Orthangonale Functies - Oefening 6

Ruben Van Assche

December 3, 2015

1 Chebychev Veeltermen

In het eerste deel van de oefening wordt gevraagd om de Chebychev veeltermen $T_4(x)$ en $T_7(x)$ te berekenen. Dit kan via de recursieve formule:

$T_0(x) = 1$
$T_1(x) = x$
$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x)$

Hierdoor bekomen we na wat rekenwerk:

$T_0(x) = 1$
$T_1(x) = x$
$T_2(x) = 2x^2 - 1$
$T_3(x) = 4x^3 - 3x$
$T_4(x) = 8x^4 - 8x^2 + 1$
$T_5(x) = 16x^5 - 20x^3 + 5x$
$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$
$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x$

2 Integralen verifieren

Het tweede deel van de oefening vraagt om enkele integralen met chebychev veeltermen uit te rekenen en te controleren of deze uitkomsten overeenkomen met de gegeven uitkomsten.

De integralen zijn van de vorm:

$\int_{-1}^{1} \frac{T_i(X)T_j(X)}{\sqrt{1-x^2}} dx$

Nu is een mogelijke oplossing simpel: we vullen de chebychev veeltermen in de intergraal in. Vervolgens bekomen we een integraal die we niet zo snel kunnen integreren waardoor we dus numerieke integratie zullen moeten uitvoeren om de uitkomst van de integraal op het interval [-1, 1] te weten. Dit zal veel rekenwerk vereisen en dat willen we beperken

Nu is het ook mogelijk om de chebychev veeltermen te herschrijven, een compactere vorm van de chebychev veelterm is:

$T_i(x) = cos(i * arccos(x))$

Wanneer we deze compactere vorm gebruiken is het veel makkelijker deze integralen op te lossen.

1. Integraal $\int_{-1}^{1} \frac{T_0(X)T_1(X)}{\sqrt{1-x^2}} dx = 0$

Vermits we hier nog met een lage graad van de chebychev veelterm werken, zullen we het recursief voorschrift gebruiken inplaats van de compactere vorm, wanneer we dan de veeltermen invullen verkijgen we:

$\int_{-1}^{1} \frac{x}{\sqrt{1-x^2}} dx$

Wanneer we hierbij een subsititutie doen met $u = 1 - x^2$ en dus $du = -2x$

$-\frac{1}{2} \int_{-1}^{1} \frac{1}{\sqrt{u}} du$

$= -\frac{1}{2}[2\sqrt{u}]_{-1}^{1}$

$= [-\sqrt{1-x^2}]_{-1}^{1}$

Dit leidt dan tot:

$-\sqrt{1-1^2} + \sqrt{1-(-1)^2} = 0$

2. Integraal $\int_{-1}^{1} \frac{T_4(X)T_7(X)}{\sqrt{1-x^2}} dx = 0$

Hierbij zullen we wel de compactere vorm van het chebychev polynoom gebruiken:

$\int_{-1}^{1} \frac{cos(4cos^{-1}(x)) * cos(7cos^{-1}(x))}{\sqrt{1-x^2}} dx = 0$

Wanneer we hierbij een substitutie doen $u = cos^{-1}(x)$ en dus $du = \frac{-1}{\sqrt{1-x^2}}$

$= -\int_{-1}^{1} cos(4u) * cos(7u) du$

D.m.v. de trigoniometrie weten we
$cos(\alpha) * cos(\beta) = \frac{1}{2}(cos(\alpha - \beta) + cos(\alpha + \beta))$

$= -\frac{1}{2} \int_{-1}^{1} cos(3u) + cos(11u) du$

$= -\frac{1}{2}\int_{-1}^{1} cos(3u)du - \frac{1}{2}\int_{-1}^{1} +cos(11u)du$

$= [-\frac{1}{6}sin(3u) - \frac{1}{22}sin(11u)]_{-1}^{1}$

$= [-\frac{1}{6}sin(3cos^{-1}(x)) - \frac{1}{22}sin(11cos^{-1}(x))]_{-1}^{1}$

$= -\frac{1}{6}sin(3cos^{-1}(1)) - \frac{1}{22}sin(11cos^{-1}(1)) + \frac{1}{6}sin(3cos^{-1}(-1)) + \frac{1}{22}sin(11cos^{-1}(-1))$

$= 0$

3. Integraal $\int_{-1}^{1} \frac{T_4(X)^2}{\sqrt{1-x^2}}dx = \frac{\pi}{2}$

Opnieuw maken we gebruik van de compactere chebychev vorm.

$\int_{-1}^{1} \frac{cos(4cos^{-1}(x))^2}{\sqrt{1-x^2}}dx$

Wanneer we hierbij een substitutie doen $u = 4cos^{-1}(x)$ en dus $du = \frac{-4}{\sqrt{1-x^2}}$

$= -\frac{1}{4}\int_{-1}^{1} cos(u)^2 du$

Uit de trigoniometrie weten we dat $cos(x)^2 = \frac{1}{2}(cos(2x) + 1)$

$= -\frac{1}{4}\int_{-1}^{1} \frac{1}{2}(cos(2u) + 1)du$

$= -\frac{1}{8}\int_{-1}^{1} cos(2u)du - \frac{1}{8}\int_{-1}^{1} du$

Wanneer we hierbij de substitutie $w = 2u$ en dus $dw = 2$

$= -\frac{1}{16}\int_{-1}^{1} cos(w)dw - \frac{1}{8}\int_{-1}^{1} du$

$= [-\frac{1}{16}sin(2u) - \frac{1}{8} * 4cos^{-1}(x)]_{-1}^{1}$

$= [-\frac{1}{16}sin(8cos^{-1}(x)) - \frac{1}{8} * 4cos^{-1}(x)]_{-1}^{1}$

$= -\frac{1}{16}sin(8cos^{-1}(1)) - \frac{1}{8} * 4cos^{-1}(1) + \frac{1}{16}sin(8cos^{-1}(-1)) + \frac{1}{8} * 4cos^{-1}(-1)$

$= -\frac{1}{16}sin(0) - \frac{1}{8} * 0 + \frac{1}{16}sin(8 * \pi) + \frac{1}{8} * 4 * \pi$

$= \frac{1}{16} * 0 + \frac{1}{8} * 4 * \pi$

$= \frac{\pi}{2}$

4. Integraal $\int_{-1}^{1} \frac{T_0(X)^2}{\sqrt{1-x^2}}dx = \pi$

Bij deze integraal laten we de compacte chebychev vorm achterwege en gebruiken we opnieuw het recursief voorschrift.

$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} dx$

$= [sin^{-1}(x)]_{-1}^{1}$

$= sin^{-1}(1) - sin^{-1}(-1)$

$= \frac{\pi}{2} - \frac{-\pi}{2}$

$= \pi$

3 Grafiek van $T_7(x)$

Maak een grafiek van $T_7(x)$ en controleer of de nulpunten en extrema beschreven door de formules in de cursus kloppen.

De formule voor nulpunten wordt beschreven door:

$T_i(cos\frac{(2j-1)\pi}{2i}) = 0$ voor j = 1, ... , i

De formule voor extrema wordt beschreven door:

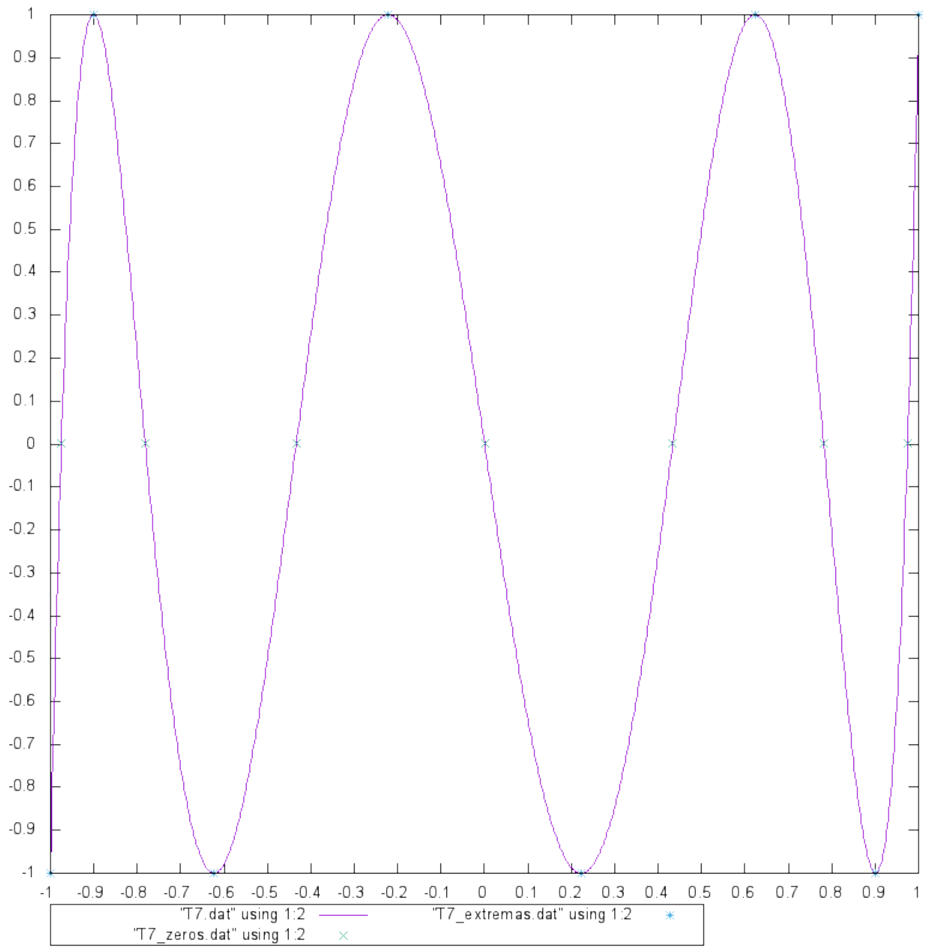
$T_i(cos(\frac{j\pi}{i}) = (-1)^j$ voor j = 0, ..., i

Om een grafiek te tekenen in GNUPlot hebben we een set van punten nodig welke we dan plotten. De punten kunnen we berekenen d.m.v. de compacte chebychev vorm $T_i(x) = cos(i * arccos(x))$. We zullen 1000 punten bereken en deze vervolgens opslaan in een bestand wat we aan GNUPlot geven.

Daarnaast zullen we ook nog de 7 x-waarden van nulpunten berekenen d.m.v. de formule in de cursus en de 8 x-waarden van extrema berkend d.m.v. de formule in de cursus. Deze 15 punten zullen we ook doorgeven aan GNUPlot waardoor we een grafiek bekomen met zowel een plot van: de functie, de nulpunten en de extrema.

De 1000 punten die we gaan plotten zullen in een interval $[a, b]$ liggen welk moet bepaald worden zodat de meest interessante grafiek wordt geplot. We zullen hiervoor een vector v construeren met de berekende x-waarden van nulpunten en extrema. Uit v halen we dan $a = min(v)$ en $b = max(v)$. Vervolgens zullen we op dit interval 1000 punten berekenen met $x = min(v) + i * (\frac{max(v)-min(v)}{1000})$ waarbij i = 0,...,1000.

Resultaat



Na het bereken blijkt dat we best onze grafiek weergeven op het interval $[-1, 1]$. Hieronder de uit formules berekende x-waarden van nulpunten en extrema:

j	Nulpunt	Extrema(x)	Extrema(y)
0	/	1	1
1	0.974928	0.900969	-1
2	0.781831	0.62349	1
3	0.433884	0.222521	-1
4	6.12323e-17	-0.222521	1
5	-0.433884	-0.62349	-1
6	-0.781831	-0.900969	1
7	-0.974928	-1	-1

Wanneer we visueel de grafiek bestuderen lijken de formules uit de cursus correct! De berekende nulpunten liggen op de coordinaten waar de nulpunten van de grafiek liggen. Bij de berekende extrema is dit hetzelfde.

4 Maak $T_7(x)$ monisch

Een monische chebychev vergelijking wordt berekend door: $\frac{T_n(x)}{2^{n-1}}$ in het geval van $T_7(x)$ zal dit dus de veelterm $\frac{T_7(x)}{2^6}$ zijn.

We willen weten wat de bovengrens is van de absolute waarde van deze functie.We hebben een formule in de cursus voor het berekenen van extrema van een chebychev veelterm, deze is makkelijk aan te passen naar een versie die kan werken met de monische chebychev veelterm.

$T_i(cos(\frac{j\pi}{i}) = \frac{(-1)^j}{2^{i-1}}$ voor j = 0, ..., i

En dus voor $T_7(x)$

$T_7(cos(\frac{j\pi}{7}) = \frac{(-1)^j}{2^6}$ voor j = 0, ..., i

Wanneer we nu het programma geschreven voor het derde deel van deze oefening aanpassen met deze nieuwe veelterm en de absolute waarde functie krijgen we volgende extrema:

j	Extrema(X)	Extrema(Y)
0	1	0.015625
1	0.900969	0.015625
2	0.62349	0.015625
3	0.222521	0.015625
4	-0.222521	0.015625
5	-0.62349	0.015625
6	-0.900969	0.015625
7	-1	0.015625

Dit is niet verwonderlijk, de extrema zijn door de absolute waarde functie allemaal positief geworden. Door het monisch maken van de veelterm wordt elke y waarde van de functie gedeeld door 64 en dus ook de extrema. Waardoor we kunnen stellen dat de functie een bovengrens 0.015625 heeft.

5 Code

T7.cpp voor berekenen van deel 3

```
#include <stdio.h>
```

```
#include <math.h>
#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include <utility>

class FileWriter{
public:
  std::string filename;
  std::vector< std::pair<double, double>> buffer;

  FileWriter(std::string name){
    this->filename = name;
    // Clean file
    std::ofstream file;
    file.open(this->filename, std::ofstream::out | std::ofstream::trunc);
    file.close();
  }

  void write(double x, double y){
    std::pair<double, double> coordinates = std::make_pair(x,y);
    buffer.push_back(coordinates);
  }

  ~FileWriter(){
    std::ofstream file;
    file.open(this->filename);
    file << "#    X    Y\n";

    for(int i = 0; i < buffer.size();i++){
      file << buffer[i].first << "    " << buffer[i].second << "\n";
    }

    file.close();
  }
};

// Caclulates the y value of the chebychev polynom
double calculatePoint(double point){
  /*
  double first = point;
  double third = first*point*point;
  double fifth = third*point*point;
  double seventh = fifth*point*point;
```

```
  return 64*seventh - 112*fifth + 56*third - 7*first;
  */

 return cos(7.0*acos(point));
}

// Caclulate the zeros of T7, index should be in range 1...7
double calculateZero(int index){
  return cos( ((2.0*index - 1.0)*M_PI)/14.0 );
}

// Calculate the extrema of T7, index should be in range 0...7
double calculateExtrema(int index){
  return cos( (index*1.0*M_PI)/7 );
}


int main (void){
  // calculate zeros and write them to a file, also store them in a vector

  FileWriter writerZeros("T7_zeros.dat");
  std::vector<double> zeros;

  std::cout << "ZEROS" << std::endl;
  std::cout << "-----" << std::endl;
  // Calculate zeros
  for(int i = 1; i <= 7;i++){
    double zero = calculateZero(i);
    zeros.push_back(zero);
    writerZeros.write(zero, 0);
    std::cout << "T7(" << zero << ") = " << calculatePoint(zero) << std::endl;
  }

  // calculate extremas and write them to a file, also store them in a vector
  FileWriter writerExtremas("T7_extremas.dat");
  std::vector<double> extremas;

  std::cout << std::endl;
  std::cout << "EXTREMA" << std::endl;
  std::cout << "-------" << std::endl;
  // Calculate zeros
  for(int i = 0; i <= 7;i++){
    double extrema = calculateExtrema(i);
    extremas.push_back(extrema);
    writerExtremas.write(extrema, pow(-1, i));
```

```
    std::cout << "T7(" << extrema << ") = " << calculatePoint(extrema) << "
(should be : " <<  pow(-1, i)  << ")"<< std::endl;;
  }

  std::cout << std::endl;
  std::cout << "CONCLUSION" << std::endl;
  std::cout << "----------" << std::endl;

  // Concatenate zeros and extremas
  std::vector<double> points;
  points.insert( points.end(), zeros.begin(), zeros.end() );
  points.insert( points.end(), extremas.begin(), extremas.end() );

  // Find minimum and maximum of zeros and extremas combined
  auto minElement = min_element(std::begin(points), std::end(points));
  auto maxElement = max_element(std::begin(points), std::end(points));
  double min = *minElement;
  double max = *maxElement;

  FileWriter writerPoints("T7.dat");
  double range = max - min;

  int n = 1000;
  double step = range/1000;

  double x = min;
  for(int i = 0; i <= n; i++){
    double y = calculatePoint(x);
    writerPoints.write(x, y);

    x += step;
  }


  std::cout << "Plot from " << min << "  till " << max << std::endl;
  std::cout << n << " points written to T7.dat" << std::endl;
  std::cout << "To generate graph, run: gnuplot T7.gnuplot" << std::endl;


  return 0;
}
```

Monisch.cpp voor berekenen van deel 4

```
#include <stdio.h>
#include <math.h>
#include <iostream>
```

```
#include <string>
#include <vector>
#include <fstream>
#include <utility>


// Caclulates the y value of the chebychev polynom
double calculatePoint(double point){
  /*
  double first = point;
  double third = first*point*point;
  double fifth = third*point*point;
  double seventh = fifth*point*point;

  return 64*seventh - 112*fifth + 56*third - 7*first;
  */

 return fabs(cos(7.0*acos(point))/64);
}

// Calculate the extrema of T7, index should be in range 0...7
double calculateExtrema(int index){
  return cos( (index*1.0*M_PI)/7 );
}


int main (void){
  // calculate extremas and write them to a file , also store them in a vector
  std::vector<double> extremas;

  std::cout << std::endl;
  std::cout << "EXTREMA" << std::endl;
  std::cout << "-------" << std::endl;
  // Calculate zeros
  for(int i = 0; i <= 7;i++){
    double extrema = calculateExtrema(i);
    extremas.push_back(extrema);
    std::cout << "T7(" << extrema << ") = " << calculatePoint(extrema) << "
(should be : " <<  1.0/64.0  << ")"<< std::endl;;
  }


  return 0;
}
```