

Numerieke Modelling en Benadering
Practicum 2
Incalza Dario & Boussé Martijn

23 mei 2013

Lijst van figuren

1	Benaderingen voor verschillende waarden van K	5
	(1.1) $K = 5, M = 28$	5
	(1.2) $K = 10, M = 28$	5
	(1.3) $K = 28, M = 28$	5
2	Enkele B-splines ter illustratie van de correctheid van het algoritme van de Boor.	8
3	Illustratie van het belang van de ligging van de knooppunten bij het construeren van een spline kleinste-kwadratenbenadering voor een $\text{sinc}(x)$ functie.	9
	(3.1) Benadering met meer knooppunten op het begin en einde van het interval.	9
	(3.2) Fout tussen de benadering en de echte functie voor meer knooppunten op het begin en einde van het interval.	9
	(3.3) Benadering met meer knooppunten in het midden van het interval.	9
	(3.4) Fout tussen de benadering en de echte functie voor meer knooppunten in het midden van het interval.	9
	(3.5) B-splines van een benadering met meer knooppunten op het begin en einde van het interval.	9
	(3.6) B-splines van een benadering met meer knooppunten in het midden van het interval.	9
4	Invloed van de knooppunten op de kleinste-kwadratenbenadering van een $\text{sinc}(x)$ functie met Gaussiaanse ruis.	10
	(4.1) Weinig knooppunten	10
	(4.2) Veel knooppunten	10
5	Kleinste-kwadratenbenadering van de solseutel uit de muziekleer.	10

1 Benaderen met trigonometrische veeltermen

Voor het benaderen van continue periodieke functies of gesloten krommen zijn klassieke veeltermen niet aangewezen. In het eerste deel van het practicum onderzoeken we daarom trigonometrische veeltermen.

1.1 Matlab-functie voor het evalueren van periodieke interpolerende en benaderende trigonometrische veeltermen

```
1 function y = periotrig(x, K, M)
2     N = size(x,1); %aantal rijen v/d matrix
3     d = size(x,2); %aantal kolommen v/d matrix
4
5     %de fft berekenen van X
6     X = fft(x);
7     %de nodige coëfficiënten op nul zetten voor
8     X(K+2:N,:) = 0;
9
10    %allocatie van de matrix
11    Y = zeros([N d]);
12    %alle coëfficiënten zijn omwille
13    %van periodiciteit gelijk aan : Y_k ==
14    %Y_(M-k)
15    for k = (M/2)+1:M-1
16        Y(k+1,:) = conj(Y(M-k+1,:));
17    end
18
19    %de juiste waarde voor Y_k kiezen volgens de
20    %voorwaarden in de opgave, toepassen op de hele matrix
21    %voor een efficiëntere berekening.
22    Y(1:N/2,:) = M*X(1:N/2,:)/N;
23    Y((N/2)+1,:) = M*X((N/2)+1,:)/(2*N);
24    Y((N/2)+2:(M/2)+1,:) = 0;
25
26    % y berekenen als de ifft van Y
27    y = ifft(Y);
28
29    % we plotten nu de oplossing
30    x_waarden = y(:,1);
31    y_waarden = y(:,2);
32    str = sprintf('Grafiek met K = %f , en M = %f',K,M);
33    plot(x_waarden,y_waarden,'b-');
34    title(str);
35
36 end
```

We gaan eerst de FFT berekenen van de ingevoerde matrix x, daarop vervolgens zetten we de nodige coëfficiënten op 0. De volgende stap bestaat eruit om de Y_k coëfficiënten te berekenen. Deze moeten voldoen aan :

$$\begin{cases} Y_k = \frac{M}{N} * X_k, & k = 0, \dots, N/2 - 1 \\ Y_k = \frac{M}{N} * \frac{X_k}{2}, & k = N/2 \\ Y_k = 0, & k = N/2 + 1, \dots, M/2 \end{cases} \quad (1)$$

Nu dient er echter nog een voorwaarde opgesteld te worden voor de coëfficiënten Y_k waarvoor geldt dat $k = M/2 + 1, \dots, M - 1$. Als gevolg van de periodiciteit en symmetrie geldt er voor deze coëfficiënten : $Y_k = \bar{Y}_{M-k}$. Al deze voorwaarden zijn geïmplementeerd van lijn 15 tot 24. Eens deze coëfficiënten bekend zijn gebruiken we de *inverse Fourier-transformatie* en bekomen zo de resultaat matrix y .

1.2 Illustreer en bespreek de invloed van de parameter K op de benadering van een periodieke veelterm.

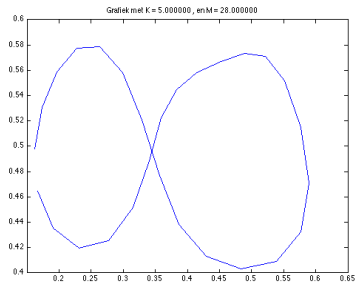
We doen dit ineens door gebruik te maken van de opgegeven functie `click()`. Wanneer we hier het oneindig teken mee genereren krijgen we als uitkomst tabel 1 van 28 punten.

<i>Coördinaten</i>	x	y
1	0.167050691244240	0.483918128654971
2	0.190092165898618	0.425438596491228
3	0.226958525345622	0.419590643274854
4	0.282258064516129	0.422514619883041
5	0.314516129032258	0.460526315789473
6	0.337557603686636	0.483918128654971
7	0.362903225806452	0.521929824561403
8	0.381336405529954	0.542397660818713
9	0.413594470046083	0.562865497076023
10	0.457373271889401	0.568713450292398
11	0.480414746543779	0.568713450292398
12	0.528801843317972	0.568713450292398
13	0.549539170506913	0.559941520467836
14	0.577188940092166	0.510233918128655
15	0.586405529953917	0.469298245614035
16	0.579493087557604	0.437134502923976
17	0.535714285714286	0.404970760233918
18	0.496543778801843	0.404970760233918
19	0.411290322580645	0.413742690058479
20	0.392857142857143	0.437134502923976
21	0.360599078341014	0.478070175438596
22	0.332949308755760	0.516081871345029
23	0.286866359447005	0.565789473684210
24	0.263824884792627	0.574561403508772
25	0.243087557603687	0.574561403508772
26	0.185483870967742	0.565789473684210
27	0.171658986175115	0.530701754385965
28	0.167050691244240	0.483918128654971

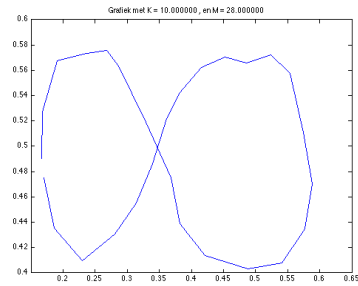
Tabel 1: x- en y-coördinaten voor 28 punten van het oneindig teken

We zetten tabel 1 om naar een matrix x en geven deze aan de zelfgeschreven functie *periotrig*. We plotten eveneens figuren voor de verschillende K waarden om zo de invloed van deze parameter, voor het benaderen door trigonometrische veeltermen, te bespreken.

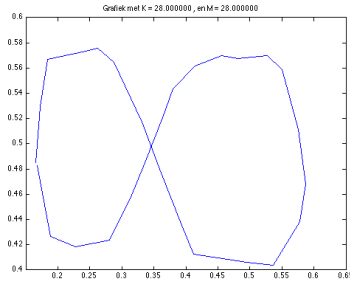
K stelt de index waarden voor en wordt gebruikt om de X_k coëfficiënten op nul te zetten voor alle coëfficiënten groter als K . We kunnen dus theoretisch vermoeden dat er een betere benadering bekomen wordt indien K hoog is. Praktisch illustreren we dit door het oneindig teken te benaderen, we gebruiken hiervoor tabel 1 als meetpunten waarop we gaan interpoleren, met verschillende waarden voor K , en bekomen zo figuur 1. We zien op figuur 11.1 duidelijk dat de benadering niet goed is. Er is een duidelijke onderbreking te zien. In figuren 11.2 en 11.3 zien we duidelijk de benadering beter worden aangezien meerdere X_k coëfficiënten gebruikt worden.



(1.1) $K = 5, M = 28$



(1.2) $K = 10, M = 28$



(1.3) $K = 28, M = 28$

Figuur 1: Benaderingen voor verschillende waarden van K .

2 Kleinste-kwadratenbenadering met splinefuncties

In dit het tweede deel van het practicum trachten we een kleinste-kwadratenbenadering op te stellen met behulp van splinefuncties. Hiertoe worden twee MATLAB functies geschreven:

- *function c = kkb_spline(t,x,f,k)*: Deze MATLAB functie berekent de kleinste-kwadratenbenadering van meerdere functies tegelijk. De output *c* is een matrix waarvan elke kolom de coëfficiënten bevat de kleinste-kwadratenbenadering voor de overeenkomstige kolom in *f*.
- *function z = de_boor(c,x,f,t,k)*: De coëfficiëntenmatrix *c*, bekomen uit de voorgaande functie, wordt nu gebruikt om de functiewaarden *z* te berekenen via het algoritme van de Boor.

2.1 Matlab-functie voor het opstellen en evalueren van een kleinste-kwadratenbenadering met kubische splinefuncties.

```
1 function c = kkb_spline(t,x,f,k)
2     %% Berekening van de matrix M
3     % grootte van de knooppuntenrij
4     [n,r] = size(t); n = n-2*k-1;
5     % grootte van de abcis vector
6     [r,r] = size(x);
7     % Allocatie van de matrix M
8     M = zeros(r,n+k);
9     for j = 1:r
10        % zoek [t(i), t(i+1)] voor gegeven abcis
11        for i = k+1:n+k+1
12            if x(j)<t(i)
13                break
14            end
15        end
16        i = i-1;
17        % Bereken de tabel N
18        N = zeros(k+1+1,k+1); % Extra rij met nullen
19        N(k+1,1) = 1;
20        for z = 2:k+1 % Voor elke kolom
21            for q=k+1:-1:1 % Helemaal tot boven in de tabel
22                w = i-(k+1)+q;
23                % Coëfficiënten
24                c1 = (x(j)-t(w))/(t(w+z-1)-t(w));
25                c2 = (t(w+z-1+1)-x(j))/(t(w+z-1+1)-t(w+1));
26                % Combineren
27                N(q,z) = c1*N(q,z-1) + c2*N(q+1,z-1);
28            end
29        end
30        % Laatste kolom van de tabel inladen in de matrix M
31        M(j,i-k:i) = N(1:k+1,k+1)';
32    end
33    %% Berekening van de coëfficiënten
34    c = M\f;
35 end
```

2.2 Evaluatie van de kleinste-kwadratenbenadering met behulp van het algoritme van de Boor.

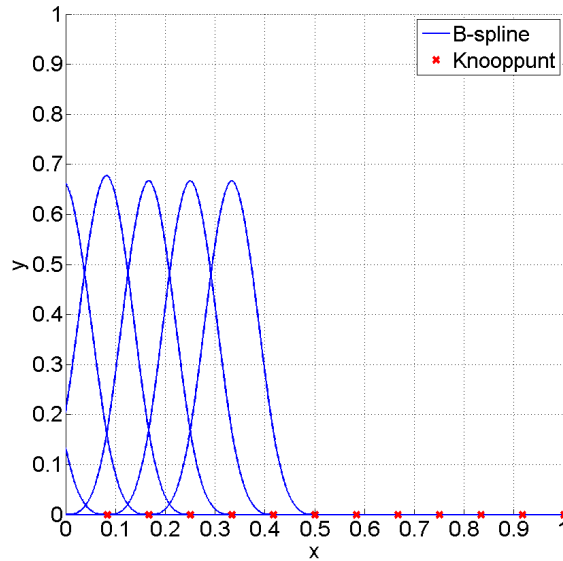
```
1 function z = de_boor(C,x,f,t,k)
2     % grootte van de knooppuntenrij
3     [n,~] = size(t); n = n-2*k-1;
4     % grootte van de abcis vector
5     [r,~] = size(f);
6     % grootte coëfficiënten matrix
7     [~,d] = size(c);
8     for v = 1:d;
9         for j = 1:r
10             % zoek [t(i), t(i+1)] voor gegeven abcis
11             for i = k+1:n+k+1
12                 if x(j)<t(i)
13                     break
14                 end
15             end
16             i = i-1;
17             P = zeros(k+1,k+1);
18             P(:,1) = c(i-k:i,v);
19             for z = 2:k+1
20                 for q = 2:k+1;
21                     w = i-(k+1)+q;
22                     alpha1 = (x(j)-t(w))/(t(w+k+1-z+1)-t(w));
23                     alpha2 = 1-alpha1;
24                     P(q,z) = alpha1*P(q,z-1)+alpha2*P(q-1,z-1);
25                 end
26             end
27             S(j,v) = P(end,end);
28         end
29     end
30     z = S;
31 end
```

2.3 Illustratie van de B-splines

Figuur 2 toont een aantal B-splines ter illustratie van de correctheid van het algoritme van de Boor. Noteer dat elke B-spline verschilt van de nulfunctie in slechts $k + 1$ intervallen, en wordt bepaald door $k + 2$ knooppunten. In een willekeurig punt van het knooppunten interval $[t_0, t_n]$ zijn hoogstens $k + 1$ B-splines verschillend van nul.

2.4 Belang van de ligging van de knooppunten

Figuur 3 toont het belang van de ligging van de knooppunten bij het construeren van een spline kleinste-kwadratenbenadering voor het klassieke regressie probleem van een $\text{sinc}(x)$ functie. In Figuur 33.1 wordt veel meer knooppunten gekozen op het begin en einde van het interval dan in het midden van het interval, vice versa voor Figuur 33.3. Figuren 33.5 en 33.6 illustreren het effect op de B-splines voor de verschillende keuzes van de knooppunten.



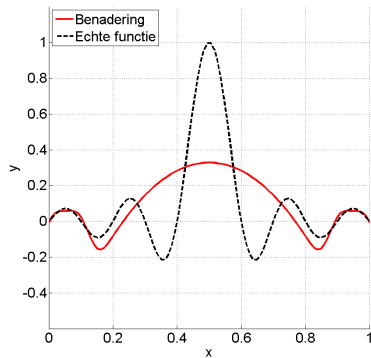
Figuur 2: Enkele B-splines ter illustratie van de correctheid van het algoritme van de Boor.

2.5 Kleinste-kwadratenbenadering van een $\text{sinc}(x)$ met Gaussiaanse ruis

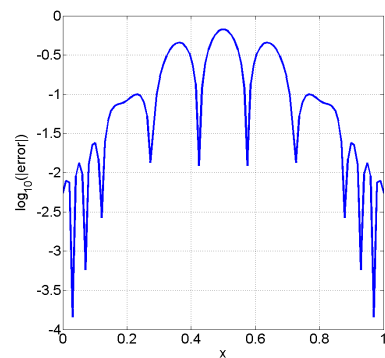
Figuur 44.1 toont de kleinste-kwadratenbenadering van een $\text{sinc}(x)$ met Gaussiaanse ruis. Belangrijk hierbij is de keuze van het aantal knooppunten; legt men bijvoorbeeld te weinig knooppunten in sommige deelgebieden, dan zal de benadering daar onvoldoende aansluiten bij de data. Legt men daarentegen teveel knooppunten, dan zal de benadering de ruis beginnen te volgen, zoals getoond in Figuur 44.2

2.6 Kleinste-kwadratenbenadering van de solseutel

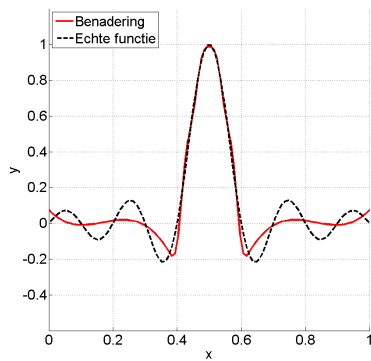
Figuur 5 toont de kleinste-kwadratenbenadering van de solseutel uit de muziek-leer. De meetpunten werden gekozen met behulp van de routine *click.m*. Dit levert twee sequenties van meetpunten die gebruikt worden als invoer voor de functie *kkb_spline*.



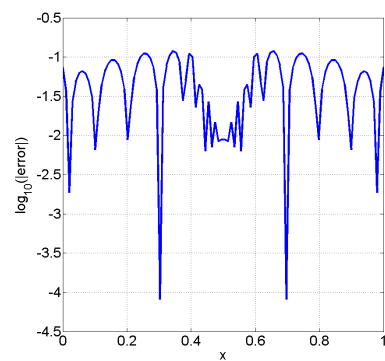
(3.1) Benadering met meer knooppunten op het begin en einde van het interval.



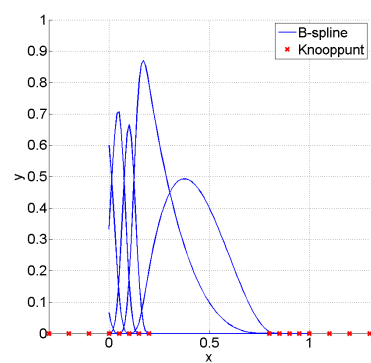
(3.2) Fout tussen de benadering en de echte functie voor meer knooppunten op het begin en einde van het interval.



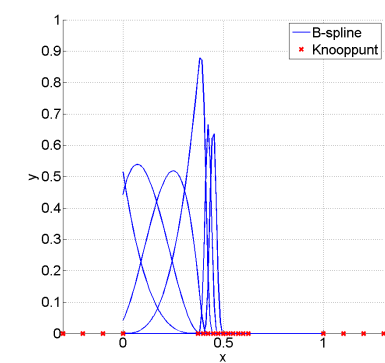
(3.3) Benadering met meer knooppunten in het midden van het interval.



(3.4) Fout tussen de benadering en de echte functie voor meer knooppunten in het midden van het interval.

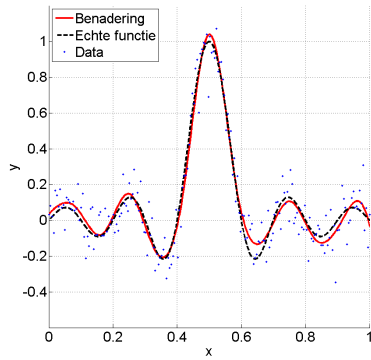


(3.5) B-splines van een benadering met meer knooppunten op het begin en einde van het interval.

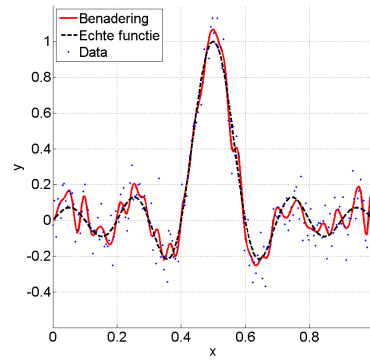


(3.6) B-splines van een benadering met meer knooppunten in het midden van het interval.

Figuur 3: Illustratie van het belang van de ligging van de knooppunten bij het construeren van een spline kleinste-kwadratenbenadering voor een $\text{sinc}(x)$ functie.

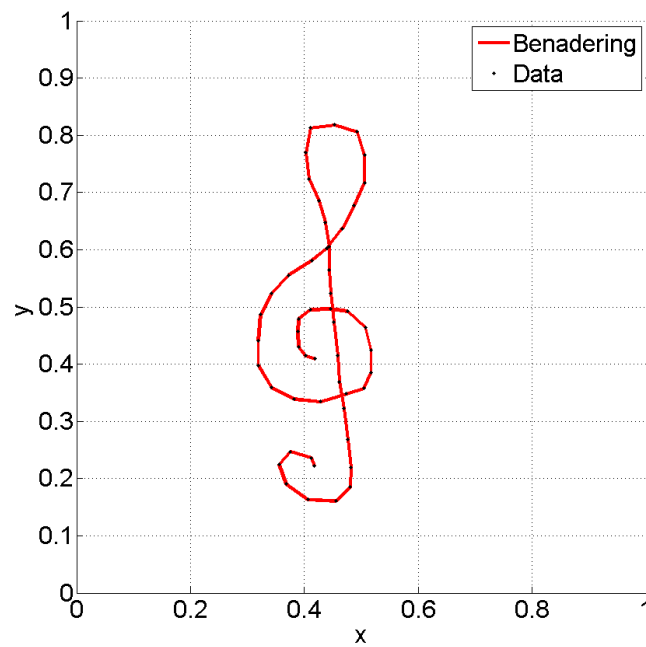


(4.1) Weinig knooppunten



(4.2) Veel knooppunten

Figuur 4: Invloed van de knooppunten op de kleinste-kwadratenbenadering van een $\text{sinc}(x)$ functie met Gaussiaanse ruis.



Figuur 5: Kleinste-kwadratenbenadering van de solsleutel uit de muziekleer.