Numerieke Modellering en Benadering: Practicum 2

Jona Beysens & Arnout Devos

21 mei 2014

Opgave 1

De nulpunten van een orthogonale veelterm gebaseerd op de waarden α_k,β_k en λ_k kunnen gevonden worden door de eigenwaarden te bepalen een n x n tridiagonale matrix *tridag(ν_{k-1}; α_k; μ_{k-1})* met $\nu_k = \frac{\beta_{k+1}}{\lambda_{k+1}}$ en $\mu_k = \frac{1}{\lambda_k}$. Dit volgt uit **stelling 3.2.5 Numerieke Modellering en Benadering**.

De MATLAB-code bevindt zich in bijlage 1.

Opgave 2

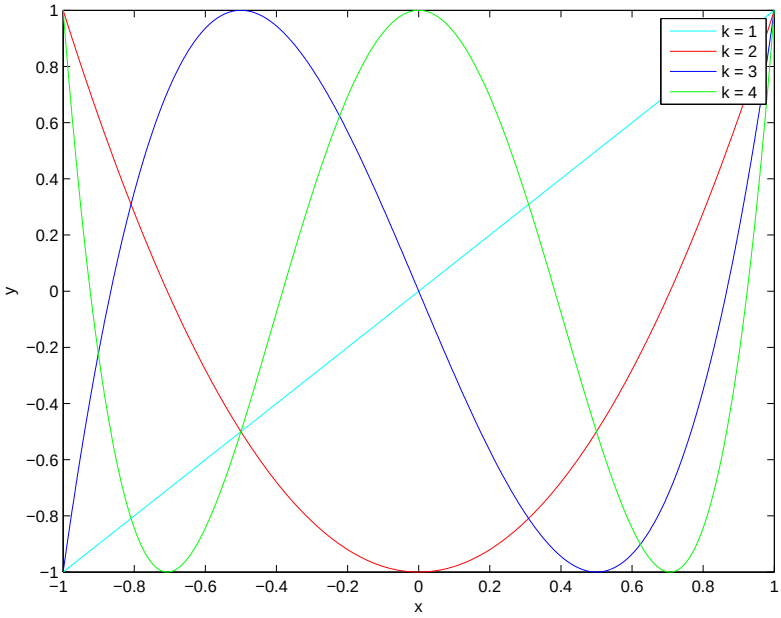
Om de orthogonale veeltermen te evalueren kan efficiënt gebruikt gemaakt worden van de drietermsrecursiebetrekking zoals gegeven in **stelling 3.2.3 Numerieke Modellering en Benadering**. Elk element uit de matrix M wordt bepaald als volgt: $M_{ij} = \psi_j(x_i)$. Aangezien in elke rij dezelfde x_i geëvalueerd moet worden, maken we voor elke rij gebruik van de recursiebetrekking. De werking van de functie wordt getoond door in figuur 1 de Chebyshev veeltermen T_1, T_2, T_3 en T_4 te evalueren in een voldoende aantal punten.

De MATLAB-code bevindt zich in bijlage 2.

Opgave 3

De interpolerende veelterm y_{n-1} voldoet aan volgende betrekking: $y_{n-1} = \sum_{i=0}^{n-1} c_i T_i$. Om de coëfficiënten c_i te vinden moet het stelsel $Mc = B$ met $M_{ij} = \psi_j(x_i)$ en $B_i = f(x_i)$ opgelost worden. Verder moet de veelterm geëvalueerd worden in een stel punten t. Dit wordt bekomen door het *rekenschema van Smith* uit te werken voor dat stel punten. Dit vermijdt de interpolerende veelterm te rangschikken naar opeenvolgende machten van x, d.w.z. de coëfficiënten d_i te bepalen zodat $y_{n-1} = \sum_{i=0}^{n-1} d_i x^i$ om daarna het *rekenschema van Horner* toe te passen.

De MATLAB-code bevindt zich in bijlage 3.



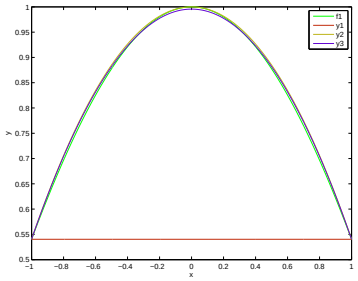
Figuur 1: Chebyshev veeltermen T_1, T_2, T_3 en T_4 van respectievelijk graad $k = 1, \ldots, 4$

Opgave 4

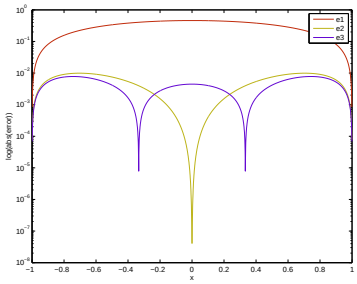
a) In figuur 2 en figuur 3 wordt de functie $f_1(x) = cos(x)$ benaderd door middel van Chebyshev veeltermen. Zowel de interpolerende veeltermen (2a en 3a) alsook de corresponderende fouten (2b en 3b) worden voorgesteld.

In figuur 2 wordt gebruik gemaakt van equidistant verdeelde interpolatiepunten. Hierbij zien we dat de fout afneemt naarmate de graad n stijgt. Vanaf graad 3 valt de interpolerende veelterm bijna volledig samen met de te benaderen functie f_1. Verder zijn de interpolatiepunten duidelijk zichtbaar op figuur 2b. In de interpolatiepunten moet de interpolerende veelterm exact door de te benaderen functie gaan waardoor de fout in die punten gelijk aan 0 is.

In figuur 3 wordt gebruik gemaakt van de nulpunten van de Chebyshev veelterm van graad $n + 1$ als interpolatiepunten. Ook hier verkleint de fout bij stijgende graad n. Verder zijn de interpolatiepunten op te merken in figuur 3b. Door het nemen van deze interpolatiepunten kan de fout verkleind worden. Dit is duidelijk te zien in de fout e_1 horende bij y_1. Bij equidistant verdeelde interpolatiepunten liggen deze voor graad 1 in de randpunten van het interval. Dit leidt tot een grote fout in het midden van het interval. Bij de nulpunten van de Chebyshev veelterm van graad $n + 1$ is dit niet het geval. De evolutie van de maximale fout in functie van graad n wordt ook voorgesteld in figuur 6a. Vanaf graad $n = 15$ neemt de fout bij equidistante interpolatie punten niet meer af.



(a) Interpolatie door $y_n(t)$



(b) De fout $e_n = |y_n(t) - f(t)|$

Figuur 2: De functie $f_1(t) = cos(t)$ wordt benaderd in equidistant verdeelde interpolatiepunten door middel van Chebyshev veeltermen y_1, y_2 en y_3 van respectievelijk graad $n = 1, \ldots, 3$

.

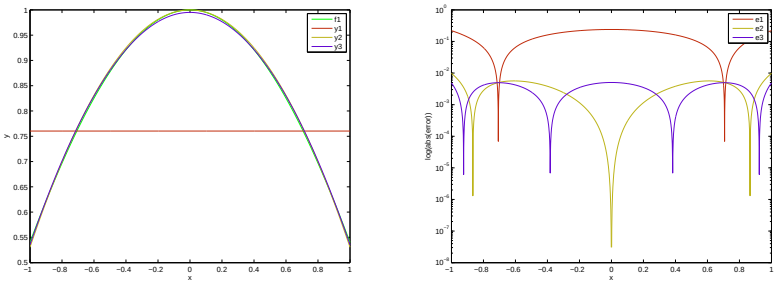
b) In figuur 4 en figuur 5 wordt de functie $f_2(t) = \frac{1}{1+6t^2}$ benaderd door middel van Chebyshev veeltermen. Zowel de interpolerende veeltermen (4a en 5a) alsook de corresponderende fouten (4b en 5b) worden voorgesteld. In dit geval worden Chebyshev veeltermen van graad $n = 1, \ldots, 9$ gebruikt in plaats van $n = 1, \ldots, 3$ aangezien de functie f_2 moeilijker te benaderen is. Bij gebruik van equidistant verdeelde interpolatiepunten wordt de fout bij stijgende graad n vanaf een bepaalde graad niet meer kleiner (zie figuur 4b). Als we de nulpunten van de Chebyshev veelterm van graad $n + 1$ nemen verkleint de fout wel(zie figuur 5b).

   De evolutie van de maximale fout in functie van graad n wordt ook voorgesteld in figuur 6b. Vanaf graad $n = 5$ neemt de fout bij equidistante interpolatiepunten niet meer af. Het is algemeen zo dat de fout bij hogere graad bij equidistante interpolatiepunten niet meer verkleint. Dit komt doordat er sterke oscillatie optreedt tussen de interpolatiepunten aan de randen van het interval. Dit fenomeen is gekend als het *Runge's phenomenon.*

Opgave 5

Het conditiegetal κ van de matrix M wordt voorgesteld in figuur 7. Het conditiegetal is een maat voor orthogonaliteit. Het conditiegetal stijgt bij gebruik van equidistante interpolatiepunten en blijft klein en constant bij gebruik van nulpunten van de Chebyshev veelterm. Dit geeft een indicatie dat de geëvalueerde Chebyshev veeltermen in equidistant verdeelde interpolatiepunten $\psi_j(x_i)$ niet orthogonaal zijn ten op zichte van een bepaalde gewichtsfunctie. Bij het gebruik van de nulpunten van de Chebyshev veelterm is dit wel het geval.

Een groot conditiegetal betekent dat het resultaat van een kleine perturbatie op de elementen van M sterk kan afwijken van het oorspronkelijk resultaat. De coëfficiënten c_i zullen dus sterk kunnen veranderen bij een kleine



(a) Interpolatie door $y_n(t)$ (b) De fout $e_n = |y_n(t) - f(t)|$

Figuur 3: De functie $f_1(t) = cos(t)$ wordt benaderd in de nulpunten van de Chebyshev-veelterm van graad $n + 1$ door middel van Chebyshev veeltermen y_1, y_2 en y_3 van respectievelijk graad $n = 1, \ldots, 3$

.

wijziging op de interpolatiepunten. Hierdoor zal de veelterm in de intervallen rond de interpolatiepunten kunnen wijzigen. In de interpolatiepunten zal de wijziging ongeveer gelijk zijn aan de perturbatie aangezien de interpolatievoorwaarde $y_n(x_i) = f(x_i)$ nog steeds voldaan moet worden.

Als we aannemen dat het stelsel in MATLAB achterwaarts stabiel wordt opgelost, dan hangt de nauwkeurigheid van de interpolerende veelterm af van het conditiegetal. Dit wordt uiteengezet in **Theorem 15.1 Trefethen and Bau**. Als het conditiegetal stijgt, dan zal de nauwkeurigheid zakken en dus de fout op de coëfficiënten vergroten. Dit betekent niet noodzakelijk dat ook de nauwkeurigheid van de interpolerende veelterm zal verminderen. Voor equidistant verdeelde interpolatiepunten stijgt het conditie getal maar toch neemt de interpolatiefout af tot en met graad $n = 15$ (zie figuur 6a). Vanwege de moeilijker te benaderen functie f_2 neemt de nauwkeurigheid van bijhorende interpolerende veelterm wel af (zie figuur 6b).

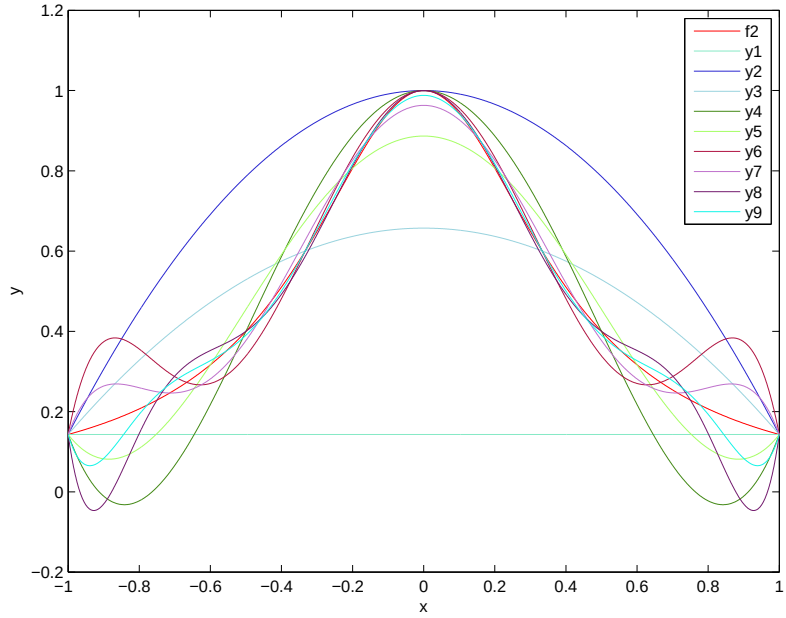
Opgave 6

De opbouw van het algoritme is als volgt:

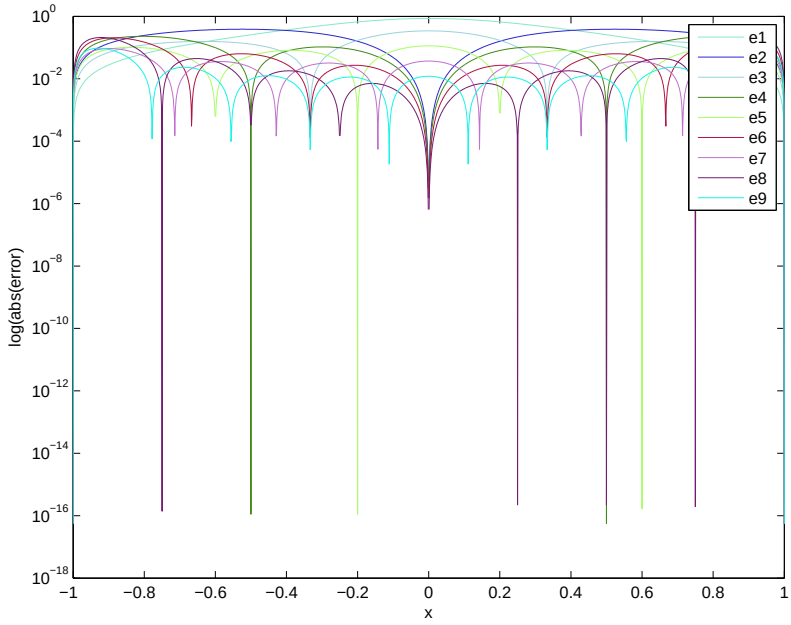
Eerst wordt het aantal rijen N gezocht van de matrix x om vervolgens de N-punts FFT uit te voeren.Vervolgens worden de coëfficiënten $X_{K+1}, \ldots, X_{\frac{N}{2}}$ gelijk aan nul gesteld. In diezelfde operatie worden ook meteen de corresponderende X_{N-k} coëfficiënten gelijk aan nul gesteld vanwege de symmetrie. Om de oorspronkelijke functie te kunnen evalueren in een aantal punten $M \geq N$, worden de volgende relaties gebruikt:

$$\begin{cases} Y_k = \frac{M}{N} X_k & k = 0, \ldots, \frac{N}{2} - 1 \\ Y_k = \frac{M}{N} \frac{X_k}{2} & k = \frac{N}{2} \\ Y_k = 0 & k = \frac{N}{2} + 1, \ldots, \frac{M}{2} \end{cases} \tag{1}$$

Tenslotte kan via de M-punts inverse FFT de benadering voor het oorspronkelijke signaal gevonden worden.



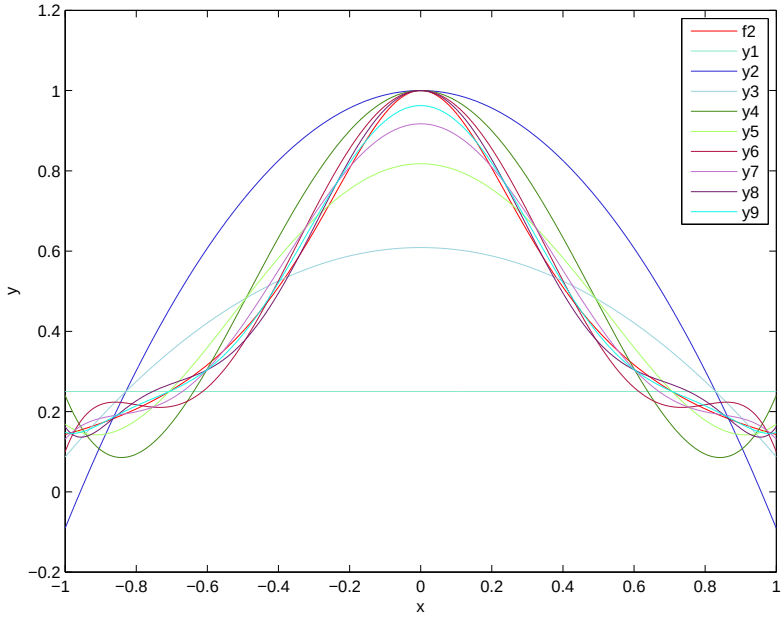
(a) Interpolatie door $y_n(t)$



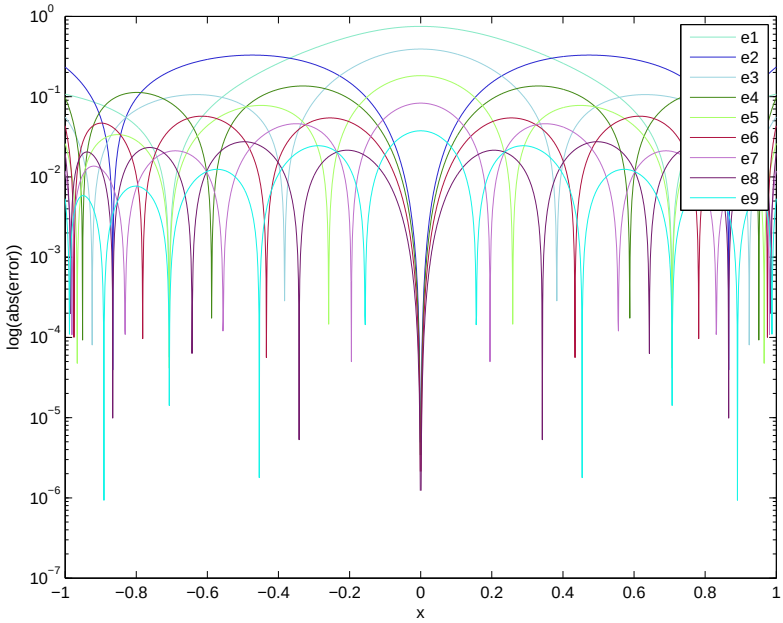
(b) De fout $e_n = |y_n(t) - f(t)|$

Figuur 4: De functie $f_2(t) = \frac{1}{1+6t^2}$ wordt benaderd in equidistant verdeelde interpolatiepunten door middel van Chebyshev veeltermen $y_1, \ldots, y_9$ van respectievelijk graad $n = 1, \ldots, 9$

.



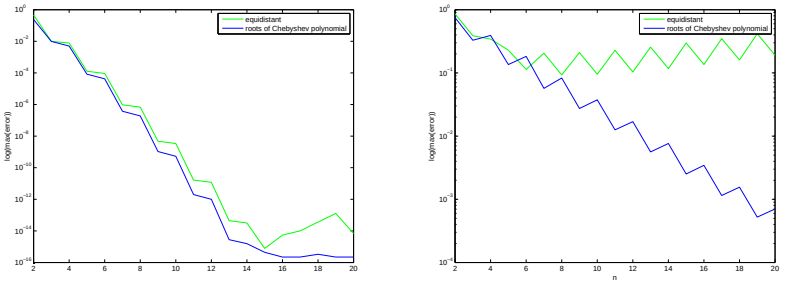
(a) Interpolatie door $y_n(t)$



(b) De fout $e_n = |y_n(t) - f(t)|$

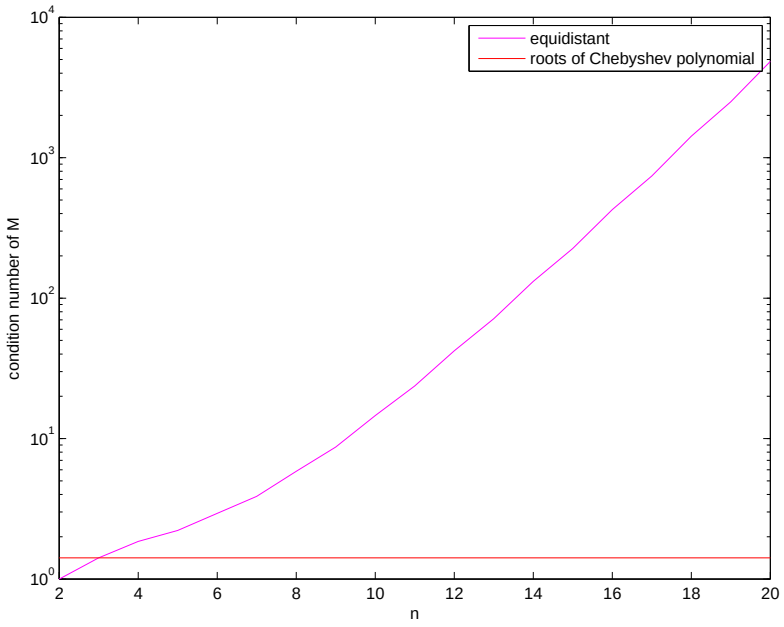
Figuur 5: De functie $f_2(t) = \frac{1}{1+6t^2}$ wordt benaderd in in de nulpunten van de Chebyshev-veelterm van graad $n+1$ door middel van Chebyshev veeltermen $y_1, \ldots, y_9$ van respectievelijk graad $n = 1, \ldots, 9$

.



(a) benadering van $f_1(t) = cos(t)$ (b) benadering van $f_2(t) = \frac{1}{1+6t^2}$

Figuur 6: maximale fout $|y_n(t) - f(t)|$ in functie van stijgende graad n voor equidistant verdeelde punten en nulpunten van de Chebyshev-veelterm van graad $n+1$



Figuur 7: conditiegetal κ van de matrix M in functie van stijgende graad n

De MATLAB-code bevindt zich in bijlage 4.

Opgave 7

Wanneer alleen de K eerste X_k coëfficiënten behouden worden, worden de hogere frequenties uit het signaal weggehaald. Dit komt overeen met een laagdoorlaatfiltering van het signaal. Er gaat enkel informatie verloren als er wel degelijk frequenties aanwezig zijn in het oorspronkelijke signaal die coëfficiënten $X_{K+1}, \ldots, X_{\frac{N}{2}}$ verschillend van nul veroorzaken.

Als voorbeeld beschouwen we een signaal in functie van de tijdsparameter t samengesteld uit 10 verschillende frequenties

$$c = \sum_{k=1}^{10} \sin 2\pi kt \tag{2}$$

Wanneer dit signaal in 512 punten gesampled wordt en er daarna de FFT van genomen wordt, bekomt men Figuur 9a. Hetzelfde resultaat wordt bekomen door $K = 256$ te nemen. Noteer dat de FFT er voor zorgt dat er ook schijnbaar frequenties aanwezig zijn bij $11, 12, \ldots$. De waarden verschillend van nul bij deze frequenties ontstaan door de keuze van het venster en worden in de verdere behandeling verwaarloosd.

Als $K = 10$ wordt genomen, worden er geen nuttige frequenties verwijderd uit het signaal en bijgevolg is het benaderende signaal exact gelijk aan het oorspronkelijke signaal zoals te zien is in Figuur 9b.

De laagdoorlaatfiltering wordt pas echt duidelijk wanneer $K = 5$ wordt gesteld. Als gevolg van deze keuze van K verdwijnt de helft van de frequenties in de benadering zoals te zien in de rechterfiguur van Figuur 8c. Aangezien de hogere frequenties eenvoudigweg op nul gezet worden, verdwijnt ook de energie die in deze frequenties vervat zit wat resulteert in een signaal met een minder grote amplitude. Ook in de linkerfiguur van Figuur 8c is te zien dat er een veel *geleidelijker* verloop is van de veelterm.

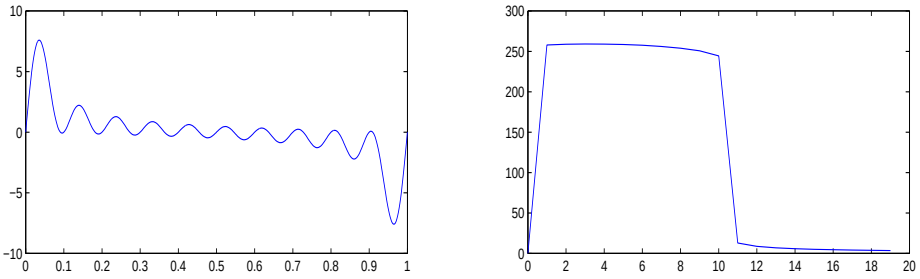
Opgave 8

In Figuur 9 wordt een benadering geconstrueerd voor het ∞ teken aan de hand van de functies *click()* en *periotrig(x,K,M)*. In de linkse deelfiguur zijn de oorspronkelijke punten weergegeven die met *click()* zijn getekend. De rechterfiguur toont een periodieke interpolerende en benaderende trigonometrische veelterm voor deze punten.

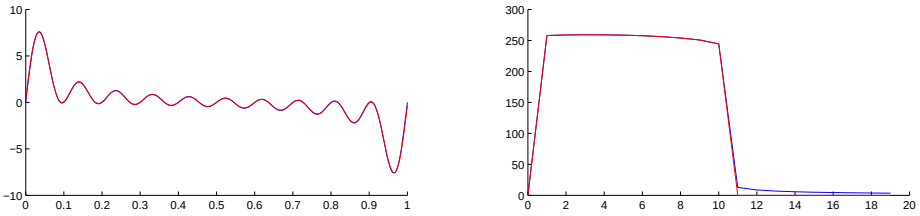
Aangezien het teken ∞ geschreven kan worden als een speciaal geval (met name *het lemniscaat van Gerono*) van de meer algemene *Lissajousfiguur*, is deze benadering uiterst geschikt omdat de parametrisaties bij een *Lissajousfiguur* van de volgende vorm zijn:

$$\begin{cases} x = \sin(\alpha t + \delta) \\ y = \sin(\beta t) \end{cases} \tag{3}$$

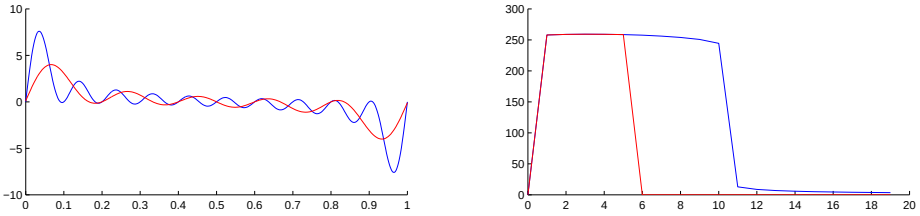
Wanneer in deze vergelijkingen de waarden $\alpha = 1$,$\beta = 2\alpha = 2$ en $\delta = 0$ worden genomen, verkrijgt men Figuur 10.



(a) Oorspronkelijke functie met $K = 256$



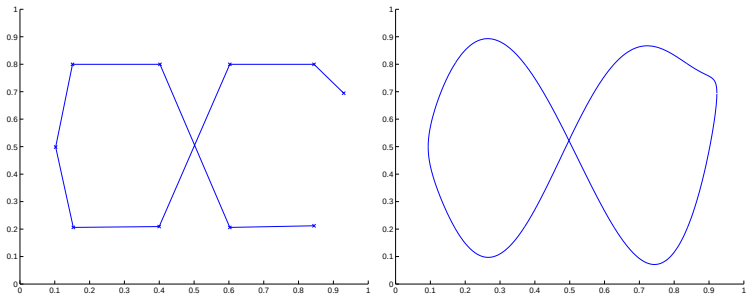
(b) Benaderende functie met $K = 10$



(c) Benaderende functie met $K = 5$

Figuur 8: Illustratie van de invloed van de parameter K op de benadering (rood) van een periodieke functie (blauw) met aan de linkerkant de veelterm en aan de rechterkant de FFT van deze veelterm weergegeven over een nuttig domein. ($M = N = 512$)

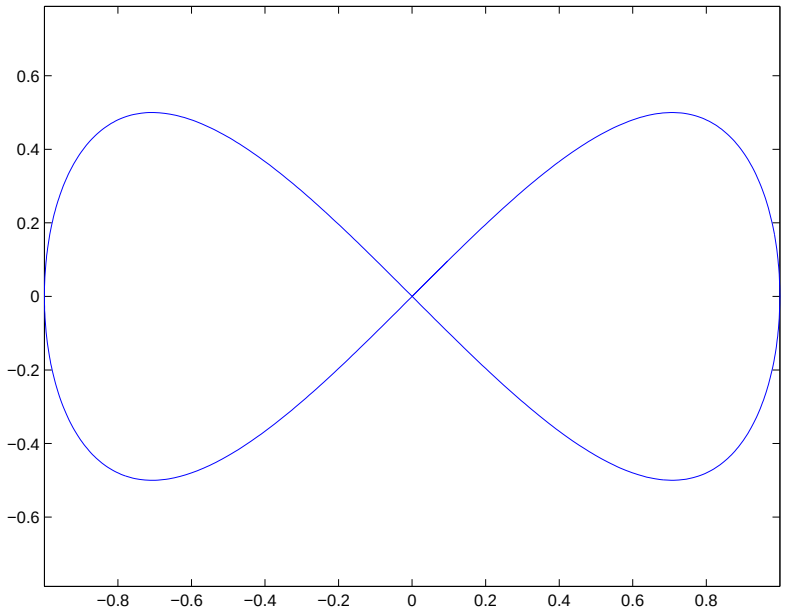
.



(a) Punten met *click()* (b) Benaderende functie

Figuur 9: ∞ teken benaderd

.



Figuur 10: Het ∞ teken onder de vorm van een Lissajousfiguur

Bijlage 1

```
function [x] = poly_zeros(n,alpha,beta,lambda)
% poly_zeros functie berekent de nulpunten van een orthogonale veelterm
% gebaseerd op de waarden alpha, beta en lambda uit de recursiebetrekking
%   n = de graad van de veelterm waarvan we de nulpunten zoeken
%   alpha = kolom vector van lengte n e.g. [a1 a2 a3]
%   beta = kolom vector van lengte n e.g. [b1 b2 b3]
%   lambda = kolom vector van lengte n+1 e.g. [l0 l1 l2 l3]
%   x = kolom vector van lengte n die de nulpunten van de veelterm bevat

% elementen van alpha op de diagonaal van A plaatsen
A = diag(alpha,0);
mu = 1./lambda(2:n+1);
nu = beta./lambda(2:n+1);
% elementen van mu en nu op respectievelijk de 1e en -1e diagonaal van A
% plaatsen
A = A + diag(mu(1:n-1),1) + diag(nu(2:n),-1);
% de eigenwaarden van de matrix A leveren de nulpunten van de veelterm
x = eig(A);
end
```

Bijlage 2

```
function M = eval_recursion(x,n,alpha,beta,lambda)
% eval_recursion functie evalueert de veeltermen gekarakteriseerd door
% alpha, beta en lambda in de opgegeven punten van x
%   x = kolom vector van lengte n die de waarden waarin gevalueerd moet worden
%       bevat
%   n = aantal punten waarin gevalueerd moet worden
%   alpha = kolom vector van lengte n e.g. [a1 a2 a3]
%   beta = kolom vector van lengte n e.g. [b1 b2 b3]
%   lambda = kolom vector van lengte n+1 e.g. [l0 l1 l2 l3]
%   M = vierkante matrix van dimensie n die de gevalueerde punten bevat

M = zeros(n,n);
M(:,1) = lambda(1);
lambda = lambda(2:n);
% De evaluatie van de veeltermen gebeurt ahv de drietermsrecursiebetrekking
for i = 1:n
    M(i,2) = x(i);
    for j = 3:n;
        M(i,j) = lambda(j-1)*(x(i)-alpha(j-1))*M(i,j-1)-beta(j-1)*M(i,j-2);
    end
end
end
```

Bijlage 3

```
function [y,M] = interpolate(x,f,alpha,beta,lambda,t)
% interpolate functie berekent de waarden van de interpolerende veelterm in
% elementen van t door middel van het rekenschema van smith
%   x = kolom vector van lengte n die de interpolatiepunten bevat
%   f = kolom vector van lengte n die de waarden van de functie in x bevat
%   alpha = kolom vector van lengte n e.g. [a1 a2 a3]
%   beta = kolom vector van lengte n e.g. [b1 b2 b3]
%   lambda = kolom vector van lengte n+1 e.g. [l0 l1 l2 l3]
%   t = kolom vector die de waarden bevat waarin de interpolerende veelterm
%       gevalueerd moet worden
%   y = waarden van de interpolerende veelterm in t
n = length(x);
M = eval_recursion(x,n,alpha,beta,lambda);
% los het stelsel op om de cofficinten te bepalen
c = M\f;
% bereken voor elke waarde van de vector t de waarde van de interpolerende
% veelterm van graad m door middel van het rekenschema van smith
% B bevat op kolom k alle cofficinten b die horen bij een enkele waarde
% van de vector t
m = n-1;
y = zeros(length(t),1);
lambda = lambda(2:m+1);
B = zeros(m+1,length(t));
for k = 1:length(t)
   y(k) = lambda(1);
   B(m+1,k) = c(m+1);
   B(m,k) = c(m)+lambda(m)*(t(k)-alpha(m))*B(m+1,k);
        for i = (m-1):-1:1
          B(i,k) = c(i)+lambda(i)*(t(k)-alpha(i))*B(i+1,k)-beta(i+1)*B(i+2,k);
        end
   y(k) = B(1,k)*y(k);
end
end
```

Bijlage 4

```
function y = periotrig(x,K,M)
%periotrig Functie voor het evalueren van periodieke interpolerende en
%   benaderende trigonometrische veeltermen
Y   = fft(x);
%t   = 0:1/M:(M-1);
[N d]   = size(x);

% Coefficienten verwijderen aan de hand van K
% Y_k = 0 for k = K+1,...,N/2 if K < N/2
Y(K+2:N-K)=0;

% Y_k = X_k*M/N for k = 0,...,N/2-1
Y(1:N/2,:) = Y(1:N/2,:)*M/N;
% Y_k = X_k*M/(N*2) for k = N/2
Y(N/2+1,:)=Y(N/2+1,:)*M/(N*2);
% Y_k = 0 for k = N/2+1,... ,M/2
Y(N/2+2:M/2+1,:)=0;
% Alleen de nodige coefficienten overhouden
Y = Y(1:M/2+1,:);

% Steunen op de symmetrie
%Y(M/2+1:M-1,:) = conj(Y(M/2:2,:));

for k = M/2+1:M-1
    Y = [Y; conj(Y(M-k+1,:))];
end
y = ifft(Y,M);
end
```