SF1547 - Numeriska metoder, grundkurs

Block 2 - Approximation och anpassning

Elias Jarlebring
KTH Royal Institute of Technology
Numerical Analysis division

Elias: Extra redovisningstillfälle för att underlätta hanteringen av redovisningar

Översikt

- Polynominterpolation
  - Vandermondematris
  - Newton-ansats
- Minstakvadratanpassning
- Olinjär minstakvadratanpassning
  - Problemformulering
  - Gauss-Newtons metod

Polynominterpolation

Polynominterpolation

(Ant-Approx.pdf §1, Sauer kap 3)

Interpolation i punkter

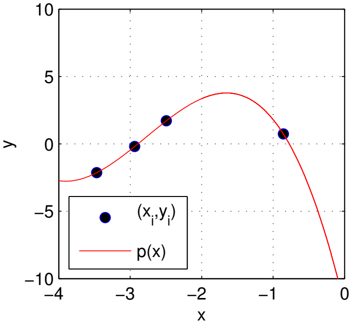
Givet $x_0 < \ldots < x_n$ och $y_0, \ldots, y_n$ hitta ett polynom $p(x)$ så att

$$p(x_j) = y_j, \quad j = 0, \ldots, n.$$

Interpolation av en funktion

Givet $x_0 < \ldots < x_n$ och $f(x)$, hitta ett polynom $p(x)$ så att

$$p(x_j) = f(x_j), \quad j = 0, \ldots, n.$$



Polynominterpolation (forts)

- Det finns ett entydigt polynom av grad n som uppfyller interpolationsvillkor. Se Sauer Teorem 3.2
- Polynomet
$$p(x) = c_0 + c_1 x + \cdots c_n x^n$$
kan konstrueras från Vandermondesystemet

$$\mathbf{f} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = A\mathbf{c}$$

Matrisen A kallas för en Vandermondematris. Ekvationen

$$A\mathbf{c} = \mathbf{f}$$

är ett linjärt ekvationssystem (som kan lösas t.ex. med backslash).

Elias: Gör ett exempel i matlab

Newton-ansatsen (Ant-Approx.pdf, Sauer kap 3.1.2)

Vandermondesystemet är numeriskt inte så pålitligt framförallt för stora n. Bättre (snabbare och bättre noggrannhet):

Newton-ansats

Alla polynom av grad n kan skrivas som

$$p(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \cdots + d_n(x - x_0)\cdots(x - x_{n-1}).$$

Det polynom som uppfyller interpolationsvillkoren uppfyller även

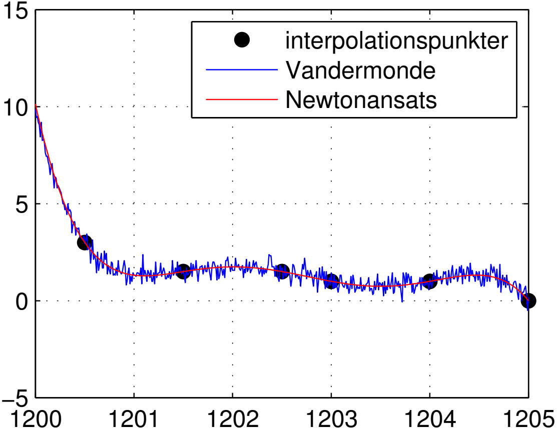
$$\mathbf{f} = A_{Newton}\mathbf{d}$$

med

$$A_{Newton} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & (x_1 - x_0) & 0 & \cdots & 0 \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & (x_n - x_0) & (x_n - x_0)(x_n - x_1) & \cdots & (x_n - x_0)\cdots(x_n - x_{n-1}) \end{bmatrix}$$

(Kan lösas med Gausseliminering som är ekvivalent med s.k. dividerade differenser. Se Sauer.)

Visa exempel: Newtonansatsexempel. Hackigt pga avrundningsfel.

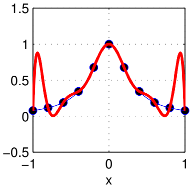


Mer om interpolation

Feluppskattning (Ant-Approx.pdf, Sauer 3.2.1)

Felet som sker vid interpolation kan uppskattas. Till exempel vid ekvidistant interpolation ($x_j - xj + 1 = h$ för all j) så gäller

$$E_n = \max_{a \leq x \leq b} |f(x) - p(x)| \leq \max_{a \leq \xi \leq b} \frac{|f^{(n+1)}(\xi)|}{4(n+1)} h^{n+1}.$$



Elias: Räkna exempel
Hit hann vi under föreläsning 3

Runges fenomen Sauer 3.2.3

Visa generellt Vandermondeprogram och Runge: $f(x) = 1/(1 + 12x^2)$

Polynominterpolation med ekvidistantanta punkter och hög polynomgrad leder ofta till höga oscillationer i utkanten av intervallet.

Polyinterpolation med hög grad ofta dåligt. En lösning är att använda...

Styckvis polynominterpolation: Ant-Approx.pdf

- Dela upp intervallet i mindre delar
- Använd polyinterpolation med lägre ordningens polynom i delintervallen

Elias Jarlebring

Enkel variation av interpolation:

Andra basfunktioner (Ant-Approx.pdf §3)

Vandermondesystemet kan generaliseras till interpolation med icke-polynom som basfunktioner

$$f(x) = c_0\psi_0(x) + c_1\psi_1(x) + \cdots + c_n\psi_n(x).$$

- Kan ge bättre approximation om man redan vet något om funktionen f.
- Entydighet mer komplicerat

Elias: visa exempel och genomför i matlab

Minstakvadratanpassning

(Delar av Sauer kap 4)

Problem

Givet $x_1 < \ldots < x_n$ och $f(x)$, hitta ett polynom $p(x)$ av grad m, med $m < n$ så att

$$p(x_j) \approx f(x_j), \;\; j = 1, \ldots, n.$$

- Från tidigare kurser: Minsta kvadratproblem. Problemet kan lösas i minstakvadratbemärkelse $^{(Elias:dk)}$, dvs så att vi hittar polynom av grad m som minimerar felkvadratsumman

$$\sum_{j=0}^{n} (p(x_j) - f(x_j))^2.$$

  Om $p(x) = c_0 + c_1 x + \cdots + c_n x^n$, så är koefficienterna lösningen till normalekvationen:

$$A^T A\mathbf{c} = A^T \mathbf{c}$$

  Kan generaliseras till linjärkombination av andra basfunktioner.
- $A\backslash b$ i matlab när A är en $n \times m$-matris ger lösningen till normalekvationen, men snabbare och mer noggrannt.
  Obs: backslash har olika beteenden när A är kvadratisk och när A är rektangulär.

Elias: Gör litet exempel

Olinjära approximation och anpassning

(Sauer Kap 4.5.1)

Exempel (Variant av ENM 4.6.)

Datapunkter för tidvatten i nordsjön:

Månad:	0	2	4	6	8	10
Tidvatten (m):	1	1.6	1.4	0.6	0.2	0.8

- Anpassning med modellfunktion:

$$g(t) = c_0 + c_1 \sin\left(\frac{2\pi t}{12}\right) + c_2 \cos\left(\frac{2\pi t}{12}\right)$$

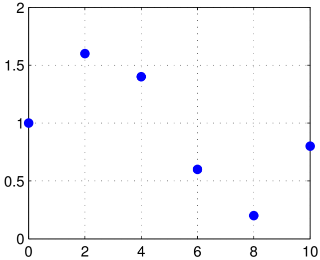
  där c_0, c_1, c_2 kan beräknas med backslash (eller normalekvationer).

Elias: Härled normalekvation för icke-polynom som basfunktion

- Anpassning med modellfunktion

$$g(t) = c_0 + c_1 \sin(c_4 t) + c_2 \cos(c_4 t)$$

  kan inte genomföras med normalekvationerna. Liknande exempel i Sauer: Exempel 4.24



Överbestämt olinjärt ekvationssystem

Antag att vi har givet en vektorvärd funktion av en vektor, dvs, $\mathbf{x} \in \mathbb{R}^m$ och

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, \ldots, x_m) \\ \vdots \\ f_n(x_1, \ldots, x_m) \end{bmatrix}.$$

Vi letar efter $\mathbf{x}_* \in \mathbb{R}^m$ så att

$$\mathbf{f}(\mathbf{x}_*) \approx \mathbf{0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n.$$

Obs:

$$\begin{array}{ccc} m & < & n \\ \text{Antalet obekanta} & < & \text{antalet ekvationer.} \end{array}$$

Vi ska nu lösa problemet i minstakvadratbemärkelse, dvs minimering:

$$\min_{\mathbf{x}} \|\mathbf{f}(\mathbf{x})\| = \|\mathbf{f}(\mathbf{x}_*)\|$$

Gauss-Newtons metod (Sauer 4.5.1)

Givet en approximation $\mathbf{x}_0$ beräknar vi nya approximationer $\mathbf{x}_1, \mathbf{x}_2, \ldots$ genom att definiera

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{h}_k, \quad k = 0, \ldots$$

där $\mathbf{h}_k$ bestäms genom

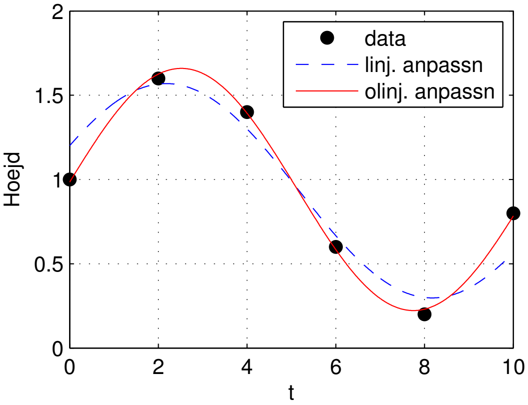
$$\min_{\mathbf{h}\in\mathbb{R}^m} \|J(\mathbf{x}_*)\mathbf{h} - f(\mathbf{x}_*)\| = \|J(\mathbf{x}_*)\mathbf{h}_k - f(\mathbf{x}_*)\|.$$

Upprepning tills $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < TOL$, där $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}} = norm(x)$

- Ekvation (*) är ett linjärt överbestämt ekvationssystem som kan lösas med backslash (eller normalekvationerna)
- Implementationen av Newtons metod och Gauss-Newtons metod i MATLAB är identiska om man använder backslash.

Elias: Gör tidvattenexemplet med olinjär anpassning

Tidvattenexempel:



Linjära och olinjära ekvationssystem och överbestämda ekvationssystem i eran grundutbildning:

$$
\begin{aligned}
n &= \text{antalet ekvationer} \\
m &= \text{antalet obekanta}
\end{aligned}
$$

	$n = m$	$n > m$
Linjärt	Gausseliminering (SF1604)	Normalekvationer (SF1604)
Olinjärt	Newtons metod (SF1547)	Gauss-Newton (SF1547)