



HAL
open science

Interpolation C de resultats C

S. Gopalsamy, Olivier Pironneau

► **To cite this version:**

S. Gopalsamy, Olivier Pironneau. Interpolation C de resultats C. RR-1000, INRIA. 1989. inria-00075559

HAL Id: inria-00075559

<https://inria.hal.science/inria-00075559>

Submitted on 24 May 2006

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.

IRIA

UNITE DE RECHERCHE
IRIA-ROCOUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tel (1) 39 63 55 11

Rapports de Recherche

N° 1000

Programme 7

**INTERPOLATION C^1
DE RESULTATS C^0**

**Sankarappan GOPALSAMY
Olivier PIRONNEAU**

Mars 1989



INTERPOLATION C^1 DE RESULTATS C^0

Sankarappan Gopalsamy (INRIA)

Olivier Pironneau (Université Pierre et Marie Curie et INRIA)

Résumé

Dans cette note on étudie une implémentation de l'algorithme de Farin pour l'interpolation C^1 de surface C^0 connue seulement par une triangulation. Cet algorithme a été proposé pour la CAO mais il nous est apparu utile pour la représentation graphique de résultats tridimensionnels issu des programmes d'éléments finis. L'algorithme à été incorporé au logiciel VISION du projet.

C^1 INTERPOLATION OF C^0 RESULTS

Abstract

In this note we report on the implementation of Farin's algorithm for the C^1 interpolation of C^0 surfaces known only by a triangulation. The algorithm was originally developed as a tool for computer aided design softwares but it appears to be useful for the visualisation of 3D computer outputs of Finite element softwares in numerical analysis.

INTERPOLATION C^1 DE RESULTATS C^0

INTRODUCTION

Une équation aux dérivées partielles résolue par la méthode des éléments finis de degré 1 fournit en général un résultat continu (C^0) mais non continuellement différentiable (C^1). La visualisation graphique de tels résultats laissent souvent apercevoir le maillage et si on veut supprimer cet effet parfois néfaste il faut interpoler les résultats par une fonction C^1 . L'algorithme de Farin [1], inventé pour le raccord des carreaux de Bézier, est aussi bien adapté à notre problème.

Le besoin d'une interpolation C^1 à des fins d'affichage graphique peut se faire sentir dans au moins trois situations:

(i) L'interpolation C^1 d'une fonction C^0 (continue) affine par morceaux sur une triangulation d'un domaine bidimensionnel pour une visualisation des résultats en perspective par exemple.

(ii) L'interpolation C^1 d'une surface (cf. figure 1) dont on ne connaît que sa triangulation et éventuellement les normales aux sommets de la triangulation.

(ii) L'interpolation C^1 d'une fonction C^0 affine par morceaux et définie sur une triangulation d'une surface de R^3 pour une visualisation par une carte de couleur affichée sur la surface.

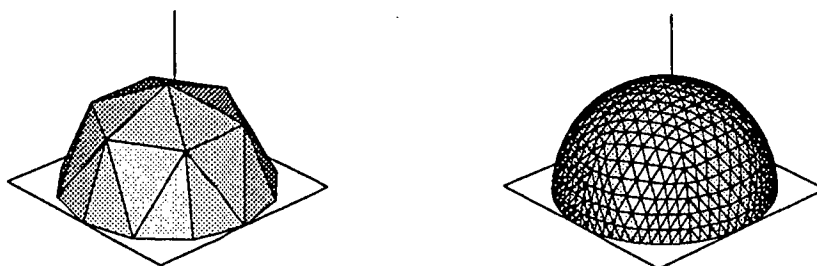


Figure 1: Exemple de résultat: avant lissage (gauche) après (droite). Les deux figures sont une représentation par facette d'une demi sphère mais dans la figure 1 (droite) les sommets des triangles ont été calculés par une interpolation C^1 des grands triangles, c'est-à-dire qu'aucune information supplémentaire que ce qu'on voit sur la figure de gauche n'a été utilisé.

L'objet de ce chapitre est de montrer comment les outils de la CAO peuvent être appliqués à l'analyse numérique et dans notre cas plus particulièrement à l'affichage graphique de résultat. Il n'y a donc rien de conceptuellement nouveau sur le plan algorithmique et l'originalité de l'étude réside dans l'implémentation de l'algorithme de Farin et l'incorporation du module dans le logiciel graphique de l'INRIA, VISION.

L'algorithme de Farin à été développé principalement pour le raccord C^1 de carreaux de Bezier, c'est à dire pour une situation très proche du cas (ii). Il est construit à partir de notion très voisines de celles utilisé par Hsieh, Clough et Tocher (HCT) pour leur élément fini C^1 . Nous choisirons donc d'exposer la méthode d'abord sur le cas (ii). Ensuite nous discuterons les deux autres cas.

2. GENERALITES

2.1 Position du problème

Une surface dont on ne connaîtrait que sa triangulation s'affiche comme une mosaïque de triangles tangents. Lorsque les triangles sont petits, on a une impression de régularité (figure 1b) mais lorsque la triangulation est trop grossière (ce qui est en général le cas des triangulation utilisées pour les calculs) alors la représentation graphique n'est pas très esthétique (figure 1a).

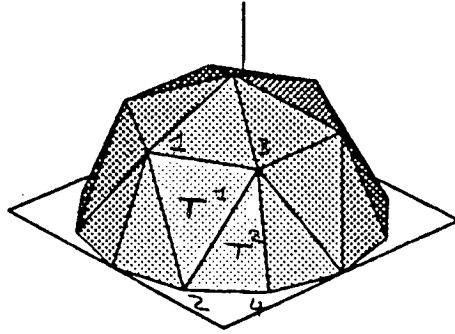


Figure 2: Affichage d'une demi-sphère par sa triangulation: $T_1^1 = 1, T_2^1 = 2, T_3^1 = 3, T_1^2 = 2, T_2^2 = 4 \dots$

Plus précisément une triangulation d'une surface S de R^3 est un ensemble ordonné de point (sommets) de R^3 , c'est-à-dire un tableau $\{P^j\}_{j=1..N}$ de vecteur, et un tableau de correspondance entre les numéro des 3 sommets de chaque triangle et leur position dans le tableau P , c'est-à-dire un tableau $\{T^k\}_{k=1..M}$ de triplets entiers T_1^k, T_2^k, T_3^k ; N, M désignent respectivement le nombre de sommets et le nombre de triangles. La surface est, interpolée linéairement à partir des sommets, c'est-à-dire que les autres points de la surface sont les combinaisons linéaires convexes des trois sommets de chaque triangle:

$$S = \left\{ \sum_{i=1,2,3} a_i P[T_i^k] : \forall a_1, a_2, a_3 \geq 0, \text{ avec } a_1 + a_2 + a_3 = 1; \forall k \right\} \quad (2.1)$$

On cherche donc une autre définition de S , aussi proche que possible que (2.1) mais qui soit C^1 .

Ce problème est identique à la recherche d'élément fini C^1 (cf. Ciarlet [2]) or on sait qu'il n'existe pas d'interpolation polynomiale C^1 de degré inférieur à 4: l'élément d'Argyris est de degré 5, l'élément HCT est de degré 3 mais il faut décomposer l'interpolation en trois polynomes et couper le triangle en trois sous triangle. On rappelle que l'élément HCT a 12 degrés de liberté disposés comme sur la figure 3:

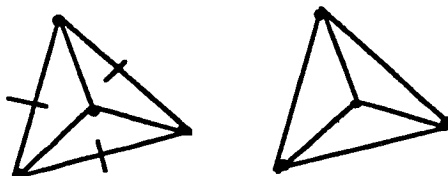


Figure 3: Dans l'élément HCT, sur chaque triangle les fonctions sont interpolées par 3 polynomes de degré 3, sur chacun des trois sous-triangles, à partir de leurs valeurs aux 3 sommets, des deux dérivées partielles en chaque sommet ainsi que par les dérivées normales aux 3 cotés du grand triangle. La position du noeud central peut être le centre du triangle. La continuité C^1 d'un triangle à l'autre est une conséquence du choix des degrés de liberté. Il existe aussi un triangle HCT réduit sans degré de liberté aux milieux des cotés

Mais l'élément HCT n'est pas bien adapté à notre problème car il y a trop de choix à faire, choisir des valeurs pour les dérivés premières aux sommets, choix des dérivés normales aux milieux des arêtes. L'élément HCT réduit par contre se prête bien a priori à l'interpolation C^1 de résultats C^0 sauf si ce qu'on doit représenter ne se présente pas sous la forme d'une surface paramétrée par les deux premières coordonnées x, y . Une construction semblable à été proposée par Farin [1] pour notre problème mais elle est basée sur la positions des points de contrôle de Bézier plutôt que sur les valeurs des dérivés premières; elle utilise les carreaux triangulaires de Bézier d'ordre 3 et 4 et le positionnement des points de contrôle interne, noeud central y compris. Comme on va le voir la construction est plus géométrique que celle de l'élément HCT réduit; elle est plus difficile à implémenter mais elle semble stable par rapport aux choix des plan tangents à la surface. L'objet de cette étude est l'implémentation de cette méthode et l'application au visualisation graphique en éléments finis; une comparaison avec l'interpolation à partir de l'élément HCT réduit sera faite ultérieurement (Gopalsamy[3]).

2.2 Surface de Bézier

Soit $B_I^n(\mathbf{u})$ le I -ème polynome de Bernstein de degré n à deux variables:

$$B_I^n(\mathbf{u}) = \frac{n!}{i_1!i_2!i_3!} u_1^{i_1} u_2^{i_2} u_3^{i_3} \quad \mathbf{u} = (u_1, u_2, u_3) \quad (2.2)$$

avec

$$I = (i_1, i_2, i_3) \quad n = |I| = i_1 + i_2 + i_3 \quad u_1 + u_2 + u_3 = 1 \quad (2.3)$$

Une carreau de Bézier S_B se définit à partir de points de contrôle $\{b_I\}_{|I|=n} \in \mathbb{R}^3$ par:

$$S_B = \{p(\mathbf{u}) = \sum_{|I|=n} b_I B_I^n(\mathbf{u}) : u_i \geq 0 \quad \sum u_i = 1\} \quad (2.4)$$

Lorsque les points de contrôle sont disposés sur un triangle curviligne comme dans la figure 4 pour $n=3$ et $n=4$, on parle de *carreau triangulaire* et on peut approcher des surfaces très générales par morceaux avec de tels carreaux. De telles approximations sont dénomées surface de Bézier.

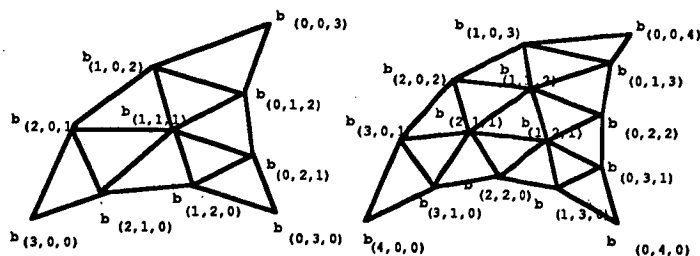


Figure 4: Points de contrôle d'un carreaux de Bézier d'ordre 3 (gauche) et d'ordre 4 (droite)

On rappelle que la surface ne passe pas nécessairement par ses points de contrôle sauf pour les sommets des triangles

Pour obtenir une approximation continue il faut évidemment que les triangles se raccordent; on démontre facilement qu'il faut et il suffit que les points de contrôles frontières de deux triangles voisins soient les mêmes.

En effet la frontière d'un carreau de Bézier est

$$\bigcup_{k=1}^3 \{p(u) : u_k = 0 \quad u_j \geq 0 \quad \sum u_j = 1\} \quad (2.5)$$

Ainsi chaque morceau de frontière est une combinaison lineaire de polynomes de Bernstein à une variable u (l'un des u_j) avec les b_I de la frontière comme poids:

$$\sum_{|I|=n, i_k=0} b_I B_I^n(u) |_{u_k=0, u_1+u_2+u_3=1} \quad (2.6)$$

or cette courbe ne dépend plus du carreau sur laquelle elle est définie mais seulement des points de contrôle du k-ème morceau de frontière.

Remarque

Dans certains articles [1],[4],[5] la notions de *visual C^1 continuity* est introduite. Ici lorsque nous parlons de continuité C^1 il s'agit bien sr de la définition mathématique usuelle.

Pour avoir un raccord C^1 entre deux carreaux aux sommets P^i on peut utiliser la proposition suivante:

Proposition 1

Si tous les points de contrôle voisin d'un sommet pour tous les triangles contenant ce sommet sont co-planaires, ceci pour chaque sommet, alors la surface de Bézier est C^1 en ce sommet.

Démonstration

On va utiliser une propriété bien connue des courbes de Bézier:

Pour que deux courbes de Bézier aient un raccord C^1 en une extrémité, il suffit que la droite définie par les deux points de contrôle de part et d'autre de cette extrémité contienne celle-ci.

En effet ceci vient de ce que l'extrémité d'une courbe de Bézier est obtenue avec $u = 0$ ou $u = 1$ et qu'en 0 par exemple:

$$\frac{\partial B_i^n}{\partial u}(0) = -n \text{ si } i = 0, n \text{ si } i = 1 \text{ et } 0 \text{ sinon}$$

de sorte que

$$\frac{\partial \sum b_i B_i^n}{\partial u}(0) = n(b_1 - b_0)$$

Ici $B_i^n(u)$ est le i^{eme} polynome de Bernstein:

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u(1-u)^{n-i}$$

Enfin si dans (2.4) on fixe l'une des composantes u_k alors on obtient une courbe de Bézier à un parametre; celle-ci sera tangente à sa voisine issue d'un carreau voisin si les points de controles voisins sont colinéaires avec le sommet commun. Comme, toujours d'après (2.4), les points de contrôle de cette courbe sont des combinaisons lineaires des points de contrôle de la surface S_B on voit que la coplanarité des points voisins de chaque sommet entraine le raccord C^1 de chaque courbe de Bézier, section de S_B ; or ceci à son tour implique le raccord C^1 des carreaux aux sommets.

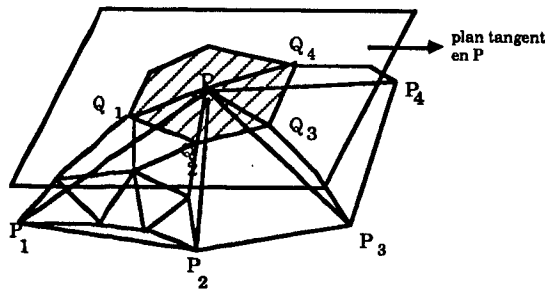


Figure 5: Continuité C^1 pour le raccord de carreaux de Bézier au sommet P

Remarque 1.

Le plan de coplanarité des voisins de chaque sommets est évidemment le plan tangent en ce sommet de la surface qu'on cherche à construire.

Pour obtenir le raccord C^1 entre deux carreaux il faut que les dérivées de $P(u)$ (cf. (2.4)) dans une direction différente de l'interface se raccordent.

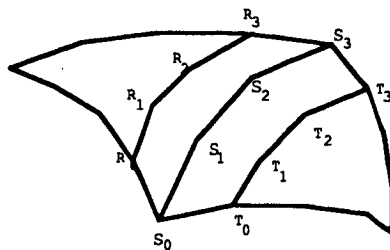


Figure 6 :raccord latéral de deux carreaux de Bézier

Proposition 2(Farin[4])

Pour obtenir un raccord C^1 le long de l'interface commune L entre deux carreaux triangulaires C_1, C_2 de Bézier de degré $n + 1$ dont les frontières seraient des courbes de Bézier de degré n il suffit qu'on ait les relations suivantes (voir figure 6):

Soit S_0, S_1, \dots, S_n les points de controles de L en temps que courbe de degré n . Soit R_0, R_1, \dots, R_n (resp T_0, T_1, \dots, T_n) les points de controle de C_1 (resp C_2) les plus proche de L

$$\exists \alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \text{ tels que, pour } 0 \leq i \leq n, \quad (2.7)$$

$$T_i = \frac{n-i}{n}(\alpha_0 R_i + \alpha_1 S_i + \alpha_2 S_{i+1}) + \frac{i}{n}(\alpha_0 R_i + \alpha_3 S_{i-1} + \alpha_4 S_i)$$

Démonstration

En temps que courbe de degré n , L peut s'écrire:

$$L(v) = \sum_{i=0}^n S_i B_i^n(v), \quad 0 \leq v \leq 1 \quad (2.8)$$

La tangente de L est donc (cf appendice, proposition 4)

$$\tau_0(v) = \frac{\partial L}{\partial v}(v) = n \sum_{i=0}^{n-1} (S_{i+1} - S_i) B_i^{n-1}(v) \quad (2.9)$$

Supposons que L soit la première courbe frontière de C_1 (i.e. obtenue avec $u_1 = 0$) et considérons la courbe G_1 définie par:

$$G_1(s) = \sum_{|I|=n} b_I^1 B_I^n(v(s)), \quad v(s) = (s, (1-s)v, (1-s)(1-v)), \quad 0 \leq s \leq 1. \quad (2.10)$$

où b_I^1 sont les points de controle de C_1 . Alors G_1 est une courbe dans C_1 joignant le point $\{0, v, 1 - v\}$ de L et le sommet opposé à L dans C_1 . En différentiant on obtient donc une tangente à C_1 différente de τ_0 :

$$\tau_1(v) = \frac{\partial G_1}{\partial s}(0) = (n+1) \sum_{i=0}^n (R_i - S_i) B_i^n(v) \quad (2.11)$$

(cf. appendice, proposition 5)

(Remarquons que les R_i sont les b_I si $i_1 = 1$)

De mme pour C_2 :

$$\tau_2(v) = (n+1) \sum_{i=0}^n (T_i - S_i) B_i^n(v) \quad (2.12)$$

Pour que C_1, C_2 aient le même plan tangent au point $\{0, v, 1 - v\}$, il suffit qu'il existe $\beta_1(v), \beta_2(v)$ tels que

$$\tau_2(v) = \beta_0(v)\tau_0(v) + \beta_1(v)\tau_1(v) \quad (2.13)$$

On obtient le résultat en prenant (cf appendice, proposition 6):

$$\beta_0(v) = \frac{n+1}{n}(\alpha_2(1-v) - \alpha_3v) \quad \beta_1(v) = \alpha_0 \quad (2.14)$$

Remarque 2

En prenant $i = 0, n$ dans (2.7) on trouve que les deux extrémités T_0, T_n de L sont des combinaison linéaire de $\{S_0, S_1, R_0\}$ et $\{S_{n-1}, S_n, R_n\}$ respectivement:

$$T_0 = \alpha_0 R_0 + \alpha_1 S_0 + \alpha_2 S_1 \quad (2.15)$$

$$T_n = \alpha_0 R_n + \alpha_3 S_{n-1} + \alpha_4 S_n \quad (2.16)$$

c'est à dire que $\{T_0, S_0, S_1, R_0\}$ (resp. $\{T_n, S_{n-1}, S_n, R_n\}$) sont coplanaires et que

$$\frac{\text{aire}(T_0 S_1 S_0)}{\text{aire}(R_0 S_0 S_1)} = \frac{\text{aire}(T_n S_n S_{n-1})}{\text{aire}(R_n S_{n-1} S_n)} \quad (2.17)$$

(Pour le montrer, mettre l'origine en S_0 , multiplier vectoriellement (2.15) par S_1 pour obtenir $\alpha_0 = T_0 \times S_1 / R_0 \times S_1$ et faire de même avec (2.16)).

3. INTERPOLATION C^1 D'UNE SURFACE C^0 TRIANGULAIRE PAR MORCEAUX

3.1 Calcul des normales aux sommets

En CAO-Solid-modeler les surfaces sont des frontières ou des intersections de solides élémentaires simples et les normales aux sommets sont donc connues. En éléments finis

cette information peut avoir été perdue car les programmes de calcul ne sont pas toujours intégrés au système de CAO; dans ce cas on définit des normales approchées n^j au sommet P^j par une interpolation $n(x)$ affine par morceaux sur la triangulation de la surface des normales aux éléments n_T en résolvant le problème suivant:

$$\int_{\Omega} n(x)w_h(x)dx = \int_{\Omega} n_T(x)w_h(x)dx \quad (3.1)$$

pour tout w_h affine par morceau continu.

Si on utilise une formule de condensation de masse (formule de Gauss avec points de quadrature aux sommets des triangles) pour le calcul de l'intégrale de gauche on obtient:

$$n(q^i) = \frac{\sum_{(T:q^i \in T)} n_T \text{area}(T)}{\sum_{(T:q^i \in T)} \text{area}(T)} \quad (3.2)$$

3.2 Calcul de l'interpolation d'ordre 3 des carreaux triangulaires

Puisque les plans tangents de la future surface C^1 sont déterminés par les normales aux sommets, si les frontières des carreaux de Bézier T^1, T^2 sont des courbes de Bézier de degré 3 alors les deux points de contrôle internes sont dans les plans tangents de l'extrémité la plus proche et (par exemple) dans le plan moyen de T^1 et T^2 de normale $n_{T^1} \text{aire}(T^1) + n_{T^2} \text{aire}(T^2)$. Pour vérifier (2.17) on fixe leurs positions de sorte que (figure 8):

$$\frac{\text{aire}(PQ_iQ_{i+1})}{\text{aire}(PP_iP_{i+1})} = \text{constante} \quad (3.3)$$

Le point de contrôle central $b_{(1,1,1)}$ est fixé arbitrairement:

$$b_{(1,1,1)} = \frac{1}{4}(b_{(1,2,0)} + b_{(2,1,0)} + b_{(0,1,2)} + b_{(0,2,1)} + b_{(1,0,2)} + b_{(2,0,1)}) - \frac{1}{6}(b_{(3,0,0)} + b_{(0,3,0)} + b_{(0,0,3)}) \quad (3.4)$$

car cette formule assure une précision quadratique (voir [1]) d'interpolation d'une surface régulière par des carreaux de Bézier de degré 3 lorsque les b_I sont sur celle-ci sauf celui du centre.

3.3 Reinterpolation des carreaux d'ordre 3 à l'ordre 4

Etape 1.

Les carreaux d'ordre 3 ainsi construits ne sont toujours pas C^1 aux interfaces. Pour obtenir un raccord C^1 on divise chaque carreau triangulaire en 3 sous-carreaux d'ordre 3 par leur points centraux $b_{(1,1,1)}$ et on construit les points de

contrôle des 3 sous-carreaux en temps que carreaux de Bézier d'ordre 3 (figure 8)

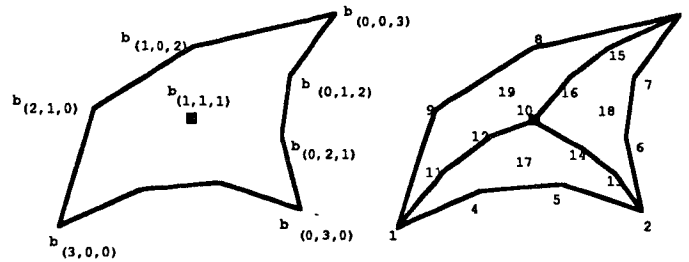


Figure 8

Etape 2.

Ensuite on construit aussi les points de contrôle de ces mêmes sous-triangles curvilignes mais vus comme des carreaux de Bézier d'ordre 4 (figure 9).

Etape 3.

Maintenant pour chaque frontière des grands triangles on modifie les points de contrôles internes voisins de manière à satisfaire aux conditions de la proposition 2. (figure 6)

Etape 4.

Enfin l'étape précédente ayant détruit le raccord des sous-triangles on rectifie les points de contrôle des interfaces internes sauf les sommets des triangles et leurs points de contrôle voisins (donc les points 15,18,21,16,19,22,13 de la figure 9) en les mettant à la moyenne des points voisins.

La surface finale est C^1 .

Analyse de l'étape 1.

Conformément à la figure 9, notons b_1, b_2, \dots, b_{10} les points de contrôles du grand triangle ($b_1 = b_{(3,0,0)}, b_2 = b_{(2,1,0)}, b_3 = b_{(1,2,0)}, \dots, b_{10} = b_{(1,1,1)}$). Dans le même ordre, les points de contrôle du premier sous-triangle sont (cf. [1])

$$\begin{aligned}
 & b_1, \quad b_2, \quad b_{10}, \quad b_4, \quad b_5, \\
 & b_{13} = \frac{1}{3}(b_{11} + b_{17} + b_{19}), \quad b_{14} = \frac{1}{3}(b_{13} + b_{18} + b_{17}) \quad b_{12} = \frac{1}{3}(b_{11} + b_{17} + b_{19}) \\
 & b_{11} = \frac{1}{3}(b_1 + b_4 + b_9) \quad b_{17} = \frac{1}{3}(b_{10} + b_4 + b_5) \tag{3.5}
 \end{aligned}$$

Analyse de l'étape 2.

La construction des points de contrôle d'ordre 4 d'un triangle d'ordre 3 donné par $b_1, b_2 \dots b_{10}$ se fait conformément aux figures 4 et 9 par:

$$b_{(4,0,0)} = b_1, \quad b_{(0,4,0)} = b_2, \quad b_{(0,0,4)} = b_3$$

$$\begin{aligned}
b_{(3,1,0)} &= \frac{3}{4}b_4 + \frac{1}{4}b_1, & b_{(2,2,0)} &= \frac{1}{2}(b_4 + b_5), & b_{(1,3,0)} &= \frac{3}{4}b_2 + \frac{1}{4}b_5 \\
b_{14} &= b_{(2,1,1)} = \frac{1}{4}(2b_{10} + b_9 + b_4)
\end{aligned} \tag{3.6}$$

Analyse de l'étape 3.

D'après (2.17) on peut calculer $\alpha_0, \alpha_1, \alpha_2$, tels que

$$T_0 = \alpha_0 R_0 + \alpha_1 S_0 + \alpha_2 S_1, \quad \alpha_0 + \alpha_1 + \alpha_2 = 1$$

et aussi α_3, α_4 tels que:

$$T_3 = \alpha_0 R_3 + \alpha_3 S_2 + \alpha_4 S_3, \quad \alpha_0 + \alpha_3 + \alpha_4 = 1$$

Ensuite pour vérifier (2.7), on pose:

$$G_i = \frac{3-i}{3}(\alpha_0 R_i + \alpha_1 S_i + \alpha_2 S_{i+1}) + \frac{i}{3}(\alpha_0 R_i + \alpha_3 S_{i-1} + \alpha_4 S_i) - T_i$$

et on change R_i en $R_i + \alpha_0 G_i / (\alpha_0^2 + 1)$ et T_i en $T_i + G_i / (\alpha_0^2 + 1)$ pour $i=0, \dots, 3$.

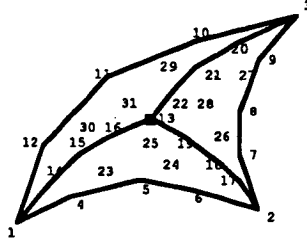


Figure 9:

Analyse de l'étape 4.

Pour chaque triangle T , on redéfinit les points $b_{15}, b_{18}, b_{21}, b_{16}, b_{19}, b_{22}, b_{13}$ dans cet ordre, avec les formules:

$$b_{15} = \frac{1}{3}(b_{14} + b_{23} + b_{30}), \quad b_{16} = \frac{1}{3}(b_{15} + b_{25} + b_{31}), \quad b_{13} = \frac{1}{3}(b_{16} + b_{19} + b_{22}),$$

4. AUTRES APPLICATIONS

4.1 Interpolation C^1 de résultats C^0 bidimensionnels

Etant donné une triangulation d'un domaine plan et les valeurs d'une fonction f aux sommets de la triangulation, on se propose de trouver une interpolation C^1 de cette fonction.

En fait ce problème est identique au précédent car $\{x_1, x_2, f(x_1, x_2)\}$ est une surface de R^3 dont on connaît une triangulation; on est donc ramené au problème précédent.

Notons cependant que pour ce problème particulier il y a une méthode plus simple qui consiste à ajuster seulement la composante verticale des noeuds sans toucher aux composantes seon x_1, x_2 (Farin[1]).

4.2 Interpolation de résultats C^0 sur une surface C^0 .

On cherche une visualisation C^1 régulière par des couleurs ou des niveaux de gris d'une fonction à valeurs sur une surface qui elle même peut être C^0 seulement.

On pourrait s'arrêter là car l'oeil n'est pas très sensible la continuité C^1 des taches de couleur, la continuité suffit. On peut donc se contenter de calculer la couleur de chaque pixel de la surface C^1 par interpolation linéaire des valeurs aux sommets . Mais si on veut effectivement un affichage C^1 alors il faut procéder comme suit.

On commence par trouver par la méthode précédente une interpolation C^1 de la surface. Ensuite comme il existe en théorie une représentation paramétrique locale de la surface avec deux variables s_1, s_2 , alors le problème se réduit à trouver une interpolation C^1 d'une fonction de s_1, s_2 . Mais la programmation de cette méthode n'est pas aisée car on ne connaît pas le système local de coordonnées.

EXEMPLE

L'algorithme a été programmé en FORTRAN sur station Apollo. La figure 10 montre un exemple de reconstitution d'une sphère et d'un tore à partir de données C^0 plus grossière.

Le programme à été incorporé dans la bibliothèque graphique de l'Inria, Vision.

Remerciements

Les auteurs souhaitent remercier M. Bercovier pour leur avoir suggéré ce problème et F. Hecht et A. Marrocco pour l'incorporation dans VISION.

REFERENCES

- 1 G. Farin: Smooth interpolation to scattered 3D data; in *surfaces in CAGD* R.E. Barnhill, W. Boehm eds. North-Holland pp43-63 (1983).
- 2 P.G. Ciarlet: *The finite element method for elliptic problems*. North-Holland(1977).
- 3 S. Gopalsamy: C^1 interpolation of C^0 data for FEM graphics (to appear).
- 4 G. Farin: A construction for visual C^1 continuity of polynomial surface patches. *Computer graphics and image processing*, 20 pp272-282 (1982).
- 5 L. Piegl: A CAGD theme: Geometric continuity and shape parameters. *Computer aided design*, 19, 10, pp 566-567 (1987).

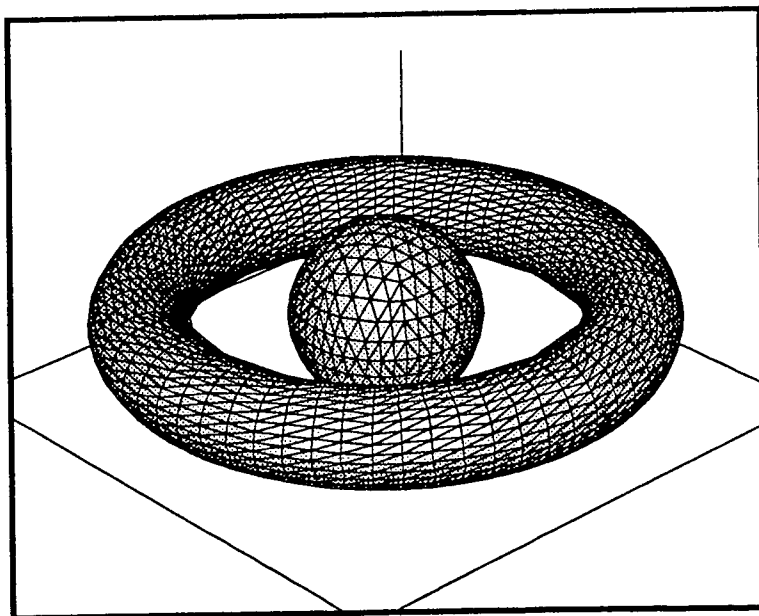
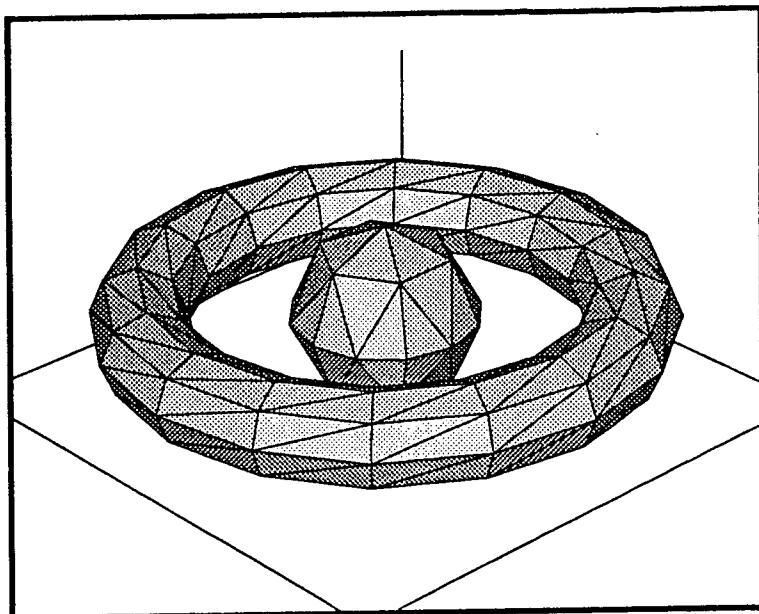


Figure 10 Résultat d'une interpolation C^1 de données C^0 . Sur la figure du haut, les données; sur la figure du bas le résultat de l'interpolation.

APPENDICE

Notation:

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, \quad 0 \leq i \leq n;$$

$$B_I^n(\underline{u}) = \frac{n!}{i_1! i_2! i_3!} u_1^{i_1} u_2^{i_2} u_3^{i_3}, \quad I = (i_1, i_2, i_3), \underline{u} = (u_1, u_2, u_3).$$

1. Passage de la représentation de degré n à celle de degré $n+1$

Proposition 3

Si $p(u) = \sum_0^n b_i B_i^n(u)$ alors $p(u) = \sum_0^{n+1} b_i^* B_i^{n+1}(u)$ avec

$$b_i^* = b_{i-1} \frac{i}{n+1} + b_i \frac{n+1-i}{n+1}$$

Démonstration :

$$\begin{aligned} p(u) &= \sum_{i=0}^n b_i B_i^n(u) \\ &= \sum_{i=0}^n b_i \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \\ &= \sum_{i=0}^n b_i \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} (u+1-u) \\ &= \sum_{i=0}^n b_i \frac{n!}{i!(n-i)!} u^{i+1} (1-u)^{n-i} + \sum_{i=0}^n b_i \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i+1} \\ &= \sum_{i=0}^n b_i \frac{i+1}{n+1} \frac{(n+1)!}{(i+1)!(n-i)!} u^{i+1} (1-u)^{n-i} + \\ &\quad \sum_{i=0}^n b_i \frac{n+1-i}{n+1} \frac{(n+1)!}{i!(n+1-i)!} u^i (1-u)^{n+1-i} \\ &= \sum_{i=1}^{n+1} b_{i-1} \frac{i}{n+1} \frac{(n+1)!}{i!(n+1-i)!} u^i (1-u)^{n+1-i} + \\ &\quad \sum_{i=0}^n b_i \frac{n+1-i}{n+1} \frac{(n+1)!}{i!(n+1-i)!} u^i (1-u)^{n+1-i} \\ &= \sum_{i=0}^{n+1} \left(b_{i-1} \frac{i}{n+1} + b_i \frac{n+1-i}{n+1} \right) B_i^{n+1}(u). \end{aligned}$$

2. Vecteur tangent à un carreau de Bézier

Proposition 4

Soit

$$\Gamma(u) = \sum_{i=0}^n S_i B_i^n(u), 0 \leq u \leq 1, S_i \in \mathbb{R}^3;$$

alors

$$\frac{\partial \Gamma}{\partial u}(u) = n \sum_{i=0}^{n-1} (S_{i+1} - S_{i-1}) B_i^{n-1}(u).$$

Démonstration :

$$\begin{aligned} \frac{\partial \Gamma}{\partial u}(u) &= \sum_{i=0}^n S_i \frac{\partial}{\partial u} \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \\ &= \sum_{i=0}^n S_i \frac{n!}{i!(n-i)!} [i u^{i-1} (1-u)^{n-i} - (n-i) u^i (1-u)^{n-i-1}] \\ &= \sum_{i=1}^n S_i n \frac{(n-1)!}{(i-1)!(n-i)!} u^{i-1} (1-u)^{n-i} - \sum_{i=0}^{n-1} S_i n \frac{(n-1)!}{i!(n-1-i)!} u^i (1-u)^{n-1-i} \\ &= \sum_{i=0}^{n-1} S_{i+1} n \frac{(n-1)!}{i!(n-1-i)!} u^i (1-u)^{n-1-i} - \sum_{i=0}^{n-1} S_i n \frac{(n-1)!}{i!(n-1-i)!} u^i (1-u)^{n-1-i} \\ &= n \sum_{i=0}^{n-1} (S_{i+1} - S_i) B_i^{n-1}(u). \end{aligned}$$

3. Tangente non parallèle à l'arête commune à deux carreaux

Proposition 5

Pour un carreau de Bézier de order $(n+1)$ avec points de control b_I , soit

$$G(s) = \sum_{|I|=n+1} b_{|I|} B_I^{n+1}(u(s)), u(s) = (s, (1-s)u, (1-s)(1-u))$$

alors

$$\frac{\partial G}{\partial s}(0) = (n+1) \left[\sum_{i=0}^n R_i B_i^n(u) - \sum_{i=0}^{n+1} S_i^* B_i^n(u) \right],$$

où $R_i = b_{(1,i,n-i)}$, $S_i^* = b_{(0,i,n+1-i)}$.

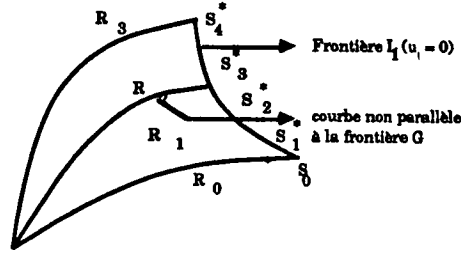


Figure A1: Points de contrôle intervenants dans la définition de la tangente à la courbe non-parallèle à la frontière du carreau.

Démonstration :

$$\begin{aligned}
\frac{\partial G}{\partial s}(0) &= \sum_{\substack{I=(i_1, i_2, i_3) \\ i_1+i_2+i_3=n+1}} b_I \frac{(n+1)!}{i_1!i_2!i_3!} \frac{\partial}{\partial s} [s^{i_1}(1-s)^{i_2}u^{i_2}(1-s)^{i_3}u^{i_3}](s=0) \\
&= \sum_{\substack{I=(i_1, i_2, i_3) \\ i_1+i_2+i_3=n+1}} b_I \frac{(n+1)!}{i_1!i_2!i_3!} [i_1 s^{i_1-1}(1-s)^{n+1-i_1} - \\
&\quad (n+1-i_1)s^{i_1}(1-s)^{n-i_1}] u^{i_2}(1-u)^{i_3} (s=0) \\
&= \sum_{\substack{I=(i_1, i_2, i_3) \\ i_1+i_2+i_3=n+1, i_1=1}} b_I \frac{(n+1)!}{i_2!i_3!} u^{i_2}(1-u)^{i_3} - \\
&\quad \sum_{\substack{I=(i_1, i_2, i_3) \\ i_1+i_2+i_3=n+1, i_1=0}} b_I \frac{(n+1)!}{i_2!i_3!} (n+1)u^{i_2}(1-u)^{i_3} \\
&= \sum_{i=0}^n b_{(1, i, n-i)} (n+1) \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} - \\
&\quad \sum_{i=0}^{n+1} b_{(0, i, n+1-i)} (n+1) \frac{(n+1)!}{i!(n+1-i)!} u^i (1-u)^{n+1-i} \\
&= (n+1) \left[\sum_{i=0}^n R_i B_i^n(u) - \sum_{i=0}^{n+1} S_i^* B_i^{n+1}(u) \right]
\end{aligned}$$

Corollaire:

Si la frontière $u_1 = 0$ est une courbe de Bézier de degré n avec points de contrôle $\{S_i\}_{0 \leq i \leq n}$ c'est-à-dire si

$$\sum_{i=0}^{n+1} S_i^* B_i^{n+1} = \sum_{i=0}^n S_i B_i^n$$

alors

$$\frac{\partial G}{\partial s}(0) = (n+1) \sum_{i=0}^n (R_i - S_i) B_i^n(u).$$

4. Une condition suffisante pour la continuité C^1

Proposition 6

Soit $S_i, R_i, T_i \in \mathbb{R}^3$ et

$$T_i = \frac{n-i}{n}(\alpha_1 S_i + \alpha_2 S_{i+1} + \alpha R_i) + \frac{i}{n}(\alpha_3 S_{i-1} + \alpha_4 S_i + \alpha R_i)$$

dont $\alpha_1, \alpha_2, \alpha_3$ sont constants, $\alpha_1 + \alpha_2 + \alpha = 1, \alpha_3 + \alpha_4 + \alpha = 1$;
pour $0 \leq u \leq 1$, soit

$$t_0(u) = n \sum_{i=0}^{n-1} (S_{i+1} - S_i) B_i^{n-1}(u)$$

$$t_1(u) = (n+1) \sum_{i=0}^n (R_i - S_i) B_i^n(u)$$

$$t_2(u) = (n+1) \sum_{i=0}^n (T_i - S_i) B_i^n(u).$$

alors

$$t_2(u) = \beta_0(u)t_0(u) + \beta_1(u)t_1(u)$$

avec

$$\beta_0(u) = \frac{n+1}{n}[\alpha_2(1-u) - \alpha_3 u], 0 \leq u \leq 1$$

$$\beta_1(u) = \alpha, 0 \leq u \leq 1.$$

Démonstration:

$$\begin{aligned} \beta_0(u)t_0(u) &= (n+1)[\alpha_2(1-u) - \alpha_3 u] \sum_{i=0}^{n-1} (S_{i+1} - S_i) B_i^{n-1}(u) \\ &= (n+1) \sum_{i=0}^{n-1} [-\alpha_3(S_{i+1} - S_i) \frac{(n-1)!}{i!(n-1-i)!} u^{i+1}(1-u)^{n-i-1} + \\ &\quad \alpha_2(S_{i+1} - S_i) \frac{(n-1)!}{i!(n-1-i)!} u^i(1-u)^{n-i}] \end{aligned}$$

$$\begin{aligned}
&= (n+1) \sum_{i=0}^{n-1} \left[-\alpha_3(S_{i+1} - S_i) \frac{i+1}{n} \frac{n!}{(i+1)!(n-i-1)!} u^{i+1}(1-u)^{n-i-1} + \right. \\
&\quad \left. \alpha_2(S_{i+1} - S_i) \frac{(n-1)}{n} \frac{n!}{i!(n-i)!} u^i(1-u)^{n-i} \right] \\
&= (n+1) \sum_{i=0}^n \left[-\alpha_3(S_i - S_{i-1}) \frac{i}{n} \frac{n!}{i!(n-i)!} u^i(1-u)^{n-i} + \right. \\
&\quad \left. \alpha_2(S_{i+1} - S_i) \frac{(n-1)}{n} \frac{n!}{i!(n-i)!} u^i(1-u)^{n-i} \right] \\
&= (n+1) \sum_{i=0}^n \left[-\alpha_3(S_i - S_{i-1}) \frac{i}{n} + \alpha_2(S_{i+1} - S_i) \frac{(n-1)}{n} \right] B_i^n(u).
\end{aligned}$$

$$\beta_1(u)t_1(u) = (n+1) \sum_{i=0}^n \alpha(R_i - S_i) B_i^n(u).$$

$$\beta_0(u)t_0(u) + \beta_1(u)t_1(u) + (n+1) \sum_{i=0}^n S_i B_i^n(u)$$

$$\begin{aligned}
&= (n+1) \sum_{i=0}^n \left[-\alpha_3(S_i - S_{i-1}) \frac{i}{n} + \alpha_2(S_{i+1} - S_i) \frac{(n-1)}{n} + \alpha(R_i - S_i) + S_i \right] B_i^n(u) \\
&= (n+1) \sum_{i=0}^n \left[\frac{n-i}{n} \{ (1 - \alpha_2 - \alpha) S_i + \alpha_2 S_{i+1} + \alpha R_i \} + \right. \\
&\quad \left. \frac{i}{n} \{ (1 - \alpha_3 - \alpha) S_i + \alpha_3 S_{i-1} + \alpha R_i \} \right] B_i^n(u) \\
&= (n+1) \sum_{i=0}^n \left[\frac{n-i}{n} (\alpha_1 S_i + \alpha_2 S_{i+1} + \alpha R_i) + \frac{i}{n} (\alpha_3 S_{i-1} + \alpha_4 S_i + \alpha R_i) \right] B_i^n(u) \\
&= (n+1) \sum_{i=0}^n T_i B_i^n(u), \quad \text{par définition.}
\end{aligned}$$

Alors

$$\beta_0(u)t_0(u) + \beta_1(u)t_1(u) = (n+1) \sum_{i=0}^n (T_i - S_i) B_i^n(u) = t_2(u).$$

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

