TANA23 Föreläsning 4

Interpolation

- Interpolationsproblemet. Introduktion.
- Polynominterpolation. Felanalys. Runges fenomen.
- Lagrange's Interpolations metod. Generalisering till $2D$.
- Tillämpning - Effektiv lösning av icke-linjära ekvationer.

Interpolation

Antag att vi har en tabell

x	x_1	x_2	$\dots$	x_{n+1}
$f(x)$	f_1	f_2	$\dots$	f_{n+1}

Hur skall vi uppskatta $f(x)$ för $x_1 \leq x \leq x_{n+1}$?

Kommentar Detta kallas *interpolationsproblemet* och kan generaliseras till både $2D$ och $3D$.

Exempel Vid gjutning av stål är det väldigt viktigt att smältan håller rätt temperatur. För att kunna ta hand om produktionsstörningar låter vi smältan vara $50\,^oC$ varmare än nödvändigt och låter skänken svalna lite innan gjutning. Svalningstiden väljs ut med hjälp av en tabell.

Antag att vi har en tabellen

$t\,[min]$	1	2	3	$\dots$	29	30
$\Delta T\,[^oC]$	-2.5	-4.3	-6.2	$\dots$	-57.3	-58.9

Hur länge skall vi låta skänken svalna om stålsmältan ligger $34.3\,^oC$ över önskad temperatur?

Närmstagranne interpolation

Lösning *Närmstagranne interpolation* innebär att man approximerar $f(x) \approx f(x_i)$, där x_i är det värde i tabellen som ligger närmast x.

Lemma Trunkeringsfelet vid närmaste granne interpolation satisfierar

$$|f(x) - f(x_i)| \leq Ch,$$

där C är en konstant och $h = \max(x_{i+1} - x_i)$ är *steglängden*.

Kommentar Bevisa med hjälp av medelvärdessatsen eller felfortplantningsfomeln. Antar att $f(x)$ är deriverbar.

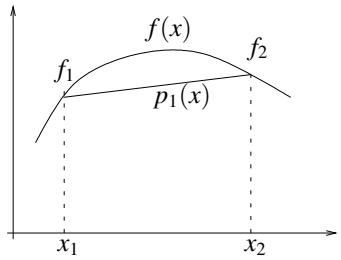
Den absolut vanligaste interpolationsmetoden. Enkel och oftast tillräckligt noggrann om tabellen är bra nog.

Polynominterpolation

Definition Ett polynom $p(x)$ *interpolerar* en funktion $f(x)$ i punkterna $x_1, x_2, \ldots, x_{n+1}$, om

$$p(x_i) = f(x_i), \qquad i = 1, 2, \ldots, n+1.$$

Frågor Polynomets gradtal? Hur skall polynomet beräknas? Feluppskattning?



Sats Det förstagrads polynom $p_1(x)$ som interpolerar *två punkter* (x_1, f_1) och (x_1, f_2) ges av

$$p_1(x) = f_1 + \frac{x - x_1}{x_2 - x_1}(f_2 - f_1).$$

Lemma Låt funktionsvärden f_1 och f_2 ha fel $|\Delta f_i| \leq \varepsilon$. Vid *linjär interpolation* fås då ett fel

$$|R_{XF}| \leq \varepsilon.$$

Bevis Eftersom vi har en *explicit formel* för fallet med linjär interpolation kan vi använda felfortplantningsformeln.

Lemma (*Rolle*) Låt $f(x)$ vara deriverbar på $[a, b]$ och antag $f(a) = f(b)$. Då finns ett $\xi \in (a, b)$ där $f'(\xi) = 0$.

Sats Låt $p_1(x)$ vara det linjära polynom som interpolerar $f(x)$ i punkterna x_1 och x_2. Då gäller

$$R_T = f(x) - p_1(x) = \frac{f''(\xi)}{2}(x - x_1)(x - x_2), \text{ där } x_1 < \xi < x_2.$$

eller

$$|R_T| \leq Ch^2, \text{ där } h = x_2 - x_1.$$

Bygger på att funktionen $f(x)$ har två kontinuerliga derivator. Måste kunna uppskatta $f''(x)$.

Polynom Interpolation

Sats Givet $n+1$ punkter (x_i, f_i) finns ett *unikt* polynom $p_n(x)$, av grad n, som *interpolerar* de givna punkterna, dvs $p_n(x_i) = f_i$, $i = 1, 2, \ldots, n+1$.

Sats Låt $p_n(x)$ vara det polynom av grad n som interpolerar $f(x)$ i punkterna $x_1, x_2, \ldots, x_{n+1}$. Då gäller

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_1) \cdots (x - x_{n+1}).$$

Funktionen $f(x)$ måste ha tillräckligt många kontinuerliga derivator. Hur skall räkningarna organiseras?

Newton's Interpolationspolynom

Hitta ett polynom $p_n(x)$ som interpolerar värdena i tabellen

x	x_1	x_2	$\ldots$	x_{n+1}
$f(x)$	f_1	f_2	$\ldots$	f_{n+1}

Definition Ansatsen

$$\begin{aligned} p_n(x) \quad = \quad & c_0 + c_1(x - x_1) + c_2(x - x_1)(x - x_2) + \ldots \\ & + c_n(x - x_1) \ldots (x - x_n), \end{aligned}$$

kallas *Newton's interpolationspolynom*.

Kommentar Syftet med ansatsen är att den leder till enkla räkningar.

I praktiken känner vi inte $f^{(n+1)}(\xi)$. Istället väljer vi en *extra punkt* (x_{n+2}, f_{n+2}) och beräknar ett nytt interpolerande polynom $p_{n+1}(x)$.

Trunkeringsfelet vid *polynominterpolation* uppskattas

$$R_T(x) \approx P_{n+1}(x) - P_n(x).$$

Kommentar Betyder att $R_T(x_i) = 0$, för $i = 1, 2, \ldots, n+1$. Antagande att felet minskar då n ökar. Används Newtons interpolationsformel är det lätt att genomföra räkningarna.

Exempel Låt

$$f(x) = \mathrm{e}^{-x/2} \cos(x/7) + (x - 0.1)^2/2,$$

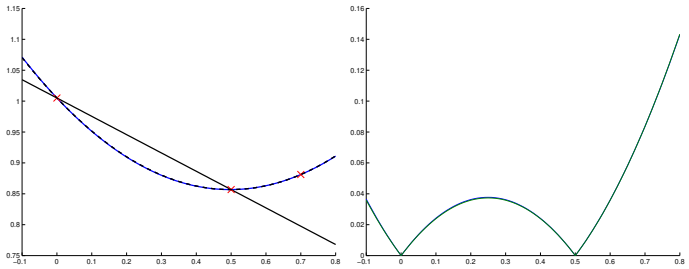
och antag att vi har följande tabell med korrekt avrundade funktionsvärden:

x	0.0	0.5	0.7
$f(x)$	1.005	0.857	0.881

Använd dessa tabellvärden för att uppskatta $f(x)$ för $x = 0.35$ genom linjär interpolation. Gör även en feluppskattning.

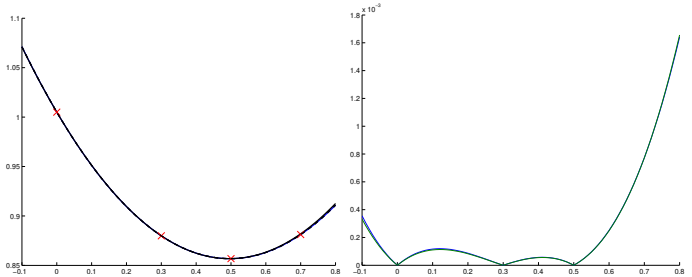
Vad händer om vi vill använda kvadratisk interpolation?

Hitta polynom $p_1(x)$ och $p_2(x)$ med hjälp av `numpy/polyfit`.



Vänster $p_1(x)$, $p_2(x)$ och $f(x)$. **Höger** Felet $|f(x) - p_1(x)|$ och $R_T \approx |p_2(x) - p_1(x)|$. Här är $|R_T| \leq 0.038$.

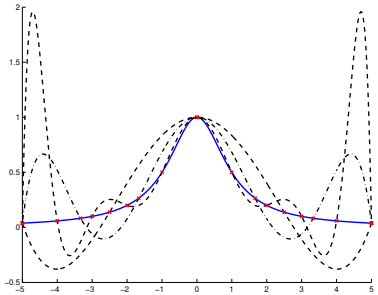
För kvadratisk interpolation behövs 3 punkter för $p_2(x)$ och en ytterligare punkt för att uppskatta R_T.



Vänster: $p_2(x)$, $p_3(x)$ och$f(x)$. **Höger:** Felet $|f(x) - p_2(x)|$ och $R_T \approx |p_3(x) - p_2(x)|$. Här är $|R_T| \leq 1.2 \cdot 10^{-4}$.

Vad händer om vi ökar polynomets gradtal ytterligare?

Runge's fenomen



Polynom av gradtal $n = 4$, 6, och 10 som interpolerar $f(x) = 1/(1+x^2)$.

Felet växer då polynomets gradtal ökar. Använd endast interpolerande polynom av *lågt* gradtal!

Lagrange's interpolationspolynom

Definition Låt $x_j, j = 1, \ldots, n$ vara distinkta punkter. Polynomen

$$\ell_i(x) = \Pi_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

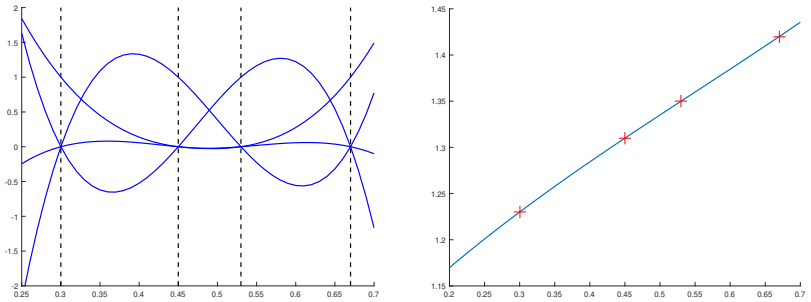
Egenskaper Det gäller att $\ell_i(x_i) = 1$ och $\ell_i(x_j) = 0$ om $i \neq j$.

Lemma Det unika polynom av grad $n-1$ som interpolerar $f(x)$ i punkterna x_j, $i = 1, 2, \ldots, n$ kan skrivas

$$p(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x).$$

Exempel Hitta det unika polynom som interpolerar tabellen

x_i	0.30	0.45	0.53	0.67
f_i	1.23	1.31	1.35	1.42



Resultat Lagrange bas polynom av grad $n = 3$. Trunkeringsfelet är $\mathcal{O}(h^4)$.

Fråga Kan vi generalisera till $2D$? Fel i funktionsvärden?

Lemma Låt $p(x)$ vara ett interpolerande polynom, av grad $n-1$. Om approximativa funktionsvärden $\bar{f}_i$ använts gäller att

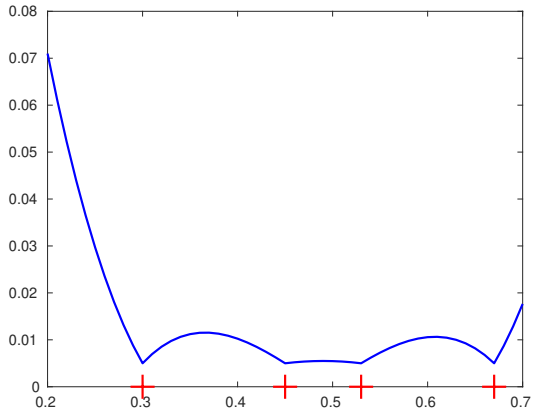
$$|\Delta p(x)| \leq \sum_{i=1}^{n} |\ell_i(x)||\Delta f_i|,$$

där $\ell_i(x)$ är Lagrange bas polynom.

Exempel Låt $p(x)$ vara det polynom som interpolerar tabellen

x_i	0.30	0.45	0.53	0.67
f_i	1.23	1.31	1.35	1.42

Uppskatta det maximala fel som orsakas av att f_i är korrekt avrundade.

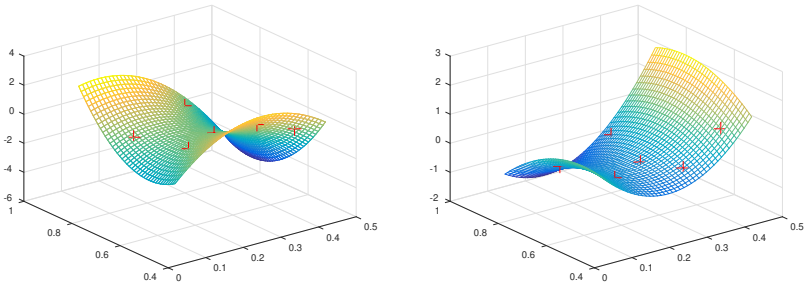


Felet $|R_{XF}|$ som orsakas av att $|\Delta f_i| \leq 0.5 \cdot 10^{-2}$. Felet är exakt $|\Delta f_i|$ vid interpolationspunkterna.

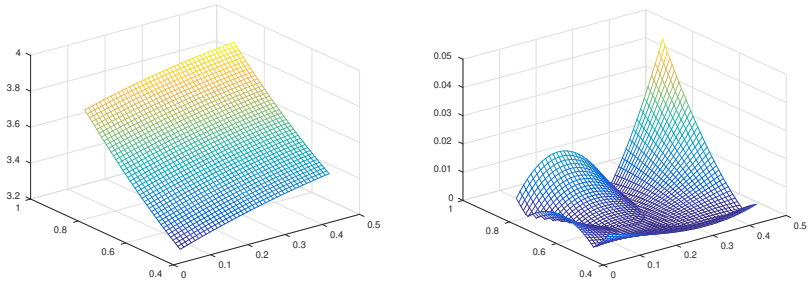
Exempel Antag att vi har $m = 6$ punkter (x_i, y_j) och vill hitta ett interpolerande polynom $u_I(x, y)$.

Introducera basfunktioner ℓ_i sådana att $\ell_i(x_j, y_j) = 1$ om $i = j$ och noll annars. Det interpolerande polynomet blir

$$u_I(x) = \sum_{i=1}^{m} u(x_i, x_i)\ell_i(x).$$



Två basfunktioner $\ell_2(x, y)$ och $\ell_3(x, y)$. Vi har $m = 6$ interpolationspunkter och termer $1, x, y, xy, x^2, y^2$.



Det interpolerande polynomet $u_I(x, y)$ och felet $|u_I - u|$, där $u = 3 + x\sqrt{y} + \cos(2x)y^2$, och $m = 6$ punkter använts.

Kommentar Enkel och flexibel metod. Ett andragradspolynom i $2D$ är unikt bestämt av $m = 6$ punkter? Vad gäller för ett tredjegradspolynom?

Tillämpning - Ekvationslösning

Problem Hitta en rot x^* till ekvationen $f(x) = 0$. Vi vill ha kvadratisk konvergens men inte beräkna f'. Vi beräknar x_{n+1} genom att hitta ett kvadratiskt polynom som interpolerar

x	x_{n-2}	x_{n-1}	x_n
$f(x)$	f_{n-2}	f_{n-1}	f_n

beräkna de två rötterna, x_i^*, $i = 1, 2$, till $p(x)$. Välj x_{n+1} som den rot med minst funktionsvärde $f(x_i^*)$.

Kommentar funktionen `polyint`, `polyval` och `roots` från `numpy` kan användas.

Exempel Lös $f(x) = \cos(x/2) + e^{-x/5} - x/2 - 4x^2 = 0$ med hjälp av kvadratisk interpolation.

k	x_k	$f(x_k)$
0	0	$2.00 \cdot 10^{0}$
1	1	$-2.80 \cdot 10^{0}$
2	2	$-1.58 \cdot 10^{1}$
3	0.61749871787530	$2.57 \cdot 10^{-3}$
4	0.61794339080406	$2.20 \cdot 10^{-6}$
5	0.61794377127614	$-1.98 \cdot 10^{-12}$
6	0.61794377127579	$4.44 \cdot 10^{-16}$

Kommentar Ser ut som kvadratisk konvergens. Hur skall vi undvika två funktionsberäkningar?

Metod Låt $y_i = f(x_i)$ och $x_i = f^{-1}(y_i)$. Hitta ett kvadratiskt polynom som interpolerar

y	f_{n-2}	f_{n-1}	f_n
$f^{-1}(y)$	x_{n-2}	x_{n-1}	x_n

Välj $x_{n+1} = p(0)$.

Kommentar En funktionsberäkning i varje steg. Interpolationssteget fungerar ej alltid. Kallas *Inverse Quadratic Interpolation* och är en av standard metoderna för att lösa ekvationer.

Kombineras ofta med någon typ av metod för att hålla reda på ett intervall där roten säkert finns för att garantera konvergens.

Exempel Lös $f(x) = \cos(x/2) + e^{-x/5} - x/2 - 4x^2 = 0$ med *Inverse quadratic interpolation*

k	x_k	$f(x_k)$
0	0	$2.00 \cdot 10^{0}$
1	1	$-2.80 \cdot 10^{0}$
2	2	$-1.58 \cdot 10^{1}$
3	0.45769147717309	$8.20 \cdot 10^{-1}$
4	0.59042717372728	$1.56 \cdot 10^{-1}$
5	0.62070311273328	$-1.60 \cdot 10^{-2}$
6	0.61795896942350	$-8.77 \cdot 10^{-5}$
7	0.61794377007804	$6.91 \cdot 10^{-9}$
8	0.61794377127579	$2.37 \cdot 10^{-16}$

Mycket snabb konvergens!