

Practicum NMB: Benaderende functies

Matthijs van Keirsblick en Harald Schäfer

donderdag 14 mei 2015

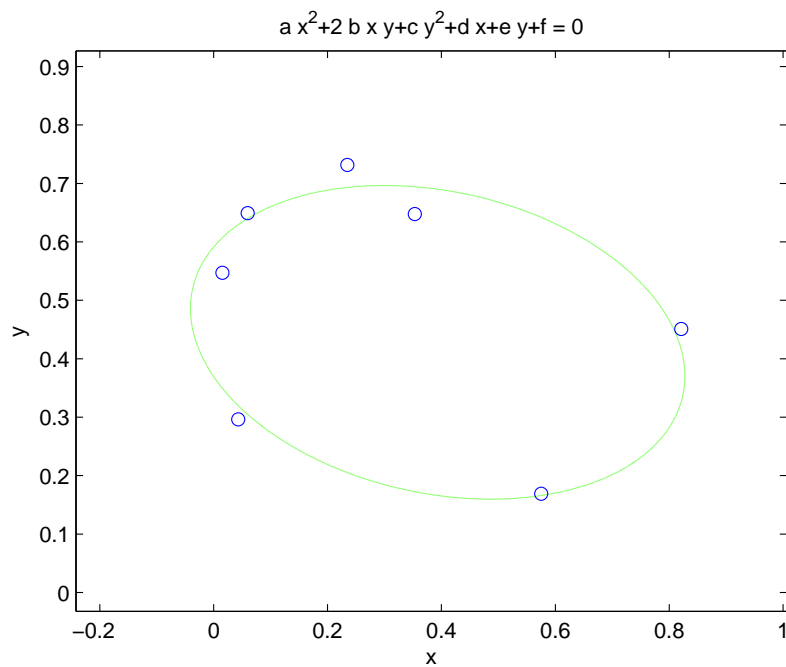
Opgave 1

Ellips

Voor het berekenen van de coëfficiënten ellips lossen we het gegeven stelsel op naar b, c, d, e en f .

$$\begin{bmatrix} 2x_1y_1 & y_1^2 - x_1^2 & x_1 & y_1 & 1 \\ 2x_1y_1 & y_1^2 - x_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2x_1y_1 & y_1^2 - x_1^2 & x_1 & y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} -x_1^2 \\ -x_2^2 \\ \vdots \\ -x_N^2 \end{bmatrix}$$

Hieruit kunnen we dan ook gemakkelijk a berekenen met de normalisatievoorwaarde $a = 1 - c$. In figuur 1 zien we de resulterende ellips die een aantal willekeurige punten benadert.



Figuur 1: Benaderende ellips voor een aantal willekeurige punten

Cirkel

Voor het berekenen van de coëfficiënten van een benaderende cirkel beginnen we weer van de vergelijking:

$$ax^2 + 2bxy + cy^2 + dx + ey + f = 0$$

Daarin vullen we de normalisatievoorwaarde in en de voorwaarde van een cirkel, deze zijn:

$$a = 1 \qquad b = 0 \qquad a = c$$

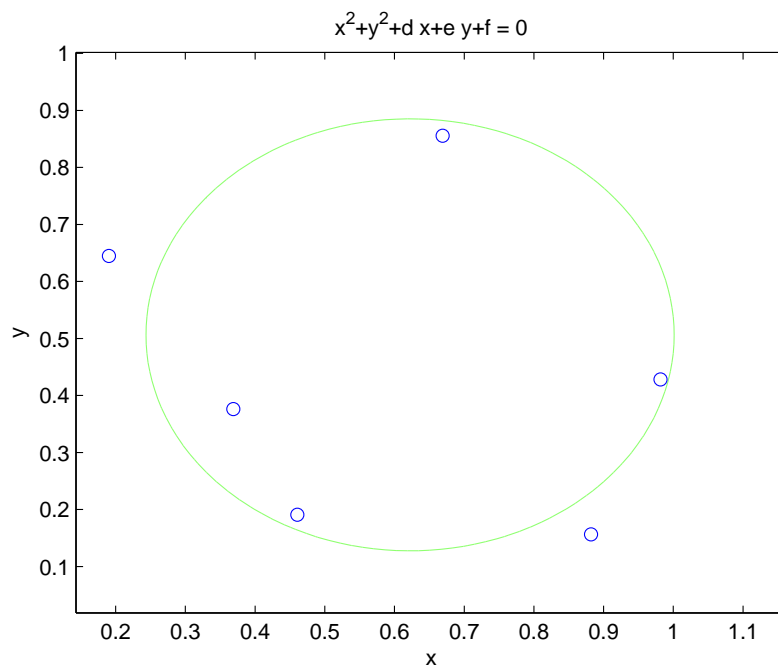
Dit geeft ons de vergelijking:

$$x^2 + y^2 + dx + ey + f = 0$$

Het stelsel om op te lossen wordt dan:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_1 & y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} -y_1^2 - x_1^2 \\ -y_2^2 - x_2^2 \\ \vdots \\ -y_N^2 - x_N^2 \end{bmatrix}$$

In figuur 2 zien we de benaderende cirkel berekend met deze methode voor een aantal willekeurige punten.



Figuur 2: Benaderende cirkel voor een aantal willekeurige punten

Tekenkegelsnede

De vergelijking van de kegelsnede in het aangepast assenstelsel is

$$\lambda_1 \bar{x}^2 + \lambda_2 \bar{y}^2 + \bar{c} = 0$$

Om een ellips(of cirkel) te zijn moet $\lambda_1 > 0$, $\lambda_2 > 0$ en $\bar{c} \geq 0$. λ_1 en λ_2 zijn de eigenwaarden van de matrix:

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

\bar{c} vinden we uit de volgende vergelijkingen.

$$B = \begin{bmatrix} d \\ f \end{bmatrix} \quad 2T^t A = -B^t \quad \bar{c} = T^t A T + B^t T + f$$

We hebben nu alle nodige vergelijkingen om na te kijken of de coëfficiënten a, b, c, d, e en f voldoen aan de voorwaarden van een ellips(of cirkel). We hebben onze functie getest door de volgende testen uit te voeren.

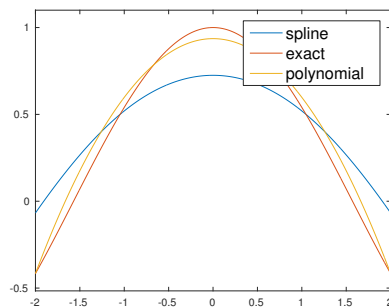
- `tekenkegelsnede()` uitgevoerd op de coëfficiënten berekend door `ellips()` toegepast op punten liggend op een willekeurige ellips, het resultaat was zoals verwacht 1
- `tekenkegelsnede()` uitgevoerd op de coëfficiënten berekend door `cirkel()` toegepast op punten liggend op een willekeurige cirkel, het resultaat was zoals verwacht 1
- `tekenkegelsnede()` uitgevoerd op de coëfficiënten berekend door `ellips()` toegepast op punten liggend op een rechte, het resultaat was zoals verwacht 0

Opgave 2

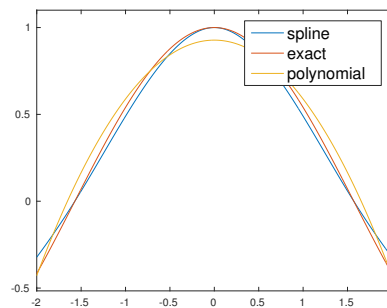
Voor de theoretische achtergrond en afleidingen, zie sectie 5.2.3 van de cursus.

Benadering $\cos(x)$

Kubische splineinterpolaties zijn pas mogelijk pas vanaf 4 interpolatiepunten. Voor 4 interpolatiepunten is de veeltermbenadering door oplossen van het Kleinste Kwadratenprobleem nog beter, maar vanaf 5 punten convergeert de splineinterpolatie bijna direct naar de exacte functie. In de figuren 3 en 4 worden deze benaderingen weergegeven. De veelterminterpolatie op de figuren gebeurt met een veelterm van graad 2. Hogere graads veeltermen geven een betere benadering van de functie op het interval, op voorwaarde dat er genoeg punten genomen worden. Veeltermen van te hoge graad introduceert wel oscillaties.



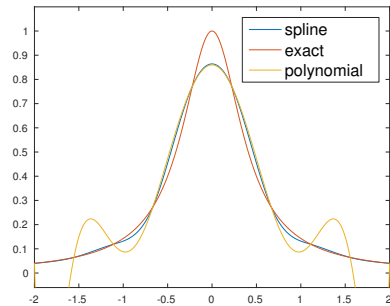
Figuur 3: 4 interpolatiepunten



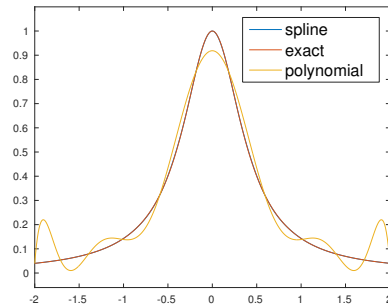
Figuur 4: 5 interpolatiepunten

Benadering $1/(1 + 6 * x^2)$

Voor beide benadermethodes bekomen we geen goede benaderingen voor N kleiner dan 10. In alle gevallen is de spline- benadering hier beter dan de benadering met veeltermen. In de grafieken 5 en 6 worden deze benaderingen weergegeven. De veelterminterpolatie op de figuren gebeurt met een veelterm van graad 10.



Figuur 5: 10 interpolatiepunten

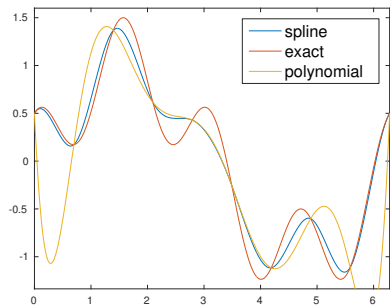


Figuur 6: 15 interpolatiepunten

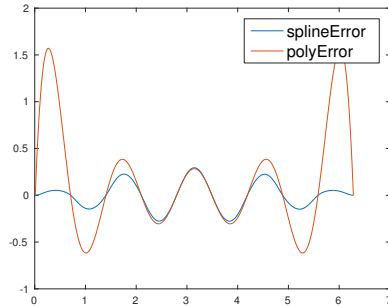
Benadering $\sin(x) + \cos(4 * x)/2$

Benadering met 10 interpolatiepunten

Ook hier is bij beide benadermethodes geen goede benadering gevonden voor N kleiner dan 10. In alle gevallen is de spline- benadering hier ook weer beter dan de benadering met veeltermen. In de figuren 7 en 8 worden deze benaderingen en de fout erop weergegeven. De veelterminterpolatie op de figuren gebeurt met een veelterm van graad 10.



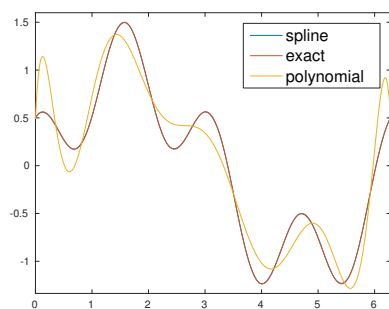
Figuur 7: 10 interpolatiepunten



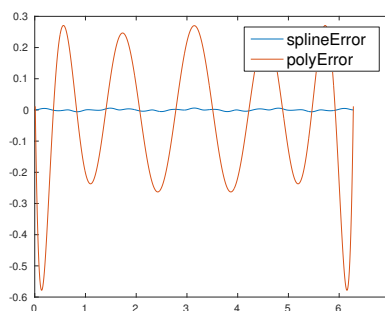
Figuur 8: fout bij 10 interpolatiepunten

Benadering met 20 interpolatiepunten

In de figuren 9 en 10 worden zien we de benaderingen en fout voor 20 interpolatiepunten.

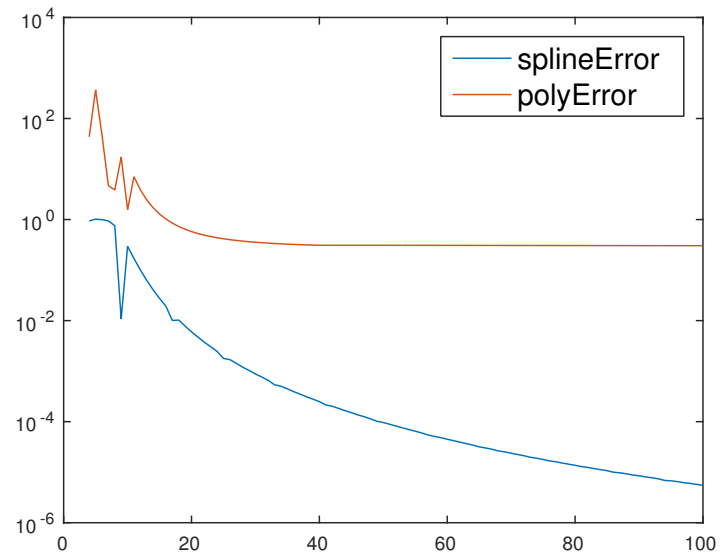


Figuur 9: 20 interpolatiepunten



Figuur 10: fout bij 20 interpolatiepunten

Beide benadermethodes convergeren naar de functie, maar de splinebenadering convergeert veel sneller. In figuur 11 wordt de evolutie van de fout in functie van het aantal interpolatiepunten weergegeven op een log-schaal. De veeltermbenadering gebruikt een veelterm van graad 10.

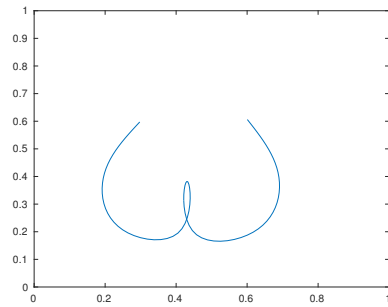


Figuur 11: De maximale fout in functie van het aantal interpolatiepunten

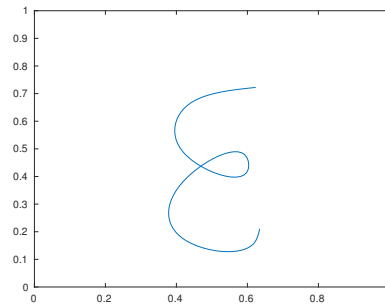
Benadering van een kromme

Met behulp van parametrisaties kunnen we willekeurige krommen benaderen met kubische splinefuncties.

Griekse letters

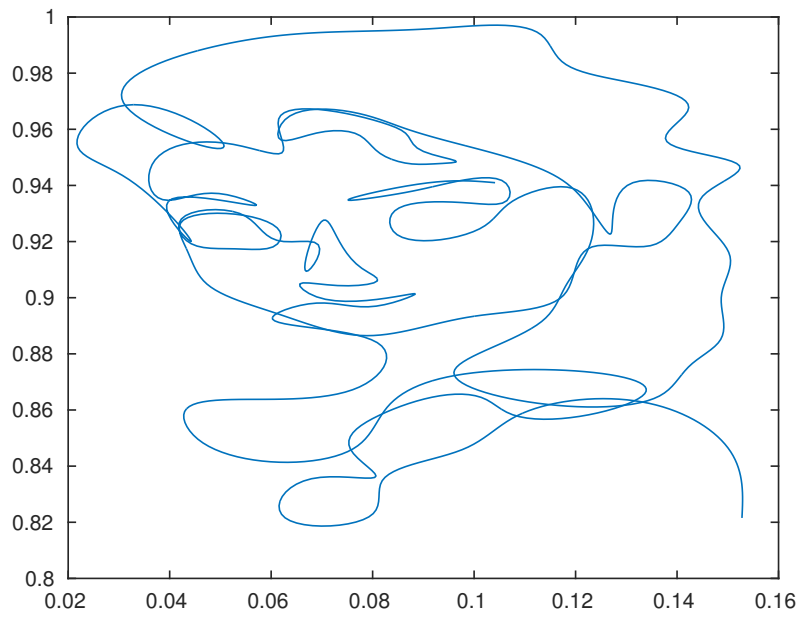


Figuur 12: omega



Figuur 13: epsilon

Disney



Figuur 14: Elsa