

MAT301 Matematikk III  
forelesningsnotater og oppgaver

Dag Nylund

8. januar 2025

# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>1</b>
<b>2</b>	<b>Funksjoner</b>	<b>2</b>
2.1	Introduksjon . . . . .	2
2.2	Funksjoner av en variabel . . . . .	2
2.3	Grafen til en funksjon . . . . .	3
2.3.1	Verteks . . . . .	3
2.3.2	Horner's regel . . . . .	4
2.3.3	Newton's metode . . . . .	5
2.3.4	Numerisk derivasjon . . . . .	6
2.3.5	Analyse av datasett . . . . .	7
2.4	Vektorfunksjoner . . . . .	8
2.5	Funksjoner av to variable $z = f(x, y)$ . . . . .	9
2.5.1	Grenseverdier, kontinuitet og deriverbarhet . . . . .	9
2.5.2	Partiellderiverte . . . . .	9
2.5.3	Tangentplan og linearisering . . . . .	9
2.5.4	Gradient . . . . .	10
2.5.5	Andrederiverte og Hessematrixen . . . . .	10
2.5.6	Eksempel . . . . .	11
2.5.7	Eksempel . . . . .	11
2.6	Dobbeltintegral . . . . .	11
2.6.1	Eksempel . . . . .	12
2.6.2	Eksempel . . . . .	12
2.6.3	Integrasjon over ikke-rektangulært område . . . . .	12
2.6.4	Eksempel: Numerisk integrasjon . . . . .	13
2.7	Programmering - tegne grafen til $f(x, y)$ . . . . .	13
2.7.1	Franke's testfunksjon . . . . .	14
2.8	Oppgaver . . . . .	15
2.8.1	. . . . .	15
2.8.2	. . . . .	15
2.8.3	. . . . .	15
2.8.4	. . . . .	15
2.8.5	. . . . .	15
2.8.6	. . . . .	16
2.8.7	. . . . .	16
2.8.8	. . . . .	16
2.8.9	. . . . .	16
2.8.10	. . . . .	16

2.8.11	16
2.8.12	16
2.8.13	17
2.8.14	17
<b>3 Vektorer og matriser</b>	<b>18</b>
3.1 Introduksjon	18
3.2 Vektorrom	18
3.3 Invers transformasjon	19
3.4 Vektorer, punkt og frames	20
3.5 Matriser	22
3.5.1 Rotasjoner	22
3.5.2 Homogene koordinater	23
3.5.3 Skalering	23
3.5.4 Translasjon	23
3.5.5 Refleksjon, projeksjon og forskyvning	24
3.5.6 Sammensatte transformasjoner	24
3.5.7 Programmering	25
3.5.8 C++ matriserepresentasjon	25
3.5.9 Viewing	25
3.5.10 Perspektivprojeksjon	27
3.5.11 Andre funksjoner i matriseklasse	27
3.6 Gausseliminasjon	28
3.6.1 Rader, kolonner og b-vektor	28
3.6.2 LU-faktorisering	28
3.6.3 Eksempel	28
3.6.4 LU-faktorisering algoritme	29
3.6.5 Eksempel	30
3.6.6 Invers matrise	30
3.6.7 Pivotering	30
3.6.8 solve() med pivotering	31
3.7 Oppgaver	32
3.7.1	32
3.7.2	32
3.7.3	32
3.7.4	32
3.7.5	32
<b>4 Konstruksjon av funksjoner</b>	<b>33</b>
4.1 Introduksjon	33
4.2 Interpolasjon	33
4.2.1 Interpolasjon av to, tre og fire punkter	35
4.2.2 Polynomrommet $\mathcal{P}^d$	37
4.3 Minste kvadraters metode	40
4.3.1 Eksempel	41
4.3.2 Eksempel	41
4.3.3 Minste kvadraters metode og andregradsfunksjon	42
4.3.4 Eksempel	42
4.3.5 Sammendrag	43
4.4 Kubisk Hermite interpolasjon	44

4.4.1	Eksempel . . . . .	44
4.4.2	Stykkevis kubisk Hermite interpolasjon . . . . .	47
4.4.3	Eksempel . . . . .	48
4.4.4	Kubisk spline interpolasjon . . . . .	48
4.5	Spline funksjoner . . . . .	50
4.5.1	Eksempel . . . . .	50
4.5.2	Eksempel . . . . .	51
4.5.3	Sammendrag . . . . .	52
4.6	Oppgaver . . . . .	54
4.6.1	. . . . .	54
4.6.2	. . . . .	54
4.6.3	. . . . .	54
4.6.4	. . . . .	54
4.6.5	. . . . .	54
4.6.6	. . . . .	55
4.6.7	. . . . .	56
4.6.8	. . . . .	56
4.6.9	. . . . .	56
4.6.10	. . . . .	56
4.6.11	. . . . .	56
4.6.12	. . . . .	57
4.6.13	. . . . .	58
4.6.14	. . . . .	58
4.6.15	. . . . .	58
4.6.16	. . . . .	58
4.6.17	. . . . .	58
<b>5</b>	<b>Affint rom</b>	<b>60</b>
5.1	Introduksjon . . . . .	60
5.1.1	Punkt og vektor . . . . .	60
5.1.2	Affin kombinasjon . . . . .	61
5.1.3	Linje . . . . .	61
5.1.4	Konveksitet . . . . .	62
5.1.5	Parameterframstilling av en rett linje . . . . .	62
5.1.6	Affin kombinasjon av tre punkter . . . . .	62
5.1.7	Barysentriske koordinater for en trekant . . . . .	62
5.2	Affint rom og basisfunksjoner . . . . .	63
5.3	Bernstein basisfunksjoner . . . . .	65
5.3.1	Lineære basisfunksjoner . . . . .	65
5.3.2	Kvadratiske basisfunksjoner . . . . .	65
5.3.3	Kubiske basisfunksjoner . . . . .	67
5.3.4	Derivasjon . . . . .	68
5.4	Lagrange basisfunksjoner . . . . .	70
5.4.1	Matriseform . . . . .	71
5.4.2	Eksempel . . . . .	71
5.5	B-spline basisfunksjoner . . . . .	73
5.5.1	Introduksjon . . . . .	73
5.5.2	Rekursiv definisjon av B-splines . . . . .	74
5.5.3	Eksempel, lineær B-spline, d=1 . . . . .	75
5.5.4	Eksempel, kvadratisk B-spline, d=2 . . . . .	76

5.5.5	Kontinuitet . . . . .	76
5.5.6	Multiple skjøter . . . . .	76
5.5.7	Sammenligning . . . . .	79
5.5.8	Affin kombinasjon av kontrollpunkter . . . . .	79
5.6	Å konstruere en kvadratisk B-spline basisfunksjon . . . . .	80
5.6.1	Beregne verdien for et punkt på en splinekurve . . . . .	82
5.6.2	Eksempel . . . . .	82
5.7	Oppgaver . . . . .	84
5.7.1	. . . . .	84
5.7.2	. . . . .	84
5.7.3	. . . . .	84
5.7.4	. . . . .	84
5.7.5	. . . . .	84
5.7.6	. . . . .	84
5.7.7	. . . . .	84
5.7.8	. . . . .	85
5.7.9	. . . . .	85
5.7.10	. . . . .	85
5.7.11	. . . . .	85
5.7.12	. . . . .	85
5.7.13	. . . . .	85
<b>6</b>	<b>Flater og trianguleringer</b>	<b>86</b>
6.1	Introduksjon . . . . .	86
6.2	Trianguleringer . . . . .	86
6.2.1	Normalvektoren til en flate . . . . .	87
6.2.2	Konstruksjon av 3d flater . . . . .	89
6.2.3	Implementering . . . . .	90
6.2.4	Anvendelser . . . . .	91
6.2.5	Søking . . . . .	92
6.2.6	Triangulering med indeksering . . . . .	92
6.2.7	Triangulering med naboinformasjon . . . . .	93
6.2.8	Topologisk søk algoritme . . . . .	95
6.2.9	Eksempel . . . . .	95
<b>7</b>	<b>Parametriske kurver</b>	<b>97</b>
7.1	Introduksjon . . . . .	97
7.2	Bezier kurver og deCasteljau algoritmen . . . . .	97
7.2.1	Egenskaper ved Bezierkurver og deCasteljau algoritmen . . . . .	99
7.2.2	C++ implementering av de Casteljau algoritmen . . . . .	99
7.2.3	Sammensatte Bezierkurver . . . . .	100
7.3	Neville's algoritme . . . . .	101
7.3.1	Eksempel . . . . .	102
7.4	Spline kurve . . . . .	105
7.4.1	Lokal support I . . . . .	106
7.5	Finne riktig intervall . . . . .	106
7.6	deBoor's algoritme . . . . .	107
7.6.1	Alternativ implementering . . . . .	108
7.6.2	Lokal support II . . . . .	109
7.7	Derivasjon . . . . .	109

7.7.1	Derivasjon av B-spline basisfunksjon . . . . .	109
7.7.2	Eksempel . . . . .	110
7.8	Oppgaver . . . . .	111
7.8.1	. . . . .	111
7.8.2	. . . . .	111
7.8.3	. . . . .	111
7.8.4	. . . . .	111
7.8.5	. . . . .	111
7.8.6	. . . . .	111
7.8.7	. . . . .	112
7.8.8	. . . . .	112
7.8.9	. . . . .	112
7.8.10	. . . . .	112
7.8.11	. . . . .	112
7.8.12	. . . . .	112
7.8.13	. . . . .	113
7.8.14	. . . . .	113
7.8.15	. . . . .	113
<b>A</b>	<b>Fasit</b>	<b>114</b>
A.1	. . . . .	114
A.2	. . . . .	120
A.3	. . . . .	122
A.4	. . . . .	124
A.5	. . . . .	128
A.6	. . . . .	130
A.7	. . . . .	136
A.8	. . . . .	138

# Kapittel 1

## Introduksjon

Disse notatene dekker de fleste av læringsmålene i Matematikk III.

# Kapittel 2

## Funksjoner

### 2.1 Introduksjon

I dette kapitlet repeterer vi noen av de mest kjente egenskapene til funksjoner av en variabel, vektorfunksjoner og funksjoner av to variable. Vi går gjennom noen eksempler på numerisk analyse. Vi har fokus på hvordan vi gjør beregninger for å kunne visualisere ulike typer funksjoner.

### 2.2 Funksjoner av en variabel

Fra grunnleggende matematisk analyse kjenner vi til funksjonsbegrepet og skrivemåten  $f(x)$ . De vanligste klasser av funksjoner er

- Polynomer, for eksempel  $f(x) = x^3 - 6x^2 + 11x - 6$
- Rasjonale funksjoner, for eksempel  $g(x) = \frac{x-3}{x^2-3x+2}$
- Eksponentialfunksjoner  $e^x$  og logaritmefunksjoner  $\ln x$
- Trigonometriske funksjoner  $\sin x$ ,  $\cos x$ ,  $\tan x$

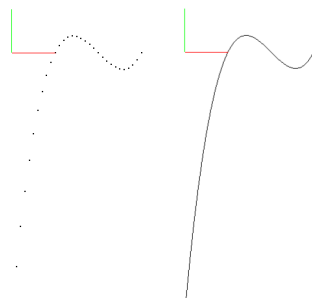
Vi kan ha kombinasjoner av disse, og vi kan analysere funksjonene. Kjente begreper er grenseverdier, kontinuitet og deriverbarhet, og grafen til funksjonen. For en funksjon  $f(x)$  gjelder at  $f$  er kontinuerlig for  $x = a$  hvis og bare hvis  $\lim_{x \rightarrow a} f(x) = f(a)$ . Husk at vi har ensidige grenseverdier som begge skal være like. En funksjon  $f(x)$  er deriverbar i  $x = a$  hvis og bare hvis høyresiden nedenfor eksisterer, og den deriverte er

$$f'(a) = \lim_{\Delta x \rightarrow 0} \frac{f(a + \Delta x) - f(a)}{\Delta x}$$

De ensidige grenseverdiene må være like, og noen ganger brukes skrivemåten  $f'(x) = \frac{df}{dx}$  for å vise hva vi deriverer med hensyn på. Ut fra denne definisjonen på den deriverte kan vi utlede formler for derivasjon av ulike klasser funksjoner.

Funksjonen  $f(x) = x^3 - 6x^2 + 11x - 6$  er opplagt deriverbar i hele  $\mathbb{R}$ , og vi har  $f'(x) = 3x^2 - 12x + 11$ . Den geometriske tolkningen av den deriverte i et





Figur 2.1:  $f(x) = x^3 - 6x^2 + 11x - 6$

punkt  $x = x_0$  er stigningstallet til tangenten. Ved å løse ligningen  $f'(x) = 0$  kan vi finne ekstremalpunkter. Bruker vi formelen for å løse andregradsligningen

$$ax^2 - bx + c = 0 \Leftrightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

får vi her  $x = \frac{12 \pm \sqrt{144 - 4 \cdot 3 \cdot 11}}{2 \cdot 6}$  som gir  $x = 2 \pm \frac{\sqrt{3}}{3}$ . For  $x = 2 - \frac{\sqrt{3}}{3}$  og  $x = 2 + \frac{\sqrt{3}}{3}$  har altså funksjonen lokale maksimums- eller minimumsverdier, og her har tangenten til grafen stigningstallet 0. I et vilkårlig punkt  $x_0$  er ligningen for tangenten  $y = f(x_0) + f'(x_0)(x - x_0)$ . For eksempel i  $x = 2$ , hvor  $f''(x) = 0$  og vi har et vendepunkt, er  $f(2) = 0$ ,  $f'(2) = -1$  og tangentligningen  $y = -x + 2$ .

## 2.3 Grafen til en funksjon

Grafen til en funksjon er en mye brukt visualisering av funksjonen, og det fins ulike verktøyer som gjør det enkelt å tegne grafen til en funksjon. Men hvordan gjør vi dette selv? Når vi gjør det manuelt, er vi vant til å skissere grafen til  $f$  ved å analysere funksjonen, regne ut noen verdier (nullpunkter, ekstremalpunkter, vendepunkter) og tegne opp en skisse. Som et utgangspunkt kan vi tenke oss at vi regner ut en del punkter (figur 2.1, til venstre) og forbinder disse med rette linjestykker. Hvis punktene ligger tett nok, vil grafen bli glatt og fin (figur 2.1, til høyre). Dette fungerer fint for  $f(x) = x^3 - 6x^2 + 11x - 6$ .

Vi kan altså tegne grafen til  $f$  ved å regne ut  $f(x)$  for mange  $x$ -verdier. La oss for eksempel se på intervallet  $[0, 3]$ . Siden  $f$  er en reell funksjon, er det uendelig mange  $x \in [0, 3]$  og vi kan ikke regne ut alle. I praksis velger vi en oppløsning  $h = \Delta x$  og regner ut funksjonsverdier for  $x_0 = 0$ ,  $x_1 = x_0 + h$ ,  $x_2 = x_1 + h = x_0 + 2h, \dots, 3$ . Til slutt kan vi tegne rette linjer mellom hvert par av punkter. Når vi skal programmere dette og bruke grafikkprossessor, kan vi sende et punktsett til grafikkprossessoren sammen med fargeegenskapen i hvert punkt og gi beskjed om at den skal tegne linjer mellom alle punktene  $(x_i, f(x_i))$ ,  $(x_{i+1}, f(x_{i+1}))$ .

### 2.3.1 Verteks

Et verteks er et punkt med koordinater og i tillegg egenskapsdata. Egenskapsdata kan være farge (rgb og eventuelt alfa-verdi/gjennomsiktighet), koordinata-

ter til en enhetsnormal  $(n_x, n_y, n_z)$ , teksturkoordinater  $(u, v)$ . Et verteks for et punkt  $P \in \mathbb{R}^3$  kan da bestå av

*xyz rgb uv eller xyz  $n_x n_y n_z$  uv*

lagret fortløpende i en array eller en struct. For å tegne grafen til funksjon  $f$  i et slikt system, kan vi altså

1. Velge et intervall  $[a, b]$  og en oppdeling  $h$
2. Velge en farge i hvert punkt (for eksempel rød  $(1.0, 0.0, 0.0)$ )
3. Regne ut punkter på grafen  $(x_i, f(x_i)) \in \mathbb{R}^2$  og angi  $z_i = 0.0$
4. Lagre alle verteksdata på en fil
5. Lese inn filen i en array i et annet program
6. Send arrayen til et shaderprogram på grafikkprosessoren med beskjed om å tegne linjestykker.

Prosessen består av både noe beregningsorientert matematikk og noe grafikk-programmering. Vi kan si at den beregningsorienterte delen avsluttes med punkt 4, og at punkt 5 og 6 er grafikkprogrammering.

### 2.3.2 Horner's regel

Hvordan kan vi regne ut funksjonsverdien  $f(x)$  effektivt? La oss bruke funksjonen ovenfor som et eksempel,  $f(x) = x^3 - 6x^2 + 11x - 6$ . For de fleste vil det sannsynligvis være naturlig å implementere dette som

```
double f(const double x) {  
    return x*x*x-6*x*x+11*x-6;  
}
```

Her er det lett å telle opp 5 multiplikasjoner og 3 addisjoner/subtraksjoner. I det generelle tilfellet med en koeffisient foran  $x^3$ -leddet, får vi 6 multiplikasjoner og 3 addisjoner/subtraksjoner. Kan vi gjøre utregningen mer effektiv?

La oss først skrive opp funksjonsuttrykket på nøstet form:

$$f(x) = -6 + x(11 - 6x + x^2) = -6 + x \cdot (11 - x \cdot (6 - x))$$

Hvis vi regner ut funksjonsverdien på denne måten, trenger vi bare 2 multiplikasjoner og 3 addisjoner/subtraksjoner. (Første minustegn representerer ingen subtraksjon, men vi vil få en multiplikasjon mer hvis vi har med en koeffisient foran  $x^3$ -leddet). Vi sparer altså 3 multiplikasjoner.

En enkel implementering blir som følger:

```
double Horner(const double x, const int n, double a[])  
{  
    double f=0.0;  
    for(auto i=n; i>=0; i--)  
        f = x*f+a[i];  
    return f;  
}
```

Vi har altså en algoritme for effektiv evaluering av for evaluering av polynomet  $f(x) = \sum_{i=0}^n a_i x^i$ . Algoritmen kalles Horner's regel. Den er også omtalt i [8], oppgave 2.14.

### 2.3.3 Newton's metode

Vi viste ovenfor et eksempel på analyse av funksjonen  $f(x) = x^3 - 6x^2 + 11x - 6$ . Hvordan kan vi finne nullpunktene uten å gjøre manuelle utregninger? Vi kan bruke Newtons metode, som er et eksempel på en numerisk metode.

Newton's metode er en kjent iterativ metode for å finne nullpunkter til en funksjon. Vi tar utgangspunkt i Taylorrekkeutvikling i punktet  $x_0$  med ett ledd,

$$f(x) = f(x_0) + f'(x_0)(x - x_0)$$

Her er både  $f$  og  $f'$  kjente uttrykk, og  $f(x_0)$  og  $f'(x_0)$  er kjente verdier. Hvis vi setter venstresiden  $f(x) = 0$  blir uttrykket

$$0 = f(x_0) + f'(x_0)(x - x_0)$$

som gir

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Når vi bruker denne ligningen gjentatte ganger, får vi en følge av tall  $x_0, x_1, \dots, x_k$  som forhåpentligvis konvergerer mot et nullpunkt for funksjonen.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2.1)$$

#### Eksempel

Vi skal prøve å finne et nullpunkt for  $f$ .

Vi gjetter først på  $x_0 = \frac{1}{2}$ . Vi får  $f'(x) = 3x^2 - 12x + 11$ , videre  $f(\frac{1}{2}) = -\frac{15}{8}$ ,  $f'(\frac{1}{2}) = \frac{23}{4}$ , slik at  $x_1 = \frac{1}{2} + \frac{15}{46} \approx 0.826$ .

I neste iterasjon får vi  $f(0.826) = -0.444$  og  $f'(0.826) = 3.135$  slik at  $x_2 = 0.826 + \frac{0.444}{3.135} = 0.968$ .

I tredje iterasjon får vi  $f(0.968) = 0.067$  og  $f'(0.968) = 2.20$  slik at  $x_3 = 0.968 + \frac{0.067}{2.20} = 0.998$ .

#### Eksempel

La  $f(x) = \sin x$ . Vi skal prøve å finne et nullpunkt for  $f$ .

Vi gjetter først på  $x_0 = \frac{\pi}{4}$ . Vi får  $f'(x) = \cos x$ , videre  $f(\frac{\pi}{4}) = f'(\frac{\pi}{4}) = \frac{\sqrt{2}}{2}$ , slik at  $x_1 = \frac{\pi}{4} - 1 \approx -0.215$ .

I neste iterasjon får vi  $\sin(-0.215) = -0.213$  og  $\cos(-0.215) = 0.977$  slik at  $x_2 = -0.215 + \frac{0.977}{0.213} = 0.003$ .

## Implementering

Listing 2.1: Newton's metode

```
double f(double x) { return std::sin(x); }
double df(double x) { return std::cos(x); }

std::pair<double, int> newton(double x0)
{
    int iter{0};
    auto fx = f(x0);
    double x = x0;
    while (std::abs(fx) > 0.01) {
        x = x - fx/df(x);
        fx = f(x);
        iter++;
    }
    return std::pair<double, int>(x, iter);
}

int main() {
    auto par = newton(0.785);
    cout << par.first << ", " << par.second << endl;
    return 0;
}
```

## Diskusjon

I eksemplene ovenfor finner vi med få iterasjoner et nullpunkt som tilfredsstiller akseptabel/oppgitt toleranse. Men vi har ikke angitt noe definisjonsområde  $D_f$  for funksjonene -  $\sin x$  er en periodisk funksjon, og antall nullpunkter avhenger av  $D_f$ . Nullpunktet vi finner, avhenger av initiell (helst kvalifisert) gjetning. Hvis vi kjører programmet ovenfor (for  $f(x) = \sin x$ ) med startverdi  $x_0 = 2.0$ , får vi  $x_1 = 4.185$ ,  $x_2 = 2.468$ ,  $x_3 = 3.266$ ,  $x_4 = 3.141$ . Programmet bruker 4 iterasjoner på å finne et annet nullpunkt. Hvis vi skal finne alle nullpunktene til en funksjon, trenger vi å bruke metoden med flere initialverdier.

### 2.3.4 Numerisk derivasjon

I mange tilfeller har vi ingen kjent underliggende funksjon, men måldata  $(x_i, y_i)$ ,  $i = 0, 1, \dots$ . Vi kan tolke  $y_i = f(x_i)$  for en ukjent funksjon  $f$ . Vi kan benytte Newton's differenskvotient som en approksimasjon (tilnærming) til den deriverte:

$$f'(a) \approx \frac{f(a+h) - f(a)}{h} \quad (2.2)$$

La oss bruke Newtons metode med Newton's differenskvotient på funksjonen  $f(x) = x^3 - 6x^2 + 11x - 6$ .

#### Eksempel

Vi skal prøve å finne et nullpunkt for  $f$ . Vi antar at vi bruker en oppdeling  $h = 0.1$  når vi regner ut verdier for  $f$

Vi gjetter først på  $x_0 = \frac{1}{2}$ , som i eksemplet ovenfor. Vi får  $f(\frac{1}{2}) = -\frac{15}{8} = -1.875$ ,  $f(\frac{1}{2} + h) = f(0.6) = -1.344$   $f'(\frac{1}{2}) \approx \frac{-1.344 + 1.875}{0.1} = 5.31$ , slik at  $x_1 = \frac{1}{2} + \frac{-1.875}{5.31} \approx 0.853$ .

I neste iterasjon får vi  $f(0.853) = -0.362$ ,  $f(0.953) = -0.101$  og  $f'(0.853) \approx \frac{-0.101+0.362}{0.1} = 2.61$ , slik at  $x_2 = 0.853 + \frac{0.362}{2.61} = 0.992$ .

Hvis vi bruker en toleranse på 0.01, finner vi i dette tilfellet et nullpunkt for  $f$  med en iterasjon mindre enn når vi bruker det eksakte uttrykket for  $f'$ .

### 2.3.5 Analyse av datasett

Tabellen nedenfor viser fire kolonner med tall, eller to kolonner med sammenhengende verdier. I praktiske situasjoner kan vi ha tusenvis av punkter. Hvis dette er alt vi vet, hvordan kan vi analysere dette? Kan vi gjette på at dette er  $x$ - og  $y$ -verdier, eller er det  $y$ - og  $x$ -verdier?

-1.875	0.500	0.274	1.700
-1.542	0.550	0.287	1.150
-1.325	0.600	0.308	1.650
-1.030	0.650	0.318	1.200
-0.838	0.700	0.330	1.250
-0.655	0.750	0.331	3.100
-0.493	0.800	0.348	1.600
-0.227	2.350	0.366	1.300
-0.212	2.850	0.399	1.400
-0.195	2.300	0.400	1.450
-0.186	2.200	0.403	1.550
-0.174	2.250	0.412	1.350
-0.149	0.900	0.471	3.150
-0.111	2.150	0.474	1.500
-0.085	2.900	0.589	3.200
-0.072	2.950	0.742	3.250
-0.054	2.100	0.924	3.300
-0.033	0.950	1.140	3.350
0.016	2.050	1.428	3.400
0.017	2.000	1.601	3.450
0.017	1.000	-0.329	2.450
0.078	3.000	-0.314	2.750
0.111	1.950	-0.311	2.550
0.159	1.900	-0.303	2.700
0.179	1.050	-0.288	2.600
0.192	3.050	-0.287	2.500
0.204	1.850	-0.284	2.400
0.242	1.100	-0.283	2.650
0.245	1.800	-0.281	0.850
0.272	1.750	-0.242	2.800

Vi kan tenke oss ulike tilnærminger:

- Dette kan være punkter i et plan, og vi kan bestemme hvilket rektangulære område de ligger i ved å finne minste og største første- og andrekoordinat.
- Det kan være punkter på en kurve (ikke nødvendigvis en funksjon). Vi kan

prøve å trekke rette linjer mellom to og to punkter, men hvordan kan vi analysere resultatet?

- Vi kan sortere punktene enten etter første- eller andrekoordinatene og på den måten finne ut om det er en ukjent underliggende funksjon her.
- Hvis vi finner en funksjon, kan vi undersøke om den har nullpunkter, og vi kan beregne en tilnærming til den deriverte.
- Vi kan undersøke om funksjonen er kontinuerlig. Når førstekoordinatene er sortert, kan vi undersøke differansen mellom alle par av påfølgende andrekoordinater. Dersom  $|y_{i+1} - y_i| > L$  for en selvvalgt grense  $L$ , kan vi anta at det er en diskontinuitet i intervallet  $(x_i, x_{i+1})$ .

## 2.4 Vektorfunksjoner

En vektor (se avsnitt 3.2 skrives ofte som  $\mathbf{u} = [u_1, u_2]$  (vektor i  $\mathbb{R}^2$ ),  $\mathbf{v} = [v_1, v_2, v_3]$  (vektor i  $\mathbb{R}^3$ ), hvor  $u_1, u_2, v_1, v_2, v_3$  er koordinater. Antall koordinater er like mange som dimensjonen til vektorrommet. Hvis vi setter inn tall, får vi for eksempel vektorene  $\mathbf{u} = [1, 2]$  i planet og  $\mathbf{v} = [1, 2, 1]$  i rommet. (Koordinatene 1,2 og 1,2,1 multipliseres med respektive basisvektorer). Hvis vi benytter et standard kartesisk koordinatsystem, med standard basis (henholdsvis  $[1,0]$ ,  $[0,1]$  og  $[1,0,0]$ ,  $[0,1,0]$ ,  $[0,0,1]$ ) kan vi tenke på  $\mathbf{u} = [1, 2]$  som en vektor fra origo som peker på punktet (1,2) i planet. Tilsvarende,  $\mathbf{v} = [1, 2, 1]$  en vektor fra origo som peker på punktet (1,2,1) i rommet. Men husk nå på at en vektor kjennetegnes ved lengde og retning, og er lik uavhengig posisjon - grunnen til at vi her innfører punktet origo er for å forklare begrepet vektorfunksjon!

Vi erstatter nå  $u_1, u_2, v_1, v_2, v_3$  med funksjoner. For eksempel kan vi erstatte  $u_1$  med  $x(t)$  og  $v_1$  med  $y(t)$ , hvor  $t$  er et reelt tall. Da får vi i stedet for  $\mathbf{u}$  en vektorfunksjon  $\mathbf{f}(t) = (x(t), y(t))$ . Vi ser på noen eksempler hvor vi setter inn funksjonsuttrykk for  $x(t)$ ,  $y(t)$ .

$x(t) = \cos t$ ,  $y(t) = \sin t$ , gir  $\mathbf{f}(t) = (\cos t, \sin t)$ . Hvis vi nå lar vektoren starte i origo, vil den peke på et punkt på enhetssirkelen. Ved å variere  $t$ , fra 0 til  $2\pi$ , kan vi få  $\mathbf{f}$  til å rotere rundt hele enhetssirkelen.

Et lignende eksempel i tre dimensjoner:  $x(t) = \cos t$ ,  $y(t) = \sin t$ ,  $z(t)$  gir  $\mathbf{f}(t) = (\cos t, \sin t, \frac{t}{10})$ . Ved å variere  $t$ , fra 0 til  $6\pi$ , kan vi få  $\mathbf{f}$  til å rotere rundt hele enhetssirkelen tre ganger, med stigende  $z$ -verdi, som en spiral.

## 2.5 Funksjoner av to variable $z = f(x, y)$

Vi kan bruke funksjoner av to variable til å lage landskap i 3d scener. Analyse for funksjoner av en variabel kan lett utvides til funksjoner av to variable, og vi skal gå gjennom noe av dette: Kontinuitet, grenseverdier, deriverbarhet, maksimums- og minimumsverdier.

Vi skal også studere elementer av hvordan vi kan implementere rendring av grafen til en funksjon av to variable.

### 2.5.1 Grenseverdier, kontinuitet og deriverbarhet

For en funksjon  $f(x)$  gjelder at  $f$  er kontinuerlig for  $x=a$  hvis og bare hvis  $\lim_{x \rightarrow a} f(x) = f(a)$ . Husk at vi har ensidige grenseverdier som begge skal være like. Når vi har en funksjon av to variable må dette gjelde samme hvilken retning vi nærmer oss et punkt  $(a, b)$  og  $f(x, y)$  er kontinuerlig for  $(x, y) = (a, b)$  hvis og bare hvis

$$\lim_{(x, y) \rightarrow (a, b)} f(x, y) = f(a, b)$$

En funksjon  $f(x)$  er deriverbar i  $x=a$  hvis og bare hvis høyresiden nedenfor eksisterer, og den deriverte er

$$f'(a) = \lim_{\Delta x \rightarrow 0} \frac{f(a + \Delta x) - f(a)}{\Delta x}$$

De ensidige grenseverdiene må være like, og noen ganger brukes skrivemåten  $f'(x) = \frac{df}{dx}$  for å vise hva vi deriverer med hensyn på. Når vi har en funksjon av to variable, kan vi derivere med hensyn på enten den ene eller den andre. Dette angir vi med å erstatte  $f'(x) = \frac{df}{dx}$  med  $f_x = \frac{\partial f}{\partial x}$  og tilsvarende for  $y$ .

### 2.5.2 Partiellderivate

Uttrykkene for de partiellderivate blir da  $\frac{\partial f}{\partial x}(x, y)$  og  $\frac{\partial f}{\partial y}(x, y)$ . Når vi partiell-deriverer med hensyn på en variabel betraktes den andre som konstant (fordi det skjer ingen endring av  $y$  i  $x$ -retning og omvendt).

#### Eksempel: Apesadelen

Funksjonen  $f(x, y) = 2x^2y$ ,  $-1 \leq x \leq 1$ ,  $-1 \leq y \leq 1$  kalles gjerne apesadelen. De partiellderivate blir  $\frac{\partial f}{\partial x} = 4xy$  og  $\frac{\partial f}{\partial y} = 2x^2$ .

#### Eksempel

$f(x, y) = \frac{2x}{y}$ . Da blir de partiellderivate  $\frac{\partial f}{\partial x} = \frac{2}{y}$  og  $\frac{\partial f}{\partial y} = -\frac{2x}{y^2}$ .

### 2.5.3 Tangentplan og linearisering

Ligningen for tangentplanet til en funksjon kalles lineariseringen. Lineariseringen til en funksjon av to variable  $f(x, y)$  om punktet  $(a, b)$  er gitt ved

$$L(x, y) = f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) \quad (2.3)$$

En forutsetning er at grenseverdien nedenfor eksisterer og er 0 (se [6], 11.2.4).

$$\lim_{(x,y) \rightarrow (a,b)} \frac{f(x,y) - L(x,y)}{\sqrt{(x-a)^2 + (y-b)^2}} = 0$$

### Eksempel

Finn tangentplanet til funksjonen  $f(x,y) = xy$  i punktet  $(1,1)$ .

$$f_x(x,y) = y$$

$$f_y(x,y) = x$$

$$f(1,1) = 1$$

$$f_x(1,1) = 1$$

$$f_y(1,1) = 1$$

$$L(x,y) = f(1,1) + f_x(1,1)(x-1) + f_y(1,1)(y-1) = 1 + (x-1) + (y-1) = x + y - 1$$

Vi skal se nærmere på tangentplan og normalvektor i kapittel 6.

### 2.5.4 Gradient

Gradienten til en funksjon  $f(x,y)$  er gitt ved

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \quad (2.4)$$

Gradienten er en vektor. Den kan regnes ut i et hvilket som helst punkt. Gradienten er normalvektor til nivåkurven for en funksjon av to variable (se [6] teorem 11.2.19).

### 2.5.5 Andrederiverte og Hessematrisen

Vi husker en del metoder for å analysere funksjoner av en variabel

- Derivere og sette  $f'(x)=0$  for å finne maks-eller min-punkt.
- Finne ligningen til tangenten i et punkt - tilsvarende å finne linearisering/tangentplan
- Den deriverte har stigningstall 0 i ekstremalpunkter (her: stasjonære punkter, de partiellderiverte er 0)
- Den andrederiverte sier noe om krumningen (f.eks  $x^2$  og  $-x^2$ ) - vi skal nå gjøre noe tilsvarende andrederiverttesten for en funksjon av to variable.

Hvis funksjonen er to ganger deriverbar, skriver vi de partiellderiverte slik:

$$\frac{\partial^2 f}{\partial x^2} \quad \frac{\partial^2 f}{\partial y \partial x} \quad \frac{\partial^2 f}{\partial x \partial y} \quad \frac{\partial^2 f}{\partial y^2}$$

I første tilfelle deriverer vi med hensyn på  $x$  to ganger. I det andre tilfellet deriverer vi med hensyn på  $y$  og deretter med hensyn på  $x$ . I det tredje tilfellet deriverer vi med hensyn på  $x$  først og deretter med hensyn på  $y$ . Dette gir samme resultat. I siste tilfelle deriverer vi med hensyn på  $y$  to ganger.



De andreordens partiellderiverte kan brukes til å klassifisere stasjonære punkter (vi husker at de stasjonære punktene er der hvor de partiellderiverte begge er null). Vi setter da opp den såkalte Hessematrisen og regner ut determinanten i de aktuelle punktene.

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial y \partial x} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{yx} \\ f_{xy} & f_{yy} \end{bmatrix} \quad (2.5)$$

$$\Delta = \det|H| = f_{xx} \cdot f_{yy} - f_{xy} \cdot f_{yx} \quad (2.6)$$

1.  $\Delta > 0$  og  $f_{xx} < 0 \Rightarrow$  lokalt maksimum.
2.  $\Delta > 0$  og  $f_{xx} > 0 \Rightarrow$  lokalt minimum.
3.  $\Delta < 0 \Rightarrow$  sadelpunkt.
4.  $\Delta = 0$  - Metoden gir ingen konklusjon.

### 2.5.6 Eksempel

Vi fortsetter med apesadelen fra 2.5.2.  $f_x = 0 \Leftrightarrow x = 0 \vee y = 0$  og  $f_y = 0 \Leftrightarrow x = 0$ . Når  $x=0$  er både  $f_x = f_y = 0$  og vi får stasjonære punkter altså for  $x=0$ .

De andreordens partiellderiverte blir  $f_{xx} = 4y$ ,  $f_{xy} = 4x$ ,  $f_{yx} = 4x$ ,  $f_{yy} = 0$ . Vi prøver med å klassifisere det stasjonære punktet ved hjelp av Hessematrisen. Vi får  $\Delta = 0$  og ingen konklusjon fra metoden.

### 2.5.7 Eksempel

Gitt  $f(x, y) = x^2y^2 - 2xy^2 + y^2 - 1$ . Finn de stasjonære punktene og klassifiser dem.

$$f_x = 2xy^2 - 2y^2 = 2y^2(x - 1) \text{ og } f_y = 2x^2y - 4xy + 2y = 2y(x^2 - 2x + 1)$$

$$f_x = 0 \Rightarrow y = 0 \vee x = 1$$

$$f_y = 0 \Rightarrow y = 0 \vee x = 1$$

Dette gir stasjonære punkter for  $x=1$  (samme hva  $y$  er) og for  $y=0$  (samme hva  $x$  er).

$$f_{xx} = 2y^2, \quad f_{xy} = 4y(x - 1), \quad f_{yy} = 2x^2 - 4x + 2 = 2(x - 1)^2$$

Vi setter inn  $(1,0)$  og får  $\Delta = 0$ , så heller ikke her gir metoden noen konklusjon.

## 2.6 Dobbelintegral

Dobbelintegraler kan tolkes som volumet under en flate, regnet over et avgrenset område i  $xy$ -planet. Hvis vi har en funksjon  $z = f(x, y)$  definert på et område  $A$  i  $xy$ -planet, kan vi finne volumet avgrenset av flaten  $z = f(x, y)$  ved å regne ut integralet

$$\int \int_A f(x, y) \, dx dy$$

Som et eksempel skal vi se hvordan vi regner ut volumet under  $f(x, y) = \sin x \sin y$  over området  $[0, \frac{\pi}{2}] \times [0, \frac{\pi}{2}]$ . Når vi setter opp dobbeltintegralet, får vi dette uttrykket:

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \sin x \sin y \, dx dy$$

Det innerste integralet blir

$$\int_0^{\frac{\pi}{2}} \sin x \sin y \, dx = [-\cos x \sin y]_0^{\frac{\pi}{2}} = 0 - (-1 \cdot \sin y) = \sin y, \text{ slik at}$$

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \sin x \sin y \, dx dy = \int_0^{\frac{\pi}{2}} \sin y \, dy = [-\cos y]_0^{\frac{\pi}{2}} = 0 - (-1) = 1.$$

### 2.6.1 Eksempel

$$\int_0^1 \int_0^2 x^2 y \, dx dy = \int_0^1 \left[ \frac{x^3}{3} y \right]_0^2 dy = \int_0^1 \left[ \frac{8}{3} y \right]_0^2 dy = \left[ \frac{8}{3} \frac{y^2}{2} \right]_0^1 = \frac{4}{3}$$

### 2.6.2 Eksempel

Integralet  $\int_0^2 \int_0^1 y x^2 dy dx$  gir samme resultat. Det er fordi vi kan bytte om  $dx$  og  $dy$  hvis vi samtidig bytter om integralene med integrasjonsgrenser.

Det er litt vanskeligere når vi ikke har et rektangulært område å integrere over.

### 2.6.3 Integrasjon over ikke-rektangulært område

Anta at vi nå skal integrere over området i  $xy$ -planet avgrenset av  $x$ -aksen,  $y$ -aksen og den rette linjen  $x+y=1$ . Vi kan jo ikke la både  $x$  og  $y$  gå fra 0 til 1, da integrerer vi over et kvadratisk område. Vi setter  $y=1-x$  og kan bruke det som øvre grense for integralet med hensyn på  $y$ . Her er det det innerste integralet. Hvis vi integrerer med hensyn på  $x$  først, det vil si bytter om på  $dx$  og  $dy$ , blir det tilsvarende  $1-y$  som øvre grense.

$$\begin{aligned} \int_0^1 \int_0^{1-x} x^2 y dy dx &= \int_0^1 \frac{1}{2} x^2 (1-x)^2 dx \\ &= \left[ \frac{1}{10} x^5 - \frac{1}{4} x^4 + \frac{1}{6} x^3 \right]_0^1 dy \\ &= \frac{1}{10} - \frac{1}{4} + \frac{1}{6} \\ &= \frac{1}{60} \end{aligned}$$

Se også [6], 11.4.2-11.4.3.

### 2.6.4 Eksempel: Numerisk integrasjon

Integraler må ofte regnes ut numerisk. Når lengden  $dx = dy = h$  på hver kant på er liten, blir arealet  $dA = dx \cdot dy$  lite. Vi tenker oss hele området A oppdelt i kvadrater  $A_{ij}$  med midtpunkter  $P_{ij} = (x_i, y_j)$ . Vi kan regne ut volumet  $V_i$  av en søyle over ett kvadrat ved midtpunktmetoden:

$$V_{ij} = h^2 \cdot f(x_i, y_j)$$

Vi regner altså ut volumet av veldig små søyler. Når vi gjentar dette og summerer alle søylene, får vi en approksimasjon til et bestemt dobbeltintegral:

$$\int \int_A f(x, y) dx dy \approx V = \sum_{ij} V_{ij} = h^2 \cdot f(x_i, y_j)$$

Listing 2.2: Numerisk integrasjon med midtpunktmetoden

```
double f(double x, double y) { return x*x*y; }

int main() {
    double h{0.10};
    const double EPS{0.001};
    double xmax{2.0};           // Ovre integrasjonsgrense
    double ymax{1.0};           // Ovre integrasjonsgrense
    double volum{0.0};

    for (double x=0.05; x<xmax+EPS; x+=h)
        for (double y=0.05; y<ymax+EPS; y+=h)
            volum = volum + h*h*f(x, y);

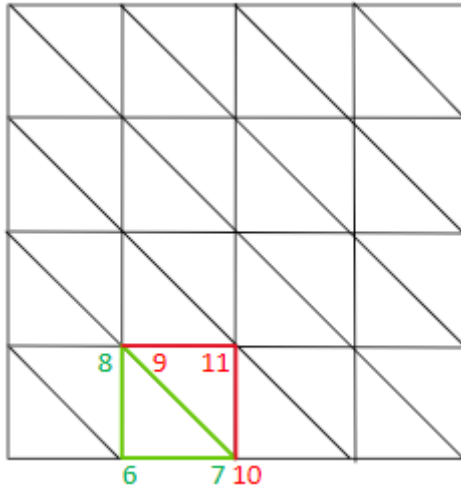
    cout << "Volumet: " << volum << endl; // blir 1.333 med h=0.10;
    return 0;
}
```

Se også oppgave 2.8.14.

## 2.7 Programmering - tegne grafen til $f(x,y)$

Vi skal se på et eksempel på hvordan vi kan tegne grafen til en funksjon  $f(x,y)$ . Vi skal konstruere og tegne en 3d flate som består av trekanter. I eksemplet nedenfor er området  $[0, 1] \times [0, 1]$  delt opp i  $4 \times 4$  kvadrater. Det gir 32 trekanter, og  $32 \cdot 3 = 96$  vertices. Funksjonen  $z = f(x, y) = \sin \pi x \cdot \sin \pi y$  beregnes i hvert vertex. Disse legges inn i en `std::vector` etter hverandre slik at tre og tre vertices definerer en trekant for `glDrawArrays()`. Den grønne og den røde trekanten på figur 2.2 viser indekseringen for de gitte vertexene i en `std::vector mVertices` for  $x = x_{min} + h$ ,  $y = y_{min}$  i indre løkke.

```
float xmin=0.0f, xmax=1.0f, ymin=0.0f, ymax=1.0f, h=0.25f;
for (auto x=xmin; x<xmax; x+=h)
    for (auto y=ymin; y<ymax; y+=h)
    {
        float z = sin(M_PI*x)*sin(M_PI*y);
        mVertices.push_back(Vertex{x,y,z,x,y,z});
        z = sin(M_PI*(x+h))*sin(M_PI*y);
```



Figur 2.2: Regulær triangulering av  $[0,1] \times [0,1]$

```

mVertices.push_back(Vertex{x+h,y,z,x,y,z});
z = sin(M_PI*x)*sin(M_PI*(y+h));
mVertices.push_back(Vertex{x,y+h,z,x,y,z});
mVertices.push_back(Vertex{x,y+h,z,x,y,z});
z = sin(M_PI*(x+h))*sin(M_PI*y);
mVertices.push_back(Vertex{x+h,y,z,x,y,z});
z = sin(M_PI*(x+h))*sin(M_PI*(y+h));
mVertices.push_back(Vertex{x+h,y+h,z,x,y,z});
}

```

Funksjonen kan lett byttes ut. Metoden kan også brukes til å lage en plan flate med  $z = f(x,y)=0$ .

### 2.7.1 Franke's testfunksjon

Franke's testfunksjon er en mye brukt funksjon i numerisk analyse. Den ser slik ut:

$$\begin{aligned}
 f(x,y) &= \frac{3}{4}e^{-((9x-2)^2+(9y-2)^2)/4} \\
 &+ \frac{3}{4}e^{-((9x+1)^2/49-(9y+1))/10} \\
 &+ \frac{1}{2}e^{-((9x-7)^2+(9y-3)^2)/4} \\
 &- \frac{1}{5}e^{-((9x-4)^2+(9y-7)^2)}
 \end{aligned} \tag{2.7}$$

Det krever litt konsentrasjon å implementere dette funksjonsuttrykket, men det er det verdt, prøv!

## 2.8 Oppgaver

### 2.8.1

- Vis hvordan stegene utføres med denne algoritmen for  $x = 3, f(x) = 4x^4 + 8x^2 + x + 2$ .
- Forklar hvordan algoritmen fungerer.
- Hva blir kjøretiden?

### 2.8.2

I denne oppgaven skal du lage datapunkter for grafen til en funksjon, og punktene skal lagres på en tekstfil med format som i avsnitt 2.3.1.

- Velg en egen funksjon  $f(x)$ , eller et datasett på fil. Velg en passende definisjonsmengde  $D_f = [a, b]$ .
- Velg antall intervaller  $n$  slik at du får  $n + 1$  punkter  $x_0, x_1, \dots, x_n$  og oppløsning  $h = \Delta x = \frac{b-a}{n}$ .
- Regn ut funksjonsverdier  $f(x_i)$  i hvert punkt. Regn også ut  $f'(x)$  eller Newton's differenskvotient i hvert punkt.
- Bestem en farge for hvert verteks slik at grafen får en farge (for eksempel grønn) når funksjonen er stigende, og en annen farge (for eksempel rød) når funksjonen er avtakende.
- Beregn funksjonsverdi og skriv ut verteksdata for hvert punkt på en tekstfil. På første linje i filen skal antall punkter stå. Deretter følger verteksdata for hvert punkt linje for linje.

### 2.8.3

Lag en fil med datapunkter for en tredimensjonal spiral, som forklart i avsnitt 2.4. Bestem selv en steglengde for parameteren  $t$  og verteksfarger. Benytt samme filformat som i forrige oppgave.

### 2.8.4

Finn de stasjonære punktene til funksjonene nedenfor:

- $f_1(x, y) = xy$
- $f_2(x, y) = x^2 + y^2$
- $f_3(x, y) = \sin xy$
- $f_4(x, y) = \sin x \cdot \sin y$

### 2.8.5

Finn tangentplanet til funksjonen  $f(x, y) = x^2 + y^2$  i punktet  $(\frac{1}{2}, \frac{1}{2})$ .

### 2.8.6

Regn ut

$$\int_3^6 \int_{-2}^1 (x^2 + xy^2) dx dy$$

### 2.8.7

Gitt  $f(x, y) = e^{x+y}$ . Regn ut volumet avgrenset av grafen til  $f$  og de rette linjene  $x \geq 0$ ,  $y \geq 0$ ,  $x + 2y \leq 3$ .

### 2.8.8

Regn ut volumet avgrenset av grafen til funksjonen  $f(x, y) = xy$ , x-aksen, y-aksen og den rette linjen  $x+y=1$ .

### 2.8.9

Gitt funksjonen  $h(x, y) = xy - x - y + 1$ .

- Finn  $\frac{\partial h}{\partial x}$  og  $\frac{\partial h}{\partial y}$  og bestem det stasjonære punktet.
- Finn de andreordens partiellderiverte, og bruk disse til å klassifisere det stasjonære punktet.
- Regn ut volumet avgrenset av grafen til  $h$ , x-aksen, y-aksen og den rette linjen  $y = 1 - x$ .

### 2.8.10

Gitt funksjonen  $f(x, y) = 3x - x^3 - 3xy^2$ .

- Finn lineariseringen til funksjonen  $f$  i punktet  $(-1, 1)$ .
- Regn ut de andreordens partiellderiverte til  $f(x, y)$ .
- Finn og klassifiser de kritiske punktene til funksjonen  $f$ .

### 2.8.11

Gitt funksjonen  $g(x, y) = 1 - x - y$  og la  $A$  være området i  $xy$ -planet avgrenset av x-aksen, y-aksen og den rette linjen  $x + y = 1$ .

Regn ut  $\int \int_A g(x, y) dA$ .

### 2.8.12

Gitt funksjonen  $f(x, y) = (1 - x)(1 - y)$ .

- Bestem normalen (med lengde 1) til  $f$  i punktet  $(\frac{3}{4}, \frac{3}{4})$ .
- Regn ut volumet som er avgrenset av grafen til  $f$ , x-aksen, y-aksen og den rette linjen  $x + y = 1$ .

### 2.8.13

1. Velg en egen funksjon av to variable, et område i xy-planet (eller xz-planet), og en oppdeling. Bruk algoritmen i 2.7 og lag en tekstfil med xyz (eventuelt også rgb og uv-koordinater) linjevis. Antall linjer skal stå øverst i fila. Fila skal kunne leses inn av et TriangleSurface objekt i Qt/3d-programmering.
2. Utvid algoritmen i 2.7 til også å regne ut enhetsnormalen  $\vec{n} = [n_x, n_y, n_z]$  i hvert punkt. Lag en tekstfil med x y z  $n_x$   $n_y$   $n_z$  (eventuelt også uv-koordinater) linjevis. Antall linjer skal stå øverst i fila.
3. Modifiser draw()-funksjonen i TriangleSurface til å tegne enhetsnormalen i hvert vertex.

Regn ut numerisk det samme integralet som i oppgave 2.6.3.

### 2.8.14

1. Regn ut det samme integralet som i oppgave 2.8.11 numerisk: Regn ut og summer integralet over alle trekantene du har laget i planet.
2. Gjenta forrige oppgave med halvert avstand mellom alle vertices.

## Kapittel 3

# Vektorer og matriser

### 3.1 Introduksjon

Vi starter dette kapitlet med litt repetisjon om vektorer og vektorrom. Vi skal senere studere praktisk lineær algebra, bruk av vektorer og matriser. Vi skal se på metoder som vi får direkte nytte av både til implementering av 3d-grafikk og i kapittel 4.

### 3.2 Vektorrom

Et vektorrom  $\mathbb{V}$  er en mengde av elementer kalt vektorer hvor operasjonene addisjon og skalarmultiplikasjon er definert. I tillegg fins et nullelement,  $\mathbf{0}$ , og assosiativ, kommutativ og distributiv lov gjelder. Hvis dimensjonen til vektorrommet er  $n$ , slik at  $\mathbb{V} = \mathbb{R}^n$ , er vektorene  $n$ -tupler med koordinater,  $\mathbf{v} = [v_1, \dots, v_n]$ .

Vi har altså

$$\text{Hvis } a \in \mathbb{R} \text{ og } \mathbf{v} \in \mathbb{V}, \text{ er } a\mathbf{v} \in \mathbb{V} \text{ og hvis } \mathbf{u}, \mathbf{v} \in \mathbb{V}, \text{ er } \mathbf{u} + \mathbf{v} \in \mathbb{V} \quad (3.1)$$

Se for eksempel [10] for en komplett formell definisjon av vektorrom.

En basis for et vektorrom  $\mathbb{R}^n$  består av  $n$  vektorer  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$  som utspenner  $\mathbb{R}^n$ . Dette er ekvivalent med at vektorene  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$  er lineært uavhengige,  $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_n\mathbf{v}_n = \mathbf{0} \Leftrightarrow a_1 = a_2 = \dots = a_n = 0$  for reelle tall  $a_i$ .

Notasjon: Vi skriver vektorer med fet skrift her. Med koordinater:  $\mathbf{v} = [v_1, v_2, v_3]$  og  $\mathbf{v} = [v_1 \ v_2 \ v_3]$  brukes om radvektorer, mens

$$\mathbf{v} = [v_1 \ v_2 \ v_3]^T \text{ og } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{ brukes om kolonnevektorer.}$$

Vektorene  $\mathbf{e}_1 = [1 \ 0 \ 0]^T$ ,  $\mathbf{e}_2 = [0 \ 1 \ 0]^T$  og  $\mathbf{e}_3 = [0 \ 0 \ 1]^T$  er lineært uavhengige vektorer som utgjør den vanligste basisen for vektorrommet  $\mathbb{R}^3$ . Vi kaller denne standardbasisen  $\mathbf{E}$  eller  $\mathbf{E}^n$ .



En vektor  $\mathbf{v} = [2, 1, 3]$  kan skrives som en lineærkombinasjon av basisvektorer,  $\mathbf{v} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3$ . Her er  $a_1 = 2$ ,  $a_2 = 1$ ,  $a_3 = 3$ :

$$\mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Vi skal nå se et eksempel på hvordan vi bruker en annen vektorbasis for  $\mathbb{R}^3$ . Vektorene  $\mathbf{v}_1 = [1 \ 0 \ 1]^T$ ,  $\mathbf{v}_2 = [1 \ 1 \ 1]^T$  og  $\mathbf{v}_3 = [0 \ 0 \ 1]^T$  er også lineært uavhengige vektorer og utgjør en annen basis for vektorrommet  $\mathbb{R}^3$ . Vi kan kalle denne basisen  $\mathbf{A}$ , altså er  $\mathbf{A} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ . (Dette navnet samsvarer med matrisen  $\mathbf{A}$  nedenfor).

Vektoren  $\mathbf{v} = [2, 1, 3]$  skrevet som en lineærkombinasjon av disse basisvektorene blir,  $\mathbf{v} = b_1\mathbf{v}_1 + b_2\mathbf{v}_2 + b_3\mathbf{v}_3$ . Her er  $b_1 = 1$ ,  $b_2 = 1$ ,  $b_3 = 1$ :

$$\mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Dette kan skrives på matriseform  $\mathbf{v} = \mathbf{A}\mathbf{b}$  hvor  $\mathbf{A} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3]^T$  og  $\mathbf{b} = [b_1 \ b_2 \ b_3]^T$ ,

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Kolonnene i  $\mathbf{A}$ -matrisen består altså av basisvektorene, og vektoren  $\mathbf{b}$  består av tallene som vi multipliserer hver basisvektor med. Vektorene  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  og  $\mathbf{v}_3$  er lineært uavhengige dersom

$$b_1\mathbf{v}_1 + b_2\mathbf{v}_2 + b_3\mathbf{v}_3 = \mathbf{0} \Leftrightarrow b_1 = b_2 = b_3 = 0$$

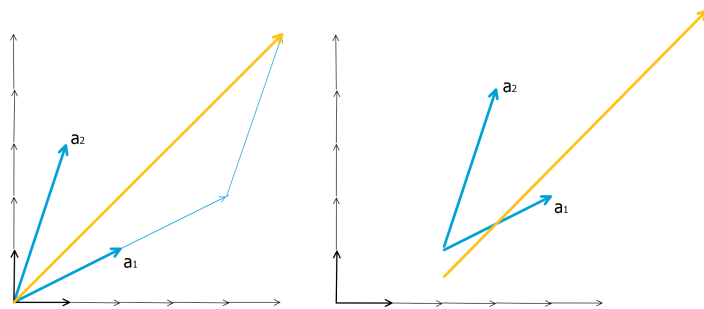
som er ekvivalent med at  $\mathbf{A}$  er invertibel,  $\det(\mathbf{A}) \neq 0$ . Dette eksemplet forteller oss:

- Vektoren  $\mathbf{v}$  er en lineærkombinasjon av basisvektorer.
- Koordinatene til  $\mathbf{v}$  uttrykt ved basisfunksjonene  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  og  $\mathbf{v}_3$  er  $[1, 1, 1]$ .
- Når vi regner ut  $b_1\mathbf{v}_1 + b_2\mathbf{v}_2 + b_3\mathbf{v}_3$ , får vi koordinatene til  $\mathbf{v}$  i standardbasen  $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ .

### 3.3 Invers transformasjon

Vi skal nå se hvordan vi kan transformere en vektor fra standardbasen  $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  til vektorbasen  $\mathbf{A} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ . Vi skal altså finne koordinatene i  $\mathbf{v}_1$ ,  $\mathbf{v}_2, \mathbf{v}_3$  for en vektor i  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ . La oss bruke  $\mathbf{b} = [1, 2, 3]$  som et eksempel.  $\mathbf{b}$  har koordinatene 1, 2 og 3 i standardbasen  $\mathbf{E}$  for  $\mathbb{R}^3$  og vi skal finne koordinatene til denne vektoren  $\mathbf{y}$  i vektorbasen  $\mathbf{A} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ . Vi har

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$



Figur 3.1: To vektorbasiser for  $\mathbb{R}^2$ . Vektorer har ikke posisjon, så lineærkombinasjonen (den orange vektoren) blir den samme i hvert tilfelle.

og

$$\mathbf{x} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}$$

Kontroller at dette stemmer ved å regne ut  $\mathbf{Ax}$ .

### 3.4 Vektorer, punkt og frames

La oss se på et eksempel i  $\mathbb{R}^2$ . Vi bruker nå basisvektorene  $[2 \ 1]^T$  og  $[1 \ 3]^T$  og vil finne koordinatene i denne basisen til en vektor  $[5, 5]$  i standardbasis.

Vi får her

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \text{ og } \mathbf{A}^{-1} = \begin{bmatrix} \frac{3}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{2}{5} \end{bmatrix}$$

slik at

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

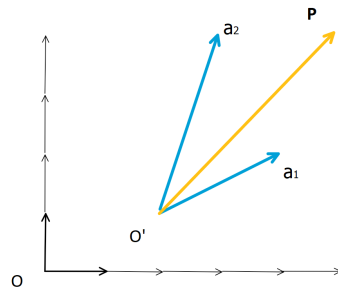
som illustrert på figur 3.1. Vi ser fra høyre del av figur 3.1 at det er vanskeligere å lese av koordinatene til den orange vektoren med hensyn på den ene eller den andre basisen. Det er enklere hvis vi flytter vektoren slik at den starter i origo.

Vi studerer nå figur 3.2. Punktet P, som den orange vektoren peker til, har koordinater (5,4) i det svarte (ortogonale) koordinatsystemet. Vi skal beregne koordinatene til P i det blå koordinatsystemet, uttrykt som en vektor  $\mathbf{x}$ . Her har vi altså to vektorbasiser for  $\mathbb{R}^2$  med hvert sitt origo og hvert sitt koordinatsystem (på engelsk brukes gjerne frame):

- Ett koordinatsystem K med origo O og standardbasisen  $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2\}$
- Ett koordinatsystem K' med origo O' og vektorbasis  $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2\}$

Vi translerer først fra origo O til O':  $\mathbf{v} = [5, 4] - [2, 1] = [3, 3]$ . Deretter,

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \begin{bmatrix} 3 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{6}{5} \\ \frac{3}{5} \end{bmatrix}$$



Figur 3.2: To vektorbasiser for  $\mathbb{R}^2$  med hvert sitt origo og et punkt P.

Vi får behov for en type matematisk rom som inneholder både vektorer og punkt. Dette kalles et affint rom og vi kommer nærmere inn på det i crefaffint.

## 3.5 Matriser

I denne seksjonen skal vi først repetere de lineære transformasjoner rotasjon, skalering og translasjon fra [6]. Vi skal videre repetere hvordan vi løser lineære ligningssystemer ved hjelp av Gausseliminering og gå gjennom hvordan dette implementeres ved LU-faktorisering, i Gausseliminering. Vi kommer til å bruke kvadratiske 4x4 matriser i eksemplene, men for eksempel LU-faktorisering er en numerisk lineær algebra-metode som akn brukes på større ligningssystemer.

Dette er kjent stoff fra [6] (kapittel 9.5, side 486-488), med figurer som viser de aller fleste aktuelle transformasjoner vi får bruk for. Vi skal nå anvende dette og programmere noe av det sjøl. Senere skal vi studere teorien bak *litt* nærmere.

### 3.5.1 Rotasjoner

Vi husker at når et punkt  $(x, y) \in \mathbf{R}^2$  roteres en vinkel  $\theta$  om origo, så kan vi finne de nye koordinatene til punktet ved å multiplisere matrisen

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

med  $[x \ y]^T$ , slik at de nye koordinatene blir

$$x' = x \cos \theta - y \sin \theta \text{ og } y' = x \sin \theta + y \cos \theta$$

Hvis for eksempel  $\theta = \frac{\pi}{4}$ , blir rotasjonsmatrisen

$$\mathbf{R}\left(\frac{\pi}{4}\right) = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

#### Øving

Hva blir koordinatene (1,1), (2,1), (2,2) og (1,2) rotert 45 grader om origo?

Rotasjon om origo i planet er det samme som rotasjon om z-aksen. Z-aksen har normalvektor  $[0 \ 0 \ 1]^T$ . Hvis vi tilføyer en dimensjon, blir rotasjonsmatrisen for rotasjon en vinkel  $\theta$  om z-aksen

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Vi ser at normalvektoren som det roteres om står i tredje kolonne. Tilsvarende matriser for rotasjon om x-aksen og y-aksen blir

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

og

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

### 3.5.2 Homogene koordinater

Det står litt om homogene koordinater i 2d i [6] kapittel 9.5. Ved å bruke homogene koordinater kan translasjon uttrykkes som en matrise multiplisert med en vektor. Når vi bruker homogene koordinater i 3d, bruker vi 4x4 matriser til transformasjoner i 3d. Med homogene koordinater blir rotasjonsmatrisene ovenfor

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Vi gjenkjenner de opprinnelige rotasjonsmatrisene som øvre venstre  $3 \times 3$  submatriser. Vi må passe på at den endelige transformasjonsmatrisen må sendes til shaderprogrammet med kolonnevise data, se 3.5.8.

### 3.5.3 Skalering

$4 \times 4$  matrisen for skalering med faktorer  $s_x$ ,  $s_y$  og  $s_z$  i henholdsvis x-retning, y-retning og z-retning er

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

### 3.5.4 Translasjon

Den homogene  $4 \times 4$  matrisen for å representere translasjonen

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} x + dx \\ y + dy \\ z + dz \end{bmatrix}$$

er

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Vi skal kort repetere hvordan vi bruker denne matrisen.

## Øving

1. Vektoren  $[2 \ 2 \ 1]^T$  skal translères med  $[1 \ 2 \ 3]^T$ . Hva blir translasjonsmatrisen i homogene koordinater? Kontroller ved å regne ut.
2. Vektoren  $[4 \ 5 \ 6]^T$  skal translères med  $[-1 \ 1 \ 1]^T$ . Hva blir translasjonsmatrisen i homogene koordinater? Kontroller ved å regne ut.

## Løsning

I homogene koordinater er siste koordinaten til en vektor 0 og siste koordinaten til et punkt 1. Vektoren  $[2 \ 2 \ 1]^T$  må utvides til  $[2 \ 2 \ 1 \ 0]^T$  i homogene koordinater. Translasjonsmatrisen multiplisert med vektoren blir

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Dette er samme vektoren som før, ingenting er endret etter multiplikasjonen. Men det er jo slik det skal være. En vektor har lengde og retning, og den skal ikke endres selv om vi prøver å flytte posisjonen.

Skal vi derimot translere punktet  $(2, 2, 1) \in \mathbf{R}^3$ , må vi føye til 1 som siste koordinat. Vi får da

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 4 \\ 1 \end{bmatrix}$$

### 3.5.5 Refleksjon, projeksjon og forskyvning

Refleksjon, projeksjon og forskyvning er forklart i [6]. Refleksjon er nyttig for eksempel ved overgang fra world koordinater til skjermkoordinater i systemer hvor y-aksen peker nedover på skjermen. Forskyvning kan er nyttig ved innlesing av geodata med misvisning til et kartesisk koordinatsystem.

### 3.5.6 Sammensatte transformasjoner

Vi kan gjøre både translasjon, rotasjon og skalering, men rekkefølgen (av rotasjon og translasjon) spiller en rolle for resultatet. Det er for eksempel en vanlig erfaring i 3d programmering at man sender et objekt i bane når man egentlig har tenkt å rotere det.

#### Rotasjon om et punkt

Man kan rotere om et annet punkt enn origo, om for eksempel z-aksen. Dette kan gjøres ved en sammensatt transformasjon: Translater til origo, roter om z-aksen og translater tilbake ([1], 4.10.1).

### Generell rotasjon

I [1], 4.10.2 beskrives også hvordan en generell rotasjon kan kombineres ved rotasjon om de tre aksene,

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$

### 3.5.7 Programmering

Disse tre transformasjonene skal implementeres i en  $4 \times 4$  matriseklasse med funksjoner **rotate**, **translate** og **scale**, med standard parametre. Vi trenger ikke overlaste disse funksjonene.

### 3.5.8 C++ matriserepresentasjon

Todimensjonale arrayer i C++ ligger lagret radvis, som eksemplet nedenfor viser:

```
#include <iostream>
struct Matrix4x4 {
    float A[4][4];
    Matrix4x4() {
        int k=0;
        // legger inn data radvis
        for (int i=0;i<4;i++)
            for (int j=0;j<4;j++)
                A[i][j] = ++k;
    }
    void print() {
        // utskrift viser at i en todimensjonal
        // array ligger data radvis
        float* tmp=A[0];
        for (int i=0;i<16;i++) {
            std::cout << (*tmp)++ << " ";
        }
    }
};
int main(){
    Matrix4x4 m;
    m.print();
}
```

Vi må huske på dette når vi skal sende data til shaderprogrammet - da skal vi sende kolonnevis data. Klassen ovenfor er forøvrig en veldig forenklet versjon.

### 3.5.9 Viewing

Det er vanlig i grafikkprogrammering API-er å ha en *LookAt()* funksjon. En *LookAt()*-funksjon har følgende parametre:

eye -  $\mathbf{e} = [e_x, e_y, e_z]$  - et punkt som angir kameraposisjonen.

at -  $\mathbf{a} = [a_x, a_y, a_z]$  - et punkt som angir fokus.

$\mathbf{up} - \mathbf{w} = [w_x, w_y, w_z]$  - en vektor som opp-retningen til kameraet.

Ut fra dette skal vi generere en matrise som transformerer world koordinater til kamera frame. Det er hensiktsmessig å bruke betegnelser  $\mathbf{F}$  for world koordinater og  $\mathbf{F}'$  for kamera frame. Kamera frame er et høyrehånds koordinatsystem med kameraposisjonen (*eye*) i origo. To ortogonale vektorer  $\mathbf{u}$  og  $\mathbf{v}$  ligger i viewplane, det vi kan sammenligne med bildebrikkeplanet/filmplanet. Normalen på viewplane, eller retningsvektoren for objektivet i en syntetisk kameramodell, blir

$$\mathbf{n} = \frac{\mathbf{a} - \mathbf{e}}{\|\mathbf{a} - \mathbf{e}\|}$$

Kryssproduktet av  $\mathbf{w}$  og  $\mathbf{n}$  blir en vektor  $\mathbf{u}$  som ligger i viewplane,

$$\mathbf{u} = \frac{\mathbf{w} \times \mathbf{n}}{\|\mathbf{w} \times \mathbf{n}\|}$$

og

$$\mathbf{v} = \frac{\mathbf{n} \times \mathbf{u}}{\|\mathbf{n} \times \mathbf{u}\|}$$

Vi har nå et ortogonalt høyrehånds koordinatsystem for kamera frame  $\mathbf{F}'$  som består av vektorene  $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ .

$$\begin{bmatrix} u_x & v_x & n_x \\ u_y & v_y & n_y \\ u_z & v_z & n_z \end{bmatrix}$$

Matrisen  $\mathbf{M}$  med disse tre vektorene som kolonnevektorer transformerer et punkt i kamera frame  $\mathbf{F}'$  til world koordinater  $\mathbf{F}$ . I homogene koordinater blir dette

$$M = \begin{bmatrix} u_x & v_x & n_x & 0 \\ u_y & v_y & n_y & 0 \\ u_z & v_z & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vi ønsker å gå motsatt vei - transformere et punkt i world koordinater  $\mathbf{F}$  til kamera frame  $\mathbf{F}'$ , og vi må da bruke den inverse matrisen  $\mathbf{M}^{-1}$ . I tillegg må vi gjøre en translasjon fra origo i world koordinater til origo i kamera frame, som er parameteren *eye*. Se også avsnitt 3.4. Gitt  $\mathbf{eye} = [eye_x, eye_y, eye_z]$  og en translasjonsmatrise i homogene koordinater

$$T = \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De vanlige parametrene til `LookAt()` er kameraposisjonen *eye*, fokuseringspunkt *at* og kameraorienteringen *up*. Vi bruker disse til å lage koordinatsystemet i kameraframe:

```
void Matrix4x4::lookAt(const Vec3 &eye, const Vec3 &at, const Vec3 &up) {
    Vec3 v = up;
    v.normalize();           // y-akse i kameraframe
```



```

Vec3 n = eye-at;
n.normalize();           // z-akse i kameraframe
Vec3 u = v.cross(n);
u.normalize();           // x-akse i kameraframe
v = n.cross(u);
v.normalize();

```

Deretter kan vi lage matrisen i LookAt()-funksjonen, som blir  $\mathbf{M}^{-1}\mathbf{T}$ :

$$\begin{bmatrix} u_x & u_y & u_z & -(eye, u) \\ v_x & v_y & v_z & -(eye, v) \\ n_x & n_y & n_z & -(eye, n) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En mer detaljert forklaring på dette er gitt i [1], kapittel 5.

### 3.5.10 Perspektivprojeksjon

Standard parametre til *frustum()*-funksjonen er *left*, *right*, *bottom*, *top*, *near*, *far*. Her er vel de fleste parametrene selvforklarende. Frustum-matrisen blir slik:

$$\begin{bmatrix} 2\frac{near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & 2\frac{near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -2 \cdot far \cdot near(far - near) \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Denne matrisen skal multipliseres med matrisen fra avsnitt 3.5.9 multiplisert med en vektor i homogene koordinater.

Standard parametre til *perspective()*-funksjonen er *fovy*, *aspectRatio*, *near*, *far*. Med beregningene

```

float top = near*std::tan(fovy);
float right = top/aspectRatio;

```

blir perspektivprojeksjonsmatrisen slik:

$$\begin{bmatrix} \frac{near}{right} & 0 & 0 & 0 \\ 0 & \frac{near}{top} & 0 & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -2 \cdot far \cdot near(far - near) \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Denne funksjonen foretrekkes ofte i stedet for frustum-funksjonen.

### 3.5.11 Andre funksjoner i matriseklasse

Vi får også bruk for blant annet å sette en matrise til identitetsmatrisen, , finne den transponerte matrisen ,finne den inverse matrisen og multiplisere to matriser. Vi trenger altså 4x4 matriser til transformasjoner, og vi vil få behov for å løse 4x4 ligningssystemer. Når vi implementerer en metode for å løse 4x4 ligningssystemer, kan vi egentlig løse større ligningssystemer med tilnærmet uendret kode. Siden vi trenger 4x4 matriser til lineære transformasjoner, passer det bra for oss å slå alt sammen til en Matrix4x4 klasse.

## 3.6 Gausseliminering

Vi får bruk for Gausseliminering til å løse ligningssystemer, spesielt 4x4 ligningssystemer:

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (3.7)$$

Skrevet på koordinatform:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Funksjonen LU() nedenfor lager en øvre triangulær matrise, men diagonalelementene er ikke nødvendigvis 1 som i matrisen på trappeform i [6], kapittel 9.

### 3.6.1 Rader, kolonner og b-vektor

Nedenfor brukes  $m$  om rader og  $n$  om kolonner, som vanlig. Det som ikke framgår av algoritmen nedenfor er radoperasjoner på høyresiden (b-vektor). Den kan legges til som en ekstra kolonne i matrisen, slik at vi får  $m=4$  og  $n=5$ . Da er første del av Gausselimineringen nesten i samsvar med [6] bortsett fra at vi ikke får 1-ere på diagonalen.

### 3.6.2 LU-faktorisering

Dette blir litt ulogisk siden klassen heter Matrix4x4. Dessuten er det en fordel med å ikke bruke en utvidet matrise og radoperasjoner på denne: Det er veldig ofte slik at man får bruk for å løse ligningen 3.7 for flere ulike vektorer  $\mathbf{b}$ . Derfor er det bedre å gjøre radoperasjoner på  $\mathbf{b}$  i del to av algoritmen.

LU-faktoriseringen tar utgangspunkt i elementære radoperasjoner kjent fra [6]:

1. Multiplisere en rad med et tall.
2. Addere et multiplum av en rad til en annen rad.
3. Bytte rekkefølgen av to rader (nedenfor kalt pivotering).

### 3.6.3 Eksempel

Matrisen nedenfor er den utvidede 3x4 matrisen til et ligningssystem  $\mathbf{Ax}=\mathbf{b}$ . Radoperasjoner på denne i henhold til [6], kapittel 9 gir

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 2 & 1 \end{bmatrix} \xrightarrow{\substack{R_2 - 2R_1 \\ R_3 - 3R_1}} \begin{bmatrix} 1 & 2 & 3 & 1 \\ 0 & -1 & -2 & -1 \\ 0 & -2 & -7 & -2 \end{bmatrix} \xrightarrow{R_3 - 2R_2} \begin{bmatrix} 1 & 2 & 3 & 1 \\ 0 & -1 & -2 & -1 \\ 0 & 0 & -3 & 0 \end{bmatrix}$$

Vi ser her at vi ikke har 1-ere på diagonalen, men det er likevel enkelt å løse ligningssystemet når vi har det på denne formen. Fra siste ligning finner vi  $z=0$ , og når vi setter inn det i andre ligning finner vi  $y=1$ . Vi gjentar for øverste ligning og finner  $x=-1$ . Vi gjør en ekstra divisjon i hvert steg her sammenlignet

med å ha 1-ere på diagonalen. Denne prosessen svarer til solve() nedenfor.

Når vi nuller ut elementene i nedre triangulære del, blir de ikke tilordnet 0 - vi bruker plassen til å lagre koeffisienter i stedet. Så hvis man studerer utskriften av en matrise etter LU-faktorisering, kan man bli forvirret over at det ikke er 0 i nedre triangulære del.

### 3.6.4 LU-faktorisering algoritme

Algoritmen nedenfor lager en øvre triangulær matrise *uten* 1-ere på diagonalen og *uten* å tilordne 0 under diagonalelementene. I stedet brukes denne plassen til å lagre faktorene som vi multipliserer en og en rad med for å nulle ut elementene under diagonalelementet.

```
void Matrix4x4::LU()
{
    for (int k=0; k<m-1; k++)
    {
        // pivot(k);

        // Ved radoperasjoner skal vi oppnå 0-ere under diagonalelementet
        // i alle rader nedenfor (altså kolonne k
        // Vi subtraherer et multiplum av k-te
        // rad fra radene nedenfor, nuller ut kolonner fra venstre

        for (int i=k+1; i<m; i++) {
            // Skal mult med denne og deretter trekke fra rad i
            // denne skal bli null og vi kan lagre faktoren her
            A[i][k] = A[i][k]/A[k][k];

            for (int j=k+1; j<n; j++) {
                // kolonnene til høyre for den som blir nullset ut
                A[i][j] = A[i][j] - A[i][k]*A[k][j];
            }
        }
    }
}
```

Algoritmen nedenfor løser deretter ligningssystemet  $\mathbf{Ax}=\mathbf{b}$  ved å bestemme verdien for en variabel om gangen, med start nedenfra. Vi skriver  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  som A.solve(b). Det er her en forutsetning at A er LU-faktorisert.

```
Vec4 Matrix4x4::solve(Vec4 &b) const
{
    Vec4 x;

    for (int k=0; k<m; k++)
        for (int i=k+1; i<n; i++)
            b[i] = b[i]-A[i][k]*b[k];
    for (int i=n-1; i>=0; i--) {
        x[i] = b[i];
```

```

        for (int j=i+1; j<n; j++)
            x[i] = x[i] - A[i][j]*x[j];
        x[i] = x[i]/A[i][i];
    }
    return x;
}

```

Første doble for-løkke gjør radoperasjoner på høyresiden. Den er lettest å forstå dersom man (midlertidig) flytter setningen til slutten av andre for-løkke i LU-faktoriseringsalgoritmen. Da ser man at b-vektoren behandles som en femte kolonne, som kjent fra manuell Gausseliminering.

```

b[i] = b[i] - A[i][k]*b[k];

```

### 3.6.5 Eksempel

En 4x4 matrise med dens inverse, eksempel til bruk for å teste implementering. Matrisen som nedenfor kalles  $A^{-1}$  har  $a_{11} = 0$  og vil ikke fungere bra til testing uten pivotering.

$$A = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

### 3.6.6 Invers matrise

Gitt en matrise  $\mathbf{A} = [a_{ij}]$  og anta at dens inverse er  $\mathbf{B} = [b_{ij}]$ . Da gjelder

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vi benevner kolonnevektor  $j$  i  $\mathbf{B}$  ved  $\mathbf{b}_j$  og enhetsvektorene  $\mathbf{e}_j$  hvor  $j$ -te element er 1 og de andre er 0. Vi kan bruke LU-faktorisering og løse

$$\mathbf{A}\mathbf{b}_j = \mathbf{e}_j \Leftrightarrow \mathbf{b} = \mathbf{A}^{-1}\mathbf{e}_j, \quad j = 1, \dots, 4$$

for å finne den inverse matrisen. Hver gang vi bruker solve(), finner vi en kolonne i den inverse matrisen. Vi kan da bruke de fire løsningsvektorene til å danne den inverse matrisen.

### 3.6.7 Pivotering

Vi husker at for et lineært ligningssystem kan vi bytte om på rekkefølgen av ligningene uten at dette endrer ligningssystemet, og pivotering betyr nettopp å bytte på rekkefølgen av ligningene - eller radene i A-matrisen. Pivotering er nødvendig dersom det fins diagonalelementer som er null eller nær null. I implementeringen av pivotering her brukes en array *permutasjon* i matriseklassen til å lagre permutasjoner til bruk på b-vektor.

```

// Pivotering, bruker en Vec4 til å lagre permutasjoner
// til retur og bruk på b-vektor
void Matrix4x4::pivot(int k)
{
    if (-EPS <= A[k][k] && A[k][k] < EPS) // zero diagonalelement
    {
        // pivotering/swap - finner rad med største element i kolonne
        if (debug) std::cout << "må pivotere.\n";

        int pivot = k+1; // initierer kandidat for rad-pivotering
        for (int i=k+2; i<m; i++)
        {
            if (A[i][k] > A[pivot][k])
                pivot = i;
        }
        if (-EPS <= A[k][k] && A[k][k] < EPS)
            // std::cout << "Kan ikke invertere\n";
            ; // exception

        // swapper rader
        for (int j=k; j<n; j++)
            std::swap(A[k][j], A[pivot][j]);
        std::swap(permutasjon[k], permutasjon[pivot]);
    }
}

```

### 3.6.8 solve() med pivotering

Med pivotering blir funksjonen for å beregne  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  som nedenfor.

```

Vec4 Matrix4x4::solve(Vec4 &b) const
{
    Vec4 x;
    Vec4 b_perm(b[permutasjon[0]], b[permutasjon[1]], b[permutasjon[2]], b[permutasjon[3]])

    for (int k=0; k<n-1; k++) // m==n
        for (int i=k+1; i<n; i++)
            b_perm[i] = b_perm[i] - A[i][k]*b_perm[k];

    for (int i=n-1; i>=0; i--)
    {
        x[i] = b_perm[i];
        for (int j=i+1; j<n; j++)
            x[i] = x[i] - A[i][j]*x[j];
        x[i] = x[i]/A[i][i];
    }
    return x;
}

```

## 3.7 Oppgaver

### 3.7.1

1. Gitt en vektorbasis **A** med to basisvektorer  $[3 \ 1]^T$  og  $[-1 \ 2]^T$  for  $\mathbb{R}^2$ . Regn ut koordinatene i standardbasen **E** for vektorene  $[1, 2]$  og  $[-1, 1]$ .
2. Regn ut koordinatene i basisen **A** for vektoren som har koordinatene  $[1, 2]$  i standardbasis **E**.

### 3.7.2

1. Gitt en vektorbasis **A** med to basisvektorer  $[3 \ 1]^T$  og  $[1 \ 2]^T$  for  $\mathbb{R}^2$ . Regn ut koordinatene i standardbasen **E** for vektorene  $[1, 1]$  og  $[2, -1]$ .
2. Regn ut koordinatene i basisen **A** for vektorene som har koordinater  $[1, 2]$  og  $[1, 1]$  i standardbasis **E**.

### 3.7.3

La nå  $[3 \ 1]^T$  og  $[1 \ 2]^T$  være basisvektorer for et koordinatsystem (frame) **K'** med origo i  $O' = (\frac{3}{2}, 2)$ . To punkter P og Q er gitt ved  $P = (\frac{7}{2}, \frac{7}{2})$  og  $Q = (\frac{5}{2}, \frac{3}{2})$  hvor koordinatene er gitt i standard koordinatsystem K med basisvektorer  $[1, 0]$  og  $[0, 1]$ .

Regn ut koordinatene til P og Q i koordinatsystemet **K'**.

### 3.7.4

Gitt tre plan med følgende ligninger:

1.  $x+y+z=1$
2.  $5x+3y+2z=2$
3.  $x+3y+2z=1$

Sett opp ligningene på matriseform og løs ligningssystemet ved å bruke et C++ program med funksjon for å regne ut invers matrise.

### 3.7.5

Løs ligningssystemet **Ax=b** nedenfor ved å bruke et C++ program med funksjon for å regne ut invers matrise:

$$\begin{bmatrix} \frac{1}{16} & -\frac{1}{2} & \frac{3}{2} & -1 \\ \frac{1}{2} & -2 & \frac{3}{2} & -1 \\ 4 & -8 & 6 & -1 \\ \frac{27}{2} & -18 & 9 & -1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} \frac{1}{16} \\ \frac{1}{2} \\ 1 \\ \frac{7}{2} \end{bmatrix}$$

## Kapittel 4

# Konstruksjon av funksjoner

### 4.1 Introduksjon

I dette kapitlet ser vi på hvordan vi konstruerer funksjoner av en variabel som passer til gitte datasett  $\{(x_i, y_i)\}$ . Vi skal bruke polynomer til dette, og lære hvordan graden til et polynom spiller en vesentlig rolle. Vi starter med interpolasjon av to, tre og fire punkter og skal deretter studere minste kvadraters metode som er et eksempel på approksimasjon av data.

Interpolasjon betyr å beregne en verdi mellom kjente verdier. Det er også vanlig å si at en funksjon  $f$  interpolerer et punkt  $(x_0, y_0)$  når funksjonen går gjennom punktet slik at  $(f(x_0) = y_0)$ . Figur 4.1 viser et kubisk polynom som interpolerer fire punkter.

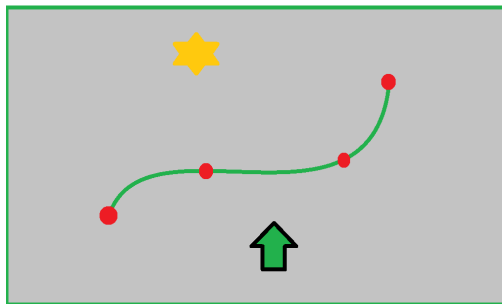
Å approksimere betyr å tilnærme, og en approksimasjon er en tilnærming (til en verdi). En approksimasjon er en ikke helt eksakt representasjon av en verdi og innebærer en avrundingsfeil. Minste kvadraters metode er et eksempel på approksimasjon.

### 4.2 Interpolasjon

Vi har inntil nå sett på flere eksempler hvor vi bruker en kjent funksjon eller kurve til å produsere kurver og flater i 2D og 3D. En trekant er en av de enkleste basisfigurene (grafiske primitiver). Når vi skal tegne en linje eller en trekant med for eksempel OpenGL eller WebGL teknologi, kan vi sende koordinater og rgb-farge for hvert hjørne til et shaderprogram (mer presist: til en vertex shader). Linjen eller trekanten som vises på skjermen har selvsagt mange flere piksler enn de to endepunktene / de tre hjørnene. I hvert piksel kan fargen bestemmes som en kombinasjon av fargen i hjørnene. Når en enkel fragmentshader lager farger på denne måten, interpolerer den fargene i hjørnene. Fargen mellom hjørnene beregnes ut fra fargen i hjørnene.

Vi kan, så langt:

- Analysere en kjent funksjon  $f(x)$ , regne ut punkter på grafen til en kjent funksjon  $f(x)$  og tegne grafen.



Figur 4.1: Bane til en npc som interpolerer fire punkter kan representeres med en 3. gradsfunksjon.

- Gjøre enkel flervariabel analyse av en kjent funksjon  $f(x, y)$ , regne ut punkter på flaten til  $f$  og tegne flaten.
- Regne ut punkter for en kjent vektorfunksjon  $\mathbf{f}(t) = [x(t), y(t), z(t)]$  og tegne kurven som funksjonen definerer.

Polynomer er en klasse funksjoner som er spesielt egnet og mye brukt til konstruksjon av kurver og flater, interpolasjon og approksimasjon. Polynomer er blant annet enkle å derivere og de har gode kontinuitetsegenskaper. Vi skal først starte med noen eksempler på interpolasjon av punkter med polynomer. Senere skal vi løse problemer med parametriske kurver.

Vi skal nå lære hvordan vi kan velge punkter og selv konstruere funksjoner, kurver og flater som passer til punktene. Vi har altså startet med dette, ved minste kvadraters metode. Vi ser på figur 4.1 at grafen til en tredjegradsfunksjon som interpolerer fire punkter kan bli glatt og jevn, og bruker det som en motivasjonsfaktor.



### 4.2.1 Interpolasjon av to, tre og fire punkter

Vi starter med et trivielt eksempel, å konstruere et polynom som interpolerer to punkter. Deretter skal vi konstruere polynomer som interpolerer tre og fire punkter, for eksempel punktene nedenfor. Våre krav er altså at funksjonens graf skal gå gjennom punktene. Hvis vi har to punkter, kan vi finne et førstegrads-polynom (lineært) som interpolerer punktene. Hvis vi har tre punkter, kan vi finne et andregradspolynom (kvadratisk) som går gjennom punktene. Og hvis vi har fire punkter, kan vi bestemme et tredjegradspolynom (kubisk) som går gjennom punktene. Noen eksempler:

1. (1,1) og (3,2)
2. (1,1), (3,3) og (5,1)
3. (0,0), (1,1), (2,0) og (3,1)

#### Interpolasjon av to punkter

Vi har gitt to punkter og konstruerer et lineært polynom som interpolerer punktene. Her bruker vi punktene ovenfor.

$$\begin{aligned}f(x) &= ax + b \\1 &= a \cdot 1 + b \\2 &= a \cdot 3 + b\end{aligned}$$

Vi kan sette opp dette på matriseform  $Ax=b$  (her er  $b$  en kolonnevektor), med

$$A = \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix}$$

$x^T = [a \ b]$  og  $b^T = [1 \ 3]$ . Løsningen blir  $x = A^{-1}b = [\frac{1}{2} \ \frac{1}{2}]$ , altså  $f(x) = \frac{1}{2}x + \frac{1}{2}$ .

#### Interpolasjon av tre punkter med andregradspolynom

Et andregradspolynom kan som kjent skrives på formen

$$f(x) = ax^2 + bx + c$$

Vi husker at vi kan sette  $f(x) = 0$  og finne nullpunktene/røttene av formelen

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Men nå er problemstillingen snudd. Vi har ikke lenger et kjent funksjonsuttrykk å jobbe med. Derimot har vi punkter (her: tre) og vi skal konstruere et polynom.

Vi vet jo fra før at det er mulig å konstruere et førstegrads-polynom (rett linje) som går gjennom to punkter. Tilsvarende kan vi konstruere et andregradspolynom som går gjennom tre punkter, og et tredjegradspolynom som går gjennom fire punkter. Og slik kan vi fortsette. Imidlertid kommer vi sjelden til å arbeide med polynomer av grad høyere enn 3. Hvis vi trenger å interpolere flere punkter

enn fire, skal vi heller skjøte sammen flere polynomer av grad 3 eller lavere.

Punktene uten å spesifisere koordinater er  $(x_0, y_0)$ ,  $(x_1, y_1)$  og  $(x_2, y_2)$ , og som alltid er det et valg man må gjøre om nummereringen skal starte på 0 eller 1.

Så hva slags problem får vi å løse? Bruker vi den kjente notasjonen vår for et andregradspolynom, får vi

$$\begin{aligned} ax_0^2 + bx_0 + c &= y_0 \\ ax_1^2 + bx_1 + c &= y_1 \\ ax_2^2 + bx_2 + c &= y_2 \end{aligned}$$

På matriseform blir dette  $\mathbf{Ax} = \mathbf{b}$  som har løsningen  $\mathbf{x} = A^{-1}\mathbf{b}$ .

Vi velger tre punkter som ikke ligger på en rett linje, (1,1), (3,3) og (5,1) og skal konstruere et andregradspolynom som interpolerer punktene.

$$\begin{aligned} f(x) &= ax^2 + bx + c \\ 1 &= a \cdot 1^2 + b \cdot 1 + c \\ 3 &= a \cdot 3^2 + b \cdot 3 + c \\ 1 &= a \cdot 5^2 + b \cdot 5 + c \end{aligned}$$

Vi kan sette opp dette på matriseform  $\mathbf{Ax}=\mathbf{b}$ , med

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 9 & 3 & 1 \\ 25 & 5 & 1 \end{bmatrix}$$

$x^T = [a \ b \ c]$  og  $b^T = [1 \ 3 \ 1]$ . Løsningen blir  $x = A^{-1}b = [-\frac{1}{2} \ 3 \ -\frac{3}{2}]$ , altså  $f(x) = -\frac{1}{2}x^2 + 3x - \frac{3}{2}$ .

### Eksempel (uten matriseformen)

Gitt punktene  $(0, \frac{3}{2})$ ,  $(1, 0)$  og  $(2, -\frac{1}{2})$ . Bestem andregradspolynomet som interpolerer punktene.

Vi får

$$\begin{aligned} a \cdot 0 + b \cdot 0 + c &= \frac{3}{2} \\ a \cdot 1 + b \cdot 1 + c &= 0 \\ a \cdot 4 + b \cdot 2 + c &= -\frac{1}{2} \end{aligned}$$

Ligning 1 gir  $c = \frac{3}{2}$ . Innsatt i ligning 2 får vi  $a + b = -\frac{3}{2} \Leftrightarrow b = -\frac{3}{2} - a$ . Setter vi dette inn i ligning 3, får vi  $4a + 2(-\frac{3}{2} - a) + \frac{3}{2} = -\frac{1}{2} \Leftrightarrow 2a - \frac{3}{2} = -\frac{1}{2} \Leftrightarrow a = \frac{1}{2}$ . Dette gir  $b = -2$  og  $p(x) = \frac{1}{2}x^2 - 2x + \frac{3}{2}$ .

### Interpolasjon av fire punkter med 3. gradspolynom

Når vi skal bestemme et polynom som interpolerer 4 punkter, trenger vi et polynom av grad 3. Vi skal se to måter å bruke tredjegrads polynom til interpolasjon,

det andre kalles kubisk Hermite interpolasjon.

Gitt fire punkter  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  og  $(x_3, y_3)$ . Vi skal bestemme et kubisk polynom (3. grads)  $p(x)$  som interpolerer punktene. Det vil si at

$$p(x_i) = y_i, \quad i = 0, 1, 2, 3.$$

Vi kan skrive det kubiske polynomet på formen

$$p(x) = ax^3 + bx^2 + cx + d \text{ hvor } a, b, c, d \in \mathbb{R},$$

altså reelle konstanter som i det kvadratiske tilfellet.

Vi skriver opp ligningene vi får av interpolasjonsbetingelsene:

$$\begin{aligned} ax_0^3 + bx_0^2 + cx_0 + d &= y_0 \\ ax_1^3 + bx_1^2 + cx_1 + d &= y_1 \\ ax_2^3 + bx_2^2 + cx_2 + d &= y_2 \\ ax_3^3 + bx_3^2 + cx_3 + d &= y_3 \end{aligned}$$

Vi velger nå punktene  $(0,0)$ ,  $(1,1)$ ,  $(2,0)$  og  $(3,1)$  og skal konstruere et tredjegradspolynom som interpolerer punktene.

$$\begin{aligned} f(x) &= ax^3 + bx^2 + cx + d \\ 0 &= a \cdot 0^3 + b \cdot 0^2 + c \cdot 0 + d \\ 1 &= a \cdot 1^3 + b \cdot 1^2 + c \cdot 1 + d \\ 0 &= a \cdot 2^3 + b \cdot 2^2 + c \cdot 2 + d \\ 1 &= a \cdot 3^3 + b \cdot 3^2 + c \cdot 3 + d \end{aligned}$$

Vi kan sette opp dette på matriseform  $\mathbf{Ax}=\mathbf{b}$ , med

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \end{bmatrix}$$

$x^T = [a \ b \ c \ d]$  og  $b^T = [0 \ 1 \ 0 \ 1]$ . Løsningen blir  $x = A^{-1}b = [\frac{2}{3} \ -3 \ \frac{10}{3} \ 0]$ , altså  $p(x) = \frac{2}{3}x^3 - 3x^2 + \frac{10}{3}x$ .

#### 4.2.2 Polynomrommet $\mathcal{P}^d$

Vi har ovenfor sett flere eksempler på at interpolasjonsbetingelsene vi får når vi skal konstruere polynomer, leder til lineære ligningssystemer. Hvis vi har  $d+1$  interpolasjonspunkter, kan vi interpolere med et polynom av grad  $\leq d$ . Vi setter vi opp  $d+1$  ligninger med  $d+1$  ukjente og bestemmer koeffisientene til polynomet. (Hvis punktene er kolleare (ligger på en rett linje), blir interpolasjonspolynomet en rett linje).

Det første vi merker oss, er at polynomer er nært beslektet med vektorer. Faktisk inneholder definisjonen av vektorrom (3.2) ikke noe krav om at en vektor

skal være et linjestykke med pil på. Derfor kan vi betrakte et polynom som en vektor, og vi kan tenke oss rom av polynomer på samme måte som vektorrom. Betegnelsen  $\mathcal{P}^d$  brukes om et lineært rom av polynomer av grad  $\leq d$  (et vektorrom kalles også et lineært rom)

Det neste vi merker oss, er at alle polynomer av grad  $\leq d$  kan skrives på formen

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_dx^d = \sum_{i=0}^d a_i x^i, \quad a_i \in \mathbb{R} \quad (4.1)$$

Vi ser av dette at polynomet  $p$  er en lineærkombinasjon av potenser av  $x$  (sammenligne med lineærkombinasjon av vektorer i avsnitt 3.2 hvor  $a_i$ -ene er konstanter (reelle tall)). Denne skrivemåten kalles potensformen eller monomialformen til  $p$  og funksjonene  $p_i(x) = x^i$  kalles monomialer. Det fins andre måter å skrive opp et polynom på som er mer effektive med tanke på evaluering, se for eksempel avsnitt 2.3.2.

Merk notasjonen av konstantene  $a_0 + a_1x + a_2x^2 + a_3x^3$  sammenlignet med tidligere notasjon  $ax^3 + bx^2 + cx + d$ . Se også avsnitt 4.2.2 nedenfor.

Hvis vi skriver opp interpolasjonsbetingelsene  $p(x_i) = y_i$  med notasjon som i likning (4.1), får vi  $d+1$  ligninger med  $d+1$  ukjente:

$$\begin{aligned} a_0 + a_1x_0 + a_2x_0^2 + a_3x_0^3 + \dots + a_dx_0^d &= y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + \dots + a_dx_1^d &= y_1 \\ a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3 + \dots + a_dx_2^d &= y_2 \\ &\vdots \\ a_0 + a_1x_d + a_2x_d^2 + a_3x_d^3 + \dots + a_dx_d^d &= y_d \end{aligned} \quad (4.2)$$

På matrisform  $\mathbf{Ax}=\mathbf{b}$  blir dette

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^d \\ 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^d \\ \vdots & & & & & \vdots \\ 1 & x_d & x_d^2 & x_d^3 & \dots & x_d^d \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix} \quad (4.3)$$

Dette er bare en generalisering av ligningssystemene i eksemplene ovenfor.

**Teorem 4.1 (Monomialbasisen)** *Monomialene  $\{1, x, x^2, \dots, x^d\}$  er en basis for polynomrommet  $\mathbf{P}^d$ .*

Det betyr at ethvert polynom av grad  $\leq d$  kan uttrykkes som en lineærkombinasjon av monomialene  $\{1, x, x^2, \dots, x^d\}$ . Vi bruker betegnelsen  $\mathbf{P}^d$  som knytter rommet til graden på polynomer. Men dimensjonen til rommet (betraktet som vektorrom) er  $\dim(\mathbf{P}^d) = d + 1$ .

Det er viktig at vi faktisk har et matematisk bevis for eksistensen til et interpolasjonspolynom (av ulik grad), og det er viktig at vi har et bevis for at det er entydig.

**Teorem 4.2 (Eksistens av interpolasjonspolynom)** *Gitt  $d+1$  distinkte punkter  $(x_i, y_i)$ . Da eksisterer et polynom  $p \in \mathbb{P}^d$  (av grad  $\leq d$ , på formen 4.1) slik at*

$$p(x_i) = y_i, \quad i = 0, \dots, d. \quad (4.4)$$

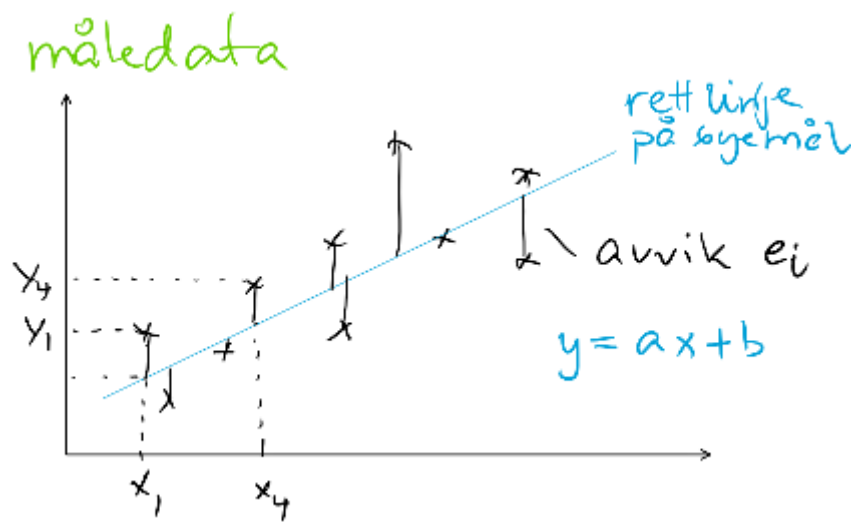
**Bevis:** Dette følger av at ligningssystemet på formen avsnitt 4.2.2 har en løsning, det vil si at  $\mathbf{A}^{-1}$  i likning (4.3) eksisterer. ■

**Teorem 4.3 (Entydighet for interpolasjonspolynomet)** *Polynomet i 4.4 er unikt.*

**Bevis:** Anta at det fins to forskjellige polynomer  $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_dx^d$  slik at  $p(x_i) = y_i$ ,  $i = 0, \dots, d$  og  $q(x) = b_0 + b_1x + b_2x^2 + \dots + b_dx^d$  slik at  $q(x_i) = y_i$ ,  $i = 0, \dots, d$ . Hvis  $p(x) = q(x)$  så må  $p(x) - q(x) = 0$  hvilket igjen impliserer  $(a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)x^2 + \dots + (a_d - b_d)x^d = 0$ . Men da må  $a_0 = b_0$ ,  $a_1 = b_1$ ,  $a_2 = b_2$ ,  $\dots$ ,  $a_d = b_d$  og vi har en selvmotsigelse. ■

### 4.3 Minste kvadraters metode

Minste kvadraters metode kan brukes til å beregne en funksjon som passer til måledata. Vi skal først se på et eksempel hvor vi prøver å finne en rett linje som passer best mulig til måledata. Dersom vi kun har to punkter, kan vi bruke



Figur 4.2: Minste kvadraters metode.

disse punktene til å bestemme ligningen for den rette linjen gjennom punktene. Vi har  $y_1 = ax_1 + b$  og  $y_2 = ax_2 + b$ , og vi kan finne konstantene  $a$  og  $b$ . Dette blir som i avsnitt 4.2.1. Så vi antar at vi har flere enn to punkter. Generelt for  $m$  punkter er  $y_i = ax_i + b$ ,  $i = 1, 2, \dots, m$ . Hvis ikke alle punktene ligger eksakt på linjen, kan vi skrive

$$y_i = ax_i + b + e_i, \quad i = 1, 2, \dots, m \quad (4.5)$$

Med  $m$  punkter får vi da følgende ligningssystem:

$$\begin{aligned} y_1 &= ax_1 + b + e_1 \\ y_2 &= ax_2 + b + e_2 \\ &\vdots \\ y_m &= ax_m + b + e_m \end{aligned} \quad (4.6)$$

hvor  $e_i$  er et lite feilledd som er positiv hvis  $y_i$  ligger over den rette linjen og negativ hvis  $y_i$  ligger under den rette linjen. Minste kvadraters metode går ut på å minimere summen av alle avvikene kvadrert:

$$\min \sum_{i=1}^m e_i^2 = \min \mathbf{e}^T \mathbf{e}$$

hvor  $\mathbf{e}^T = [e_1 \ e_2 \ \dots \ e_m]$ . Ligning 4.6 blir på matriseform

$$\mathbf{y} = \mathbf{Ax} + \mathbf{e} \quad (4.7)$$

eller

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & 1 \\ x_m & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix}$$

Fra 4.6 følger

$$\begin{aligned} \mathbf{e} &= \mathbf{Ax} - \mathbf{y} \\ \mathbf{e}^T \mathbf{e} &= (\mathbf{y} - \mathbf{Ax})^T (\mathbf{y} - \mathbf{Ax}) \\ &= (\mathbf{Ax} - \mathbf{y})^T (\mathbf{Ax} - \mathbf{y}) \\ &= (\mathbf{Ax})^T (\mathbf{Ax} - \mathbf{y}) - \mathbf{y}^T (\mathbf{Ax} - \mathbf{y}) \\ &= \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - \mathbf{x}^T \mathbf{A}^T \mathbf{y} - \mathbf{y}^T \mathbf{Ax} + \mathbf{y}^T \mathbf{y} \\ &= \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - 2\mathbf{y}^T \mathbf{Ax} + \mathbf{y}^T \mathbf{y} \\ &= \mathbf{x}^T \mathbf{Bx} - 2\mathbf{c}^T \mathbf{x} + \mathbf{y}^T \mathbf{y} \end{aligned} \quad (4.8)$$

hvor vi har brukt regnereglene for matriser:  $(AB)^T = B^T A^T$ ,  $\mathbf{B} = \mathbf{A}^T \mathbf{A}$  og  $\mathbf{c} = \mathbf{A}^T \mathbf{y}$ .

Minste kvadraters metode er et kvadratisk minimeringsproblem. Vi ser av siste linje ovenfor at

$$\mathbf{e}^T \mathbf{e} = \mathbf{x}^T \mathbf{Bx} - 2\mathbf{c}^T \mathbf{x} + \mathbf{y}^T \mathbf{y}$$

$\mathbf{y}^T \mathbf{y}$  er alltid er større eller lik null, og det kan vises at løsningen er

$$\mathbf{Bx} = \mathbf{c} \Leftrightarrow \mathbf{x} = \mathbf{B}^{-1} \mathbf{c}$$

### 4.3.1 Eksempel

Gitt punktene (1,2), (2,1) og  $(3, \frac{5}{2})$ . Vi får  $\mathbf{x}^T = [a \ b]$  og

$$\mathbf{y} = \begin{bmatrix} 2 \\ 1 \\ \frac{5}{2} \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \quad \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix} \quad \mathbf{A}^T \mathbf{y} = \begin{bmatrix} \frac{23}{2} \\ \frac{11}{2} \end{bmatrix}$$

Løsning: For en 2x2 matrise  $\mathbf{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$  er  $\mathbf{B}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ , slik

at her er  $\mathbf{B}^{-1} = \frac{1}{6} \begin{bmatrix} 3 & -6 \\ -6 & 14 \end{bmatrix}$  og vi får  $\mathbf{x} = \mathbf{B}^{-1} \mathbf{c} = [\frac{1}{4}, \frac{4}{3}]$ , altså  $a = \frac{1}{4}$  og  $b = \frac{4}{3}$  og ligningen for linjen blir  $y = \frac{1}{4}x + \frac{4}{3}$ .

### 4.3.2 Eksempel

Gitt punktene  $(1, \frac{1}{2})$ ,  $(\frac{3}{2}, \frac{3}{2})$ , (2,1) og  $(3, \frac{5}{2})$ . Vi skal bruke minste kvadraters metode til å bestemme ligningen for den rette linjen som passer best mulig til punktene.

$$\mathbf{y} = \begin{bmatrix} \frac{1}{2} \\ \frac{3}{2} \\ 1 \\ \frac{5}{2} \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} 1 & \frac{3}{2} & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 1 \\ \frac{3}{2} & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \quad \mathbf{A}^T \mathbf{A} = \begin{bmatrix} \frac{65}{2} & \frac{15}{2} \\ \frac{15}{2} & 4 \end{bmatrix} \quad \mathbf{A}^T \mathbf{y} = \begin{bmatrix} \frac{49}{4} \\ \frac{11}{2} \end{bmatrix}$$

Videre,  $\mathbf{B}^{-1} = \frac{4}{35} \begin{bmatrix} 4 & -\frac{15}{2} \\ -\frac{15}{2} & \frac{65}{4} \end{bmatrix}$  og  $\mathbf{x} = B^{-1}\mathbf{c} = \frac{4}{35} \begin{bmatrix} 49 - \frac{165}{8} \\ -\frac{735}{8} + \frac{715}{8} \end{bmatrix} = \begin{bmatrix} \frac{31}{35} \\ -\frac{2}{7} \end{bmatrix}$   
 Altså er  $a = \frac{31}{35}$  og  $b = -\frac{2}{7}$ .

Vi kontrollerer at dette stemmer med punktene, og konkluderer med at vi trenger et verktøy slik at vi slipper å gjøre så mye brøkgregning for hånd.

### 4.3.3 Minste kvadraters metode og andregradsfunksjon

Ligningssystemet 4.6 blir på formen

$$\begin{aligned} y_1 &= ax_1^2 + bx_1 + c + e_1 \\ y_2 &= ax_2^2 + bx_2 + c + e_2 \\ &\vdots \\ y_m &= ax_m^2 + bx_m + c + e_m \end{aligned} \quad (4.9)$$

Vi ser at vi da får en ekstra kolonne i  $\mathbf{A}$  matrisen og at  $\mathbf{x}^T = [a \ b \ c]$ . Antall ligninger er det samme som antall målepunkter.  $\mathbf{B} = \mathbf{A}^T \mathbf{A}$  blir en  $3 \times 3$  matrise.

### 4.3.4 Eksempel

Gitt  $m=5$  punkter  $(-2.0, 4.1)$ ,  $(-1.0, 0.9)$ ,  $(0.0, 0.1)$ ,  $(1.0, 1.1)$  og  $(2.0, 3.9)$ . Vi skal bruke minste kvadraters metode til å bestemme en andregradsfunksjon som passer best mulig til punktene. (Vi trenger jo bare tre punkter for å bestemme en andregradsfunksjon, så vi har to punkter ekstra).

$$\mathbf{A} = \begin{bmatrix} 4.00 & -2.00 & 1.00 \\ 1.00 & -1.00 & 1.00 \\ 0.00 & 0.00 & 1.00 \\ 1.00 & 1.00 & 1.00 \\ 4.00 & 2.00 & 1.00 \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} 4.00 & 1.00 & 0.00 & 1.00 & 4.00 \\ -2.00 & -1.00 & 0.00 & 1.00 & 2.00 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix}$$

$$\mathbf{B} = \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 34 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 5 \end{bmatrix} \quad \mathbf{A}^T \mathbf{y} = \begin{bmatrix} 4.00 & 1.00 & 0.00 & 1.00 & 4.00 \\ -2.00 & -1.00 & 0.00 & 1.00 & 2.00 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix}$$

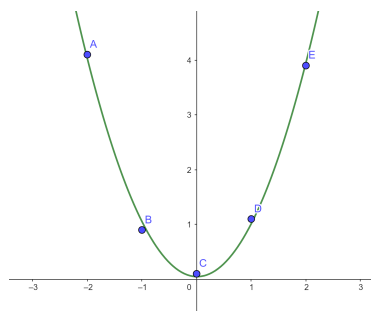
Løsning:

$$\mathbf{B}^{-1} = \begin{bmatrix} 0.0714 & 0 & -0.1429 \\ 0 & 0.1000 & 0 \\ -0.1429 & 0 & 0.4857 \end{bmatrix}$$

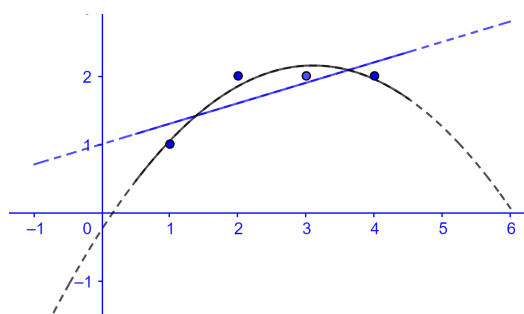
$$\mathbf{c} = \mathbf{A}^T \mathbf{y} = \begin{bmatrix} 34.0 \\ -0.2 \\ 10.1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 0.9858 \\ -0.0190 \\ 0.0484 \end{bmatrix}$$





Figur 4.3: Minste kvadraters metode og andregradsfunksjon.



Figur 4.4: Hva passer best, en rett linje eller andregradsfunksjon? Figuren viser løsning på oppgave 4.6.7.

#### 4.3.5 Sammendrag

Minste kvadraters metode kan brukes til å bestemme en funksjon som beskriver et datasett best mulig, eller passer best mulig til dataene. Vi har sett eksempler på hvordan en lineær funksjon og en andregradsfunksjon kan brukes, og det bør ikke være vanskelig å utvide til høyere grad. Det er ikke slik at alle datapunktene ligger på grafen til funksjonen. Vi sier at funksjonen som vi konstruerer, approksimerer dataene. Når datapunktene ligger på grafen til funksjonen, eller på en parametrisk kurve, sier vi at funksjonen eller kurven interpolerer punktene.

## 4.4 Kubisk Hermite interpolasjon

I stedet for å konstruere et kubisk polynom som interpolerer fire punkter  $(x_i, y_i)$  kan vi interpolere to funksjonsverdier og to deriverte. Det kalles kubisk Hermite interpolasjon. Da må vi kjenne to punkter  $(x_0, y_0)$  og  $(x_1, y_1)$  og i tillegg stigningstallet til en tangent til funksjonen i disse punktene. Vi ser på et eksempel.

Gitt punktene  $P_0(-2, -\frac{15}{2})$  og  $P_1(3, 0)$ . I stedet for å bruke interpolasjonsbetingelsene i ytterligere to punkter for å sette opp et ligningssystem med fire ligninger og fire ukjente, kan vi stille betingelser til de deriverte i  $P_0$  og  $P_1$ . ( $P_1$  her er det samme punktet som  $P_3$  i oppgave 4.6.3).

La oss kreve at den deriverte til interpolasjonspolynomet  $p$  i  $x_0 = -2$  og  $x_1 = 3$  skal være  $p'(-2) = \frac{23}{2}$  og  $p'(3) = 4$ . Vi har da følgende fire interpolasjonsbetingelser:

1.  $p(-2) = -\frac{15}{2}$
2.  $p(3) = 0$
3.  $p'(-2) = \frac{23}{2}$
4.  $p'(3) = 4$

Polynomet vi skal bestemme er på formen  $p(x) = ax^3 + bx^2 + cx + d$ , og når vi deriverer får vi  $p'(x) = 3ax^2 + 2bx + c$ . Vi setter inn og regner ut og får

$$\begin{aligned} -8a + 4b - 2c + d &= -\frac{15}{2} \\ 27a + 9b + 3c + d &= 0 \\ 12a - 4b + c &= \frac{23}{2} \\ 27a + 6b + c &= 4 \end{aligned}$$

Vi kan nå sette opp dette på matriseform  $\mathbf{Ax}=\mathbf{b}$  og løse ligningssystemet på vanlig måte. Løsningen blir, skrevet på potensform  $p(x) = \frac{1}{2}x^3 - \frac{3}{2}x^2 - \frac{1}{2}x + \frac{3}{2}$ .

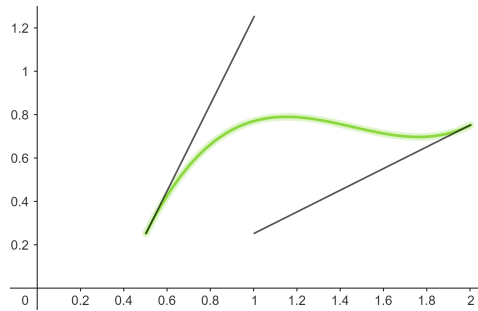
### 4.4.1 Eksempel

Vi kan sette opp en tredjegradsfunksjon på potensform som tidligere, med uttrykket for den deriverte:

$$\begin{aligned} p(x) &= ax^3 + bx^2 + cx + d \\ p'(x) &= 3ax^2 + 2bx + c \end{aligned} \tag{4.10}$$

La oss se på et eksempel hvor vi interpolerer en funksjon i punktene  $(x_0, y_0)$  og  $(x_1, y_1)$  hvor  $x_0 < x_1$ . Vi tegner figur og setter inn noen tall:

- Endepunkter til intervallet:  $x_0 = \frac{1}{2}$ ,  $y_0 = \frac{1}{4}$ ,  $x_1 = 2$ ,  $y_1 = \frac{3}{4}$
- De deriverte i endepunktene:  $y'_0 = 2$  og  $y'_1 = \frac{1}{2}$



Figur 4.5: Hermite interpolasjon, eksempel.

### Interpolasjonsbetingelser

Ved kubisk Hermite interpolasjon på intervallet  $[x_0, x_1]$  krever vi

$$\begin{aligned} p(x_0) &= y_0, \quad p(x_1) = y_1 \\ p'(x_0) &= y'_0, \quad p'(x_1) = y'_1 \end{aligned} \quad (4.11)$$

Interpolasjonsbetingelsene gir

$$\begin{aligned} p(x_0) = y_0 &\Rightarrow ax_0^3 + bx_0^2 + cx_0 + d = y_0 \\ p(x_1) = y_1 &\Rightarrow ax_1^3 + bx_1^2 + cx_1 + d = y_1 \end{aligned}$$

Når vi setter inn tall, får vi

$$\begin{aligned} \frac{1}{8}a + \frac{1}{4}b + \frac{1}{2}c + d &= \frac{1}{4} \\ 8a + 4b + 2c + d &= \frac{3}{4} \end{aligned}$$

Vi har så langt to ligninger med fire ukjente, og vi trenger to ligninger til for å bestemme konstantene a, b, c og d. De siste to ligningene får vi av de deriverte i endepunktene. Vi får av ligning 4.10  $p'(x) = 3ax^2 + 2bx + c$  og videre

$$\begin{aligned} p'(x_0) = y'_0 &\Rightarrow p'\left(\frac{1}{2}\right) = 2 \Rightarrow \frac{3}{4}a + b + c = 2 \\ p'(x_1) = y'_1 &\Rightarrow p'(2) = \frac{1}{2} \Rightarrow 12a + 4b + c = \frac{1}{2} \end{aligned}$$

På matriseform  $\mathbf{Ax}=\mathbf{b}$  blir dette

$$\begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \\ 8 & 4 & 2 & 1 \\ \frac{3}{4} & 1 & 1 & 0 \\ 12 & 4 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \\ 2 \\ \frac{1}{2} \end{bmatrix}$$

I matrisen ovenfor utgjør interpolasjonsbetingelsene de to første radene, og derivertbetingelsene de to siste radene.

Numerisk løsning av dette ligningssystemet gir  $p(x) = 0.815x^3 - 3.556x^2 + 4.944x - 1.435$ .

Vi kan også bestemme Hermite-interpolasjonspolynomet uten å bruke matriseformen:

**Lemma 4.4 (Kubisk Hermite interpolasjon)** *Gitt reelle tall  $x_0 < x_1$ ,  $y_0, y_1, y'_0, y'_1$  og la  $h = x_1 - x_0$ . Polynomet  $p$  gitt ved*

$$p(x) = y_0 + y'_0(x - x_0) + q(x - x_0)^2 + c(x - x_0)^3 \quad (4.12)$$

hvor

$$q = \frac{3 \frac{y_1 - y_0}{h} - 2y'_0 - y'_1}{h} \quad (4.13)$$

$$c = \frac{-2 \frac{y_1 - y_0}{h} + y'_0 + y'_1}{h^2} \quad (4.14)$$

er det entydige kubisk polynomet slik at  $p(x_0) = y_0$ ,  $p(x_1) = y_1$ ,  $p'(x_0) = y'_0$  og  $p'(x_1) = y'_1$

Vi kommer ikke til å bevise denne setningen her, selv om det ikke er veldig vanskelig. Vi kommer heller ikke nærmere inn på hvordan disse formlene framkommer. Setningen er hentet fra [12] og er som nevnt tatt med for å kunne regne ut Hermite interpolasjonspolynomer på en litt enklere måte enn ved å løse 4x4 ligningssystem. Metoden gir litt raskere beregninger (bedre matriser) enn ved å bruke potensbasisen direkte.

### Eksempel

Vi skal finne kubisk Hermite interpolant til  $x^4$  på  $[0,1]$  ved å bruke resultatet ovenfor. Vi får

$$x_0 = 0, x_1 = 1, h = 1, y_0 = 0, y_1 = 1, y'_0 = 0, y'_1 = 4$$

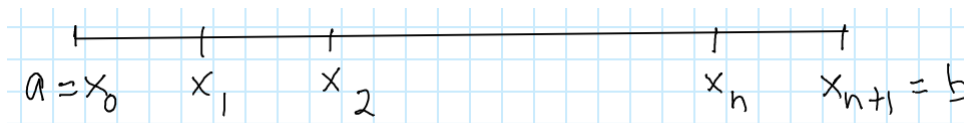
Videre  $q = -1$  og  $c = 2$  (sett inn i formlene og regn ut dette) og til slutt  $p(x) = -x^2 + 2x^3$ .

Kontroll:  $p(0) = 0$ ,  $p(1) = 1$ ,  $p'(x) = -2x + 6x^2 \Rightarrow p'(0) = 0$ ,  $p'(1) = 4$ . Alle de fire interpolasjonsbetingelsene stemmer, og da må resultatet stemme. Tegn figur!

### Eksempel

Gitt fire punkter  $(0,1)$ ,  $(\frac{1}{2}, 2)$ ,  $(\frac{3}{2}, 1)$ ,  $(2, 2)$ . Hvordan kan vi gjøre kubisk Hermite interpolasjon med disse punktene?

La første og siste punkt være endepunkter til en ukjent funksjon, slik at  $x_0 = 0, x_1 = 2, h = 2, y_0 = 1$  og  $y_1 = 2$  i lemma 4.4. Videre trenger vi de deriverte i endepunktene. Vi kan bruke vi Newtons differenskvotient ((2.2)) til å beregne  $y'_0 = \frac{2-1}{\frac{1}{2}-0} = 2$  og  $y'_1 = \frac{2-1}{2-\frac{3}{2}} = 2$ .



Figur 4.6: Stykkevis Hermite interpolasjon, intervaller. Punktene hvor lokale polynomer møtes kalles gjerne skjøter.

Vi får  $q = -\frac{9}{4}$ ,  $c = \frac{3}{4}$  og

$$p(x) = 1 + 2x - \frac{9}{4}x^2 + \frac{3}{4}x^3$$

Kontroll:  $p(0) = 1$ ,  $p(2) = 2$ ,  $p'(0) = 2$ ,  $p'(1) = 2$ .

#### 4.4.2 Stykkevis kubisk Hermite interpolasjon

Før vi går videre, repeterer vi ideen med å bruke kubiske polynomer til interpolasjon. Hvis vi skal interpolere  $n + 1$  punkter, kan vi gjøre det med et polynom av grad  $\leq n$ . Når  $n$  blir stor, leder dette til store ligningssystemer som er tunge å løse, og andre ulemper med polynomer av høy grad.

Vi skal nå bruke kubisk Hermite interpolasjon med flere polynombiter/intervaller, se 4.6. Vi har  $n+2$  punkter  $a = x_0 < x_1 < \dots < x_n < x_{n+1} = b$  og skal konstruere en approksimasjon  $F$  til en gitt funksjon  $f$  ved å sette sammen biter av kubiske Hermite interpolanter.

Anta at  $s_i = f'(x_i)$ ,  $i = 0, \dots, n+1$  eksisterer. På hvert intervall  $[x_i, x_{i+1}]$  har vi et kubisk Hermite polynom:

$$\begin{aligned} p_i(x_i) &= f(x_i), & p_i(x_{i+1}) &= f(x_{i+1}), \\ p'_i(x_i) &= s_i, & p'_i(x_{i+1}) &= s_{i+1} \end{aligned} \quad (4.15)$$

Den sammensatte funksjonen

$$F(x) = \begin{cases} p_0(x) & x_0 \leq x < x_1 \\ p_1(x) & x_1 \leq x < x_2 \\ \vdots & \\ p_{n-1}(x) & x_{n-1} \leq x < x_n \\ p_n(x) & x_n \leq x \leq x_{n+1} \end{cases} \quad (4.16)$$

er konstruert slik at

$$\begin{aligned} p_{i-1}(x_i) &= f(x_i) = p_i(x_i) \\ p'_{i-1}(x_i) &= f'(x_i) = p'_i(x_i) \quad 1 \leq i \leq n \end{aligned} \quad (4.17)$$

Siden  $F$  er et polynom på hvert intervall og deriverbar i skjøtene, er  $F$  deriverbar på  $[a, b]$ . Vi kan nå bruke setning 4.12 på hvert intervall, ved å re-indeksere:  $x_0 := x_i$ ,  $x_1 := x_{i+1}$ ,  $y_0 := f(x_i)$ ,  $y_1 := f(x_{i+1})$ ,  $y'_0 := s_i$ ,  $y'_1 := s_{i+1}$ . Dette gir for hvert polynom  $p_i$

$$p_i(x) = f(x_i) + s_i(x - x_i) + q_i(x - x_i)^2 + c_i(x - x_i)^3 \quad (4.18)$$

Her er

$$\begin{aligned} q_i &= \frac{3d_i - 2s_i - s_{i+1}}{h_i} \\ c_i &= \frac{-2d_i + s_i + s_{i+1}}{h_i^2} \\ d_i &= \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \\ h_i &= x_{i+1} - x_i \end{aligned} \quad (4.19)$$

Vi kan skjøte sammen flere kubiske polynomer for å interpolere mer enn fire punkter, og den resulterende funksjonen kalles da en splinefunksjon. Splines er stykkevis polynomer. Vi kan for eksempel ha stykkevis lineære polynomer, kvadratiske, kubiske. Vi kan ha polynomer i flere variable. En Hermite spline er et stykkevis interpolerende polynom hvor både en verdi og en derivert er gitt i kontrollpunktene/skjøtene. Hver polynombit avhenger kun av sine endepunktsbetingelser og kan justeres lokalt uten at hele det sammensatte polynomet må rekalkuleres.

Kubisk spline interpolasjon ligner på kubisk Hermite interpolasjon, men det er ikke noe krav om å interpolere den deriverte i skjøtene. I stedet krever man at venstre og høyre polynombit har samme derivert i skjøtene.

#### 4.4.3 Eksempel

La oss se på et eksempel på stykkevis kubisk Hermite interpolasjon med to intervaller. Vi skal finne den stykkevis kubiske Hermite interpolanten til  $f(x) = x^4$ . Vi velger  $a=0$  og  $b=2$ . Med to intervaller blir  $n = 1$  og  $\{x_0, x_1, x_2\} = \{0, 1, 2\}$ .

Vi finner først  $p_1$  ved å sette  $i=1$  i ligningene 4.18 og 4.19. Dette gir  $f(x_1) = 1$ ,  $f(x_2) = 16$ ,  $s_1 = 4$ ,  $s_2 = 32$ ,  $h_1 = 1$ ,  $d_1 = 15$ ,  $q = 5$ ,  $c = 6$ .

Innsatt i 4.19 får vi

$$p_1(x) = 1 + 4(x-1) + 5(x-1)^2 + 6(x-1)^3, \quad 1 \leq x \leq 2.$$

#### Øving

Vis på samme måte at  $p_0(x) = 2x^3 - x^2$ ,  $0 \leq x \leq 1$  (samme resultat som i eksempel 4.4.1).

#### Resultat

$$F = \begin{cases} p_0(x) = 2x^3 - x^2, & 0 \leq x \leq 1 \\ p_1(x) = 1 + 4(x-1) + 5(x-1)^2 + 6(x-1)^3, & 1 \leq x \leq 2 \end{cases}$$

#### 4.4.4 Kubisk spline interpolasjon

Vi bruker fortsatt intervallet  $a = x_0 < x_1 < \dots < x_{n+1} = b$ . Vi har fortsatt kjente funksjonsverdier  $f(x_i)$  og et stykkevis polynom som i likning (4.16). kubiske polynomer  $p_i(x)$ ,  $i = 0, \dots, n$ . De deriverte  $f'(x_i)$  i Hermite interpolasjon-ligningen avsnitt 4.4.2 er i praksis ofte ukjent. Vi fjerner kravet om interpolasjon av den deriverte i avsnitt 4.4.2. I stedet krever vi at det stykkevis polynomet

( $F(x)$  i likning (4.16) skal være to ganger kontinuerlig deriverbar på i  $I = [a, b]$ . Dette betyr

$$p''_{i-1}(x_i) = p''_i(x_i), \quad i = 1, \dots, n \quad (4.20)$$

Vi må finne de ukjente deriverte i hver skjøt. Dette kan løses ved å sette opp et  $n \times n$  tridiagonalt ligningssystem. Vi skal senere studere denne type problemer med B-splines (avsnitt 5.5).

## 4.5 Spline funksjoner

Vi har allerede definert splines som stykkevis polynomer og stiftet bekjentskap med Hermite splines. En spline funksjon  $f(x)$  er altså en funksjon som består av flere polynomer  $p_i(x)$ ,  $i = 0, 1, \dots, n-1$  slik at  $f(x) = p_i(x)$ ,  $x_i \leq x < x_{i+1}$ ,  $i = 0, 1, \dots, n$  hvor selvsagt  $n \geq 2$ . B-splines vil grundig gjennomgått i kapittel 5 og kapittel 7. Vi skal nå her på noen flere eksempler som forhåpentligvis illustrerer behovet for splines.

### 4.5.1 Eksempel

La oss først se på eksempel 4.4.1 på nytt. Vi har fire punkter  $(0, 1)$ ,  $(\frac{1}{2}, 2)$ ,  $(\frac{3}{2}, 1)$ ,  $(2, 2)$ , og Hermite interpolasjon gir  $p(x) = 1 + 2x - \frac{9}{4}x^2 + \frac{3}{4}x^3$ . Kan vi klare å konstruere et lignende stykkevis polynom av grad 2?

Kubisk hermite interpolasjon krever to funksjonsverdier og to deriverte. Her prøver vi å konstruere to kvadratiske polynomer  $p_0(x) = a_0x^2 + b_0x + c_0$  definert for  $0 \leq x < 1$  og  $p_1(x) = a_1x^2 + b_1x + c_1$  definert for  $1 \leq x \leq 2$ . Kravene fra eksempel 4.4.1 gir

1.  $p_0(0) = 1$
2.  $p'_0(0) = 2$
3.  $p_1(2) = 2$
4.  $p'_1(2) = 2$

Krav til kontinuitet gir

5.  $p_0(1) = p_1(1)$
6.  $p'_0(1) = p'_1(1)$

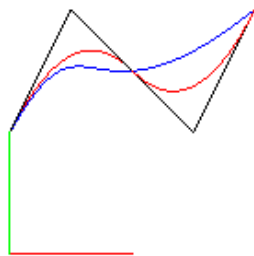
Vi har nå seks ligninger som kan brukes til å bestemme de seks ukjente  $a_0, b_0, c_0, a_1, b_1$  og  $c_1$ . Vi deriverer og får  $p'_0(x) = 2a_0x + b_0$  og  $p'_1(x) = 2a_1x + b_1$

1.  $c_0 = 1$
2.  $p'_0(0) = 2 \Leftrightarrow b_0 = 2$
3.  $p_1(2) = 2 \Leftrightarrow 4a_1 + 2b_1 + c_1 = 2$
4.  $p'_1(2) = 2 \Leftrightarrow 4a_1 + b_1 = 2$
5.  $p_0(1) = p_1(1) \Leftrightarrow a_0 + b_0 + c_0 = a_1 + b_1 + c_1$
6.  $p'_0(1) = p'_1(1) \Leftrightarrow 2a_0 + b_0 = 2a_1 + b_1$

(4) innsatt i (3) gir  $b_1 + c_1 = 0$ . (1), (2) og siste resultat benyttet i (5) gir  $a_0 + 3 = a_1$ . Setter vi alt dette inn i (6), får vi  $2a_0 + 2 = 2(a_0 + 3) + b_1$  som gir  $b_1 = -4$ ,  $c_1 = 4$ . (4) gir nå  $a_1 = \frac{3}{2}$ , til slutt  $a_0 = a_1 - 3 = -\frac{3}{2}$ . Vi har da

$$p_0(x) = -\frac{3}{2}x^2 + 2x + 1 \text{ og } p_1(x) = \frac{3}{2}x^2 - 4x + 4$$





Figur 4.7: Den svarte kurven viser kontrollpolygonet til punktene  $(0,1)$ ,  $(\frac{1}{2},2)$ ,  $(\frac{3}{2},1)$ ,  $(2,2)$  i eksempel 4.4.1. Den blå kurven viser kubisk Hermite interpolant til de samme punktene. Den røde kurven viser de to kvadratiske polynomene som utgjør splinefunksjonen i eksemplet ovenfor.

Funksjonen  $f(x) = p_i(x)$ ,  $x_i \leq x < x_{i+1}$ ,  $i = 0, 1, 2$  blir da et eksempel på kvadratisk spline interpolasjon.

#### 4.5.2 Eksempel

Funksjonen  $f(x) = x^2$  er den enkleste andregradsfunksjonen (eller det enkleste kvadratiske polynomet) vi kjenner. Vi kan enkelt skalere og translere dette. Polynomet  $p_0(x) = \frac{1}{2}x^2$  er  $f(x)$  skalert, og er enkelt å tegne opp, og det er enkelt å regne ut  $p_0(1) = \frac{1}{2}$ . Notasjonen  $p_0$  her er valgt fordi vi skal ha flere polynomer, og her ikke noe med graden (som er 2) å gjøre.

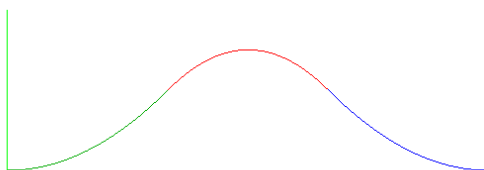
Det er ofte nyttig å skrive polynomer på en annen måte enn på standardform/potensform. Hvis vi skriver  $p_2(x) = \frac{1}{2}(x-3)^2$ , så er det ganske enkelt å se at et variabelskifte  $u = x-3$  gir et uttrykk  $p_2(u) = \frac{1}{2}u^2$ , det er altså  $p_0(x)$  flyttet tre enheter i positiv x-retning. Det er enkelt å regne ut  $p_2(2) = \frac{1}{2}$ . Videre ser vi at  $p_0(0) = p_2(3) = 0$ . Disse to polynomene har hver sin symmetrilinje, og de er innbyrdes symmetriske om  $x = \frac{3}{2}$ .

La nå  $p_0(x)$  være definert for  $0 \leq x \leq 1$  og la  $p_2(x)$  være definert for  $2 \leq x \leq 3$ .

Hvordan kan vi finne et kvadratisk polynom  $p_1(x)$ ,  $1 \leq x \leq 2$  som skjøter sammen disse to andre polynombitene? Vi skal ha et polynom på formen  $q(x) = ax^2 + bx + c$ , da er  $q'(x) = 2ax + b$ . Vi krever at den deriverte er kontinuerlig i skjøtene, noe som gir betingelsene  $p'_0(1) = 1 = q'(1)$  og  $q'(2) = p'_2(2) = -1$ . Av dette får vi to ligninger med to ukjente a og b, med løsning a = -1 og b = 3. Altså er  $q(x) = -x^2 + 3x + c$ . Vi kan bestemme c av betingelsen  $p_1(\frac{1}{2}) = q(\frac{1}{2})$  og får  $c = -\frac{3}{2}$ .

Vi har nå bestemt en polynombit som passer mellom  $p_0(x)$  og  $p_2(x)$ ,

$$p_1(x) = -x^2 + 3x - \frac{3}{2} = -(x - \frac{3}{2})^2 + \frac{3}{4}$$



Figur 4.8: En splinefunksjon som er en kvadratisk B-spline basisfunksjon.

Hvis vi nå tegner opp disse tre polynombitene i et egnet verktøy, ser vi at de passer finst sammen, de er kontinuerlige og faktisk deriverbare i skjøtene  $x = 1$  og  $x = 2$ , se figur 4.8.

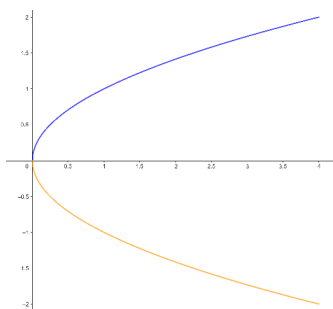
### Øving

Vis ved regning at  $p'_0(1) = p'_1(1)$  og  $p'_1(2) = p'_2(2)$ .

De tre polynombitene danner sammen en kvadratisk splinefunksjon, faktisk det som kaller en B-spline basisfunksjon, som vi kommer tilbake til i avsnitt 5.5.4 nedenfor. Her er også funksjonene tegnet opp og regnet ut ved rekursjonsformelen for B-splines.

### 4.5.3 Sammendrag

Vi har studert ulike metoder å konstruere funksjoner på ved å benytte polynomer, interpolasjon og approksimasjon. Vi har blant annet sett at vi kan velge å interpolere fire punkter med et kubisk polynom, eller at vi i stedet kan interpolere endepunktene og de derivate i endepunktene. I begge tilfeller får vi et  $4 \times 4$  ligningssystem som bestemmer konstantene til det kubiske polynomet. Imidlertid har funksjoner noen klare begrensninger. Men metodene som vi går gjennom her for funksjoner av en variabel vil i mange tilfeller kunne generaliseres til parametriske kurver.



Figur 4.9: Vi trenger parametriske kurver!

La oss se på eksemplet  $f(x) = x^2$ . Når vi roterer grafen til funksjonen 90

grader som på figur 4.9, er det plutselig nødvendig med to funksjoner  $f(x) = \sqrt{x}$  og  $g(x) = -\sqrt{x}$  for å beskrive hele kurven medfunksjonsuttrykk. Vi har tilsvarende situasjon for en sirkel. Ligningen for enhetssirkelen er  $x^2 + y^2 = 1$ , dette gir funksjonene  $f(x) = \sqrt{1-x^2}$  og  $g(x) = -\sqrt{1-x^2}$ . En kjent parametrisering er  $x(t) = \cos t$ ,  $y(t) = \sin t$ ,  $0 \leq t < 2\pi$ .

På samme måte kan vi parametrisere kurven på figur 4.9,  $x(t) = t^2$ ,  $y(t) = t$ . Dette er to eksempler på at det generelt er mer hensiktsmessig å benytte parametriske kurver enn funksjoner.

## 4.6 Oppgaver

I oppgavene nedenfor er det meningen at du skal tegne eller skissere polynomene du regner ut i tillegg til å sette opp og løse ligningssystemene. Du kan gjerne bruke et passende tegneprogram. Senere skal vi programmere tegning av slike kurver sjøl.

### 4.6.1

Bestem et polynom som interpolerer punktene  $(0, 1)$ ,  $(2, 0)$  og  $(3, 4)$ .

### 4.6.2

Interpolasjon med andregradspolynom.

1. La  $(x_0, y_0) = (\frac{1}{2}, \frac{5}{4})$ ,  $(x_1, y_1) = (2, -1)$  og  $(x_2, y_2) = (4, 3)$ . Du skal finne et andregradspolynom som interpolerer disse punktene. Sett opp ligningene på matrisform.
2. Bestem interpolasjonspolynomet (andregradsfunksjonen).
3. Tegn/skisser funksjonen.

### 4.6.3

Interpolasjon med tredjegradspolynom.

1. Sett opp matrisen  $\mathbf{A}$  og vektorene  $\mathbf{x}$  og  $\mathbf{b}$  for det generelle tilfellet.
2. La  $(x_0, y_0) = (-2, -\frac{15}{2})$ ,  $(x_1, y_1) = (\frac{1}{2}, \frac{15}{16})$ ,  $(x_2, y_2) = (2, -\frac{3}{2})$  og  $(x_3, y_3) = (3, 0)$ . Du skal finne et tredjegradspolynom som interpolerer disse punktene. Sett opp ligningene på matrisform.
3. Bestem interpolasjonspolynomet ved å bruke en implementering av 4x4 matrise.
4. Tegn/skisser funksjonen.

I oppgavene nedenfor er problemstillingen ikke like detaljert, men framgangsmåten blir den samme.

### 4.6.4

Interpolasjon med tredjegradspolynom: Bestem et polynom av grad  $\leq 3$  som interpolerer punktene  $(0, 1)$ ,  $(\frac{1}{2}, \frac{9}{16})$ ,  $(\frac{3}{2}, -\frac{5}{16})$  og  $(3, 4)$

### 4.6.5

Interpolasjon av halvsirkel.

1. Bestem et polynom av grad  $\leq 2$  som interpolerer punktene  $(-1, 0)$ ,  $(0, 1)$ ,  $(1, 0)$

2. Bestem et polynom av grad  $\leq 3$  som interpolerer punktene  $(-1, 0)$ ,  $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$ ,  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ ,  $(1, 0)$
3. Sammenligne svarene. Tegn opp grafene sammen med øvre halvdel av enhetssirkelen.

#### 4.6.6

Interpolasjon av  $\sin x$ .

1. Bestem et polynom av grad  $\leq 2$  som interpolerer punktene  $(0, 0)$ ,  $(\frac{\pi}{2}, 1)$ ,  $(\pi, 0)$
2. Bestem et polynom av grad  $\leq 2$  som interpolerer punktene  $(\pi, 0)$ ,  $(\frac{3\pi}{2}, -1)$ ,  $(2\pi, 0)$
3. Tegn begge polynomene sammen med funksjonen  $\sin x$ ,  $0 \leq x \leq 2\pi$

#### 4.6.7

Gitt punktene  $(1,1)$ ,  $(2,2)$ ,  $(3,2)$  og  $(4,2)$ . Bruk minste kvadraters metode til å bestemme ligningen for en rett linje som passer best mulig til punktene.

Gitt de samme punktene  $(1,1)$ ,  $(2,2)$ ,  $(3,2)$  og  $(4,2)$ . Du skal nå bruke minste kvadraters metode til å bestemme en andregradsfunksjon som passer best mulig til punktene.

1. Sett opp matrisen  $\mathbf{A}$  og vektorene  $\mathbf{x}$  og  $\mathbf{y}$  i ligningssystemet  $\mathbf{Ax}=\mathbf{y}$ .
2. Sett opp  $\mathbf{A}^T$  og regn ut  $\mathbf{B} = \mathbf{A}^T\mathbf{A}$  og  $\mathbf{c} = \mathbf{A}^T\mathbf{y}$
3. Regn ut  $\mathbf{B}^{-1}$  og bestem koeffisientene  $a, b, c$  til andregradsfunksjonen som passer best mulig til punktene.

Oppgavene 4.6.8-4.6.12 er oppgaver med minste kvadraters metode:

#### 4.6.8

Gitt punktene  $(1, \frac{1}{2})$ ,  $(2, 0)$  og  $(4, 1)$ . Bruk minste kvadraters metode til å bestemme ligningen for den rette linjen som er best mulig tilpasset disse punktene.

#### 4.6.9

Bruk minste kvadraters metode til å bestemme ligningen for den rette linjen som passer best mulig til disse punktene:  $(4, 3)$ ,  $(0, 1)$ ,  $(2, 0)$  og  $(3, 4)$ .

#### 4.6.10

I en spillscene skal spilleren bevege seg langs en parabelbane og plukke trofeer langs banen. Trofeene har koordinater

$$(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{3}{2}), (\frac{5}{2}, -1), (3, \frac{1}{2}), (4, 3)$$

Parabelbanen skal konstrueres ved hjelp av minste kvadraters metode.

1. Skriv opp matrisene  $\mathbf{A}$ ,  $\mathbf{A}^T$  og vektoren  $\mathbf{y} = \mathbf{A}^T\mathbf{b}$
2. Bestem ligningen for andregradsfunksjonen som definerer parabelbanen.

#### 4.6.11

I en spillscene skal spilleren bevege seg langs en parabelbane og plukke trofeer langs banen. Trofeene har koordinater

$$(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{3}{2}), (\frac{5}{2}, -1), (3, \frac{1}{2}), (4, 3)$$

Parabelbanen skal konstrueres ved hjelp av minste kvadraters metode.

1. Skriv opp matrisene  $\mathbf{A}$ ,  $\mathbf{A}^T$  og vektoren  $\mathbf{y} = \mathbf{A}^T\mathbf{b}$
2. Bestem ligningen for andregradsfunksjonen som definerer parabelbanen.

#### 4.6.12

Bruk av minste kvadraters metode i spill.

1. Bestem koordinater til 7-8 punkter i planet for en scene. Tegn en skisse.
2. Vi skal lage en parabelbane som passer best mulig til disse punktene.
3. Bruk et C++ bibliotek til å regne ut matrisene som trengs for å bruke minste kvadraters metode, og vis utregningen. Dokumenter i rapport.
4. Bruk et egnet verktøy til å regne ut løsningen.
5. Beregn punkter (x,y) og lagre i en array eller på fil.
6. Punktene og grafen til funksjonen skal visualiseres.

#### 4.6.13

Bestem et polynom av grad  $\leq 3$  som interpolerer punktene  $p_0(1, 0)$  og  $p_1(\pi, 0)$  og hvor de deriverte er henholdsvis 1 og -1 (vi skriver  $p'(x_0) = 1$  og  $p'(x_1) = -1$ ).

#### 4.6.14

La  $x_0 = 0$ ,  $x_1 = 1$ ,  $y_0 = 0$  og  $y_1 = 1$  og gitt funksjonen  $g(x) = \sin \frac{\pi}{2}x$ . La  $p(x)$  være polynomet for kubisk Hermite interpolasjon av  $g$  på intervallet  $[0, 1]$ .

- Regn ut  $y'_0 = g'(0)$ ,  $y'_1 = g'(1)$  og sett opp interpolasjonsproblemet på matriseform  $\mathbf{Ax}=\mathbf{b}$  hvor  $\mathbf{b}^T = [y_0 \ y_1 \ y'_0 \ y'_1]$ .
- Bruk resultatet fra a) til å bestemme interpolasjonspolynomet  $p(x)$ .

#### 4.6.15

Kubiske Hermite interpolasjon av  $x^4$  på intervallet  $[0, 1]$ . Bruk ligningene 4.10 og 4.11 og sett opp først de to ligningene for interpolasjon i endepunktene, deretter de to ligningene for interpolasjon av de deriverte i endepunktene.

Sett så opp ligningene på matriseform  $\mathbf{Ax}=\mathbf{b}$  hvor  $\mathbf{x}^T = [a \ b \ c \ d]$  og  $\mathbf{b}^T = [y_0 \ y_1 \ y'_0 \ y'_1]$ . Vis at

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

og  $\mathbf{b}^T = [0 \ 1 \ 0 \ 4]$ . Bestem til slutt  $\mathbf{A}^{-1}$  (kalles Hermite matrisen) og  $\mathbf{x}$  og sett opp interpolasjonspolynomet.

#### 4.6.16

Vi har nå to metoder for å gjøre kubisk Hermite interpolasjon. La oss for anledningen kalle den uavhengige variabelen  $t$  i stedet for  $x$ . Bestem en kubisk Hermite interpolant for følgende fire sett av interpolasjonsbetingelser:

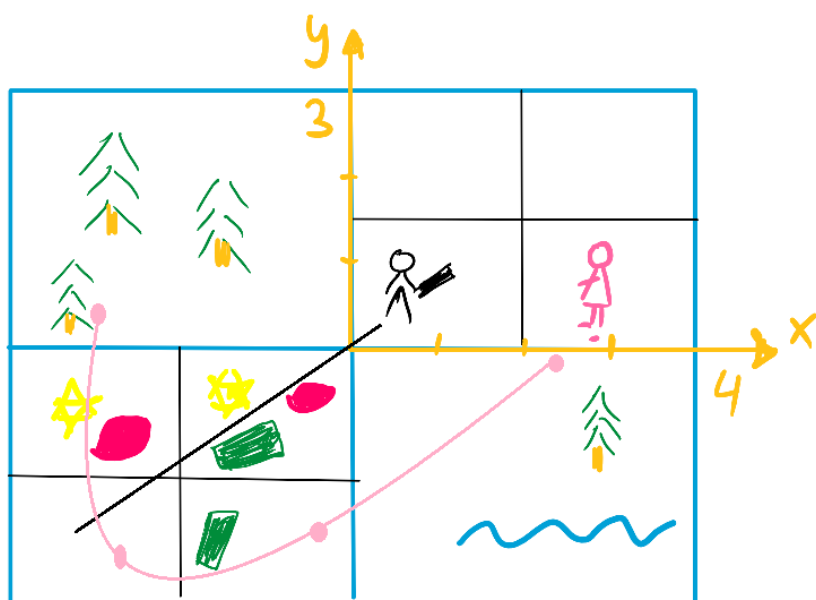
$t_0$	$y_0$	$y'_0$	$t_1$	$y_1$	$y'_1$
0	1	-3	1	0	0
0	0	3	1	0	0
0	0	0	1	0	-3
0	0	0	1	1	3

#### 4.6.17

Bruk av interpolasjon i spill. Et forslag til scene er figur 4.10

- Bestem koordinater til 4 punkter for interpolasjon med kubisk polynom. Sett opp matriser og vis nødvendige utregninger.
- Beregn punkter (x,y) og lagre i en array eller på fil.
- Punktene og grafen til funksjonen skal visualiseres.





Figur 4.10: Eksempel på scene. Kurven er ikke grafen til en funksjon!

# Kapittel 5

## Affint rom

### 5.1 Introduksjon

Vi har vektorrom, men utgangspunktet er i praksis ofte punkter. Et vektorrom har en algebra (tillatte regneoperasjoner som gir mening, for eksempel skalar-multiplikasjon og addisjon, en nullvektor) og vi ønsker et rom som inneholder både inneholder vektorer og punkter og som har en algebra for punkter. Dette kalles et affint rom. I dette kapitlet går vi gjennom noen grunnleggende egenskaper ved affine rom: Hva er et punkt, koordinatfri geometri, parameterframstilling av en rett linje, barysentriske koordinater. (Dette er grundigere forklart for eksempel i [4], [5], som anbefales).

#### 5.1.1 Punkt og vektor

Hva er et punkt? For å svare på spørsmålet, er det kanskje en ide å sammenligne med noe vi kan, en vektor. Vi kjenner til vektorer fra avsnitt 3.2. En vektor  $\mathbf{v}$  kan vi skrives som en lineærkombinasjon av vektorer  $v_0, \dots, v_n$  og konstanter  $c_0, \dots, c_n$

$$\mathbf{v} = \sum_{i=0}^n c_i \mathbf{v}_i, \quad (5.1)$$

En vektor har størrelse og retning, men ingen posisjon. Et punkt har posisjon, men ingen størrelse eller retning. Vektoraddisjon er definert, og differansen mellom to punkter  $P$  og  $Q$  er en vektor

$$\mathbf{v} = Q - P$$



Figur 5.1: Barysentriske koordinater for et punkt på en linje gjennom to punkter.

fra  $Q$  til  $P$ . Men hva med summen  $P+Q$ ? Kun i enkelte tilfeller gir en sum av punkter mening. Midtpunktet på linjen mellom  $P$  og  $Q$  kan skrives som

$$\frac{P+Q}{2} = P + \frac{Q-P}{2} = P + \frac{\mathbf{v}}{2} \quad (5.2)$$

Her ser vi at et punkt kan uttrykkes som et punkt + en vektor, hvor  $\mathbf{v} = Q - P$ . Disse punktene er forskjellige, men tilfeldig valgt. Lar vi  $Q$  være et fast punkt og  $P$  variere, blir  $\mathbf{v}_i = Q - P_i$

### 5.1.2 Affin kombinasjon

Vi skal nå vise at det fins en lignende måte å skrive et punkt som en kombinasjon av andre punkter på. Vi har sett i likning (5.2) at det kan gi mening for to punkter. Hva skal til for at det gir mening for flere enn to punkter? Altså, for punkter  $P_0, \dots, P_n$  og konstanter  $c_0, \dots, c_n$  skal vi kunne skrive et punkt som en kombinasjon av andre punkter,

$$P = \sum_{i=0}^n c_i P_i$$

Siden  $\mathbf{v} = Q - P$  kan vi altså skrive  $P = Q - \mathbf{v}$ . Videre setter inn uttrykket for  $\mathbf{v}$  fra likning (5.1), med  $\mathbf{v}_i = Q - P_i$  og notasjonen  $\tilde{c}_i$  for konstantene.

$$\begin{aligned} P &= Q - \sum_{i=0}^n \tilde{c}_i \mathbf{v}_i \\ &= Q + \sum_{i=0}^n c_i \mathbf{v}_i \\ &= Q + \sum_{i=0}^n c_i P_i - \sum_{i=0}^n c_i Q \\ &= Q + \sum_{i=0}^n c_i P_i - Q \left( \sum_{i=0}^n c_i \right) \end{aligned}$$

hvor  $c_i = -\tilde{c}_i$ .

Vi ser fra siste linje at vi kan skrive

$$P = \sum_{i=0}^n c_i P_i, \quad \text{hvis} \quad \sum_{i=0}^n c_i = 1 \quad (5.3)$$

Dette kalles en affin kombinasjon eller en barysentrisk kombinasjon.

### 5.1.3 Linje

Linjen som går gjennom to punkter  $P$  og  $Q$  (se figur 5.1) kan uttrykkes ved en parametrisering.

$$(1-t)P + tQ = P + t\mathbf{v} \quad (5.4)$$

hvor  $\mathbf{v} = Q - P$  som ovenfor og  $t \in \mathbf{R}$ . Mellom  $P$  og  $Q$  er  $0 \leq t \leq 1$ . Til venstre for  $P$  er  $t < 0$  og til høyre for  $Q$  er  $t > 1$ . Venstresiden av ligningen er altså en affin kombinasjon av to punkter, et nytt punkt.

### 5.1.4 Konveksitet

Et rom (eller en mengde) er konvekst dersom den rette linjen mellom to vilkårlige punkter også ligger i rommet (mengden). Linjen mellom P og Q er dermed en konveks mengde, og for  $0 \leq t \leq 1$  kalles affinkombinasjonen i (5.4) en konvekskombinasjon.

### 5.1.5 Parameterframstilling av en rett linje

Vi går tilbake til figur 5.1. En parameterframstilling av linja gjennom P og Q er  $x(t) = (1-t)P + tQ$ . Alternativt med vektoren  $\mathbf{v} = Q - P$  får vi  $x(t) = P + t\mathbf{v} = P + t(Q - P)$ .  $1-t$  og  $t$  kalles de barysentriske koordinatene til P og Q, og når  $0 \leq t \leq 1$  får vi et punkt på linjestykket mellom P og Q.

Vi kan finne igjen et hvilket som helst punkt på linja slik: Vi starter i punktet P. Deretter går vi et tall, en konstant  $c$  multiplisert med retningsvektoren. Vi får en affin sum som er slik:  $P + c\mathbf{v}$ . Alternativt kan du tenke deg vektoren  $\mathbf{a} = \vec{OP}$  og starte i origo.

- $c = 0$  gir punktet P.
- $c = \frac{1}{2}$  gir punktet midt mellom P og Q.
- $c = 1$  gir punktet Q.
- $c = 2$  gir et punkt like langt unna Q som P er, på motsatt side.

Dette hittil anonyme punktet på linja som vi har omtalt, har koordinater  $(x, y)$ . Når vi har med en parametrisering å gjøre, er ikke  $y$  en funksjon av  $x$ , men både  $x$  og  $y$  er parametriske med  $t$  som parameter. Vi skriver gjerne

$$x = x(t) \text{ og } y = y(t)$$

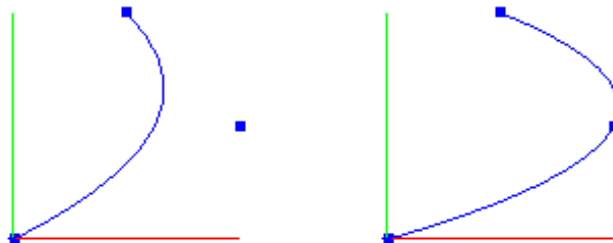
Vi kaller gjerne punktet  $\mathbf{x}(t) = (x(t), y(t))$ , og i tre dimensjoner har vi i tillegg  $z(t)$ . Da skrives  $\mathbf{x}(t) = (x(t), y(t), z(t))$ .

### 5.1.6 Affin kombinasjon av tre punkter

En affin kombinasjon av to punkter er altså en rett linje gjennom punktene. Hva hvis vi øker til tre punkter, hva blir da en affin kombinasjon? Øvre summasjonsgrense i (5.3) åpner jo opp for å kombinere flere enn to punkter  $P_i$  så lenge koeffisientene/vektene  $c_i$  summerer seg til 1. La oss først se på et par eksempler, på figur 5.2. For begge kurvene på figur 5.2 gjelder at et hvilket som helst punkt på kurven er en affin kombinasjon av de tre punktene markert med blå ruter. Begge kurvene er altså laget ved å blande, eller kombinere, tre punkter i henhold til (5.3). Vi får mye bruk for denne type kurver, og det er derfor veldig interessant for oss å finne et matematisk verktøy for å beregne koeffisientene/vektene  $c_i$  i (5.3).

### 5.1.7 Barysentriske koordinater for en trekant

Vi skal nå innføre et eget, lokalt koordinatsystem for *en* trekant. Vi har nettopp lært om barysentriske koordinater for en linje, se figur 5.1. Vi har sett at vi kan



Figur 5.2: To eksempler på affin kombinasjon av punktene  $P_0(0, 0, 0)$ ,  $P_1(1, \frac{1}{2}, 0)$  og  $P_2(\frac{1}{2}, 1, 0)$ .

kombinere tre punkter og få en kurve. Vi kan også legge til ett ekstra punkt (fra linjeeksemplet) og få en trekant! Se figur 5.3. Her har de barysentriske koordinatene for linjen gjennom P og Q fått nye navn. Summen av dem er fortsatt 1. Det er ganske opplagt at det tredje punktet R ikke kan ha noen innvirkning på et punkt på linjen gjennom P og Q.

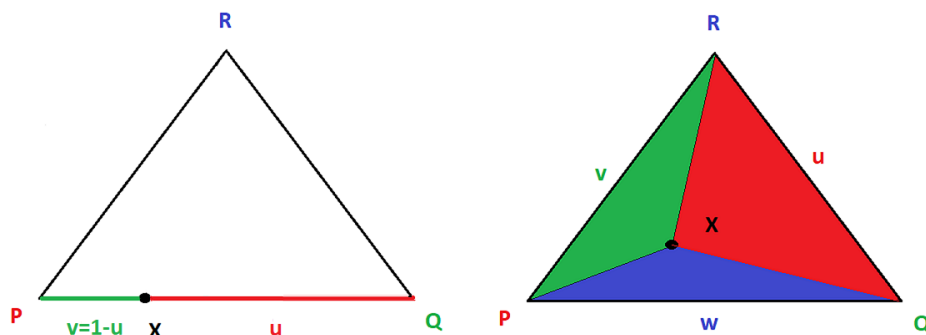
Hvis vi nå flytter punktet X inn i trekanten, kan vi (forhåpentligvis) forestille oss intuitivt at X kan skrives som en kombinasjon av P, Q og R. De barysentriske koordinatene til punktet X med hensyn på P, Q og R er henholdsvis  $u$ ,  $v$  og  $w$ . Vi har  $u + v + w = 1$  og størrelsen på disse barysentriske koordinatene er indikert med en farget trekant. Vi ser at når X ligger på linjen gjennom P og Q, forsvinner den blå trekanten og  $w = 0$ . Tilsvarende gjelder for de to andre barysentriske koordinatene. Et punkt X kan altså skrives som en barysentrisk kombinasjon av hjørnene P, Q og R til en trekant T, og denne trekanten T er valgt vilkårlig.

La  $Boundary(T)$  betegne kantene som omslutter trekanten T og la  $Int(T)$  betegne det indre området til T. Vi har

$$\begin{aligned}
 u + v + w &= 1 \\
 u \geq 0, v \geq 0, w \geq 0, & \quad X \in T \\
 u > 0, v > 0, w > 0, & \quad X \in Int(T) \\
 u = 0 \parallel v = 0 \parallel w = 0 & \quad X \in Boundary(T) \\
 u < 0 \parallel v < 0 \parallel w < 0 & \quad X \notin T
 \end{aligned} \tag{5.5}$$

## 5.2 Affint rom og basisfunksjoner

Et affint rom  $\mathbb{A}$  består altså av både punkter og vektorer. Hvis vi for eksempel starter med tre punkter  $P, Q, R \in \mathbb{R}^2$  som ikke er kolineære (ligger på en rett linje), kan vi bestemme de to lineært uavhengige vektorene  $Q - P$  og  $R - P$ . Differanse mellom to punkter er definert fra vektorrom, gir mening i et affint rom, og det affine rommet er opplagt assosiert med et vektorrom. I [4] brukes formuleringen å utvide det affine rommet (som i utgangspunktet består av



Figur 5.3: Barysentriske koordinater til et punkt i en trekant. Se også figur 5.1.

punkter) med vektorer.

Hvis vi har tre ikke kolineære punkter  $P, Q, R \in \mathbb{R}^3$ , kan vi også bestemme de to lineært uavhengige vektorene  $Q - P$  og  $R - P$ . Disse to utspenner nå et plan, altså et underrom av  $\mathbb{R}^3$ . Dimensjonen til et affint rom  $\mathbb{A}$  kan derfor være lavere enn antall koordinater til punktene. Dimensjonen til et rom er som vi husker antall lineært uavhengige vektorer i rommet. Det vanligste for oss vil være å bruke et affint rom med dimensjon 2 eller dimensjon 3.

Addisjon av punkter, altså affinkombinasjon, krever at  $\sum c_i = 1$ . Hvis vi har 3 punkter, trenger vi 3 koeffisienter  $c_0, c_1, c_2$  med sum  $c_0 + c_1 + c_2 = 1$ . Det har vi nettopp sett eksempler på. Hvis vi har 4 punkter, trenger vi 4 koeffisienter  $c_0, c_1, c_2, c_3$  med sum  $c_0 + c_1 + c_2 + c_3 = 1$ . Vi skal i resten av dette kapitlet se eksempler på hvordan vi kan bruke ulike typer basisfunksjoner til å gjøre affinkombinasjoner av punkter. Vi skal benytte (lineært uavhengige) basisfunksjoner som summerer seg opp til 1 - dermed kan vi bruke basisfunksjoner som koeffisienter i affinkombinasjoner.

Vi har tidligere, i kapittel 4 sett hvordan vi bruker basisfunksjoner (monomial-basisen) til å konstruere interpolasjonspolynomer. Nå skal vi bruke basisfunksjoner på en annen måte, til å konstruere parametriske kurver. Vi skal studere de tre typene Bernstein basisfunksjoner, Lagrange basisfunksjoner og B-spline basisfunksjoner. Senere i kapittel 7 skal vi studere algoritmer knyttet til disse: deCasteljau algoritmen, Neville's algoritme og deBoor's algoritme.

## 5.3 Bernstein basisfunksjoner

Vi skal nå kombinere det vi kan om vektorbasiser (avsnitt 3.2) og parameterframstillingen av en rett linje til å introdusere de lineære, kvadratiske og kubiske Bernstein basisfunksjonene. Kurven til venstre på figur 5.2 er tegnet med kvadratiske Bernstein basisfunksjoner.

### 5.3.1 Lineære basisfunksjoner

La oss nå bytte ut navnet på  $x$ -aksen til  $t$ -aksen og tegne opp de rette linjene  $y = t$  og  $y = 1 - t$  for  $0 \leq t \leq 1$ . Det er klart at

$$1 - t + t = 1, \quad (5.6)$$

vi har en affin kombinasjon av punktene. I tillegg plasserer vi punktene P og Q fra figur 5.1 som vist på figur 5.4. Vi ser at

- $t$  og  $1 - t$  summerer seg til 1 (opplagt).
- $t$  er 1 der hvor  $1 - t$  er 0 og omvendt (også opplagt).
- $t$  og  $1 - t$  er lineært uavhengige.

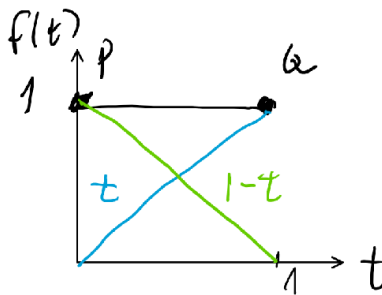
Men da er  $t$  og  $1 - t$  lineære basisfunksjoner. Vi kaller dem her Bernstein basisfunksjoner av grad 1.

### 5.3.2 Kvadratiske basisfunksjoner

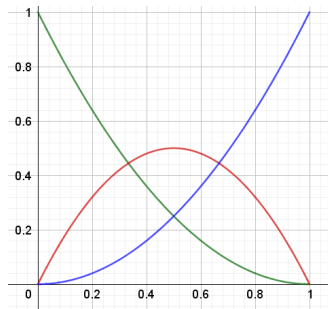
Hvis vi kvadrerer hver side av likning (5.6), og betrakter det som en ligning, må venstresiden fortsatt være 1. Vi får

$$((1 - t) + t)^2 = (1 - t)^2 + 2t(1 - t) + t^2 \quad (5.7)$$

De tre uttrykkene på høyre side er alle kvadratiske og summerer seg opp til 1 uavhengig av hva  $t$  er. De kalles Bernstein basisfunksjoner av grad 2.



Figur 5.4: Lineære basisfunksjoner.



Figur 5.5: Kvadratiske Bezierkurver kan lages ved å kombinere disse Bernstein-basisfunksjonene av grad  $d=2$ .

### Matriseform

Når vi skriver de kvadratiske Bernstein basisfunksjonene på standardform (potensform) får vi  $(1-t)^2 = t^2 - 2t + 1$  og  $2t(1-t) = -2t^2 + 2t$ . På matriseform blir dette

$$\mathbf{B}\mathbf{x} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}$$

hvor  $\mathbf{x}^T = [t^2 \ t \ 1]$  Vi kan da skrive ligning 5.11 på matriseform

$$\mathbf{c}^T \mathbf{B}\mathbf{x}$$

### Eksempel

I dette eksemplet viser vi hvordan vi gjør utregninger med parametriske kurver. Her skal vi bruke de kvadratiske Bernstein basisfunksjonene og kontrollpunkter som ligger i planet (de har to koordinater). Parameterframstillingen av en kurve kaller vi  $\mathbf{p}(t) = (x(t), y(t))$ . Vi kan også betrakte  $\mathbf{p}(t)$  som en vektorfunksjon.

La  $\mathbf{c}$  være gitt ved  $c_0(4, 1)$ ,  $c_1(1, 2)$  og  $c_2(3, 4)$ . Vi skal regne ut  $\mathbf{p}(t) = \mathbf{c}^T \mathbf{B}\mathbf{x}$ .

Vi får

$$\mathbf{p}(t) = (4, 1) \cdot (t^2 - 2t + 1) + (1, 2) \cdot (-2t^2 + 2t) + (3, 4) \cdot t^2$$

$\mathbf{p}(t) = (x(t), y(t))$  og vi regner ut  $x(t)$  og  $y(t)$  hver for seg:

$$\begin{aligned} x(t) &= 4 \cdot (t^2 - 2t + 1) + 1 \cdot (-2t^2 + 2t) + 3 \cdot t^2 = 5t^2 - 6t + 4 \\ y(t) &= 1 \cdot (t^2 - 2t + 1) + 2 \cdot (-2t^2 + 2t) + 4 \cdot t^2 = t^2 + 2t + 1 \end{aligned}$$

Når vi setter inn  $t = 0$  får vi  $(4, 1)$  og  $t = 1$  gir  $(3, 4)$ . Den parametriske kurven interpolerer altså  $c_0$  og  $c_2$ . Vi kan i tillegg regne ut punkter på kurven for så mange  $t$ -verdier vi måtte ønske. For eksempel  $t = \frac{1}{2}$  gir  $(\frac{9}{4}, \frac{9}{4})$ .

Det siste punktet kan beregnes på en annen måte: Finn midtpunktet  $c_{01}$  mellom  $c_0$  og  $c_1$ , og midtpunktet  $c_{12}$  mellom  $c_1$  og  $c_2$ . Finn deretter midtpunktet mellom  $c_{01}$  og  $c_{12}$ . Dette skal vi se nærmere på i avsnitt 7.2.





Figur 5.6: Kubiske Bezierkurver kan lages ved å kombinere disse Bernsteinbasisfunksjonene av grad  $d=3$ .

### 5.3.3 Kubiske basisfunksjoner

Hvis vi fortsetter enda et steg, får vi

$$\begin{aligned}
 ((1-t) + t)^3 &= ((1-t) + t)((1-t) + t)^2 \\
 &= ((1-t) + t)[(1-t)^2 + 2t(1-t) + t^2] \\
 &= (1-t)[(1-t)^2 + 2t(1-t) + t^2] + t[(1-t)^2 + 2t(1-t) + t^2] \\
 &= (1-t)^3 + 3t(1-t)^2 + 3t^2(1-t) + t^3
 \end{aligned} \tag{5.8}$$

Dette er Bernstein basisfunksjonene av grad 3. Det er som vi ser fire av dem, og vi så ovenfor at vi har tre i det kvadratiske tilfellet. Med binomialkoeffisientene

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = \frac{1 \cdot 2 \dots n}{1 \cdot 2 \dots r \cdot 1 \cdot 2 \dots (n-r)} = \frac{(n-r+1) \dots n}{1 \cdot 2 \dots r}$$

kan hver og en av dem skrives

$$B_i^d(t) = \binom{d}{i} t^i (1-t)^{d-i} \tag{5.9}$$

Vi har sett at vi har 3 basisfunksjoner av grad  $d=2$  og 4 basisfunksjoner av grad  $d=3$ . Antall basisfunksjoner for en bestemt grad  $d$  er altså  $d+1$ . Ved å kombinere basisfunksjoner av grad  $d$  på forskjellige måter, kan vi uttrykke et hvilket som helst polynom av grad  $d$ . Vi har tidligere sett at monomialene  $\{1, x, x^2, \dots, x^d\}$  utgjør en basis for rommet av polynomer av grad mindre eller lik  $d$ . Vi har nå ytterligere en basis for dette rommet, slik at for et polynom av grad mindre eller lik  $d$  kan vi skrive

$$p(t) = \sum_{i=0}^d c_i B_i^d(t) \tag{5.10}$$

hvor  $c_i$  er reelle tall. Dette gjelder for en dimensjon, vi kan tenke oss  $t$  som uavhengig variabel og  $p(t)$  som en funksjon av  $t$ . Når vi får med kurver i  $\mathbf{R}^2$  å gjøre, gjelder resultatet for  $\mathbf{p}(t) = (x(t), y(t))$ . For kurver i  $\mathbf{R}^3$  skriver vi  $\mathbf{p}(t) = (x(t), y(t), z(t))$ . Vi har altså

$$\mathbf{p}(t) = \sum_{i=0}^d c_i \mathbf{B}_i^d(t) = \mathbf{c}^T \mathbf{B}(t) \tag{5.11}$$

hvor  $c_i$  er reelle tall og hver  $\mathbf{B}(t) = (x(t), y(t), z(t))$ . Skrivemåten

$$\mathbf{p}(t) = \mathbf{c}^T \mathbf{B}(t), \quad \mathbf{c}^T = [c_0 \ \dots \ c_n] \in \mathbf{R}^n$$

er mye brukt.

Et matematisk rom kan som vi har sett tidligere være et rom som består av alle mulige polynomer av grad mindre eller lik 3. Dette rommet kan ha betegnelsen  $\mathcal{P}^3$ . Funksjonene i ligning 5.8 danner en basis for  $\mathcal{P}^3$ . Det betyr at et hvilket som helst polynom i dette rommet,  $p \in \mathcal{P}^3$ , kan uttrykkes som en lineærkombinasjon av basisfunksjonene.

### Matriseform

Vi kan bruke ligning (5.9) til å bestemme Bernstein basis funksjonene av grad 3 i ligning (5.8),  $(1-t)^3$ ,  $3t(1-t)^2$ ,  $3t^2(1-t)$ ,  $t^3$ . Når vi skriver polynomene på standardform (potensform) får vi

$$\begin{aligned} (1-t)^3 &= -t^3 + 3t^2 - 3t + 1 \\ 3t(1-t)^2 &= 3t^3 - 6t^2 + 3t \\ 3t^2(1-t) &= -3t^3 + 3t^2 \\ t^3 & \end{aligned}$$

Vi kan sette opp dette på matriseform,

$$\mathbf{B}\mathbf{x} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

hvor  $\mathbf{x}^T = [t^3 \ t^2 \ t \ 1]$  Vi kan da skrive ligning 5.11 på matriseform

$$\mathbf{c}^T \mathbf{B}\mathbf{x}$$

Sammenligne med oppgave 5.7.2.

### 5.3.4 Derivasjon

Vi oppsummerer først noe av det vi har gått gjennom om Bernstein basisfunksjoner og Bezierkurver. En (parametrisk) Bezierkurve  $p(t) = (x(t), y(t))$  av grad  $d$  kan skrives

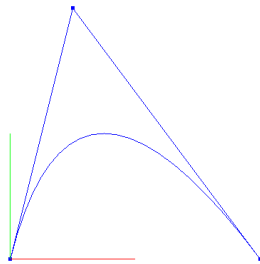
$$p(t) = \sum_{i=0}^d P_i b_i(t)$$

hvor  $P_i$ ,  $i = 0, 1, \dots, d$  er kontrollpunktene og  $b_i(t)$ ,  $i = 0, 1, \dots, d$  er Bernstein basisfunksjonene av grad  $d$ . Med  $d = 2$  får vi

$$p(t) = P_0 b_0(t) + P_1 b_1(t) + P_2 b_2(t) \quad (5.12)$$

hvor

$$\begin{aligned} b_0(t) &= t^2 - 2t + 1 \\ b_1(t) &= -2t^2 + 2t \\ b_2(t) &= t^2 \end{aligned}$$



Figur 5.7: Kvadratisk Bezierkurve med kontrollpunktene  $P_0(0,0)$ ,  $P_1(\frac{1}{2}, 2)$  og  $P_2(2,0)$ .

Når vi regner ut uttrykket for kurven (som i eksemplet ovenfor) får vi

$$\begin{aligned} x(t) &= 0 + \frac{1}{2} \cdot (-2t^2 + 2t) + 2 \cdot t^2 = t^2 + t \\ y(t) &= 0 + 2 \cdot (-2t^2 + 2t) + 0 = -4t^2 + 4t \\ p\left(\frac{1}{2}\right) &= \left(\frac{3}{4}, 1\right) \end{aligned}$$

Den deriverte til en parametrisk kurve  $p(t) = (x(t), y(t))$  i en parameterverdi  $t_0$  er en vektor som er tangent til kurven i  $t_0$ . Figur 5.7 viser et nytt eksempel på en kvadratisk Bezierkurve, med kontrollpunkter og kontrollpolygon. Kurven tangerer kontrollpolygonet i endepunktene. Da må vektoren  $P_1 - P_0 = [\frac{1}{2}, 2]$  være den deriverte i  $t = 0$  skalert, og vektoren  $P_2 - P_1 = [\frac{3}{2}, -2]$  må være den deriverte i  $t = 1$  skalert. Vi kan beregne den deriverte i et hvilket som helst punkt på kurven ved å derivere likning (5.12):

$$p'(t) = P_0 b'_0(t) + P_1 b'_1(t) + P_2 b'_2(t) \quad (5.13)$$

hvor

$$\begin{aligned} b'_0(t) &= 2t - 2 \\ b'_1(t) &= -4t + 2 \\ b'_2(t) &= 2t \end{aligned}$$

Vi får da

$$\begin{aligned} x'(t) &= 0 + \frac{1}{2} \cdot (-4t + 2) + 2 \cdot 2t = 2t + 1 \\ y'(t) &= 0 + 2 \cdot (-4t + 2) + 0 = -8t + 4 \end{aligned}$$

Altså er  $p'(t) = (x'(t), y'(t)) = (2t + 1, -8t + 4)$ . (Her kunne vi benyttet hakeparenteser for å understreke at den deriverte er en vektor). Når vi setter inn for  $t = 0$  og  $t = 1$  får vi

$$\begin{aligned} p'(0) &= [1, 4] = 2(P_1 - P_0) \\ p'(1) &= [3, -4] = 2(P_2 - P_1) \end{aligned}$$

Hvis vi nå bruker likning (5.13) for  $t = \frac{1}{2}$ , får vi  $x'(\frac{1}{2}) = 2$ ,  $y'(\frac{1}{2}) = 0$  og  $p'(\frac{1}{2}) = [2, 0]$  som er en vektor parallell med  $x$ -aksen og tangent til kurven i punktet  $(\frac{3}{4}, 1)$ .

## 5.4 Lagrange basisfunksjoner

Kurven til høyre på figur 5.2 er tegnet med kvadratiske Lagrange basisfunksjoner. Vi har nå sett eksempler på at en rett linje mellom punktene  $P_0$  and  $P_1$  kan parametriseres  $(1-t)P_0 + tP_1$ ,  $0 \leq t \leq 1$ . Hvis vi endrer parameterintervallet fra  $[0, 1]$  til  $[t_0, t_1]$  kan vi uttrykke den samme linjen ved

$$\frac{t_1 - t}{t_1 - t_0}P_0 + \frac{t - t_0}{t_1 - t_0}P_1, \quad t_0 \leq t \leq t_1$$

Gitt  $d$  punkter  $P_i$ ,  $i = 0, \dots, d$ , reelle tall  $t_i$ ,  $i = 0, \dots, d$ , og funksjoner

$$f_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i}, \quad i = 0, \dots, d-1 \quad (5.14)$$

Da har vi for  $d = 1$   $1 - f_{i,d}(t) = 1 - \frac{t-t_i}{t_{i+1}-t_i} = \frac{t_{i+1}-t}{t_{i+1}-t_i}$ , og for hvert par av punkter  $\{P_i, P_{i+1}\}$  kan den rette linjen  $P_i P_{i+1}$  fra  $P_i$  til  $P_{i+1}$  skrives som en affin kombinasjon

$$(1 - f_{i,1}(t))P_i + f_{i,1}(t)P_{i+1} = \frac{t_{i+1} - t}{t_{i+1} - t_i}P_i + \frac{t - t_i}{t_{i+1} - t_i}P_{i+1}, \quad t_i \leq t \leq t_{i+1} \quad (5.15)$$

Vi ser tydelig at  $f_{i,1}(t)$  interpolerer  $P_i$  i  $t = t_i$  og  $P_{i+1}$  i  $t_{i+1}$ ,  $i = 0, \dots, d-1$ .

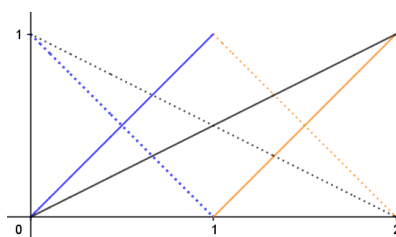
Lagrange funksjonene (se for eksempel [11] for en grundigere beskrivelse), § 2.3) er gitt ved

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

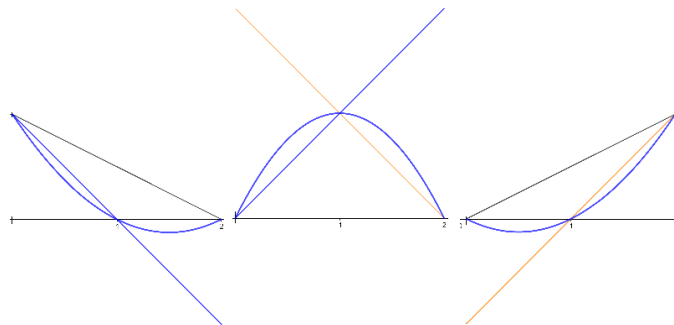
Gitt et polynom av grad  $d$  kan Lagrange basisfunksjonene skrives

$$L_j^d(t) = \prod_{i=0, i \neq j}^d \frac{t - t_i}{t_j - t_i} \quad (5.16)$$

([5], §2.5). Vi gjenkjenner  $f_{i,d}(t)$  fra (5.14) som  $w_{i,d}(t)$  i (5.18) og de er relatert til  $L_i^d(t)$  i (5.16).



Figur 5.8: Lineære funksjoner  $f_{i,d}(t)$  i (5.14) for  $i=0,1,2$  aog  $d=1,2$  brukes til å konstruere den kvadratiske Lagrange basisen  $L_0^2(t), L_1^2(t), L_2^2(t)$ . For eksempel  $L_0^1(t) = \frac{t-t_0}{t_1-t_0} = t$ .



Figur 5.9: Kvadratiske Lagrange basis funksjoner på 0, 1, 2.

### 5.4.1 Matriseform

Når vi skriver ut de kvadratiske Lagrange basisfunksjonene i (5.16), får vi

$$L_0^2(t) = \frac{(t-t_1)(t-t_2)}{(t_0-t_1)(t_0-t_2)} \quad L_1^2(t) = \frac{(t-t_0)(t-t_2)}{(t_1-t_0)(t_1-t_2)} \quad L_2^2(t) = \frac{(t-t_0)(t-t_1)}{(t_2-t_0)(t_2-t_1)}$$

og med  $t_0 = 0$ ,  $t_1 = 1$ ,  $t_2 = 2$  blir dette på standardform

$$L_0^2(t) = \frac{1}{2}t^2 - \frac{3}{2}t + 1, \quad L_1^2(t) = -t^2 + 2t, \quad L_2^2(t) = \frac{1}{2}t^2 - \frac{1}{2}t$$

Hvis vi har punkter  $c_0$ ,  $c_1$ ,  $c_2$ , kan vi på samme måte som i (5.10) uttrykke en parametrisk kurve som en affinkombinasjon med Lagrange basisfunksjoner av disse punktene:

$$p(t) = \sum_{i=0}^d c_i L_i^d(t) \quad (5.17)$$

På matriseform blir dette

$$\mathbf{L}\mathbf{x} = \begin{bmatrix} \frac{1}{2} & -\frac{3}{2} & 1 \\ -1 & 2 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix} \cdot \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}$$

hvor  $\mathbf{x}^T = [t^2 \ t \ 1]$  Vi kan da skrive ligning 5.17 på matriseform  $\mathbf{c}^T \mathbf{L}\mathbf{x}$  på samme måte som for Bernstein basisfunksjoner.

### 5.4.2 Eksempel

Gitt tre kontrollpunkter  $c_0(1, 3)$ ,  $c_1(1, 1)$  og  $c_2(3, 1)$ . Vi skal regne ut en kvadratisk parametrisk kurve som interpolerer disse punktene ved å bruke (5.17) (eller matriseformen).

$$p(t) = \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot (1, 3) + (-t^2 + 2t) \cdot (1, 1) + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot (3, 1)$$

Vi har  $p(t) = (x(t), y(t))$  og får et uttrykk for  $x(t)$  og et uttrykk for  $y(t)$ .

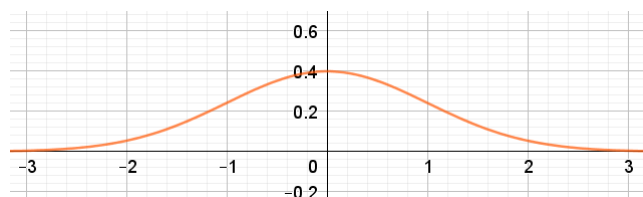
$$x(t) = \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot 1 + (-t^2 + 2t) \cdot 1 + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot 3 = t^2 - t + 1$$

$$y(t) = \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot 3 + (-t^2 + 2t) \cdot 1 + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot 1 = t^2 - 3t + 3$$

Altså

$$p(t) = (x(t), y(t)) = (t^2 - t + 1, t^2 - 3t + 3)$$

Innsetting av  $t$ -verdier gir  $p(0) = (1, 3)$ ,  $p(1) = (1, 3)$  og  $p(2) = (3, 1)$ , som det skal være, for hver av parameterverdiene  $(0, 1, 2)$  er en av Lagrange basisfunksjonene 1 og de andre to er 0.



Figur 5.10: Normalfordelingen.

## 5.5 B-spline basisfunksjoner

### 5.5.1 Introduksjon

Normalfordelingsfunksjonen (Gaussfordelingsfunksjonen) er gitt ved

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty$$

Med  $\mu = 0$  og  $\sigma = 1$  blir funksjonen

$$\frac{1}{\sqrt{2\pi}} e^{-x^2}$$

Dette er en av de mest berømte funksjonene i matematikken. Integralet

$$\int_{-\infty}^{\infty} g(x) dx = 1,$$

men det er umulig å regne ut analytisk. Den enklere funksjonen

$$h(x) = e^{-x^2}$$

er det heller ikke mulig å integrere analytisk. Grafen til begge funksjonene er klokkeformet. Hvis vi klarer å approksimere disse eksponentialfunksjonene med polynomer, blir det i det minste mulig å regne ut et integral. Med den stykkevis lineære funksjonen

$$f_1(x) = \begin{cases} \frac{1}{2}x + 1 & -2 \leq x < 0 \\ -\frac{1}{2}x + 1 & 0 \leq x < 2 \end{cases}$$

får vi en funksjon som følger  $h$  relativt godt ( $|h(x) - f_1(x)|$  er et lite tall) .

På samme måte, med den stykkevis lineære funksjonen

$$f_2(x) = \begin{cases} \frac{1}{4}x + \frac{1}{2} & -2 \leq x < 0 \\ -\frac{1}{4}x + \frac{1}{2} & 0 \leq x < 2 \end{cases}$$

får vi en funksjon som følger  $g$  relativt godt ( $|g(x) - f_2(x)|$  er et lite tall) og

$$\int_{-2}^2 f_2(x) dx = 1$$

Dette motiverer til å benytte splines, som er stykkevise polynomer. Vi ser på det enkleste stykkevise polynomet:

$$f(x) = \begin{cases} x & 0 \leq x < 1 \\ 2 - x & 1 \leq x < 2 \end{cases}$$

Dette er regnet ut som en lineær B-spline basisfunksjon i eksemplet nedenfor og tegnet med rødt på figur 5.11.

### 5.5.2 Rekursiv definisjon av B-splines

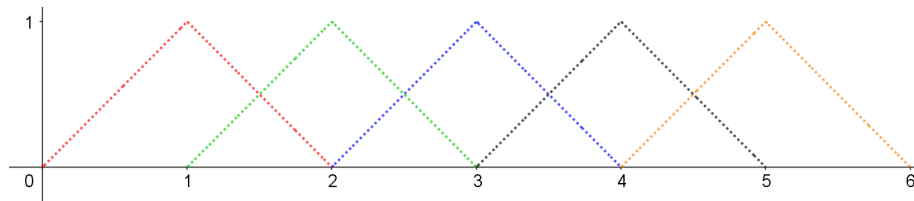
B-spline basisfunksjoner er polynomer av en grad  $d$ , definert på en såkalt skjøtvektor (knot vector på engelsk).  $\tau = \{t_0, t_1, \dots, t_{n+d+1}\}$ . (Som vi kan se, er ikke dette en vektor i det hele tatt, men den har fått dette navnet).  $t_0, t_1, \dots$  kalles skjøter. Skjøtvektoren må være progressiv, det betyr at verdien til en skjøt må være større eller lik den foregående skjøten.  $\{0, 1, 3, 2\}$  er altså ikke en tillatt skjøtvektor. B-spline basisfunksjoner kan defineres rekursivt som i (5.18) ([2]).

$$\begin{aligned} B_{i,d}(t) &= w_{i,d}(t) \cdot B_{i,d-1}(t) + (1 - w_{i+1,d}(t)) \cdot B_{i+1,d-1}(t) \\ \text{hvor } w_{i,d}(t) &= \begin{cases} \frac{t-t_i}{t_{i+d}-t_i} & t_i < t_{i+d} \\ 0 & \text{ellers} \end{cases} \\ \text{og } B_{i,0} &= \begin{cases} 1 & t_i \leq t < t_{i+1} \\ 0 & \text{ellers} \end{cases} \end{aligned} \quad (5.18)$$

#### B-spline notasjon

Vi bruker her notasjonen  $B_{i,d}(t)$  hvor  $d$  er graden til polynomet og  $i$  utgjør indekseringen i henhold til en skjøtvektor. Med denne notasjonen er det underforstått at vi har en skjøtvektor. Andre notasjoner som er brukt, er for eksempel  $b_{i,d}(t)$ ,  $b_{i,d}(t \mid t_0, \dots, t_{n+d+1})$ . Indekseringen angir hvor på skjøtvektoren den aktuelle basisfunksjonen har sitt domene, som vi ser på figur 5.11. En splinekurve betegnes gjerne med  $\mathbf{f}(t)$  eller bare  $f(t)$ .

B-spline basisfunksjoner brukes til å konstruere splinefunksjoner og splinekurver. En splinefunksjon av en variabel  $t$  konstrueres altså ved å kombinere B-spline basisfunksjoner.



Figur 5.11: Lineære B-spline basisfunksjoner på en uniform skjøtvektor.



### 5.5.3 Eksempel, lineær B-spline, d=1

Vi får

$$\begin{aligned} B_{i,1}(t) &= w_{i,1}(t) \cdot B_{i,0}(t) + (1 - w_{i+1,1}(t)) \cdot B_{i+1,0}(t) \\ &= \frac{t - t_i}{t_{i+1} - t_i} \cdot B_{i,0}(t) + \left(1 - \frac{t - t_{i+1}}{t_{i+2} - t_{i+1}}(t)\right) \cdot B_{i+1,0}(t) \end{aligned}$$

La oss benytte en uniform skjøtvektor, sette  $t_0 = 0, t_1 = 1, t_2 = 2, \dots$  og  $i = 0$  og regne ut den første lineære B-spline basisfunksjonen

$$\begin{aligned} B_{0,1}(t) &= w_{0,1}(t) \cdot B_{0,0}(t) + (1 - w_{1,1}(t)) \cdot B_{1,0}(t) \\ &= \frac{t - t_0}{t_1 - t_0} \cdot B_{0,0}(t) + \left(1 - \frac{t - t_1}{t_2 - t_1}\right) \cdot B_{1,0}(t) \\ &= t \cdot B_{0,0}(t) + (1 - (t - 1)) \cdot B_{1,0}(t) \\ &= t \cdot B_{0,0}(t) + (2 - t) \cdot B_{1,0}(t) \end{aligned}$$

Hva slags funksjon er dette? Her må vi bruke siste linje i B-spline definisjonen (5.18). Vi ser at  $B_{0,0}(t)$  er definert for  $0 \leq t < 1$  og at  $B_{1,0}(t)$  er definert for  $1 \leq t < 2$ . Det betyr at funksjonen vi har funnet, er en funksjon satt sammen av  $y = t$  for  $0 \leq t < 1$  og  $y = 2 - t$  for  $1 \leq t < 2$ . Figur 5.11 viser noen lineære B-splines på en uniform skjøtvektor.

#### Øving

Bestem uttrykkene for  $B_{i,1}(t)$ ,  $i = 1, \dots, 4$  på figur 5.11.



Figur 5.12: Kvadratisk B-spline  $B_{0,2}(t)$  med  $t_0 = 0, t_1 = 1, t_2 = 2, t_3 = 3$ .

### 5.5.4 Eksempel, kvadratisk B-spline, d=2

Vi har

$$B_{1,1}(t) = (t-1)B_{1,0}(t) + (3-t)B_{2,0}(t)$$

og kan nå regne ut  $B_{0,2}(t)$  ved hjelp av definisjonen (som en kombinasjon av  $B_{0,1}(t)$  og  $B_{1,1}(t)$ ). Vi får

$$\begin{aligned} B_{0,2}(t) &= w_{0,2}(t) \cdot B_{0,1}(t) + (1 - w_{1,2}(t)) \cdot B_{1,1}(t) \\ &= \frac{t-t_0}{t_2-t_0} B_{0,1}(t) + \left(1 - \frac{t-t_1}{t_3-t_1}\right) B_{1,1}(t) \\ &= \frac{t}{2} B_{0,1}(t) + \frac{3-t}{2} B_{1,1}(t) \\ &= \frac{t}{2} (t \cdot B_{0,0}(t) + (2-t) \cdot B_{1,0}(t)) \\ &\quad + \frac{3-t}{2} ((t-1)B_{1,0}(t) + (3-t)B_{2,0}(t)) \\ &= \frac{t^2}{2} B_{0,0} + \frac{t(2-t)}{2} B_{1,0} + \frac{(3-t)(t-1)}{2} B_{1,0} + \frac{(3-t)^2}{2} B_{2,0} \\ &= \frac{t^2}{2} B_{0,0} + (-t^2 + 3t - \frac{3}{2}) B_{1,0} + \frac{(3-t)^2}{2} B_{2,0} \end{aligned}$$

Hva har vi egentlig her? Når vi bruker definisjonsområdet for  $B_{i,0}$ , ser vi at vi har en andregradsfunksjon på intervallet  $[0, 1)$ , en annen andregradsfunksjon på  $[1, 2)$  og en tredje på  $[2, 3)$ . Se også figur 5.12.

### 5.5.5 Kontinuitet

En kvadratisk B-spline basisfunksjon er  $C^1$  kontinuerlig. Ved å derivere ser vi at denne funksjonen har kontinuerlig derivert, siden den er deriverbar i skjøtene.

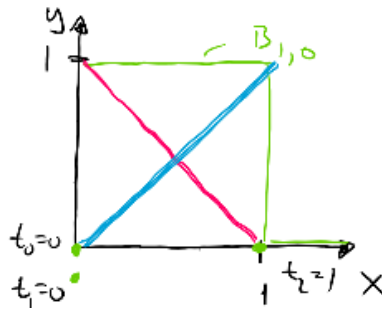
$$\begin{aligned} \frac{d}{dt} \left( \frac{t^2}{2} \right) &= t \\ \frac{d}{dt} \left( -t^2 + 3t - \frac{3}{2} \right) &= -2t + 3 \\ \frac{d}{dt} \left( \frac{(3-t)^2}{2} \right) &= t - 3 \end{aligned}$$

Når vi setter inn  $t=1$  i de to første uttrykkene, får vi at den deriverte er 1. Når vi setter inn  $t=2$  i de to siste uttrykkene, får vi at den deriverte er -1.

### 5.5.6 Multiple skjøter

En skjøt i skjøtvektoren kan gjentas flere ganger, og vi sier at multiplisiteten til en skjøt er antall ganger den forekommer i skjøtvektoren.

Bezierkurver er et spesialtilfelle av B-spline kurver, se avsnitt 7.2 og spesielt figur 5.6. På samme måte er Bernstein basisfunksjonene spesialtilfeller av B-spline basisfunksjoner. Hvordan kan vi uttrykke Bernstein basisfunksjoner med B-spline basisfunksjoner? La oss starte med å lage lineære B-spline basisfunksjoner som på figur 5.4, vist med tilpasninger på figur 5.13.



Figur 5.13:  $B_{0,1}(t)$  og  $B_{1,1}(t)$  med  $t_0 = 0$ ,  $t_1 = 0$ ,  $t_2 = 1$ ,  $t_3 = 1$ .

Vi velger skjøter  $t_0 = 0, t_1 = 0, t_2 = 1, t_3 = 1$ . Hvordan finner vi  $B_{0,0}(t)$ ? Vi ser av definisjonen (5.18) at

$$B_{0,0}(t) = \begin{cases} 1 & t_0 \leq t < t_1 \\ 0 & \text{ellers} \end{cases}$$

Første del (øverste linje) blir umulig å oppfylle når vi har multiple skjøter  $t_0 = t_1$ , derfor blir  $B_{0,0}(t) = 0$ . Videre får vi

$$B_{0,1}(t) = w_{0,1}B_{0,0}(t) + (1 - w_{1,1}(t))B_{1,0}(t) = 0 + (1 - t) \cdot 1 = 1 - t$$

og

$$B_{1,1}(t) = w_{1,1}(t)B_{1,0}(t) + (1 - w_{2,1}(t))B_{2,0}(t) = \frac{t - t_1}{t_2 - t_1} \cdot 1 + 0 = t$$

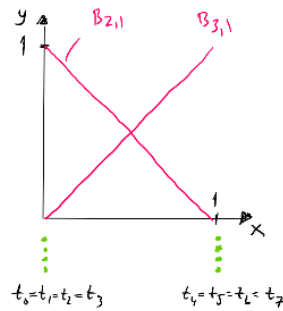
Her har vi utelatt  $B_{0,0}$  og  $B_{1,0}$  faktorene i siste ledd. Siden vi har multiple skjøter, går begge basisfunksjonene over hele intervallet  $[0,1]$ .

$B_{2,0}(t) = 0$  av samme grunn som at  $B_{0,0}(t) = 0$ . Det hender at vi utelater argumentet i notasjonen og bare skriver  $B_{i,d}$  i stedet for  $B_{i,d}(t)$ , som nedenfor.

### Øving

La  $t_0 = t_1 = t_2 = t_3 = 0$  og  $t_4 = t_5 = t_6 = t_7 = 1$ . Bruk rekursjonsformelen (5.18) til å regne ut

1.  $B_{i,0}$ ,  $i = 0, \dots, 7$
2.  $B_{i,1}$ ,  $i = 0, \dots$
3.  $B_{i,2}$ ,  $i = 0, \dots$
4.  $B_{i,3}$ ,  $i = 0, \dots$



Figur 5.14:  $B_{2,1}$  og  $B_{3,1}$  med  $t_0 = t_1 = t_2 = t_3 = 0$ ,  $t_4 = t_5 = t_6 = t_7 = 1$ .

### Lineære B-spline basisfunksjoner

På figur 5.14 er skjøtene markert med prikker, og vi finner ved samme argument som ovenfor at  $B_{0,0} = B_{1,0} = B_{2,0} = 0$  og  $B_{4,0} = B_{5,0} = B_{6,0} = 0$ , slik at kun  $B_{3,0} = 1$ ,  $0 \leq x < 1$ . Dette forplanter seg til de lineære basisfunksjonene. Kun to av dem blir noe annet enn 0:

$$\begin{aligned} B_{2,1}(t) &= w_{2,1}B_{2,0} + (1 - w_{3,1})B_{3,0} = \frac{t-0}{1} \cdot 0 + \left(1 - \frac{t-t_3}{t_5-t_3}\right) \cdot 1 = 1-t \\ B_{3,1}(t) &= w_{3,1} \cdot B_{3,0} + 0 = t \end{aligned}$$

### Kvadratiske B-spline basisfunksjoner

$B_{1,2}(t)$  går over  $t_1, t_2, t_3, t_4$ :

$$B_{1,2}(t) = w_{1,2}B_{1,1} + (1 - w_{2,2})B_{2,1} = 0 + 1 - \frac{t-t_2}{t_4-t_2}(1-t) = (1-t)^2$$

Videre,  $B_{2,2}(t)$  går over  $t_2, t_3, t_4, t_5$ :

$$\begin{aligned} B_{2,2}(t) &= w_{2,2}B_{2,1} + (1 - w_{3,2})B_{3,1} \\ &= \frac{t-0}{1}(1-t) + \left(1 - \frac{t-0}{1}\right)t \\ &= t(1-t) + t(1-t) = 2t(1-t) \end{aligned}$$

og  $B_{3,2}(t)$  går over  $t_3, t_4, t_5, t_6$ :

$$B_{3,2}(t) = w_{3,2}B_{3,1} + (1 - w_{4,2})B_{4,1} = \frac{t-0}{1} \cdot t + 0 = t^2$$

Vi har her tre 2. gradsfunksjoner, definert på hvert sitt intervall. Disse intervallene er bestemt av  $B_{i,0}$ .

### Kubiske B-spline basisfunksjoner

Vi fortsetter med skjøtvektoren på figur 5.14:

$$B_{0,3}(t) = w_{0,3}B_{0,2} + (1 - w_{1,3})B_{1,2} = \frac{t}{1} \cdot 0 + \left(1 - \frac{t}{1}\right)(1-t)^2 = (1-t)^3$$

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

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

$$B_{3,3}(t) = w_{3,3}B_{3,2} + (1 - w_{4,3})B_{4,2} = t \cdot t^2 + 0 = t^3$$

En kubisk B-spline basisfunksjon  $B_{i,3}(t)$  går fra skjøt  $t_i$  til skjøt  $t_{i+3}$ , over intervallet  $[t_i, t_{i+3}]$ . Siden vi har fire like skjøter i hver ende i dette tilfellet, betyr det at alle de fire kubiske B-spline basisfunksjonene som vi har regnet ut, går over samme intervallet  $[0, 1]$ . Vi kan da oppsummere med følgende:

### 5.5.7 Sammenligning

På intervallet  $[0, 1]$  har vi regnet ut tredjegrads basisfunksjoner på tre forskjellige måter:

- Ved kubisk Hermite interpolasjon i avsnitt 5.7.2
- Ved å ekspandere  $(1-t+t)^3$  som på figur 5.6
- Ved å bruke Cox-deBoor rekursjonsformel, (5.18) på skjøtvektor  $\{0, 0, 0, 0, 1, 1, 1, 1\}$  (som er noe annet enn deBoor's algoritme).

Selvsagt var de deriverte i avsnitt 5.7.2 konstruert for dette formålet. Men uansett har vi brukt ulike metoder og kommet til samme resultat, og det er en nyttig erfaring. Vi skal nå gå litt mer detaljert til verks og studere egenskaper til B-spline basisfunksjonene.

### 5.5.8 Affin kombinasjon av kontrollpunkter

B-spline basisfunksjoner summerer seg til 1 og kan brukes til å beregne en affin kombinasjon av kontrollpunkter: Gitt en grad  $d$ ,  $n$  kontrollpunkter og en skjøtvektor  $\mathbf{t} = \{t_0, \dots, t_{n+d}\}$ . Da er

$$f(t) = \sum_{i=0}^{n-1} c_i B_{i,d}(t) \quad (5.19)$$

Det er kun et begrenset antall B-spline basisfunksjoner som er større eller lik 0 for en gitt  $t$ . Når vi har funnet det korrekte skjøtintervallet for den aktuelle  $t$ -verdien,  $t_\mu \leq x < t_{\mu+1}$ , reduseres denne summen til

$$f(t) = \sum_{i=\mu-d}^{\mu} c_i B_{i,d}(t) \quad (5.20)$$

Hvis koeffisientene  $c_i$  i likning (5.19) og likning (5.20) er punkter  $\mathbf{c}_i \in \mathbf{R}^2$  eller  $\mathbf{c}_i \in \mathbf{R}^3$ , representerer uttrykket  $\mathbf{f}(t)$  en vektorfunksjon og definerer en parametrisk kurve.

Vi kan konstruere de lineære B-spline basisfunksjonene ved å bruke definisjonen, og vi kan tegne dem opp ganske enkelt.

## 5.6 Å konstruere en kvadratisk B-spline basisfunksjon

Vi skal se hvordan vi konstruerer en kvadratisk B-spline basisfunksjon. figur 5.15 viser den kvadratiske B-spline basisfunksjonen  $B_{3,2}(t)$  på et utsnitt av en skjøtvektor. Denne aktuelle basisfunksjonen starter i  $t_3$  og ender i  $t_6$ , og  $t_4$  og  $t_5$  er uniformt fordelt i mellom. Mer trenger vi ikke vite om denne skjøtvektoren for å konstruere basisfunksjonen! Det er fordi den har lokal støtte (support).

Den er altså positiv over  $d + 1 = 3$  intervaller. Figuren viser at denne ene basisfunksjonen er satt sammen av tre kvadratiske komponenter (vi så ovenfor at en lineær basisfunksjon er satt sammen av to lineære komponenter). Den første (svart) og den siste (orange) kvadratiske delen er konstruert ved at to lineære funksjoner er multiplisert.

Hvis vi bruker (5.19) finner vi de eksplisitte uttrykkene:

$$\begin{aligned}
 B_{3,2}(t) &= w_{3,2}(t) \cdot B_{3,1}(t) + (1 - w_{4,2}(t))B_{4,1}(t) \\
 &= w_{3,2}(t) [w_{3,1}(t) \cdot B_{3,0}(t) + (1 - w_{4,1}(t)) B_{4,0}(t)] \\
 &+ (1 - w_{4,2}(t)) [w_{4,1}(t) B_{4,0}(t) + (1 - w_{5,1}(t)) B_{5,0}(t)] \\
 &= w_{3,2}(t) w_{3,1}(t) B_{3,0}(t) \\
 &+ w_{3,2}(t)(1 - w_{4,1}(t)) B_{4,0}(t) + (1 - w_{4,2}(t)) w_{4,1}(t) B_{4,0}(t) \\
 &+ (1 - w_{4,2}(t))(1 - w_{5,1}(t)) B_{5,0}(t) \\
 &= \frac{(t - t_3)^2}{(t_5 - t_3)(t_4 - t_3)} B_{3,0}(t) \\
 &+ \left( \frac{t - t_3}{t_5 - t_3} \frac{t_5 - t}{t_5 - t_4} + \frac{t_6 - t}{t_6 - t_4} \frac{t - t_4}{t_5 - t_4} \right) B_{4,0}(t) \\
 &+ \frac{(t - t_6)^2}{(t_6 - t_4)(t_6 - t_5)} B_{5,0}(t)
 \end{aligned}$$

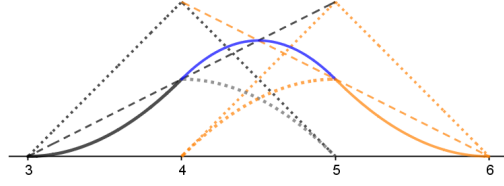
Alternativt, og i henhold til figur 5.15,

$$B_{3,2}(t) = \frac{t-t_3}{t_5-t_3} B_{3,1}(t) + (1 - \frac{t-t_4}{t_6-t_4}) B_{4,1}(t) = \frac{t-t_3}{2} B_{3,1}(t) + \frac{6-t}{2} B_{4,1}(t).$$

Den prikkede svarte linjen representerer den lineære basisfunksjonen  $B_{3,1}(t)$ . Den stiplede svarte linjen representerer  $w_{3,2}(t)$ . Når disse to multipliseres, får vi den første komponenten til  $B_{3,2}(t)$ , den heltrukne svarte kurven for  $t_3 \leq t \leq t_4$ . Vi får også den prikkede svarte kurven for  $t_4 \leq t \leq t_5$ .

Den prikkede orange linjen representerer den lineære basisfunksjonen  $B_{4,1}(t)$ . Den stiplede orange linjen representerer  $1 - w_{4,2}(t)$ . Når disse to multipliseres, får vi den prikkede orange kurven for  $t_4 \leq t \leq t_5$  og den siste komponenten til  $B_{3,2}(t)$ , den heltrukne orange kurven for  $t_5 \leq t \leq t_6$ .

Til slutt: Den andre komponenten til  $B_{3,2}(t)$  lages ved å addere de to prikkede kurvene for  $t_4 \leq t \leq t_5$ . Den er tegnet med en heltrukken blå kurve på figur 5.15. Hvis vi setter inn verdier for en uniform skjøtvektor  $t_i = i$ , får vi



Figur 5.15: Konstruksjon av de tre delene til  $B_{3,2}(t)$  på en uniform skjøtvektor.

$$B_{3,2}(t) = \frac{1}{2}(t-3)^2 B_{3,0}(t) - \frac{1}{2}((t-3)(t-5) + (t-4)(t-6))B_{4,0}(t) + \frac{1}{2}(t-6)^2 B_{5,0}(t)$$

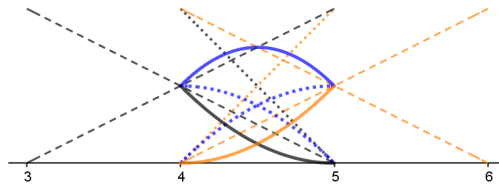
Denne skrivemåten angir at

$$B_{3,2}(t) = \begin{cases} \frac{1}{2}(t-3)^2 & t_3 \leq t < t_4 \\ -\frac{1}{2}((t-3)(t-5) + (t-4)(t-6))B_{4,0}(t) & t_4 \leq t < t_5 \\ \frac{1}{2}(t-6)^2 & t_5 \leq t < t_6 \end{cases}$$

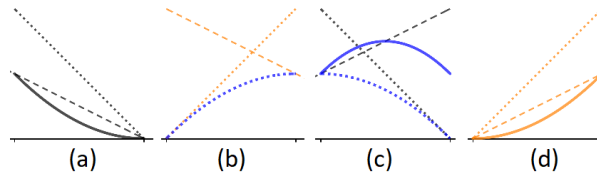
Det generelle uttrykket for en kvadratisk B-spline basisfunksjon blir

$$\begin{aligned} B_{i,2}(t) &= \frac{(t-t_i)^2}{(t_{i+2}-t_i)(t_{i+1}-t_i)} B_{i,0}(t) \\ &- \left( \frac{(t-t_i)(t-t_{i+2})}{(t_{i+2}-t_i)(t_{i+2}-t_{i+1})} + \frac{(t-t_{i+3})(t-t_{i+1})}{(t_{i+3}-t_{i+1})(t_{i+2}-t_{i+1})} \right) B_{i+1,0}(t) \\ &+ \frac{(t-t_{i+3})^2}{(t_{i+3}-t_{i+1})(t_{i+3}-t_{i+2})} B_{i+2,0}(t) \\ &= w_{i,1}w_{i,2}B_{i,0} \\ &+ ((1-w_{i+1,2})w_{i+1,1} + w_{i,2}(1-w_{i+1,1})) B_{i+1,0} \\ &+ (1-w_{i,2})(1-w_{i+1,1})B_{i+2,0} \end{aligned} \quad (5.21)$$

Ligning (5.20) kan brukes til å beregne verdien til en kvadratisk splinefunksjon eller splinekurve på en gitt skjøtvektor. I dette tilfellet viser figur 5.16 bidraget fra hver basisfunksjon i tilfellet  $\mu = 4$ , det vil si  $4 \leq t < 5$  (les bare av funksjonsverdien for en gitt parameterverdi). Den endelige kurven er en sum av tre basisfunksjoner multiplisert med de respektive kontrollpunktene. Ligning (5.20) med  $d = 2$  og  $\mu = 4$  blir  $f(t) = c_2 B_{2,2}(t) + c_3 B_{3,2}(t) + c_4 B_{4,2}(t)$



Figur 5.16: Kvadratiske basisfunksjoner brukt i likning (5.20) på et gitt intervall.  $B_{2,2}(t)$  er den heltrukne svarte kurven,  $B_{3,2}(t)$  er den heltrukne blå kurven og  $B_{4,2}(t)$  er den heltrukne orange kurven.  $B_{3,2}(t)$  er satt sammen av to kvadratiske kurver.



Figur 5.17: Detaljer fra figuren ovenfor. Figuren samsvarer med linjene i (5.21).

### 5.6.1 Beregne verdien for et punkt på en splinekurve

Et punkt på en splinekurve  $f(t)$  kan beregnes ved en affin kombinasjon av kontrollpunkter som i likning (5.20). Figur 5.16 viser hvilke B-spline basisfunksjoner som gir bidrag for  $t_\mu = 4 \leq t < t_{\mu+1} = 5$ . Som nevnt, ligning (5.20) med  $d = 2$  og  $\mu = 4$  blir  $f(t) = c_2 B_{2,2}(t) + c_3 B_{3,2}(t) + c_4 B_{4,2}(t)$ .

Den første B-spline basisfunksjonen  $B_{\mu-2,2}$  starter i  $t_{\mu-2} = 2$  og ender i  $t_{\mu+1} = 5$  (den svarte funksjonen i figur 5.17 a). På intervallet  $t_\mu = 4 \leq t < t_{\mu+1} = 5$  kan de beregnes som produktet av de to svarte prikkede/stiplede linjene,

$$B_{\mu-2,2}(t) = (1 - w_{\mu,1}(t)) \cdot (1 - w_{\mu-1,2}(t)) \quad (5.22)$$

Den midterste B-spline basisfunksjonen  $B_{\mu-1,2}(t)$  starter i  $t_{\mu-1} = 3$  og ender i  $t_{\mu+2} = 6$ . Figur 5.17 b-c viser hvordan den beregnes: Første ledd svarer til (b) på figuren og andre ledd svarer til (c) på figuren.

$$B_{\mu-1,2}(t) = w_{\mu,1}(t) \cdot (1 - w_{\mu,2}(t)) + (1 - w_{\mu,1}(t)) \cdot w_{\mu-1,2}(t) \quad (5.23)$$

Den siste B-spline basisfunksjonen  $B_{\mu,2}$  starter i  $t_\mu = 4$  (den orange funksjonen i figur 5.17 d). Verdien av den er produktet av de to orange prikkede/stiplede linjene,

$$B_{\mu,2}(t) = w_{\mu,1}(t) \cdot w_{\mu,2}(t) \quad (5.24)$$

### 5.6.2 Eksempel

La oss bruke skjøttvektor  $\mathbf{t} = \{0, 0, 0, 1, 1, 1\}$  og grad  $d = 2$  og regne ut b-spline basisfunksjonene for  $t = \frac{2}{3}$ .

Vi har fra tidligere eksempel basisfunksjonene  $B_{0,2}(t) = (1 - t)^2$ ,  $B_{1,2}(t) = 2t(1 - t)$  og  $B_{2,2}(t) = t^2$ . Vi får  $B_{0,2}(\frac{2}{3}) = \frac{1}{9}$  og  $B_{2,2}(\frac{2}{3}) = \frac{4}{9}$ . Siden basisfunksjonene har sum 1, blir  $B_{1,2}(\frac{2}{3}) = \frac{4}{9}$ .

La oss bruke skjøttvektor  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$  og grad  $d = 2$  og regne ut b-spline basisfunksjonene for  $t = \frac{4}{3}$ .

Her må vi først avgjøre hvilke basisfunksjoner som er positive og bidrar.  $B_{0,2}(t)$  har domene  $[t_0, \dots, t_3] = [0, 1]$ . Altså er det  $B_{1,2}(t)$ ,  $B_{2,2}(t)$ ,  $B_{3,2}(t)$ , som er positive.



Fra (5.22) får vi, med  $\mu = 3$

$$\begin{aligned} B_{1,2}(t) &= (1 - w_{3,1}(t)) \cdot (1 - w_{2,2}(t)) \\ &= \frac{t_4 - t}{t_4 - t_3} \cdot \frac{t_4 - t}{t_4 - t_2} = \frac{2 - \frac{4}{3}}{2 - 1} \cdot \frac{2 - \frac{4}{3}}{2 - 0} = \frac{2}{9} \end{aligned}$$

og fra (5.24)

$$\begin{aligned} B_{3,2}(t) &= w_{3,1}(t) \cdot w_{3,2}(t) \\ &= \frac{t - t_3}{t_4 - t_3} \cdot \frac{t - t_3}{t_5 - t_3} = \frac{\frac{4}{3} - 1}{2 - 1} \cdot \frac{\frac{4}{3} - 1}{2 - 1} = \frac{1}{9} \end{aligned}$$

Da blir

$$B_{2,2}(t) = 1 - \frac{2}{9} - \frac{1}{9} = \frac{6}{9}$$

## 5.7 Oppgaver

### 5.7.1

1. Hva er uttrykket for de lineære Bernstein basis funksjonene?
2. Bruk ligning 5.9 til å bestemme Bernstein basis funksjonene av grad 2 i ligning 5.8.
3. Skriv polynomene på standardform (potensform) og sett opp koeffisientmatrisen. Sammenligne med oppgave 5.7.2.

### 5.7.2

Vis at de deriverte til Bernstein basis funksjonene av grad 3 i ligning 5.8 kan skrives

$$\begin{array}{c} -3(1-t)^2 \\ 3(1-t)^2 - 6t(1-t) \\ -3t^2 + 6t(1-t) \\ 3t^2 \end{array}$$

Hint: Bruk produktregelen og kjerneregelen. Sett opp på matriseform.

### 5.7.3

Gitt kontrollpunktene  $P_0(1, 1)$ ,  $P_1(2, 3)$  og  $P_2(3, 0)$  for en kvadratisk Bezierkurve. Bruk Bernstein basisfunksjonene til å regne ut uttrykket for kurven  $\mathbf{p}(t) = (x(t), y(t))$ , sett inn for  $t = 0$ ,  $t = \frac{1}{2}$  og  $t = 1$  og skisser kurven.

### 5.7.4

Gitt kontrollpunktene  $P_0(1, 0)$ ,  $P_1(2, 2)$  og  $P_2(4, 1)$  for en kvadratisk Bezierkurve. Bruk Bernstein basisfunksjonene til å regne ut uttrykket for kurven  $\mathbf{p}(t) = (x(t), y(t))$ , sett inn for  $t = 0$ ,  $t = \frac{1}{2}$  og  $t = 1$  og skisser kurven.

### 5.7.5

Gitt tre kontrollpunkter  $c_0(4, 1)$ ,  $c_1(1, 2)$  og  $c_2(3, 3)$ . Regn ut en kvadratisk parametrisert kurve som interpolerer disse punktene ved å bruke (5.17) (eller matriseformen).

### 5.7.6

Gitt tre kontrollpunkter  $c_0(1, 0)$ ,  $c_1(2, 2)$  og  $c_2(4, 1)$ . Regn ut en kvadratisk parametrisert kurve som interpolerer disse punktene ved å bruke (5.17) (eller matriseformen).

### 5.7.7

La  $t_0 = 0$ ,  $t_1 = \frac{1}{2}$ ,  $t_2 = 1$ ,  $t_3 = \frac{3}{2}$ ,  $t_4 = 2$ . Bestem alle lineære og kvadratiske B-splines.

### 5.7.8

La  $\mathbf{t} = \{0, 0, 0, 1, 1, 1\}$ . Bestem alle lineære og kvadratiske B-splines.

### 5.7.9

La  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$ . Bestem alle lineære og kvadratiske B-splines.

### 5.7.10

La  $\mathbf{t} = \{0, 0, 0, 1, 2, 3, 3, 3\}$ . Bestem alle lineære og kvadratiske B-splines.

### 5.7.11

Gitt

- Lineære B-spline basisfunksjoner på uniform skjøtvektor som i figur 5.11
- $t_i = i$
- B-spline definisjon ligning (5.18)

Bestem alle B-spline basisfunksjoner av grad  $d \leq 3$ .

### 5.7.12

Gitt en skjøtvektor  $\mathbf{t} = \{0, 1, 1, 3\}$ . Bestem alle lineære og kvadratiske B-splines.

### 5.7.13

Deriver de kvadratiske B-spline basisfunksjonene i oppgave 5.7.9.

## Kapittel 6

# Flater og trianguleringer

### 6.1 Introduksjon

I dette kapitlet studerer vi funksjoner av to variable  $f(x, y)$  definert over en flate satt sammen av trekanter, og vi starter med å se på hvordan man kan dele opp et område i planet i trekanter.

Vi trenger flater i mange slags spill og simuleringer. I denne seksjonen skal vi studere flater som er satt sammen av trekanter. Vi har allerede sett eksempel på en slik flate i avsnitt 2.7, da vi skulle implementere en funksjon av to variable. Vi trenger å studere trekanter i litt mer detalj. Vi må kunne lage en topologi for rendring med OpenGL/Vulkan hvor vi bruker en programmerbar grafikkprosessor. Vi trenger å vite hvor mye minne som kreves for å lage en topologi av en mengde trekanter.

Vi må skille mellom parametriske flater som for eksempel en ball og flater som kan uttrykkes som en funksjon  $z = f(x, y)$ . Vi så i avsnitt 2.5 at en funksjon av to variable  $z = f(x, y)$  kan ha partiellderiverte  $\frac{\partial f}{\partial x}$  og  $\frac{\partial f}{\partial y}$ , og at vi kan bestemme et tangentplan til flaten i et punkt  $(x_0, y_0)$  ved

$$z = f(x_0, y_0) + \frac{\partial f}{\partial x}(x_0, y_0)(x - x_0) + \frac{\partial f}{\partial y}(x_0, y_0)(y - y_0)$$

Flatene vi skal studere her er altså funksjoner som er satt sammen av trekanter. De er definert over et område  $\Omega \subset \mathbb{R}^2$ . Dette området er delt opp i trekanter, og denne oppdelingen kalles en triangulering  $\Delta$ . Vi får da stykkevis bilineære funksjoner som er kontinuerlige, men som har noen slags knekker der trekanter møtes, og dermed ikke er deriverbare over alt. Det fins et eget matematisk rom for å betegne alle slike funksjoner: Splinerommet  $\mathbf{S}_0^1(\Delta)$  hvor tallet 1 indikerer at det er snakk om lineære (bilineære) funksjoner og tallet 0 angir kontinuiteten ( $C^0$  kontinuitet betyr kontinuerlig, men ikke deriverbar).

### 6.2 Trianguleringer

En triangulering av et område i planet består av triangler, vertekser (noder) og kanter. En kant er linjen mellom to noder. To kanter kan ikke krysse hverandre

hvis det ikke er en node i skjæringspunktet. En node er da endepunktet for minst to felles kanter, og graden til en node betyr antall kanter som har noden som endepunkt. Det er vanlig å indeksere nodene ( $V_i$ ) og kantene ( $E_i$ ). Trekantene kan enten indeksere ved  $T_i$  eller ved verteksindeksene ( $i, j, k$ ) i positiv omlopsretning. En Delaunay triangulering er en triangulering som er lokalt optimal for alle trekanter. Det betyr at den omskrevne sirkelen til en vilkårlig trekant  $T_i$  ikke skal inneholde noen andre noder  $V_i$ . Vi kommer nærmere inn på dette i ??.

### 6.2.1 Normalvektoren til en flate

Når vi har funnet ligningen for tangentplanet til flaten  $f(x, y)$ , er det enkelt å regne ut en normalvektor til flaten. I tangeringspunktet blir jo det samme som normalen til tangentplanet.

1. La  $P_0 = (x_0, y_0, z_0)$ . Her er  $z_0 = f(x_0, y_0)$ .
2. Velg to punkter  $P_1$  og  $P_2$  og bruk ligningen for tangentplanet til å regne ut verdiene.
3. Regn ut vektorene  $\mathbf{u} = P_1 - P_0$  og  $\mathbf{v} = P_2 - P_0$ .
4. Regn ut vektoren  $\mathbf{n} = \mathbf{u} \times \mathbf{v}$  ved å bruke determinanter:

$$\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{vmatrix} = [u_y v_z - v_y u_z, -(u_x v_z - v_x u_z), u_x v_y - v_x u_y]$$

(6.1)

5. Normaliser normalvektoren slik at den får lengde 1.

#### Eksempel

La  $f(x, y) = 1 - xy$  og  $(x_0, y_0) = (\frac{1}{2}, \frac{1}{4})$ . Vi finner  $z_0 = 1 - \frac{1}{2} \cdot \frac{1}{4} = \frac{7}{8}$ .

De partiellderiverte blir  $\frac{\partial f}{\partial x} = -y$  og  $\frac{\partial f}{\partial y} = -x$ . Innsatt for punktet  $(\frac{1}{2}, \frac{1}{4})$  får vi  $\frac{\partial f}{\partial x}(\frac{1}{2}, \frac{1}{4}) = -\frac{1}{4}$  og  $\frac{\partial f}{\partial y}(\frac{1}{2}, \frac{1}{4}) = -\frac{1}{2}$ .

Vi har altså et punkt  $P_0 = (\frac{1}{2}, \frac{1}{4}, \frac{7}{8})$  hvor vi skal finne tangentplan og normal.

Ligningen for tangentplanet i  $P_0 = (\frac{1}{2}, \frac{1}{4}, \frac{7}{8})$  blir  $z = \frac{7}{8} - \frac{1}{4}(x - \frac{1}{2}) - \frac{1}{2}(y - \frac{1}{4}) = \frac{9}{8} - \frac{1}{4}x - \frac{1}{2}y$ .

Vi skal nå regne ut en normalvektor til flaten.

1. Vi kan velge oss to punkter  $P_1 = (1, 0, z_1)$ , og  $P_2 = (0, 1, z_2)$  og nå skal vi bruke ligningen for tangentplanet til å beregne z-verdiene.  
 $P_1 = (1, 0, z_1) \Rightarrow z_1 = \frac{9}{8} - \frac{1}{4} - 0 = \frac{7}{8}$  og  $P_2 = (0, 1, z_2) \Rightarrow z_2 = \frac{9}{8} - 0 - \frac{1}{2} = \frac{5}{8}$
2. Vektorene  $\mathbf{u} = P_1 - P_0 = (1, 0, \frac{7}{8}) - (\frac{1}{2}, \frac{1}{4}, \frac{7}{8}) = [\frac{1}{2}, -\frac{1}{4}, 0]$   
og  $\mathbf{v} = P_2 - P_0 = (0, 1, \frac{5}{8}) - (\frac{1}{2}, \frac{1}{4}, \frac{7}{8}) = [-\frac{1}{2}, \frac{3}{4}, -\frac{1}{4}]$ .

3. Normalvektoren  $\mathbf{n} = \mathbf{u} \times \mathbf{v}$  blir

$$\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{1}{2} & -\frac{1}{4} & 0 \\ -\frac{1}{2} & \frac{3}{4} & -\frac{1}{4} \end{vmatrix} = \left[ \frac{1}{16} \ \frac{1}{8} \ \frac{1}{4} \right]^T$$

4. Med desimaltall blir  $\mathbf{n} = [0.0625, 0.125, 0.25]$ .

5. Det som gjenstår nå er å normalisere normalvektoren slik at den får lengde 1. Vi finner  $|\mathbf{n}|^2 = \frac{21}{256} \approx 0.0820 \Rightarrow n = 0.286$ .

Enhetsnormalvektoren blir da  $[0.218, 0.436, 0.873]$ .

Kontroller utregningene!

Vi har i eksemplet ovenfor sett hvordan vi kan bruke partiellderiverte til å lage et tangentplan, og deretter hvordan vi lager en normalvektor ved hjelp av tangentplanet.

Vi skal evaluere en kjent funksjon  $f(x, y)$  i en del punkter, og vi vil trenge normalvektor i hvert punkt. Må vi da finne en ny ligning for et nytt tangentplan i hvert punkt? Og hvordan skal vi velge ut punktene i tangentplanet?

La oss studere eksemplet ovenfor litt nærmere. Det som er klart er at uttrykket for de partiellderiverte er  $\frac{\partial f}{\partial x} = -y$  og  $\frac{\partial f}{\partial y} = -x$ . Videre, for hvert punkt  $(x_0, y_0)$  kan vi regne ut verdien til de partiellderiverte ved innsetting. Ovenfor fikk vi  $\frac{\partial f}{\partial x}(\frac{1}{2}, \frac{1}{4}) = -\frac{1}{4}$  og  $\frac{\partial f}{\partial y}(\frac{1}{2}, \frac{1}{4}) = -\frac{1}{2}$ .

Det betyr:

1. I x-retning endrer z-verdien seg med  $-\frac{1}{4}$  når x øker med 1 (og y er konstant).

En vektor i tangentplanet i x-retning er da  $\mathbf{u} = [1, 0, -\frac{1}{4}]$ .

2. I y-retning endrer z-verdien seg med  $-\frac{1}{2}$  når y øker med 1 (og y er konstant).

En vektor i tangentplanet i y-retning er da  $\mathbf{v} = [0, 1, -\frac{1}{2}]$ .

Vi kan da finne en normalvektor av disse to:  $\mathbf{n} = \mathbf{u} \times \mathbf{v}$

$$\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & -\frac{1}{4} \\ 0 & 1 & -\frac{1}{2} \end{vmatrix} = \left[ \frac{1}{4} \ \frac{1}{2} \ 1 \right]^T = 4 \left[ \frac{1}{16} \ \frac{1}{8} \ \frac{1}{4} \right]^T$$

Vi ser at vi her har funnet samme normalvektor som i eksemplet ovenfor, bortsett fra lengden. Når vi normaliserer, får vi samme vektor. Og vi har ikke lenger noe behov for å regne ut ligningen for tangentplanet. Et generelt uttrykk for normalvektoren blir da

$$\mathbf{n} = \left[ 1, 0, \frac{\partial f}{\partial x} \right] \times \left[ 0, 1, \frac{\partial f}{\partial y} \right]$$

og normalisert

$$\frac{\mathbf{n}}{||\mathbf{n}||} = \frac{[n_x, n_y, n_z]}{\sqrt{n_x^2 + n_y^2 + n_z^2}}$$

## 6.2.2 Konstruksjon av 3d flater

Vi skiller mellom det å lage en 3d flate og selve rendringen. Noen har kanskje vært borti enkleste form for en 3d flate - et heightmap, hvor vi strengt tatt kan angi høyden i hvert punkt  $(x,y)$  for et rektangulært område.

### Regulært grid (rektangulært mesh)

En generell metode for å generere en 3d flate er å evaluere en funksjon  $f(x,y)$  i en dobbel for-løkke. Man får da et rektangulært område i  $xy$ -planet (eller  $xz$ -planet) med  $z$ -verdier i tillegg til  $x$ - og  $y$ -verdier. Siden vi skal tegne trekanten, må hvert rektangel (quad) deles i to. Vi har tidligere gått gjennom et eksempel på dette (figur 2.2). Med normaler i vertexene blir dette som i oppgave ??.

### Triangulering basert på irregulære datapunkter

GPU-en trenger normaler. Normalene beregnes ulikt avhengig av hvilken shadermodell som skal benyttes. Det er ikke nødvendig å regne ut tangentplan og normaler i hvert vertex som i eksemplet ovenfor. Vi kan gjøre noen forenklinger og likevel få et akseptabelt resultat.

Ettersom vi jobber tett på GPU med trekanten, skal vi fokusere på å bruke flater sammensatt av trekanten, men ikke nødvendigvis trekanten som ligger ordnet. Slike flater går gjerne under betegnelsen *mesh* når det kommer til implementering. Vi starter med å fokusere på *en* trekant.

### Stykkevis bilineær interpolant

En flate som er satt sammen av trekanten er lineær i  $x$  og  $y$  og kontinuerlig, men generelt ikke deriverbar over kantene og hjørnene. En slik flate kalles en stykkevis bilineær interpolant. Flaten er definert over et område  $\Omega \subset \mathbf{R}^2$  og interpolerer et gitt datasett. Vi har studert kurver, deriverbarhet og kontinuitet og skjønner at det blir mye jobb å lage glatte flater fra bunnen av. Vi skal derfor nøye oss med å bruke trekantbaserte flater. De er såpass viktige at de har en egen betegnelse i matematikken: Splinerommet  $\mathbf{S}_0^1(\Omega)$ .

### Kileprodukt

For vektorer  $\mathbf{u} = [u_1, u_2]$  og  $\mathbf{v} = [v_1, v_2]$  i planet har vi noe som kalles kileproduktet ([7], kapittel 2.2) definert ved

$$\mathbf{u} \wedge \mathbf{v} = u_1 \cdot v_2 - u_2 \cdot v_1$$

Sett  $\mathbf{x}_1 = Q - P$  og  $\mathbf{x}_2 = R - P$  på figur 5.3. Vektorene  $\mathbf{x}_1$  og  $\mathbf{x}_2$  utspenner da et parallelogram med areal  $|\mathbf{x}_1 \wedge \mathbf{x}_2|$ . Alternativt kan vi utvide  $\mathbf{x}_1$  og  $\mathbf{x}_2$  med  $z$ -koordinat lik 0 og regne ut kryssprodukt på vanlig måte,  $\mathbf{x}_1 \times \mathbf{x}_2$ . Kryssproduktet har som vi kanskje husker to geometriske tolkninger: Det er en vektor vinkelrett på både  $\mathbf{x}_1$  og  $\mathbf{x}_2$ , og fortegnet til  $z$ -koordinaten angir retningen (eller hvilken side den er på). Lengden av denne vektoren er arealet av parallelogrammet utspent av  $\mathbf{x}_1$  og  $\mathbf{x}_2$  og selvsagt det dobbelte til arealet av trekanten  $T$ . Siden vi får bruk for både lengde og fortegn når vi skal regne ut barysentriske koordinater, er det best å bruke kileproduktet.

Videre, sett  $\mathbf{u}_1 = Q - X$  og  $\mathbf{u}_2 = R - X$  på figuren. Vektorene  $\mathbf{u}_1$  og  $\mathbf{u}_2$  utspenner da et parallellogram med areal  $\mathbf{u}_1 \wedge \mathbf{u}_2$ . Lengden av denne vektoren er arealet av parallellogrammet utspent av  $\mathbf{u}_1$  og  $\mathbf{u}_2$  og altså det dobbelte til arealet av den røde trekanten. Fortegnet har også her betydning!

Vi gjør det samme med de to siste fargede trekantene.

$\mathbf{v}_1 = R - X$  og  $\mathbf{v}_2 = P - X$ . Den grønne trekantens areal er halvparten av arealet  $\mathbf{v}_1 \wedge \mathbf{v}_2$  (og husk på fortegnet).

$\mathbf{w}_1 = P - X$  og  $\mathbf{w}_2 = Q - X$ . Den blå trekantens areal er halvparten av arealet  $\mathbf{w}_1 \wedge \mathbf{w}_2$  (og husk på fortegnet).

Vi har da følgende formler for utregning av de barysentriske koordinatene:

$$\begin{aligned} u &= \frac{\mathbf{u}_1 \wedge \mathbf{u}_2}{\mathbf{x}_1 \wedge \mathbf{x}_2} \\ v &= \frac{\mathbf{v}_1 \wedge \mathbf{v}_2}{\mathbf{x}_1 \wedge \mathbf{x}_2} \\ w &= \frac{\mathbf{w}_1 \wedge \mathbf{w}_2}{\mathbf{x}_1 \wedge \mathbf{x}_2} \end{aligned} \tag{6.2}$$

±

Fortegnet til kileproduktet, eller retningen på vektoren (dersom man regner ut kryssprodukt i stedet for kileprodukt), forteller noe om plasseringen til et gitt punkt i forhold til trekanten. Det spiller en viktig rolle i anvendelse av barysentriske koordinater. Hva skjer hvis vi for eksempel flytter punktet X på undersiden av linjen gjennom P og Q? Eller ut over en av de to andre kantene til T?

### 6.2.3 Implementering

Gitt en funksjon  $z = f(x, y)$  eller en stykkevis bilinear funksjon, altså en samling trekanter - som vi er vant til nå. Vi kan regne ut de barysentriske koordinatene med 3-dimensjonale vektorer. Men det gir unødvendig mange operasjoner (multiplikasjoner/divisjoner - kostbar utregning) sammenlignet med å regne ut de barysentriske koordinatene til punkter i planet. Projeksjonen av en tredimensjonal triangulær flate i planet danner en todimensjonal triangulær flate og de barysentriske koordinatene kan regnes ut på denne, det blir samme resultat.

Funksjonen nedenfor ligger i en klasse **Vec2** med overlastede operatorer for subtraksjon og kryssprodukt/kileprodukt (hattetegnet brukes vanligvis til kryssprodukt operator), og en funksjon *length()* som beregner lengden. Funksjonen har tre parametre inn: (x,y) koordinatene til den aktuelle trekanten, og kalles opp av punktet som vi skal ha de barysentriske koordinatene til. De barysentriske koordinatene returneres i en Vec3.

```
Vec3 barycentricCoordinates(const Vec2& p1, const Vec2& p2, const Vec2& p3)
{
```



```

Vec2 p12 = p2-p1;
Vec2 p13 = p3-p1;

Vec3 n = p12^p13;
float areal_123 = n.length(); // dobbelt areal

Vec3 baryc; // til retur. Husk
// u
Vec2 p = p2 - *this;
Vec2 q = p3 - *this;
n = p^q;
baryc.x = n.z/areal_123;
// v
p = p3 - *this;
q = p1 - *this;
n = p^q;
baryc.y = n.z/areal_123;
// w
p = p1 - *this;
q = p2 - *this;
n = p^q;
baryc.z = n.z/areal_123;

return baryc;
}

```

#### 6.2.4 Anvendelser

Barysentriske koordinater har mange praktiske anvendelser. Blant annet kan en fragmentshader bruke barysentriske koordinater til å interpolere farger i vertices.

##### Beregne høyde $z = f(x, y)$

Anta at vi har et spillerobjekt eller et annet interaktivt objekt som skal bevege seg på et tredimensjonalt underlag. Vi kan styre spilleren i xy-planet for eksempel med tastene wasd. Men hvordan beregne riktig høyde? Vi kan bruke barysentriske koordinater til å beregne høyden. Når vi har funnet ut hvilken trekant i xy-planet spilleren er i, kan vi beregne høyden som en kombinasjon av høyden i hvert vertex. Terrenget vårt er jo generert ved å regne ut høyden i punkter med jevn avstand og deretter generere en regulær triangulering. Hvis spilleren har posisjon (x,y) og er i trekant PQR, regner vi ut de barysentriske koordinatene u, v, w med hensyn på (x,y) og regner ut høyden

$$f(x, y) = u \cdot f(P_x, P_y) + v \cdot f(Q_x, Q_y) + w \cdot f(R_x, R_y)$$

Men hvordan finne igjen riktig trekant? I TriangleSurface klassen ligger vertex-data sekvensielt i en vektor. Gitt spillerens (x,y) koordinater kan vi da søke gjennom alle vertexer helt til vi finner en trekant hvor de barysentriske koordinatene  $u, v, w \in [0, 1]$ . Dette er  $O(n)$  søk, men det fungerer, også for irregulære trianguleringer.

```

42
0.0 0.0 2.0    1.5 0.0 2.0    0.0 1.0 2.0
0.0 1.0 2.0    1.5 0.0 2.0    2.5 1.0 1.0
0.0 1.0 2.0    2.5 1.0 1.0    2.0 2.0 2.0
0.0 1.0 2.0    2.0 2.0 2.0    1.0 3.0 1.0
0.0 1.0 2.0    1.0 3.0 1.0    0.0 4.0 2.0
1.5 0.0 2.0    4.0 0.5 2.0    2.5 1.0 1.0
1.5 0.0 2.0    4.0 0.0 2.0    4.0 0.5 2.0
3.5 2.5 1.0    4.0 0.5 2.0    4.0 4.0 2.0
2.0 2.0 2.0    2.5 1.0 1.0    4.0 0.5 2.0
2.0 2.0 2.0    4.0 0.5 2.0    3.5 2.5 1.0
2.5 4.0 2.0    3.5 2.5 1.0    4.0 4.0 2.0
2.0 2.0 2.0    3.5 2.5 1.0    2.5 4.0 2.0
1.0 3.0 1.0    2.0 2.0 2.0    2.5 4.0 2.0
0.0 4.0 2.0    1.0 3.0 1.0    2.5 4.0 2.0

```

Figur 6.1: Datasett for triangulering, med linjevis opplisting av verteks koordinater for hver trekant.

### 6.2.5 Søking

Barysentriske koordinater kan også brukes til søking. La oss se på et eksempel. Nedenfor er en fil med vertex data for 12 punkter gjengitt, riktignok uten farge/normal og teksturkoordinater. Dette er altså rene posisjonsdata. Hver linje representerer en trekant (og det går fint an å lese inn filen slik, selv om vi vanligvis ville legge ett vertex på hver linje. Det er tilsammen 14 trekanter. Uten noen form for topologi eller indeksering må vi duplisere disse punktene for å tegne en flate med `glDrawArrays()`, og vi får da  $14 \cdot 3 = 42$  vertices. Vi trenger i tillegg som nevnt farge/normal og teksturkoordinater.

#### Lineært søk

Selv med denne enkle (eller kanskje heller tungvinte) representasjonen av en triangulær flate, har barysentriske koordinater interessante anvendelser. Hvis vi skal finne igjen hvilket triangel  $T_i$  et gitt punkt  $P$  tilhører, kan vi gjennomløpe arrayen sekvensielt, lese inn tre punkter om gangen og regne ut de barysentriske koordinatene til  $P$  med respekt på  $T_i$ . Hvis alle  $u$ ,  $v$ ,  $w$  er positive, har vi funnet det riktige triangellet. Hvis ikke, fortsetter søket.

Her blir søketiden  $O(n)$  og vi bruker unødige mye lagringsplass. Men det virker, og noen ganger trenger vi også `glDrawArrays()`.

### 6.2.6 Triangulering med indeksering

Gitt  $n$  punkter trenger vi bare å lagre  $n$  forekomster av vertex data. For å kunne definere trekanter og rendre noe, må vi da i tillegg angi en topologi/indeksering og bruke `glDrawElements()`. For eksemplet i 6.2.5 ovenfor blir da geometrien 12 linjer med vertex data:

```

12
0.0 0.0 2.0
0.0 1.0 2.0
0.0 4.0 2.0
1.0 3.0 1.0
1.5 0.0 2.0
2.0 2.0 2.0
2.5 1.0 1.0
2.5 4.0 2.0
3.5 2.5 1.0
4.0 0.0 2.0
4.0 0.5 2.0
4.0 4.0 2.0

```

og topologien 14 linjer med indekser:

```

14
0 4 1
1 4 6
1 6 5
1 5 3
1 3 2
4 10 6
4 9 10
8 10 11
5 6 10
5 10 8
7 8 11
5 8 7
3 5 7
2 3 7

```

Sammenlign med figur 6.2. Merk at indekseringen starter på 0 og at alle vertices er angitt i positiv omløpsretning. Konsekvens er viktig her!

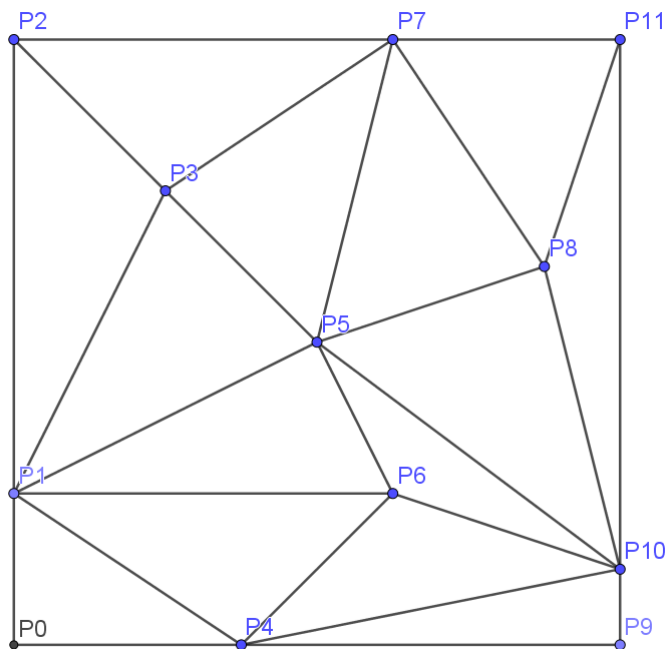
Vi oppnår her et mindre datasett, noe som er generelt er ønskelig. Men søking etter et triangel som inneholder et gitt punkt,  $P \in T_i$  må fortsatt foregå lineært selv om vi bruker topologien. Søketiden er dermed fortsatt  $O(n)$ .

### 6.2.7 Triangulering med naboinformasjon

Vi kan utvide indekseringen ovenfor til å inkludere naboinformasjon. Vi har her

- Geometri lagret unikt og separat.
- Indeksering av trekanter. Hver trekant har implisitt en egen indeks/nummer som starter på 0 og svarer til posisjon i array/vektor.
- Vi tilføyer informasjon om nabotrekanten på hver linje.

En trekant har like mange naboer som kanter, bortsett fra trekanter langs randen til det triangulerte området. To trekanter er **naboer** hvis de har en felles



Figur 6.2: En triangulering for datasett 6.2.5.

kant. Vi skal se på et eksempel hvordan naboene kan angis for hver trekant.

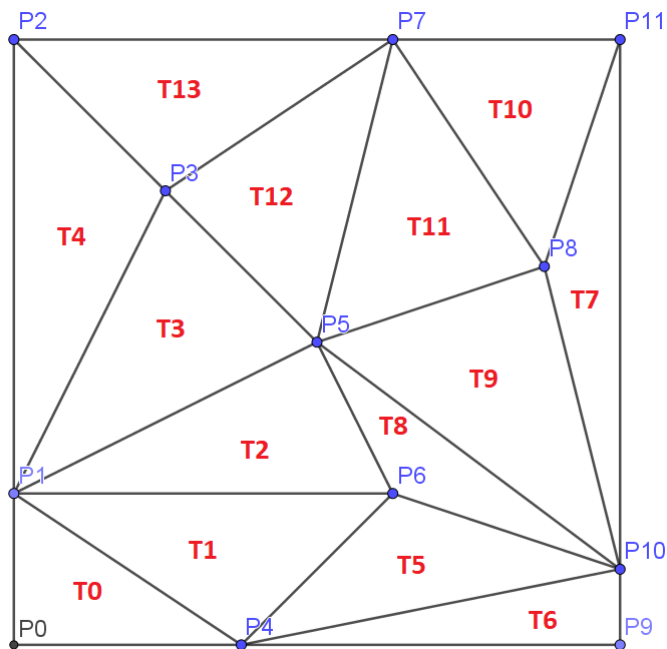
Med samme datasettet som i 6.2.5 kan vi angi en komplett triangulering med naboinformasjon på følgende vis:

```

14
0 4 1      1 -1 -1
1 4 6      5 2 0
1 6 5      8 3 1
1 5 3      12 4 2
1 3 2      13 -1 3
4 10 6     8 1 6
4 9 10     -1 5 -1
8 10 11    -1 10 9
5 6 10     5 9 2
5 10 8     7 11 8
7 8 11     7 -1 11
5 8 7      10 12 9
3 5 7      11 13 3
2 3 7      12 -1 4

```

Vi ser at dersom en trekant ikke har noen nabo over en gitt kant, angis dette med -1. Sammenlign med figur 6.3



Figur 6.3: En triangulering med naboinformasjon.

### 6.2.8 Topologisk søk algoritme

Vi kan nå utnytte naboinformasjonen for hvert triangel til en søkealgoritme. Problemet er som nevnt tidligere:

Gitt et punkt  $P$  og en triangulering

$$\Delta = \{T_i\}_{i=0}^{n-1}$$

1. Velg en initiell trekant og beregn barysentriske koordinater til  $P$  med hensyn på denne.
2. Hvis  $u, v, w \geq 0$  er trekanten funnet.
3. Ellers fortsett søket på den trekanten som svarer til den minste (mest negative) barysentriske koordinaten

Dette kan lett omskrives til en løkke, men krever en grundigere forklaring. Igjen bruker vi eksempeldata som ovenfor.

### 6.2.9 Eksempel

Med naboinformasjon får vi følgende  $O(\log_3 n)$  søkealgoritme for å finne et punkt med koordinater  $P = (x^*, y^*)$ :

- La  $P = (x^*, y^*)$
- Start søket i vilkårlig trekant  $i$  (bruk gjerne forrige verdi).

- **do** {  
   Beregn barysentriske koordinater for trekant i  
   **if**  $u, v, w \geq 0$  funnet = true;  
   **else** i = nabo som svarer til minste barysentriske koordinaten  
   **}** **while** (!funnet)

Sett  $P = (3.5, 3.5)$  og test algoritmen på datasettet ovenfor!

## Kapittel 7

# Parametriske kurver

### 7.1 Introduksjon

Det fins mange kjente uttrykk for parametriske kurver, for eksempel  $x(t) = \cos t, y(t) = \sin t$  eller  $x(t) = \cos t, y(t) = \sin t, z(t) = \frac{t}{10}$  som nevnt i avsnitt 2.4. Vi skal her studere hvordan vi konstruerer parametriske kurver uten noen underliggende funksjon. Vi starter med Bezierkurver.

### 7.2 Bezier kurver og deCasteljau algoritmen

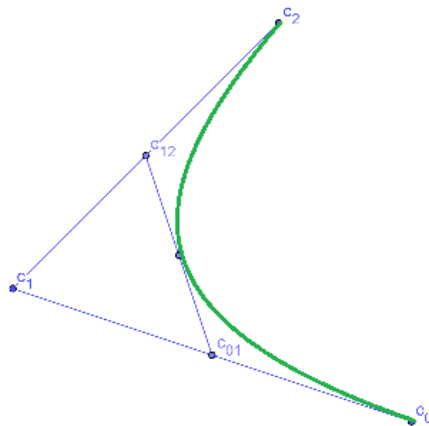
Bezierkurver kjennetegnes ved en grad  $d$ , et sett kontrollpunkter  $\mathbf{c} = \{\mathbf{c}_i\}$  som oftest har to eller tre koordinater, og et kontrollpolygon som består av rette linjer mellom kontrollpunktene. En bezierkurve ligger i et affint rom. Vi kan bruke matriseformen av Bezierpolynomer til beregninger, men det er mer vanlig å benytte deCasteljau algoritmen. Vi skal vise et eksempel for en kvadratisk Bezierkurver, og deretter et eksempel for en kubisk Bezierkurve.

#### Eksempel, kvadratisk Bezierkurve

I dette eksemplet har vi kontrollpunktene  $c_0, c_1$  og  $c_2$  som vist på figur 7.1. På figuren er kontrollpolygonet inntegnet. Vi vet nå at

$$\frac{c_0 + c_1}{2} \text{ og } \frac{c_1 + c_2}{2}$$

er tillatte affinkombinasjoner av punkter. Vi finner  $c_{01} = \frac{c_0 + c_1}{2}$  og  $c_{12} = \frac{c_1 + c_2}{2}$ , trekker en ny linje mellom disse punktene og finner igjen midtpunktet på linjen (affinkombinasjonen  $\frac{c_{01} + c_{12}}{2}$ ). Dette punktet er et punkt på den kvadratiske Bezierkurven definert av  $c_0, c_1$  og  $c_2$ . En Bezierkurve interpolerer både endepunktene og de deriverte i endepunktene, og til sammen kan vi ut fra dette lage en brukbar skisse av kurven. Dette kan vi gjøre helt uten å bruke koordinater, vi kaller det koordinatfri geometri. I eksemplet i avsnitt 5.3.2 har vi regnet ut uttrykket for denne kurven ved å bruke koordinater og Bernstein basisfunksjoner, hvilket var betraktelig mer arbeid.



Figur 7.1: Illustrasjon av deCasteljau algoritmen for grad  $d=2$  og  $t = \frac{1}{2}$ .

### Eksempel, kubisk Bezierkurve

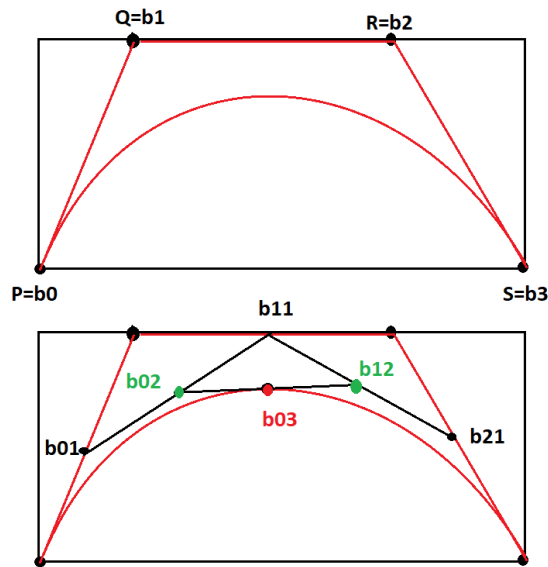
Figur 7.2 viser hvordan algoritmen fungerer for en kubisk Bezierkurve. Øverst på figuren har vi en kurve med fire kontrollpunkter PQRS (som like gjerne kunne vært en funksjon). Den nederste figuren illustrerer hvordan deCasteljau algoritmen fungerer for en Bezierkurve av grad 3, med 4 kontrollpunkter.

Vi skal nå finne punktet som svarer til parameterverdien  $t = \frac{1}{2}$  i en rekursiv prosess.

1. Først må **kontrollpunktene** tegnes opp. Endepunktene her svarer til endepunktene i kubisk Hermite interpolasjon. De to siste kontrollpunktene blir ikke interpolert, men brukes til å styre tangenten i endepunktene.
2. Deretter tegner vi opp **kontrollpolygonet** som består av rette linjer mellom kontrollpunktene.
3. For hvert sett av kontrollpunkter har vi en rett linje som kan uttrykkes ved en affinkombinasjon av punktene. Hvis vi er ute etter et punkt på linjen mellom P og Q velger vi  $t$  og finner en affin kombinasjon  $tP + (1-t)Q$ . Midtpunktet på linjen svarer som kjent til  $t = \frac{1}{2}$ .
4. Dette gjentas for to de gjenstående delene av kontrollpolygonet. Vi får tre nye punkter  $\mathbf{b}_0^1$ ,  $\mathbf{b}_1^1$ ,  $\mathbf{b}_2^1$ .
5. Så gjør vi samme prosessen med de tre punktene  $\mathbf{b}_0^1$ ,  $\mathbf{b}_1^1$ ,  $\mathbf{b}_2^1$  og får to nye punkter  $\mathbf{b}_0^2$ ,  $\mathbf{b}_1^2$ .
6. Til slutt danner vi  $\mathbf{b}_0^3$  av  $\mathbf{b}_0^2$ ,  $\mathbf{b}_1^2$ .

Indekseringen her gjenspeiler rekkefølgen av kontrollpunktene og hvor mange ganger vi har gjentatt prosessen.





Figur 7.2: Illustrasjon av deCasteljau algoritmen for grad  $d=3$  og  $t = \frac{1}{2}$ .

### 7.2.1 Egenskaper ved Bezierkurver og deCasteljau algoritmen

deCasteljau algoritmen kalles en hjørnekuttignsalgoritme, og det er lett å forstå når man studerer disse to eksemplene. En Bezierkurve kan altså konstrueres på minst to forskjellige måter: Enten som en affinkombinasjon av Bernstein basisfunksjoner og kontrollpunkter, eller ved deCasteljau algoritmen (koordinatfri geometri). En Bezierkurve har altså en grad  $d$ ,  $d+1$  kontrollpunkter som danner et kontrollpolygon,  $d+1$  Bernstein basisfunksjoner av grad  $d$  som summerer seg til 1. I tillegg danner kontrollpolygonet en konveks innhylning for kurven.

### 7.2.2 C++ implementering av de Casteljau algoritmen

Gitt en egen C++ klasse **BezierCurve** med grad  $d$  og en objektvariabel **Vector3D** `c[4]`; for å lagre kontrollpunktene. Anta at kontrollpunktene inneholder noen ekte verdier. deCasteljau algoritmen blir da som nedenfor.

Listing 7.1: deCasteljau algoritmen

```
Vector3D BezierCurve::evaluateBezier(float t)
{
    Vector3D a[4]; // 4=d+1 for kubisk Bezier
    for (int i=0; i<4; i++)
        a[i] = c[i];

    for (int k=d; k>0; k--) //for (int k=1; k<=d; k++)
    {
        for (int i=0; i<k; i++) //for (int i=0; i<=d-k; i++)
            a[i] = a[i] * (1-t) + a[i+1] * t;
    }
    return a[0];
}
```

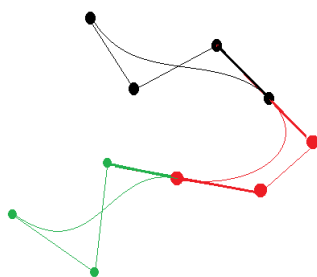
Sammenlign denne algoritmen og spesielt  $k$ -ene med figur 7.2.

### 7.2.3 Sammensatte Bezierkurver

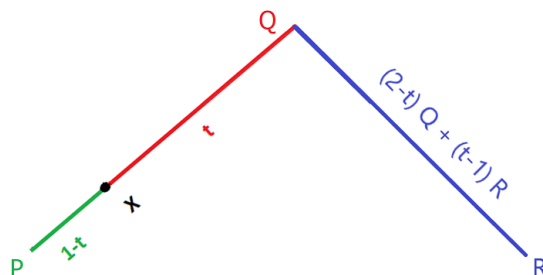
Hva gjør vi når vi har flere enn fire kontrollpunkter, som vi vanligvis har? Vi må sette sammen Bezierkurver på en spesiell måte for å få kontinuerlig derivert i skjøtene. Vi må velge kontrollpunktene slik at tre og tre punkter ligger på en rett linje. Figur 7.3 illustrerer dette. De svarte kontrollpunktene tilhører en kubisk Bezierkurve,  $B_s$ . Denne skjøter vi sammen med den røde  $B_r$  på en slik måte at to svarte kontrollpunkter og det ene røde ligger på en rett linje. Det samme gjør vi med den røde og den grønne kurven  $B_g$ . Vi trenger altså 3 kubiske Bezierkurver til 10 kontrollpunkter: Hver kurve har 4 kontrollpunkter, og kurvene har parvis ett felles kontrollpunkt.

Det er viktig å være klar over at vi nå ikke kan endre på en av kurvene (for eksempel flytte ett kontrollpunkt og samtidig opprettholde  $C^1$  kontinuitet i skjøtene).

Det er mer vanlig å bruke B-splines til slike oppgaver.



Figur 7.3: Tre kubiske Bezierkurver satt sammen med kontinuerlig derivert i skjøtene.



Figur 7.4: Neville's algoritme, ide.

### 7.3 Neville's algoritme

Vi skal nå utvide figur 5.1 med et punkt og en linje, denne gangen uten å ha fokus på trekanter. Framstillingen her er delvis hentet fra [5]. Vi har sett at vi kan kombinere punktene P og Q til en linje mellom P og Q ved å la parameter  $t$  gå fra 0 til 1. Tilsvarende kan vi kombinere Q og R ved å la parameter  $u$  gå fra 0 til 1, eller  $t$  gå fra 1 til 2 når  $t = 1 + u$ . Vi kan altså finne et punkt  $X$  mellom P og Q og et punkt  $Y$  mellom Q og R på samme måte,

$$\begin{aligned} X &= (1-t)P + tQ, \quad 0 \leq t \leq 1 \\ &\text{og} \\ Y &= (2-t)Q + (t-1)R, \quad 1 \leq t \leq 2 \end{aligned} \quad (7.1)$$

Vi ser at  $t=0$  i første linje gir punktet P,  $t=1$  i både første og andre linje gir punktet Q, og  $t=2$  i andre linje gir punktet R. Legg merke til at summen av vektene for hvert punkt er 1 i begge tilfeller.

La oss nå gjenta dette, altså lage en affin kombinasjon (eller en barysentrisk kombinasjon, eller en konveks kombinasjon) av punkt X og Y. Siden X ligger på linjen PQ og Y ligger på linjen QR, blir dette som en kombinasjon av linjene PQ og QR. Vi får da

$$\begin{aligned} P_{PQR}(t) &= X \cdot \frac{2-t}{2} + Y \cdot \frac{t-1}{2} \\ &= ((1-t)P + tQ) \frac{2-t}{2} + ((2-t)Q + (t-1)R) \cdot \frac{t-1}{2} \\ &= \frac{1}{2}(1-t)(2-t)P + t(2-t)Q + \frac{1}{2}t(t-1)R \end{aligned} \quad (7.2)$$

hvor  $0 \leq t \leq 2$ . Vi må dividere på 2 her for at vektene for X og Y skal summeres til 1.

Hvis vi absolutt vil regne ut de kvadratiske uttrykkene, får vi

$$P_{PQR} = \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right)P + (2t - t^2)Q + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right)R = p(t) \cdot P + q(t) \cdot Q + r(t) \cdot R \quad (7.3)$$

De tre andregradspolynomene her kalles Lagrangefunksjonene av grad 2 og danner en basis for  $\mathbf{P}^2$  - rommet av polynomer av grad mindre eller lik 2. Se også avsnitt 5.4. Vi kan sette inn  $t = 0, 1, 2$  i dette uttrykket eller ligning (7.2) og vi får da

	$p(t)$	$q(t)$	$r(t)$
$t = 0$	1	0	0
$t = 1$	0	1	0
$t = 2$	0	0	1
punkt	$P$	$Q$	$R$

Dette viser at  $P_{PQR}(t)$  interpolerer alle punktene P, Q og R. Hvis vi legger til enda et punkt og gjentar prosessen kan vi konstruere en kubisk kurve som interpolerer fire punkter.

Punktene P, Q, R ovenfor kalles **kontrollpunkter**, siden man kan kontrollere kurven ved valg og variasjon av disse punktene. For en kubisk kurve har vi fire kontrollpunkter. Her trenger vi ikke tenke på å finne en funksjon som interpolerer punktene, algoritmen tar seg av den saken.

### 7.3.1 Eksempel

Vi ser på et eksempel med kontrollpunktene  $P_0(1, 3)$ ,  $P_1(1, 1)$  og  $P_2(3, 1)$ . Vi skal konstruere en kvadratisk parametrisk kurve med Neville's algoritme. Igjen, vi kan tenke på  $p(t)$  som en parametrisk kurve eller en vektorfunksjon.

Vi har her noe som minner om en skjøtvektor,  $\mathbf{t} = \{t_0, t_1, t_2\} = \{0, 1, 2\}$  hvor  $t_i$ -ene er parameterverdier i kontrollpunktene. Vi skal ta affinkombinasjoner på hvert intervall. Vi bruker (7.1)

$$P_{01} = (1 - t)P_0 + tP_1, \quad 0 \leq t \leq 1 \quad \text{og} \quad P_{12} = (2 - t)P_1 + (t - 1)P_2, \quad 1 \leq t \leq 2$$

og gjenta dette ett steg til. I siste steg blir vektene som forklart i (7.2).

Vi tegner først opp punktene og kontrollpolygonet. Så regner vi ut punkter på kurven for parameterverdiene  $t = \frac{1}{2}$  og  $t = \frac{3}{2}$  (husk at  $t$  går fra 0 til 2 her). Vi starter med å regne ut et punkt på kurven for parameterverdien  $t = \frac{1}{2}$ : Det blir et punkt mellom  $P_0$  og  $P_1$ . Siden vi skal ha affinkombinasjoner med sum lik 1, får vi  $1 - t = -\frac{1}{2}$  i utregningen av  $P_{12}$ :

$$\begin{aligned} P_{01} &= \frac{1}{2} \cdot P_0 + \frac{1}{2} \cdot P_1 = \frac{1}{2} \cdot (1, 3) + \frac{1}{2} \cdot (1, 1) = (1, 2) \\ P_{12} &= \frac{3}{2} \cdot P_1 - \frac{1}{2} \cdot P_2 = \frac{3}{2} \cdot (1, 1) - \frac{1}{2} \cdot (3, 1) = (0, 1) \\ P_{012} &= \frac{3}{4} \cdot P_{01} + \frac{1}{4} \cdot P_{12} = \frac{3}{4} \cdot (1, 2) + \frac{1}{4} \cdot (0, 1) = \left(\frac{3}{4}, \frac{7}{4}\right) \end{aligned}$$

$P_{01}$  er både en affinkombinasjon og en konvekskombinasjon av  $P_0$  og  $P_1$  og ligger derfor på linjen mellom  $P_0$  og  $P_1$ .

$P_{12}$  er en affinkombinasjon av  $P_1$  og  $P_2$  og ligger derfor på linjen gjennom  $P_1$  og  $P_2$ . Men fordi det ikke er en konvekskombinasjon av  $P_1$  og  $P_2$ , ligger  $P_{12}$  ikke mellom  $P_1$  og  $P_2$ . Når vi i siste linje kombinerer  $P_{01}$  og  $P_{12}$ , får vi et punkt på

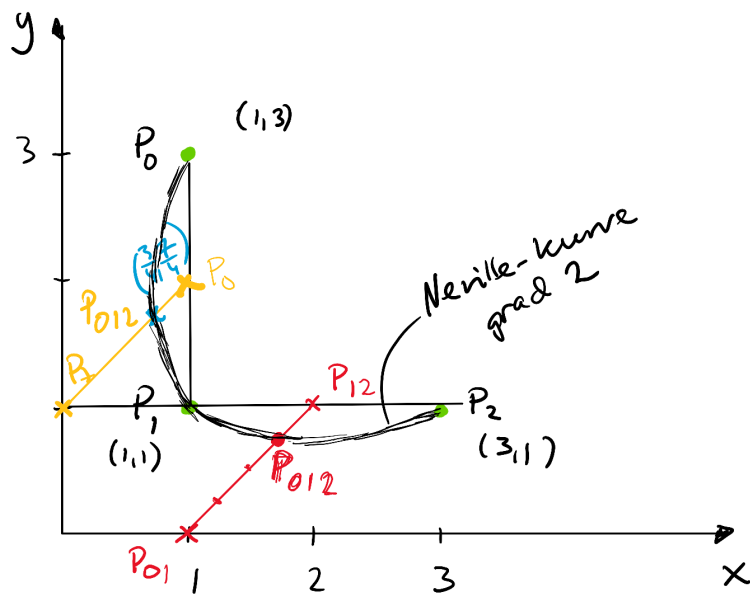
kurven,  $P_{012}$  med gul/orange skrift.

Vi fortsetter med å regne ut et punkt på kurven for parameterverdien  $t = \frac{3}{2}$ :  
Det blir et punkt mellom  $P_1$  og  $P_2$ :

$$\begin{aligned} P_{01} &= -\frac{1}{2} \cdot P_0 + \frac{3}{2} \cdot P_1 = -\frac{1}{2} \cdot (1, 3) + \frac{3}{2} \cdot (1, 1) = (1, 0) \\ P_{12} &= \frac{1}{2} \cdot P_1 + \frac{1}{2} \cdot P_2 = \frac{1}{2} \cdot (1, 1) + \frac{3}{2} \cdot (3, 1) = (2, 1) \\ P_{012} &= \frac{1}{4} \cdot P_{01} + \frac{3}{4} \cdot P_{12} = \frac{1}{4} \cdot (1, 0) + \frac{3}{4} \cdot (2, 1) = \left(\frac{7}{4}, \frac{3}{4}\right) \end{aligned}$$

$P_{01}$  er en affinkombinasjon av  $P_0$  og  $P_1$  og ligger derfor på linjen gjennom  $P_0$  og  $P_1$ . Men fordi det ikke er en konvekskombinasjon av  $P_0$  og  $P_1$ , ligger  $P_{01}$  ikke mellom  $P_0$  og  $P_1$ .

$P_{12}$  er både en affinkombinasjon og en konvekskombinasjon av  $P_1$  og  $P_2$  og ligger derfor på linjen mellom  $P_1$  og  $P_2$ . Når vi i siste linje kombinerer disse to,  $P_{01}$  og  $P_{12}$ , får vi et punkt på kurven,  $P_{012}$  med rød skrift.



Figur 7.5: Eksempel på kontrollpunkter  $(1, 3)$ ,  $(1, 1)$ , og  $(3, 1)$  og interpolerende kurve.

Implementering av Neville's algoritme kan gjøres som nedenfor:

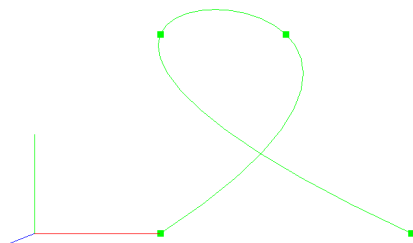
Listing 7.2: Neville's algoritme for kubisk polynom

```
// Neville's algoritme for grad d=3,
// interpolasjon i parameterverdier 0, 1, 2, 3
//      P0123
//      P012 - P123
//      P01 - P12 - P13
//      P0 - P1 - P2 - P3
Vec3 NevilleCurve::nevilleSimple(double t)
{
    // mP0123 initieres med kontrollpunktene
    std::array<Vec3,4> P=mP0123;

    // Hvilket nivå i pyramiden er vi paa
    int level=3;

    for (int i=level; i>=0; i--)
    {
        // Hvilken grad er det paa polynomene vi regner ut
        int d = level-i;

        // Vi kombinerer to og to punkter, start til venstre
        // og legger resultatet tilbake i arrayen
        for (int j=0; j<i; j++)
            P[j] = P[j]*(mt[j+1+d]-t)/(d+1) + P[j+1]*(t-mt[j])/(d+1);
    }
    // Den aktuelle verdien p(t) ligger naa i P[0]
    return P[0];
}
Vec3 NevilleCurve::neville(double t) // Normalisert
{
    std::array<Vec3,4> P=mP0123;
    int level=3;
    for (int i=level; i>=0; i--)
    {
        int d = level-i;
        for (int j=0; j<i; j++)
            P[j] = P[j]*(mt[j+1+d]-t)/(mt[j+1+d]-mt[j])
                + P[j+1]*(t-mt[j])/(mt[j+1+d]-mt[j]);
    }
    return P[0];
}
```



Figur 7.6: Eksempel på kontrollpunkter  $(1,0)$ ,  $(2,2)$ ,  $(1,2)$  og  $(3,0)$  og interpolerende kurve.

## 7.4 Spline kurve

Anta at  $c_i$ -ene i likning (5.19) er punkter  $\mathbf{c}_i \in \mathbf{R}^2$  eller  $\mathbf{c}_i \in \mathbf{R}^3$ . Funksjonen  $\mathbf{f}(t)$  har da to eller tre koordinater for hver parameterverdi  $t$ . Den kalles en vektorfunksjon og definerer en parametrisk kurve. En spline kurve er både en generalisering fra spline funksjon, og en generalisering fra Bezier kurve. Hvis vi har å gjøre med punkter i  $\mathbf{R}^2$ , blir  $\mathbf{f}(t) = (x(t), y(t))$ . Splinekurven er altså en parametrisering, eller en vektorfunksjon.

En spline kurve er definert ved antall kontrollpunkter  $n$ , graden  $d$  og en skjøtvektor  $\mathbf{t} = \{t_0, t_1, \dots, t_{n+d}\}$

Vi kan finne uttrykket for en kvadratisk Bezierkurve ved å bruke de kvadratiske Bernsteinbasisfunksjonene (likning (5.7)). Vi kan finne uttrykket for en kvadratisk Nevillekurve ved å bruke de kvadratiske Lagrange basisfunksjonene (likning (7.3)).

Vi skal se på et eksempel med tre kontrollpunkter i planet, P(1,1), Q(3,3) og R(4,1). En B-spline kurve med disse tre kontrollpunktene går fra P til R og interpolerer altså P og R. Kontrollpolygonet består av den rette linjen fra P til Q og den rette linjen fra Q til R. Kurven ligger i den konvekse innhylningen (Convex Hull) til kontrollpolygonet som her er trekanten PQR.

Vi har  $n = 3$  kontrollpunkter, og trenger  $n = 3$  kvadratiske basisfunksjoner. Gitt en grad  $d$  går hver basisfunksjon over  $d + 2$  skjøter. Vi trenger  $n + d + 1$  skjøter totalt.

En splinekurve som er festet i (interpolerer) første og siste kontrollpunkt, kalles en *clamped* kurve. Dette kan vi oppnå med  $d + 1$  like skjøter i hver ende av skjøtvektoren. Her er  $n = 3$ ,  $d = 2$  og  $n + d + 1 = 6$  og vi kan bruke skjøtvektoren  $\{0, 0, 0, 1, 1, 1\}$ .

Fra oppgave 5.7.8 (som har samme skjøtvektor) får vi basisfunksjonene  $B_{0,2}(t) = (1 - t)^2$ ,  $B_{1,2}(t) = 2t(1 - t)$  og  $B_{2,2}(t) = t^2$ .

$$\begin{aligned}x(t) &= 1 \cdot (1 - t)^2 + 3 \cdot 2t(1 - t) + 4 \cdot t^2 = -t^2 + 4t + 1 \\y(t) &= 1 \cdot (1 - t)^2 + 3 \cdot 2t(1 - t) + 1 \cdot t^2 = -4t^2 + 4t + 1 \\ \mathbf{f}(t) &= (-t^2 + 4t + 1, -4t^2 + 4t + 1)\end{aligned}$$

Hvis vi setter inn  $t = 0$ , ser vi at  $\mathbf{f}(0) = (x(0), y(0)) = (1, 1)$ . Hvis vi setter inn  $t = 1$ , ser vi at  $\mathbf{f}(1) = (x(1), y(1)) = (4, 1)$ .

$$\mathbf{f}\left(\frac{1}{2}\right) = \left(x\left(\frac{1}{2}\right), y\left(\frac{1}{2}\right)\right) = \left(\frac{11}{4}, 2\right)$$

Det kan vises at kurven tangerer kontrollpolygonet i endepunktene. Det er derfor lett å skissere kurven ved hjelp av kontrollpolygonet og ett eller to utregnede punkter.

### 7.4.1 Lokal support I

Vi har sett i flere eksempler og oppgaver at

- B-spline basisfunksjoner av grad  $d$   $B_{i,d}(t) \geq 0$  over høyst  $d + 1$  intervaller i skjøtvektoren. For andre parameterverdier er verdien  $B_{i,d}(t) = 0$ . Vi tar med et eksempel for grad 1.  $B_{i,1}(t) \geq 0$  i intervallene  $[t_i, t_{i+1}]$  og  $[t_{i+1}, t_{i+2}]$  eller over høyst  $d + 2 = 3$  skjøter  $t_i, t_{i+1}, t_{i+2}$ .
- På hvert intervall  $[t_i, t_{i+1}]$  er det høyst  $d$  basisfunksjoner som er forskjellig fra 0. Vi trenger derfor ikke bruke mer enn noen få av basisfunksjonene for å beregne verdien til en splinefunksjon eller en splinekurve (likning (5.20)).
- Vi må finne ut hvilke intervaller/basisfunksjoner/kontrollpunkter som skal være med i beregningen.
- Vi beregner verdien av splinefunksjonen/splinekurven ved deBoor's algoritme, som ligner mye på deCasteljau's algoritme.

## 7.5 Finne riktig intervall

B-spline funksjoner og kurver har altså lokal support, som forklart ovenfor og i avsnitt 4.5. Vi kan derfor beregne verdien til en splinefunksjon ved å bruke likning (5.20) i stedet for likning (5.19). Enten vi bruker denne formelen eller deBoor's algoritme til beregning, må vi finne riktig intervaller, og vi bruker  $\mu$  for å indikere dette. Når vi skal beregne splinefunksjonen  $f(t)$  eller splinekurven  $\mathbf{f}(t)$  for en gitt parameterverdi  $t$ , er vi ute etter  $\mu$  slik at

$$t_\mu \leq t < t_{\mu+1} \quad (7.4)$$

### Eksempel

Anta  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$  som i oppgave 5.7.9 og  $t = 1.5$ . Da blir  $\mu = 3$ . Det er kun basisfunksjonene  $b_1, b_2$  og  $b_3$  som er positive for denne parameterverdien.

Anta  $t = 2.0 = t_4 = t_5 = t_6$ . Da er basisfunksjonene  $b_1, b_2$  og  $b_3$  ikkenegative.  $b_1(2.0) = b_2(2.0) = 0$  og bare  $b_3(2.0) = 1$  er gir bidrag:  $c_1 b_1(2.0) + c_2 b_2(2.0) + c_3 b_3(2.0) = 0 + 0 + c_3 \cdot 1$ . Se figur A.2.

Med multiplisitet  $d + 1$  i endene av skjøtvektoren, kan aldri  $\mu$  bli større enn  $n - 1$ . Antall skjøter totalt er  $n + d + 1$ , indeksen til den siste blir da  $n + d$ , og når vi trekker fra  $d + 1$  får vi altså  $n - 1$ . Siden vi har  $t_\mu \leq x$  i den venstre ulikheten, kan minste verdi for  $\mu$  bli  $d$ .

Gitt  $n$  kontrollpunkter og grad  $d$  vil hver basisfunksjon  $B_{i,d}(t) \geq 0$  på intervallet  $(t_i, t_{i+d+1})$ , altså over  $d+2$  skjøter eller  $d+1$  skjøtintervaller. Den  $n$ -te basisfunksjonen skal multipliseres med det  $n$ -te kontrollpunktet og vil da være ikkenegativ på intervallet  $(t_n, t_{n+d+1})$ . Når indekseringen starter på 0 som i C++, vil det  $n$ -te kontrollpunktet ha indeks  $n-1$ .

Følgende algoritme kan brukes til å finne  $\mu$  dersom vi har interpolasjon i endene ( $d+1$  skjøter i hver ende av skjøtvektoren, kalles **clamped** kurve):



Listing 7.3: Finne  $\mu$ 

```
int BSplineCurve::findKnotInterval(float x) {
    int my = n-1; // indeks til siste kontrollpunkt
    while (x < t[my])
        my--;
    return my;
}
```

Hvis vi ikke har dette kravet om  $d+1$  skjøter i hver ende, kan vi legge til en ekstra test og bruke denne modifiserte algoritmen:

Listing 7.4: Finne  $\mu$ 

```
int BSplineCurve::findKnotInterval(float x) {
    int my = n-1; // indeks til siste kontrollpunkt
    while (x < t[my] && my>d)
        my--;
    return my;
}
```

## 7.6 deBoor's algoritme

En B-spline basisfunksjon av grad  $d$  er definert over en skjøtvektor  $\mathbf{t}$ , og har en indeks  $i$ . Den kan være en vanlig funksjon eller en vektorfunksjon (som nedenfor). En spline funksjon eller en spline kurve er definert over en av grad  $d$  er definert over en skjøtvektor  $\mathbf{t}$ , og har  $n$  kontrollpunkter  $\mathbf{c} = c_0, \dots, c_{n-1}$ .

En klasse BSplineCurve kan da ha variablene  $d$ ,  $n$ ,  $\mathbf{t} = \{t_0, \dots, t_{n+d}\}$ , og  $\mathbf{c} = \{c_0, \dots, c_{n-1}\}$ . Algoritmen nedenfor returnerer en tredimensjonal vektor for gitt parameterverdi og er altså for en splinekurve i  $\mathbf{R}^3$ . Operatoroverlasting for datatypen Vector3d er nødvendig.

**Algoritme 7.1 (Cox-deBoor)** Vector3d BSplineCurve::evaluateBSplineSimple(float x)

```
{
    int my = findKnotInterval(x);

    std::vector<Vector3d> a;
    a.reserve(d+1);

    for (int j=0; j<=d; j++) {
        a[d-j] = c[my-j];
    }
    for (int k=d; k>0; k--) {
        int j = my-k;
        for (int i=0; i<k; i++) {
            j++;
            float w = (x-t[j])/(t[j+k]-t[j]);
            a[i] = a[i] * (1-w) + a[i+1] * w;
        }
    }
    return a[0];
}
```

Algoritmen er en triangulær algoritme, og i mange bøker er det satt opp triangelskjemaer for å forklare algoritmen. Vi skal sammenligne med deCasteljau's

algoritme og tar utgangspunkt i figur 5.11.

Anta at vi nå skal evaluere en funksjonsverdi eller et punkt på en splinekurve for  $x \in [2, 3)$ . Vi må da først finne riktig  $\mu = 2$ . På dette intervallet, generelt  $[t_\mu, t_{\mu+1})$  er det kun basisfunksjonene  $B_{\mu-d,d}(t), \dots, B_{\mu,d}(t)$  som er positive. Her blir dette  $B_{\mu-1,1}(t), \dots, B_{\mu,1}(t)$  som markert på figuren.

Vi observerer også følgende: Basisfunksjonene har to komponenter/ledd i henhold til avsnitt 5.5.2, og på det aktuelle intervallet  $[t_\mu, t_{\mu+1}) = [t_2, t_3) = [2, 3)$  er det  $(1-w)$ -delen (markert rød på figuren) som gir bidrag for  $B_{\mu-1,1}(t)$  og  $w$ -delen (markert grønn på figuren) som gir bidrag for  $B_{\mu,1}(t)$ . (Intervallet svarer til den karakteristiske funksjonen  $B_{2,0}(t)$ ).

Videre er  $(1-w)$ -delen faktisk indeksert  $(1-w_{i+1,d}(x))$  i avsnitt 5.5.2. Dette samsvarer selvsagt med at  $(1-w)$ -delen (rød) og  $w$ -delen (grønn) summerer seg opp til 1.

Algoritmen er veldig lik deCasteljau algoritmen. Her er de samme variabelnavnene brukt. Endringene fra deCasteljau algoritmen er vist i linjene nedenfor:

```
int j = my-k;
for (int i=0; i<k; i++) {
    j++;
    float w = (x-t[j])/(t[j+k]-t[j]);
```

Her er  $j$  startindeks for intervallet til den første basisfunksjonen som skal regnes med, og den ekstra linjen  $j++$ ; er tatt med for å understreke at første del av denne basisfunksjonen ikke gir bidrag i beregningen. For  $k=d$  blir altså  $j = \mu - 1$ . Videre beregnes i siste linje i deCasteljau algoritmen

$$a_i = a_i(1 - t) + a_{i+1} \cdot t$$

mens i siste linje i deBoor's algoritme

$$a_i = a_i(1 - w) + a_{i+1} \cdot w$$

### 7.6.1 Alternativ implementering

Man kan bruke en  $d \times d$  matrise til beregningene, og erstatte alle forekomster av  $a_i$  med  $a_{ik}$ . For grad 3 blir dette

```
Matrix4x4 BSplineCurve::evaluateBSpline(float x) {
    int my = findKnotInterval(x);
    Matrix4x4 A;
    for (int j=0; j<=d; j++) {
        A[d-j][d] = c[my-j];
    }
    for (int k=d; k>0; k--) {
        int j = my-k;
        for (int i=0; i<k; i++) {
            j++;
            float w = (x-t[j])/(t[j+k]-t[j]);
```

```

        A[i][k] = A[i][k] * (1-w) + A[i+1][k] * w;
    }
    return A;
}

```

## 7.6.2 Lokal support II

Anta at vi  $n = 3$  kontrollpunkter, grad  $d = 2$  og en skjøtvektor med  $d+1$  like skjøter i hver ende og ingen flere skjøter. Totalt har vi da  $2d+2$  skjøter (6 for grad  $d=2$ ). Hver basisfunksjon går over  $d+2$  skjøter (her: 4) og  $n + d + 1 = 6$ . Med skjøtvektoren  $\{0, 0, 0, 1, 1, 1\}$  svarer dette til B-spline versjonen av en kvadratisk Bezier kurve (oppgave 5.7.8).

I oppgave 5.7.10 har vi  $n = 5$  kontrollpunkter, grad  $d = 2$  og en skjøtvektor med  $n + d + 1 = 8$  elementer. De tre første elementene er 0, og de tre siste elementene er 3. Vi får da en **clamped** kurve - en kurve som interpolerer endepunktene.

Hvis vi bruker deBoor algoritmen til å evaluere kurven, trenger vi for hver parameterverdi  $t$  å finne  $\mu$ . Vi ser på figur A.2 at uansett hva  $t$  er, er det kun  $d + 1 = 3$  kvadratiske basisfunksjoner som er aktive i beregningen. De andre er 0. Dermed blir hvert punkt på kurven er vektet sum av tre kontrollpunkter, siden kontrollpunktene multipliseres med basisfunksjonene (se også avsnitt 5.7.8).

Selve kurven her er ikke spesielt spennende, men er lett å skissere (tegn kontrollpolygonet først).

Et eksempel på anvendelse av en splinekurve i spill fins i [9]. Dette er en kurve i  $R^2$  som kan løftes opp på en flate. For å oppnå konstant hastighet langs kurven, er reparametrisering nødvendig.

## 7.7 Derivasjon

B-spline kurver kan evalueres etter likning (5.20), og vi kan da beregne den deriverte til en B-spline kurve ved å derivere hver B-spline basisfunksjon og summere som før.

$$p'(t) = \sum_{i=\mu-d}^{\mu} c_i B'_{i,d}(t) \quad (7.5)$$

Vi skal nå vise hvordan vi deriverer B-spline basisfunksjonene.

### 7.7.1 Derivasjon av B-spline basisfunksjon

La  $d = 1$  i den rekursive definisjonen (5.18). Da er

$$B'_{i,1} = \frac{1}{t_{i+1} - t_i} B_{i,0} - \frac{1}{t_{i+2} - t_{i+1}} B_{i+1,0} = w'_{i,1} B_{i,0} - w'_{i+1,1} B_{i+1,0} \quad (7.6)$$

Dette er lett å oppnå ved å derivere (5.18) ved hjelp av produktregelen. Resultatet forteller oss at den deriverte til en første grads B-spline basisfunksjon er

en differanse mellom to basisfunksjoner av en grad lavere. Telleren i brøkene er graden  $d = 1$ . Det kan vises at dette gjelder generelt.

**Teorem 7.2** *Den deriverte av en B-spline basisfunksjon av grad  $d$  er*

$$\begin{aligned} B'_{i,d}(t) &= \frac{d}{t_{i+d} - t_i} B_{i,d-1}(t) - \frac{d}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(t) \\ &= d \left( w'_{i,d}(t) B_{i,d-1}(t) - w'_{i+1,d}(t) B_{i+1,d-1}(t) \right) \end{aligned} \quad (7.7)$$

### 7.7.2 Eksempel

La oss velge grad  $d = 2$  og skjøtvektoren  $\{3, 3, 3, 4, 5, 6, 6, 6\}$  som samsvarer med figur 5.16. Vi skal beregne den deriverte til en splinekurve i  $t = \frac{9}{2}$  etter likning (7.5). Da er  $\mu = 4$  og ligningen blir

$$p'(t) = c_2 B'_{2,2}(t) + c_3 B'_{3,2}(t) + c_4 B'_{4,2}(t)$$

Ligning (7.7) gir

$$\begin{aligned} B'_{2,2}\left(\frac{9}{2}\right) &= B_{2,1}\left(\frac{9}{2}\right) - B_{3,1}\left(\frac{9}{2}\right) = 0 - \frac{1}{2} = -\frac{1}{2} \\ B'_{3,2}\left(\frac{9}{2}\right) &= B_{3,1}\left(\frac{9}{2}\right) - B_{4,1}\left(\frac{9}{2}\right) = 0 \\ B'_{4,2}\left(\frac{9}{2}\right) &= B_{4,1}\left(\frac{9}{2}\right) - B_{5,1}\left(\frac{9}{2}\right) = \frac{1}{2} + 0 = \frac{1}{2} \end{aligned}$$

siden nevnerne  $t_{i+d} - t_i = t_{i+d+1} - t_{i+1} = d = 2$  når vi har en uniform skjøtvektor. Når vi setter inn disse verdiene i uttrykket ovenfor, får vi den deriverte uttrykt ved kontrollpunktene

$$p'\left(\frac{9}{2}\right) = \frac{c_4 - c_2}{2}$$

Kontrollpunktene  $c_i \in \mathbf{R}^3$ , så dette er en vektor i  $\mathbf{R}^3$ .

La oss nå beregne den deriverte i  $t = \frac{17}{4}$ . Da er fortsatt  $\mu = 4$  og vi får samme uttrykk for den deriverte som ovenfor. Når vi setter inn parameterverdien i (7.7), får vi

$$\begin{aligned} B'_{2,2}\left(\frac{17}{4}\right) &= B_{2,1}\left(\frac{17}{4}\right) - B_{3,1}\left(\frac{17}{4}\right) = 0 - \frac{3}{4} = -\frac{3}{4} \\ B'_{3,2}\left(\frac{17}{4}\right) &= B_{3,1}\left(\frac{17}{4}\right) - B_{4,1}\left(\frac{17}{4}\right) = \frac{3}{4} - \frac{1}{4} = \frac{2}{4} \\ B'_{4,2}\left(\frac{17}{4}\right) &= B_{4,1}\left(\frac{17}{4}\right) - B_{5,1}\left(\frac{17}{4}\right) = \frac{1}{4} - 0 = \frac{1}{4} \end{aligned}$$

Dette gir

$$p'\left(\frac{17}{4}\right) = \frac{-3 \cdot c_2 + 2 \cdot c_3 + c_4}{4}$$

## 7.8 Oppgaver

### 7.8.1

1. Tegn en skisse av hvordan deCasteljau algoritmen fungerer ved å velge  $t = \frac{1}{4}, \frac{1}{3}, \frac{2}{3}, \frac{3}{4}$
2. Skriv en egen klasse BezierCurve i C++.
3. Implementer nødvendige konstruktører og funksjoner slik at du kan tegne en kubisk Bezierkurve med kontrollpolygon i valgt farge.

### 7.8.2

Gitt kontrollpunkter (5,1), (1,2) og (3,4). Bruk deCasteljau's algoritme til å tegne opp den tilhørende kvadratiske Bezierkurven. Kontroller svaret ved utregning (bruk likning (5.10)).

### 7.8.3

Gitt kontrollpunkter (1,2), (3,4) og (3,1). Bruk deCasteljau's algoritme til å tegne opp den tilhørende kvadratiske Bezierkurven. Kontroller svaret ved utregning (bruk likning (5.10)).

### 7.8.4

Gitt kontrollpunkter (5,1), (1,2), (3,4) og (3,1). Bruk nå deCasteljau's algoritme til å tegne opp den tilhørende kvadratiske Bezierkurven. Kontroller svaret ved utregning (bruk likning (5.10)).

### 7.8.5

Skriv en C++ klasse BezierCurve som arver VisualObject. Et BezierCurve objekt skal representere en kubisk Bezierkurve. Funksjonene init() og draw() skal implementeres slik at både kontrollpolygonet og den kubiske Bezierkurven kan rendres med en valgt farge. Lever skjermbilde av Bezierkurve med kontrollpolygon.

### 7.8.6

La A(4,0), B(0,0), C(0,4) og D(4,4) være kontrollpunktene til en kubisk Bezierkurve.

- a) Tegn opp kontrollpolygonet og vis hvordan du bruker deCasteljau algoritmen til å finne punktet på kurven som svarer til parameterverdien  $t = \frac{1}{2}$ .
- b) Tegn opp et nytt kontrollpolygon og vis hvordan du bruker deCasteljau algoritmen til å finne punktet på kurven som svarer til parameterverdien  $t = \frac{3}{4}$ .
- c) Skisser Bezierkurven på figuren fra a) eller b).
- d) Vis ved utregning hvordan du kommer fram til de to punktene på Bezierkurven i a) og b)

### 7.8.7

La  $A(4,3)$ ,  $B(0,1)$ ,  $C(2,0)$  og  $D(3,4)$  være kontrollpunktene til en kubisk Bezierkurve.

- a) Tegn opp kontrollpolygonet og vis hvordan du bruker deCasteljau algoritmen til å finne punktet på kurven som svarer til parameterverdien  $t = \frac{1}{2}$ . Skisser: Se tidligere eksempler. Det kreves minimum en skisse hvor det er tydelig at kurven går gjennom det utregnede punktet og hvordan den deriverte er i endepunktene.
- b) Vis ved utregning hvordan du kommer fram til punktet på Bezierkurven i a).

### 7.8.8

Gitt kontrollpunkter  $(5,1)$ ,  $(1,2)$  og  $(3,4)$ . Bruk Neville's algoritme til å tegne opp den kvadratiske kurven som interpolerer kontrollpunktene. Kontroller svaret ved utregning som vist i 7.3.

### 7.8.9

Gitt kontrollpunkter  $(1,2)$ ,  $(3,4)$  og  $(3,1)$ . Bruk Neville's algoritme til å tegne opp den kvadratiske kurven som interpolerer kontrollpunktene. Kontroller svaret ved utregning som vist i 7.3.

### 7.8.10

Gitt kontrollpunkter  $(5,1)$ ,  $(1,2)$ ,  $(3,4)$  og  $(3,1)$ . Bruk nå Neville's algoritme til å tegne opp den kubiske kurven som interpolerer kontrollpunktene. Kontroller svaret ved utregning som vist i 7.3.

### 7.8.11

Gitt en kvadratisk kurve som interpolerer punktene  $P(0,2)$ ,  $Q(2,0)$  og  $R(3,2)$ .  $PQR$  er altså kontrollpunktene.

- a) Bruk Neville's algoritme til å regne ut punktet på kurven som svarer til parameterverdien  $t = \frac{1}{2}$ .
- b) Bruk Neville's algoritme til å regne ut punktet på kurven som svarer til parameterverdien  $t = \frac{3}{2}$ .
- c) La  $t \leq 2$  og skisser kurven.

### 7.8.12

Bruk skjøtvektoren og basisfunksjonene i oppgave 5.7.8. La  $\mathbf{c} \in \mathbf{R}^2$  være gitt ved  $\mathbf{c} = \{(0,1), (0,0), (1,0)\}$ .

1. Bestem den kvadratiske splinekurven med kontrollpunkter  $\mathbf{c}$ .
2. Tegn splinekurven.

### 7.8.13

Bruk skjøtvektoren og basisfunksjonene i oppgave 5.7.10. La  $\mathbf{c} \in \mathbf{R}^2$  være gitt ved  $\mathbf{c} = \{(0, 1), (1, 0), (2, 1), (3, 0), (4, 1)\}$ . Bruk deBoor algoritmen til å skissere den kvadratiske splinekurven med kontrollpunkter  $\mathbf{c}$ .

### 7.8.14

La  $\mathbf{t} = \{0, 0, 0, 1, 1, 1\}$  som i oppgave 5.7.7. La  $\mathbf{c}_0(1, 0, 0)$ ,  $\mathbf{c}_1(1, 1, 2)$  og  $\mathbf{c}_2(1, 2, 0)$  være et kontrollpolygon for en kvadratisk splinekurve  $\mathbf{f}(\mathbf{t})$ .

1. Regn ut  $\mathbf{f}(\mathbf{t})$  for  $t$ -verdiene  $0, \frac{1}{2}, 1$ .
2. Bruk lemma 7.2 til å bestemme uttrykkene for  $B'_{0,2}$ ,  $B'_{1,2}$  og  $B'_{2,2}$ .
3. Regn ut  $\mathbf{f}'(\mathbf{t})$  for  $t$ -verdiene  $0, \frac{1}{2}, 1$ .
4. Regn ut  $\mathbf{f}'(\mathbf{t})$  for  $t$ -verdiene  $0, \frac{1}{2}, 1$  og kontrollpunktene  $(0,0,0)$ ,  $(0,1,0)$ ,  $(0,2,0)$ .

### 7.8.15

1. Bruk formelen for derivasjon av B-splines ((7.7)) til å derivere B-spline basisfunksjonene i oppgave 5.7.13.
2. La  $\mathbf{c}_0(0, 1, 0)$ ,  $\mathbf{c}_1(1, 1, 2)$ ,  $\mathbf{c}_2(2, 1, 2)$ , og  $\mathbf{c}_3(3, 1, 0)$  være et kontrollpolygon for en kvadratisk splinekurve  $\mathbf{f}(\mathbf{t})$ . Regn ut  $\mathbf{f}'(\mathbf{t})$  for  $t$ -verdiene  $0, \frac{1}{2}, 1, \frac{3}{2}, 2$ .
3. La  $\mathbf{c}_0(0, 2, 0)$ ,  $\mathbf{c}_1(1, 2, 0)$ ,  $\mathbf{c}_2(2, 2, 0)$ , og  $\mathbf{c}_3(3, 2, 0)$  være et kontrollpolygon for en kvadratisk splinekurve  $\mathbf{f}(\mathbf{t})$ . Regn ut  $\mathbf{f}'(\mathbf{t})$  for  $t$ -verdiene  $0, \frac{1}{2}, 1, \frac{3}{2}, 2$ .

# Tillegg A

## Fasit

### A.1

#### Fasit for 2.8.1

Vi får 4 addisjoner og 4 multiplikasjoner, altså  $O(n)$ .

$$\begin{aligned}a = 4 : \quad f &= 3 \cdot 0 + 4 &= 4 \\a = 3 : \quad f &= 3 \cdot 4 + 0 &= 12 \\a = 2 : \quad f &= 3 \cdot 12 + 8 &= 44 \\a = 1 : \quad f &= 3 \cdot 44 + 1 &= 133 \\a = 0 : \quad f &= 3 \cdot 133 + 2 &= 401\end{aligned}$$

Vi kan skrive polynomet på nøstet form og ser da hvordan utregningen foregår:

$$\begin{aligned}2 + x(1 + x(8 + x(0 + x(4)))) \\0 + 3 \cdot 4 \\8 + 3 \cdot 12 \\1 + 3 \cdot 44 \\2 + 3 \cdot 133\end{aligned}$$

#### Fasit til 2.8.4

Vi må først finne de partiellderiverte, og deretter løse ligningene vi får når vi setter disse lik 0.  $f_1(x, y) = xy$   $\frac{\partial f}{\partial x}(x, y) = y$  og  $\frac{\partial f}{\partial y}(x, y) = x$   
 $\frac{\partial f}{\partial x} = 0 \Leftrightarrow y = 0$  og  $\frac{\partial f}{\partial y} = 0 \Leftrightarrow x = 0$ . Funksjonen har ett stasjonært punkt  $(0, 0)$ .

$f_2(x, y) = x^2 + y^2$ . De partiellderiverte blir:  $\frac{\partial f}{\partial x}(x, y) = 2x$  og  $\frac{\partial f}{\partial y}(x, y) = 2y$   
 $\frac{\partial f}{\partial x} = 0 \Leftrightarrow x = 0$  og  $\frac{\partial f}{\partial y} = 0 \Leftrightarrow y = 0$  Funksjonen har ett stasjonært punkt  $(0, 0)$ .

$f_3(x, y) = \sin xy$ . Vi finner her de partiellderiverte ved å bruke kjerneregelen:  
 $\frac{\partial f}{\partial x}(x, y) = y \cos xy$  og  $\frac{\partial f}{\partial y}(x, y) = x \cos xy$ .  $\frac{\partial f}{\partial x} = 0 \Leftrightarrow y = 0 \vee \cos xy = 0 \Leftrightarrow x =$



$0 \vee y = 0$ . Vi får de samme løsningene fra ligningen  $\frac{\partial f}{\partial y} = 0$ .

Dette gir

1.  $(0,0)$
2.  $\cos xy = 0 \Leftrightarrow xy = \frac{\pi}{2} \Leftrightarrow y = \frac{\pi}{2x}$  fra begge ligninger gir stasjonære punkter over en hyperbel i xy-planet.
3.  $\cos xy = 0 \Leftrightarrow xy = -\frac{\pi}{2} \Leftrightarrow y = -\frac{\pi}{2x}$  fra begge ligninger gir stasjonære punkter over en hyperbel i xy-planet.
4. Flere løsninger har vi ikke.  $x = 0$  fra ligning 1 sammen med  $\cos xy = 0 \Leftrightarrow xy = 1 \Leftrightarrow y = \frac{1}{x}$  fra ligning 2 er en umulig kombinasjon (gir null i nevner).

$$f_4(x, y) = \sin x \cdot \sin y.$$

$$\text{Partiellderiverte: } f_x = \frac{\partial f}{\partial x}(x, y) = \cos x \sin y \text{ og } f_y = \frac{\partial f}{\partial y}(x, y) = \sin x \cos y.$$

$$f_x = 0 \Leftrightarrow \cos x = 0 \vee \sin y = 0 \Leftrightarrow x = \frac{\pi}{2} \vee x = \frac{3\pi}{2} \vee y = 0 \vee y = \pi.$$

$$f_y = 0 \Leftrightarrow \sin x = 0 \vee \cos y = 0 \Leftrightarrow x = 0 \vee x = \pi \vee y = \frac{\pi}{2} \vee y = \frac{3\pi}{2}.$$

For hver x-verdi i ligning 1 får vi en y-verdi fra ligning 2:

$$\left(\frac{\pi}{2}, \frac{\pi}{2}\right), \left(\frac{\pi}{2}, \frac{3\pi}{2}\right), \left(\frac{3\pi}{2}, \frac{\pi}{2}\right), \left(\frac{3\pi}{2}, \frac{3\pi}{2}\right).$$

Tilsvarende, for hver x-verdi fra ligning 2 en y-verdi fra ligning 1:

$$(0, 0), (0, \pi), (\pi, 0), (\pi, \pi)$$

### Fasit til 2.8.5

Finn tangentplanet til funksjonen  $f(x, y) = x^2 + y^2$  i punktet  $(\frac{1}{2}, \frac{1}{2})$ .

$$f_x(x, y) = 2x$$

$$f_y(x, y) = 2y$$

$$f\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2}$$

$$f_x\left(\frac{1}{2}, \frac{1}{2}\right) = 1$$

$$f_y\left(\frac{1}{2}, \frac{1}{2}\right) = 1$$

$$L(x, y) = \frac{1}{2} + 1 \cdot \left(x - \frac{1}{2}\right) + 1 \cdot \left(y - \frac{1}{2}\right) = x + y - \frac{1}{2}$$

### Fasit til 2.8.6

$$\begin{aligned} \int_3^6 \int_{-2}^1 (x^2 + xy^2) dx dy &= \int_3^6 \left[ \frac{1}{3}x^3 + \frac{1}{2}x^2y^2 \right]_{-2}^1 dy = \int_3^6 \left[ \left(\frac{1}{3} + \frac{1}{2}y^2\right) - \left(-\frac{8}{3} + 2y^2\right) \right] dy = \\ &= \int_3^6 \left( 3 - \frac{3}{2}y^2 \right) dy = \left[ 3y - \frac{y^3}{2} \right]_3^6 = -\frac{171}{2} \end{aligned}$$

Integralet er negativt, og et volum må være positivt.

**Fasit til 2.8.7**

$$\begin{aligned}
\int_0^3 \int_0^{\frac{3-x}{2}} e^{x+y} dy dx &= \int_0^3 [e^{x+y}]_0^{\frac{3-x}{2}} dx = \int_0^3 \left( e^{x+\frac{3-x}{2}} - e^{x+0} \right) dx = \\
&= \int_0^3 \left( e^{\frac{x+3}{2}} - e^x \right) dx = \left[ 2e^{\frac{x+3}{2}} - e^x \right]_0^3 = (2e^3 - e^3) - (2e^{\frac{3}{2}} - e^0) = \\
&= e^3 - 2e^{\frac{3}{2}} + 1 = (e^{\frac{3}{2}} - 1)^2.
\end{aligned}$$

**Fasit til 2.8.8**

$$\begin{aligned}
\int_0^1 \int_0^{1-x} xy dy dx &= \int_0^1 \left[ \frac{1}{2} xy^2 \right]_0^{1-x} dx \\
&= \int_0^1 \frac{1}{2} x(1-x)^2 dx \\
&= \int_0^1 \left( \frac{1}{2} x^3 - x^2 + \frac{1}{2} x \right) dx \\
&= \left[ \frac{1}{8} x^4 - \frac{1}{3} x^3 + \frac{1}{4} x^2 \right]_0^1 \\
&= \frac{1}{8} - \frac{1}{3} + \frac{1}{4} = \frac{3-8+6}{24} = \frac{1}{24}.
\end{aligned}$$

**Fasit til 2.8.9**

- a)  $\frac{\partial h}{\partial x} = y - 1$  og  $\frac{\partial h}{\partial y} = x - 1$ . Det stasjonære punktet blir enkelt  $(1, 1)$ .
- b)  $\frac{\partial^2 h}{\partial x^2} = 0$ ,  $\frac{\partial^2 h}{\partial x \partial y} = 1$  og  $\frac{\partial^2 h}{\partial y^2} = 0$ .  
 $\Delta = \frac{\partial^2 h}{\partial x^2} \cdot \frac{\partial^2 h}{\partial y^2} - \left( \frac{\partial^2 h}{\partial x \partial y} \right)^2 = -1$ .  
 Det betyr at  $(1, 1)$  er et sadelpunkt.
- c) Regn ut volumet avgrenset av grafen til  $h$ , x-aksen, y-aksen og den rette linjen  $y = 1 - x$ .

$$\begin{aligned}
\int_0^1 \int_0^{1-x} (xy - x - y + 1) dy dx &= \int_0^1 \left[ \frac{1}{2} xy^2 - xy - \frac{1}{2} y^2 + y \right]_0^{1-x} dx \\
&= \int_0^1 \left( \frac{1}{2} x(1-x)^2 - x(1-x) - \frac{1}{2} (1-x)^2 + (1-x) \right) dx \\
&= \int_0^1 \left( \frac{1}{2} x^3 - \frac{1}{2} x^2 - \frac{1}{2} x + \frac{1}{2} \right) dx \\
&= \left[ \frac{1}{8} x^4 - \frac{1}{6} x^3 - \frac{1}{4} x^2 + \frac{1}{2} x \right]_0^1 \\
&= \frac{1}{8} - \frac{1}{6} - \frac{1}{4} + \frac{1}{2} = \frac{3-4-6+12}{24} = \frac{5}{24}.
\end{aligned}$$

**Fasit til 2.8.10**

- a) (oppgaven er hentet fra [3], 14.10)

$$\frac{\partial f}{\partial x} = 3 - 3x^2 - 3y^2 = 3(1 - (x^2 + y^2)) \text{ og } \frac{\partial f}{\partial y} = -6xy$$

Vi har  $f(-1, 1) = 1$ ,  $\frac{\partial f}{\partial x}(-1, 1) = -3$  og  $\frac{\partial f}{\partial y}(-1, 1) = 6$ . Ligningen for tangentplanet blir da

$$L(x, y) = 1 - 3(x + 1) + 6(y - 1) = -3x + 6y - 8$$

b)

$$\frac{\partial^2 f}{\partial x^2} = -6x, \quad \frac{\partial^2 f}{\partial x \partial y} = -6y, \quad \text{og} \quad \frac{\partial^2 f}{\partial y^2} = -6x$$

c)  $\frac{\partial f}{\partial x} = 0 \Leftrightarrow x^2 + y^2 = 1$ , altså ligger de kritiske punktene på enhetssirkelen. I tillegg  $\frac{\partial f}{\partial y} = 0 \Leftrightarrow x = 0 \vee y = 0$ .

Dette gir følgende kritiske/stasjonære punkter:  $(1, 0)$ ,  $(0, 1)$ ,  $(-1, 0)$  og  $(0, -1)$ . Vi må da bruke determinanten til Hessematrixen for hvert av disse punktene. Vi regner ut  $\Delta = \det(H)$  hvor

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial y \partial x} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} f_{xx} & f_{yx} \\ f_{xy} & f_{yy} \end{bmatrix} = 36(x^2 - y^2)$$

1.  $(1, 0)$  gir  $f_{xx} = -6$ ,  $f_{yx} = 0$ ,  $f_{xy} = -6$ ,  $f_{yy} = 0$ .  $\Delta = 36$  og  $f_{xx} < 0 \Rightarrow$  lokalt maksimum.
2.  $(0, 1)$  gir  $f_{xx} = 0$ ,  $f_{yx} = -6$ ,  $f_{xy} = -6$ ,  $f_{yy} = -0$ .  $\Delta = -36 \Rightarrow$  sadelpunkt.
3.  $(-1, 0)$  gir  $f_{xx} = 6$ ,  $f_{yx} = 0$ ,  $f_{xy} = 0$ ,  $f_{yy} = 6$ .  $\Delta = 36$  og  $f_{xx} > 0 \Rightarrow$  lokalt minimum.
4.  $(0, -1)$  gir  $f_{xx} = 0$ ,  $f_{yx} = 6$ ,  $f_{xy} = 6$ ,  $f_{yy} = 0$ .  $\Delta = -36$  og  $f_{xx} < 0 \Rightarrow$  sadelpunkt.

### Fasit til 2.8.11

Vi finner grensene for integralet og setter opp (vi kan bytte grensene og dy med dx hvis vi vil, det kan føre til enklere utregninger):

$$\int_0^{1-x} \int_0^1 (1-x-y) \, dx \, dy = \int_0^1 \int_0^{1-x} (1-x-y) \, dy \, dx$$

Vi regner først ut det indre integralet.

$$\int_0^{1-x} (1-x-y) \, dy = \left[ y - xy - \frac{1}{2}y^2 \right]_0^{1-x} = \frac{1}{2} - x + \frac{1}{2}x^2$$

slik at

$$\begin{aligned} \int_0^1 \int_0^{1-x} (1-x-y) \, dy \, dx &= \int_0^1 \left( \frac{1}{2} - x + \frac{1}{2}x^2 \right) dx \\ &= \left[ \frac{x}{2} - \frac{x^2}{2} + \frac{x^3}{6} \right]_0^1 = \frac{1}{6} \end{aligned}$$

**Fasit til 2.8.12****Normal:**

$$\begin{aligned}
f(x, y) &= (1-x)(1-y) = xy - x - y + 1 \\
f\left(\frac{3}{4}, \frac{3}{4}\right) &= \frac{9}{16} - \frac{3}{4} - \frac{3}{4} + 1 = \frac{1}{16} \text{ (men trengs ikke for normalen)} \\
\frac{\partial f}{\partial x} &= y - 1 \\
\frac{\partial f}{\partial y} &= x - 1 \\
\frac{\partial f}{\partial x}\left(\frac{3}{4}, \frac{3}{4}\right) &= -\frac{1}{4} \\
\frac{\partial f}{\partial y}\left(\frac{3}{4}, \frac{3}{4}\right) &= -\frac{1}{4} \\
\vec{n} &= [1, 0, -\frac{1}{4}] \times [0, 1, -\frac{1}{4}] = [\frac{1}{4}, \frac{1}{4}, 1] \\
|\vec{n}| &= \frac{\sqrt{18}}{4} \\
\frac{\vec{n}}{|\vec{n}|} &= [\frac{1}{\sqrt{18}}, \frac{1}{\sqrt{18}}, \frac{4}{\sqrt{18}}] = [\frac{\sqrt{2}}{6}, \frac{\sqrt{2}}{6}, \frac{4\sqrt{2}}{6}] \approx [0.236, 0.236, 0.943]
\end{aligned}$$

**Integral:** Vi integrerer over en trekant i  $xy$ -planet, derav følger integrasjonsgrensene.

$$\begin{aligned}
&\int_0^1 \int_0^{1-x} (1-x)(1-y) \, dy \, dx \\
&= \int_0^1 \left( (1-x) \left[ y - \frac{y^2}{2} \right]_{y=0}^{1-x} \right) dx \\
&= \int_0^1 \left( (1-x) \left( (1-x) - \frac{1}{2}(1-x)^2 \right) \right) dx \\
&= \int_0^1 \left( (1-x)^2 - \frac{1}{2}(1-x)^3 \right) dx \\
&= \left[ -\frac{1}{3}(1-x)^3 + \frac{1}{8}(1-x)^4 \right]_0^1 \\
&= \frac{1}{3} - \frac{1}{8} = \frac{5}{24}
\end{aligned}$$

hvor vi i siste integrasjon har brukt  $u = 1 - x$  og  $du = -dx$ .

## Fasit til 2.8.13

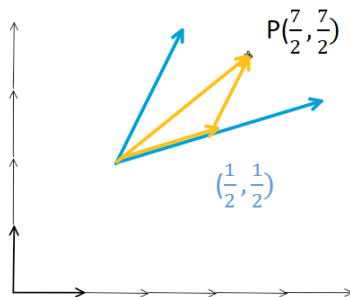
Listing A.1: Numerisk integrasjon med midtpunktmetoden

```
#include <iostream>
using namespace std;

// Funksjonen i eksempel 2.6.1
double f(double x, double y) { return x*x*y; }

int main() {
    double h{0.05};
    const double EPS{0.001};
    double volum{0.0};
    double xmax{1.0};           // Ovre integrasjonsgrense
    double ymax;                 // Ovre integrasjonsgrense

    // Regner ut integralet numerisk med midtpunktmetoden
    for (double x=0.05; x<xmax+EPS; x+=h) {
        ymax = 1.0 - x;
        for (double y=0.05; y<ymax+EPS; y+=h)
            volum = volum + h*h*f(x,y);
    }
    cout << "Volumet: " << volum;
    return 0;
}
Svaret blir 0.0187.
```



Figur A.1: Illustrasjon til oppgave 3.7.3.

## A.2

### Fasit for 3.7.1

Vi får

$$\mathbf{A} = \begin{bmatrix} 3 & -1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{A}^{-1} = \begin{bmatrix} \frac{2}{7} & \frac{1}{7} \\ -\frac{1}{7} & \frac{3}{7} \end{bmatrix}$$

og 1) Transformasjon fra A til E:  $\mathbf{Ax}_1 = [1, 5]$ ,  $\mathbf{Ax}_2 = [-4, 1]$ , 2) transformasjon fra E til A:  $\mathbf{A}^{-1}\mathbf{x} = [\frac{4}{7}, \frac{5}{7}]$ .

### Fasit for 3.7.2

Vi får

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{A}^{-1} = \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{3}{5} \end{bmatrix}$$

og 1) Transformasjon fra A til E:  $[4, 3]$ ,  $[5, 0]$ , 2) transformasjon fra E til A:  $[0, 1]$  og  $[\frac{1}{5}, \frac{2}{5}]$

### Fasit for 3.7.3

1) Med notasjon som i avsnitt 3.4 får vi ved translasjon fra O til O' for P:  $\mathbf{v} = [\frac{7}{2}, \frac{7}{2}] - [(\frac{3}{2}, 2)] = [2, \frac{3}{2}]$ . Vi får

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{A}^{-1} = \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{3}{5} \end{bmatrix}$$

og

$$x = \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{3}{5} \end{bmatrix} \cdot \begin{bmatrix} 2 \\ \frac{3}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

2) Translasjon fra O til O' for Q:  $\mathbf{v} = [\frac{5}{2}, \frac{3}{2}] - [(\frac{3}{2}, 2)] = [1, -\frac{1}{2}]$ .

$$\mathbf{A}^{-1} \mathbf{v} = [\frac{1}{2}, -\frac{1}{2}]$$

### Fasit til 3.7.4

Skjæringspunktet mellom de tre planene blir  $\frac{1}{4}(1, -3, 6)$ .

**Fasit til 3.7.5**

Verdiene i matrisen over er funksjonsverdiene til et tredjegradspolynom på formen  $ax^3 + bx^2 + cx + d$  i punktene  $x = \frac{1}{2}, x = 1, x = 2$  og  $x = 3$ . bf x vektoren inneholder de ukjente koeffisientene  $a, b, c, d$ . Svaret blir  $[\frac{1}{2} \ -2 \ 3 \ -1]$  - altså er polynomet  $p(x) = \frac{1}{2}x^3 - 2x^2 + 3x - 1$ .

## A.3

### Fasit til 4.6.1

Vi skal konstruere et andregradspolynom som interpolerer punktene  $(0, 1)$ ,  $(2, 0)$  og  $(3, 4)$ .

Vi får et polynom på formen  $p(x) = ax^2 + bx + c$  og ser at siden  $(0, 1)$  er et interpolasjonspunkt, må vi ha  $c = 1$ . Interpolasjonsbetingelsene  $p(x) = ax^2 + bx + c$  reduseres da til to ligninger med to ukjente:

$$\begin{aligned} 0 &= a \cdot 2^2 + b \cdot 2 + 1 \\ 4 &= a \cdot 3^2 + b \cdot 3 + 1 \end{aligned}$$

som kan skrives

$$\begin{aligned} -1 &= a \cdot 2^2 + b \cdot 2 \\ 3 &= a \cdot 3^2 + b \cdot 3 \end{aligned}$$

På matriseform  $\mathbf{Ax} = \mathbf{b}$ , med

$$\begin{bmatrix} 4 & 2 \\ 9 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

For en  $2 \times 2$  matrise  $\mathbf{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$  er  $\mathbf{B}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$  Vi får da

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$$-\frac{1}{6} \begin{bmatrix} 3 & -2 \\ -9 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ -\frac{7}{2} \end{bmatrix}$$

$$f(x) = \frac{3}{2}x^2 - \frac{7}{2}x + 1.$$

### Fasit til 4.6.2

$$p(x) = x^2 - 4x + 3$$

### Fasit til 4.6.3

$$p(x) = \frac{1}{2}x^3 - \frac{3}{2}x^2 - \frac{1}{2}x + \frac{3}{2}.$$

### Fasit til 4.6.4

Vi kan sette opp fire ligninger med fire ukjente for å finne konstantene  $\mathbf{x}^T = [a \ b \ c \ d]$ . Koeffisientmatrisen blir

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \\ \frac{27}{8} & \frac{9}{4} & \frac{3}{2} & 1 \\ 27 & 9 & 3 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1 \\ \frac{9}{16} \\ -\frac{5}{16} \\ 4 \end{bmatrix}$$

Løsningen blir  $p(x) = \frac{1}{2}x^3 - x^2 - \frac{1}{2}x + 1$ .



### Fasit til 4.6.5

På matriseform  $\mathbf{Ax}=\mathbf{b}$  med  $\mathbf{x}^T = [a \ b \ c]$

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

(husk at hver rad i A-matrisen svarer til  $x_i^2 \ x_i^1 \ x_i^0$  som blir til  $x_i^2 \ x_i \ 1$ ). Vi får  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  som blir

$$\frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -1 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Altså blir  $p(x) = -x^2 + 1$ .

2) Her blir hver rad i A-matrisen  $x_i^3 \ x_i^2 \ x_i^1 \ x_i^0$

$$A = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -\frac{1}{8} & \frac{1}{4} & -\frac{1}{2} & 1 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  blir

$$\frac{1}{6} \begin{bmatrix} -4 & 8 & -8 & 4 \\ 4 & -4 & -4 & 4 \\ 1 & -8 & 8 & -1 \\ -1 & 4 & 4 & -1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{2}{\sqrt{3}} \\ 0 \\ \frac{2}{\sqrt{3}} \end{bmatrix}$$

Her blir da  $p(x) = -\frac{2}{\sqrt{3}}x^2 + \frac{2}{\sqrt{3}}$ . Til tross for at vi prøver å beregne et polynom av grad opp til 3, ender vi opp med et kvadratisk polynom.

### Fasit til 4.6.6

Her får vi tre ligninger med tre ukjente a, b og c:

$$\begin{aligned} a \cdot 0 + b \cdot 0 + c &= 0 \\ a \frac{\pi^2}{4} + b \frac{\pi}{2} + c &= 1 \\ a \cdot \pi^2 + b \cdot \pi + c &= 0 \end{aligned}$$

På matriseform med  $\mathbf{x}^T = [a \ b \ c]$  og

$$A = \begin{bmatrix} 0 & 0 & 1 \\ \frac{\pi^2}{4} & \frac{\pi}{2} & 1 \\ \pi^2 & \pi & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Vi finner  $p_1(x) = -\frac{4}{\pi^2}x^2 + \frac{4}{\pi}x$ .

Vi kan regne ut  $p_2(x)$  på tilsvarende måte, men også ved et annet resonnement: Vi vet at polynomet er av grad 2 og vi kjenner to nullpunkter, slik at polynomet må være på formen  $p_2(x) = a(x - \pi)(x - 2\pi)$ . Vi kan da bestemme konstanten  $a$  ved å sette inn verdiene for det siste (midterste) interpolasjonspunktet:  $a(\frac{\pi}{2} - \pi)(\frac{3\pi}{2} - 2\pi) = -1 \Leftrightarrow a = \frac{4}{\pi^2}$  og vi får  $p_2(x) = \frac{4}{\pi^2}(x - \pi)(x - 2\pi) = \frac{4}{\pi^2}(x^2 - 3\pi x + 2\pi^2) = \frac{4}{\pi^2}x^2 - \frac{12}{\pi}x + 8$ .

## A.4

### Fasit til 4.6.7

Vi skal først bestemme en lineær minste kvadraters tilpasning og får

$$\mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 2 \\ 4 & 2 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

og videre

$$\mathbf{A}^T = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{B} = \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 30 & 10 \\ 10 & 4 \end{bmatrix} \quad \mathbf{c} = \mathbf{A}^T \mathbf{y} = \begin{bmatrix} 19 \\ 7 \end{bmatrix}$$

Løsning:  $\mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{5} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$  gir  $\mathbf{x} = \mathbf{B}^{-1} \mathbf{c} = [\frac{3}{10}, 1]$ , altså  $a = \frac{3}{10}$  og  $b = 1$  og ligningen for linjen blir  $y = \frac{3}{10}x + 1$ .

Nå skal vi bruke minste kvadraters metode og andregradspolynom med de samme punktene. Vi skal ha et polynom på formen  $ax^2 + bx + c$  og får  $\mathbf{x}^T = [a \ b \ c]$  og

$$\mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 2 \\ 9 & 3 & 2 \\ 16 & 4 & 2 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

Videre,

$$\mathbf{A}^T = \begin{bmatrix} 1 & 4 & 9 & 16 \\ 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{B} = \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 354 & 100 & 30 \\ 100 & 30 & 10 \\ 30 & 10 & 4 \end{bmatrix} \quad \mathbf{c} = \mathbf{A}^T \mathbf{y} = \begin{bmatrix} 59 \\ 19 \\ 7 \end{bmatrix}$$

Løsning:

$$\mathbf{B}^{-1} = \begin{bmatrix} 0.25 & -1.25 & 1.25 \\ -1.25 & 6.45 & -6.75 \\ 1.25 & -6.75 & 7.75 \end{bmatrix}$$

og vi får  $\mathbf{x} = \mathbf{B}^{-1} \mathbf{c} = [-0.25, 1.55, -0.25]$ , altså  $a = -0.25, b = 1.55$  og  $c = -0.25$  og ligningen for andregradsfunksjonen blir  $f(x) = -0.25x^2 + 1.55x - 0.25$ .

### Fasit til 4.6.8

Vi får ved minste kvadraters metode  $a = \frac{3}{14}$  og  $b = 0$ , altså er ligningen for den rette linjen  $y = \frac{3}{14}x$ .

### Fasit til 4.6.9

Punkter:  $(4, 3), (0, 1), (2, 0)$  og  $(3, 4)$ . Vi får  $\mathbf{x}^T = [a \ b]$  og

$$\mathbf{y} = \begin{bmatrix} 3 \\ 1 \\ 0 \\ 4 \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} 4 & 0 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 4 & 1 \\ 0 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \quad \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 29 & 9 \\ 9 & 4 \end{bmatrix} \quad \mathbf{A}^T \mathbf{y} = \begin{bmatrix} 24 \\ 8 \end{bmatrix}$$

Løsning: , slik at her er  $\mathbf{B}^{-1} = \frac{1}{35} \begin{bmatrix} 4 & -9 \\ -9 & 29 \end{bmatrix}$  og vi får  $\mathbf{x} = \mathbf{B}^{-1}\mathbf{c} = [\frac{24}{35}, \frac{16}{35}]$ ,  
altså  $a = \frac{24}{35}$  og  $b = \frac{16}{35}$  og ligningen for linjen blir  $y = \frac{24}{35}x + \frac{16}{35} \approx 0.69x + 0.46$ .

#### Fasit til 4.6.10

For hvert punkt skal vi sette inn  $(x_i, y_i)$  i ligningen  $y_i = ax_i^2 + bx_i + c$  se også (4.9). Vi har her 5 punkter og vi får da 5 rader i A-matrisen.

```
A =
    0.25    0.50    1.00
    0.25    0.50    1.00
    6.25    2.50    1.00
    9.00    3.00    1.00
   16.00    4.00    1.00
```

```
A^T =
    0.25    0.25    6.25    9.00   16.00
    0.50    0.50    2.50    3.00    4.00
    1.00    1.00    1.00    1.00    1.00
```

```
y = [ 46.75, 12.00, 4.50 ]
```

```
B=A^T * A =
   376.1875   106.8750   31.7500
   106.8750    31.7500   10.5000
    31.7500    10.5000    5.0000
```

```
B^-1 =
    0.1254   -0.5198    0.2951
   -0.5198    2.2573   -1.4396
    0.2951   -1.4396    1.3491
```

```
x = [ 0.9541, -3.6912, 2.593 ]
```

$f(x) = 0.9541x^2 - 3.6912x + 2.593$ .

#### Fasit til 4.6.11

For hvert punkt skal vi sette inn  $(x_i, y_i)$  i ligningen  $y_i = ax_i^2 + bx_i + c$  se også (4.9). Vi har her 5 punkter og vi får da 5 rader i A-matrisen.

```
A =
    0.25    0.50    1.00
    0.25    0.50    1.00
    6.25    2.50    1.00
    9.00    3.00    1.00
   16.00    4.00    1.00
```

```
A^T =
    0.25    0.25    6.25    9.00   16.00
    0.50    0.50    2.50    3.00    4.00
```

```

1.00    1.00    1.00    1.00    1.00

y = [ 46.75, 12.00, 4.50 ]

```

```

B=A^T * A =
376.1875  106.8750  31.7500
106.8750   31.7500  10.5000
31.7500   10.5000   5.0000

```

```

B^-1 =
0.1254  -0.5198   0.2951
-0.5198   2.2573  -1.4396
0.2951  -1.4396   1.3491

```

```

x = [ 0.9541, -3.6912, 2.593 ]

```

$$f(x) = 0.9541x^2 - 3.6912x + 2.593.$$

#### Fasit til 4.6.13

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \pi^3 & \pi^2 & \pi & 1 \\ 3 & 2 & 1 & 0 \\ 3\pi^2 & 2\pi & 1 & 0 \end{bmatrix}$$

og  $\mathbf{b}^T = [0 \ 0 \ 1 \ -1]$  gir  $p(x) = -0.46x^2 + 1.93x - 1.46$ .

#### Fasit til 4.6.14

La  $x_0 = 0$ ,  $x_1 = 1$ ,  $y_0 = 0$  og  $y_1 = 1$  og gitt funksjonen  $g(x) = \sin \frac{\pi}{2}x$ . La  $p(x)$  være polynomet for kubisk Hermite interpolasjon av  $g$  på intervallet  $[0, 1]$ .

- a) Vi har  $g'(x) = \frac{\pi}{2} \cos \frac{\pi}{2}x$  og får da  $y'_0 = \frac{\pi}{2}$  og  $y'_1 = 0$ .  
 $p(x) = ax^3 + bx^2 + cx + d$  og  $p'(x) = 3ax^2 + 2bx + c$  innsatt for  $x_0 = 0$  og  $x_1 = 1$  gir på matriseform  $\mathbf{A}\mathbf{x}=\mathbf{b}$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

$$\mathbf{x}^T = [a \ b \ c \ d] \text{ og } \mathbf{b}^T = [0 \ 1 \ \frac{\pi}{2} \ 0].$$

- b) To ukjente kan bestemmes direkte av ligning 1 og 3,  $d = 0$  og  $c = \frac{\pi}{2}$ . Da gjenstår to ligninger med to ukjente og vi får  $a = \frac{\pi}{2} - 2$  og  $b = 3 - \pi$ . Dette gir  $p(x) = (\frac{\pi}{2} - 2)x^3 + (3 - \pi)x^2 + \frac{\pi}{2}x$ .

#### Fasit til 4.6.15

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Hermite  $x^4, x \in [0, 1]$   $(x^4)' = 4x^3$ ,  $x_0 = 0$ ,  $x_1 = 1$ ,  $y_0 = 0$ ,  $y_1 = 1$ ,  $y'_0 = 0$ ,  $y'_1 = 4 \Rightarrow q = -1$ ,  $c = 2$ ,  $p(x) = 2x^3 - x^2$ .

## A.5

### Fasit til 5.7.2

Ligningene 4.10 gir i hvert tilfelle samme matrise

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

med

$$\mathbf{b}_1 = [1 \ 0 \ -3 \ 0]^T \quad \mathbf{b}_2 = [0 \ 0 \ 3 \ 0]^T \quad \mathbf{b}_3 = [0 \ 0 \ -3 \ 0]^T \quad \mathbf{b}_4 = [0 \ 1 \ 0 \ 3]^T$$

$$\mathbf{x}_1 = [-1 \ 3 \ -3 \ 1]^T \quad \mathbf{x}_2 = [3 \ -6 \ 3 \ 0]^T \quad \mathbf{x}_3 = [-3 \ 3 \ 0 \ 0]^T \quad \mathbf{x}_4 = [1 \ 0 \ 0 \ 0]^T$$

Dette gir

$$\begin{aligned} b_0(t) &= -t^3 + 3t^2 - 3t + 1 \\ b_1(t) &= 3t^3 - 6t^2 + 3t \\ b_2(t) &= -3t^3 + 3t^2 \\ b_3(t) &= t^3 \end{aligned}$$

### Fasit til 5.7.3

Kontrollpunkter  $P_0(1, 1)$ ,  $P_1(2, 3)$  og  $P_2(3, 0)$ . Vi skal regne ut  $\mathbf{p}(t) = (x(t), y(t))$  og skissere Bezierkurven.

$$\mathbf{p}(t) = (1, 1) \cdot (t^2 - 2t + 1) + (2, 3) \cdot (-2t^2 + 2t) + (3, 0) \cdot t^2$$

$\mathbf{p}(t) = (x(t), y(t))$  og vi regner ut  $x(t)$  og  $y(t)$  hver for seg:

$$\begin{aligned} x(t) &= 1 \cdot (t^2 - 2t + 1) + 2 \cdot (-2t^2 + 2t) + 3 \cdot t^2 = 2t + 1 \\ y(t) &= 1 \cdot (t^2 - 2t + 1) + 3 \cdot (-2t^2 + 2t) + 0 \cdot t^2 = -5t^2 + 4t + 1 \end{aligned}$$

Innsetting:  $t = 0$  gir  $(1, 1)$ ,  $t = \frac{1}{2}$  gir  $(2, \frac{7}{4})$  og  $t = 1$  gir  $(3, 0)$ . Når vi vet at kurven tangerer kontrollpolygonet i endepunktene, er det mulig å skissere kurven med tre punkter.

### Fasit til 5.7.4

Kontrollpunkter  $P_0(1, 0)$ ,  $P_1(2, 2)$  og  $P_2(4, 1)$ . Vi skal regne ut  $\mathbf{p}(t) = (x(t), y(t))$  og skissere Bezierkurven. Vi regner ut  $x(t)$  og  $y(t)$  hver for seg:

$$\begin{aligned} x(t) &= 1 \cdot (t^2 - 2t + 1) + 2 \cdot (-2t^2 + 2t) + 4 \cdot t^2 = t^2 + 2t + 1 \\ y(t) &= 0 \cdot (t^2 - 2t + 1) + 2 \cdot (-2t^2 + 2t) + 1 \cdot t^2 = -3t^2 + 4t \end{aligned}$$

Innsetting:  $t = 0$  gir  $(1, 0)$ ,  $t = \frac{1}{2}$  gir  $(\frac{9}{4}, \frac{5}{4})$  og  $t = 1$  gir  $(4, 1)$ .

**Fasit til 5.7.5**

Gitt tre kontrollpunkter  $c_0(4, 1)$ ,  $c_1(1, 2)$  og  $c_2(3, 3)$ . Vi skal regne ut en kvadratisk parametrisk kurve som interpolerer disse punktene ved å bruke (5.17) (eller matriseformen).

$$p(t) = \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot (4, 1) + (-t^2 + 2t) \cdot (1, 2) + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot (3, 3)$$

Vi har  $p(t) = (x(t), y(t))$  og får et uttrykk for  $x(t)$  og et uttrykk for  $y(t)$ .

$$\begin{aligned} x(t) &= \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot 4 + (-t^2 + 2t) \cdot 1 + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot 3 = \frac{5}{2}t^2 - \frac{11}{2}t + 4 \\ y(t) &= \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot 1 + (-t^2 + 2t) \cdot 2 + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot 3 = t + 1 \end{aligned}$$

Altså

$$p(t) = (x(t), y(t)) = \left(\frac{5}{2}t^2 - \frac{11}{2}t + 4, -t + 1\right)$$

Kontroll: Innsetting av  $t$ -verdier gir  $p(0) = (4, 1)$ ,  $p(1) = (1, 2)$  og  $p(2) = (3, 3)$ .

**Fasit til 5.7.6**

Kontrollpunkter  $c_0(1, 0)$ ,  $c_1(2, 2)$  og  $c_2(4, 1)$ .

$$p(t) = \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot (1, 0) + (-t^2 + 2t) \cdot (2, 2) + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot (4, 1)$$

Vi har  $p(t) = (x(t), y(t))$  og får et uttrykk for  $x(t)$  og et uttrykk for  $y(t)$ .

$$\begin{aligned} x(t) &= \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot 1 + (-t^2 + 2t) \cdot 2 + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot 4 = \frac{1}{2}t^2 + \frac{1}{2}t + 1 \\ y(t) &= \left(\frac{1}{2}t^2 - \frac{3}{2}t + 1\right) \cdot 0 + (-t^2 + 2t) \cdot 2 + \left(\frac{1}{2}t^2 - \frac{1}{2}t\right) \cdot 1 = -\frac{3}{2}t^2 + \frac{7}{2}t \end{aligned}$$

Kontroll: Innsetting av  $t$ -verdier gir  $p(0) = (1, 0)$ ,  $p(1) = (2, 2)$  og  $p(2) = (4, 1)$ .

## A.6

### Fasit til 5.7.7

Vi har 5 skjøter, og hver B-spline basisfunksjon av grad  $d$  går over  $d+1$  skjøter. Vi får tre lineære, to kvadratiske og en kubisk B-spline basisfunksjon. De lineære basisfunksjonene kan vi finne ved å bruke rekursjonsformelen eller ved å tegne opp og lese av grafisk. Vi får

#### Lineære basisfunksjoner

$$\begin{aligned}B_{0,1} &= 2xB_{0,0} + (2-2x)B_{1,0} \\B_{1,1} &= (2x-1)B_{1,0} + (3-2x)B_{2,0} \\B_{2,1} &= (2x-2)B_{2,0} + (4-2x)B_{3,0}\end{aligned}$$

Notasjonen med de konstante b-spline basisfunksjonene brukes her til å angi hvilket intervall funksjonens første og andre del er definert over.  $B_{i,0}$  notasjonen betyr at funksjonen er definert på det halvåpne intervallet  $[t_i, t_{i+1})$ . Vi ser for eksempel at  $B_{1,0}$  inngår i både første og andre lineære funksjon. På intervallet  $[t_1, t_2) = [\frac{1}{2}, 1)$  fins to lineære b-spline basisfunksjoner.

#### Kvadratiske basisfunksjoner

$$\begin{aligned}B_{0,2} &= w_{0,2}B_{0,1} + (1-w_{1,2})B_{1,1} = \\&= x(2xB_{0,0} + (2-2x)B_{1,0}) + (\frac{3}{2}-x)((2x-1)B_{1,0} + (3-2x)B_{2,0}) = \\&= 2x^2B_{0,0} + (2x-2x^2)B_{1,0} + (-2x^2+4x-\frac{3}{2})B_{1,0} + (2x^2-6x+\frac{9}{2})B_{2,0} = \\&= 2x^2B_{0,0} + (-4x^2+6x-\frac{3}{2})B_{1,0} + (2x^2-6x+\frac{9}{2})B_{2,0}.\end{aligned}$$

$$\begin{aligned}B_{1,2} &= w_{1,2}B_{1,1} + (1-w_{2,2})B_{2,1} = \\&= (x-\frac{1}{2})((2x-1)B_{1,0} + (3-2x)B_{2,0}) + (2-x)((2x-2)B_{2,0} + (4-2x)B_{3,0}) = \\&= (2x^2-2x+\frac{1}{2})B_{1,0} + (-4x^2+10x-\frac{11}{2})B_{2,0} + (2x^2-8x+8)B_{3,0}.\end{aligned}$$

### Fasit til 5.7.8

Vi har  $d = 1, 2$  og  $\mathbf{t} = \{0, 0, 0, 1, 1, 1\}$ .

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

### Fasit til 5.7.9

Her er  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$ . Vi får

$$\begin{aligned}B_{0,0} &= B_{1,0} = 0 \\B_{2,0} &= 1, \quad 0 \leq t < 1 \\B_{3,0} &= 1, \quad 1 \leq t < 2 \\B_{4,0} &= B_{5,0} = 0\end{aligned}$$



De lineære B-spline basisfunksjonene blir

$$B_{0,1} = 0$$

$$B_{1,1} = 0 + (1 - x)B_{2,0}$$

$$B_{2,1} = xB_{2,0} + (2 - x)B_{3,0}$$

$$B_{3,1} = (x - 1)B_{3,0} + 0$$

og de kvadratiske:

$$B_{0,2}(x) = 0 + (1 - w_{1,2})B_{1,1} = (1 - \frac{x-0}{1-0})(1-x)B_{2,0} = (1-x)^2 B_{2,0}$$

$$\begin{aligned} B_{1,2}(x) &= w_{1,2}B_{1,1} + (1 - w_{2,2})B_{2,1} \\ &= \frac{x-t_1}{t_3-t_1}B_{1,1} + (1 - \frac{x-t_2}{t_4-t_2})B_{2,1} \\ &= \frac{x-0}{1-0}B_{1,1} + (1 - \frac{x-0}{2-0})B_{2,1} \\ &= xB_{1,1} + (1 - \frac{x}{2})B_{2,1} \\ &= x(1-x)B_{2,0} + (1 - \frac{x}{2})(xB_{2,0} + (2-x)B_{3,0}) \\ &= (x-x^2)B_{2,0} + (x - \frac{x^2}{2})B_{2,0} + (1 - \frac{x}{2})(2-x)B_{3,0} \\ &= (x-x^2)B_{2,0} + (x - \frac{x^2}{2})B_{2,0} + (1 - \frac{x}{2})(2-x)B_{3,0} \\ &= (2x - \frac{3}{2}x^2)B_{2,0} + (2 - 2x + \frac{x^2}{2})B_{3,0} \end{aligned}$$

$$\begin{aligned} B_{2,2}(x) &= w_{2,2}B_{2,1} + (1 - w_{3,2})B_{3,1} \\ &= \frac{x-t_2}{t_4-t_2}B_{2,1} + (1 - \frac{x-t_3}{t_5-t_3})B_{3,1} \\ &= \frac{x}{2}B_{2,1} + (2-x)B_{3,1} \\ &= \frac{x}{2}(xB_{2,0} + (2-x)B_{3,0}) + (2-x)(x-1)B_{3,0} \\ &= \frac{x^2}{2}B_{2,0} + (x - \frac{x^2}{2})B_{3,0} + (-x^2 + 3x - 2)B_{3,0} \\ &= \frac{x^2}{2}B_{2,0} + (-\frac{3}{2}x^2 + 4x - 2)B_{3,0} \end{aligned}$$

$$B_{3,2}(x) = w_{3,2}B_{3,1} = \frac{x-t_3}{t_5-t_3}B_{3,1} = (x-1)^2 B_{3,0}$$

Basisfunksjonene summerer seg til 1 på hvert skjøtintervall.

### Fasit til 5.7.10

Vi har  $d=2$ ,  $\mathbf{t} = \{0, 0, 0, 1, 2, 3, 3, 3\}$ .

Vi har grad 2 og 3 skjøter i hver ende, så de første B-spline basisfunksjonene blir som i 5.7.9.

$$\begin{aligned}
B_{0,1}(x) &= 0 \\
B_{1,1}(x) &= 0 + (1-x)B_{2,0} \\
B_{2,1}(x) &= xB_{2,0} + (2-x)B_{3,0} \\
B_{3,1}(x) &= (x-1)B_{3,0} + (3-x)B_{4,0} \\
B_{4,1}(x) &= (x-2)B_{4,0} + 0 \\
B_{5,1}(x) &= 0
\end{aligned}$$

Videre trenger vi for å regne ut de kvadratiske B-spline basisfunksjonene

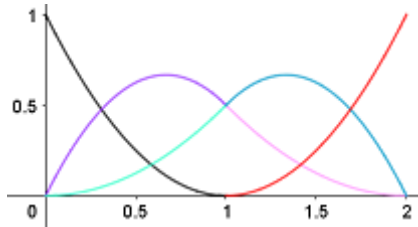
$$\begin{aligned}
w_{1,2}(x) &= \frac{x-t_1}{t_3-t_1} = \frac{x-0}{1} = x \\
w_{2,2}(x) &= \frac{x-t_2}{t_4-t_2} = \frac{x-0}{2} = \frac{x}{2} \\
w_{3,2}(x) &= \frac{x-t_3}{t_5-t_3} = \frac{x-1}{2} \\
w_{4,2}(x) &= \frac{x-t_4}{t_6-t_4} = x-2
\end{aligned}$$

Siden de fem første skjøtene er som i oppgave 5.7.9, blir de to første kvadratiske basisfunksjonene de samme:

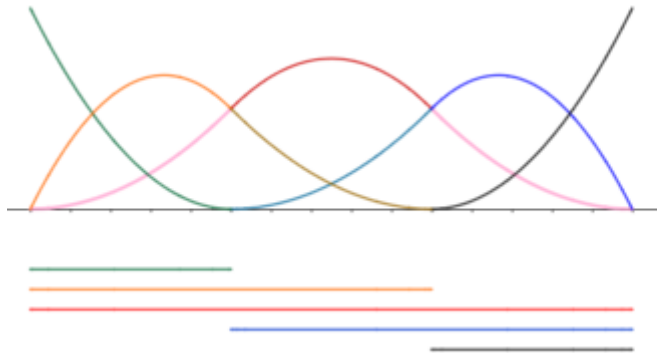
$$\begin{aligned}
B_{0,2}(x) &= (1-x)^2 B_{2,0} \\
B_{1,2}(x) &= (2x - \frac{3}{2}x^2)B_{2,0} + (2 - 2x + \frac{x^2}{2})B_{3,0}
\end{aligned}$$

Videre

$$\begin{aligned}
B_{2,2}(x) &= \frac{1}{2}x^2 B_{2,0} + (-x^2 + 3x - \frac{3}{2})B_{3,0} + \frac{1}{2}(3-x)^2 B_{4,0} \\
B_{3,2}(x) &= \frac{x-1}{2}B_{3,1} + (1 - \frac{x-2}{1})B_{4,1} \\
&= \frac{(x-1)^2}{2}B_{3,0} + (2x - \frac{x^2}{2} - \frac{3}{2})B_{4,0} + (5x - x^2 - 6)B_{4,0} \\
&= \frac{(x-1)^2}{2}B_{3,0} + (7x - \frac{3}{2}x^2 - \frac{15}{2})B_{4,0} \\
B_{4,2}(x) &= (x-2)^2 B_{4,0}
\end{aligned}$$



Figur A.2: Kvadratiske B-spline basisfunksjoner  $B_{i,2}(x)$  med  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$ .



Figur A.3: Kvadratiske B-spline basisfunksjoner  $B_{i,2}(x)$  med  $\mathbf{t} = \{0, 0, 0, 1, 2, 3, 3, 3\}$ .

#### Fasit til 5.7.11

Vi starter med å sette opp uttrykkene for  $w_{i,d}(x)$  og  $1 - w_{i,d}(x)$ :

$$\begin{aligned} w_{i,d}(x) &= \frac{x - t_i}{t_{i+d} - t_i} = \frac{x - i}{d} \\ 1 - w_{i,d}(x) &= 1 - \frac{x - t_i}{t_{i+d} - t_i} = \frac{i + d - x}{d} \end{aligned}$$

Med  $\mathbf{t} = \{0, 1, 2, 3, 4, 5\}$  får vi

$$\begin{aligned} B_{0,1} &= (x - 0)B_{0,0} + (2 - x)B_{1,0} \\ B_{1,1} &= (x - 1)B_{1,0} + (3 - x)B_{2,0} \\ B_{2,1} &= (x - 2)B_{2,0} + (4 - x)B_{3,0} \\ B_{3,1} &= (x - 3)B_{3,0} + (5 - x)B_{4,0} \end{aligned}$$

Videre

$$\begin{aligned}
B_{0,2} &= \frac{x-0}{2}B_{0,1} + \frac{3-x}{2}B_{1,1} \\
&= \frac{1}{2}(x-0)^2B_{0,0} + \frac{1}{2}(x-0)(2-x)B_{1,0} + \frac{1}{2}(3-x)(x-1)B_{1,0} + \frac{1}{2}(3-x)^2B_{2,0} \\
&= \frac{1}{2}(x-0)^2B_{0,0} + (-x^2 + 3x - \frac{3}{2})B_{1,0} + \frac{1}{2}(3-x)^2B_{2,0} \\
B_{1,2} &= \frac{x-1}{2}B_{1,1} + \frac{4-x}{2}B_{2,1} \\
&= \frac{1}{2}(x-1)^2B_{1,0} + \frac{1}{2}(x-1)(3-x)B_{2,0} + \frac{1}{2}(4-x)(x-2)B_{2,0} + \frac{1}{2}(4-x)^2B_{3,0} \\
&= \frac{1}{2}(x-1)^2B_{1,0} + (-x^2 + 5x - \frac{11}{2})B_{2,0} + \frac{1}{2}(4-x)^2B_{3,0} \\
B_{2,2} &= \frac{x-2}{2}B_{2,1} + \frac{5-x}{2}B_{3,1} \\
&= \frac{1}{2}(x-2)^2B_{2,0} + \frac{1}{2}(x-2)(4-x)B_{3,0} + \frac{1}{2}(5-x)(x-3)B_{3,0} + \frac{1}{2}(5-x)^2B_{4,0} \\
&= \frac{1}{2}(x-2)^2B_{2,0} + (-x^2 + 7x - \frac{23}{2})B_{3,0} + \frac{1}{2}(5-x)^2B_{4,0}
\end{aligned}$$

og

$$\begin{aligned}
B_{0,3} &= \frac{x-0}{3}B_{0,2} + \frac{4-x}{3}B_{1,2} \\
B_{1,3} &= \frac{x-1}{3}B_{1,2} + \frac{5-x}{3}B_{2,2}
\end{aligned}$$

Dette gir

$$\begin{aligned}
B_{0,3} &= \frac{x-0}{3}B_{0,2} + \frac{4-x}{3}B_{1,2} \\
&= \frac{1}{3}(x-0) \left( \frac{1}{2}(x-0)^2B_{0,0} + \frac{1}{2}(x-0)(2-x)B_{1,0} + \frac{1}{2}(3-x)(x-1)B_{1,0} + \frac{1}{2}(3-x)^2B_{2,0} \right) \\
&\quad + \frac{1}{3}(4-x) \left( \frac{1}{2}(x-2)^2B_{2,0} + \frac{1}{2}(x-2)(4-x)B_{3,0} + \frac{1}{2}(5-x)(x-3)B_{3,0} + \frac{1}{2}(5-x)^2B_{4,0} \right) \\
&= \frac{x^3}{6}B_{0,0} + (-\frac{x^3}{2} + 2x^2 - 2x + \frac{2}{3})B_{1,0} \\
&\quad + (\frac{x^3}{2} - 4x^2 + 10x - \frac{22}{3})B_{2,0} + (-\frac{1}{6}x^3 + 2x^2 - 8x + \frac{32}{3})B_{3,0}
\end{aligned}$$

### Fasit til 5.7.12

$n + d + 1 = 4$ . Hver basisfunksjon går over  $d + 1$  skjøter, altså har vi to lineære basisfunksjoner og en kvadratisk basisfunksjon.

$$\begin{aligned}
w_{0,1}(t) &= t \\
w_{1,1}(t) &= 0 \\
w_{2,1}(t) &= \frac{t-1}{2} \\
B_{0,1}(t) &= tB_{0,0}(t) + (1 - w_{1,1}(t))B_{1,0}(t) = tB_{0,0}(t) \\
B_{1,1}(t) &= \frac{3-t}{2}B_{2,0}(t)
\end{aligned}$$

$$\begin{aligned}
w_{0,2}(t) &= \frac{t-t_0}{t_2-t_0} = t \\
w_{1,2}(t) &= \frac{t-t_1}{t_3-t_1} = \frac{t-1}{2} \\
B_{0,2}(t) &= w_{0,2}B_{0,1}(t) + (1-w_{1,2})B_{1,1}(t) \\
&= t \cdot B_{0,1}(t) + (1-\frac{t-1}{2})B_{1,1}(t) \\
&= t^2B_{0,0}(t) + (\frac{3-t}{2})^2B_{2,0}(t) \\
&= t^2B_{0,0}(t) + (\frac{1}{4}t^2 - \frac{3}{2}t + \frac{9}{4})B_{2,0}(t)
\end{aligned}$$

### Fasit til 5.7.13

Vi har  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$  og fra oppgave 5.7.9 de kvadratiske B-spline basis-funksjonene

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

Vi deriverer de kvadratiske uttrykkene og får

$$\begin{aligned}
B'_{0,2}(t) &= 2(t-1)B_{2,0} \\
B'_{1,2}(t) &= (-3t+2)B_{2,0} + (t-2)B_{3,0} \\
B'_{2,2}(t) &= tB_{2,0} + (-3t+4)B_{3,0} \\
B'_{3,2}(t) &= 2(t-1)B_{3,0}
\end{aligned}$$

## A.7

**Fasit til ??**

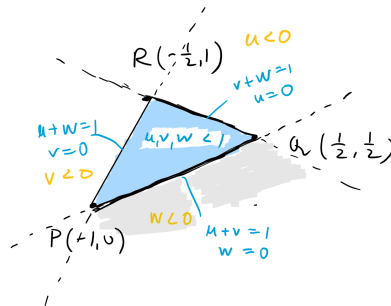
Setter  $P=(-1,0)$ ,  $Q = (\frac{1}{2}, \frac{1}{2})$  og  $R = (-\frac{1}{2}, 1)$ . Vi får  $\vec{u} = Q - P = [\frac{3}{2}, \frac{1}{2}]$  og  $\vec{v} = R - P = [\frac{1}{2}, 1]$  som gir  $A = \vec{u} \wedge \vec{v} = \frac{5}{4}$ .

**a)** Setter  $X = (-\frac{1}{4}, \frac{1}{2})$  og kaller de barysentriske koordinatene  $u, v, w$ .

$$\begin{aligned} u &= (Q - X) \wedge (R - X) / A = [\frac{3}{4}, 0] \wedge [-\frac{1}{4}, \frac{1}{2}] / A = \frac{3}{8} \cdot \frac{4}{5} = \frac{3}{10} \\ v &= (R - X) \wedge (P - X) / A = [-\frac{1}{4}, \frac{1}{2}] \wedge [-\frac{3}{4}, -\frac{1}{2}] / A = \frac{4}{8} \cdot \frac{4}{5} = \frac{4}{10} \\ w &= (P - X) \wedge (Q - X) / A = [-\frac{3}{4}, -\frac{1}{2}] \wedge [\frac{3}{4}, 0] / A = \frac{3}{8} \cdot \frac{4}{5} = \frac{3}{10} \end{aligned}$$

**b)** Setter nå  $X(1, 1)$  og kaller de barysentriske koordinatene  $u, v, w$ .

$$\begin{aligned} u &= (Q - X) \wedge (R - X) / A = [-\frac{1}{2}, -\frac{1}{2}] \wedge [-\frac{3}{2}, 0] / A = -\frac{3}{5} \\ v &= (R - X) \wedge (P - X) / A = [-\frac{3}{2}, 0] \wedge [-2, -1] / A = \frac{6}{5} \\ w &= (P - X) \wedge (Q - X) / A = [-2, -1] \wedge [-\frac{1}{2}, -\frac{1}{2}] / A = \frac{2}{5} \end{aligned}$$



Figur A.4: Oppgave ?? c) hvordan fortegn på de barysentriske koordinatene forteller noe om plasseringen til et punkt i forhold til det aktuelle trianglet.

**Fasit til ??**

Setter  $P=(0,1)$ ,  $Q = (2,0)$  og  $R = (4,3)$ . Vi får  $\vec{u} = Q - P = [2, -1]$  og  $\vec{v} = R - P = [4, 2]$  som gir  $A = |\vec{u} \wedge \vec{v}| = 8$ .

**a)** Setter  $X = (2, 1)$  og kaller de barysentriske koordinatene  $u, v, w$ .

$$\begin{aligned} u &= (Q - X) \wedge (R - X)/A = [0, -1] \wedge [2, 2]/A = \frac{2}{8} \\ v &= (R - X) \wedge (P - X)/A = [2, 2] \wedge [-2, 0]/A = \frac{4}{8} \\ w &= (P - X) \wedge (Q - X)/A = [-2, 0] \wedge [0, -1]/A = \frac{2}{8} \end{aligned}$$

**b)** Setter nå  $Y(3, 4)$  og kaller de barysentriske koordinatene  $u, v, w$ .

$$\begin{aligned} u &= (Q - Y) \wedge (R - Y)/A = [-1, -4] \wedge [1, -1]/A = \frac{5}{8} \\ v &= (R - Y) \wedge (P - Y)/A = [1, -1] \wedge [-3, -3]/A = -\frac{6}{8} \\ w &= (P - Y) \wedge (Q - Y)/A = [-3, -3] \wedge [-1, -4]/A = \frac{9}{8} \end{aligned}$$

**c)** Fortegn på de barysentriske koordinatene forteller noe om plasseringen til et punkt i forhold til det aktuelle triangel. Hvis alle barysentriske koordinater er positive, ligger punktet i trekanten. Hvis en av de barysentriske koordinatene er 0, ligger punktet på en kant, og hvis en av de barysentriske koordinatene er negativ, ligger punktet utenfor trekanten. En skisse som i forrige oppgave vil supplere forklaringer.

## A.8

### Fasit til 7.8.6

Bruker notasjonen  $\mathbf{c}_0, \mathbf{c}_1 \dots$

$$\begin{aligned}\mathbf{c}_{00} &= \frac{1}{2}(4, 0) + \frac{1}{2}(0, 0) = (2, 0) \\ \mathbf{c}_{10} &= \frac{1}{2}(0, 0) + \frac{1}{2}(0, 4) = (0, 2) \\ \mathbf{c}_{20} &= \frac{1}{2}(0, 4) + \frac{1}{2}(4, 4) = (2, 4) \\ \mathbf{c}_{01} &= \frac{1}{2}(2, 0) + \frac{1}{2}(0, 2) = (1, 1) \\ \mathbf{c}_{11} &= \frac{1}{2}(0, 2) + \frac{1}{2}(2, 4) = (1, 3) \\ \mathbf{c}_{20} &= \frac{1}{2}(1, 1) + \frac{1}{2}(1, 3) = (1, 2)\end{aligned}$$

$$\begin{aligned}\mathbf{c}_{00} &= \frac{1}{4}(4, 0) + \frac{3}{4}(0, 0) = (1, 0) \\ \mathbf{c}_{10} &= \frac{1}{4}(0, 0) + \frac{3}{4}(0, 4) = (0, 3) \\ \mathbf{c}_{20} &= \frac{1}{4}(0, 4) + \frac{3}{4}(4, 4) = (3, 4) \\ \mathbf{c}_{01} &= \frac{1}{4}(1, 0) + \frac{3}{4}(0, 3) = (\frac{1}{4}, \frac{9}{4}) \\ \mathbf{c}_{11} &= \frac{1}{4}(0, 3) + \frac{3}{4}(3, 4) = (\frac{9}{4}, \frac{15}{4}) \\ \mathbf{c}_{20} &= \frac{1}{4}(\frac{1}{4}, \frac{9}{4}) + \frac{3}{4}(\frac{9}{4}, \frac{15}{4}) = (\frac{14}{8}, \frac{27}{8})\end{aligned}$$

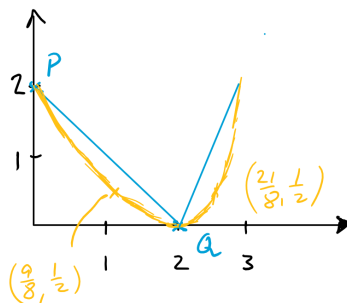
### Fasit til 7.8.7

Bruker notasjonen  $\mathbf{c}_0, \mathbf{c}_1 \dots$

$$\begin{aligned}\mathbf{c}_{00} &= \frac{1}{2}(4, 3) + \frac{1}{2}(0, 1) = (2, 2) \\ \mathbf{c}_{10} &= \frac{1}{2}(0, 1) + \frac{1}{2}(2, 0) = (1, \frac{1}{2}) \\ \mathbf{c}_{20} &= \frac{1}{2}(2, 0) + \frac{1}{2}(3, 4) = (\frac{5}{2}, 2) \\ \mathbf{c}_{01} &= \frac{1}{2}(2, 2) + \frac{1}{2}(1, \frac{1}{2}) = (\frac{3}{2}, \frac{5}{4}) \\ \mathbf{c}_{11} &= \frac{1}{2}(1, \frac{1}{2}) + \frac{1}{2}(\frac{5}{2}, 2) = (\frac{7}{4}, \frac{5}{4}) \\ \mathbf{c}_{20} &= \frac{1}{2}(\frac{3}{2}, \frac{5}{4}) + \frac{1}{2}(\frac{7}{4}, \frac{5}{4}) = (\frac{13}{8}, \frac{5}{4})\end{aligned}$$



### Fasit til 7.8.11



Figur A.5: Oppgave ?? c) hvordan fortegn på de barysentriske koordinatene forteller noe om plasseringen til et punkt i forhold til det aktuelle trianlet.

### Fasit til 7.8.12

Vi har basisfunksjonene fra oppgave 5.7.8 og får

$$\begin{aligned} \mathbf{p}(t) &= \mathbf{c}_0 B_{0,2}(t) + \mathbf{c}_1 B_{1,2}(t) + \mathbf{c}_2 B_{2,2}(t) \\ \mathbf{p}(t) &= \mathbf{c}_0 (1-t)^2 + \mathbf{c}_1 2t(1-t) + \mathbf{c}_2 t^2 \end{aligned}$$

Dette er en parametrisk kurve. For hver  $t$ -verdi regner vi ut både  $x(t)$  og  $y(t)$  ved å bruke  $x$ -koordinatene og  $y$ -koordinatene i  $\mathbf{c}$ .

For eksempel er for  $t = \frac{1}{2}$

$$x = 0 \cdot (1 - \frac{1}{2})^2 + 0 \cdot 2 \cdot \frac{1}{2} \cdot (1 - \frac{1}{2}) + 1 \cdot (\frac{1}{2})^2 = \frac{1}{4}$$

Her får vi

$$\begin{aligned} x(t) &= 0 \cdot (1-t)^2 + 0 \cdot 2 \cdot t \cdot (1-t) + 1 \cdot t^2 = t^2 \\ y(t) &= 1 \cdot (1-t)^2 + 0 \cdot 2 \cdot t \cdot (1-t) + 0 \cdot t^2 = (1-t)^2 \end{aligned}$$

### Fasit til 7.8.14

Her kan vi bruke deBoor's algoritme eller (5.20) som blir

$$\mathbf{f}(t) = \mathbf{c}_0 B_{0,2}(t) + \mathbf{c}_1 B_{1,2}(t) + \mathbf{c}_2 B_{2,2}(t)$$

Basisfunksjonene er de samme som i oppgave 5.7.8. Siden skjøtvektoren er  $\{0, 0, 0, 1, 1, 1\}$  vil kurven interpolere endepunktene, slik at  $\mathbf{f}(0) = (1, 0, 0)$  og  $\mathbf{f}(1) = (1, 2, 0)$ .

Det gjenstår kun ett punkt å regne ut,

$$\mathbf{f}(\frac{1}{2}) = (1, 0, 0) \cdot \frac{1}{4} + (1, 1, 2) \cdot \frac{1}{2} + (1, 2, 0) \cdot \frac{1}{4} = (1, 1, 1)$$

Ligning (7.7) med  $d = 2$  blir

$$B'_{i,2}(t) = \frac{2}{t_{i+2} - t_i} B_{i,1}(t) - \frac{2}{t_{i+3} - t_{i+1}} B_{i+1,1}(t)$$

slik at

$$B'_{0,2}(t) = \frac{2}{t_2 - t_0} B_{0,1}(t) - \frac{2}{t_3 - t_1} B_{1,1}(t) = 0 - 2B_{1,1} = 2(t - 1)$$

På samme måte regner vi ut

$$\begin{aligned} B'_{1,2}(t) &= \frac{2}{t_3 - t_1} B_{1,1}(t) - \frac{2}{t_4 - t_2} B_{2,1}(t) \\ &= 2B_{1,1}(t) - 2B_{2,1}(t) \\ &= 2(1 - t) - 2t \\ &= 2 - 4t \end{aligned}$$

og

$$B'_{2,2}(t) = \frac{2}{t_4 - t_2} B_{2,1}(t) - \frac{2}{t_5 - t_3} B_{3,1}(t) = 2t$$

Med skjøtvektoren i denne oppgaven blir de kvadratiske B-spline basisfunksjonene de samme som de kvadratiske Bernstein-basisfunksjonene i ligning 5.7.

Vi bruker nå (7.5) for å beregne de deriverte til splinekurven,

$$\mathbf{f}'(t) = \mathbf{c}_0 B'_{0,2}(t) + \mathbf{c}_1 B'_{1,2}(t) + \mathbf{c}_2 B'_{2,2}(t)$$

og vi får

$$\begin{aligned} \mathbf{f}'(0) &= (1, 0, 0) \cdot (-2) + (1, 1, 2) \cdot 2 + (1, 2, 0) \cdot 0 = [0, 2, 4] \\ \mathbf{f}'\left(\frac{1}{2}\right) &= (1, 0, 0) \cdot (-1) + (1, 1, 2) \cdot 0 + (1, 2, 0) \cdot 1 = [0, 2, 0] \\ \mathbf{f}'(1) &= (1, 0, 0) \cdot 0 + (1, 1, 2) \cdot (-2) + (1, 2, 0) \cdot 2 = [0, 2, -4] \end{aligned}$$

Legg merke til at dette er vektorer.

Med kontrollpunktene (0,0,0), (0,1,0), (0,2,0):

$$\begin{aligned} \mathbf{f}'(0) &= (0, 0, 0) \cdot (-2) + (0, 1, 0) \cdot 2 + (0, 2, 0) \cdot 0 = [0, 2, 0] \\ \mathbf{f}'\left(\frac{1}{2}\right) &= (0, 0, 0) \cdot (-1) + (0, 1, 0) \cdot 0 + (0, 2, 0) \cdot 1 = [0, 2, 0] \\ \mathbf{f}'(1) &= (0, 0, 0) \cdot 0 + (0, 1, 0) \cdot (-2) + (0, 2, 0) \cdot 2 = [0, 2, 0] \end{aligned}$$

### Fasit til 7.8.15

Vi har  $\mathbf{t} = \{0, 0, 0, 1, 2, 2, 2\}$  og fra oppgave 5.7.9 de lineære B-spline basisfunksjonene

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

Når vi bruker ligning (7.7) med  $d = 2$  blir

$$B'_{i,2}(t) = \frac{2}{t_{i+2} - t_i} B_{i,1}(t) - \frac{2}{t_{i+3} - t_{i+1}} B_{i+1,1}(t)$$

slik at de deriverte til basisfunksjonene blir

$$\begin{aligned} B'_{0,2}(t) &= -2B_{1,1}(t) = 2(t-1)B_{2,0} \\ B'_{1,2}(t) &= (-3t+2)B_{2,0} + (t-2)B_{3,0} \\ B'_{2,2}(t) &= tB_{2,0} + (-3t+4)B_{3,0} \\ B'_{3,2}(t) &= 2(t-1)B_{3,0} \end{aligned}$$

Dette er selvsagt samme resultat som i oppgave 5.7.13.

Vi skal beregne den deriverte  $\mathbf{f}'(t)$  i  $t = 0, \frac{1}{2}, 1, \frac{3}{2}, 2$  med kontrollpunktene  $(0, 1, 0)$ ,  $(1, 1, 2)$ ,  $(2, 1, 2)$ , og  $(3, 1, 0)$ .

1.  $B'_{0,2}(0) = -2, B'_{1,2}(0) = 2, B'_{2,2}(0) = 0$ , slik at  
 $\mathbf{f}'(0) = -2(0, 1, 0) + 2(1, 1, 2) = [2, 0, 4]$
2. Vi har  $B'_{0,2}(\frac{1}{2}) = -1, B'_{1,2}(\frac{1}{2}) = \frac{1}{2}$  og  $B'_{2,2}(\frac{1}{2}) = \frac{1}{2}$ .  
 $\mathbf{f}'(\frac{1}{2}) = -(0, 1, 0) + \frac{1}{2}(1, 1, 2) + \frac{1}{2}(2, 1, 2) = [\frac{3}{2}, 0, 2]$
3.  $B'_{1,2}(1) = -1, B'_{2,2}(1) = 1, B'_{3,2}(1) = 0$ .  
 $\mathbf{f}'(1) = -(1, 1, 2) + (2, 1, 2) = [1, 0, 0]$
4.  $B'_{1,2}(\frac{3}{2}) = -\frac{1}{2}, B'_{2,2}(\frac{3}{2}) = -\frac{1}{2}, B'_{3,2}(\frac{3}{2}) = 1$ .  
 $\mathbf{f}'(\frac{3}{2}) = -\frac{1}{2}(1, 1, 2) - \frac{1}{2}(2, 1, 2) + (3, 1, 0) = [\frac{3}{2}, 0, -2]$
5.  $B'_{1,2}(2) = 0, B'_{2,2}(2) = -2, B'_{3,2}(2) = 2$ .  
 $\mathbf{f}'(2) = -2(2, 1, 2) + 2(3, 1, 0) = [2, 0, -4]$

Kontroller gjerne de deriverte til basisfunksjonene med figur A.2.

# Bibliografi

- [1] D. Angel, E. Shreiner. *Interactive Computer Graphics A Top-Down Approach with WebGL Seventh edition*. Pearson, 2015.
- [2] C deBoor. *A Practical Guide to Splines*. Springer, 1978.
- [3] D.E. Edwards, C.H. Penney. *Calculus and analytic geometry*. Prentice-Hall, 1982.
- [4] J Gallier. *Geometric methods and applications fro Computer Science and Engineering*. Springer, 2011.
- [5] R Goldman. *Pyramid Algorithms*. Morgen Kaufman Elsevier, 2003.
- [6] M.G.Gulbrandsen, J.Kleppe, T.A.Kro, J.E.Vatne. *Matematikk for ingeniørfag med numeriske beregninger*. Gyldendal akademisk, 2013.
- [7] Arne Lakså. *Blendingsteknikker i kurve- og flatekonstruksjoner*. Geofo, 2022.
- [8] M.A.Weiss. *Data Structures and Algorithm Analysis i C++ Fourth Edition*. Pearson, 2014.
- [9] D.Nylund, O.E.Flaten. *A dynamically changing spline curve for the motion of a game character*. NIK 2018, 2018.
- [10] W. Rudin. *Principles of Mathematical Analysis Third Edition*. McGraw Hill, 1976.
- [11] L.L Schumaker. *Spline Functions: Basic Theory Third Edition*. Cambridge Mathematical Library, 2007.
- [12] O. Dahl, T. Lyche, R. Winter. *Construction and analysis of numerical methods*. Universitetet i Oslo, 1985.