

Numerieke Modelling en Benadering: Practicum 2

Ellen Anthonissen Marte Biesmans

donderdag 25 mei 2017

1 Bivariate kleinste-kwadraten veeltermbenadering

Opgave 1

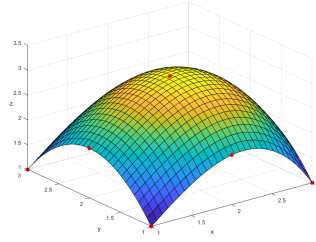
Onderstaande functie berekent de coëfficiëntenmatrix $C \in \mathbb{R}^{(n+1) \times (m+1)}$, gegeven de vectoren $x \in \mathbb{R}^M$ en $y \in \mathbb{R}^N$ als 2D meetpunten, de matrix $F \in \mathbb{R}^{N \times M}$ met functie- of meetwaarden en de parameters $m, n \in \mathbb{N}$ die de graad in x en y respectievelijk van de benaderende veelterm voorstelt.

```
1 function C = kkb(x,y,F,m,n)
2 % Benader een oppervlak op basis van de functiewaarden F op een
   rechthoekig
3 % puntenrooster (x_i, y_i) met als benaderende functie een
   bivariate
4 % veelterm van graad m in x en graad n in y.
5 % x: vector met M waarden, interpolatiepunten op x-as
6 % y: vector met N waarden
7 % F: matrix met functiewaarden
8 % m: graad in x van benaderende functie
9 % n: graad in y van benaderende functie
10
11 %construct A and B
12 A = zeros(size(x,2),m+1);
13 B = zeros(size(y,2),n+1);
14 for k = 0:m
15     A(:,k+1) = x'.^k;
16 end
17 for l = 0:n
18     B(:,l+1) = y'.^l;
19 end
20 %calculate C
21 C = B\F * pinv(A)';
22 end
```

Er werd een kleine test op deze functie uitgevoerd. Hierbij is de input:
 $x = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, y = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, m = 3, n = 2$ en

$$F = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

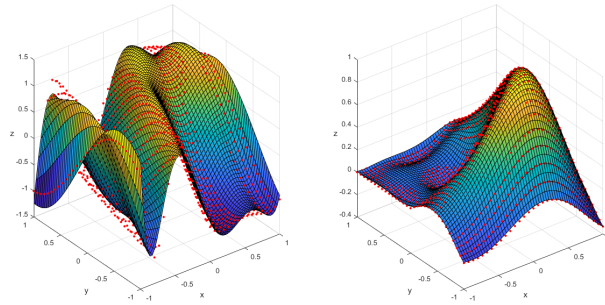
De punten werden samen geplott met het benaderend oppervlak in figuur 1.



Figuur 1: veeltermbenadering van F van graad 3 in x en graad 2 in y

Opgave 2

Voor de eerste dataset, gegenereerd door $f(x, y) = \sin((2x - 1)^2 + 2y)$ op elk punt van het rooster toe te passen, worden de veeltermbenadering van graad 7 in x en y in het interval $[-1, 1] \times [-1, 1]$ en de exacte functiewaarden weergegeven op de linkse grafiek van figuur 2. Voor de tweede dataset, gegenereerd door $F = \text{membrane}(1, 15)$ in MATLAB, worden de veeltermbenadering van graad 7 in x en y en de exacte data weergegeven op de rechtse grafiek van figuur 2.

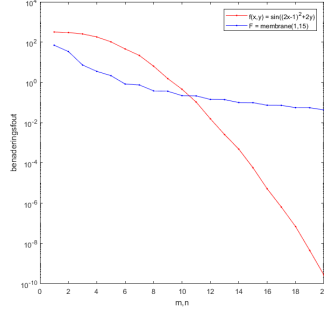


Figuur 2: veeltermbenadering van graad 7 in x en y van de functie $f(x, y) = \sin((2x - 1)^2 + 2y)$ (links) en $F = \text{membrane}(1, 15)$ (rechts)

Opgave 3

De opeenvolgende waarden van de benaderingsfout $\|f - z_{mn}\|_2$ voor $m = n = 1, \dots, 20$ voor beide datasets van opgave 2 worden weergegeven in figuur 3. We zien hier duidelijk dat voor de tweede dataset veeltermoppervlakken een geschikte benadering is. De eerste dataset heeft een slechte veeltermbenadering

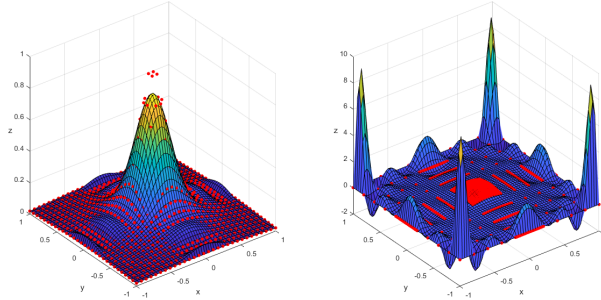
voor kleine m, n . Dat komt omdat de functie $f(x, y) = \sin((2x - 1)^2 + 2y)$ een zekere periodiciteit bevat en kan dus beter benaderd worden door middel van trigonometrische functies.



Figuur 3: benaderingsfout $\|f - z_{mn}\|_2^2$ van de veeltermbenadering van de functie $f(x, y) = \sin((2x - 1)^2 + 2y)$ en $F = \text{membrane}(1, 15)$ in functie van de graad in x en y

Opgave 4

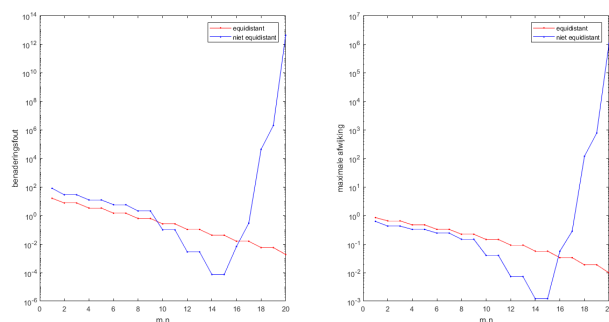
De veeltermbenadering van de functie van Runge $f(x, y) = \frac{1}{1+25(x^2+y^2)}$ gebruik makend van twee verschillende roosters wordt getoond in figuur 4. Zowel een rooster met equidistante punten (links) als een rooster met niet-equidistante punten (rechts) werd gebruikt ter vergelijking.



Figuur 4: veeltermbenadering van graad 10 in x en y van de functie van Runge $f(x, y) = \frac{1}{1+25(x^2+y^2)}$ gebruik makend van equidistante roosterpunten (links) en niet-equidistante roosterpunten (rechts)

Het is hier duidelijk dat het rooster dat gebruik maakt van een equidistante verdeling te verkiezen is. Dit komt doordat bij de niet-equidistante verdeling er weinig aandacht gaat naar de randen van het 2D interval. Bij het verhogen van de graad van de veeltermbenadering gaan er zich namelijk grote schommelingen voordoen aan de rand van het interval. Dit zorgt voor een zeer grote maximale fout bij hogere machten, die voorkomt in de hoekpunten (zie figuur 5 rechts).

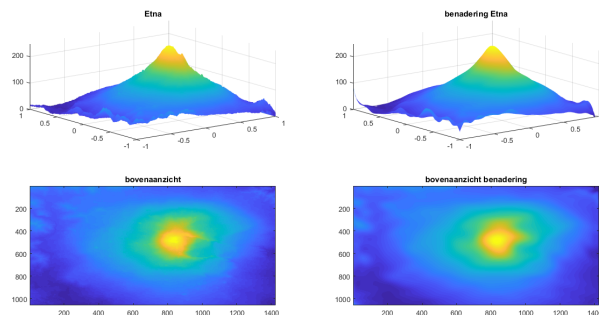
We zien ook dat de benaderingsfout $\|f - z_{mn}\|_2^2$ (figuur 5) en de maximale afwijking (figuur 5) voor een niet-equidistante verdeling toch beperkt blijft voor $m, n \leq 15$, terwijl het benaderende veeltermoppervlak toch niet zo goed is. De fout bij de roosterpunten blijft inderdaad beperkt, maar de fout buiten de roosterpunten is daarentegen wel groot.



Figuur 5: benaderingsfout $\|f - z_{mn}\|_2^2$ (links) en de maximale afwijking (rechts) van de veeltermbenadering van de functie van Runge $f(x, y) = \frac{1}{1+25(x^2+y^2)}$ in functie van de graad in x en y

Opgave 5

De veeltermbenadering van de Etna van graad 25 in x en y wordt getoond in figuur 6. Voor M en N werd het aantal pixels gekozen in de x en y richting respectievelijk gekozen.



Figuur 6: 3D-weergave en bovenaanzicht van de Etna (links) en van de benadering (rechts)

2 Interpolerende splinefuncties en -curven

Opgave 1

In de functie `naturalspline` stellen we een natuurlijke kubische splinefunctie op op basis van een stel gegeven punten $[x, f]$. Hierbij evalueren we dit resultaat voor de gegeven waarden t . Dit gebeurt in verschillende stappen:

1. Stel een stelsel op met de tridiagonale coëfficiëntenmatrix (5.19) en het rechterlid (5.20) uit het handboek.
2. Los dit stelsel op om de tweede afgeleiden in elke abscis te bepalen.
3. Bepaal de onbekende constanten c_{1i} en c_{2i} voor elke i met de formules (5.14) en (5.15) uit het handboek.
4. Met al deze gegevens y bepalen aan de hand van formule (5.13) uit het handboek.

Het kan voorvallen dat de waarde van t buiten het interval van de gegeven abscissen ligt. Dit is niet ideaal, maar als dit voorvalt geven we y de functiewaarde die hoort bij de abscis op de rand. Ook als t juist op een abscis ligt, kunnen we rekenwerk besparen en geven we y de functiewaarde uit de gegeven punten.

Om de correctheid van de implementatie te testen hebben we de functie $\sin(x)$ benaderd op het interval $[-2,2]$ met equidistante punten. Hierbij gebruikten we 4 equidistante abscissen en 11 evaluatiepunten. Door dit zowel op papier als in matlab uit te rekenen kunnen stap per stap de waarden voor A , b , S , c_1 , c_2 en y gecontroleerd worden.

```

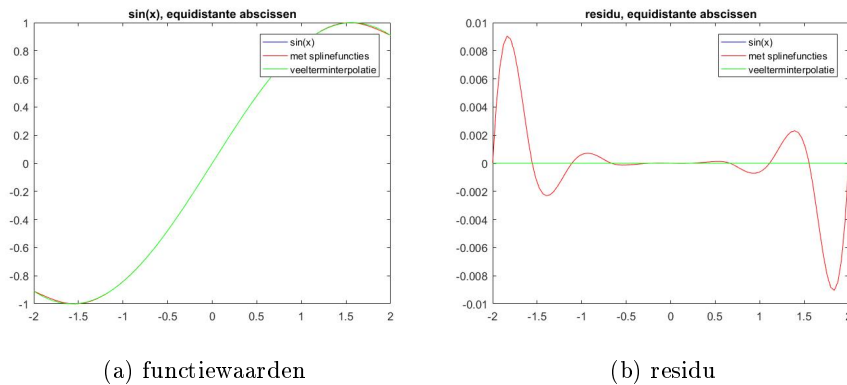
1 function y = naturalspline(x,f,t)
2 % construct and evaluate a natural cubic spline
3 % x: abscissen, row vector of length n+1          x_i in x(1,
4           i+1)
5 % f: functiewaarden, row vector of length n+1      f_i in f(1,
6           i+1)
7 % t: evaluatiepunten, row vector of length N
8 % y: functiewaarden in de evaluatiepunten, row vector of length N
9
10 %stelsel opstellen (HB p. 112, 5.19 en 5.20)
11 main_diag = zeros(1,n-1);
12 side_diag = zeros(1,n-1);
13 b = zeros(1,n-1);
14 for i=1:n-1
15     main_diag(1,i) = 2*(x(1,i+2)-x(1,i)); % 2 * (x_{i+1} - x_{i-1})
16     side_diag(1,i) = x(1,i+2)-x(1,i+1); % x_{i+1} - x_i
17     b(1,i) = (f(1,i+2)-f(1,i+1)) / (x(1,i+2)-x(1,i+1)) ... %f_{i
18           +1 - f_i / x_{i+1} - x_i
19           - (f(1,i+1)-f(1,i)) / (x(1,i+1)-x(1,i)); %f_i
20           - f_{i-1} / x_i - x_{i-1}
21 end
22 b = 6.*b;
23 A = diag(main_diag, 0) + diag(side_diag(1:n-2), -1) + diag(
24     side_diag(1:n-2), 1);
25 %stelsel oplossen, S is een kolomvector met s"(x_1) tem s"(x_{n-1})
26 S = A\b';
27 %constanten c1i en c2i berekenen en bewaren in een rijvector

```

```

26 c_1 = zeros(1,n);
27 c_2 = zeros(1,n);
28 for j=1:n
29     if (j==1) % s"(x_0) = 0
30         c_1(1,j) = f(1,j+1)/(x(1,j+1)- x(1,j)) - (x(1,j+1)-x(1,j))
                    /6*S(j,1);
31         c_2(1,j) = f(1,j)/(x(1,j+1)-x(1,j));
32     elseif (j==n) % s"(x_n)=0
33         c_1(1,j) = f(1,j+1) / (x(1,j+1)-x(1,j));
34         c_2(1,j) = f(1,j)/(x(1,j+1)-x(1,j)) - (x(1,j+1)-x(1,j))/6*S
                    (j-1,1);
35     else
36         c_1(1,j) = f(1,j+1)/(x(1,j+1)- x(1,j)) - (x(1,j+1)-x(1,j))
                    /6*S(j,1);
37         c_2(1,j) = f(1,j)/(x(1,j+1)-x(1,j)) - (x(1,j+1)-x(1,j))/6*S
                    (j-1,1);
38     end
39 end
40
41 %y bepalen
42 y = zeros(1,N);
43 for k=1:N
44     %bepalen tussen welke x-waarden de huidige t-waarde ligt
45     l=0;
46     if (t(1,k)>x(1,n+1))
47         l = n+1;
48     else
49         while(t(1,k)>x(1,l+1))
50             l = l+1;
51         end
52     end
53
54     %y_t berekenen (HB p. 111, 5.13)
55     if (l==0) % t ligt links van de eerste abscis, gebruik eerste
        functiewaarde
56         y(1,k) = f(1,1);
57     elseif (l==n+1) % t ligt rechts van de laatste abscis, gebruik
        laatste functiewaarde
58         y(1,k) = f(1,n+1);
59     elseif (t(1,k)==x(1,l+1)) % t ligt juist op een abscis
60         y(1,k) = f(1,l+1);
61     elseif (l==1) % t ligt in eerste interval, s"(x_0) = 0
62         y(1,k) = (t(1,k) - x(1,l))^3 / (6*(x(1,l+1)-x(1,l))) * S(l
            ,1) ...
63         + c_1(1,l) * (t(1,k) - x(1,l)) ...
64         + c_2(1,l) * (x(1,l+1) - t(1,k));
65     elseif (l==n) % t ligt in laatste interval, s"(x_n) = 0
66         y(1,k) = - (t(1,k) - x(1,l+1))^3 / (6*(x(1,l+1)-x(1,l)))
            * S(l-1,1) ...
67         + c_1(1,l) * (t(1,k) - x(1,l)) ...

```



Figuur 7: interpolant door 10 equidistante absciswaarden van $\sin(x)$

```

68         + c_2(1,l) * (x(1,l+1) - t(1,k));
69     else
70         y(1,k) = (t(1,k) - x(1,l))^3 / (6*(x(1,l+1)-x(1,l))) * S(l
71             ,1) ...
72         - (t(1,k) - x(1,l+1))^3 / (6*(x(1,l+1)-x(1,l))) * S(l
73             -1,1) ...
74         + c_1(1,l) * (t(1,k) - x(1,l)) ...
75         + c_2(1,l) * (x(1,l+1) - t(1,k));
76     end
end
end

```

Opgave 2

We gebruiken bovenstaande code om $\sin(x)$ en $\frac{1}{1+7x^2}$ te benaderen op het interval $[-2,2]$ met een equidistante set van 10 abscissen voor $\sin(x)$ en 9 abscissen voor de tweede functie. We vergelijken dit ook met een veelterminterpolant door deze punten. Het eerste resultaat is te vinden in figuur 7, het tweede resultaat in figuur 8. We geven steeds een plot van de functiewaarden en van het residu weer.

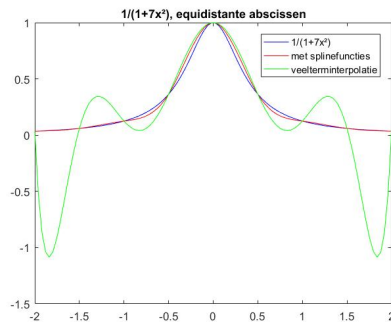
De functie $\frac{1}{1+7x^2}$ is een klassiek voorbeeld van een functie die slecht te banderen is met een interpolerende veelterm door equidistante punten. Dit is duidelijk te zien in de plot van het residu. De splinevoorstelling presteert hier een stuk beter.

TODO verdere uitleg over verschil veelterm vs. spline bij \sin ??

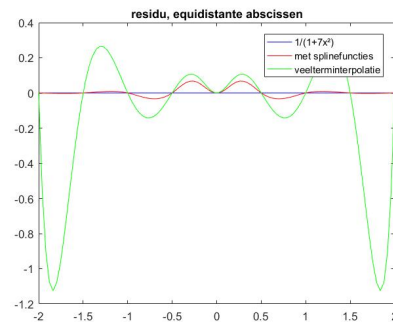
Opgave 3

Nu doen we hetzelfde, maar in plaats van equidistante abscissen gebruiken we nu Chebyshev-punten. Het resultaat voor $\sin(x)$ is te vinden in figuur 9, het resultaat voor $\frac{1}{1+7x^2}$ in figuur 10. We geven ook hier steeds een plot van de functiewaarden en van het residu weer.

Het gebruik van Chebyshev-punten is een duidelijke verbetering, vooral aan de

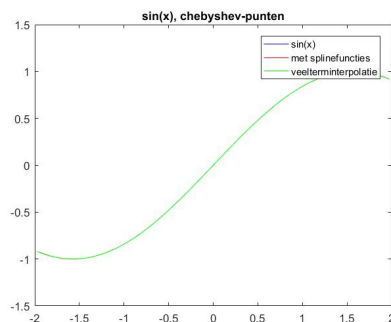


(a) functiewaarden

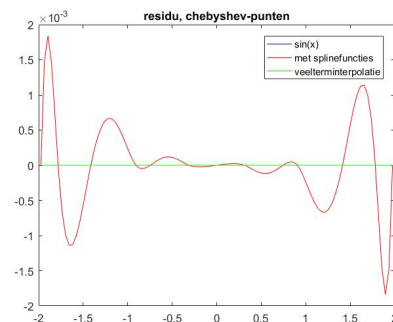


(b) residu

Figuur 8: interpolant door 9 equidistante absciswaarden van $\frac{1}{1+7x^2}$



(a) functiewaarden



(b) residu

Figuur 9: interpolant door 10 Chebyshev-punten van $\sin(x)$

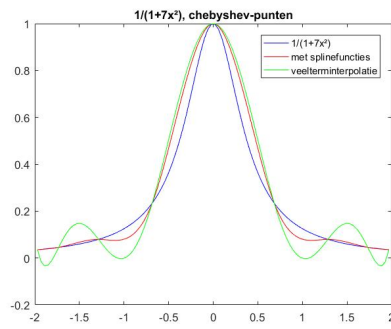
rand van het interval.

TODO verdere uitleg over verschil veelterm vs. spline

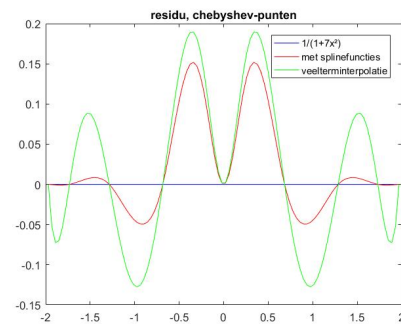
Opgave 4

We kozen als griekse letters de letters delta en phi aangezien deze geen scherpe knikken bevatten. Het resultaat is te zien in figuur 11. De aangeduide punten zijn de inputgegevens, de rode lijn is de berekende curve.

In figuur 12 zie je op dezelfde manier de splinevoorstelling van het mannetje van La Linea. Ook hier ziet men duidelijk dat knikken niet worden weergegeven. Dit is logisch aangezien dit een ander probleem is dan het opstellen van een natuurlijke, interpolerende, kubische splinecurve. Er zouden immers andere voorwaarden opgelegd worden aan de verschillende afgeleiden.

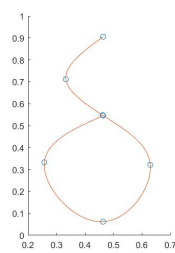


(a) functiewaarden

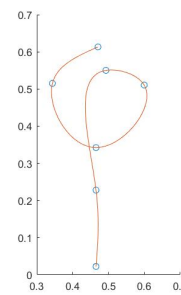


(b) residu

Figuur 10: interpolant door 9 Chebyshev-punten van $\frac{1}{1+7x^2}$

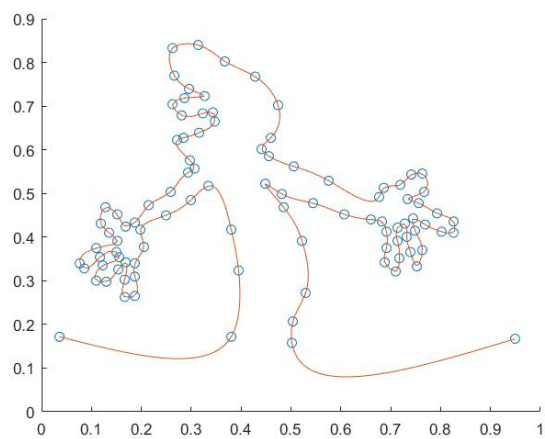


(a) delta



(b) phi

Figuur 11: griekse letters benaderd door interpolerende, kubische splinecurven



Figuur 12: splinevoorstelling van het mannetje van La Linea