MA8020 Tekniska beräkningar

Något om interpolation

Mikael Hindgren

HÖGSKOLAN
I HALMSTAD

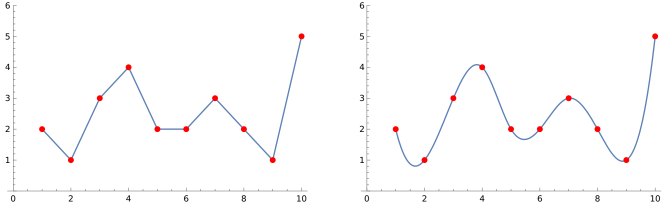
6 november 2025

Vad är interpolation?

Metod för att konstruera nya datapunkter inom intervallet för ett givet dataset.

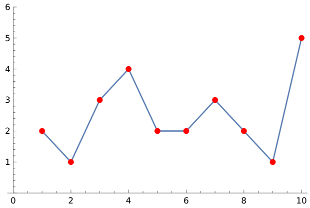
Exempel på tillämpningar:

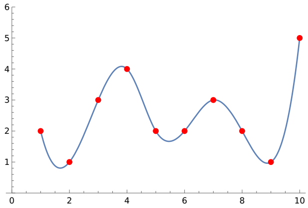
- *Datorgrafik*: Utjämning och kurvanpassning i grafisk rendering.
- *Dataanalys*: Uppskatta mellanvärden i experimentella data.
- *Signalbehandling:* DA-omvandling
- *Maskininlärning*: Uppskatta saknade värden i ett dataset eller göra förutsägelser om nya datapunkter baserat på känd data.
- *Teknisk design*: Skapa mjuka övergångar i CAD/CAM-design.
- *Naturvetenskap*: Modellera fysikaliska fenomen där data är ofullständig.



Typer av Interpolation

- *Polynominterpolation*: Konstruerar ett polynom som går genom alla givna datapunkter.
- *Linjär interpolation*: Använder linjesegment mellan varje par av datapunkter.
- *Spline-interpolation*: Använder låggradspolynom i varje intervall för att säkerställa mjuk övergång vid datapunkterna.
- *Närmaste-granne-interpolation*:[1] Tilldelar det närmaste kända värdet till varje datapunkt.



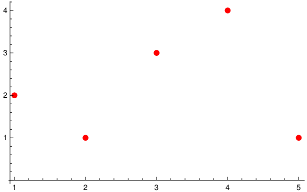


[1] Metoden används ofta inom bildbehandling och data där snabbhet prioriteras framför precision.

Interpolation

Antag att vi har en samling data för en okänd funktion $f(x)$:

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



- $x_1 < x_2 < \cdots < x_{n+1}$ kallas interpolationspunkter.
- Hur ska vi uppskatta $f(x)$ för $x_1 \leq x \leq x_{n+1}$?
- Vi söker en funktion $P(x)$ som interpolerar f i punkterna $x_1, x_2, ..., x_{n+1}$ dvs

$$P(x_i) = f_i,\ i = 1, 2, ..., n+1. \quad (*)$$

  Interpolanten $P(x)$ kan sedan användas för att uppskatta $f(x)$ för $x \neq x_i$.
- Om $x \in [x_1, x_{n+1}]$ kallas detta interpolation annars extrapolation.
- $(*) \Rightarrow P(x)$ kan konstrueras som en linjärkombination av basfunktioner $\varphi_k(x)$:

$$P(x) = \sum_{k=0}^{n} c_k \varphi_k(x)$$

Interpolation

Vilka basfunktioner ska vi välja? Några vanligt förekommande exempel:

- $\varphi_k(x) = x^k \Rightarrow P(x)$ polynom av grad n
- Ortogonala basfunktioner:
  $\varphi_k(x) = \sin kx$, $\varphi_k(x) = \cos kx$, Legendrepolynom...

Interpolation

Newtons interpolationspolynom

Sats 1

Antag att vi har $n+1$ punkter (x_i, f_i). Det finns då ett unikt polynom $p_n(x)$ av grad n som interpolerar de givna punkterna, dvs $p_n(x_i) = f_i$, $i = 1, 2, ..., n+1$.

Vi vill hitta ett polynom $p_n(x)$ som interpolerar värdena i tabellen

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

Ansats:

$$p_n(x) = c_0 + c_1(x - x_1) + c_2(x - x_1)(x - x_2) + \cdots c_n(x - x_1)(x - x_2)\cdots(x - x_n).$$

Villkoren $p_n(x_i) = f(x_i)$ ger

$$\begin{aligned}
p_n(x_1) &= c_0 = f_1 \\
p_n(x_2) &= c_0 + c_1(x_2 - x_1) = f_2 \Leftrightarrow c_1 = \frac{f_2 - c_0}{x_2 - x_1} \\
p_n(x_3) &= c_0 + c_1(x_3 - x_1) + c_2(x_3 - x_1)(x_3 - x_2) = f_3 \Leftrightarrow c_2 = \frac{f_3 - c_0 - c_1(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} \\
&\vdots
\end{aligned}$$

Interpolation

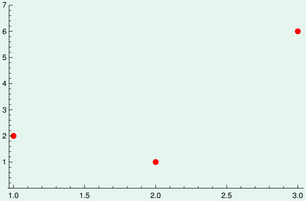
Newtons interpolationspolynom

Man kan enkelt lägga till datapunkter utan att behöva konstruera en helt ny interpolant.

Exempel 1

Bestäm Newtons interpolationspolynom av grad 2 till följande data:

x	1	2	3
y	2	1	6



$$
\begin{aligned}
\text{Ansats:}\quad p_2(x) &= c_0 + c_1(x-1) + c_2(x-1)(x-2)\\
p_2(1) &= c_0 = 2\\
p_2(2) &= c_0 + c_1(2-1) = 1 \Leftrightarrow c_1 = -1\\
p_2(3) &= c_0 + c_1(3-1) + c_2(3-1)(3-2) = 6 \Leftrightarrow c_2 = 3
\end{aligned}
$$

$\therefore p_2(x) = 2 - (x-1) + 3(x-1)(x-2) = 3x^2 - 10x + 9.$

Interpolation

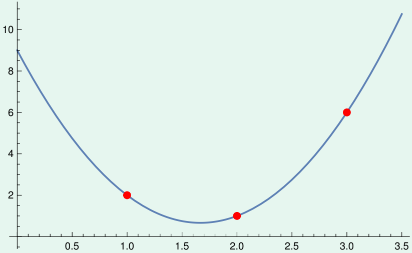
Newtons interpolationspolynom

Exempel 1

Kontroll med Mathematica:

```
Remove["Global`*"]
data = {{1, 2}, {2, 1}, {3, 6}};
lp = ListPlot[data, PlotStyle → {Red, PointSize[Large]}, PlotRange → {0, 7}];
p = Fit[data, {1, x, x^2}, x]
Show[Plot[p, {x, 0, 3.5}], lp]
```

$9. - 10.\, x + 3.\, x^2$



Interpolation

Feluppskattning

Sats 2

Antag att vi har $n+1$ punkter (x_i, f_i) och ett polynom $p_n(x)$ av grad n som interpolerar de givna punkterna.

Om $f_i = f(x_i)$ där f är en funktion med $n+1$ kontinuerliga derivator i $I = [x_1, x_{n+1}]$ så är

$$R_T = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_1)(x - x_2)\cdots(x - x_{n+1})$$

för något $\xi \in I$

I praktiken känner vi inte $f^{(n+1)}(\xi)$. Vi väljer istället en extra punkt (x_{n+2}, f_{n+2}), beräknar $p_{n+1}(x)$ och använder approximationen

$$f^{(n+1)}(\xi) \approx p_{n+1}^{(n+1)}(\xi) = (n+1)!c_{n+1}$$

Detta innebär att trunkeringsfelet uppskattas som

$$R_T \approx p_{n+1}(x) - p_n(x)$$

Interpolation

Feluppskattning

Exempel 2

Antag att

$$f(x) = e^{-x/5} \cos\left(\frac{x}{2}\right) + \frac{(x-0.1)^2}{3}$$

och uppskatta $f(0.5)$ med hjälp av tabellen

x	0.0	0.4	0.7
$f(x)$	1.0033	0.9347	0.9367

genom

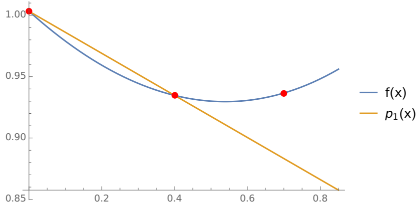
1. linjär interpolation
2. kvadratisk interpolation

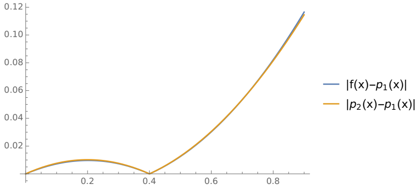
Gör även feluppskattning genom att lägga till punkten $(0.8, 0.9482)$.

Interpolation
Feluppskattning

Resultat:

$$p_1(x) = 1.00333 - 0.171545x \Rightarrow p_1(0.5) = 0.917561$$





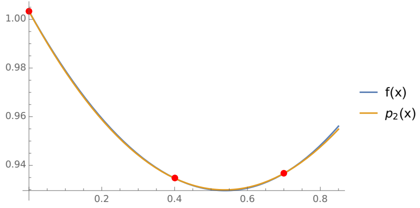
- Felet $|R_T| = |f(x) - p_1(x)| \leq 0.1$ i intervallet $[x_1, x_3] = [0, 0.8]$
- För $x = 0.5$ är $|R_T| < 0.02$.

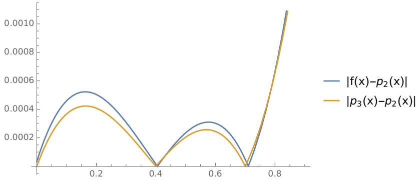
Interpolation

Feluppskattning

Resultat:

$$p_2(x) = 1.00333 - 0.273258x + 0.254282x^2 \Rightarrow p_2(0.5) = 0.930275$$





- Felet $|R_T| = |f(x) - p_2(x)| \leq 0.001$ i intervallet $[x_1, x_3] = [0, 0.8]$
- För $x = 0.5$ är $|R_T| < 0.0003$.

Interpolation

Runges fenomen

Blir alltid approximationen bättre om man ökar antalet interpolationspunkter dvs interpolationspolynomets grad?

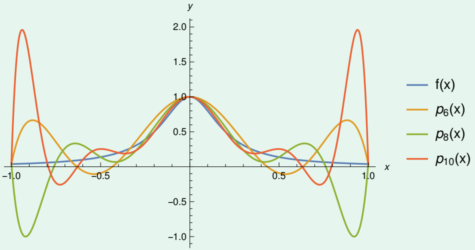
Exempel 3

Funktionen

$$f(x) = \frac{1}{1+25x^2}$$

interpoleras med $p_n(x)$ i punkterna

$$x_i = -1 + (i-1)\frac{2}{n}, \quad i = 1, 2, ..., n+1$$



Man kan visa att felet

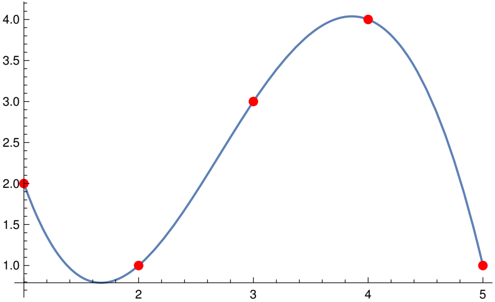
$$\max_{-1 \le x \le 1} |f(x) - p_n(x)| \to \infty \text{ då } n \to \infty$$

Slutsats: Interpolera endast med polynom av lågt gradtal!

Interpolation

Spline-interpolation

- För att undvika Runges fenomen vid interpolation med ett polynom kan man använda olika polynom i olika delar av intervallet.
- En sådan metod är sk Spline-interpolation
- Antag att $f(x)$ är känd i noderna $x_1, x_2, ..., x_{n+1}$.
- Vi vill hitta en funktion $s(x) \approx f(x)$ på $[x_1, x_{n+1}]$ sådan att $s(x_i) = f(x_i)$.



Interpolation

Spline-interpolation

Definition 1

Antag att $a = x_1 < x_2 < \cdots < x_{n+1} = b$ och att funktionsvärdena $f(x_i)$, $i = 1, ..., n+1$, är kända. En funktion $s(x)$ sammansatt av polynom av grad $2m+1$ så att

- $s(x)$ är $2m$ gånger kontinuerligt deriverbar
- $s(x_i) = f(x_i)$, $i = 1, 2, ..., n+1$

kallas en interpolerande spline-funktion av grad $2m+1$ på intervallet $[a, b]$.

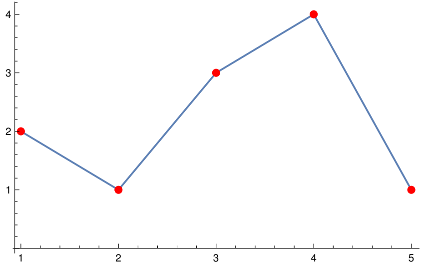
- $m = 0$ ger linjära och $m = 1$ kubiska spline-funktioner.
- Det är mest kubiska spline-funktioner som används i praktiken.

Interpolation

Linjär spline-interpolation

$s(x)$ är en interpolerande linjär splinefunktion med noder $[x_1, ..., x_{n+1}]$ om:

1. $s(x_i) = f(x_i)$, $i = 1, 2, ..., n + 1$.
2. $s(x)$ är kontinuerlig på $[x_1, x_{n+1}]$.
3. $s(x)$ är en rät linje på varje delintervall $[x_i, x_{i+1}]$.



En linjär splinefunktion $s(x)$ är alltså sammansatt av n räta linjer

$$s_i(x) = a_i + b_i(x - x_i), \quad x_i \leq x \leq x_{i+1}.$$

Interpolation

Linjär spline-interpolation

$s(x)$ kan uttryckas med hjälp av basfunktioner $\varphi_i(x)$ som kallas "hattfunktioner":

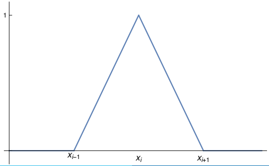
$$s(x) = \sum_{i=1}^{n+1} f(x_i)\varphi_i(x)$$

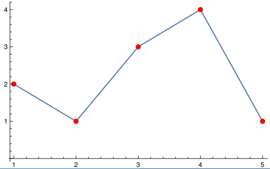
där

$$\varphi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} < x < x_i \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i < x < x_{i+1} \\ 0 & \text{annars} \end{cases} \Rightarrow \quad \varphi_i(x_j) = \delta_{ij} = \begin{cases} 1, \; i = j \\ 0, \; i \neq j \end{cases}$$

I ändpunkterna används “halvhattar”:

$$\varphi_1(x) = \begin{cases} \dfrac{x_2 - x}{x_2 - x_1}, & x_1 < x < x_2 \\ 0, & \text{annars} \end{cases} \quad \text{respektive} \quad \varphi_{n+1} = \begin{cases} \dfrac{x - x_n}{x_{n+1} - x_n}, & x_n < x < x_{n+1} \\ 0, & \text{annars} \end{cases}$$





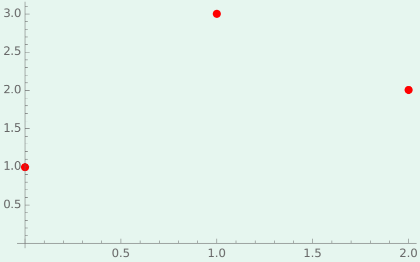
Interpolation

Linjär spline-interpolation

Exempel 4

Bestäm en linjär splinefunktion $s(x)$ som interpolerar

x	0	1	2
$f(x)$	1	3	2



Interpolation

Linjär spline-interpolation

Exempel 4 (forts)

Ansats:

$$s(x) = \begin{cases} s_1(x) = c_0 + c_1(x-0), & 0 \leq x < 1 \\ s_2(x) = d_0 + d_1(x-1), & 1 \leq x \leq 2 \end{cases}$$

Kontinuitetsvillkoret på $s(x)$ ger nu

$$\begin{aligned} s(0) &= s_1(0) = c_0 = f(0) = 1 \\ s(1) &= s_1(1) = c_0 + c_1 = s_2(1) = d_0 = f(1) = 3 \\ s(2) &= s_2(2) = d_0 + d_1 = f(2) = 2 \end{aligned}$$

$$\Leftrightarrow (c_0, c_1, d_0, d_1) = (1, 2, 3, -1)$$

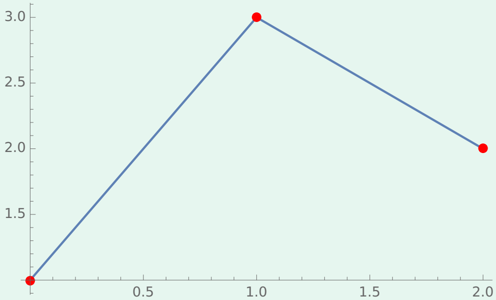
Interpolation

Linjär spline-interpolation

Exempel 4 (forts)

Vi får den linjära spline-funktionen

$$s(x) = \begin{cases} s_1(x) = 1 + 2x, & 0 \leq x < 1 \\ s_2(x) = 4 - x, & 1 \leq x \leq 2 \end{cases}$$



Interpolation

Kubisk spline-interpolation

$s(x)$ är en interpolerande kubisk splinefunktion med noder $x_1, \ldots, x_{n+1}$ om:

1. $s(x_i) = f(x_i)$, $i = 1, 2, \ldots, n+1$.
2. $s(x)$, $s'(x)$ och $s''(x)$ är kontinuerliga på $[x_1, x_{n+1}]$
3. $s(x)$ är ett tredjegrads polynom på varje delintervall $[x_i, x_{i+1}]$.

$s(x)$ blir entydigt bestämd om vi anger två ändpunktsvillkor. Alternativ:

- Naturliga: $s''(x_1) = s''(x_{n+1}) = 0$
- Rätta: Om derivatan av f i ändpunkterna är känd:
  $s'(x_1) = f'(x_1)$ och $s'(x_{n+1}) = f'(x_{n+1})$
- Periodiska: Om $f(x)$ är periodisk med perioden $x_{n+1} - x_1$:
  $s'(x_1) = s'(x_{n+1})$ och $s''(x_1) = s''(x_{n+1})$

Interpolation

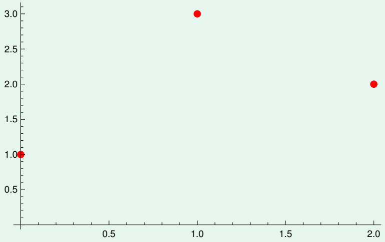
Kubisk spline-interpolation

Exempel 5

Bestäm en kubisk splinefunktion $s(x)$ som interpolerar

x	0	1	2
$f(x)$	1	3	2

med rätta ändpunktsvillkor då $f'(0) = 1$ och $f'(2) = -1$.



Interpolation

Kubisk spline-interpolation

Exempel 5 (forts)

Ansats:

$$s(x) = \begin{cases} s_1(x) = c_0 + c_1(x-0) + c_2(x-0)^2 + c_3(x-0)^3, & 0 \le x < 1 \\ s_2(x) = d_0 + d_1(x-1) + d_2(x-1)^2 + d_3(x-1)^3, & 1 \le x \le 2 \end{cases}$$

Villkoren för en kubisk splinefunktion med rätta ändpunktsvillkor leder till

$$\begin{aligned}
s(0) &= s_1(0) = c_0 = f(0) = 1 \\
s(1) &= s_1(1) = c_0 + c_1 + c_2 + c_3 = s_2(1) = d_0 = f(1) = 3 \\
s(2) &= s_2(2) = d_0 + d_1 + d_2 + d_3 = f(2) = 2 \\
s_1'(1) &= c_1 + 2c_2 + 3c_3 = s_2'(1) = d_1 \\
s_1''(1) &= 2c_2 + 6c_3 = s_2''(1) = 2d_2 \\
s'(0) &= s_1'(0) = c_1 = f'(0) = 1 \\
s'(2) &= s_2'(2) = d_1 + 2d_2 + 3d_3 = f'(2) = -1
\end{aligned}$$

Interpolation

Kubisk spline-interpolation

Exempel 5 (forts)

Mathematica löser enkelt ekvationssystemet:

$$s(x) = \begin{cases} s_1(x) = 1 + x + 3.25x^2 - 2.25x^3, & 0 \leq x < 1 \\ s_2(x) = -3 + 13x - 8.75x^2 + 1.75x^3, & 1 \leq x \leq 2 \end{cases}$$