

# ALGORITMA SUBBAHAGIAN DALAM CAGD

oleh

WAN KHAIRIYAH HULAINI BINTI WAN RAMLI

Projek diserahkan untuk memenuhi  
sebahagian keperluan bagi  
Ijazah Sarjana Sains Matematik Pengajaran

Mei 2009

## PENGHARGAAN

Syukur ke hadrat Ilahi kerana dengan limpah kurnia dan keizinannya, saya dapat menyiapkan projek ini. Jutaan terima kasih diucapkan kepada penyelia saya, Prof Madya Mohd Nain Bin Haji Awang atas bimbingan dan nasihat beliau membantu saya menyiapkan projek ini. Terima kasih juga kepada Pusat Pengajian Sains Matematik yang memberi peluang kepada saya untuk melanjutkan pelajaran di peringkat Ijazah Sarjana Sains Matematik Pengajaran.

Seterusnya, saya juga ingin mengucapkan terima kasih kepada suami, Mohamed Affiz Bin Rahimi, ibu saya Wan Nadzirah Binti Wan Daud serta keluarga saya yang sentiasa memberi dorongan kepada saya untuk tidak berputus asa dalam usaha saya mengikuti kursus Sarjana Sains Matematik Pengajaran ini. Terima kasih kepada pensyarah UiTM, Prof Madya Dr Jusoh Yacob yang sentiasa memberi nasihat yang berguna untuk saya. Tidak lupa juga ucapan terima kasih buat teman-teman sekursus iaitu Norlaila, Maziah dan Che Hayati yang tidak jemu-jemu memberi semangat untuk terus berusaha bersama-sama. Semoga kejayaan menjadi milik kita.

## ISI KANDUNGAN

PENGHARGAAN		ii
ISI KANDUNGAN		iii
SENARAI RAJAH		v
ABSTRAK		viii
ABSTRACT		ix
BAB 1	Pengenalan	1
	1.1 Objektif Projek	4
BAB 2	Lengkung Bezier	5
	2.1 Polinomial Bernstein	5
	2.2 Definisi Lengkung Bezier	6
	2.3 Ciri-ciri Lengkung Bezier	7
	2.4 Contoh-contoh Lengkung Bezier	7
	2.5 Keselajaran Geometri Dan Keselajaran Parameter	10
BAB 3	Penghasilan Lengkung Bezier Dengan Menggunakan Algoritma de Casteljau	12
	3.1 Penghasilan Lengkung Kuadratik	12
	3.2 Penghasilan Lengkung Kubik	18

BAB 4	PENGHASILAN LENGKUNG DENGAN MENGUNAKAN ALGORITMA CHAIKIN	25
4.1	Penghasilan Lengkung Kubik	25
4.2	Penghasilan Huruf S Dan Huruf Bersambung	33
BAB 5	PERMUKAAN	45
5.1	Hasil Darab Tensor Untuk Tampilan Permukaan	45
5.2	Subbahagian Permukaan	47
5.3	Permukaan Tertutup	53
BAB 6	KESIMPULAN	57
RUJUKAN		60

## SENARAI RAJAH

Rajah 2.1	: Lengkung Kuadratik Dengan Poligon Kawalan	8
Rajah 2.2	: Polinomial Bernstein Darjah 3	9
Rajah 2.3	: Lengkung Kubik Dengan Poligon Kawalan	10
Rajah 3.1(a)	: Poligon Kawalan Untuk Lengkung Kuadratik	12
Rajah 3.1(b)	: Poligon Kawalan Untuk Lengkung Kuadratik	13
Rajah 3.1(c)	: Poligon Kawalan Untuk Lengkung Kuadratik	14
Rajah 3.2	: Lengkung Bezier Kuadratik Dengan $t = 0.2$	16
Rajah 3.3	: Lengkung Bezier Kuadratik Dengan $t = 0.5$	17
Rajah 3.4	: Lengkung Bezier Kuadratik Dengan $t = 0.7$	17
Rajah 3.5(a)	: Poligon Kawalan Untuk Lengkung Kubik	18
Rajah 3.5(b)	: Poligon Kawalan Untuk Lengkung Kubik	19
Rajah 3.5(c)	: Poligon Kawalan Untuk Lengkung Kubik	19
Rajah 3.5(d)	: Poligon Kawalan Untuk Lengkung Kubik	20
Rajah 3.6	: Lengkung Bezier Kubik Dengan $t = 0.2$	23
Rajah 3.7	: Lengkung Bezier Kubik Dengan $t = 0.5$	23
Rajah 3.8	: Lengkung Bezier Kubik Dengan $t = 0.7$	24
Rajah 4.1(a)	: Poligon Kawalan Untuk Lengkung Kubik	26
Rajah 4.1(b)	: Poligon Kawalan Untuk Lengkung Kubik	27
Rajah 4.1(c)	: Poligon Kawalan Untuk Lengkung Kubik	28
Rajah 4.1(d)	: Poligon Kawalan Untuk Lengkung Kubik	29
Rajah 4.1(e)	: Poligon Kawalan Selepas Lelaran Pertama	30

Rajah 4.2	: Lengkung Kubik Selepas Lelaran Pertama	30
Rajah 4.3	: Lengkung Kubik Selepas Lelaran Kedua	31
Rajah 4.4	: Lengkung Kubik Selepas Lelaran Ketiga	32
Rajah 4.5	: Lengkung Kubik Selepas Lelaran Keempat	32
Rajah 4.6(a)	: Poligon Kawalan Huruf S	33
Rajah 4.6(b)	: Poligon Kawalan Huruf S	35
Rajah 4.6(c)	: Poligon Kawalan Huruf S	36
Rajah 4.6(d)	: Poligon Kawalan Huruf S	38
Rajah 4.6(e)	: Poligon Kawalan Selepas Lelaran Pertama	39
Rajah 4.7	: Huruf S Selepas Lelaran Pertama	40
Rajah 4.8	: Huruf S Selepas Lelaran Kedua	40
Rajah 4.9	: Huruf S Selepas Lelaran Ketiga	41
Rajah 4.10	: Huruf S	41
Rajah 4.11	: Titik Kawalan Untuk Penghasilan Huruf Bersambung	42
Rajah 4.12	: Huruf Bersambung	43
Rajah 4.13	: Titik Kawalan Untuk Penghasilan Huruf Bersambung	43
Rajah 4.14	: Huruf Bersambung	44
Rajah 5.1	: Kedudukan Titik Kawalan Permukaan Bezier Bikuadratik	46
Rajah 5.2	: Kedudukan Titik Kawalan Permukaan Bezier Bikubik	47
Rajah 5.3	: Jaring Kawalan Permukaan Bikuadratik	48
Rajah 5.4	: Jaring Kawalan Permukaan Bikuadratik Selepas Satu Lelaran $u$	48
Rajah 5.5	: Jaring Kawalan Permukaan Bikuadratik Selepas Satu Lelaran $v$	49
Rajah 5.6	: Permukaan Bikuadratik Selepas Satu Lelaran	50
Rajah 5.7	: Jaring Kawalan Permukaan Bikuadratik	51
Rajah 5.8	: Permukaan Bikuadratik Selepas Satu Lelaran	51

Rajah 5.9	: Permukaan Bikuadratik Selepas Dua Lelaran	52
Rajah 5.10	: Permukaan Bikuadratik Selepas Tiga Lelaran	52
Rajah 5.11	: Permukaan Bikuadratik Selepas Empat Lelaran	53
Rajah 5.12	: Jaring Kawalan Permukaan Pasu	54
Rajah 5.13	: Permukaan Pasu Selepas Satu Lelaran	55
Rajah 5.14	: Permukaan Pasu Selepas Dua Lelaran	55
Rajah 5.15	: Permukaan Pasu Selepas Tiga Lelaran	56

## ABSTRAK

CAGD adalah singkatan untuk “Computer Aided Geometric Design” atau di dalam Bahasa Melayu ialah Rekabentuk Geometri Bantuan Komputer iaitu RGBK. Membina dan mengawal lengkung menjadi keutamaan di dalam rekabentuk berbantuan komputer. Teknik penjanaan lengkung dan permukaan menggunakan polinomial Bernstein dalam CAGD ini mula diperkenalkan oleh Paul de Casteljaou dan Pierre Bezier. Projek ini membincangkan dua algoritma subbahagian yang digunakan dalam proses untuk menghasilkan lengkung dan permukaan iaitu algoritma de Casteljaou dan algoritma Chaikin. Algoritma de Casteljaou menghasilkan lengkung dengan cara menginterpolasi titik kawalan pertama dan titik kawalan terakhir manakala algoritma Chaikin menjana lengkung dengan cara memotong setiap penjuru poligon kawalan asal. Lengkung dan permukaan Bezier kini digunakan secara meluas sebagai asas matematik dalam sistem CAD dan menjadi alat utama dalam perkembangan kaedah-kaedah baru untuk keperluan lengkung dan permukaan dalam CAD.



## **SUBDIVISION ALGORITHM IN CAGD**

### **ABSTRACT**

CAGD is the abbreviation for the Computer Aided Geometric Design or in Malay Language, it is called “Rekabentuk Geometri Bantuan Komputer” or RGBK. Generating and controlling curves are important in computer aided design. The techniques for generating the curves and surfaces in CAGD were developed by Paul de Casteljau and Pierre Bezier. This project will discuss about 2 subdivision algorithms used in the process of generating curves and surfaces, that are de Casteljau algorithm and Chaikin algorithm. De Casteljau algorithm generates curves by interpolating the first control point and last control point. Chaikin algorithm generates curves by cutting every corner of original control polygon. The Bezier curves and surfaces nowadays are established as the mathematical basis of many CAD systems and have formed a major tool for the development of new methods for curves and surfaces descriptions.

# BAB 1

## PENGENALAN

Projek ini membincangkan algoritma subbahagian yang digunakan dalam proses untuk menghasilkan lengkung dan permukaan. Terdapat dua algoritma yang dibincangkan di dalam projek ini iaitu algoritma Chaikin dan algoritma de Casteljau. Setiap bab secara keseluruhannya menyentuh tentang penjanaan lengkung dan permukaan Bezier dengan menggunakan proses subbahagian.

Industri rekabentuk berasaskan sistem rekabentuk bantuan komputer (CAD) seperti yang terdapat dalam industri automobil, perkapalan, kapal terbang serta objek model paparan komputer memerlukan kaedah pameran rupabentuk atau corak objek yang lebih berkesan. Membina dan mengawal lengkung menjadi keutamaan di dalam rekabentuk berbantuan komputer. Antara perwakilan lengkung berparameter yang paling penting digunakan di dalam komputer grafik dan rekabentuk berbantuan komputer ialah lengkung Bezier [3].

Bab Dua membincangkan tentang lengkung Bezier. Pada tahun 1959, sebuah syarikat kenderaan di Perancis iaitu Citroen telah mengambil Paul de Faget de Casteljau untuk bekerja dengan syarikat tersebut. Casteljau telah menggunakan polinomial Bernstein dalam hasil kerjanya yang membawa kepada penghasilan algoritma de Casteljau. Walaubagaimanapun, hasil kerja Casteljau dirahsiakan oleh Citroen daripada umum. Casteljau berhenti dari Citroen pada 1989 dan menumpukan

perhatian dalam menerbitkan hasil kerjanya. Pada masa yang sama, pesaing Citroen iaitu Renault diterajui oleh Pierre Bezier. Hasil kerja Bezier menyamai Casteljau, hanya teknik matematik yang berbeza. Hasil kerja Bezier diterbitkan secara meluas dan dikenali sebagai Lengkung Bezier [8].

Bab ini juga membincangkan polinomial Bernstein yang digunakan untuk menghasilkan persamaan untuk lengkung dan permukaan Bezier dan ciri-ciri lengkung Bezier. Selain itu, bab ini juga menyentuh tentang keselajaran parameter dan keselajaran geometri.

Bab Tiga membincangkan kaedah untuk menghasilkan lengkung Bezier dengan menggunakan algoritma subbahagian. Algoritma ini dikenali sebagai algoritma de Casteljau. Lengkung yang dihasilkan hanya menginterpolasi titik kawalan pertama dan titik kawalan terakhir. Bab ini menunjukkan proses lelaran yang dilakukan dengan menggunakan algoritma ini untuk menghasilkan lengkung kuadratik dan lengkung kubik yang licin. Proses lelaran ini sangat penting kerana pada setiap lelaran, bentuk poligon kawalan yang lebih baik akan dihasilkan untuk menghampiri bentuk lengkung sebenar yang dikehendaki [1].

Bab Empat menyentuh tentang satu lagi kaedah penjanaan lengkung dengan menggunakan proses subbahagian iaitu algoritma Chaikin. Pada tahun 1974, George Chaikin telah memperkenalkan satu prosedur untuk menghasilkan lengkung berdasarkan titik-titik kawalan. Algoritma ini dianggap menarik kerana ia menggunakan kaedah pemotongan pepenjuru terhadap set titik kawalan atau poligon kawalan untuk menghasilkan lengkung. Pengkaji seperti Bezier menghasilkan

lengkung berdasarkan titik-titik kawalan yang memfokuskan kepada polinomial Bernstein. Chaikin pula mencipta algoritma yang berasaskan terus kepada poligon kawalan. Lengkung dijana berdasarkan pemotongan pepenjuru di mana poligon kawalan baru dihasilkan dengan cara memotong setiap penjuru poligon kawalan asal [9].

Kesinambungan daripada penghasilan lengkung ialah penghasilan permukaan. Topik ini dibincangkan dalam Bab Lima. Dalam bab ini, kita akan membincangkan hasil darab tensor yang digunakan untuk menghasilkan persamaan bagi permukaan dan contoh-contoh permukaan yang dijanakan menggunakan persamaan tersebut.

Seterusnya, bab ini juga membincangkan proses subbahagian dalam penghasilan permukaan dengan menggunakan algoritma Chaikin. Proses subbahagian untuk menghasilkan permukaan adalah sama dengan proses subbahagian untuk menghasilkan lengkung. Cuma untuk permukaan, subbahagian akan dilakukan pada dua arah iaitu  $u$  dan  $v$ . Contoh permukaan bikuadratik yang dijanakan dengan menggunakan algoritma Chaikin juga ditunjukkan dalam bab ini. Selain itu, bab ini juga menunjukkan contoh permukaan tertutup yang dihasilkan iaitu permukaan pasu. Kesimpulan keseluruhan projek ini akan dibincangkan dalam bab Enam.

## 1.1 Objektif Projek

Objektif saya menjalankan projek bertajuk Algoritma Subbahagian Dalam CAGD ini ialah:

- 1) Menjana lengkung Bezier dengan menggunakan algoritma de Casteljaou.
- 2) Menggunakan algoritma Chaikin untuk menghasilkan lengkung dan permukaan.
- 3) Menjana huruf bersambung dengan menggunakan algoritma Chaikin.
- 4) Menghasilkan objek dengan menggunakan algoritma Chaikin.

Penjanaan lengkung dan permukaan bagi projek ini menggunakan perisian Matlab 6.5 dan Mathematica 6.0.

## BAB 2

### LENGKUNG BEZIER

#### 2.1 Polinomial Bernstein

Polinomial Bernstein berdarjah  $n$  ditakrifkan sebagai :

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, 1, \dots, n, \quad 0 \leq t \leq 1 \quad (2.1)$$

di mana  $t$  ialah parameter. Oleh sebab

$$t^i (1-t)^{n-i} \geq 0, \quad 0 \leq t \leq 1,$$

maka persamaan (2.1) menunjukkan bahawa

$$B_i^n(t) \geq 0$$

Kita dapat buktikan

$$\sum_{i=0}^n B_i^n(t) = 1, \quad 0 \leq t \leq 1$$

seperti berikut:

$$\begin{aligned} \sum_{i=0}^n B_i^n(t) &= \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} \\ &= \binom{n}{0} t^0 (1-t)^n + \binom{n}{1} t^1 (1-t)^{n-1} + \binom{n}{2} t^2 (1-t)^{n-2} + \dots + \binom{n}{n} t^n (1-t)^0 \\ &= [(1-t) + t]^n \\ &= 1 \end{aligned}$$

Seterusnya, satu persamaan rekursi diperolehi iaitu

$$\begin{aligned}
 B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\
 &= \left[ \binom{n-1}{i} + \binom{n-1}{i-1} \right] t^i (1-t)^{n-i} \\
 &= \binom{n-1}{i} t^i (1-t)^{n-i} + \binom{n-1}{i-1} t^i (1-t)^{n-i} \\
 &= (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \quad 0 \leq t \leq 1
 \end{aligned}$$

## 2.2 Definisi Lengkung Bezier

Persamaan untuk lengkung Bezier  $P(t)$  iaitu hasil gabungan polinomial Bernstein dan titik kawalan ditakrifkan oleh persamaan (2.2) berikut:

$$P(t) = \sum_{i=0}^n v_i B_i^n(t) \quad , \quad 0 \leq t \leq 1 \quad (2.2)$$

di mana,  $v_i$  ialah titik kawalan dan  $B_i^n(t)$  ialah polinomial Bernstein,

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

dan

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

Di sini,  $n$  ialah darjah fungsi Bernstein.

## 2.3 Ciri-ciri Lengkung Bezier

Terdapat beberapa ciri-ciri lengkung Bezier iaitu :

- i. Darjah polinomial,  $n$  adalah kurang satu daripada bilangan titik kawalan yang ditakrifkan,  $n+1$ . Contohnya, lengkung kuadratik mempunyai tiga titik kawalan.
- ii. Lengkung yang dihasilkan bergantung kepada bentuk poligon kawalan yang ditentukan.
- iii. Lengkung hanya menginterpolasi titik kawalan pertama dan titik kawalan terakhir.
- iv. Lengkung hanya menghampiri poligon kawalan.

## 2.4 Contoh-contoh Lengkung Bezier

Merujuk persamaan (2.2), fungsi Bezier ialah

$$P(t) = \sum_{i=0}^n v_i B_i^n(t) \quad , \quad 0 \leq t \leq 1$$

Subtopik berikut menunjukkan contoh-contoh fungsi Bezier.

### 2.4.1 Bezier Kuadratik

Untuk lengkung kuadratik ( $n = 2$ ), bilangan titik kawalan adalah tiga.

Persamaan (2.2) akan menjadi

$$P(t) = \sum_{i=0}^2 v_i B_i^2(t)$$

$$P(t) = v_0 B_0^2(t) + v_1 B_1^2(t) + v_2 B_2^2(t)$$



Nilai bagi polinomial Bernstein  $B_i^2(t)$  :

$$i = 0 : B_0^2(t) = \binom{2}{0} t^0 (1-t)^2 = (1-t)^2$$

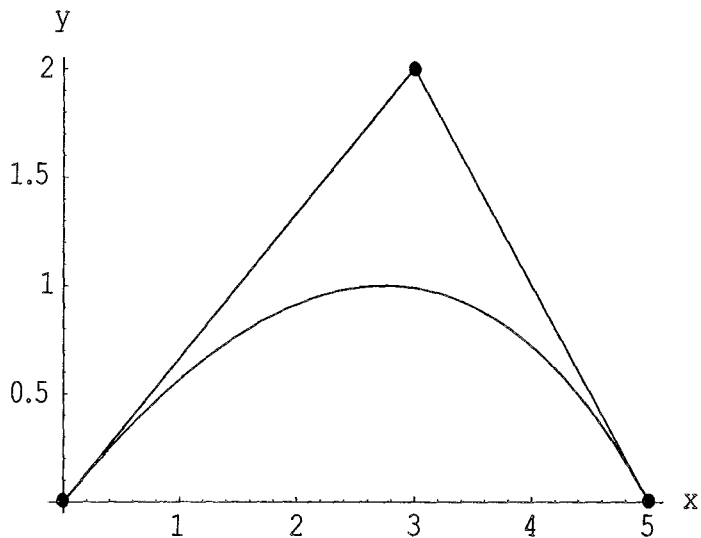
$$i = 1 : B_1^2(t) = \binom{2}{1} t^1 (1-t)^1 = 2t(1-t)$$

$$i = 2 : B_2^2(t) = \binom{2}{2} t^2 (1-t)^0 = t^2$$

Oleh itu, kita memperolehi persamaan lengkung Bezier kuadratik iaitu

$$P(t) = v_0(1-t)^2 + 2v_1t(1-t) + v_2t^2 \quad (2.3)$$

Rajah 2.1 menunjukkan contoh lengkung Bezier kuadratik.



Rajah 2.1 : Lengkung Kuadratik Dengan Poligon Kawalan

#### 2.4.2 Bezier Kubik

Untuk lengkung kubik ( $n = 3$ ), bilangan titik kawalan adalah empat.

Persamaan (2.2) akan menjadi

$$P(t) = \sum_{i=0}^3 v_i B_i^3(t)$$

$$P(t) = v_0 B_0^3(t) + v_1 B_1^3(t) + v_2 B_2^3(t) + v_3 B_3^3(t)$$

Nilai bagi polinomial Bernstein  $B_i^3(t)$  :

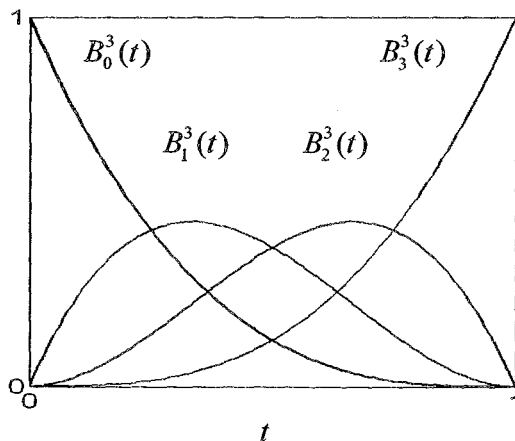
$$i = 0 : B_0^3(t) = \binom{3}{0} t^0 (1-t)^3 = (1-t)^3$$

$$i = 1 : B_1^3(t) = \binom{3}{1} t^1 (1-t)^2 = 3t(1-t)^2$$

$$i = 2 : B_2^3(t) = \binom{3}{2} t^2 (1-t)^1 = 3t^2(1-t)$$

$$i = 3 : B_3^3(t) = \binom{3}{3} t^3 (1-t)^0 = t^3$$

Rajah 2.2 menunjukkan polinomial Bernstein darjah 3.

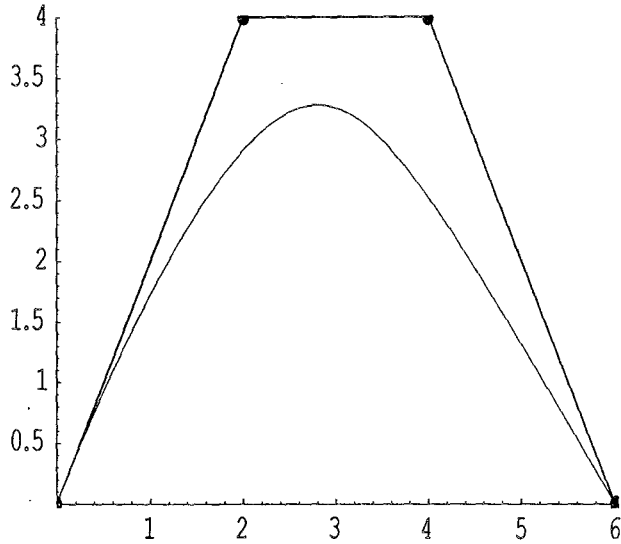


Rajah 2.2 : Polinomial Bernstein Darjah 3

Dengan ini, kita memperoleh persamaan lengkung Bezier kubik iaitu

$$P(t) = v_0(1-t)^3 + 3v_1t(1-t)^2 + 3v_2t^2(1-t) + v_3t^3 \quad (2.4)$$

Rajah 2.3 menunjukkan contoh lengkung Bezier kubik.



Rajah 2.3 : Lengkung Kubik Dengan Poligon Kawalan

## 2.5 Keselajaran Geometri Dan Keselajaran Parameter

Bila kita ingin menyambung dua atau lebih lengkung Bezier, kita perlu mengambil kira keselajaran pada titik penyambungan antara lengkung-lengkung tersebut. Terdapat dua jenis keselajaran, iaitu keselajaran parameter berdarjah  $r$  iaitu  $C^r$  dan keselajaran geometri berdarjah  $r$  iaitu  $G^r$ .

$$P(t) = \begin{cases} P_1(t) & , \quad a - \xi < t \leq a \\ P_2(t) & , \quad a \leq t < a + \delta \end{cases}$$

di mana  $\xi$  dan  $\delta$  nombor yang sangat kecil. Syarat keselajaran bagi  $P(t)$  pada titik  $t = a$  adalah seperti berikut [4].

i) Keselajaran Posisi

Lengkung  $P(t)$  memenuhi syarat  $G^0$  dan  $C^0$  sekiranya lengkung  $P_1(t)$  dan  $P_2(t)$  bertemu pada titik  $a$  iaitu

$$P_1(a) = P_2(a)$$

ii) Keselajaran tangen

Lengkung  $P(t)$  memenuhi syarat  $G^1$  sekiranya tangen pada titik penyambungan kedua-dua lengkung adalah sama iaitu

$$P_2'(a) = \beta P_1'(a), \quad \beta > 0$$

Jika  $\beta = 1$ ,  $G^1$  akan menjadi  $C^1$ .

iii) Keselajaran vektor kelengkungan

Lengkung  $P(t)$  memenuhi syarat  $G^2$  sekiranya vektor kelengkungan pada titik penyambungan kedua-dua lengkung adalah sama iaitu

$$P_2''(a) = \beta^2 P_1''(a) + \gamma P_1'(a)$$

di mana  $\gamma$  ialah nombor nyata dan  $\beta > 0$ . Jika  $\beta = 1$  dan  $\gamma = 0$ ,  $G^2$  akan menjadi  $C^2$ .

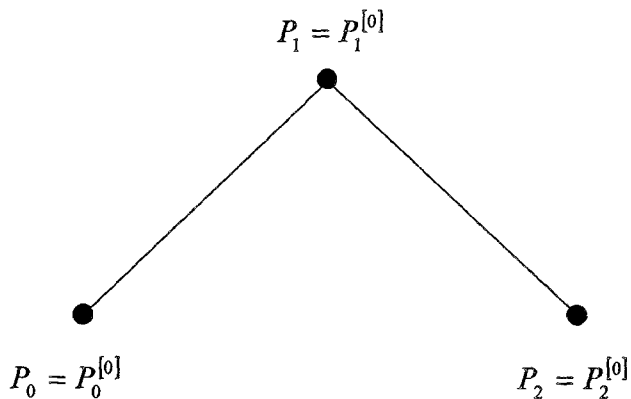
## BAB 3

### PENGHASILAN LENGKUNG BEZIER DENGAN MENGGUNAKAN ALGORITMA DE CASTELJAU

Dalam bab ini, kita akan membincangkan algoritma subbahagian bagi lengkung Bezier dengan menggunakan algoritma de Casteljau. Diberi titik-titik  $P_0, P_1, \dots, P_n$ , matlamat di sini ialah menghasilkan lengkung Bezier iaitu  $P(t)$ , di mana  $t \in [0,1]$ .

#### 3.1 Penghasilan Lengkung Kuadratik

Untuk menghasilkan lengkung kuadratik ( $n = 2$ ), 3 titik kawalan diperlukan iaitu  $P_0, P_1, P_2$ . Poligon kawalan ditunjukkan seperti dalam Rajah 3.1(a).

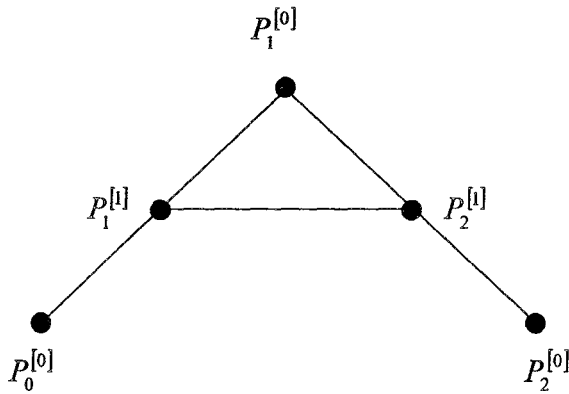


Rajah 3.1(a) : Poligon Kawalan Untuk Lengkung Kuadratik

Setiap segmen garis merah akan dibahagikan mengikut nisbah  $t : 1 - t$ , menghasilkan titik-titik berwarna hijau iaitu  $P_1^{[1]}$  dan  $P_2^{[1]}$ . Titik-titik ini jika disambungkan akan menghasilkan segmen garis yang berwarna hijau [2, 4, 7]. Rajah 3.1(b) menunjukkan titik dan segmen garis tersebut.

$$P_1^{[1]}(t) = (1-t)P_0^{[0]} + tP_1^{[0]}$$

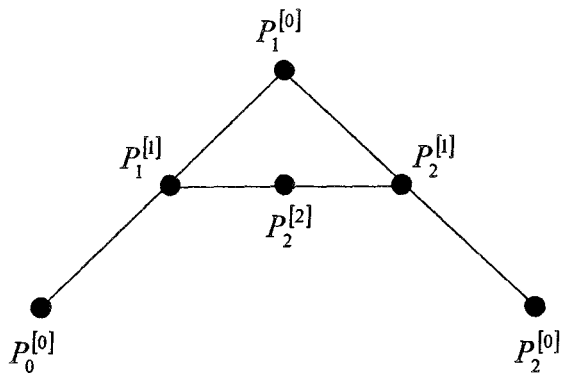
$$P_2^{[1]}(t) = (1-t)P_1^{[0]} + tP_2^{[0]}$$



Rajah 3.1(b) : Poligon Kawalan Untuk Lengkung Kuadratik

Proses yang sama diulangi untuk mencari titik  $P_2^{[2]}$  berdasarkan segmen garis berwarna hijau. Titik ini diwakili oleh titik berwarna biru dalam Rajah 3.1(c).

$$P_2^{[2]}(t) = (1-t)P_1^{[1]}(t) + tP_2^{[1]}(t)$$



Rajah 3.1(c) : Poligon Kawalan Untuk Lengkung Kuadratik

Segmen pertama lengkung Bezier diwakili oleh titik-titik kawalan  $P_0^{[0]}, P_1^{[1]}$  dan  $P_2^{[2]}$  manakala segmen kedua lengkung Bezier diwakili oleh titik-titik kawalan  $P_2^{[2]}, P_2^{[1]}$  dan  $P_2^{[0]}$ . Titik kawalan  $P_2^{[2]}$  akan menjadi titik akhir bagi segmen pertama lengkung kuadratik dan titik awal bagi segmen kedua lengkung kuadratik. Persamaan untuk segmen pertama lengkung ialah

$$P_1(t) = P_0^{[0]}(1-t)^2 + 2P_1^{[1]}t(1-t) + P_2^{[2]}t^2$$

dan persamaan untuk segmen kedua lengkung ialah

$$P_2(t) = P_2^{[2]}(1-t)^2 + 2P_2^{[1]}t(1-t) + P_2^{[0]}t^2$$

Langkah seterusnya adalah menggantikan nilai  $t$  dari 0 hingga 1.

Untuk  $t = 0$ :

$$\begin{aligned} P_1(t) &= P_0^{[0]}(1-t)^2 + 2P_1^{[1]}t(1-t) + P_2^{[2]}t^2 \\ P_1(0) &= P_0^{[0]} \\ P_2(t) &= P_2^{[2]}(1-t)^2 + 2P_2^{[1]}t(1-t) + P_2^{[0]}t^2 \\ P_2(0) &= P_2^{[2]} \end{aligned}$$

Untuk  $t = 0$ , bagi segmen pertama lengkung, kita mendapat titik  $P_0$  manakala bagi segmen kedua lengkung, kita mendapat titik  $P_2^{[2]}$ .

Untuk  $t = 0.1$  :

$$\begin{aligned}P_1(t) &= P_0^{[0]}(1-t)^2 + 2P_1^{[1]}t(1-t) + P_2^{[2]}t^2 \\P_1(0.1) &= (0.9)^2 P_0^{[0]} + 2P_1^{[1]}(0.1)(0.9) + (0.1)^2 P_2^{[2]} \\&= T_1\end{aligned}$$

$$\begin{aligned}P_2(t) &= P_2^{[2]}(1-t)^2 + 2P_2^{[1]}t(1-t) + P_2^{[0]}t^2 \\P_2(0.1) &= (0.9)^2 P_2^{[2]} + 2(0.1)(0.9)P_2^{[1]} + (0.1)^2 P_2^{[0]} \\&= U_1\end{aligned}$$

Untuk  $t = 0.1$ , bagi segmen pertama lengkung, kita mendapat titik  $T_1$  manakala bagi segmen kedua lengkung, kita mendapat titik  $U_1$ .

Untuk  $t = 0.2$  :

$$\begin{aligned}P_1(t) &= P_0^{[0]}(1-t)^2 + 2P_1^{[1]}t(1-t) + P_2^{[2]}t^2 \\P_1(0.2) &= (0.8)^2 P_0^{[0]} + 2(0.2)(0.8)P_1^{[1]} + (0.2)^2 P_2^{[2]} \\&= T_2\end{aligned}$$

$$\begin{aligned}P_2(t) &= P_2^{[2]}(1-t)^2 + 2P_2^{[1]}t(1-t) + P_2^{[0]}t^2 \\P_2(0.2) &= (0.8)^2 P_2^{[2]} + 2(0.2)(0.8)P_2^{[1]} + (0.2)^2 P_2^{[0]} \\&= U_2\end{aligned}$$

Untuk  $t = 0.2$ , bagi segmen pertama lengkung, kita mendapat titik  $T_2$  manakala bagi segmen kedua lengkung, kita mendapat titik  $U_2$ . Proses diulangi dengan menggantikan  $t = 0.3, 0.4, \dots, 0.9, 1.0$ .

Untuk  $t = 1.0$  :

$$\begin{aligned}P_1(t) &= P_0^{[0]}(1-t)^2 + 2P_1^{[1]}t(1-t) + P_2^{[2]}t^2 \\P_1(1) &= P_2^{[2]} \\P_2(t) &= P_2^{[2]}(1-t)^2 + 2P_2^{[1]}t(1-t) + P_2^{[0]}t^2 \\P_2(1) &= P_2^{[0]}\end{aligned}$$