Numerieke Modellering en Benadering

De Wolf Peter
Vekemans Wout

10 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

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

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

%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);
size(nuMat)
size(muMat)

%maak matrix A
A= alphaMat+nuMat+muMat;

%bereken de eigenwaarden van A en dus de nulpunten van de veelterm
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.

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

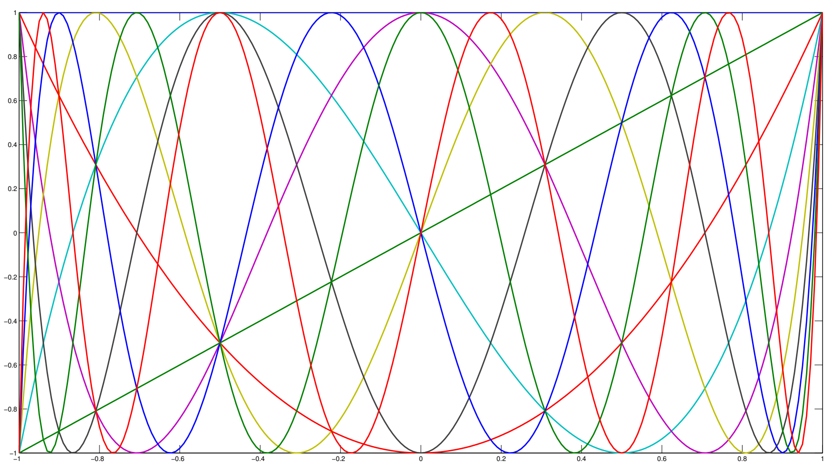
$$\phi_k(x) = \lambda_k(x - \alpha_k)\phi_{k-1}(x)) - \beta_k\phi_{k-2}(x) \tag{1b}$$

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)

%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
values=eval_recursion(t,length(alpha),alpha,beta,lambda);
res=values*c;
```

Als we met deze functie een interpolerende veelterm voor de functie $\cos(x)$ berekenen krijgen we figuur **??**. Op de grafiek zijn de interpolaties van graad 3, 5 en 10 weergegeven. De interpolatiefout voor die drie interpolerende functies zijn weergegeven op figuur **??**. We zien duidelijk dat een benadering van hogere graad nauwkeuriger is. Dit wordt ook geïllustreerd door figuur **??**. Hierop is de maximale fout in functie van de graad van de interpolerende veelterm weergegeven.