} (12\mathbf{Q}_1 - 12\mathbf{Q}_2 - 12\mathbf{Q}_3 + 12\mathbf{Q}_4)} \quad (132)$$

$$\ddot{\mathbf{c}}(t_{n+1}) = \frac{1}{24} \begin{bmatrix} t_{n+1}^2 & t_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 12 & -48 & 72 & -48 & 12 \\ -24 & 72 & -72 & 24 & 0 \\ 12 & -12 & -12 & 12 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-4} \\ \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (133)$$

$$\boxed{\ddot{\mathbf{c}}(t_{n+1}) = \frac{1}{24} (12\mathbf{Q}_{n-3} - 12\mathbf{Q}_{n-2} - 12\mathbf{Q}_{n-1} + 12\mathbf{Q}_n)} \quad (134)$$

4. Conditions limites pour la dérivée tertiaire

$$\dddot{\mathbf{c}}(\kappa) = \frac{1}{24} \begin{bmatrix} \kappa & 1 \end{bmatrix} \begin{bmatrix} 24 & -96 & 144 & -96 & 24 \\ -24 & 72 & -72 & 24 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{h-4} \\ \mathbf{Q}_{h-3} \\ \mathbf{Q}_{h-2} \\ \mathbf{Q}_{h-1} \\ \mathbf{Q}_h \end{bmatrix} \quad (135)$$

$$\dddot{\mathbf{c}}(t_5) = \frac{1}{24} \begin{bmatrix} t_5 & 1 \end{bmatrix} \begin{bmatrix} 124 & -96 & 144 & -96 & 24 \\ -24 & 72 & -72 & 24 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \\ \mathbf{Q}_5 \end{bmatrix} \quad (136)$$

$$\boxed{\dddot{\mathbf{c}}(t_5) = \frac{1}{24} (-24\mathbf{Q}_1 + 72\mathbf{Q}_2 - 72\mathbf{Q}_3 + 24\mathbf{Q}_4)} \quad (137)$$

$$\dddot{\mathbf{c}}(t_{n+1}) = \frac{1}{24} \begin{bmatrix} t_{n+1} & 1 \end{bmatrix} \begin{bmatrix} 124 & -96 & 144 & -96 & 24 \\ -24 & 72 & -72 & 24 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{n-4} \\ \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} \quad (138)$$

$$\boxed{\dddot{\mathbf{c}}(t_{n+1}) = \frac{1}{24} (-24\mathbf{Q}_{n-3} + 72\mathbf{Q}_{n-2} - 72\mathbf{Q}_{n-1} + 24\mathbf{Q}_n)} \quad (139)$$

Ainsi, les conditions limites que nous obtenons sont les suivantes

$$\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \end{bmatrix} = \frac{1}{12} \begin{bmatrix} 12 & -18 & 11 & -3 \\ 12 & -6 & -1 & 1 \\ 12 & 6 & -1 & -1 \\ 12 & 18 & 11 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_5) \\ \dot{\mathbf{c}}(t_5) \\ \ddot{\mathbf{c}}(t_5) \\ \ddot{\mathbf{c}}(t_5) \end{bmatrix} \quad (140)$$

$$\begin{bmatrix} \mathbf{Q}_{n-3} \\ \mathbf{Q}_{n-2} \\ \mathbf{Q}_{n-1} \\ \mathbf{Q}_n \end{bmatrix} = \frac{1}{12} \begin{bmatrix} 12 & -18 & 11 & -3 \\ 12 & -6 & -1 & 1 \\ 12 & 6 & -1 & -1 \\ 12 & 18 & 11 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{c}(t_{n+1}) \\ \dot{\mathbf{c}}(t_{n+1}) \\ \ddot{\mathbf{c}}(t_{n+1}) \\ \ddot{\mathbf{c}}(t_{n+1}) \end{bmatrix} \quad (141)$$

6.5 NURBS : Non-Rational Uniform B-Splines

Les courbes paramétriques faisant appel aux NURBS sont définies par l'équation suivante :

$$\mathbf{c}(\kappa) = \frac{1}{\sum_{j=1}^n B_{j,d}(\kappa) \mathbf{w}_j} \sum_{j=1}^n \mathbf{Q}_j B_{j,d}(\kappa) \mathbf{w}_j \quad (142)$$

où

- d est le degré de la courbe paramétrique ;
- n est le nombre total de points de contrôle ;
- $B_{j,d}$ sont les fonctions splines polynomiales ;
- \mathbf{Q}_j sont les points de contrôle ;
- \mathbf{w}_j est une valeur du vecteur de masse associée à un point de contrôle \mathbf{Q}_j .

Pour chaque valeur de κ appartenant à l'intervalle choisie du vecteur de nœuds t (défini plus bas), un point de contrôle \mathbf{Q}_j est associé à une unique fonction spline $B_{j,d}$ qui, lorsque non-nulle, permet à la fonction $\mathbf{c}(\kappa)$ de retourner un point appartenant à la courbe paramétrique. Chaque fois qu'une intervalle de nœuds est complétée, il en est de même pour un morceau de la courbe paramétrique. Ainsi, il est possible de créer une courbe polynomiale paramétrique par morceau (appelé *patch*). Le nombre de patches dépend entièrement du nombre d'intervalles de nœuds permis et ainsi, du nombre de points de contrôle et du degré de la courbe. Les seules différences entre les NURBS et les B-Splines sont le type de vecteur de noeuds choisi et le vecteur de masse \mathbf{w} .

Le vecteur de noeuds pour les NURBS est de type non-uniforme, c'est-à-dire que l'espacement entre deux valeurs de noeuds consécutifs n'est pas constant. À titre, d'exemple :

$$\begin{aligned} \mathbf{t} &= [t_1 \ t_2 \ t_3 \ \dots \ t_{n+d-1} \ t_{n+d} \ t_{n+d+1}] \\ &= [1 \ 1.5 \ 5 \ 8 \ 9 \ 11 \ \dots \ n+d-1 \ n+d \ n+d+1] \end{aligned} \quad (143)$$

Toutefois, la définition des fonctions splines restent la même, à la seule différence près que nous ne pouvons pas exprimer les fonctions splines résultantes sous forme polynomiale et ainsi, sous forme matricielle.

En ce qui concerne le vecteur de masse \mathbf{w}_j , chaque valeur du vecteur est associée à un point de contrôle \mathbf{Q}_j . La valeur de la masse affecte l'effet qu'a le point de contrôle sur la forme du patch auquel il est associé. Cet effet peut être comparé à un aimant : si $\mathbf{w}_j > 1$, le patch auquel appartient le point de contrôle \mathbf{Q}_j associé se rapproche du point de contrôle. Si $\mathbf{w}_j < 0$, le patch auquel appartient le point de contrôle \mathbf{Q}_j associé s'éloigne du point de contrôle. Finalement, si $\mathbf{w}_j = 1$, aucun effet n'est appliqué sur le patch auquel appartient le point de contrôle \mathbf{Q}_j associé.