Matematisk modellering og numeriske metoder

Lektion 14

Morten Grud Rasmussen

9. november 2015

1 Interpolation

[Bogens afsnit 19.3 side 805]

1.1 Interpolationspolynomier

Enhver kontinuert funktion f på et lukket og begrænset interval J kan approksimeres vilkårligt godt af et polynomium p i den forstand, at for enhver tolerance $\varepsilon > 0$, kan vi finde et polynomium p_ε, så der for alle $x \in J$ gælder at $|f(x) - p_\varepsilon(x)| < \varepsilon$. Dette faktum kaldes *Weierstraß' approksimationssætning*, og vi vil blot bruge det som motivation for at approksimere funktioner med polynomier.

Antag nu, at vi har to n'tegradspolynomier p_n og q_n og $n+1$ punkter $x_0, \ldots, x_n$ som opfylder, at $p_n(x_i) = q_n(x_i)$ for $i = 0, \ldots, n$. Så er $p_n - q_n$ (højst) et n'tegradspolynomium med $n+1$ rødder i $x_0, \ldots, x_n$. Men et n'tegradspolynomium kan højst have n rødder (algebraens fundamentalsætning), med mindre det er identisk lig 0, og p_n må altså være lig q_n. Vi har vist følgende sætning:

Sætning 1.1 (Entydighed af approksimerende polynomier). *Hvis p_n og q_n er n'tegradspolynomier med samme værdi i de $n+1$ punkter $x_0, \ldots, x_n$, så er $p_n = q_n$.*

Vi vil i det følgende beskæftige os med at approksimere funktioner med polynomier. Lad f betegne den funktion, vi ønsker at approksimere. Antag, at vi kender funktionens værdi i de $n+1$ punkter $x_0. \ldots, x_n$. Disse punkter, $f(x_0), \ldots, f(x_n)$, vil vi i det følgende benævne $f_0, \ldots, f_n$ hvor altså $f_i = f(x_i)$. Jf. ovenstående Sætning 1.1 kan vi højst finde ét n'tegradspolynomium, som går gennem $f_0, \ldots, f_n$. Det viser sig, at der *altid* eksisterer et sådant polynomium (vi har altså både *eksistens* og *entydighed*).

Vi antager nu for en kort bemærkning, at $x_0 < \cdots < x_i < x_{i+1} < \ldots < x_n$ og at p_n er det entydige n'tegradspolynomium, som opfylder, at $p_n(x_i) = f_i$ for alle $i = 0, \ldots, n$. Idéen med polynomiumsinterpolation er så, at for ethvert $x \in [x_0, x_n]$ er den *interpolerede værdi* $p_n(x)$ et kvalificeret bud på, hvad $f(x)$ er. Det kaldes tilsvarende *ekstrapolation*, såfremt $x < x_0$ eller $x > x_n$, og x altså ikke ligger *imellem* ("inter") x_i'erne, men *udenfor* ("ekstra" – som i *extraterrestial*).

De følgende to afsnit præsenterer to forskellige *konstruktive* metoder til at finde et n'tegradspolynomium p_n, som opfylder, at $p_n(x_i) = f_i$ for alle $i = 0, \dots, n$, og er hver især i sig selv *konstruktive beviser* for eksistensen. Vores primære interesse i de to metoder er dog, at de resulterende polynomier (som pr. entydigheden er identiske) er på så forskellig form rent algebraisk, at de i numerisk sammenhæng er interessante af forskellige årsager.

1.2 Lagrange-interpolation

Idéen i Lagrange-interpolation er lidt i familie med idéen bag basisfunktionerne, vi så i forbindelse med finite element-metoden: Vi finder passende funktioner (her n'tegradspolynomier) L_i, som opfylder, at $L_i(x_j) = \delta_{ij}$, hvor $\delta_{ij} = 0$ for $i \neq j$ og $\delta_{ii} = 1$. Vi kan så konstruere p_n ved blot at sætte:

$$p_n(x) = \sum_{i=0}^{n} f_i L_i(x).$$

Vi tjekker nu, om $p_n(x_j) = f_j$:

$$p_n(x_j) = \sum_{i=0}^{n} f_i L_i(x_j) = \sum_{i=0}^{n} f_i \delta_{ij} = f_1 \cdot 0 + \cdots + f_{j-1} \cdot 0 + f_j \cdot 1 + f_{j+1} \cdot 0 + \cdots + f_n \cdot 0 = f_j$$

som ønsket. Tilbage står konstruktionen af L_i'erne. Hvis $n = 1$, kan vi vælge

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \qquad \text{og} \qquad L_1(x) = \frac{x - x_0}{x_1 - x_0},$$

og hvis $n = 2$, kan vi vælge

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \qquad L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \qquad \text{og} \qquad L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

Den vakse læser vil nu muligvis have gættet mønsteret for et vilkårligt n:

$$L_i(x) = \frac{\ell_i(x)}{\ell_i(x_i)} \qquad \text{hvor} \qquad \ell_i(x) = \prod_{j \neq i} (x - x_j) = (x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n),$$

hvor $\prod$ er et såkaldt *produkttegn*, som svarer til sumtegnet blot med den forskel, at man ganger ting sammen i stedet for at addere dem, og hvor $j \neq i$ indikerer, at produktet er over alle værdier af j som ikke er i, altså $j = 0, \dots, i-1, i+1, \dots, n$. Da produktet således består af n faktorer på formen $(x - x_i)$, vil resultatet ℓ_i være et n'tegradspolynomium. Sætter vi x_j med $j \neq i$ ind i ℓ_i, vil én af faktorerne være 0 og resultatet derfor ligeledes være nul. Sætter vi x_i ind i ℓ_i, så er der derimod tale om et produkt af n tal, som ikke er nul, og resultatet er derfor heller ikke nul, og vi kan da i definitionen af L_i tillade os at dividere med $\ell_i(x_i)$, således at

$$L_i(x_i) = \frac{\ell_i(x_i)}{\ell_i(x_i)} = 1.$$

Vi har hermed vist, at $L_i(x_j) = \delta_{ij}$ som ønsket.

Eksempel 1.2 (Bogens Example 1 side 807, Example 2 side 808 og Example 3 side 809)**.** Vi vil approksimere $\ln(9.2)$ ved interpolation ud fra værdierne af $\ln(9.0)$ og $\ln(9.5)$:

$$p_1(x) = \sum_{i=0}^{1} f_i L_i(x) = \ln(9.0)\frac{x-9.5}{9.0-9.5} + \ln(9.5)\frac{x-9.0}{9.5-9.0} \qquad \text{og dermed} \qquad p_1(9.2) = 2.21885.$$

Vi har dog ikke styr på, hvor præcist dette resultat er. Vi prøver derfor at se, hvad der sker, hvis vi tager et punkt mere med i interpolationen, eksempelvis $\ln(11.0)$. Da er

$$\begin{aligned} p_2(x) &= \sum_{i=0}^{2} f_i L_i(x) \\ &= \ln(9.0)\frac{(x-9.5)(x-11.0)}{(9.0-9.5)(9.0-11.0)} + \ln(9.5)\frac{(x-9.0)(x-11.0)}{(9.5-9.0)(9.5-11.0)} \\ &\quad + \ln(11.0)\frac{(x-9.0)(x-9.5)}{(11.0-9.0)(11.0-9.5)} \\ &= 2.21916. \end{aligned}$$

Forskellen $p_2(9.2)-p_1(9.2) = 0.00031$ er et *estimat* på fejlen. Snyder vi og finder den korrekte værdi af $\ln(9.2)$, vil vi se, at den korrekte fejl er 0.00035, og vores estimat er altså ikke helt i skoven.

Vi vil fremhæve to pointer i forhold til dette eksempel. For det første bemærker vi, at det er næsten som at begynde forfra, når man kender p_n og vil have fat i p_{n+1} (her med $n = 1$), og for det andet noterer vi os vigtigheden af fejlvurderinger i forbindelse med polynomier. Vi vil om lidt se, at der findes en anden metode til at konstruere de interpolerende polynomier, hvor p_n genbruges i udtrykket for p_{n+1}. Hvad angår fejlvurderinger, så kan det vises, at fejlen kan udtrykkes

$$\varepsilon_n(x) = f(x) - p_n(x) = (x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{(n+1)}(t_x)}{(n+1)!} \tag{1}$$

for et passende t_x, som afhænger af x. Når $x \in [x_0, x_n]$ vil $t_x \in [x_0, x_n]$ og hvis $x \notin [x_0, x_n]$, vil $t_x \in [\min(x_0, x), \max(x_n, x)]$. Ved at vælge største og mindste værdi af $f^{(n+1)}$ i intervallet kan vi altså komme med en præcis nedre og øvre grænse for fejlen.[1]

Eksempel 1.3 (Example 3 side 809 fortsat)**.** Der findes et $t_{9.2}$ så fejlen fra sidste eksempel kan skrives

$$\varepsilon_1(9.2) = (9.2-9.0)(9.2-9.5)\frac{\ln''(t_{9.2})}{2!}.$$

Vi vil finde en øvre og nedre grænse for fejlen. Først finder vi $\ln''$:

$$\ln'(x) = \frac{1}{x} \qquad \text{så} \qquad \ln''(x) = -\frac{1}{x^2}.$$

På intervallet $[x_0, x_1] = [9.0, 9.5]$ antager $\ln''$ altså sin minimale og sin maksimale værdi i hhv. 9.0 og 9.5. Derfor må

$$(9.2-9.0)(9.2-9.5)\frac{-\frac{1}{9.0^2}}{2} = 0.00033 \leq \varepsilon_1(9.2) \leq 0.00037 = (9.2-9.0)(9.2-9.5)\frac{-\frac{1}{9.5^2}}{2}.$$

I dette tilfælde giver (1) altså et ret godt indtryk af fejlens størrelse.

[1]"Præcis" i den forstand, at fejlen nødvendigvis må ligge mellem den øvre og nedre grænse. Der kan imidlertid sagtens være stor afstand mellem øvre og nedre grænse, så vi bliver altså ikke nødvendigvis særligt meget klogere på fejlens størrelse.

I tilfælde, hvor (1) giver et stort spænd mellem øvre og nedre grænse, hvor de afledte af f i sig selv er svære at udregne, eller hvor f's afledte simpelthen er ukendte, er det stadig en god idé at sammenligne p_n med p_{n+1}, når man skal estimere fejlen.

1.3 Newtons generelle divideret differens-metode

Som sagt findes der netop ét n'tegradspolynomium p_n som går gennem $n+1$ givne datapunkter (x_i, f_i), $i = 0, \dots, n$, og i Lagrange-metoden skulle man begynde forfra, hvis man pludselig fik et datapunkt mere, og derfor mulighed for at vælge et $(n+1)$'stegradspolynomium i stedet. I Newtons divideret differens-metode søger man i stedet $g_n = p_n - p_{n-1}$ således, at p_n kan skrives

$$p_n = \sum_{i=0}^{n} g_i, \tag{2}$$

hvor g_i er et i'tegradspolynomium, som går gennem det i'te punkt (x_i, f_i) men er 0 i (x_j, f_j) for $j < i$. Altså må g_i kunne skrives som

$$g_i(x) = f[x_0, \dots, x_i](x - x_0) \cdots (x - x_{i-1}) = f[x_0, \dots, x_i] \prod_{j<i} (x - x_j), \tag{3}$$

Hvor $f[x_0, \dots, x_i]$ er en passende koefficient. Det slemme er nu, at koefficienten er givet ved en fæl, rekursiv[2] formel. Det er klart, at koefficienten (bestemt af punktet (x_0, f_0)) i nultegradspolynomiet $g_0 = p_0 \equiv f[x_0]$ som går gennem (x_0, f_0) må være

$$f[x_0] = f_0. \tag{4}$$

Alle andre koefficienter (bestemt af de $k+1$ punkter (x_i, f_i), $i = 0, \dots, k$) er så rekursivt defineret ved

$$f[x_0, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0}, \tag{5}$$

altså en brøk, hvor tælleren er differensen mellem koefficienter bestemt af de k punkter, som er tilbage, hvis man smider hhv. (x_0, f_0) og (x_k, f_k) væk, og nævneren er differensen $x_k - x_0$. Det overlades til læseren at tjekke, at de påståede egenskaber er opfyldt! Sætter vi (2), (3), (4) og (5) sammen, fås

$$\begin{aligned} p_n(x) &= \sum_{i=0}^{n} f[x_0, \dots, x_i] \prod_{j<i} (x - x_j) \\ &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &\quad + \cdots + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1}). \end{aligned} \tag{6}$$

Vi bemærker, at i rekursionen (5) "æder" man enten det yderste venstre eller det yderste højre x_i. Det betyder, at de eneste koefficienter $f[\cdots]$, vi behøver at kende for at udregne $f[x_0, \dots, x_k]$, er dem på formen $f[x_i, \dots, x_{i+j}]$, hvor $i \geq 0$ og $i + j \leq k$ og ingen x_l mellem x_i og x_{i+j} udelades. Udregningen kan derfor organiseres i følgende skema:

[2]Rekursion er ca. det modsatte af iteration. Ved iteration finder man næste skridt ud fra de skridt, der allerede er taget; i rekursion finder man udtrykkene i skridt n ved at kende udtrykkene fra skridt $n-1$.

i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_{i-1}, x_i, x_{i+1}]$	$f[x_{i-1}, x_i, x_{i+1}, x_{i+2}]$	$\cdots$	$f[x_0, \ldots, x_k]$
0	f_0					
		$\dfrac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f_1 - f_0}{x_1 - x_0}$				
			$\dfrac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} =$			
1	f_1					
			$\dfrac{\frac{f[x_2]-f[x_1]}{x_2-x_1} - \frac{f[x_1]-f[x_0]}{x_1-x_0}}{x_2 - x_0}$			
				$\dfrac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} =$		
		$\dfrac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{f_2 - f_1}{x_2 - x_1}$				
				$\dfrac{\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} - \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}}{x_3-x_0}$		
			$\dfrac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} =$			
2	f_2				$\ddots$	
			$\dfrac{\frac{f[x_3]-f[x_2]}{x_3-x_2} - \frac{f[x_2]-f[x_1]}{x_2-x_1}}{x_3 - x_1}$			
				$\dfrac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1} =$		
		$\dfrac{f[x_3] - f[x_2]}{x_3 - x_2} = \frac{f_3 - f_2}{x_3 - x_2}$				
				$\dfrac{\frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2} - \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}}{x_4-x_1}$		
			$\dfrac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} =$			
3	f_3				$\cdots$	$f[x_0, \ldots, x_k]$
			$\dfrac{\frac{f[x_3]-f[x_2]}{x_3-x_2} - \frac{f[x_2]-f[x_1]}{x_2-x_1}}{x_3 - x_1}$			
				$\dfrac{f[x_3, x_4, x_5] - f[x_2, x_3, x_4]}{x_5 - x_2} =$		
		$\dfrac{f[x_4] - f[x_3]}{x_4 - x_3} = \frac{f_4 - f_3}{x_4 - x_3}$				
				$\dfrac{\frac{f[x_4,x_5]-f[x_3,x_4]}{x_5-x_3} - \frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}}{x_5-x_1}$		
$\vdots$	$\vdots$		$\cdots$			
		$\dfrac{f[x_k] - f[x_{k-1}]}{x_k - x_{k-1}} = \frac{f_k - f_{k-1}}{x_k - x_{k-1}}$				
k	f_k					

Denne tabel er måske i første omgang mere forvirrende end informativ, men pointen er, at første søjle blot er de kendte data, anden søjle er blot den parvise forskel mellem en “underbo” og en “overbo” i første søjle divideret med forskellen på de to tilhørende x_i’er, og generelt udregnes en søjle som forskellen på en “underbo” og en “overbo” i søjlen før divideret med forskellen på de tilhørende x_i’er med hhv. størst og mindst indeks. Skulle det stadig være uklart, regner vi nu et konkret eksempel, som bør sammenlignes med ovenstående.

Eksempel 1.4 (Bogens Example 4 på side 812)**.** Vi fortsætter med eksemplet med $\ln$. Indtil nu har vi haft tre datapunkter, $(9.0, \ln(9.0))$, $(9.5, \ln(9.5))$ og $(11.0, \ln(11.0))$. Vi tilføjer $(8.0, \ln(8.0))$ og sætter $x_0 = 8.0$, $x_1 = 9.0$, $x_2 = 9.5$ og $x_3 = 11.0$. Så ser tabellen ud som følger:[3]

i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_{i-1}, x_i, x_{i+1}]$	$f[x_0, x_1, x_2, x_3, x_4]$
0	$f_0 = \ln(8.0) = 2.07944$			
		$\frac{\ln(9.0)-\ln(8.0)}{9.0-8.0} = 0.117783$		
1	$f_1 = \ln(9.0)$		$\frac{0.108134-0.117783}{9.5-8.0} = -0.00653240$	
		$\frac{\ln(9.5)-\ln(9.0)}{9.5-9.0} = 0.108134$		$\frac{-0.00519940-(-0.00653240)}{11.0-8.0} = 0.000411000$
2	$f_2 = \ln(9.5)$		$\frac{0.0977356-0.108134}{11.0-9.0} = -0.00519940$	
		$\frac{\ln(11.0)-\ln(9.5)}{11.0-9.5} = 0.0977356$		
3	$f_3 = \ln(11.0)$			

Sågiver (6)

$$\begin{aligned} p_3(x) &= \sum_{i=0}^{3} f[x_0, \ldots, x_i] \prod_{j<i} (x - x_j) \\ &= 2.07944 + 0.117783 \cdot (x - 8.0) - 0.00653240 \cdot (x - 8.0)(x - 9.0) \\ &\quad + 0.00041100 \cdot (x - 8.0)(x - 9.0)(x - 9.5), \end{aligned}$$

som giver $p_3(9.2) = 2.21921$. Som bekendt var $p_1(9.2) = 2.21885$ og $p_2(9.2) = 2.21916$ mens $\ln(9.2) = 2.21920$, og vi ser, at præcisionen stiger med graden af polynomiet/antallet af målepunkter.

1.4 Newtons *forward difference*-formel

Hvis datapunkterne er indsamlet med jævne mellemrum i den forstand, at $x_{i+1} - x_i = h$ for alle $i = 0, \ldots, n-1$, så kan både (5) og (6) simplificeres en del. Vi springer udledningen over og går direkte til formlerne:

$$p_n(x) = \sum_{i=0}^{n} (\Delta^i f_0) \binom{\frac{x - x_0}{h}}{i}, \tag{7}$$

hvor $\Delta^k f_j$ er defineret rekursivt ved $\Delta^k f_j = \Delta^{k-1} f_{j+1} - \Delta^{k-1} f_j$ for $k \geq 1$ og $\Delta^0 f_j = f_j$, mens

$$\binom{r}{s} = \frac{\prod_{i=0}^{s-1} (r - i)}{s!} = \frac{r(r-1)(r-2)\cdots(r-(s-1))}{s \cdot (s-1) \cdots 2 \cdot 1} \quad \text{for} \quad s \in \mathbb{N} \quad \text{og} \quad \binom{r}{0} = 1.$$

[3] Alle tal er angivet med 6 betydende cifre men udregnet med langt højere præcision.

Vi bemærker, at Δ^k her ikke har noget med differentialoperatorer at gøre. Navnet *forward difference* skyldes, at $\frac{1}{h}\Delta f_j$ er den *forward difference*, vi så i forbindelse med finite difference-metoden, mens $\frac{1}{h^k}\Delta^k f_j$ er højereordens differenskvotienter i stil med den centrale andenordens differenskvotient, vi også stiftede bekendtskab med i samme ombæring, og som altså vil gå mod den k'te afledede, hvis h går mod 0.

Da $\Delta^k f_j$ er defineret rekursivt på en måde, som minder meget om koefficienterne $f[\cdots]$ (faktisk er $f[x_0,\ldots,x_k] = \frac{1}{k!h^k}\Delta^k f_0$), så kan $\Delta^k f_j$ også med fordel udregnes vha. en tilsvarende tabel og tilsvarende fremgangsmåde, den største forskel værende at der ikke skal divideres med en forskel mellem x_i'er.

Det kan vises, at for $r = \frac{x-x_0}{h}$ og et $t_x \in [x_0, x_n]$ som afhænger af x, så er

$$\varepsilon_n(x) = f(x) - p_n(x) = \frac{h^{n+1}}{(n+1)!} r(r-1)\cdots(r-n) f^{(n+1)}(t_x)$$

for $x \in [x_0, x_n]$. Vi bemærker, at produktet $r(r-1)\cdots(r-n)$ er mindre jo mere centralt x ligger mellem x_i'erne. Dermed er fejlen også mindre, desto mere centralt x ligger. Det kan endvidere vises, at fejlen $\varepsilon_n(x)$ er af samme størrelsesorden som $g_{n+1}(x)$, som dermed kan bruges som estimat for fejlen.

1.5 Newtons *backward difference*-formel

Holder vi os til faste afstande mellem x_i'erne, findes der oplagt en *backward difference*-pendant til (7). Den ser ud som følgende.

$$p_n(x) = \sum_{i=0}^{n} (\nabla^i f_n) \binom{\frac{x-x_0}{h} + i - 1}{i},$$

hvor $\nabla^k f_j$ er defineret rekursivt ved $\nabla^k f_j = \nabla^{k-1} f_j - \nabla^{k-1} f_{j-1}$ for $k \geq 1$ og $\nabla^0 f_j = f_j$, mens

$$\binom{r}{s} = \frac{\prod_{i=0}^{s-1}(r-i)}{s!} = \frac{r(r-1)(r-2)\cdots(r-(s-1))}{s\cdot(s-1)\cdots 2\cdot 1} \quad \text{for} \quad s \in \mathbb{N} \quad \text{og} \quad \binom{r}{0} = 1$$

som før. Eksempler på brug af Newtons *forward* og *backward diffence*-formler kan findes i bogen på side 814 og 816, hhv.