Numerieke Modellering en Benadering

De Wolf Peter
Vekemans Wout

13 mei 2014

Inhoudsopgave

Lijst van figuren

Benaderen met veeltermen

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

Berekenen van nulpunten van orthogonale veeltermen

```
1  function x = poly_zeros(n, alpha, beta, lambda)
2
3  %maak diagonaalmatrix met de elementen van alpha
4  alphaMat = diag(alpha);
5
6  %maak matrices met mu en nu elementen op respectievelijk eerste
       boven- en
7  %onderdiagonaal
8  mu=zeros(n-1,1);
9  nu=zeros(n-1,1);
10 for i=1:n-1
11     nu(i) = beta(i+1)/lambda(i+1);
12     mu(i) = 1/lambda(i);
13 end
14 muMat = diag(mu,1);
15 nuMat = diag(nu,-1);
16 %maak matrix A
17 A= alphaMat+nuMat+muMat;
18
19 %bereken de eigenwaarden van A en dus de nulpunten van de
       veelterm
20 x=eig(A);
```

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.

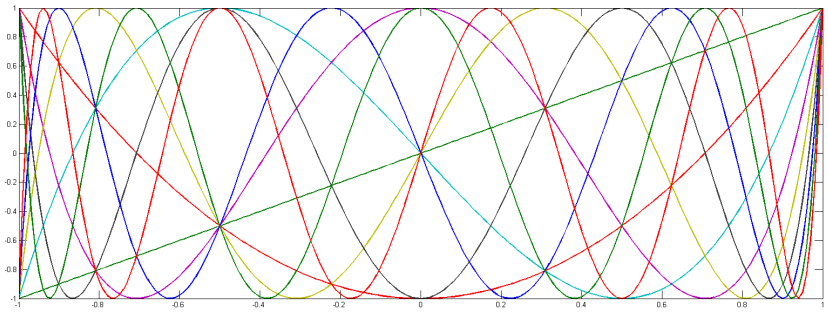
$$\phi_0(x) = \lambda_0 \; ; \; \phi_1(x) = \lambda_1(x - \alpha_1)\phi_0(x)) \tag{1a}$$

$$\phi_k(x) = \lambda_k(x - \alpha_k)\phi_{k-1}(x)) - \beta_k\phi_{k-2}(x) \tag{1b}$$

Evauleren van drietermsrecursiebetrekking

```
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
```

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.



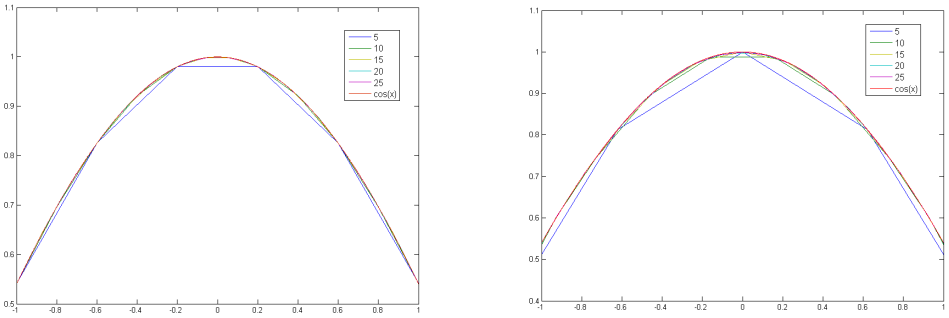
Figuur 1: Evaluatie van eerste tien Cheybchev veeltermen op 200 punten in [-1,1]

Berekenen van interpolerende veelterm

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

b=[];
n=length(alpha);
```

```
%maak M matrix
M=eval_recursion(x,length(alpha),alpha,beta,lambda);

%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
```

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.

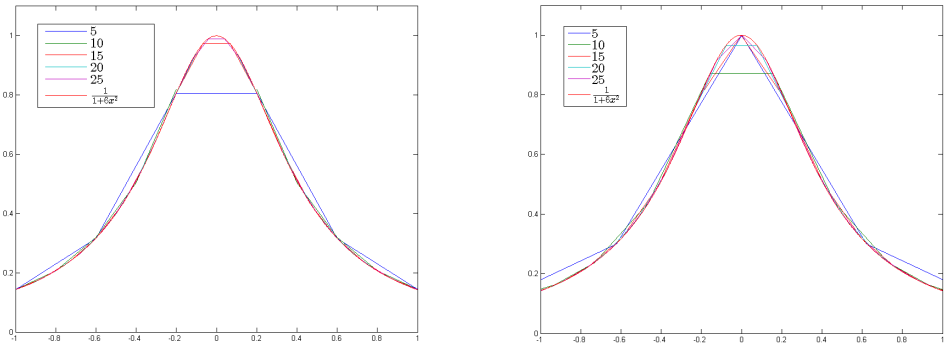


(a) Equidistant

(b) Nulpunten van T_n

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

Bij een verdere analyse van de interpolatiefout blijkt echter dat de nulpunten van de Chebychev veeltermen een veel beter gespreide fout geven. Voor illustratie, zie figuren 4 en 6. Bij de equidistante punten is er aan de rand

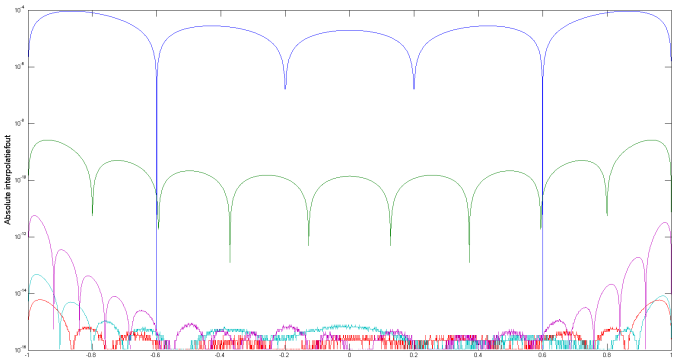


(a) Equidistant (b) Nulpunten van T_n

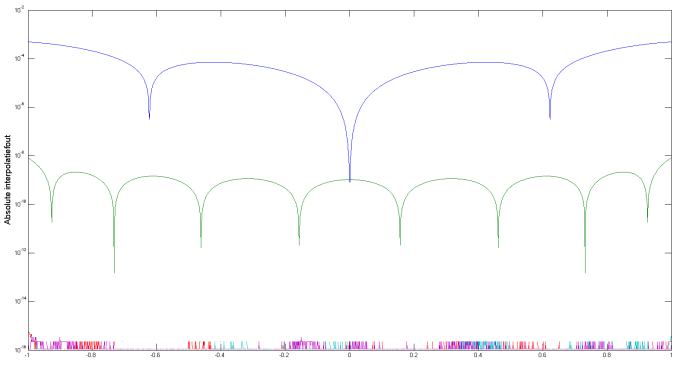
Figuur 3: Enkele interpolerende veeltermen van $\frac{1}{1+6x^2}$

van het interval een veel grotere fout aanwezig. Bij benaderingen van lagere graad is de absolute fout wel kleiner bij de equidistante punten.

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 bepaald punt. De Chebychev interpolatie daarentegen daalt in het begin ongeveer even snel, maar blijft bij hogere graad constant rond ϵ_{mach}. Dit wordt duidelijk op figuur **??**

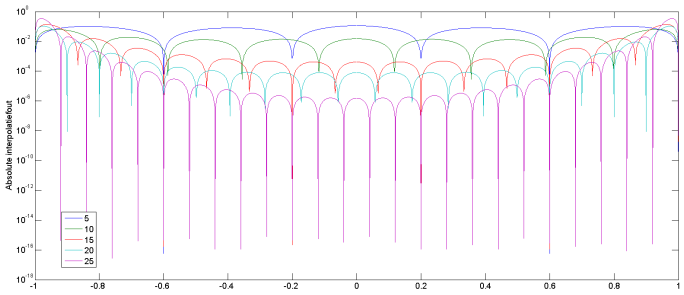


(a) Equidistant

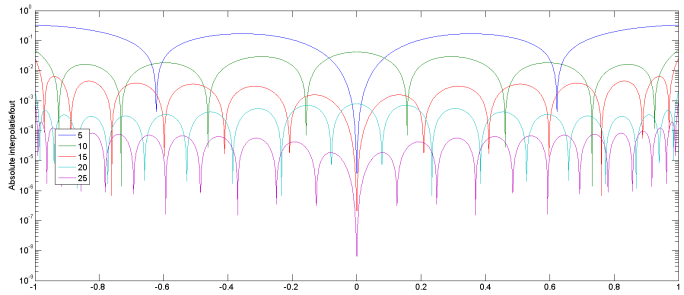


(b) Nulpunten van T_n

Figuur 4: Absolute fout bij het interpoleren van $\cos(x)$

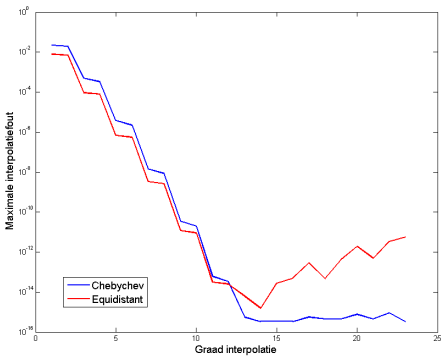


(a) Equidistant

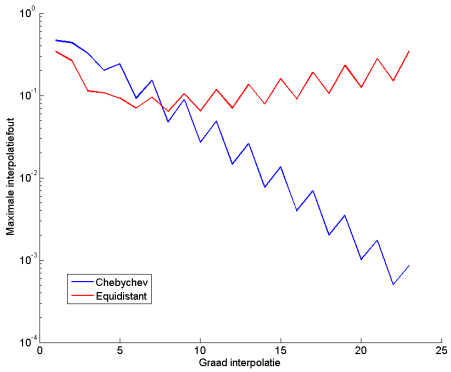


(b) Nulpunten van T_n

Figuur 5: Absolute fout bij het interpoleren van $\frac{1}{1+6x^2}$



(a) $\cos x$



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

Figuur 6: Maximale interpolatiefout