

Διπλωματική Εργασία
*Πολύωνυμα Bernstein και
 εφαρμογές τους στο CAGD*

*Τμήμα Μηχανικών Η/Υ,
 Τηλεπικοινωνιών & Δικτύων*

Βόλος, Οκτώβριος 2011

Όνομα : Αναγνωστοπούλου Ελένη

Επιβλέποντες : Ακρίτας Αλκιβιάδης

Αγάς Κωνσταντίνος

Διανύουμε μια εποχή με ραγδαίες εξελίξεις στον τομέα των επιστημών και ιδιαίτερα εκείνης των υπολογιστών. Δε θα χαρακτηριζόταν περιττή όμως μια συσχέτιση του παρελθόντος με το παρόν καθώς πολλές εφαρμογές που σήμερα θεωρούνται δεδομένες, και συνεχώς βελτιώνονται, είχαν πρακτικό σημείο έναρξης στα μισά του προηγούμενου αιώνα ενώ το θεωρητικό τους υπόβαθρο είχε αναπτυχθεί κάποιες εκατοντάδες χρόνια νωρίτερα. Πιο συγκεκριμένα, τα πολυώνυμα, που όλοι έχουμε συναντήσει από τα μαθητικά μας χρόνια, συνετέλεσαν στην ποιοτική αναπαράσταση καμπυλών και επιφανειών στην οθόνη των υπολογιστών. Εκμεταλλευόμενοι τις ιδιότητες διαφορετικών βάσεων πολυωνύμων οδηγήμασε σε διαφορετικές αναπαραστάσεις, πιο λεπτομερείς είτε πιο εύκολες στην ανάπτυξη είτε πιο “φθηνές” υπολογιστικά.

Η γραφική αναπαράσταση των καμπυλών και των επιφανειών εν γένει αποτελεί θέμα μεγάλης σημασίας σε πολλούς τομείς ερευνών και εφαρμογών, κυρίως λόγω της ραγδαίας εξέλιξης της σχεδίασης χρήσει υπολογιστή (CAD) και της τεχνολογίας των παιχνιδιών. Κρίθηκε σκόπιμο στην παρούσα διπλωματική εργασία να εξετασθούν τα Bernstein πολυώνυμα καθώς και η συνεισφορά τους στον τομέα του Computer Aided Geometric Design, αφού αποτελούν την βάση για την μια απο τις δυο δημοφιλέστερες μεθόδους αναπαράστασης στον χώρο του CAGD, τις Bézier καμπύλες.

Κανείς μας δεν μπορεί να υπερηφανευθεί για τα κατορθώματά του χωρίς να αναγνωρίσει πως εν τέλει δεν έφτασε στο στόχο του μόνος του. Πάντα υπάρχουν άνθρωποι που μας στηρίζουν, μας καθοδηγούν, μας συμβουλεύουν και μας δίνουν την ώθηση να κάνουμε ό,τι καλύτερο μπορούμε. Η εν λόγω διπλωματική εργασία ολοκληρώθηκε με τη βοήθεια πολλών ανθρώπων, ο καθένας από τους οποίους συνεισέφερε με τον δικό του τρόπο. Κάνοντας την αρχή από τους διδάσκοντες του πανεπιστημίου και ιδιαίτερος από τον καθηγητή μου, κ. Ακρίτα, ο οποίος με στήριξε και μου έδωσε κίνητρα καθ' όλη τη διάρκεια των σπουδών μου όπως και κατά τη διάρκεια σύνταξης της διπλωματικής μου εργασίας, συνεχίζοντας με τους φίλους μου που με ενθάρρυναν στις δύσκολες στιγμές, θα ήθελα τέλος να ευχαριστήσω και την οικογένειά μου για την άνευ όρων αγάπη και στήριξή της.

Στον κ. Ακρίτα Αλκιβιάδη

&

Στην οικογένεια μου

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή.....	5
2. Θεωρία των Bernstein πολυωνύμων	6
2.1 Ορισμός των Bernstein πολυωνύμων.....	6
2.2 Ιδιότητες των Bernstein πολυωνύμων.....	9
2.2.1 Αναδρομικός ορισμός των Bernstein πολυωνύμων.....	9
2.2.2 Τα πολυώνυμα Bernstein είναι μη αρνητικά (All Non-Negative).....	9
2.2.3 Τα πολυώνυμα Bernstein αθροίζουν στη μονάδα (Form Partition of unity).....	10
2.2.4 Degree Raising.....	11
2.3 Χαρακτηριστικά των πολυωνύμων.....	13
2.3.1 Παράγωγοι.....	13
2.3.2 Τα Bernstein πολυώνυμα σαν βάση χώρου	14
2.3.3 Αναπαράσταση των Bernstein πολυωνύμων με πίνακες	15
3. Base switches μεταξύ Power Basis και Bernstein πολυωνύμων	16
3.1 Αναπαράσταση των Bernstein πολυωνύμων σε όρους Power Basis	16
3.2 Αναπαράσταση των Power Basis πολυωνύμων σε όρους Bernstein.....	17
3.3 Ανάπτυξη διαδικασίας για εναλλαγή μεταξύ Bernstein και Power Basis βάσεων πολυωνύμων	17
3.3.1 Θεωρητική προσέγγιση	17
3.3.2 Η διαδικασία στο Mathematica.....	21
4. Bernstein πολυώνυμα και η συμβολή τους στο χώρο του CAGD.....	23
4.1 Εισαγωγή.....	23
4.2 Αρχικοί Σχεδιασμοί.....	23
4.3 Οι De Casteljaou και Bézier	25
4.4 Bézier curves Technology	27
4.4.1 Bézier curves and surfaces.....	28
4.4.2 Ορισμός καμπυλών.....	31
4.5 Τμήματα καμπυλών (curve segments).....	32
4.6 Bézier υποδιαίρεση (Bézier subdivision)	33
4.7 Rational Bézier curves	33
4.8 Bézier επιφάνειες (Bézier surfaces).....	34
4.9 Αλγόριθμος de Casteljaou.....	35

4.10	Εξετάζοντας θέματα πραγματικού χρόνου	37
4.10.1	<i>Υπολογισμοί</i>	37
4.10.2	<i>Sculpting</i>	37
4.11	Εφαρμογές	38
4.11.1	<i>Collision Detection</i>	38
4.11.2	<i>Soft objects</i>	40
4.11.3	<i>Αυτόματη επιλογή ανάλυσης (LOD)</i>	41
4.11.4	<i>Μοντελοποίηση αντικειμένων</i>	42
4.12	Υλοποίηση	42
4.12.1	<i>Απλός υπολογισμός</i>	42
4.12.2	<i>Forward Difference (Προς τα εμπρός διαφορές)</i>	43
4.12.3	<i>Αναδρομική υποδιαίρεση (recursive subdivision)</i>	44
4.12.4	<i>Βοηθητικοί Πίνακες (look-up tables)</i>	44
4.12.5	<i>Φορητότητα (portability)</i>	45
4.13	Συγκρίσεις	46
5.	Αναπαραστάσεις χρήσει του υπολογιστικού προγράμματος Mathematica	48
	Βιβλιογραφία	53

1. Εισαγωγή

Στην επιστήμη των μαθηματικών ένα αδιαμφισβήτητο χρήσιμο εργαλείο είναι τα πολυώνυμα, και όχι άδικα καθώς παρουσιάζουν πολλά πλεονεκτήματα :

- Ορίζονται εύκολα
- Η αριθμητική τους τιμή υπολογίζεται γρήγορα στα συστήματα υπολογιστών
- Αναπαριστούν μεγάλη ποικιλία συναρτήσεων
- Διαφορίζονται και ολοκληρώνονται εύκολα
- Με συνδυασμό τους σχηματίζονται *splines* και προσεγγίζεται οποιαδήποτε συνάρτηση σε οποιαδήποτε επιθυμητή ακρίβεια

Η πιο κοινή μορφή αναπαράστασης τους είναι η *power basis*, όπου κάθε πολυώνυμο αναπαρίσταται σαν γραμμικός συνδυασμός των ακόλουθων μονωνύμων μιας μεταβλητής $\{1, t, t^2, \dots, t^n\}$, με την εξής μορφή

$$p(t) = a_n t^n + a_{n-1} t^{n-1} + \dots + a_1 t + a_0$$

Κατά τον ίδιο τρόπο μπορεί να γραφεί και οποιαδήποτε πολυωνυμική συνάρτηση βαθμού μικρότερου είτε ίσου με n καθώς

- Το σετ των μονωνύμων βαθμού μικρότερου είτε ίσου με n σχηματίζει ένα διανυσματικό χώρο όπου τα μονώνυμα προστίθενται μεταξύ τους, πολλαπλασιάζονται με συντελεστές και ισχύουν όλες οι ιδιότητες που αφορούν τους διανυσματικούς χώρους
- Το σετ των μονωνύμων $\{1, t, t^2, \dots, t^n\}$ αποτελεί και βάση για τον διανυσματικό χώρο, οπότε οποιοδήποτε πολυώνυμο βαθμού μικρότερου είτε ίσου με n μπορεί να γραφεί μοναδικά σαν γραμμικός συνδυασμός τους

Αυτή η βάση ονομάζεται *power basis* και δεν είναι παρά μόνο μια από τις δυνατές βάσεις αναπαράστασης στον χώρο των πολυωνύμων. Στις ακόλουθες ενότητες θα αναφερθούμε στην αναπαράσταση των *Bernstein πολυωνύμων*, στην συσχέτιση τους με την προαναφερθείσα αναπαράσταση και θα εξετάσουμε σε ποιο βαθμό είναι εφικτή μια εναλλαγή μεταξύ των δυο βάσεων.

Στη συνέχεια εστιάζουμε στην ιστορία του CAGD και στις μεθόδους αναπαράστασης καμπυλών στους υπολογιστές. Στο τελευταίο τμήμα παρατίθεται ο συσχετισμός των ανωτέρω με το υπολογιστικό πρόγραμμα *Mathematica*.

2. Θεωρία των Bernstein πολυωνύμων

2.1 Ορισμός των Bernstein πολυωνύμων

Τα πολυώνυμα *Bernstein* βαθμού n ορίζονται σύμφωνα με την ακόλουθη σχέση

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \text{ για } i = 0, 1, \dots, n \text{ και } t \in [0, 1]$$

$$\text{ενώ } \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Υπάρχουν $n+1$ Bernstein πολυώνυμα βαθμού n , ενώ για μαθηματική ευκολία συνήθως θέτουμε $B_{i,n} = 0$, εάν $i < 0$ ή $i > n$.

Επίσης τα πολυώνυμα Bernstein είναι εύκολο να γραφούν καθώς ισχύουν τα ακόλουθα : οι συντελεστές $\binom{n}{i}$ προκύπτουν από το τρίγωνο του Pascal, ο εκθέτης των όρων t αυξάνεται καθώς αυξάνεται το i , ενώ ο εκθέτης των όρων $(1-t)$ μειώνεται όσο αυξάνεται το i .

Στη συνέχεια παρατίθενται οι πιο απλές περιπτώσεις των Bernstein πολυωνύμων ανάλογα με τον βαθμό που επιλέγουμε να παρουσιάζονται.

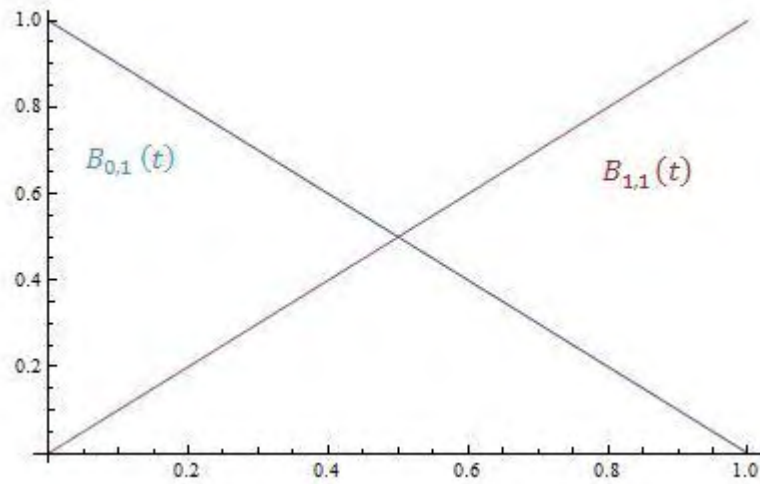
1^ο Βαθμού

Τα πολυώνυμα που συμμετέχουν στον επιλεγμένο βαθμό είναι τα ακόλουθα

$$B_{0,1}(t) = 1 - t$$

$$B_{1,1}(t) = t$$

Η γραφική τους αναπαράσταση για $0 \leq t \leq 1$ παρουσιάζεται στην (Εικόνα 2.1)



Εικόνα 2.1 Γραφική αναπαράσταση πολυωνύμων Bernstein 1^{ου} βαθμού

2^{ου} Βαθμού

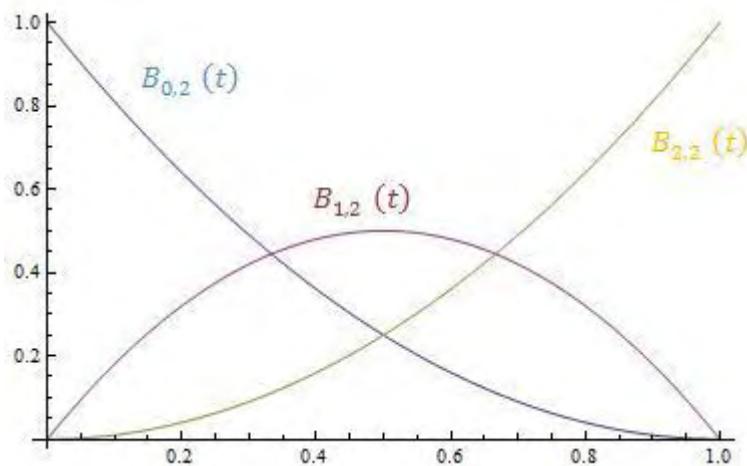
Τα πολυώνυμα που συμμετέχουν στον επιλεγμένο βαθμό είναι τα ακόλουθα

$$B_{0,2}(t) = (1 - t)^2$$

$$B_{1,2}(t) = 2t(1 - t)$$

$$B_{2,2}(t) = t^2$$

Η γραφική τους αναπαράσταση για $0 \leq t \leq 1$ παρουσιάζεται στην (Εικόνα 2.2)



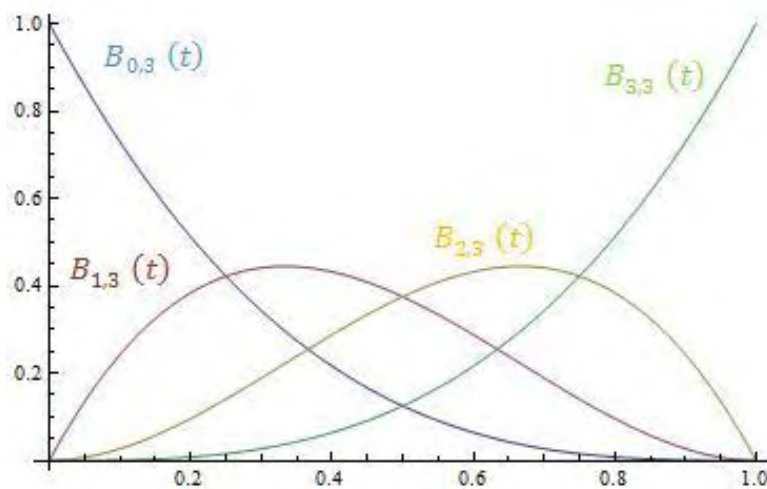
Εικόνα 2.2 Γραφική αναπαράσταση πολυωνύμων Bernstein 2^{ου} βαθμού

3^ο Βαθμού

Τα πολυώνυμα που συμμετέχουν στον επιλεγμένο βαθμό είναι τα ακόλουθα

$$\begin{aligned} B_{0,3}(t) &= (1-t)^3 \\ B_{1,3}(t) &= 3t(1-t)^2 \\ B_{2,3}(t) &= 3t^2(1-t) \\ B_{3,3}(t) &= t^3 \end{aligned}$$

Η γραφική τους αναπαράσταση για $0 \leq t \leq 1$ παρουσιάζεται στην (Εικόνα 2.3)



Εικόνα 2.3 Γραφική αναπαράσταση πολυωνύμων Bernstein 3^{ου} βαθμού

Το πιο σύνηθες διάστημα εντός του οποίου εξετάζουμε τα Bernstein πολυώνυμα είναι το $[0,1]$. Αξίζει όμως να αναφέρουμε και τη μορφή των πολυωνύμων σε επιλεγμένο διάστημα αναφοράς $t \in [a,b]$ και ορίζεται από την ακόλουθη μαθηματική σχέση.

$$B_{i,n}(t) = \binom{n}{i} \frac{(t-a)^i (b-t)^{n-i}}{(b-a)^n}, \text{ για } i = 0, 1, \dots, n$$

2.2 Ιδιότητες των Bernstein πολυωνύμων

Οι ιδιότητες που παρουσιάζουν τα Bernstein πολυώνυμα είναι οι ακόλουθες :

2.2.1 Αναδρομικός ορισμός των Bernstein πολυωνύμων

Τα Bernstein πολυώνυμα βαθμού n μπορούν να γραφούν σαν συνδυασμός δυο Bernstein πολυωνύμων βαθμού $n-1$. Δηλαδή το k -οστό n -οστού βαθμού πολυώνυμο Bernstein γράφεται ως εξής

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

Απόδειξη :

$$\begin{aligned} (1-t)B_{k,n-1}(t) + t B_{k-1,n-1}(t) &= (1-t) \binom{n-1}{k} t^k (1-t)^{n-1-k} + \\ &\quad t \binom{n-1}{k-1} t^{k-1} (1-t)^{n-1-(k-1)} \\ &= \binom{n-1}{k} t^k (1-t)^{n-k} + \binom{n-1}{k-1} t^k (1-t)^{n-k} \\ &= \left[\binom{n-1}{k} + \binom{n-1}{k-1} \right] t^k (1-t)^{n-k} \\ &= \binom{n}{k} t^k (1-t)^{n-k} \\ &= B_{k,n}(t) \end{aligned}$$

2.2.2 Τα πολυώνυμα Bernstein είναι μη αρνητικά (All Non-Negative)

Μια συνάρτηση $f(t)$ είναι non-negative σε διάστημα $[a,b]$ εάν ισχύει $f(t) \geq 0$ για $t \in [a,b]$. Στην περίπτωση των Bernstein πολυωνύμων βαθμού n διακρίνουμε πως είναι non-negative εντός του διαστήματος $[0,1]$.

Απόδειξη:

Είναι προφανές πως οι συναρτήσεις $B_{0,1}(t)=1-t$ και $B_{1,1}(t)=t$ είναι non-negative στο διάστημα $0 \leq t \leq 1$. Αν υποθέσουμε πως όλα τα πολυώνυμα Bernstein βαθμού μικρότερου από k είναι non-negative, τότε χρησιμοποιώντας την αναδρομική ιδιότητα των Bernstein πολυωνύμων μπορούμε να γράψουμε

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

και να υποστηρίξουμε πως τα $B_{k,n}(t)$ είναι επίσης non-negative στο διάστημα $[0,1]$ δεδομένου ότι και τα δυο μέρη του δεξιού μέλους της άνωθεν ισότητας είναι non-negative στο $[0,1]$. Με επαγωγή, όλα τα Bernstein πολυώνυμα είναι non-negative στο $[0,1]$. Στη διάρκεια της απόδειξης επίσης έγινε εμφανές πως κάθε ένα από τα Bernstein πολυώνυμα είναι θετικό όταν $0 \leq t \leq 1$.

2.2.3 Τα πολυώνυμα Bernstein αθροίζουν στη μονάδα (Form Partition of unity)

Ένα σει συναρτήσεων $f_i(t)$ σχηματίζουν partition of unity εάν το άθροισμά τους για κάθε τιμή του t είναι μονάδα. Τα $k+1$ Bernstein πολυώνυμα βαθμού k αθροίζουν στη μονάδα.

Απόδειξη :

Για να αποδείξουμε την ισχύ της προαναφερθείσας πρότασης θα ξεκινήσουμε αποδεικνύοντας ένα λίγο διαφορετικό γεγονός

Για κάθε k , το άθροισμα των $k+1$ Bernstein πολυωνύμων βαθμού k ισούται με το άθροισμα των k Bernstein πολυωνύμων βαθμού $k-1$. Δηλαδή

$$\sum_{i=0}^k B_{i,k}(t) = \sum_{i=0}^{k-1} B_{i,k-1}(t)$$

Ο υπολογισμός μπορεί να γίνει με ευθεία απόδειξη κάνοντας χρήση της αναδρομικής ιδιότητας των πολυωνύμων και αναδιάταξη των αθροισμάτων, έχοντας βέβαια υποθέσει πως $B_{k,k-1}(t) = B_{-1,k-1}(t) = 0$

$$\begin{aligned} \sum_{i=0}^k B_{i,k}(t) &= \sum_{i=0}^k [(1-t)B_{i,k-1}(t) + t B_{i-1,k-1}(t)] \\ &= (1-t) \left[\sum_{i=0}^{k-1} B_{i,k-1}(t) + B_{k,k-1}(t) \right] \\ &\quad + t \left[\sum_{i=1}^k B_{i-1,k-1}(t) + B_{-1,k-1}(t) \right] \\ &= (1-t) \sum_{i=0}^{k-1} B_{i,k-1}(t) + t \sum_{i=1}^k B_{i-1,k-1}(t) \end{aligned}$$

$$\begin{aligned}
&= (1-t) \sum_{i=0}^{k-1} B_{i,k-1}(t) + t \sum_{i=0}^k B_{i,k-1}(t) \\
&= \sum_{i=0}^k B_{i,k-1}(t)
\end{aligned}$$

Οπότε μετά την απόδειξη της πιο πάνω ισότητας, είναι απλό να γράψουμε πως

$$\begin{aligned}
\sum_{i=0}^n B_{i,n}(t) &= \sum_{i=0}^{kn-1} B_{i,n-1}(t) = \sum_{i=0}^{n-2} B_{i,n-2}(t) = \dots = \sum_{i=0}^1 B_{i,1}(t) = (1-t) + t \\
&= 1
\end{aligned}$$

Η προαναφερθείσα ιδιότητα είναι πολύ σημαντική όταν χρησιμοποιούνται τα πολυώνυμα Bernstein σε γεωμετρικούς σχεδιασμούς (geometric modeling) και σε γραφικά υπολογιστών (computer graphics). Συγκεκριμένα, για οποιοδήποτε σετ σημείων P_0, P_1, \dots, P_n του τρισδιάστατου χώρου και για κάθε t , η έκφραση

$$P(t) = P_0 B_{0,n}(t) + P_1 B_{1,n}(t) + \dots + P_n B_{n,n}(t)$$

αποτελεί έναν “καθαρό” συνδυασμό των σημείων P_0, P_1, \dots, P_n και στην περίπτωση που ισχύει $0 \leq t \leq 1$ αυτός ο συνδυασμός είναι κυρτός (convex).

2.2.4 Degree Raising

Κάθε Bernstein πολυώνυμο μικρότερου βαθμού ($<n$) μπορεί να εκφραστεί σαν γραμμικός συνδυασμός Bernstein πολυωνύμων βαθμού n . Συγκεκριμένα κάθε Bernstein πολυώνυμο βαθμού $n-1$ μπορεί να γραφεί σαν γραμμικός συνδυασμός Bernstein πολυωνύμων βαθμού n .

Απόδειξη :

Αρχικά έχουμε

$$\begin{aligned}
t B_{i,n}(t) &= \binom{n}{i} t^{i+1} (1-t)^{n-i} \\
&= \binom{n}{i} t^{i+1} (1-t)^{(n+1)-(i+1)} \\
&= \frac{\binom{n}{i}}{\binom{n+1}{i+1}} B_{i+1,n+1}(t) \\
&= \frac{i+1}{n+1} B_{i+1,n+1}(t)
\end{aligned}$$

και

$$\begin{aligned}(1-t) B_{i,n}(t) &= \binom{n}{i} t^i (1-t)^{n+1-i} \\ &= \frac{\binom{n}{i}}{\binom{n+1}{i}} B_{i,n+1}(t) \\ &= \frac{n-i+1}{n+1} B_{i,n+1}(t)\end{aligned}$$

Οπότε τελικά είναι

$$\begin{aligned}\frac{1}{\binom{n}{i}} B_{i,n}(t) + \frac{1}{\binom{n}{i+1}} B_{i+1,n}(t) &= t^i (1-t)^{n-i} + t^{i+1} (1-t)^{n-(i+1)} \\ &= t^i (1-t)^{n-i-1} ((1-t) + t) \\ &= t^i (1-t)^{n-i-1} \\ &= \frac{1}{\binom{n-1}{i}} B_{i,n-1}(t)\end{aligned}$$

Χρησιμοποιώντας την τελευταία ισότητα, μπορούμε πλέον να γράψουμε οποιοδήποτε Bernstein πολυώνυμο με όρους Bernstein πολυωνύμων μεγαλύτερου βαθμού.

Δηλαδή,

$$\begin{aligned}B_{i,n-1}(t) &= \binom{n-1}{i} \left[\frac{1}{\binom{n}{i}} B_{i,n}(t) + \frac{1}{\binom{n}{i+1}} B_{i+1,n}(t) \right] \\ &= \left(\frac{n-i}{n} \right) B_{i,n}(t) + \left(\frac{i+1}{n} \right) B_{i+1,n}(t)\end{aligned}$$

το οποίο εκφράζει ένα Bernstein πολυώνυμο βαθμού $n-1$ σαν γραμμικό συνδυασμό Bernstein πολυωνύμων βαθμού n .

Επεκτείνοντας αυτή την σχέση μπορούμε να δούμε πως οποιοδήποτε Bernstein πολυώνυμο βαθμού k ($<n$) μπορεί να γραφεί σαν γραμμικός συνδυασμός Bernstein πολυωνύμων βαθμού $n-e$ και για παράδειγμα ένα Bernstein πολυώνυμο βαθμού $n-2$ μπορεί να γραφεί σαν γραμμικός συνδυασμός δυο Bernstein πολυωνύμων βαθμού $n-1$, κάθε ένα από τα οποία με τη σειρά του γράφεται σαν γραμμικός συνδυασμός δυο Bernstein πολυωνύμων βαθμού n και ούτω καθ' εξής.

2.3 Χαρακτηριστικά των πολυωνύμων

2.3.1 Παράγωγοι

Η παράγωγος ενός *Bernstein* πολυωνύμου βαθμού n είναι γραμμικός συνδυασμός *Bernstein* πολυωνύμων βαθμού $n-1$. Συγκεκριμένα είναι

$$\frac{d}{dt} B_{k,n}(t) = n \left(B_{k-1,n-1}(t) - B_{k,n-1}(t) \right), \text{ για } 0 \leq k \leq n$$

Απόδειξη :

Με απευθείας παραγωγή έχουμε

$$\begin{aligned} \frac{d}{dt} B_{k,n}(t) &= \frac{d}{dt} \binom{n}{k} t^k (1-t)^{n-k} \\ &= \frac{kn!}{k!(n-k)!} t^{k-1} (1-t)^{n-k} + \frac{(n-k)!n!}{k!(n-k)!} t^k (1-t)^{n-k-1} \\ &= \frac{n(n-1)!}{(k-1)!(n-k)!} t^{k-1} (1-t)^{n-k} \\ &\quad + \frac{n(n-1)!}{k!(n-k-1)!} t^k (1-t)^{n-k-1} \\ &= n \left(\frac{(n-1)!}{(k-1)!(n-k)!} t^{k-1} (1-t)^{n-k} \right. \\ &\quad \left. + \frac{(n-1)!}{k!(n-k-1)!} t^k (1-t)^{n-k-1} \right) \\ &= n \left(B_{k-1,n-1}(t) - B_{k,n-1}(t) \right) \end{aligned}$$

Επομένως διαπιστώνουμε πως η παράγωγος του πολυωνύμου ισούται με τον βαθμό του n πολλαπλασιασμένο με τη διαφορά των δυο πολυωνύμων *Bernstein* βαθμού $n-1$.

2.3.2 Τα Bernstein πολυώνυμα σαν βάση χώρου

Τα Bernstein πολυώνυμα βαθμού n σχηματίζουν βάση για τον χώρο των πολυωνύμων βαθμού μικρότερου είτε ίσου με n καθώς

Καλύπτουν τον χώρο των πολυωνύμων :

Κάθε πολυώνυμο βαθμού μικρότερου είτε ίσου με n μπορεί να γραφεί σαν γραμμικός συνδυασμός των Bernstein πολυωνύμων. Αυτό προκύπτει εύκολα δεδομένου ότι η βάση της ύψωσης σε δύναμη (power basis) καλύπτει τον χώρο όλων των πολυωνύμων και επιπλέον κάθε μέλος αυτής της βάσης μπορεί να γραφεί σαν γραμμικός συνδυασμός των Bernstein πολυωνύμων.

Είναι γραμμικά ανεξάρτητα :

δηλαδή, στην περίπτωση που υπάρχουν σταθερές c_0, c_1, \dots, c_n ώστε η ακόλουθη ταυτότητα

$$0 = c_0 B_{0,n}(t) + c_1 B_{1,n}(t) + \dots + c_n B_{n,n}(t)$$

να ισχύει για κάθε t , τότε όλα τα c_i πρέπει να είναι μηδέν.

Στην περίπτωση που κάτι τέτοιο ήταν αληθές θα μπορούσαμε να γράψουμε

$$\begin{aligned} 0 &= c_0 B_{0,n}(t) + c_1 B_{1,n}(t) + \dots + c_n B_{n,n}(t) \\ &= c_0 \sum_{i=0}^n (-1)^i \binom{n}{i} \binom{i}{0} t^i + c_1 \sum_{i=1}^n (-1)^{i-1} \binom{n}{i} \binom{i}{1} t^i + \dots \\ &\quad + c_n \sum_{i=n}^n (-1)^{i-n} \binom{n}{i} \binom{i}{n} t^i \\ &= c_0 \left[\sum_{i=0}^1 c_1 \binom{n}{1} \binom{1}{1} \right] t^1 + \dots + \left[\sum_{i=0}^n c_i \binom{n}{n} \binom{n}{n} \right] t^n \end{aligned}$$

Καθώς η βάση ύψωσης σε δύναμη αποτελεί γραμμικώς ανεξάρτητο σετ, πρέπει να ισχύει

$$c_0 = 0$$

$$\sum_{i=0}^1 c_i \binom{n}{1} \binom{1}{1} = 0$$

⋮

$$\sum_{i=0}^n c_i \binom{n}{i} \binom{n}{n-i} = 0$$

το οποίο συνεπάγεται πως $c_0=c_1=\dots=c_n=0$, αφού το c_0 έχει μηδενική τιμή και στη συνέχεια αντικαθιστώντας το στην δεύτερη ισότητα προκύπτει πως επίσης $c_1=0$ και ούτω καθ' εξής.

2.3.3 Αναπαράσταση των Bernstein πολυωνύμων με πίνακες

Σε κάποιες εφαρμογές είναι χρήσιμη η αναπαράσταση των Bernstein πολυωνύμων με την χρήση πινάκων. Δεδομένου πολυωνύμου γραμμένου σαν γραμμικός συνδυασμός των συναρτήσεων βάσης Bernstein

$$B(t) = c_0 B_{0,n}(t) + c_1 B_{1,n}(t) + \dots + c_n B_{n,n}(t)$$

είναι εύκολο να γραφεί σαν γινόμενο δυο διανυσμάτων

$$B(t) = [B_{0,n}(t) \quad B_{1,n}(t) \quad \dots \quad B_{n,n}(t)] \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Κάτι που μπορεί να μετασχηματιστεί ως εξής

$$B(t) = [1 \quad t \quad t^2 \quad \dots \quad t^n] \begin{bmatrix} b_{0,0} & 0 & 0 & \dots & 0 \\ b_{1,0} & b_{1,1} & 0 & \dots & 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n,0} & b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

όπου τα $b_{i,j}$ είναι οι συντελεστές της βάσης power που χρησιμοποιήθηκαν για τον καθορισμό των σχετικών Bernstein πολυωνύμων. Επίσης φαίνεται πως ο εν λόγω πίνακας είναι κάτω τριγωνικός.

Στην περίπτωση των πολυωνύμων δεύτερου βαθμού ($n=2$), η αναπαράσταση με την χρήση πινάκων είναι

$$B(t) = [1 \quad t \quad t^2] \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$$

και αντίστοιχα στην κυβική περίπτωση ($n=3$) ισχύει

$$B(t) = [1 \quad t \quad t^2 \quad t^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

3. Base switches μεταξύ Power Basis και Bernstein πολυωνύμων

3.1 Αναπαράσταση των Bernstein πολυωνύμων σε όρους Power Basis

Καθώς η βάση της ύψωσης σε δύναμη $\{1, t, t^2, \dots, t^n\}$ σχηματίζει τη βάση για τον χώρο των πολυωνύμων βαθμού μικρότερου είτε ίσου με n , τότε οποιοδήποτε Bernstein πολυώνυμο βαθμού n μπορεί να γραφεί χρησιμοποιώντας όρους της power basis. Αυτή η πρόταση μπορεί να υπολογιστεί απευθείας με τη βοήθεια του ορισμού των Bernstein πολυωνύμων και του δυωνυμικού θεωρήματος όπως φαίνεται ακολούθως:

$$\begin{aligned} B_{k,n}(t) &= \binom{n}{k} t^k (1-t)^{n-k} \\ &= \binom{n}{k} t^k \sum_{i=0}^{n-k} (-1)^i \binom{n-k}{i} t^i \\ &= \sum_{i=0}^{n-k} (-1)^i \binom{n}{k} \binom{n-k}{i} t^i \\ &= \sum_{i=k}^n (-1)^{i-k} \binom{n}{k} \binom{n-k}{i-k} t^i \\ &= \sum_{i=k}^n (-1)^{i-k} \binom{n}{i} \binom{i}{k} t^i \end{aligned}$$

Όπου χρησιμοποιήσαμε το δυωνυμικό θεώρημα για να επεκτείνουμε το $(1-t)^{n-k}$.

3.2 Αναπαράσταση των Power Basis πολυωνύμων σε όρους Bernstein

Σε αντιστοιχία για να αποδείξουμε πως κάθε στοιχείο της βάσης ύψωσης σε δύναμη (power basis) μπορεί να γραφεί σαν γραμμικός συνδυασμός Bernstein πολυωνύμων χρησιμοποιούμε τεχνικές degree raising και επαγωγή:

$$\begin{aligned}t^k &= t(t^{k-1}) \\&= \sum_{i=k-1}^n \frac{\binom{i}{k-1}}{\binom{n}{k-1}} B_{i,n-1}(t) \\&= \sum_{i=k}^n \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} t B_{i-1,n-1}(t) \\&= \sum_{i=k-1}^{n-1} \frac{\binom{i}{k-1}}{\binom{n}{k-1}} \frac{i}{n} B_{i,n}(t) \\&= \sum_{i=k-1}^{n-1} \frac{\binom{i}{k}}{\binom{n}{k}} B_{i,n}(t)\end{aligned}$$

3.3 Ανάπτυξη διαδικασίας για εναλλαγή μεταξύ Bernstein και Power Basis βάσεων πολυωνύμων

3.3.1 Θεωρητική προσέγγιση

Η μετάβαση μεταξύ διαφορετικών βάσεων πολυωνύμων αποτελεί θεμελιώδες πρόβλημα στο χώρο του Computer Aided Geometric Design (CAGD), στον οποίο θα αναφερθούμε στην επόμενη ενότητα. Ας λάβουμε υπόψη μια κοινή περίπτωση μετατροπής μεταξύ των power basis και Bernstein πολυωνύμων. Ακριβώς πιο πριν καταλήξαμε πως εν γένει μια τέτοια μετατροπή είναι εφικτή. Για την εύρεση όμως των ισοδύναμων συντελεστών, μεταξύ των δυο αναπαράστασεων, με ακρίβεια απαιτείται η ανάδειξη μιας διαδικασίας.

Έστω λοιπόν, ένα πολυώνυμο Bernstein

$$B(t) = \sum_{i=0}^n b_i \binom{n}{i} (1-t)^{n-i} t^i$$

καθώς και ένα πολυώνυμο power basis

$$P(t) = \sum_{i=0}^n p_i t^i$$

Το πρόβλημα που καλούμαστε να επιλύσουμε είναι αμφίδρομο . Δεδομένων των συντελεστών b_i ενός Bernstein πολυωνύμου επιθυμούμε να βρούμε τους ισοδύναμους συντελεστές p_i των power basis πολυωνύμων (Bernstein to power basis conversion) είτε δεδομένων των p_i να καταλήξουμε στα ισοδύναμα b_i (power basis to Bernstein conversion).

Μια λύση που θα μπορούσε ταυτόχρονα να χαρακτηριστεί “κομψή” αποτελεί η χρήση σειρών Taylor. Το ανάπτυγμα κατά Taylor του $B(t)$ ορίζεται ως

$$B(t) \equiv B(0) + B'(0)t + \frac{B''(0)t^2}{2} + \dots + \frac{B^{(i)}(0)t^i}{i} + \dots$$

και ένα power basis πολυώνυμο θεωρείται ισοδύναμο του Bernstein πολυωνύμου ($P(t) \equiv B(t)$) άν και μόνο αν

$$\frac{P^{(i)}(0)t^i}{i!} = \frac{B^{(i)}(0)t^i}{i!}, \quad i = 0, \dots, n$$

Όμως για τα power basis πολυώνυμα ισχύει

$$\frac{P^{(i)}(0)t^i}{i!} = p_i$$

επομένως

$$p_i = \frac{B^{(i)}(0)}{i!}, \quad i = 0, \dots, n \quad (\Sigma\chi. 3.1)$$

Οι όροι $B^{(i)}(0)$ είναι εφικτό να υπολογιστούν μέσω ενός πίνακα διαφορών. Ας ακολουθήσουμε όμως την διαδικασία βήμα βήμα. Αρχικά διαπιστώνουμε πως οι συντελεστές των παραγώγων υπολογίζονται ως εξής

Δεδομένου ότι

$$B(0) = \sum_{i=0}^n b_i \binom{n}{i} (1-0)^{n-i} 0^i = b_0$$

Έχουμε

$$B'(0) = n(b_1 - b_0) , \quad B''(0) = n(n-1)(b_2 - 2b_1 + b_0)$$

και εν γένει

$$B^{(i)}(0) = n(n-1) \dots (n-i+1) \sum_{j=0}^i (-1)^{(i-j+1)} \binom{i}{j} b_j$$

Αναπαριστώντας το πιο πάνω αποτέλεσμα χρήσει αναδρομικής ιδιότητας έχουμε

$$b_i^j = b_{i+1}^{j-1} - b_i^{j-1} , \text{ με } b_i^0 \equiv b_i$$

Συνεπώς

$$B^{(i)}(0) = n(n-1) \dots (n-i+1) b_i^0 = \frac{n!}{(n-i)!} b_i^0$$

Από την (Σχ.3.1) έχουμε

$$p_i = \frac{n!}{(n-i)! i!} b_i^0 = \binom{n}{i} b_i^0$$

Με αυτό τον τρόπο το αρχικό πρόβλημα απλουστεύεται σε εκείνο της εύρεσης των τιμών των b_0^i , το οποίο απαιτεί την χρήση μόνο ενός πίνακα διαφορών :

$b_0^0 = b_0 = p_0$	$b_1^0 = b_1$...	$b_n^0 = b_n$
$b_0^1 = b_1^0 - b_0 = \frac{p_1}{\binom{n}{1}}$	$b_1^1 = b_2^0 - b_1^0$...	$b_n^1 = b_n^0 - b_{n-1}^0$
$b_0^2 = b_1^1 - b_0^1 = \frac{p_1}{\binom{n}{2}}$	$b_1^2 = b_2^1 - b_1^1$...	$b_n^2 = b_n^1 - b_{n-1}^1$
...
$b_0^{n-1} = b_1^{n-2} - b_0^{n-2} = \frac{p_{n-1}}{\binom{n}{n-1}}$	$b_1^{n-1} = b_2^{n-2} - b_1^{n-2}$		
$b_0^n = b_1^{n-1} - b_0^{n-1} = p_n$			

Συμπεραίνουμε λοιπόν τα ακόλουθα

Για να μετατρέψουμε ένα πολυώνυμο Bernstein σε μορφή power basis , εισάγουμε τους συντελεστές του στην πρώτη σειρά ενός πίνακα που τα υπόλοιπα στοιχεία του συμπληρώνονται όπως ορίζεται ένας πίνακας διαφορών. Για να λάβουμε τους συντελεστές των power basis πολλαπλασιάζουμε την πρώτη στήλη του ολοκληρωμένου πίνακα με $\binom{n}{i}$.

Αντίστοιχα για να υλοποιήσουμε μια power basis σε Bernstein αναπαράσταση πολυωνύμου, αρχικά διαιρούμε τους συντελεστές με $\binom{n}{i}$, έπειτα συμπληρώνουμε την αριστερή στήλη ενός πίνακα διαφορών με τα εν λόγω αποτελέσματα, συνεχίζουμε με τα υπόλοιπα στοιχεία του πίνακα διαφορών εκτελώντας τον αλγόριθμο ανάποδα και καταλήγουμε με τους Bernstein συντελεστές στην πρώτη σειρά του πίνακα.

Παρουσιάζουμε το ακόλουθο παράδειγμα για να γίνει πιο κατανοητή η πιο πάνω προσέγγιση.

Παράδειγμα :

Έστω Bernstein πολυώνυμο τετάρτου βαθμού ,με συντελεστές $\{1,3,4,6,8,\}$, το οποίο επιθυμούμε να μετατρέψουμε σε power basis αναπαράσταση. Ο πίνακας διαφορών συμπληρώνεται ως εξής

$$\left[\begin{array}{ccc} a & b & c \\ -a + b & -b + c & -c + d \\ -(-a + b) + (-b + c) & -(-b + c) + (-c + d) & \\ -(-(-a + b) + (-b + c) + (-(-b + c) + (-c + d))) & & \end{array} \right] d$$

Επομένως στην εξεταζόμενη περίπτωση έχουμε

$$\begin{array}{cccc} 1 & 3 & 4 & 6 & 8 \\ 2 & 1 & 2 & 2 & \\ -1 & 1 & 0 & & \\ 2 & -1 & & & \\ -3 & & & & \end{array}$$

και οι συντελεστές power basis είναι οι ακόλουθοι

$$\begin{aligned} p_0 &= 1 \binom{4}{0} = 1 & p_3 &= 2 \binom{4}{3} = 8 \\ p_1 &= 2 \binom{4}{1} = 8 & p_4 &= -3 \binom{4}{4} = -3 \\ p_2 &= -1 \binom{4}{2} = -6 \end{aligned}$$

Δηλαδή πολυώνυμο $p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3 + p_4 x^4$

Ποιο συγκεκριμένα αρχικά έχουμε το πολυώνυμο Bernstein 4^{ου} βαθμού

$$\begin{aligned} B(t) &= 1 B_{0,4}(t) + 3 B_{1,4}(t) + 4 B_{2,4}(t) + 6 B_{3,4}(t) + 8 B_{4,4}(t) \\ &= (1 - t)^4 + 3 4t(1 - t)^3 + 4 6t^2(1 - t)^2 + 6 4t^3(1 - t) + 8 t^4 \end{aligned}$$

Εισάγουμε το πιο πάνω πολυώνυμο στο Mathematica και στη συνέχεια ζητάμε να υπολογίσει την αναλυτική μορφή του. Επομένως έχουμε

```
In[1] := B[t_]:= (1-t)^4+12 t (1-t)^3+24 t^2 (1-t)^2+24 t^3 (1-t)+8 t^4
```

```
In[2] := Expand[B[t]]
```

```
Out[2] := 1+8t-6t^2+8t^3-3t^4
```

Αποτέλεσμα που ταιριάζει απόλυτα με τους συντελεστές στους οποίους καταλήξαμε αμέσως νωρίτερα

3.3.2 Η διαδικασία στο Mathematica

Για πολύ καιρό τα Bernstein πολυώνυμα δεν είχαν καταφέρει να πείσουν τους επιστήμονες για την χρησιμότητά τους. Η εν λόγω λοιπόν μετατροπή τους σε power basis πολυώνυμα ήταν σχεδόν απαραίτητη. Άλλωστε πολλοί αλγόριθμοι στον χώρο της άλγεβρας, όπως εκείνοι εύρεσης ριζών πολυωνύμων, έχουν αναπτυχθεί με σκοπό την διαχείριση τέτοιου τύπου πολυωνύμων.

Ας δούμε λοιπόν με ποιο τρόπο όσα αναπτύχθηκαν πιο πάνω μπορούν να μεταφραστούν σε δυο εντολές του Mathematica, γλιτώνοντας μας χρόνο και κόπο. Στα πλαίσια της εν λόγω διπλωματικής αναπτύξαμε μια δική μας συνάρτηση στο Mathematica, τη `BernToPower[coeff,n]` η οποία δέχεται σαν ορίσματα μια λίστα με τους συντελεστές των Bernstein πολυωνύμων και τον βαθμό τους, και επιστρέφει σαν έξοδο το ισοδύναμο power basis πολυώνυμο.

Έχουμε στη διάθεσή μας τις ακόλουθες συναρτήσεις

Differences[list] : δίνει μια λίστα με τις διαφορές των στοιχείων της λίστας εισόδου

NestList[f,expr,n] : δίνει μια λίστα με τα αποτελέσματα εφαρμογής της f στην $expr$ από 0 έως n φορές

Part[list,n] : δίνει το n -οστό στοιχείο της λίστας $list$

Take[list,n] : δίνει τα πρώτα n στον αριθμό στοιχεία της λίστας $list$

Καταλήγουμε λοιπόν στον σχεδιασμό της ακόλουθης συνάρτησης

```
BernToPower[coeff_,n_] := Module[ { m = 0, m2 = Table[m,{m,n+1} ] },  
  Bern = Table[Take[Part[NestList[Differences,coeff,n],m],1],{m,m2});  
  For[ i=1; p=0, i≤n+1, i++,  
    p = Take[Part[Bern,i]]*Binomial[n,i-1]*x^(i-1)+p; p ]
```

Εφόσον “τρέξουμε” στο φύλλο εργασίας μας τον πιο πάνω ορισμό, εκτελούμε τις δυο ακόλουθες εντολές

```
In[2] := coeff = {1,3,4,6,8} ;  
        BernToPower[coeff,4]
```

```
Out[2] := {1 + 8x - 6x2 + 8x3 - 3x4}
```

που ταυτίζεται με το αποτέλεσμα στο οποίο είχαμε καταλήξει και προηγουμένως.

4. Bernstein πολυώνυμα και η συμβολή τους στο χώρο του CAGD

4.1 Εισαγωγή

Το 1974 στην πολιτεία των Ηνωμένων Πολιτειών Γιούτα έλαβε χώρα το πρώτο συνέδριο για τον σχεδιασμό επιφανειών με τη χρήση υπολογιστή. Οι διοργανωτές του B. Barnhill και R. Riskenfeld επινόησαν για αυτή την συγκέντρωση τον – γνωστό πλέον – όρο CAGD (Computer Aided Geometric Design). Για την εν λόγω περίπτωση συγκεντρώθηκαν πολλοί ερευνητές τόσο από τις Ηνωμένες Πολιτείες όσο και από την Ευρώπη. Τα αποτελέσματα του εν λόγω συνεδρίου αποδείχθηκαν ισχυρή επιρροή για τις μετέπειτα έρευνες.

Οι πρώτες σημειώσεις, σε μορφή βιβλίου, εκδόθηκαν το 1979 από τους I. Faux και M. Pratt με τίτλο “Computational Geometry for Design and Manufacture”. Η πρώτη δημοσίευση άρθρου με τίτλο “Computer Aided Geometric Design” πραγματοποιήθηκε το 1984 από τους R. Barnhill και W. Boehm. Ένα ακόμη συνέδριο διοργανώθηκε το 1971 στην πόλη του Παρισιού. Κεντρικός άξονας ήταν ο σχεδιασμός στην αυτοκινητοβιομηχανία και υπεύθυνος ήταν ο P. Bézier, που τότε ήταν πρόεδρος της Société des ingénieurs de l'automobile. Όσα προέκυψαν από τις συζητήσεις εκδόθηκαν σε άρθρο με τίτλο “Ingénieurs de l'automobile”.

4.2 Αρχικοί Σχεδιασμοί

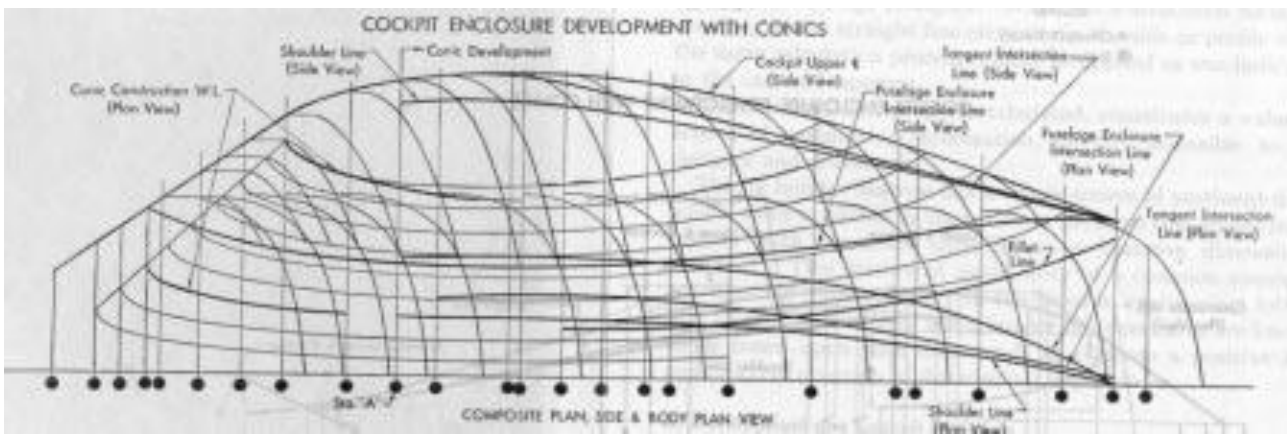
Η πιο παλιά καταγεγραμμένη χρήση καμπυλών (curves) στο κατασκευαστικό περιβάλλον συναντάται στα Ρωμαϊκά χρόνια για την εξυπηρέτηση της ναυπηγικής. Τότε οι πλευρές του πλοίου παράγονταν χάρη βασικού προσχεδίου, το οποίο θα χρησιμοποιούνταν ξανά και ξανά. Έτσι το προσχέδιο για την δημιουργία των σκαφών ήταν δυνατό να αποθηκευτεί και δεν απαιτούνταν να κατασκευαστεί ξανά εξ' αρχής όποτε προέκυπτε η ίδια ανάγκη. Η τεχνική αυτή τελειοποιήθηκε από τους Βενετούς μεταξύ του 13^{ου} και του 16^{ου} αιώνα.

Η μορφή των πλευρών του πλοίου καθοριζόταν από συνεχόμενα εφαπτόμενα κυκλικά τόξα, NURBS στη σύγχρονη διάλεκτο. Ο σκελετός του πλοίου δημιουργούνταν μεταβάλλοντας το σχήμα των πλευρών, ακολουθώντας την καρίνα του πλοίου, και αποτελεί μια πρωταρχική χρήση των τανυστών στον καθορισμό και σχεδιασμό επιφανειών. Τα πρώτα σχέδια για τον ορισμό της μορφής του σκελετού του πλοίου έγιναν γνωστά στην Αγγλία τον 17^ο αιώνα, οπότε και η χρήση των “spline” ξεκίνησε. Βέβαια, δεν επρόκειτο για τίποτα περισσότερο από μια ξύλινη δοκό, η οποία εφευρέθηκε προκειμένου να σχεδιάζονται ομαλές καμπύλες. Η πιο παλιά διαθέσιμη αναφορά σε “splines” έγινε το 1756. Η ναυπηγική αποτέλεσε την πρώτη εφαρμογή γεωμετρικών σχεδίων για τον καθορισμό μορφών επιφανειών πέρα από διαθέσιμα μοτίβα.

Ένα ακόμη κομβικό σημείο για την εδραίωση του CAGD αποτέλεσε η αεροναυτική. Το 1944, ο R. Liming έγραψε ένα βιβλίο με τίτλο “Analytical Geometry with Application to Aircraft”. Ο Liming κατά τη διάρκεια του Δευτέρου Παγκοσμίου Πολέμου εργαζόταν για την NAA (North America Aviation), εταιρεία που κατασκεύαζε πολεμικά αεροσκάφη όπως τα θρυλικά Mustang (Εικόνα 4.1). Στο βιβλίο του συνδύαζε για πρώτη φορά κλασικές μεθόδους σχεδίασης με υπολογιστικές τεχνικές. Κατά κύριο λόγο χρησιμοποιούνταν κώνοι για τον σχεδιασμό των αεροσκαφών όπως και των πλοίων, βασισμένοι σε κατασκευές της εποχής των Pascal και Monge.

Παραδοσιακά τότε οι σχεδιαστές χρησιμοποιούσαν προσχέδια για την πρωταρχικό ορισμό των επιθυμητών κατασκευών. Ο Liming εισήγαγε την ιδέα πως πιο αποτελεσματική θα ήταν η αποθήκευση των σχεδίων σε μορφή αριθμητικών όρων, παρά χρήσει προτύπων σχεδίων που είχαν γίνει με το χέρι. Έτσι προχώρησε στη “μετάφραση” των κλασικών σχεδιαστικών οδηγιών σε αριθμητικούς αλγόριθμους.

Το αποτέλεσμα ήταν η αποθήκευση των στοιχείων των σχεδίων σε αριθμητικούς πίνακες, χωρίς το περιθώριο αμφισβήτησης τους, καταργώντας το φαινόμενο των οποιονδήποτε προσωπικών ερμηνειών στα πρότυπα σχέδια. Η συνεισφορά του Liming είναι εμφανής στη δεκαετία του 1950, καθώς η πρότασή του υιοθετήθηκε ευρέως από τις εταιρείες αεροναυτικής στις Ηνωμένες Πολιτείες εκείνη την χρονική περίοδο.



Εικόνα 4.1 Liming : κωνική κατασκευή του πιλοτηρίου πολεμικού αεροσκάφους

Μια ακόμη εφαρμογή που επηρέασε την ανάπτυξη του πεδίου CAGD ήταν η άφιξη των αριθμητικών ελέγχων (numerical control – NC) τη δεκαετία του 1950. Οι πρώτοι υπολογιστές είχαν την δυνατότητα να παράξουν αριθμητικές ακολουθίες που μετασχηματίζονταν σε εντολές μηχανών κατασκευής σχεδίων σε μεταλλικά μέρη. Στο MIT σχεδιάστηκε ειδική γλώσσα προγραμματισμού - APT - για την διεκπαιρέωση του συγκεκριμένου σκοπού. Παρά τις προσπάθειες, ένα πρόβλημα παρέμενε άλυτο : όλη η σχετική πληροφορία ήταν αποθηκευμένη σε χειρόγραφα πρότυπα σχέδια και δεν ήταν ξεκάθαρος ο τρόπος με τον οποίο μπορούσε να μετατραπεί σε μορφή αντιληπτή από τους υπολογιστές που καθοδηγούσαν τις μηχανές.

Η ψηφιοποίηση σημείων των σχεδίων και η χρήση καμπυλών για την προσέγγιση των αρχικών σχεδίων, με μεθόδους σαν την παρεμβολή σημείων του Lagrange, απέτυχαν στα αρχικά στάδια. Ενώ αναζητούνταν νέες ιδέες για την μεταφορά των σχεδίων στον υπολογιστή, στην Γαλλία δυο μαθηματικοί, ο de Casteljau και ο Bézier, ξεπέρασαν κατά πολύ τον συγκεκριμένο στόχο δίνοντας την δυνατότητα στους σχεδιαστές να εγκαταλείψουν την μέχρι τότε ευρεία ακολουθούμενη μέθοδο των σχεδίων προτύπων.

Η αυτοκινητοβιομηχανία General Motors ανέπτυξε το πρώτο CAD/CAM σύστημα DAC-I (Design Augmented by Computer). Έκανε χρήση βασικών καμπυλών και διαδικασιών αναπαράστασης επιφανειών που σχεδιάστηκαν από ερευνητές όπως οι B. De Boor και ο W.Gordon αλλά και ο S.Coons του MIT των Ηνωμένων Πολιτειών.

Όλες οι προαναφερθείσες μελέτες έλαβαν χώρα στη δεκαετία του 1960. Για σεβαστό χρονικό διάστημα βρίσκονταν στο περιθώριο, έως τη δεκαετία του 1970 οπότε άρχισε να γίνεται εμφανής η συμβολή των διαφορετικών ερευνητικών προσεγγίσεων που οδήγησαν στην ανάπτυξη του τομέα CAGD. Η έλευση των υπολογιστών ήταν εκείνη που έδωσε το έναυσμα για την έρευνα και την εξέλιξη τομέων όπως ο CAGD.

Βέβαια, η συμβολή των υπολογιστών δεν θα μπορούσε να θεωρηθεί εξ'αρχής της ίδιας βαρύτητας. Η πρωταρχική τους χρήση δεν ήταν για τον υπολογισμό σύνθετων καμυλών και σχεδίων αλλά για την παραγωγή χρήσιμης πληροφορίας για σχεδιαστικές μηχανές. Η απαραίτητη πληροφορία αποτυπωνόταν σε διάτρητες κάρτες από τους υπολογιστές και έπειτα εισάγονταν στην μονάδα ελέγχου της μηχανής.

Το σχεδιαστικό ενδιαφέρον βέβαια δεν προσανατολιζόταν σε τέτοιου είδους μηχανές αλλά σε plotters (εκτυπωτές σχεδίων), οι οποίοι μπορούσαν να αναπαραστήσουν γραφικά την ιδέα του σχεδιαστή πολύ γρήγορα. Οι πρώτοι σχεδιογράφοι είχαν το μέγεθος ενός τραπεζιού μπιλιάρδου ή μεγαλύτεροι, κάτι λογικό όμως για τα δεδομένα εκείνης της εποχής, καθώς όλα τα μηχανικά μέρη των αυτοκινήτων σχεδιάζονταν σε κλίμακα.

Αυτού του τύπου οι σχεδιασμοί και οι εκτυπώσεις αποδείχτηκαν σημαντικοί σε βαθμό τέτοιο που σχεδόν όλο το CAD επικεντρώθηκε στο να παράγει σχέδια. Ακόμη και ο όρος CAD(Computer Aided Design) από αρκετούς “μεταφραζόταν” σε Computer Aided Drafting. Πριν την άφιξη των υπολογιστών και των σχετικών μέσων, η δημιουργία σχεδίου για την αναπαραστάση ενός αντικειμένου, ακόμη και με την βοήθεια σχετικών υλοποιημένων σχεδίων, μπορούσε να απαιτήσει μια ολόκληρη εβδομάδα ή και παραπάνω, ενώ πλέον η εν λόγω εργασία γινόταν εντός λεπτών.

4.3 Οι De Casteljau και Bézier

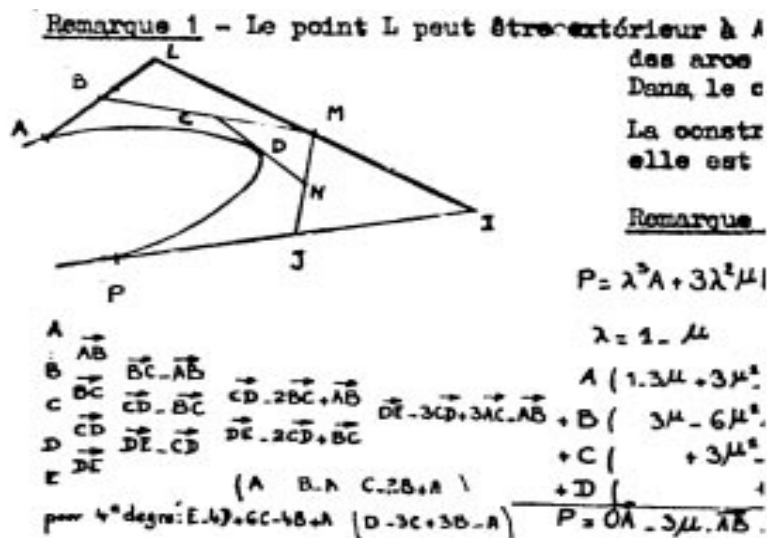
Το 1959, η γαλλική αυτοκινητοβιομηχανία Citroën προσέλαβε ένα νεαρό μαθηματικό προκειμένου να επιλύσει τα θεωρητικά προβλήματα που προέκυπταν από την πρόκληση της εποχής : τη μεταφορά των πρότυπων σχεδίων στη γλώσσα των υπολογιστών. Δεν ήταν άλλος από τον *Paul de Faget de Casteljau*, ο οποίος μόλις είχε ολοκληρώσει την διδακτορική του διατριβή. Προτίμησε να αναπτύξει ένα σύστημα που πρωτίστως είχε στόχο να δημιουργήσει σχέδια καμυλών και επιφανειών *ab initio* (εξ'αρχής), παρά να επικεντρωθεί στον τρόπο μεταφοράς των έτοιμων σχεδίων από το χαρτί στον υπολογιστή (όπως αρκετοί είχαν προσπαθήσει μέχρι τότε).

Για τους ορισμούς των καμυλών και των επιφανειών εν γένει υιοθέτησε, από τις πρώτες του μελέτες, τη χρήση των Bernstein πολυωνύμων σε συνδυασμό με την -πλέον γνωστή σε όλους- ιδέα του, τον *αλγόριθμο de Casteljau*. Η διορατικότητά του φάνηκε από το γεγονός πως χρησιμοποίησε πολύγωνα ελέγχου (control polygons), που δεν είχαν συμμετάσχει σε καμία προσπάθεια που είχε γίνει μέχρι τότε. Με αυτό τον τρόπο, για να μεταβάλλουμε μια καμπύλη(επιφάνεια) δεν αλλάζουμε την ίδια άμεσα αλλά το control polygon στο οποίο βασίζεται, και η καμπύλη (επιφάνεια) το ακολουθεί “ενστικτωδώς”.

Η Citroën κρατούσε τη δουλειά του de Casteljau κρυφή για μεγάλο χρονικό διάστημα. Η πρώτη δημόσια αναφορά στον εν λόγω αλγόριθμο έγινε παραλείποντας το όνομα του εφευρέτη του, ενώ ο πρώτος που αναγνώρισε

επίσημα την προσφορά του στην ερευνητική κοινότητα ήταν ο W.Boehm στα τέλη της δεκαετίας του 1970, οπότε και ανακάλυψε τις τεχνικές αναφορές του εφευρέτη και ήταν εκείνος που επινόησε τον όρο “de Casteljau algorithm” (Εικόνα 4.2).

Ο de Casteljau αποσύρθηκε από την Citroën το 1989 και τότε έγινε ενεργός στον χώρο των επιστημονικών δημοσιεύσεων.



Εικόνα 4.2 Περιγραφή του de Casteljau για τον αλγόριθμό του

Στο Παρίσι βρίσκεται επίσης η αυτοκινητοβιομηχανία της Renault, η οποία είχε ενημερωθεί για τις προσπάθειες της ανταγωνίστριας της Citroën στο κομμάτι του CAGD. Στις αρχές του 1960, ο επικεφαλής του τμήματος σχεδίασης, Pierre Bézier, επίσης συνειδητοποίησε την ανάγκη της αναπαράστασης των μηχανικών σχεδίων στον υπολογιστή. Ήταν ενήμερος για τον τρόπο με τον οποίο προσέγγιζαν στην Citroën το συγκεκριμένο κομμάτι και επηρεάστηκε αρκετά από τα αποτελέσματα των ερευνών τους, προσεγγίζοντας όμως την λύση του προβλήματος με το δικό του τρόπο. Η αρχική του ιδέα ήταν η αναπαράσταση μιας “βασικής καμπύλης” (basic curve) σαν την τομή δυο ελλειπτικών κυλίνδρων, ορισμένων στα όρια παραλληλεπίπεδου. Συσχετισμένοι μετασχηματισμοί του παραλληλεπίπεδου οδηγούν σε συσχετισμένους μετασχηματισμούς και στις καμπύλες ¹.

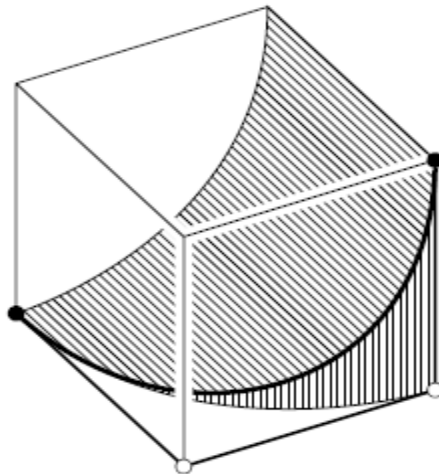
Αργότερα ο Bézier προχώρησε την ιδέα του χρησιμοποιώντας πολυώνυμα για την κεντρική του ιδέα και την επέκτεινε για μεγαλύτερους βαθμούς πολυωνύμων. Το αποτέλεσμα οπτικά είναι ίδιο με εκείνο των καμπυλών του de Casteljau, μόνο που οι μαθηματικοί όροι που συμμετέχουν για το τελικό αποτέλεσμα είναι διαφορετικοί. Ένας μέλος της ομάδας του Bézier, D.Vernet ανέπτυξε ανεξάρτητα τον αλγόριθμο de Casteljau.

¹ Ένας μετασχηματισμός που διατηρεί την παραλληλία μεταξύ των γραμμών ονομάζεται **συσχετισμένος μετασχηματισμός**

Η δουλειά του Bézier δημοσιεύθηκε και πολύ σύντομα τράβηξε το ενδιαφέρον του A.R.Forrest, ο οποίος συνειδητοποίησε πως υπήρχε η δυνατότητα να εκφραστούν οι καμπύλες του Bézier σε όρους των Bernstein πολυωνύμων – κάτι που είχε εφαρμόσει ο de Casteljau ήδη από τα τέλη του 1950. Το άρθρο που δημοσίευσε “Interactive interpolation and approximation by Bézier polynomials” το 1972 είχε αντίκτυπο στον ερευνητικό κόσμο και βοήθησε στη διάδοση των καμπυλών Bézier.

Το CAM/CAD σύστημα της Renault, UNISURF, βασίστηκε εξ’ ολοκλήρου στις καμπύλες και επιφάνειες Bézier. Επηρέασε επίσης έρευνες και κατασκευές της γαλλικής εταιρείας αεροναυτικής Dassault, η οποία δημιούργησε ένα σύστημα που το ονόμασε EVE. Το σύστημα αυτό συμμετείχε αργότερα και στο CATIA (Computer Aided Three-Dimensional Interactive Application).

Ο Bézier εφηύρε επίσης και μια μέθοδο με την οποία μπορούσε να παραμορφώσει συμπλέγματα επιφανειών, εισάγοντας τις αρχικά σε ένα κύβο και έπειτα τους άλλαζε σχήμα χρησιμοποιώντας “Bézier cubes” τριών μεταβλητών, με βήμα της μεθόδου να αναπαρίσταται στην ακόλουθη εικόνα (Εικόνα 4.3).



Εικόνα 4.3 Η “κυβική” καμπύλη του Bézier

4.4 Bézier curves Technology

Οι γραφικές αναπαραστάσεις καμπυλών και επιφανειών εν γένει, είναι ένας τομέας που κεντρίζει το ενδιαφέρον των ερευνητών και αυτό συμβαίνει τουλάχιστον λόγω της ραγδαίας εξέλιξης του Computer Aided Design και της τεχνολογίας των γραφικών των παιχνιδιών. Δυο από τις πιο δημοφιλείς αναπαραστάσεις του εν λόγω τομέα είναι οι Bézier και B-spline καμπύλες και επιφάνειες.

Τις δυο τελευταίες δεκαετίες ο τομέας των γραφικών υπολογιστών έχει κυριολεκτικά γνωρίσει εκρηκτική ανάπτυξη προς όλες τις κατευθύνσεις, και σε αυτό έχουν συντελέσει γεγονότα όπως οι απαιτήσεις στα γραφικά, οι πιο γρήγοροι υπολογιστές και το νέο υλικό (hardware) που εξελίσσεται και υποστηρίζει τα νέα υπολογιστικά συστήματα. Μια τέτοιου τύπου κατεύθυνση είναι η αναπαράσταση ομαλών καμπυλών και επιφανειών γενικά, κατάλληλη για αναπαράσταση ανάγλυφου εδαφών, προσώπων και άλλων επιθυμητών αντικειμένων. Σε αυτό το σημείο άλλωστε γίνεται πιο έντονη η ανάγκη των καμπυλών Bézier και B-spline.

Οι δυο αναφερόμενοι τύποι καμπυλών είναι κατασκευασμένες από αυθαίρετο κάθε φορά αριθμό σημείων ελέγχου, τα οποία σχηματίζουν έναν οδηγό τον οποίο προσεγγίζει η υπολογιζόμενη καμπύλη. Το αποτέλεσμα της προσέγγισης ποικίλει ανάλογα με τις ανάγκες που υπάρχουν : μπορούν να χρησιμοποιηθούν σημεία ελέγχου με βάρη, διαφορετικός βαθμός αναπαράστασης της καμπύλης και πολλά ακόμη παραμετρικά δεδομένα. Είναι πιθανό κάποιος να αναρωτηθεί γιατί οι παραμετρικές αναπαραστάσεις με τη χρήση καμπυλών αξίζει να προτιμηθεί από την αναπαράσταση με τη βοήθεια πολυγωνικών αναπαραστάσεων. Οι πιο απλοί λόγοι είναι το πολύ μικρό μέγεθος των καμπυλών, η κλιμάκωση που παρουσιάζουν (μπορούν να μεταφερθούν μεταξύ μικρών και μεγάλων συστημάτων με ευκολία) και πολυάριθμα ακόμη πλεονεκτήματα που θα αναφερθούν και στη συνέχεια της ενότητας.

4.4.1 Bézier curves and surfaces

Αρχικά θα εστιάσουμε στις κυβικές πολυωνυμικές καμπύλες καθώς είναι οι πιο χρησιμοποιούμενες στον χώρο των γραφικών υπολογιστών (computer graphics) και ταυτόχρονα είναι εύκολο να επεκταθεί η εν λόγω προσέγγιση για πολυώνυμα μεγαλύτερου βαθμού. Για τον σχηματισμό των καμπυλών Bézier χρησιμοποιούνται σαν βασική συνάρτηση τα πολυώνυμα Bernstein, που παρουσιάστηκαν στην Ενότητα 2.

Μια κυβική καμπύλη Bézier αποτελείται από τέσσερα στον αριθμό σημεία ελέγχου, που προσδίδουν στην καμπύλη την θέση αλλά και το σχήμα της, $P_0 \dots P_3$. Τα ενδιάμεσα σημεία P_1 και P_2 χρησιμοποιούνται για να καθορίσουν τα εφαιπόμενα διανύσματα στα άκρα της καμπύλης, καθώς εκείνη διέρχεται από τα P_0 και P_3 ενώ προσεγγίζει τα άλλα δυο σημεία ελέγχου. Για να επιτευχθεί αυτός ο στόχος είναι απαραίτητη μια συνάρτηση με βάρη, η οποία θα καθορίζει την επιρροή των σημείων ελέγχου σε κάθε σημείο της καμπύλης. Παρά το γεγονός πως είναι δυνατό να χρησιμοποιηθεί οποιαδήποτε συνάρτηση για το σκοπό αυτό, πιο σύνηθες είναι να χρησιμοποιούνται τα πολυώνυμα Bernstein :

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad (\text{Σχ. 4.1})$$

ή αλλιώς

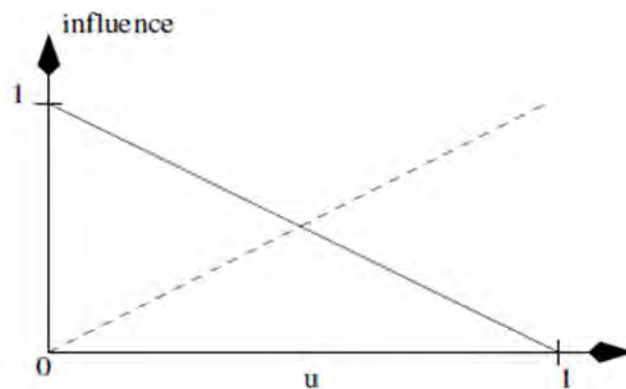
$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (\text{Σχ. 4.2})$$

όπου το i απεικονίζει το σημείο ελέγχου στο οποίο δίνεται βαρύτητα και το n ορίζει τον βαθμό της καμπύλης που θα χρησιμοποιηθεί σαν βασική συνάρτηση. Τα πολυώνυμα, όπως έχουμε αναφέρει ήδη, είναι συμμετρικά και αθροίζουν στη μονάδα, ιδιότητες που αποκτά και η καμπύλη Bézier που προκύπτει από την χρήση των εν λόγω πολυωνύμων.

Για να γίνει πιο απτή η συμπεριφορά της παραμέτρου $u \in (0,1)$ θα μπορούσε να χρησιμοποιηθεί η εξής προσέγγιση, ας αναπαρασταθεί λοιπόν με ένα σωματίδιο το οποίο προσπαθεί να παρεμβληθεί των σημείων ελέγχου με διακριτά βήματα. Καθώς κινείται κατά μήκος της τροχιάς, του ασκείται “έλξη” από το σημείο ελέγχου $B_{i,3}(u)$, στην περίπτωση της κυβικής προσέγγισης.

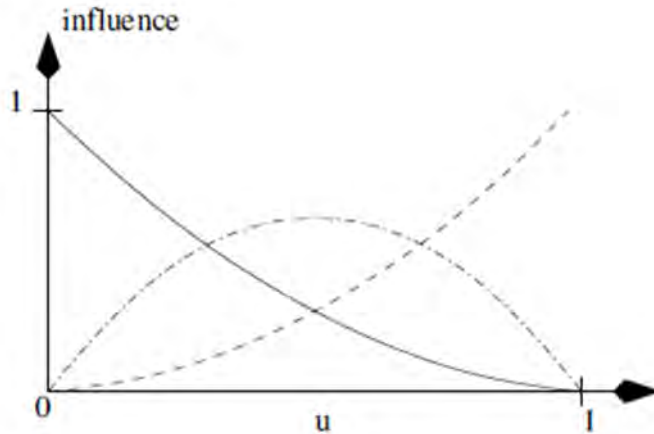
Με τη βοήθεια των σχημάτων που ακολουθούν θα γίνει πιο αντιληπτή η απεικόνιση των πολυωνύμων σε συνδυασμό με την προαναφερθείσα απλουστευμένη θεωρία.

Ας ξεκινήσουμε με την περίπτωση που $n=1$, δηλαδή τα Bernstein πολυώνυμα είναι πρώτου βαθμού(η έννοια της καμπύλης χάνεται καθώς το αποτέλεσμα είναι μια ευθεία γραμμή). Όταν $u=0$ το δεύτερο σημείο ελέγχου (στην ακόλουθη εικόνα αναπαρίσταται με διακεκομμένη γραμμή) δεν επηρεάζει καθόλου το u . Καθώς όμως το τελευταίο κινείται εντός του διαστήματος $0 \leq u \leq 1$ το πρώτο σημείο ελέγχου κάνει σταδιακά την επίδρασή του στο u και τόσο αυξάνεται η επιρροή του δεύτερου. Το αποτέλεσμα δεν είναι άλλο από μια ευθεία γραμμή μεταξύ των δυο σημείων ελέγχου. Οι βασικές συναρτήσεις που χρησιμοποιούνται στην ακόλουθη εικόνα (Εικόνα 4.4) είναι οι γνωστές $B_{0,1}$ και $B_{1,1}$.



Εικόνα 4.4 Βασικές συναρτήσεις πρώτου βαθμού

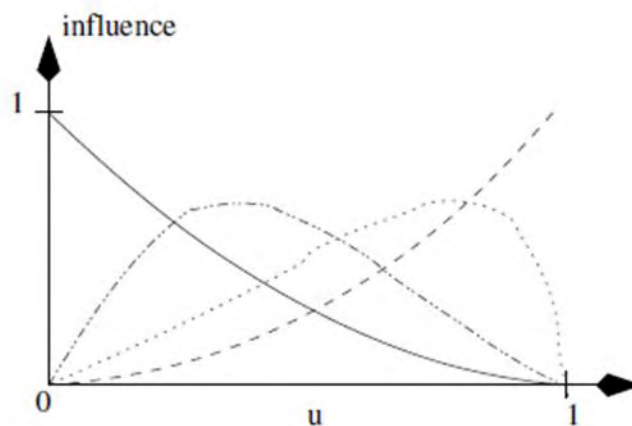
Ακολουθώς εξετάζουμε την περίπτωση του δεύτερου βαθμού πολυωνύμων όπου $n=2$ και οι βασικές συναρτήσεις είναι πολυώνυμα δευτέρου βαθμού. Τα δυο πηγουόμενα σημεία ελέγχου παρουσιάζουν απότομη αλλαγή στο μέγεθος της επιρροής που ασκούν – προσθετικά ή αφαιρετικά- χωρίς όμως να χάνουν ποτέ ολοκληρωτικά την επιρροή τους, παρά μόνο στα δυο ακραία σημεία. Αυτή η ιδιότητα είναι άλλωστε που δίνει την δυνατότητα στην καμπύλη Bézier να είναι κοίλη ή κυρτή (concave ή convex αντίστοιχα), όπως όλα τα πολυώνυμα δευτέρου βαθμού. Οι βασικές συναρτήσεις που χρησιμοποιούνται σε αυτό το βήμα, και είναι οι $B_{0,2}$, $B_{1,2}$ και $B_{2,2}$, εμφανίζονται στην (Εικόνα 4.5).



Εικόνα 4.5 Βασικές συναρτήσεις δευτέρου βαθμού

Αξίζει να αναφέρουμε πως μέχρι εδώ δεν υπάρχει κάποιο τοπικό σημείο ελέγχου, καθώς μεταβάλλοντας τη θέση ενός σημείου ελέγχου αλλάζει εξ'ολοκλήρου το σχήμα της καμπύλης.

Συνεχίζοντας με την περίπτωση των κυβικών πολυωνύμων σαν βασική συνάρτηση, εδώ $n=3$, διαπιστώνουμε πως το αποτέλεσμα δεν διαφέρει σε τίποτα από μια κοινή κυβική καμπύλη που ταυτόχρονα είναι κοίλη και κυρτή. Ο συγκεκριμένος βαθμός, σύμφωνα με την πλειοψηφία των τακτικών χρηστών, είναι και ο πιο ευέλικτος και εύπλαστος για να εργαστεί κανείς αποδοτικά. Στις περιπτώσεις που $n < 3$ δεν υπάρχει αρκετή "ελευθερία" για να φτάσει κανείς σε ένα ικανοποιητικό αποτέλεσμα, και όταν $n > 3$ η καμπύλη γίνεται δύσχρηστη καθώς αποκτά την τάση να "ξεφεύγει" απότομα σε μη επιθυμητά σημεία χωρίς να υπάρχουν περιθώρια τιθάσευσης. Στην (Εικόνα 4.6) εμφανίζονται οι βασικές συναρτήσεις $B_{0,3}$, $B_{1,3}$, $B_{2,3}$, $B_{3,3}$.



Εικόνα 4.6 Βασικές συναρτήσεις τρίτου βαθμού

Δεν είναι περιττό να αναφέρουμε ξανά πως, σε οποιαδήποτε από τις πιο πάνω εικόνες, οι διαφορετικές τιμές των καμπυλών αθροίζουν στη μονάδα, άσχετα με το βαθμό της καμπύλης.

4.4.2 Ορισμός καμπυλών

Έχοντας στο μυαλό μας τους πιο πάνω ορισμούς προχωρούμε στον ορισμό της καμπύλης Bézier βαθμού n

$$C(u) = \sum_{i=0}^n P_i B_{i,n}(u), u \in (0,1) \text{ (Σχ. 4.3)}$$

και για τον υπολογισμό της καμπύλης επαναλαμβάνονται τα $n+1$ σημεία ελέγχου και υπολογίζουμε την αντίστοιχη επιρροή.

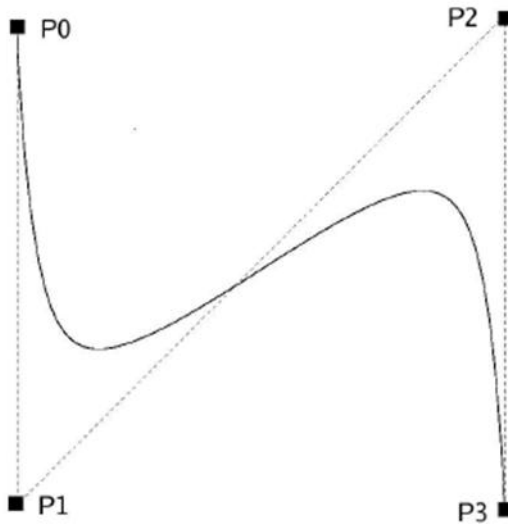
Για κάποιον εξοικειωμένο με την γραμμική άλγεβρα ο προαναφερθής ορισμός μπορεί να απεικονιστεί χρήσει πινάκων ως εξής

$$C(u) = U M_B G_B, \text{ (Σχ. 4.4)}$$

όπου U είναι το διάνυσμα των $[u^n, u^{n-1}, \dots, u, 1]$, M_B είναι ο πίνακας βάσης Bézier και G_B είναι το διάνυσμα των $n+1$ σημείων ελέγχου. Βέβαια η ακόλουθη προσέγγιση θα γίνει βάσει της πρώτης και πιο αναλυτικής αναπράστασης καθώς είναι πιο βολική από την οπτική γωνία των σχεδιαστών.

Στην ακόλουθη εικόνα (Εικόνα 4.6) είναι εμφανές το πως αντανakλά η συμμετρία των Bernstein πολυωνύμων σε μια καμπύλη. Το ίδιο ισχύει και στην περίπτωση που χρησιμοποιούταν ο αλγόριθμος de Casteljau. Από τον ορισμό (Σχ 4.3) ανακαλύπτουμε πως ανεξαρτήτως του βαθμού της καμπύλης, δεν υπάρχει κάποια ιδιότητα τοπικού ελέγχου. Όλα τα σημεία ελέγχου επηρεάζουν σε κάποιο βαθμό την τελική μορφή της καμπύλης, κάτι που μπορεί να θεωρηθεί μειονέκτημα στην προσπάθεια σχεδιασμού λείων καμπυλών και σχημάτων χρήσει της καμπύλης Bézier. Βέβαια για την αντιμετώπιση του εν λόγω προβλήματος έχει προταθεί η ακόλουθη λύση : χρήση καμπυλών που έχουν σχεδιαστεί από πολλά τμήματα Bézier καμπυλών, θέμα που αναλύεται στην επόμενη υποενότητα.

Για να ολοκληρώσουμε τον ορισμό των καμπυλών Bézier, απεικονίζεται στο ακόλουθο σχήμα (Εικόνα 4.6) μια τρίτου βαθμού καμπύλη με τέσσερα στον αριθμό σημεία ελέγχου, ενώ εμφανίζεται και το πολύγωνο ελέγχου διαγραμματισμένο. Αξίζει να παρατηρήσουμε τον τρόπο που τα διανύσματα P0P1 και P2P3 καθορίζουν την κατεύθυνση της καμπύλης στα ακραία σημεία. Η κυρτή περίμετρος (convex hull) προσδιορίζεται από τον χώρο που εσωκλείει το πολύγωνο που δημιουργεί η ένωση των σημείων ελέγχου με ευθύγραμμα τμήματα, θεμελιώδης ιδιότητα των καμπυλών Bézier. Με όποιο τρόπο και αν μεταχειριστούμε τα σημεία ελέγχου, μπορούμε να είμαστε πάντα σίγουροι πως η καμπύλη θα βρίσκεται εντός της κυρτής περιμέτρου.

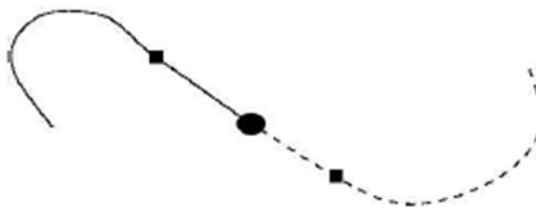


Εικόνα 4.6 Καμπύλη Βέζιερ και πολύγωνο ελέγχου

4.5 Τμήματα καμπυλών (curve segments)

Χρησιμοποιώντας πολλές καμπύλες Βέζιερ σε αλληλουχία έχουμε την δυνατότητα να αποκτήσουμε τοπικό έλεγχο. Κάτι τέτοιο υλοποιείται μοιράζοντας τα δυο ακραία σημεία ελέγχου σε γειτονικές καμπύλες, κάτι που προφανώς επηρεάζει το πρώτο και τελευταίο τμήμα της κάθε καμπύλης. Είναι αναπόφευκτο πως θα δημιουργηθούν ασυνέχειες στο κοινό σημείο ελέγχου. Μια καμπύλη Βέζιερ έχει συνέχεια C^1 στο διάστημα που έχει οριστεί και C^0 στις κοινές κορυφές. Αυτό έχει ως αποτέλεσμα να μην είναι ομαλή η μετάβαση μεταξύ των συμμετεχόντων τμημάτων παρά μόνο εάν δημιουργηθεί χειροκίνητα, κάτι που μπορεί να αποδειχθεί θετικό είτε αρνητικό, ανάλογα με τις απαιτήσεις κάθε εφαρμογής.

Για παράδειγμα, ένα τετράγωνο δεν μπορεί να δημιουργηθεί με κερματισμένες C^1 συνεχείς καμπύλες εκτός και αν δημιουργήσουμε οι ίδιοι ασυνέχεια στις κορυφές του, ενώ ο σχεδιασμός ενός κύκλου απαιτεί ομαλές μεταβάσεις και συνέχεια μεταξύ των καμπυλών.



Εικόνα 4.7 Περιορισμοί συνέχειας

Ας υποθέσουμε πως έχουμε καμπύλες $P = P_0, P_1, \dots, P_{n+1}$ και $Q = P_{n+1}, Q_1, \dots, Q_{m+1}$ βαθμών n και m αντίστοιχα. Προκειμένου να έχουμε εξαναγκασμένη συνέχεια μεταξύ αυτών των δυο καμπυλών πρέπει να υπάρχει διανυσματική συνέχεια μεταξύ των P_n, P_{n+1} και Q_1 . Στην εικόνα που προηγήθηκε (Εικόνα 4.7), έχουμε σημειώσει με τετράγωνο τα σημεία P_n και Q_1 και με κύκλο το σημείο P_{n+1} , που αποτελεί και το σημείο ένωσης των δυο καμπυλών. Το μειονέκτημα αυτού του σχεδιασμού είναι η απώλεια ενός βαθμού ελευθερίας σε κάθε καμπύλη καθώς εκείνες “κλειδώνουν” με σκοπό να επιτευχθεί η επιθυμητή ομαλή μετάβαση μεταξύ τους.

4.6 Bézier υποδιαίρεση (Bézier subdivision)

Στην περίπτωση που μια καμπύλη Bézier δεν είναι αρκετά ευέλικτη έχουμε την δυνατότητα να την “οπάσουμε” σε δεδομένο σημείο u' και να δημιουργήσουμε δυο καμπύλες Bézier που ενώνονται στο εν λόγω σημείο. Ένας αλγόριθμος που υλοποιεί την εν λόγω προσέγγιση είναι ο de Casteljau, και κάνει χρήση τεχνικών γεωμετρικών κατασκευών.

Για να το θέσουμε πιο επίσημα, στόχος είναι η εύρεση σημείων L_0, L_1, L_2, L_3 και R_0, R_1, R_2, R_3 έτσι ώστε η αρχική καμπύλη Bézier να υπολογίζεται τμηματικά με ακριβώς L σημεία για $0 \leq u \leq 0.5$ και άλλα R σημεία για $0.5 \leq u \leq 1.0$. Καθώς πλέον οι L και R θεωρούνται διαφορετικές καμπύλες, υπολογίζονται τοπικά με $0 \leq u \leq 1.0$.

Ας υποθέσουμε πως η αρχική καμπύλη αποτελείται από P_0, P_1, P_2, P_3 . Όταν κατασκευάσουμε ένα σημείο H που διαιρεί τα P_0P_1, P_1P_2 και P_2P_3 με αναλογία $(1-u)$, δημιουργείται μια περίμετρος με σημεία $L_0L_1L_2L_3$ και $R_0R_1R_2R_3$ και τελικά αποκτούμε δυο σει σημείων ελέγχου που εμπεριέχονται στην περίμετρο των σημείων ελέγχου της αρχικής καμπύλης. Άλλωστε οι L και R δεν είναι τίποτε άλλο από αθροίσματα με βάρη που αθροίζουν στην P .

Αυτή είναι η βασική περιγραφή και του αλγορίθμου de Casteljau για τον οποίο θα αναφερθούμε στην υποενότητα 4.9.

4.7 Rational Bézier curves

Rational χαρακτηρίζονται οι καμπύλες για τις οποίες τα σημεία ελέγχου έχουν επεκταθεί κατά μια συντεταγμένη η οποία αναπαριστά το *βάρος* ή αλλιώς το μέγεθος της επιρροής του. Όσο μεγαλύτερο είναι αυτό το μέγεθος τόσο πιο κοντά του τραβά την καμπύλη το σημείο - κορυφή, διαφορετικά την απομακρύνει από εκείνο. Κάτι τέτοιο είναι εξαιρετικής σημασίας για κάποιον που σχεδιάζει καμπύλες για την οθόνη ενός υπολογιστή, καθώς η διαθέσιμη επιφάνεια είναι περιορισμένη. Αντί να διώχνουμε το σημείο ελέγχου πολύ μακριά, σε κάποιες περιπτώσεις και εκτός ορίων του χώρου σχεδιασμού, απλά αυξάνουμε το βάρος συμμετοχής του.

Αυτή η προσθήκη δεν επιφέρει σημαντικές αλλαγές στον τρόπο που υπολογίζαμε έως τώρα τα πράγματα και ισχύει ο ακόλουθος ορισμός

$$C(u) = \frac{\sum_{i=0}^n w_i P_i B_{i,n}(u)}{\sum_{i=0}^n w_i B_{i,n}(u)} \quad , (\Sigma\chi. 4.5)$$

με το w_i να αναπαριστά το βάρος του P_i . Τα δυο αθροίσματα μπορούν να υπολογιστούν επαναλαμβάνοντας τα ίδια σημεία ελέγχου , με αποτέλεσμα σε μικρό υπολογιστικό χρόνο να έχουμε στην κατοχή μας μια καμπύλη που αποδεικνύεται πιο χρήσιμη.

4.8 Bézier επιφάνειες (Bézier surfaces)

Οι Bézier επιφάνειες είναι απλά μια επέκταση των καμπυλών Bézier σε δυο παραμετρικές κατευθύνσεις, δηλαδή ένα διανυσματικό γινόμενο δυο καμπυλών. Ο τύπος υπολογισμού είναι ο ακόλουθος

$$C(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} B_{i,n}(u) B_{j,m}(v) \quad , (\Sigma\chi. 4.6)$$

όπου οι παραμετρικές κατευθύνσεις είναι εξ'ορισμού n και m βαθμού, αν και πολύ συχνά συναντάται η περίπτωση $n=m$. Αποδίδονται με αυτό τον τρόπο $(n+1)*(m+1)$ σημεία ελέγχου.

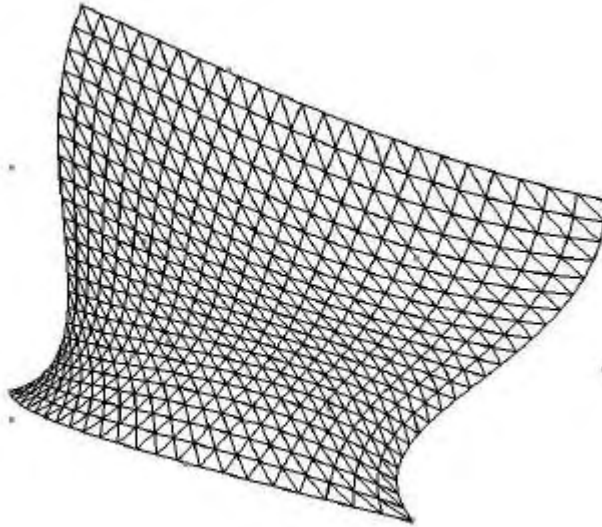
Η πιο πάνω ισότητα προτείνει εμμέσως την αποθήκευση των σημείων ελέγχου σε πίνακα, χωρίς ωστόσο κάτι τέτοιο να αποδεικνύεται απαραίτητο καθώς η σειρά υπολογισμού τους δεν σχετίζεται με το τελικό αποτέλεσμα.

Η ακόλουθη ισότητα αναπαριστά τον υπολογισμό μιας *rational* επιφάνειας, ανάλογη της περίπτωσης με μια παραμετρική κατεύθυνση.

$$C(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} P_{i,j} B_{i,n}(u) B_{j,m}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{i,j} B_{i,n}(u) B_{j,m}(v)} \quad , (\Sigma\chi. 4.7)$$

Όσον αφορά στην αλληλουχία των επιφανειών εφαρμόζουμε τους ίδιους περιορισμούς με την ενότητα 4.5, μόνο που στην προκειμένη περίπτωση προσαρμόζονται για δυο παραμετρικές κατευθύνσεις.

Στην ακόλουθη εικόνα (Εικόνα 4.8) αναπαρίσταται μια κυβική επιφάνεια Bézier με κατασκευασμένα εσωτερικά πολύγωνα.



Εικόνα 4.8 Bézier επιφάνεια

4.9 Αλγόριθμος de Casteljau

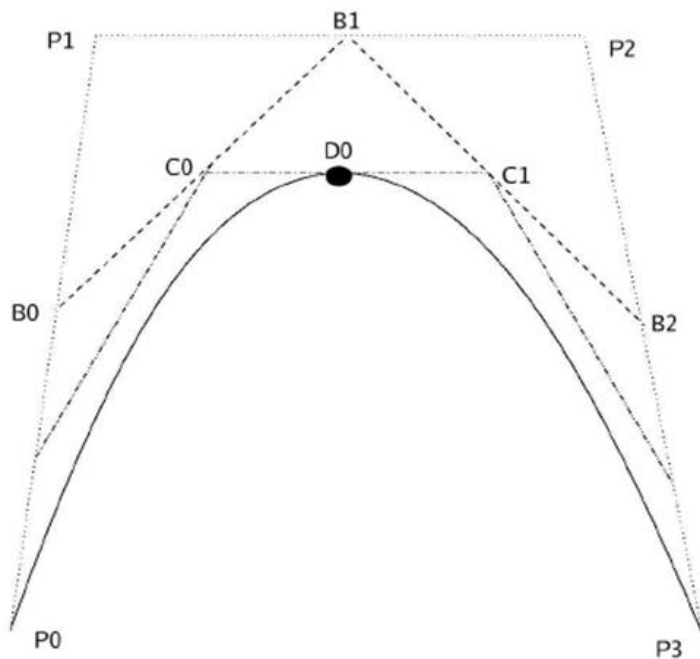
Όπως έχουμε αναφέρει και στην αρχή της ενότητας, ο de Casteljau εφύηρε ένα γεωμετρικό αλγόριθμο (ή καλύτερα έναν αλγόριθμο τον οποίο μπορούμε να ερμηνεύσουμε και γεωμετρικά), ο οποίος προσεγγίζει για οποιοδήποτε σημείο καμπύλη Bézier οποιοσδήποτε βαθμού.

Στην ακόλουθη εικόνα (Εικόνα 2.9) φαίνεται πως ο αλγόριθμος επιλύει για $u=0.5$ μια κυβική καμπύλη Bézier υποδιαιρώντας τα σημεία ελέγχου με αναλογία $1-u$. Κάτι τέτοιο ερμηνεύεται ως εξής: σχεδιάζουμε αρχικά μια περίμετρο μεταξύ των γραμμών $P_0P_1^* 0.5$, $P_1P_2^* 0.5$ και $P_2P_3^* 0.5$ και συνεχίζουμε με τον ίδιο τρόπο χρήσει των νέων σημείων ελέγχου. Εφόσον η καμπύλη είναι τρίτου βαθμού η διαδικασία τερματίζει σε αυτό το σημείο και έχουμε βρει την τιμή της. Η ίδια μεθοδολογία εφαρμόζεται ανεξαρτήτως του βαθμού της καμπύλης, με την αναδρομική διαδικασία όμως να φτάνει σε διαφορετικό βάθος, ανάλογα με τον βαθμό, και είναι επεκτάσιμη για κάθε $u \in (0,1)$.

Ο εν λόγω αλγόριθμος μπορεί να αναπαρασταθεί με ένα τριγωνικό σχήμα, με αρχή τα τέσσερα σημεία ελέγχου, τα οποία σταδιακά μειώνονται έως ότου απομείνει ένα μοναδικό σημείο.

$$\begin{array}{cccc}
 P_0 & & & \\
 P_1 & B_0 & & \\
 P_2 & B_1 & C_0 & \\
 P_3 & B_2 & C_1 & D_0
 \end{array}$$

Όταν έχουμε στη διάθεσή μας το ακόλουθο σχήμα (Εικόνα 4.9) γίνονται ξεκάθαρα τα μαθηματικά που κρύβονται πίσω από τους προηγούμενους υπολογισμούς.



Εικόνα 4.9 Αναπαράσταση του αλγορίθμου de Casteljau

Μια ακόμη αναπαράσταση που μπορεί να θεωρηθεί κοινή είναι η απόδοση της αναδρομικής σχέσης του αλγορίθμου de Casteljau. Ξανά αναφερόμαστε σε τριγωνικό σχήμα, μόνο που αυτή τη φορά η απόδοση γίνεται με την μορφή μαθηματικής σχέσης. Θεωρούμε σαν $P_{0,j}$ το P_j για $j=0,1, \dots, n$ όπου n είναι ο βαθμός της εξεταζόμενης καμπύλης, καθώς $i=1,2, \dots, n$ και $j=0,1, \dots, n-i$. Τότε σημείο $P_{i,j}$ της καμπύλης υπολογίζεται όπως ορίζει η ακόλουθη σχέση ως

$$P_{i,j} = (1 - u) P_{i-1,j} + u P_{i-1,j+1}$$

4.10 Εξετάζοντας θέματα πραγματικού χρόνου

4.10.1 Υπολογισμοί

Μια καμπύλη Bézier μπορεί να υπολογιστεί με μικρό κόστος ακόμη και στην περίπτωση που επιλεγθούν σημεία με μικρά Δu (διαφορά μεταξύ των δυο διαδοχικών σημείων υπολογισμού). Το Δu δεν απαιτείται να είναι σταθερό, αλλά μπορεί να παραμετροποιηθεί ανάλογα με τα διαστήματα στα οποία αναφέρεται βάσει κριτηρίων πολυγώνου ελέγχου.

Ο πιο συνήθης τρόπος για να επαυξήσει κανείς μια επιφάνεια χρειάζεται να την επεκτείνει με μεθόδους, οι οποίες απαιτούν σημαντικό υπολογιστικό χρόνο σε σύγκριση με τις απαιτήσεις του βασικού υπολογισμού των συντεταγμένων της επιφάνειας και τη μεταβολή των υπολογισμένων πολυγώνων μεταξύ διαδοχικών βημάτων υπολογισμού.

4.10.2 Sculpting

Ένα σχεδιαστικό μέσο Bézier καμπυλών ,και επιφανειών εν γένει, θα μπορούσε να θεωρηθεί τουλάχιστον επαρκές, εφόσον επιτρέπει την παραμετροποίηση των βαθμών των συμμετεχουσών καμπυλών και των βαρών των σημείων ελέγχου, με αυτόματη C^1 συνέχεια διατηρώντας αλληλουχία στα νεοεισαγόμενα τμήματα. Στην περίπτωση επιφανειών είναι επιθυμητές παροχές για αυτόματες περιστροφές όπως επίσης και παρεχόμενες επιλογές για διαφορετικούς τύπους μετάβασης μεταξύ διαδοχικών βημάτων υπολογισμού, σκίασης και λοιπών οπτικών αντικειμένων.

Στα βήματα των εφαρμογών, μόνο το πολύγωνο ελέγχου σχεδιάζεται σε πραγματικό χρόνο καθώς μεταχειρίζεται τα σημεία ελέγχου, δεδομένου του μεγέθους του υπολογιστικού κόστους που απαιτείται για τη συνεχή ενημέρωση του οπτικού αποτελέσματος μεταξύ των βημάτων υπολογισμού. Έτσι οδηγούμαστε σε μια πιο “σκληρή” απεικόνιση του τελικού αντικειμένου που αποκαθίσταται με την ολοκλήρωση των απαραίτητων ενημερώσεων. Ένας τρόπος να διατηρηθεί χαμηλά ο φόρτος ενημέρωσης είναι η εκκίνηση με το αντικείμενο να βρίσκεται στην απλούστερη μορφή του (χωρίς χρωματισμό ή τροποποίηση των πολυγώνων μετά από κάθε βήμα υπολογισμού).

Ένα ακόμη ενδιαφέρον θέμα αποτελεί το πόσο πολύπλοκη γίνεται η επεξεργασία επιφανειών (ή τρισδιάστατων καμπυλών) δεδομένου ότι η οθόνη του υπολογιστή παραμένει δισδιάστατη. Χρειάζεται να προβάλουμε την καμπύλη (είτε την επιφάνεια) στη διάσταση της οθόνης (2D) από όπου μπορούμε να διαχειριστούμε τα σημεία ελέγχου με το ποντίκι. Αφότου ολοκληρωθεί και αυτό το βήμα προβάλουμε ξανά το αποτέλεσμα στον υπολογιστικό χώρο (3D).

Δεν αρνείται κανείς πως μετά την περιστροφή της μια επιφάνεια μπορούμε να την διαχειριστούμε έμμεσα σαν να είναι τρισδιάστατη, όμως το αρχικό πρόβλημα δεν εξαλείφεται δεδομένης της δυσκολίας στον ορισμό του βάθους όταν απουσιάζουν καλής ποιότητας δεδομένα που βοηθούν στην σωστή εκτίμηση του (όπως αναφορές σε παρεμφερή αντικείμενα, σκίαση, εστίαση και ούτω καθ'εξής).

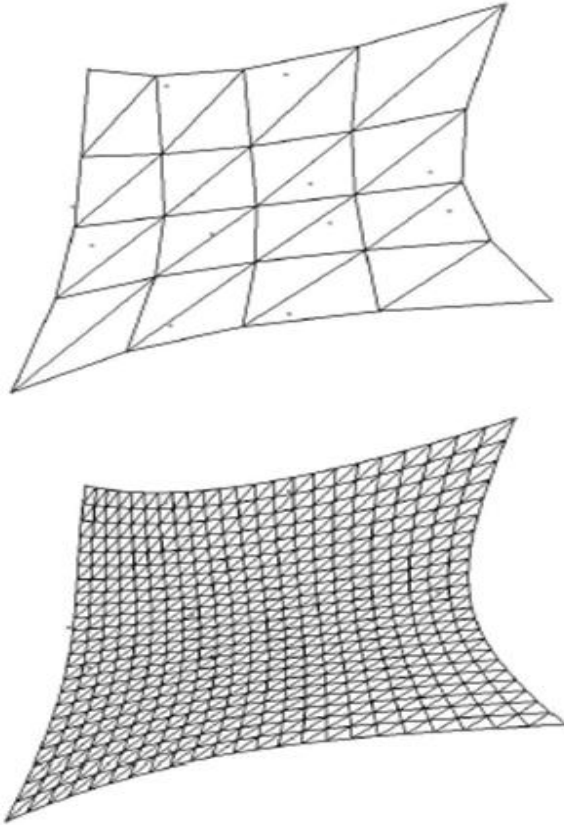
4.11 Εφαρμογές

4.11.1 Collision Detection

Βασικό πρόβλημα του θέματος collision detection δε θα μπορούσε να χαρακτηριστεί άλλο από το πως θα αποφασιστεί εάν δυο αντικείμενα συγκρούονται, με την ελάχιστη απαιτούμενη υπολογιστική ισχύ, διατηρώντας ταυτόχρονα σταθερή τη συμπεριφορά των αντικειμένων. Το παρόν αποτελεί μείζον θέμα για αυτό και θα επικεντρωθούμε στα πλεονεκτήματα που παρουσιάζει η χρήση παραμετρικών αναπαραστάσεων για καμπύλες και επιφάνειες εν γένει, όπως και η επιλογή του τρόπου συμμετοχής τους σε μια διαδραστική εφαρμογή.

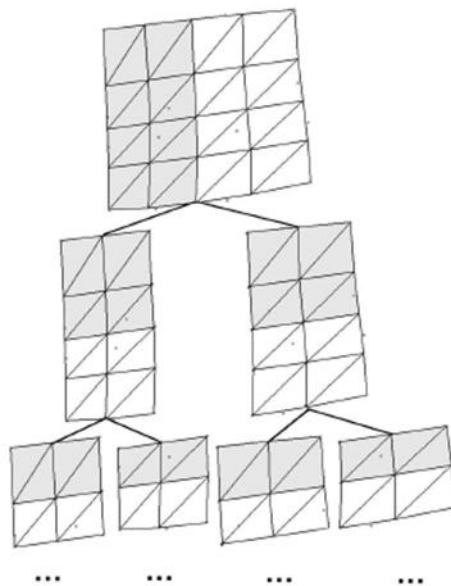
Αρχικά ας θεωρήσουμε κάποια θεμελιώδη στοιχεία τα οποία χρησιμοποιούνται ευρέως στον τομέα του collision detection, όπως δισδιάστατες επιφάνειες, σφαίρες, γραμμές και λοιπά. Στόχος είναι η σύνθεση ενός αντικειμένου χρήσει των προαναφερθέντων βασικών στοιχείων με τέτοιο τρόπο ώστε τυχόν τομή μεταξύ δυο συμμετεχόντων αντικειμένων να ανιχνεύεται πολύ σύντομα. Υπάρχουν γρήγοροι αλγόριθμοι για τους περισσότερους συνδυασμούς όπως σφαίρα-γραμμή, πλαίσιο-δισδιάστατη επιφάνεια και δισδιάστατη επιφάνεια-γραμμή, των οποίων θα γίνει χρήση προκειμένου να γίνει μια βασική ανίχνευση συγκρούσεων μεταξύ αντικειμένων και να χρησιμεύσουν σαν οριοθέτες.

Σε μια διεργασία, όχι πραγματικού χρόνου, διαβάζουμε τα σημεία ελέγχου και στη συνέχεια δημιουργούμε δυο εκδοχές της επιφάνειας, στην πρώτη περίπτωση ικανοποιώντας την απαιτούμενη ανάλυση της επιθυμητής οπτικής αναπαράστασης και μια ακόμη με την ανάλυση που είναι κάθε φορά χρήσιμη στην μηχανή ανίχνευσης συγκρούσεων. Στην ακόλουθη εικόνα (Εικόνα 4.10 άλλωστε παρατηρούμε πως τα σημεία ελέγχου παραμένουν σταθερά (δεν αλλάζουν θέση) ο υπολογισμός όμως της επιφάνειας εν τέλει γίνεται σε δυο διαφορετικές αναλύσεις.



Εικόνα 4.10 Διαφορετικό μέγεθος ανάλυσης

Η εν λόγω τακτική συχνά συνδυάζεται με μια βοηθητική δομή δέντρου. Σε αυτή την περίπτωση χωρίζουμε την επιφάνεια σε πολλαπλούς ιεραρχικά οριοθετημένους κόμβους όπου κάθε κορυφή –με εξαίρεση την ρίζα και τα φύλλα του δέντρου- έχει τέτοιους όγκους σαν πατέρα και παιδιά. Η ακόλουθη εικόνα (Εικόνα 4.11) αναπαριστά την θεωρητική μας προσέγγιση



Εικόνα 4.11 Κόμβοι ορίων

Η συγκεκριμένη τεχνική θεωρείται κλασικός τρόπος προσέγγισης του αρχικού θέματος με όσο το δυνατόν λιγότερους “αχρείαστους” υπολογισμούς. Αρχικά επιχειρούμε να ανιχνεύσουμε σύγκρουση στον κόμβο πατέρα. Στην περίπτωση που δεν υπάρχουν τομές ολοκληρώνουμε την ανίχνευση με λίγους “άσκοπους” υπολογισμούς, διαφορετικά συνεχίζουμε την διαδικασία με τους κόμβους παιδιά.

Είναι δυνατό να μην υπολογιστεί το δέντρο σε πραγματικό χρόνο και η εναπομένουσα δυσκολία είναι η πρόβλεψη του βάθους του δέντρου, καθώς εξαρτάται απόλυτα από το είδος της εφαρμογής. Για παράδειγμα, αν η επιφάνεια είναι μια σημαία την οποία προσπερνάμε με μεγάλη ταχύτητα, χρήσιμα θεωρούνται τα δυο κορυφαία επίπεδα. Από την άλλη θα μπορούσε να αποτελεί ένα αντικείμενο του οποίου θα απαιτούνταν λεπτομερής εξέταση από τον χρήστη, όπως το τμήμα ενός χάρτη στοιχείων σε ένα παιχνίδι δράσης.

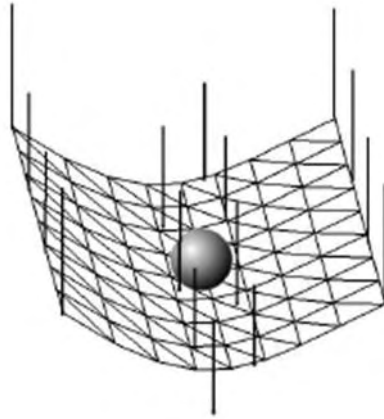
Στην τελευταία περίπτωση είναι συνήθως επιθυμητό να διαχωριστούν οι κόμβοι με τέτοιο τρόπο ώστε πολύγωνα της ίδιας ομάδας να ενταχθούν στη συνέχεια στο ίδιο επίπεδο ή πλαίσιο, προκειμένου να γίνει χρήση εξετασμένων αλγόριθμων τομών που επιλύουν γρήγορα.

Η χρήση προσαρμοστικών υποδιαιρέσεων εφαρμοσμένων στην εκδοχή της επιφάνειας με την μικρότερη ανάλυση κάθε άλλο παρά περιττή γνώση θα μπορούσε να χαρακτηριστεί. Χρησιμοποιώντας σωστά την εν λόγω τεχνική σε συνδυασμό με κριτήρια εύρεσης επιπέδων επιφανειών, θα μπορούσαμε να ελαχιστοποιήσουμε τις ανάγκες σε υπολογιστική δύναμη και το πλεόνασμα να χρησιμοποιηθεί σε άλλους πιο απαιτητικούς τομείς. Καθώς πάνω από την μισή εργασία των σχεδιαστών παιχνιδιών καταλαμβάνει η εύρεση έξυπνων λύσεων (tricks and cheats) για καλύτερη απόδοση, συχνά θυσιάζουμε λίγη ακρίβεια για λιγότερους υπολογισμούς συγκρούσεων, ιδιαίτερα σε μια περίπτωση σαν την τελευταία.

Η προαναφερθείσα τεχνική με το δέντρο είναι εφαρμόσιμη και σε καμπύλες, όπου δημιουργούμε όρια χρήσει πλαισίων για να απομονώσουμε όλο και μικρότερα τμήματα της καμπύλης.

4.11.2 Soft objects

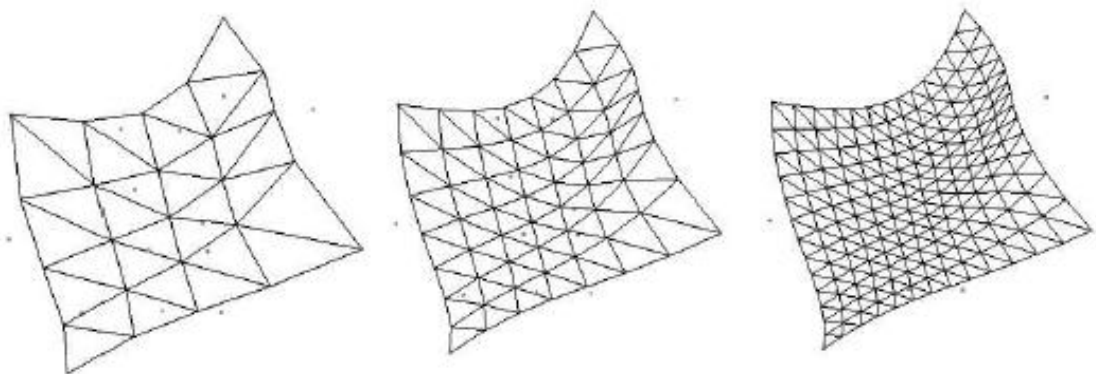
Ένα θέμα πρόκληση για τον τομέα της φυσικής και των γραφικών υπολογιστών είναι ο σχεδιασμός ενός μοντέλου για την όσο το δυνατόν πιο πιστή αναπαράσταση ρούχων και άλλων παραμορφώσιμων μαλακών αντικειμένων. Ένα καλό αποτέλεσμα επιτυγχάνεται τοποθετώντας στην επιφάνεια έναν αριθμό τμημάτων για τον υπολογισμό των δυνάμεων που αναπτύσσονται. Οι δυνάμεις αυτές μπορούν να εφαρμοστούν και στα σημεία ελέγχου της επιφάνειας. Η ακόλουθη εικόνα (Εικόνα 4.12) αναπαριστά μια επιφάνεια με συνδεδεμένα τμήματα σε κάθε σημείο ελέγχου της.



Εικόνα 4.12 Επιφάνεια Βέζιερ με τμήματα δυνάμεων

4.11.3 Αυτόματη επιλογή ανάλυσης (LOD)

Το *LOD* (Level of Detail) καθορίζει το μέγεθος της λεπτομέρειας στην αναπαράσταση κάποιου αντικειμένου. Μια συχνά χρησιμοποιημένη τεχνική, που εφαρμόζουν πολλοί σχεδιαστές, είναι η προσφορά πολλαπλών εκδοχών του ίδιου αντικειμένου εξ'αρχής. Το εν λόγω αντικείμενο θα μπορούσε να είναι μια φωτογραφία, ένα μοντέλο, μια επιφάνεια ή οτιδήποτε άλλο με πολυγωνική αναπαράσταση. Καθώς η κάμερα πλησιάζει το αντικείμενο, γίνεται εναλλαγή μεταξύ των διαφορετικών επιπέδων ανάλυσης *LOD*. Μια απεικόνιση αντικειμένου σε υποτιθέμενη κοντινή απόσταση θα μπορούσε να περιέχει κατά προσέγγιση 500 στον αριθμό πολύγωνα, ενώ μια μακρινή αναπαράσταση του ίδιου αντικειμένου να κάνει χρήση μόλις 5 στον αριθμό πολύγωνων. Παρατίθεται η ακόλουθη εικόνα (Εικόνα 4.13), από όπου προκύπτει και η ομοιότητα με την τεχνική της ενότητας 4.11.1



Εικόνα 4.13 Βέζιερ ένθετο (patch) με διαφορετικό *LOD*

Υπάρχουν πολλά θέματα για μελέτη σχετικά με τους *LOD* υπολογισμούς, και αφορούν στην επιλογή του τρόπου μετάβασης μεταξύ δυο παρεχόμενων αναλύσεων του αντικειμένου. Το κύριο ζήτημα είναι η αναγνώριση της δύναμης της παραμετρικής αναπαράστασης. Ακόμη όμως και στην περίπτωση που η εν λόγω προσέγγιση δεν είναι διαθέσιμη, υπάρχουν πολλές τεχνικές γνωστές σαν *imploding*, στις οποίες μειώνεται ο αριθμός των συμμετεχόντων πολυγώνων κάνοντας χρήση ορισμένων κανόνων σε σχέση με την απόσταση του θεατή.

4.11.4 Μοντελοποίηση αντικειμένων

Ένα ακόμη θέμα που θα έπρεπε να αναφερθεί είναι η δυνατότητα δημιουργίας αντικειμένων στον περιβάλλοντα χώρο παιχνιδιών, που αποτελεί μια κοινή εφαρμογή. Στην πλειοψηφία των περιπτώσεων γίνεται χρήση λογισμικού σχεδίασης 3D, χωρίς να απαξιώνεται η χρήση υπολογιστικών εφαρμογών αντίστοιχα.

Ένα διάσημο παράδειγμα της δύναμης των παραμετρικών επιφανειών είναι η “Utah teapot”, που όπως περιγράφει το όνομα, είναι ένα εικονικό μοντέλο μιας τσαγιέρας και αποτέλεσε ένα από τα πρώτα πραγματικά αντικείμενα που αναπαραστάθηκαν χρήσει ενθέτων επιφανειών (surface patches). Χρησιμοποιώντας κάποιες Βέζιερ καμπύλες έχουμε την δυνατότητα να αναδημιουργήσουμε μια πολύ ρεαλιστική τσαγιέρα.

Θεωρήστε την βασική μέθοδο πολυγώνων, όπου απαιτούνται τουλάχιστον 200 στον αριθμό σημεία προσδιορισμού, ενώ στην περίπτωση της παραμετρικής μεθόδου το ίδιο αποτέλεσμα επιτυγχάνεται με σαφώς λιγότερα σημεία (περίπου 50 στον αριθμό). Ας συμπεριλάβουμε στα άνωθεν την ιδιότητα που μας επιτρέπει την αναπαράσταση σε διαφορετική ανάλυση με μοναδική αλλαγή εκείνη στο διάστημα υπολογισμού.

4.12 **Υλοποίηση**

4.12.1 Απλός υπολογισμός

Μέχρι τώρα έχουμε περιγράψει τον υπολογισμό καμπυλών και επιφανειών σαν μια διαδικασία στην οποία θέτουμε διάστημα Δu και ακολούθως επαναυπολογίζουμε την καμπύλη. Παρά το κοινό της προσέγγισης μπορεί να μην αποδειχθεί αποτελεσματική καθώς το σχήμα της καμπύλης ποικίλει ανάλογα με το διάστημα που επιλέγεται. Μπορεί να είναι εντελώς επίπεδη, και να παρουσιάζει μια απότομη κορυφή σε κάποιο σημείο, οδηγώντας σε σπατάλη του επεξεργαστή.

4.12.2 Forward Difference (Προς τα εμπρός διαφορές)

Πολλοί επιστήμονες στο χώρο των υπολογιστών αλλά και των μαθηματικών έχουν έρθει αντιμέτωποι με την μέθοδο των προς τα εμπρός διαφορών, καθώς αποτελεί ένα τρόπο προσαρμογής του Δu σύμφωνα με καθορισμένα κριτήρια – σε αυτή την περίπτωση του πόσο επίπεδη είναι κάποια επιφάνεια.

Η διαφορά προς τα εμπρός $\Delta f(t)$ της συνάρτησης $f(t)$ δίνεται από την ακόλουθη σχέση

$$\Delta f(t) = f(t + \delta) - f(t) \quad , \quad (\Sigma\chi. 4.8)$$

που μπορεί να γραφεί και ως

$$f_{n+1} = f_n + \Delta f_n \quad (\Sigma\chi.4.9)$$

όπου η f υπολογίζεται σε ομοιόμορφο διάστημα μεγέθους δ . Για ένα κυβικό πολυώνυμο, που αποτελεί τον πιο κοινό βαθμό για παραμετρικές καμπύλες και επιφάνειες, έχουμε

$$f(t) = at^3 + bt^2 + ct + d \quad (\Sigma\chi. 4.10)$$

οπότε η προς τα εμπρός διαφορά υπολογίζεται ως εξής

$$\Delta f(t) = a(t + \delta)^3 + b(t + \delta)^2 + c(t + \delta) + d - (at^3 + bt^2 + ct + d) \quad (\Sigma\chi. 4.11)$$

Προκειμένου να διευκολύνουμε τον υπολογισμό εφαρμόζουμε την ίδια τεχνική στο $\Delta f(t)$. Οι σχέσεις (Σχ.4.9) και (Σχ. 4.11) μετασχηματίζονται σε

$$\Delta^2 f(t) = \Delta f(t + \delta) - \Delta f(t) \Rightarrow$$

$$\Delta^2 f(t) = 6a\delta^2 + 6a\delta^3 + sb\delta^2$$

από το οποίο προκύπτει

$$\Delta^2 f_n = \Delta f_{n+1} - \Delta f_n \quad \text{ή αλλιώς} \quad \Delta f_n = \Delta f_{n-1} + \Delta^2 f_{n-1} \quad (\Sigma\chi 4.12)$$

Δεδομένου ότι ασχολούμαστε κατά κύριο λόγο με πολυώνυμα τρίτου βαθμού η ακόλουθη προς τα εμπρός διαφορά θα είναι σταθερή και υπολογίζεται όπως αναφέρθηκε πιο πάνω. Συγκεντρωτικά λοιπόν έχουμε

$$f_{n+1} = f_n + \Delta f_n$$

$$\Delta f_n = \Delta f_{n-1} + \Delta^2 f_{n-1}$$

$$\Delta^2 f_{n-1} = \Delta^2 f_{n-2} + 6a\delta^3$$

Είναι λογικό οι πιο πάνω υπολογισμοί να φαίνονται υπεραρκετοί όταν φορούν στον υπολογισμό ενός και μόνο γραμμικού τμήματος, όμως οι απαιτούμενοι υπολογισμοί για κάθε 3D σημείο έχουν μειωθεί από τις δέκα προσθέσεις και τους 9 πολλαπλασιασμούς σε μόλις εννέα προσθέσεις, εάν για παράδειγμα χρησιμοποιήσουμε καμπύλες Bézier.

4.12.3 Αναδρομική υποδιαίρεση (recursive subdivision)

Ένας τρόπος για να βελτιωθεί η αποτελεσματικότητα είναι η αναδρομική υποδιαίρεση (recursive subdivision). Η θεωρητική προσέγγιση έχει αναφερθεί στις προηγούμενες υποενότητες, ενώ για να γίνει καλύτερα αντιληπτή βοηθάει ο ακόλουθος ψευδοκώδικας

```
Draw(Curve c, Error e)
  if (Straight(c,e)) then DrawLine(curve)
  else  SubdivideCurve(c, leftCurve, rightCurve)
        draw(leftCurve,e)
        draw(rightCurve,e)
  endif
end Draw
```

Ο αλγόριθμος διαιρεί και βασιλεύει (Divide-and-Conquer) είναι απλός στην υλοποίηση και αναμένει όταν ένα διαφεμένο τμήμα καμπύλης (εάν κάποιο) είναι αρκετά επίπεδο για να σχεδιαστεί σαν ευθεία γραμμή. Θα μπορούσε να αφορά στην περιοχή που εσωκλείεται από τα ακραία σημεία μιας καμπύλης, η γωνία μεταξύ των ακραίων σημείων και ούτω καθ' εξής.

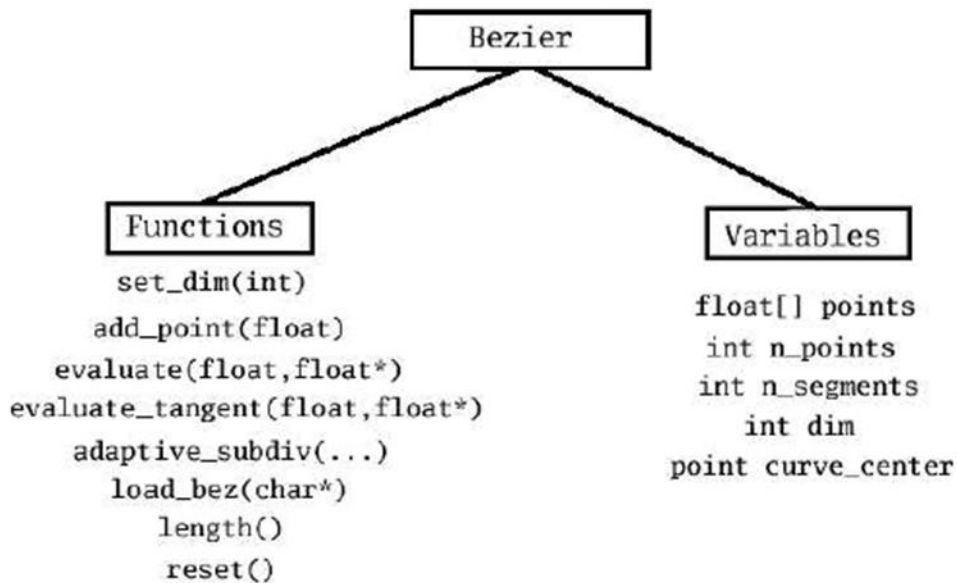
4.12.4 Βοηθητικοί Πίνακες (look-up tables)

Οι βοηθητικοί πίνακες χρησιμοποιούνται συχνά στην περίπτωση που έχουμε γνώση της υπολογιστικής ανάλυσης μιας συνάρτησης. Παραδείγματος χάριν, στο περιβάλλον παιχνιδιών χρησιμοποιούνται σε τριγωνομετρικές συναρτήσεις, όμως μπορούν να χρησιμοποιηθούν εξίσου αποτελεσματικά σε παραμετρικές επιφάνειες και καμπύλες. Για κάθε σημείο ελέγχου, δημιουργούμε ένα πίνακα του οποίου το περιεχόμενο είναι η τιμή της βασικής συνάρτησης σε κάθε διάστημα υπολογισμού. Στις περιπτώσεις που παρουσιάζουμε είτε τροποποιούμε μια επιφάνεια σε πραγματικό χρόνο, δεν είμαστε υποχρεωμένοι να κάνουμε ξανά χρήση των περίπλοκων συναρτήσεων βαρών.

Βέβαια οι πίνακες μπορεί να αποκτήσουν αρκετά μεγάλο μέγεθος, επομένως το τίμημα για την προαναφερθείσα διευκόλυνση είναι ο απαιτούμενος αποθηκευτικός χώρος. Συνήθως το κέρδος σε χρόνο επεξεργαστή είναι μεγαλύτερης βαρύτητας σε σχέση με το αντίστοιχο κέρδος σε απαιτούμενη μνήμη, και για αυτό το λόγο γίνεται συχνή χρήση της εν λόγω μεθόδου.

4.12.5 Φορητότητα (*portability*)

Ένα από τα πιο σημαντικά χαρακτηριστικά των παραμετρικών καμπυλών, το οποίο καλούνται να ικανοποιήσουν οι αλγόριθμοι που τις μεταχειρίζονται, είναι η φορητότητα. Έχοντας αυτό το στοιχείο στο μυαλό μας και συνδυάζοντας το με μια αντικειμενο-κεντρική προσέγγιση, προκύπτει μια τάξη (class) που θα μπορούσε να θεωρηθεί αρκετά γενική. Ας λάβουμε υπόψη το ακόλουθο διάγραμμα τάξεων(class) - κοινός τρόπος αναπαράστασης ιεραρχίας αντικειμένων/τάξεων και περιεχομένου - που βασίζεται στη μηχανή γραφικών Fly3D (Εικόνα 4.14).

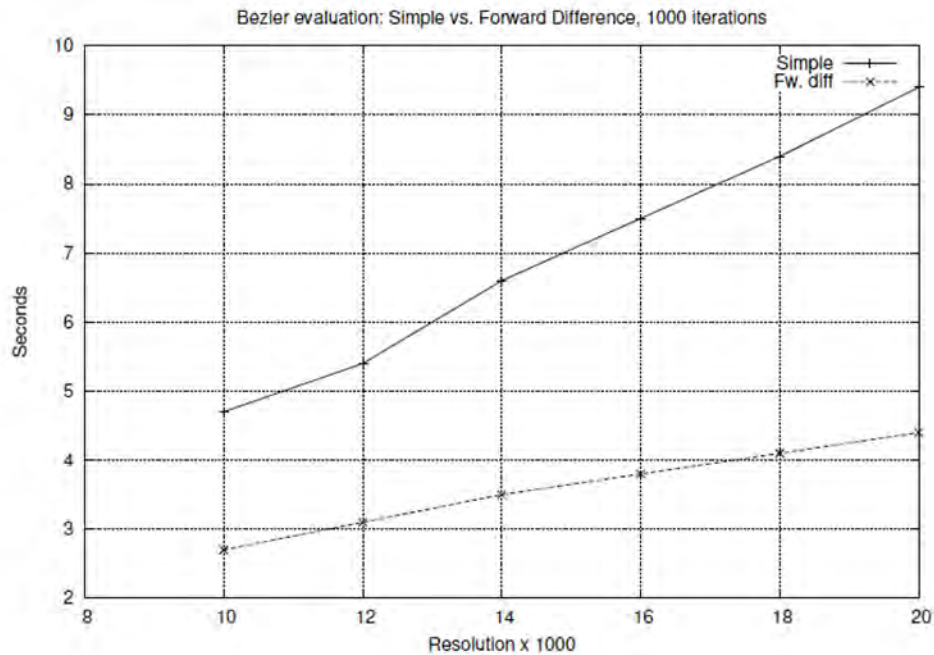


Εικόνα 4.14 Διάγραμμα τάξεων Bézier

Πρόκειται για μια βασική τάξη που διατηρεί τις απαραίτητες πληροφορίες για τους αλγόριθμους και τις συναρτήσεις που χρειάζονται για τον υπολογισμό μιας καμπύλης Bézier. Όλες οι αναφερθείσες συναρτήσεις διατηρούν ανεξαρτησία εκτέλεσης σε σχέση με την διαθέσιμη κάθε φορά πλατφόρμα υπολογισμών, με μικρή εξαίρεση τη `load_bez` καθώς η ανάγνωση δεδομένων μπορεί να διαφέρει σε κάποιες πλατφόρμες. Τον ρόλο της κάθε συνάρτησης αλλά και της κάθε μεταβλητής δεν είναι πολύ δύσκολο να τα αντιληφθεί κανείς. Το μόνο που χρειάζεται από εδώ και πέρα είναι μια σχεδιαστική συνάρτηση, η οποία ποικίλει μεταξύ των διαθέσιμων εργαλείων εργασίας.

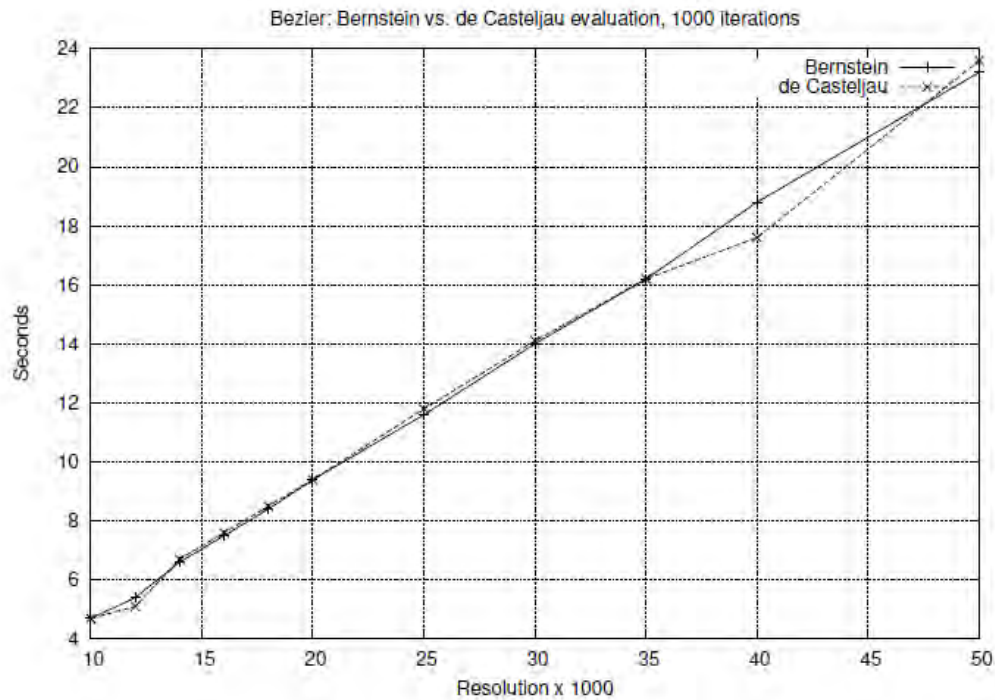
4.13 Συγκρίσεις

Παρατίθενται μερικές συγκρίσεις μεταξύ των διαφορετικών υλοποιήσεων και η κλιμάκωση που παρουσιάζουν. Ζητούμενο είναι ο υπολογισμός μιας τυπικής καμπύλης για διαφορετικές αναλύσεις. Οι μετρήσεις έγιναν με ρολόι συστήματος – το οποίο δεν επεξεργάζεται μοναδική διεργασία- και οι πίνακες συμπληρώθηκαν με τους μέσους όρους τριών διαδορικών δοκιμών.



Εικόνα 4.15 Απλός υπολογισμός vs. Προς τα εμπρός διαφορές

Όπως είχε αναφερθεί στην αντίστοιχη υποενότητα, η μέθοδος των προς τα εμπρός διαφορών ελάττωσε κατά σημαντικό ποσοστό το πλήθος των απαιτούμενων πράξεων για τον υπολογισμό μιας καμπύλης και η διαφορά φαίνεται καθαρά στο πιο πάνω διάγραμμα (Εικόνα 4.15).



Εικόνα 4.16 Πολυώνυμα Bernstein vs. de Casteljau

Η μόνη διαφορά μεταξύ των δυο καταλήγει να είναι η θεωρητική σκοπιά: ο αλγόριθμος de Casteljau αναπαρίσταται γεωμετρικά ενώ τα πολυώνυμα Bernstein τα μεταχειριζόμαστε πιο αναλυτικά. Περιέχουν τον ίδιο αριθμό πολλαπλασιασμών, υπολογίζουν τα ίδια σημεία, όμως η ιστορία τους είναι διαφορετική. Καταλήγουμε μόνο σε μικρές τοπικές διαφορές, όπως φαίνεται και στην πιο πάνω εικόνα (Εικόνα 4.16), και στο συμπέρασμα πως η επιλογή μεταξύ των δυο εξαρτάται από την ευελιξία στην υλοποίηση του καθενός.

5. Αναπαραστάσεις χρήσει του υπολογιστικού προγράμματος Mathematica

Μαθηματική έκφραση	Κώδικας στο Mathematica
$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$ <p>Bernstein polynomial</p>	<pre>Bernstein[i_,n_,t_]:= Binomial[n,i] t^i (1-t)^(n-i)</pre>
$B(t) = \sum_{i=0}^n P_i B_{i,n}(t)$ <p>Bézier curve</p>	<pre>BezierCurve[pts:{{_,_}..} {{_,_}..},t_] := Module[{n=Length[pts]-1}, Simplify[Table[Bernstein[i,n,t],{i,0,n}].pts]]</pre>
$S(s,t) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_{i,m}(t)B_{j,n}(t)$ <p>Bézier surface</p>	<pre>BezierSurface[pts:{{{_,_,_}..}..},{s_,t_}]:= Module[{m=Length[pts]-1, n=Length[First[pts]]-1,U,V},{U,V} =MapThread[Table[Bernstein[i,#1,#2], {i,0,#1}]&{{m,n},{s,t}}; Plus@@@ (U.pts*V) //Simplify]</pre>
$B(t) = \frac{\sum_{i=0}^n w_i P_i B_{i,n}(t)}{\sum_{i=0}^n w_i B_{i,n}(t)}$ <p>Rational Bézier curve</p>	<pre>RationalBezierCurve[pts:{{_,_}..} {{_,_}..}, weights List,t_]:= Module[{n=Length[pts]-1,lisfun}, If[Length[pts] == Length[weights], lisfun=Table[Bernstein[i,n,t],{i,0,n}; Simplify[(Plus @@@ (pts*weights*lisfun)) / (lisfun.weights)],Message[RationalBezierCurve::badnum]]]]</pre>
$S(s,t) = \frac{\sum_{i=0}^m \sum_{j=0}^n w_{ij} P_{ij} B_{i,m}(t)B_{j,n}(t)}{\sum_{i=0}^m \sum_{j=0}^n w_{ij} B_{i,m}(t)B_{j,n}(t)}$ <p>Rational Bézier surface</p>	<pre>RationalBezierSurface[pts:{{{_,_,_}..}..}, weights ?MatrixQ,{s_,t_}]:= Module[{m=Length[pts]-1, n=Length[First[pts]]-1,U,V}, If[Take[Dimensions[pts],2]== Take[Dimensions[weights],2], {U,V}=MapThread[Table [Bernstein[i,#1,#2],{i,0,#1}]&{{m,n},{s,t}}; Plus@@@ (U.(pts*weights)*V)/(Plus@@@ (U.weights*V))//Simplify,Message[RationalBezierSurface::badnum]]]]</pre>

Θεωρείται αυτονόητο πως οι πιο πάνω ορισμοί αναφέρονται για εγκυκλοπαιδικούς λόγους και φυσικά δεν απαιτείται η γνώση τους για τον χειρισμό του Mathematica. Έχουν σχεδιαστεί κατάλληλα όλες οι συναρτήσεις ώστε με τη προσθήκη μόνο των επιθυμητών ορισμάτων να μπορεί ο εκάστωτε χρήστης να επωφεληθεί αυτών και να ολοκληρώσει γρήγορα την εργασία του. Ακολουθώς παρατίθενται εντολές του Mathematica συνοδευόμενες από τα αντίστοιχα αποτελέσματα.

Για τα **Bernstein πολυώνυμα** έχει ορισθεί η ακόλουθη συνάρτηση

`BernsteinBasis[d,n,x]` που αναπαριστά την n-οστή βασική συνάρτηση βαθμού d στο σημείο x. Επομένως έχουμε

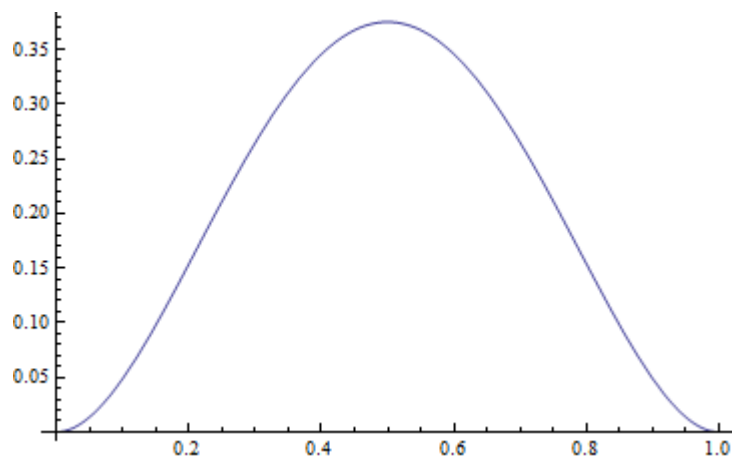
```
In[1] := BernsteinBasis[4,4,0.75]
```

```
Out[1] := 0.316406
```

Για την 2^η βασική συνάρτηση των Bernstein πολυωνύμων 4^{ου} βαθμού στο διάστημα [0,1] εκτελούμε την ακόλουθη εντολή και το αποτέλεσμα εμφανίζεται στην (Εικόνα 5.1)

```
In[2] := Plot[BernsteinBasis[4,2,x],{x,0,1}]
```

```
Out[2] :=
```



Εικόνα 5.1 Αναπαράσταση της $B_{1,4}(t)$ στο $[0,1]$

Η συνάρτηση που χρησιμοποιήθηκε παραμετρικά για να δημιουργηθούν όλες οι εικόνες της Ενότητας 2 παρατίθεται ακολούθως

```
In[3]:=
```

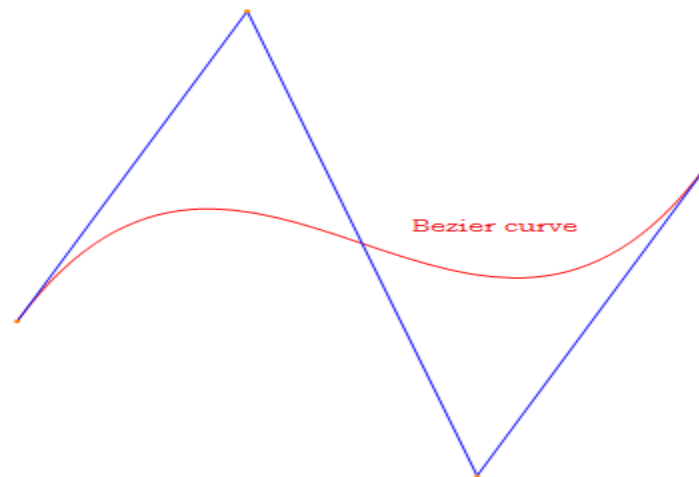
```
Plot[Evaluate[Table[BernsteinBasis[3,i,t],{i,0,3}],{t,0,1}]
```

Για τις **καμπύλες Bézier** έχει οριστεί η ακόλουθη συνάρτηση

`BezierCurve[{pt1,pt2,...}]` η οποία αναπαριστά μια Bézier καμπύλη με σημεία ελέγχου τα pt_i . Επομένως για την αναπαράσταση Bézier καμπύλης σημείων ελέγχου $\{0,0\},\{1,2\},\{2,-1\},\{3,1\}$, έχουμε το αποτέλεσμα της (Εικόνας 5.2)

```
In[4] := pts={{0,0},{1,2},{2,-1},{3,1}}
Graphics[{Red,BezierCurve[pts],Blue,Line[pts],Orange,Point[pts]}]
```

Out[4] :=



Εικόνα 5.2 Bézier curve

```
In[5] := Graphics[Table[
{Hue[RandomReal[]],BezierCurve[RandomReal[1,{4,2}]]},{8}]]
```

Με την πιο πάνω εντολή δημιουργούνται 8 στον αριθμό καμπύλες Bézier, με την `RandomReal` να παράγει τυχαία 4 σημεία ελέγχου δυο διαστάσεων για όλες τις καμπύλες. Το αποτέλεσμα παρουσιάζεται στην ακόλουθη (Εικόνα 5.3)

Out[5] :=

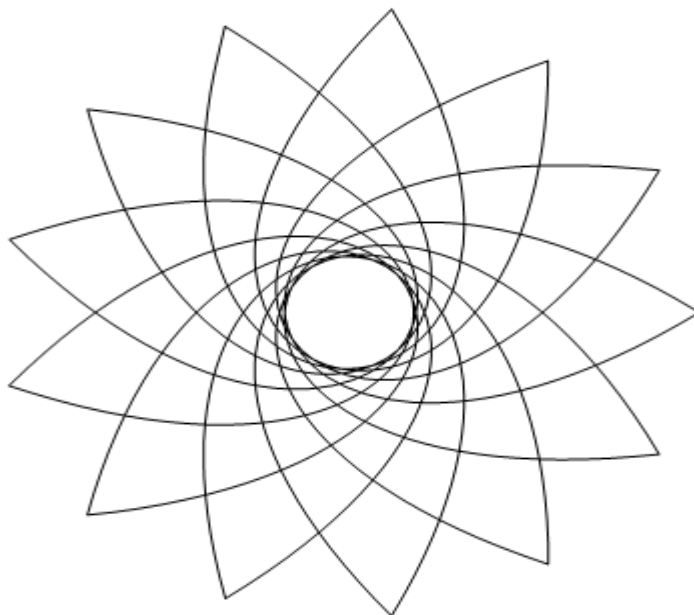


Εικόνα 5.3 Τυχαίες καμπύλες Bezier

Μια μικρή απόδειξη της “καλλιτεχνικής” φύσης των καμπυλών Bezier είναι το αποτέλεσμα της αναπαράστασης που ακολουθεί (Εικόνα 5.4)

In[6] := **Graphics[BezierCurve[Table[{Cos[2k Pi/13],Sin[2 k Pi/13]},
{k,0,156,4}]]]**

Out[6] :=

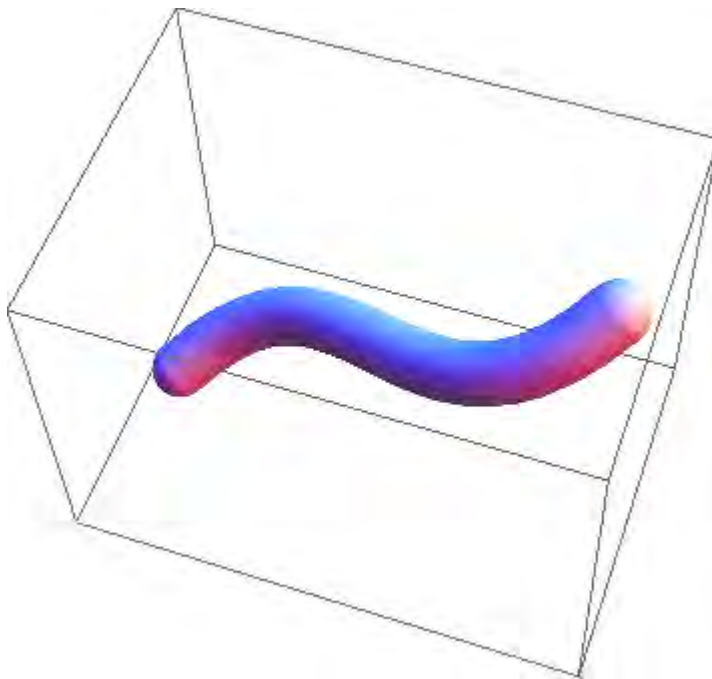


Εικόνα 5.4 Καμπύλες Bezier

Ένα τελευταίο παράδειγμα για τον τρισδιάστατο χώρο, στον οποίο δεν έχουμε κάνει κάποια αναφορά έως τώρα. Η κατασκευή ενός σωλήνα χρήσει καμπυλών Bezier ο οποίος αναπαρίσταται στην (Εικόνα 5.5)

```
In[7] := pts = {{0,0,0},{1,1,1},{2,-1,1},{3,0,2}}  
Graphics3D[Tube[BezierCurve[pts],0.2]]
```

```
Out[7] :=
```



Εικόνα 5.5 Bezier σωλήνας

Βιβλιογραφία

- [1] *Bernstein Polynomials*
Kenneth I. Joy, Department of Computer Science - University of California, USA
- [2] *A History of Curves and Surfaces in CAGD*
Gerald Farin, Computer Science and Engineering – University of Arizona, USA
- [3] *Bézier and B-spline Technology*
Fredrik Andersson
- [4] *Computer Aided Geometric Design and Computer Graphics: High Level computation programs (Mathematica and Matlab) in CAGD*
Andres Iglesias, Department of Applied Mathematics and Computational Sciences – University of Cantabria, Spain
- [5] *How Symbolic Computation can benefit Computer Aided Geometric Design*
Andres Iglesias, Department of Applied Mathematics and Computational Sciences – University of Cantabria, Spain
- [6] *Computer Aided Geometric Design*
Thomas W. Sederberg
- [7] *The Basic Curves and Surfaces of CAGD*
Colm Mulcahy, Department of Mathematics – Spelman College Atlanta, USA
- [8] *Generalized Bernstein Polynomials and Bézier Curves : An application of umbral calculus to CAGD*
Rudolf Winkel, Institut für Reine und Angewandte Mathematik, RWTH - Aachen, Germany
- [9] *CAGD : History and Outlook*
Gerald Farin, Computer Science and Engineering – University of Arizona, USA
- [10] *The de Casteljau Algorithm for Evaluating Bézier curves*
From Rockwood “Interactive Curves and Surfaces”
JDill de Casteljau.doc

Ιστοσελίδες

<http://www.wikipedia.org>
<http://www.wolfram.com>