Source:

-[wp1] <https://fr.wikipedia.org/wiki/Polyn%C3%B4me_de_Bernstein>

-[sm] Polynomial real root finding in Bernstein Basis par Spencer M.

-[at] Modélisation Géométrique par Contraintes : Solveurs basés sur l’arithmétique des intervalles par Abdou El Karim TAHARI

- [wp2] https://fr.wikipedia.org/wiki/Matrice\_de\_Casteljau

Polynôme de Bernstein

Pour un degré n >= 0, il y a n+1 polynôme de Bernstein définis, sur l’intervalle [0 ;1], par où les sont les coefficients binomiaux.[wp1]  
Les n+1 polynôme de Bernstein forment une base de l’espace vectoriel des polynômes de degré au plus n.

Propriétés

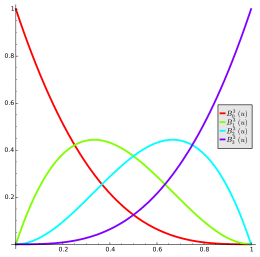
Les polynômes de Bernstein présentent plusieurs propriétés importantes :

Pour tout t appartenant à [0 ;1]

-Partition de l’unité :

-Positivité :

-Symétrie :



Polynôme de Bernstein de degré 3 [wp1]

Puisque les polynômes de Bernstein sont définis sur l’intervalle [0 ;1], or il faut étudier des polynômes sur l’intervalle [a ; b].

On va donc introduire un changement de variable [sm] :

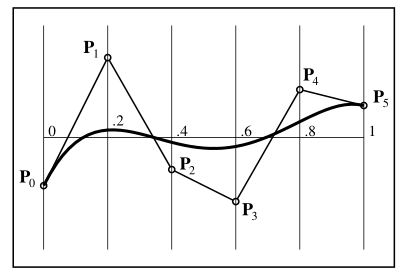
Ce qui permet d’avoir un intervalle [a ; b] d’étude, on obtient la nouvelle formule suivante :

On obtient la forme polynomiale dans la base de Bernstein :

Avec comme coefficient de Bernstein

A partir de cette formule on obtient la courbe de Bézier explicite avec des points de contrôle :

Avec



Un polynôme de Bernstein avec ça courbe de Bézier explicite [sm]

Théorème utilisé :

Théorème des valeurs intermédiaires :

Si la fonction f est continue et strictement monotone sur [a ; b] et si le réel m est compris entre f(a) et f(b), alors l'équation f(x) = m a une seule solution dans [a ; b].

Théorème de Cauchy pour les limites [sm] :

Toutes les racines réelles de f(t) sont contenue dans l’intervalle [-B, B]

B=1+max(abs(ai))

Théorème Modifier de Cauchy pour les limites [sm] :

Toutes les racines réelles de f(t) sont contenue dans l’intervalle [-B, B].

Soit N la valeur absolue de la valeur du coefficient le plus négatif de f(t) alors

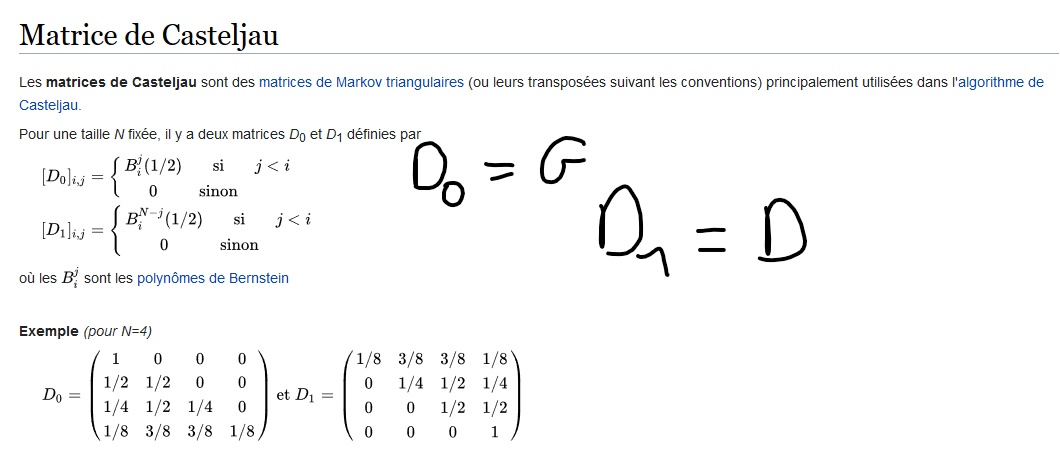
B=1+N

Algorithme de Casteljau :

On utilisera l’algorithme de Casteljau subdiviser un intervalle en deux parties ce qui permet d’augmenter la précision de la position de la racine.

Par exemple sur un intervalle [0,1] on obtiendra deux intervalles [0, 1/2] et [1/2, 0]

On utilisera les matrices de Casteljau [wp2] :



Condition de racine :

Une fois les points contrôlent ou les coefficients de Bernstein obtenues dans un intervalle [a, b], on peut regarder s’il y a un ou plusieurs changements de signe dans les coefficients de Bernstein pour savoir s’il y a une racine dans cet intervalle d’après le théorème des valeurs intermédiaires cela garantie l’existence de la racine dans l’intervalle. S’il n’y a pas de racine dans l’intervalle on peut l’abandonner et répéter le test sur tous les autres intervalles. [sm] et [at] ont décrit cette méthode pour savoir l’existence d’une racine.

Méthode pour calculer la racine :

L’utilisation de la base de Bernstein pour isoler la racine permet de garantir la convergence des algorithmes classique, comme la méthode de newton ou par dichotomie. Puisque nous faisons un solveur en base de Bernstein nous avons décidé d’utiliser les points de contrôle de la courbe de Bézier explicite, pour rester en base de Bernstein pour ne pas utiliser la base canonique du polynôme.

Algorithme Principale :

Changement de base :

On va représenter les polynômes de Bernstein sous forme de matrice, comme on peut représenter les polynômes ,f(x) = a0+a1\*x+a2\*x^2 + …, an\*x^n sous la forme f(X) = XF

Avec X=(1,x,x^2,…x^n) pour variable et F= (a0,a1,a2, … an)^T

On suivra en partie la méthode détailler dans [at]

La conversion de entre la base canonique et la base de Bernstein est une application linéaire, représentable par une matrice M de taille (n+1) x (n+1) telle que B=X.M , M est construite par les coefficients de la base de Bernstein.

La matrice B représente la base de Bernstein B= (B0, n(t), B1, n (t), .., Bn, n(t))

Pour un polynôme de degrés 3 sur l’intervalle [0,1] on à

Pour faire le changement de base, on utilise le fait que B=X.M on obtient :

f(x)=X.F

=(B.M^-1).F

=B.P

B représente les polynômes de Bernstein tandis que, P=(y0,y1,…,yn)^T représente les coefficients de Bernstein, ils peuvent facilement transformer en points de contrôle de la courbe de Bézier explicite avec a=0 et b=1.

Test de l’existence d’une racine :

Maintenant que nous avons les points de contrôle on peut faire le test pour savoir s’il y a une racine dans l’intervalle choisi, ici [0,1], s’il n’y a pas de racine dans ce cas on peut arrêter de calculer, s’il y a une racine et que l’intervalle est trop grand pour calculer précisément la racine on fera l’algorithme de Casteljau.

Utilisation de l’algorithme de Casteljau

Puisque l’on utilise des matrices, on se servira des matrices de Casteljau pour diviser l’intervalle en 2.

Par exemple pour le degré 3 on a la matrice gauche

On fait Pgauche=G.P et Pdroite=D.P

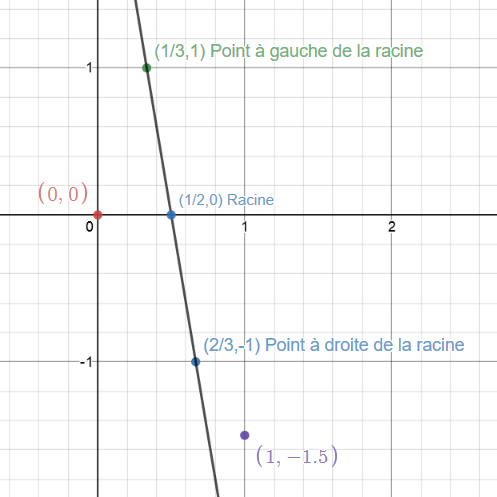
Pgauche sont les points de control pour l’intervalle [0, ½].

Pdroite sont les points de control pour l’intervalle [1/2, 0].

A partir de là on peut refaire le test pour savoir s’il y a une racine et refaire l’algorithme de Casteljau tant que la taille de l’intervalle et supérieur à la précision voulue.

Calcul de la racine :

Une fois que la précision voulue est atteinte on peut calculer approximativement où se trouve la racine dans l’intervalle, il y a plusieurs méthodes pour faire cela par exemple tracé une droite d’équation g=ax+b entre le point à gauche de la racine et à droite et on résout l’équation pour ax+b=0 ce qui donne une approximation de la racine, on peut aussi prendre un point de l’intervalle pour approximer la racine si l’intervalle est suffisamment petit l’erreur sera acceptable.



Méthode en traçant un segment d’équation y=ax+b entre le point à gauche de la racine et à droite de la racine, et résoudre l’équation pour ax+b=0 .

Note sur notre algorithme :

Le calcul de toute les matrices de changement de base et de Casteljau, pour chaque degré n’est pas nécessaire. On peut utiliser la matrice de passage vers la base de Bernstein, en utilisant la matrice pour les polynômes de degré 5 pour des polynômes de degré inférieur. Il suffit de mettre des coefficients du polynômes à 0. Dans notre approche nous avons privilégier la rapidité des calculs. Le nombre d’opération augmente énormément en fonction des matrices utilisé, car il y a des inversions de matrice et des multiplications entre matrice. Donc utiliser des matrices plus grandes que nécessaire augmente énormément le nombre d’opération et le temps de calcul, au bénéfice du temps de programmation. Donc dans notre approche qui vise à être la plus rapide nous utilisons la matrice adaptée au polynôme. Il a donc fallu calculer les matrices de passage pour les polynômes de degré 3,4 et 5.

Conclusion :

Le solveur utilisant les polynômes de Bernstein permet de converger efficacement sûr les racines du polynôme, en évitant des calculs inutiles que pourrait faire d’autre solveur qui isole les racines de manière plus naïve. Le solveur a tout de même des limites à partir du degrés 20 il devient beaucoup lents comparait à d’autres méthodes. [at], [sm]