Numerieke Modellering en Benadering

De Wolf Peter
Vekemans Wout

23 mei 2014

Lijst van figuren

Benaderen met veeltermen

In dit practicum willen we eerst enkele functies benaderen met behulp van veeltermen door interpolatie. We gebruiken hiertoe nulpunten van orthogonale veeltermen als interpolatiepunten.

Berekenen van nulpunten van orthogonale veeltermen

Zie bijlage `poly_zeros.m`
Deze functie berekent de nulpunten van de orthogonale veeltermen gebaseerd op de waarden α_k, β_k, en λ_k uit de recursiebetrekking voor orthogonale veeltermen (1). Het doet dit door een tridiagonale matrix op te stellen met elementen afgeleid van de coëfficienten uit die recursiebetrekking. De nulpunten van de veelterm zijn dan gelijk aan de eigenwaarden van die matrix.

$$\phi_0(x) = \lambda_0 \; ; \; \phi_1(x) = \lambda_1(x - \alpha_1)\phi_0(x)) \tag{1a}$$
$$\phi_k(x) = \lambda_k(x - \alpha_k)\phi_{k-1}(x)) - \beta_k\phi_{k-2}(x) \tag{1b}$$

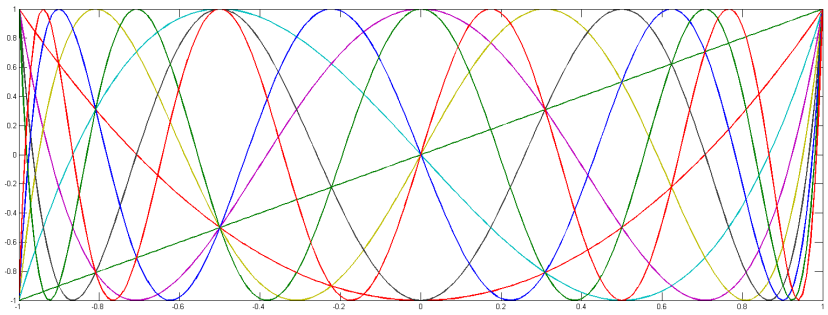
Evauleren van drietermsrecursiebetrekking

Zie bijlage `eval_recursion.m`
Deze functie evalueert de drietermsrecursiebetrekking in de punten gegeven in de vector x. Als we deze methode toepassen op de 10 eerste Chebychev veeltermen krijgen we figuur 1. Deze functie is eenvoudig toe te passen omdat Chebychev veeltermen duidelijk voldoen aan vergelijking (1). Dit is makkelijk in te zien als men in de recursiebetrekking volgende waarden invult: $\lambda_0 = 1, \lambda_1 = 1$ en alle andere λ_k gelijk aan 2. α_k is gelijk aan 0, en β_k is 1.

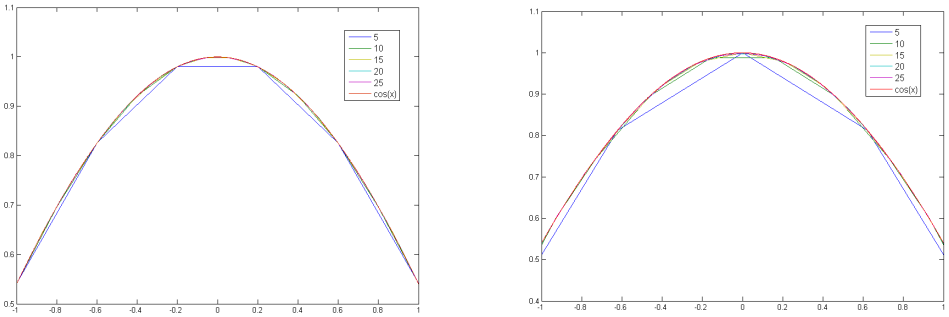
Berekenen van interpolerende veelterm

Zie bijlage `interpolate.m`
Dit algoritme berekent een interpolerende veelterm door de waarden $f(x)$ en evalueert die veelterm in de punten gegeven in t. Als we dit toepassen op respectievelijk $\cos(x)$ en $\frac{1}{1+6x^2}$ krijgen we grafiek 2 en 3. We zien op het eerste zicht niet duidelijk verschil tussen het gebruik van equidistante interpolatiepunten en de nulpunten van de Chebychev-veelterm T_n.

Bij een verdere analyse van de interpolatiefout blijkt echter dat de nulpunten van de Chebychev veeltermen een veel beter gespreide fout geven, de fout is



Figuur 1: Evaluatie van eerste tien Cheybchev veeltermen op 200 punten in [-1,1]



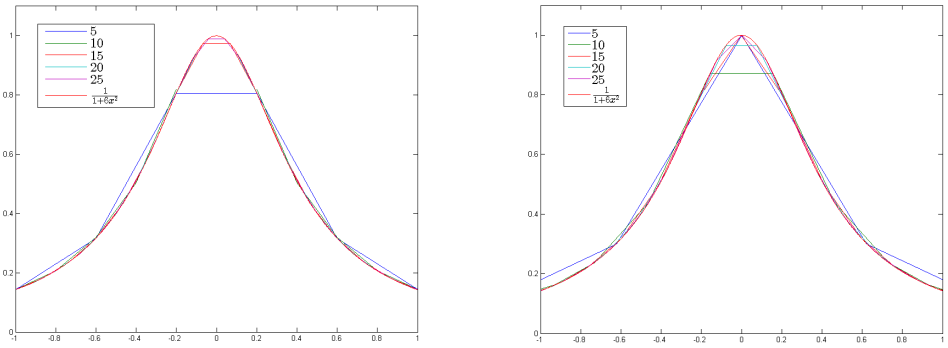
(a) Equidistant

(b) Nulpunten van T_n

Figuur 2: Enkele interpolerende veeltermen van $\cos(x)$

ongeveer overal even groot. Voor illustratie, zie figuur 4. Bij de equidistante punten is er aan de rand van het interval een veel grotere fout aanwezig. Bij benaderingen van lagere graad is de absolute fout wel kleiner bij de equidistante punten. Het verschil aan de rand van het interpolatie-interval wordt veroorzaakt door de locatie van de interpolatiepunten. Als de nulpunten van T_n worden gebruikt, liggen die veel meer geconcentreerd aan de rand van het interval, waardoor de fout aan de rand ongeveer even groot wordt als in het midden van het interval. Dit zorgt voor een fout die gemiddeld hoger is, maar minder hoge pieken heeft dan bij equidistante interpolatiepunten.

Als we de maximale fout in functie van de graad van de benadering uitzetten zien we ook dat bij hogere graad de equidistante punten weer onderdoen. We merken zelfs een stijging in de maximale fout vanaf een bepaalde punt. De Chebychev interpolatie daarentegen daalt in het begin ongeveer even snel,



(a) Equidistant (b) Nulpunten van T_n

Figuur 3: Enkele interpolerende veeltermen van $\frac{1}{1+6x^2}$

maar blijft bij hogere graad constant rond ϵ_{mach}. Dit wordt duidelijk op figuur 5. De stijgende śtaartv́an de grafiek voor equidistante punten wordt veroorzaakt door de zeer slechte benadering aan de rand van het interval.

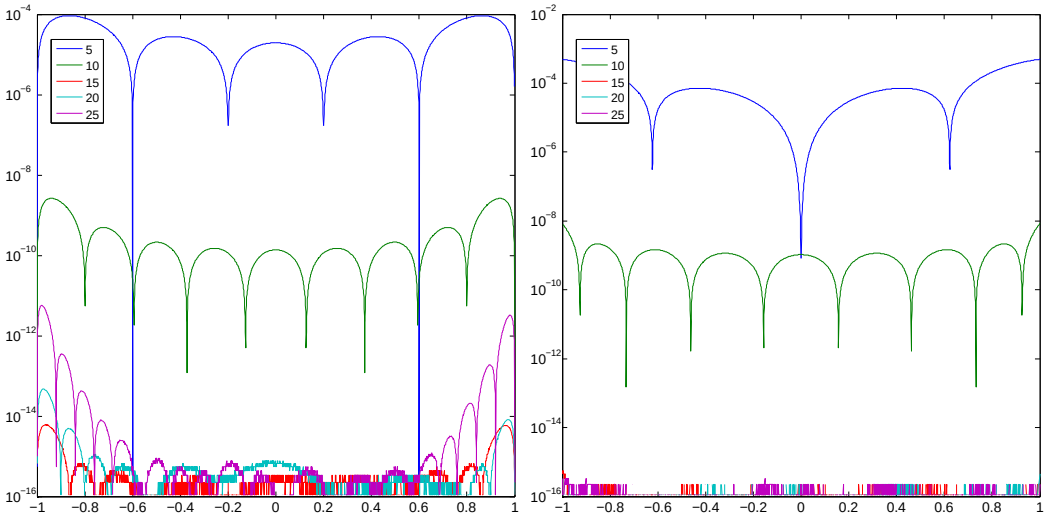
Tot slot bestuderen we het conditiegetal van de matrix M die gebruikt wordt in `eval_recursion.m`. Als we dit voor beide methodes (equidistant en Chebychev) tonen in functie van de graad van benadering, zien we dat $k(M)$ exponentiëel stijgt voor beide methodes, maar minder snel bij equidistante interpolatiepunten. Bij een interpolatie van graad 25 is k al van grootte-orde 10^7 voor interpolatie in de nulpunten van T_n. Dit wil zeggen dat de interpolerende curve zelfs bij benaderingen van relatief lage graad al zeer gevoelig is aan kleine afwijkingen op de opgemeten waarden van $f(x)$. De toename van het conditiegetal wordt geïllustreerd in figuur 6.

Benaderen met trigonometrische veeltermen

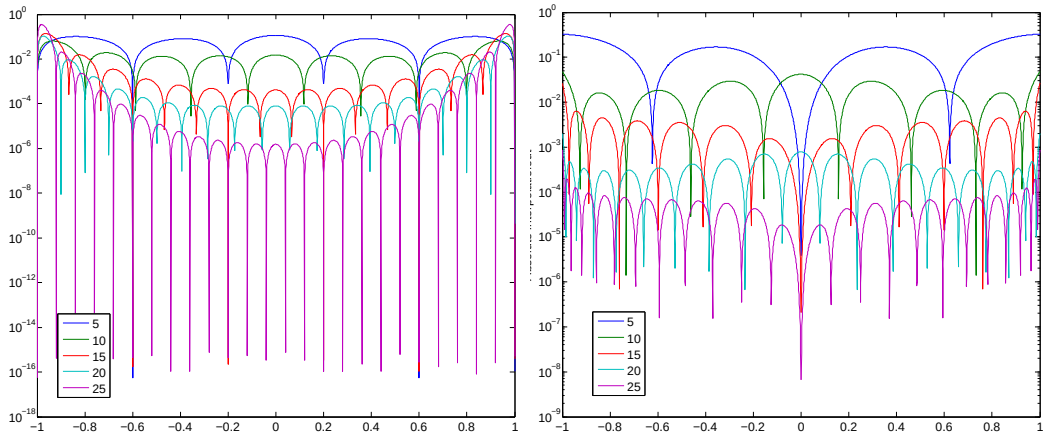
Zie bijlage `periotrig.m`.

Invloed van de parameter K op de benadering

Als te benaderende functie nemen we $\sin(t) + \cos(2t) + \sin(4t) + \cos(8t) + \sin(16t) + \cos(32t)$. Deze functie wordt benaderd door trigoniometrische functies. De K-waarde geeft de indexwaarde van de laatste X_k-coëfficient die

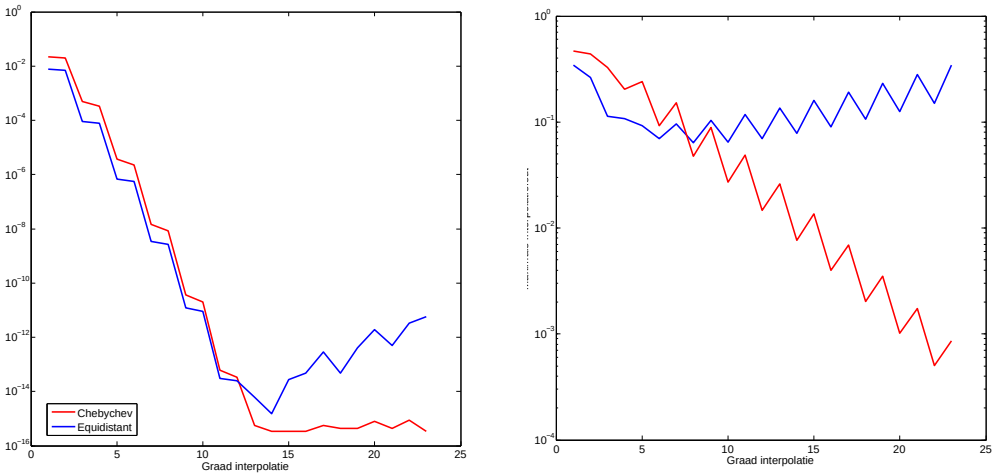


(a) Equidistant, $\cos(x)$ (b) Nulpunten van T_n, $\cos(x)$



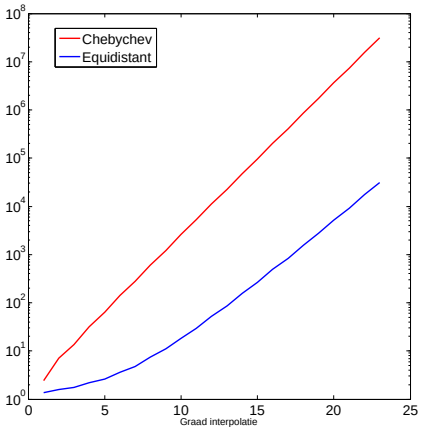
(c) Equidistant, $\frac{1}{1+6x^2}$ (d) Nulpunten van T_n, $\frac{1}{1+6x^2}$

Figuur 4: Absolute fout bij het interpoleren

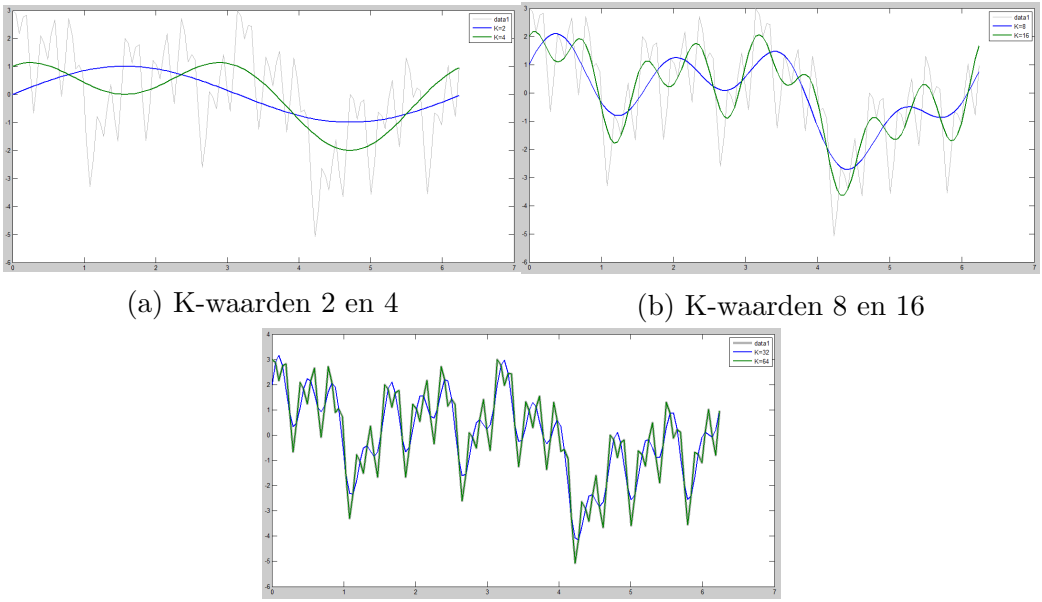


(a) $\cos x$ (b) $\frac{1}{1+6x^2}$

Figuur 5: Maximale interpolatiefout



Figuur 6: Conditiegetal van M



(a) K-waarden 2 en 4

(b) K-waarden 8 en 16

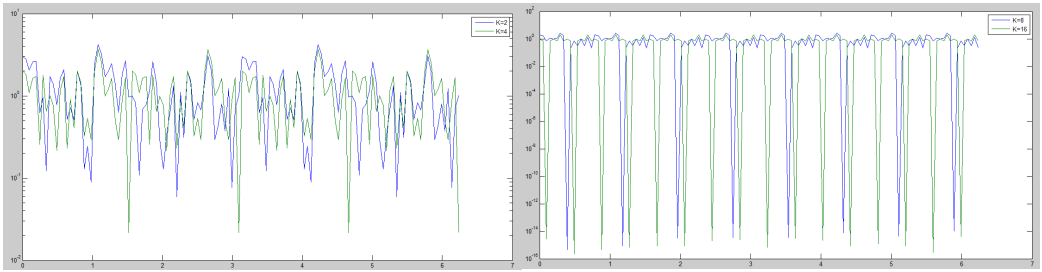
(c) K-waarden 32 en 64

Figuur 7: De benaderingen

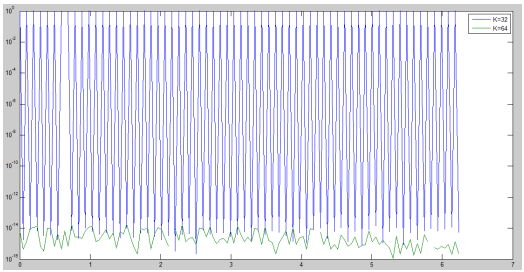
behouden wordt opgegeven. Hoe lager de K-waarde, hoe minder termen er in de benadering aanwezig zijn. Een hogere K-waarde zal dus een betere benadering opleveren. Dit is ook te zien op de figuren. Figuur 7 toont de functie en de verschillende benaderingen. Figuur 8 geeft de norm van de fout weer van elke benadering. Hier is ook te zien dat bij een K-waarde van 64 de norm van de fout schommelt rond 10^{-16}. De fout ligt dus heel dicht bij de machinenauwkeurigheid. Er is dan ook geen verschil meer te zien op de figuren waar de functie benadert wordt voor een K-waarde van 64.

Benaderen van een gesloten kromme

De click-functie geeft als output een vector met X-waarden en een vector met Y-waarden. Deze 2 vectoren kunnen gebruikt worden als input van de periotrig functie. In de figuren is te zien dat het uitvoeren van de periotrig functie met een hogere K hier een slechtere benadering geeft voor het symbool. De hoofdreden hiervoor is dat de punten handmatig worden ingegeven en niet nauwkeurig zijn. Een hogere orde benadering zal nauwkeuriger door de punten gaan waardoor de bekomen figuur minder goed op een oneindig-symbool lijkt dan wanneer de benadering met een lagere orde gebeurt. Indien de punten wel goed liggen geeft een benadering met een hoge K-waarde echter nog wel een goed beeld als benadering van het oneindig-symbool.

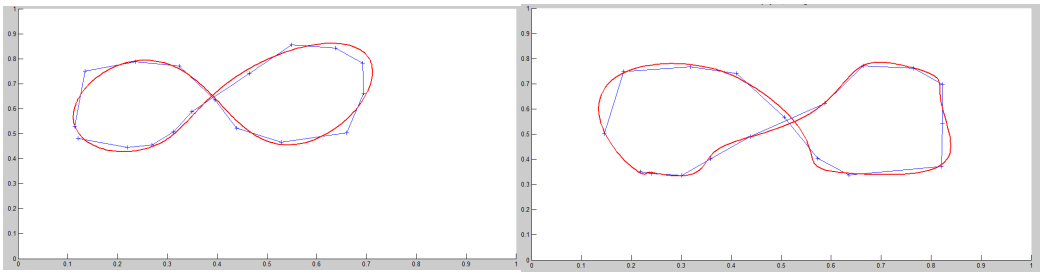


(a) K-waarden 2 en 4 (b) K-waarden 8 en 16

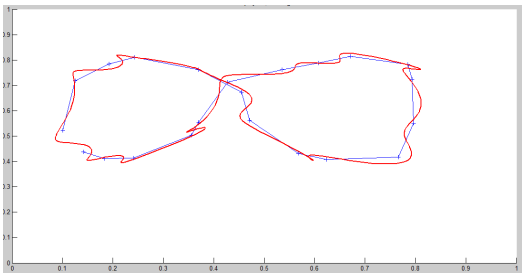


(c) K-waarden 32 en 64

Figuur 8: De norm van de fout



(a) K-waarden 2 en 4 (b) K-waarden 8 en 16



(c) K-waarden 32 en 64

Figuur 9: Benaderingen voor het oneindig-symbool

Bijlagen

```
function x = poly_zeros(n, alpha, beta, lambda)

%maak diagonaalmatrix met de elementen van alpha
alphaMat = diag(alpha);

%maak matrices met mu en nu elementen op respectievelijk eerste boven- en
%onderdiagonaal
mu=zeros(n-1,1);
nu=zeros(n-1,1);
for i=1:n-1
    nu(i) = beta(i+1)/lambda(i+1);
    mu(i) = 1/lambda(i);
end
muMat = diag(mu,1);
nuMat = diag(nu,-1);
%maak matrix A
A= alphaMat+nuMat+muMat;

%bereken de eigenwaarden van A en dus de nulpunten van de veelterm
x=eig(A);
```

```
function M = eval_recursion(x, n, alpha, beta, lambda)

m = length(x);
M=zeros(m,n);
for i=1:m %itereer over rijen
    M(i,1)=lambda(1);
    M(i,2)=lambda(2)*(x(i)-alpha(1))*M(i,1);
    for j=3:n %itereer over kolommen
        M(i,j)=lambda(j)*(x(i)-alpha(j))*M(i,j-1)-beta(j)*M(i,j-2);
    end
end
```

```
function [y,conditie] = interpolate(x, f, alpha, beta, lambda, t)

b=[];
n=length(alpha);
%maak M matrix
```

```
M=eval_recursion(x,length(alpha),alpha,beta,lambda);
conditie=cond(M);

%bereken de coefficienten van de interpolerende veelterm
c=M\f;

%bereken de waarden van de interpolerende veelterm in t
for j=1:length(t)
    b(n+2,1)=0;
    b(n+1,1)=0;
    b(n) = c(n);
    if(n>1)
       b(n-1,1) = c(n-1) + lambda(n,1)*(t(j)-alpha(n,1))*b(n);
    end
    for i=n-2:-1:1
        b(i,1) = c(i,1) + b(i+1,1)*lambda(i+1,1)*(t(j)-alpha(i
           +1,1)) - b(i+2,1)*beta(i+2,1);
    end
    y(j,1) = b(1,1)*lambda(1,1);
end
```

```
function y = periotrig(x, K, M)

    %Geeft een periodische benadering terug van de kolommen van
       x

    %Parameters:
    % x: Een N x d matrix waarbij N even moet zijn
    % K: De graad van de trigonioetrische veelterm.
    % M: Het nieuwe aantal interpolatiepunten.

    [N, O] = size(x);

    if(mod(N, 2) ~= 0)
        disp('N must be even');
        return;
    end;

    %Maak de nieuwe matrix aan waar alle nieuwe interpolatie
       punten
    %inpassen.
    y = zeros(M, O);

    for col = 1:O
        Xk = fft(x(:,col)); %Bereken de fourrier cofficienten
           van x.
        i = linspace(1, 0, M+1)';
```

```
24          i = i(1:M);
25
26          %Berekent de sommatie van formule (3) van de opgave.
27          for k = 1:K-1
28              y(:,col) = y(:,col) + real(Xk(k+1)) * cos(2*pi*k*i)
                    + imag(Xk(k+1)) * sin(2*pi*k*i);
29          end;
30
31          y(:,col) = y(:,col) * 2; %Factor 2 in rekening brengen
32          y(:,col) = y(:,col) + Xk(1); %X0 term van formmule (3)
                in rekening brengen.
33
34          if(K == N/2)
35              y(:,col) = y(:,col) + Xk(N/2 + 1) * cos(pi*N*i);
36          end;
37
38          y(:,col) = y(:,col) / N; %Het geheel delen door N
39          %Deze y term is nu gelijk aan de yterm van formule (3)
40      end
41  end
```