Numerieke Modellering en Benadering
Taak 2: Chebyshev Benadering

Thibault Lahaye - r0713047, Ward Kerkhofs - r0750166

2021-2022

1 Continue kleinste kwadratenbenadering met Chebyshev veeltermen

1. Toon aan dat de som van de coëfficiënten van een Chebyshev veelterm $T_k(x)$ gelijk is aan 1, $\forall k \in N$. We prove this via induction. From looking at the definition of the Chebyshev Polynomials we can conclude this proof is trivial for T_0 and T_1. We start the proof for T_{k+1} by assuming that T_k and T_{k-1} both satisfy the theorem.

$$T_{k+1}(x) = 2x * T_k(x) - T_{k-1}(x)$$

We calculate the the sum of the coefficients by evaluating the function in 1.

$$T_{k+1}(1) = 2 * 1 * T_k(1) - T_{k-1}(1)$$

Now from the induction we can fill in 1 for $T_k(1)$ and $T_{k-1}(1)$, which simplifies the equation to $2 - 1 = 1$. And thus the theorem is proven.

2. Ga na dat de veeltermen $T_k(x)$ orthogonaal zijn t.o.v. het volgende scalair product gedefiniëerd op de vectorruimte $C([1,1])$ van continue, reële functies op $[1,1]$. The Chebyshev polynomials are orthogonal on $[-1,1]$ with respect to the weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$

$$\int_{-1}^{+1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \int_{-1}^{+1} \frac{\cos(n \arccos(x)) \cos(m \arccos(x))}{\sqrt{1-x^2}} dx$$

$$\textbf{Substitute: } \theta = \arccos(x) \implies d\theta = \frac{-1}{\sqrt{1-x^2}} dx \implies dx = -\sqrt{1-x^2} d\theta$$

$$\int_{-1}^{+1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = -\int_{\pi}^{0} \cos(n\theta)\cos(m\theta) d\theta = \int_{0}^{\pi} \cos(n\theta)\cos(m\theta) d\theta$$

$$\textbf{For } n \neq m : \cos(n\theta)\cos(m\theta) = \frac{1}{2}\left[\cos(n+m)\theta + \cos(n-m)\theta\right]$$

$$\int_{-1}^{+1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \frac{1}{2}\int_{0}^{\pi} \cos((n+m)\theta) d\theta + \frac{1}{2}\int_{0}^{\pi} \cos((n-m)\theta) d\theta$$

$$= \left[\frac{1}{2(n+m))} \sin((n+m)\theta) + \frac{1}{2(n-m)} \sin((n-m\theta))\right]_0^{\pi} = 0$$

$$\textbf{For } n = m : \int_{-1}^{+1} \frac{T_n(x)^2}{\sqrt{1-x^2}} dx = \int_{0}^{\pi} \cos(n\theta)^2 d\theta = \begin{cases} \pi \text{ } when \ n = m = 0 \\ \frac{\pi}{2} \text{ } when \ n = m \neq 0 \end{cases}$$

$$\textbf{Conclusion: } \int_{-1}^{+1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} \pi \text{ } when \ n = m = 0 \\ \frac{\pi}{2} \text{ } when \ n = m \neq 0 \\ 0 \text{ } when \ n \neq 0 \end{cases}$$

3. Schrijf een Matlab functie function $v = evalCheb(a, x)$ die als invoer een vector $a = (a_0, a_1, ..., a_n) \in R^{n+1}$ en een vector $x = (x_1, ..., x_N) \in [1, 1]^N$ neemt en als uitvoer een vector $v = (f_1, ..., f_N) \in R^N$ teruggeeft, maar zonder expliciete functie-evaluatie van de basisfuncties T_k. Gebruik hiervoor het recursieve rekenschema van Smith. Alpha is always 0, so we ignore it for this function. Beta is always 1, so this can be ignored too. Lambda is always 2, except lambda(0) = 1 and lambda(1) = 1.

```
function [v] = evalCheb(a,x)
    n = size(a,2)-1;

    lambda = zeros(1,n+2);
    lambda(:, 1:2) = 1;
    lambda(1,3:end) = 2;

    v = zeros(1,size(x,2));
    for i = 1 : size(x,2)
        b = zeros(1, n+3);
        for k = n : -1 : 0
            b(k+1) = a(k+1) + lambda(k+2)*x(i)*b(k+2) - b(k+3);
        end
        v(i) = b(1);
    end
end
```

4. We definiëren V_n als de deelvectorruimte van $C([1,1])$ opgespannen door $T_0(x),...,T_n(x)$. Toon aan dat de beste benadering voor een functie $f(x) \in C([1,1])$ in V_n gegeven wordt door volgende formule. The best approximation of $f(x)$ in $P_n[-1,1]$ is given by $\|f-P\| = \inf \|f-P\|_2$

$$\begin{aligned} p(x) &= a_1\phi_1(x) + a_2\phi_2(x) + ... + a_n\phi_n(x) \\ &= \frac{\langle f(x), \phi_1(x)\rangle}{\langle \phi_1(x), \phi_1(x)\rangle}\phi_1(x) + \frac{\langle f(x), \phi_2(x)\rangle}{\langle \phi_2(x), \phi_2(x)\rangle}\phi_2(x) + ... + \frac{\langle f(x), \phi_n(x)\rangle}{\langle \phi_n(x), \phi_n(x)\rangle}\phi_n(x) \\ &= \frac{\langle f(x), T_1(x)\rangle}{\langle T_1(x), T_1(x)\rangle}T_1(x) + \frac{\langle f(x), T_2(x)\rangle}{\langle T_2(x), T_2(x)\rangle}T_2(x) + ... + \frac{\langle f(x), T_n(x)\rangle}{\langle T_n(x), T_n(x)\rangle}T_n(x) \end{aligned}$$

$$\langle f(x), T_k\rangle = \int_{-1}^{+1} \frac{f(x)\cos(k\arccos(x))}{\sqrt{1-x^2}}dx$$

$$\textbf{Substitute: } \theta = \arccos(x) \implies d\theta = \frac{-1}{\sqrt{1-x^2}}dx \implies dx = -\sqrt{1-x^2}d\theta$$

$$\langle f(x), T_k\rangle = \int_0^\pi \frac{f(\cos(\theta))\cos(k\theta)\sqrt{1-x^2}}{\sqrt{1-x^2}}d\theta = \int_0^\pi f(\cos(\theta))\cos(k\theta)d\theta$$

$$\langle T_k, T_k\rangle = \begin{cases} \pi \ \textit{when } n = m = 0 \\ \frac{\pi}{2} \ \textit{when } n = m \neq 0 \end{cases}$$

$$p(x) = \frac{1}{\pi}\int_0^\pi f(\cos(\theta))d\theta + \frac{2}{\pi}\int_0^\pi f(\cos(\theta)\cos(\theta))d\theta + \\ \frac{2}{\pi}\int_0^\pi f(\cos(\theta)\cos(2\theta))d\theta + ... + \frac{2}{\pi}\int_0^\pi f(\cos(\theta)\cos(n\theta))d\theta$$

5. Voor voorgaande integralen bestaat niet altijd een gesloten formule. Deze integralen kunnen echter efficiënt benaderd worden door het interval $[0,\pi]$ op te delen in n deelintervallen en de trapeziumregel toe te passen op elk deelinterval. Dit is de samengestelde trapeziumregel voor numerieke integratie. Leid op deze manier een benadering af voor de integralen. The integral can be approximated by calculating the sum of all tiny trapezoids. The area A of each trapezoid can be determined by using following formula, where $g(x)$ equals the function that is being integrated.

$$A = \frac{g(m) + g(m+1)}{2} * \frac{\pi}{n}$$

We can use this to approximate a_k like so:

$$a_k \approx \frac{\delta(k)}{\pi} * \sum_{i=0}^{n-1} \frac{[f(cos(\frac{\pi}{n}i)) * cos(\frac{k\pi}{n}i)] + [f(cos(\frac{\pi}{n}(i+1))) * cos(\frac{k\pi}{n}(i+1))]}{2} * \frac{\pi}{n}$$

Where $\delta(k) = 1$ for $k = 0$ and 2 for $k > 0$. We can rewrite this as:

$$a_k \approx \frac{\delta(k)}{n} * (\frac{f(1)}{2} + \sum_{i=1}^{n-1} f(cos(\frac{\pi}{n} * i)) * cos(\frac{k\pi}{n} * i) + \frac{f(-1) * cos(k\pi)}{2})$$

6. In de formules voor de coëfficiënten a_k, zoals afgeleid in de vorige deelvraag, herken je de inverse discrete cosinus transformatie (IDCT) van een reële rij $f(z_0), f(z_1), ..., f(z_n)$. Waarbij je de waarde van $z_l, l = 0, ..., n$, bekomen hebt in de vorige deelvraag. Toon aan dat z_l de punten zijn waarin $T_n(x)$ zijn maximale en minimale waarde bereikt op het interval $[1, 1]$. We find z_l in the previous exercise. We use the following definition of the Chebyshev Polynomials to start this proof: $T_n(x) = cos(n * arccos(x))$

$$z_l = cos(\frac{\pi}{n}l)$$

$$T_n(z_l) = cos(n * arccos(cos(\frac{\pi}{n}l))) = cos(n * \frac{\pi}{n}l) = cos(\pi l)$$

From this we can easily see that the maximum values of the cosine (and the polynomial), -1 and 1 are reached when l is odd and even respectively.

7. Schrijf een Matlab functie function $a = approxCheby(f, n)$ met als invoer een function handle f en een getal n, en als uitvoer een vector a met daarin de coëfficiënten van de beste benadering $y_n(x)$ van f in V_n. Gebruik daarvoor de IDCT.

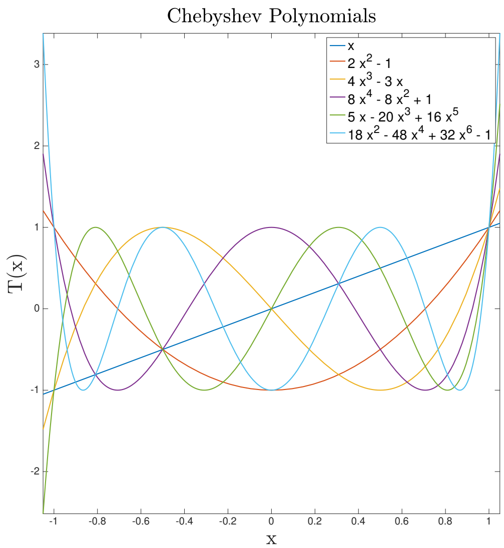
```
function a = approxCheby(f,n)
    % Determining the real series f(z0), f(z1), ... , f(zn)
    z = zeros(1,n+1);
    v = zeros(1,n+1);
    for l = 1:n+1
        z(1,l) = cos(pi*(l-1)/n);
        v(1,l) = f(z(l));
    end

    % Calculating factors ak
    w = fliplr(v(2:end-1));
    v_even = [v w];
    V = fft(v_even)/n;
    V = real(V(1:n+1));
    a = [V(1)/2 V(2:end)];
end
```

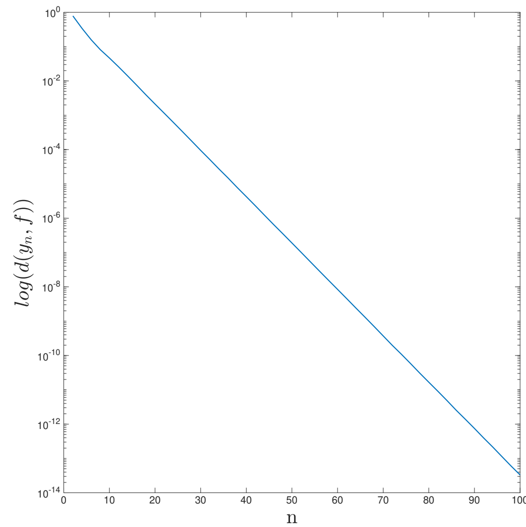
8. In deze opgave gaan we de beste benadering berekenen van de functie $f(x) = \frac{1}{10x^2+1}$ in de deelruimtes $V_2, V_4 ... V_{100}$.

8.1. Geef de basisfuncties $T_k(x), k = 1...10$, weer op dezelfde grafiek. Wat valt je op? The Chebyshev polynomials go to infinity very fast outside of $[-1, 1]$, they also lose their orthogonality. All zero crossings are in this interval too. We can also see that the highest values for the polynomials are -1 and 1. The polynomials of even k are even, and those of odd k are odd.

8.2. Bereken vervolgens de benaderende functies $y_n(x)$ en de bereken de gevraagde foutwaarde. Maak een grafiek van deze foutwaarde in functie van n .



8.1



8.2: Error

2 Interpolatie in Chebyshev knooppunten

1. Leid een gesloten formule af voor de nulpunten van de k-de Chebyshev veelterm $T_k(x)$. We determine a formula for the zeros x_i for $i = 1, ..., k$. We start this derivation from the following definition of the Chebyshev Polynomials: $T_k(x) = cos(k * arccos(x))$

$$T_k(x) = cos(k * arccos(x)) = 0$$

$$\Rightarrow k * arccos(x_i) = \frac{\pi}{2} * (2i - 1) \Rightarrow arccos(x_i) = \frac{\pi}{2} * \frac{2i-1}{k} \Rightarrow x_i = cos(\frac{\pi}{2} * \frac{2i-1}{k})$$

2. Schrijf een Matlab-functie voor het berekenen van de interpolerende veelterm van een functie f in n punten $\{x_k\}_{k=1}^{n}$. De functie $function[c, kappa] = interpolate(x, f)$ krijgt als invoer een vector x met n interpolatiepunten en een function handle f. De functie berekent de vector c met coëfficiënten $c_0, ..., c_{n-1}$ van de interpolerende veelterm $g_n(x)$ en het 2-norm conditiegetal $kappa$ van de matrix M.

```
function [c,kappa] = interpolate(x, f)
    n = size(x, 2);

    M = zeros(n,n);
    for j = 1:n
        for i = 1:n
            M(i,j) = cos((j-1)*acos(x(i)));
        end
    end

    b = zeros(n,1);
    for i = 1:n
        b(i,1) = f(x(i));
    end

    c = linsolve(M,b);
    kappa = cond(M);
end
```

3. Bereken de interpolerende veelterm van de functies $f_1(x) = x^{10} - 1$ en $f_2(x) = \frac{1}{10x^2+1}$ op het interval $[-1, 1]$, voor $n = 2, 4, ..., 100$. Gebruik als interpolatiepunten achtereenvolgens n equidistant verdeelde punten op $[-1, 1]$ en de nulpunten van $T_n(x)$.

3.1 Maak voor beide functies telkens 2 grafieken: een eerste grafiek met alle interpolaties g_n en een tweede grafiek met het verschil $g_n - f$. Om de interpolant te evalueren gebruik je je geïmplenteerde functie *evalCheb*. Wat valt je op? Hoe verloopt de fout in de verschillende gevallen? Even functions are only approximated by even Chebyshev polynomials. The uneven coefficients are zero. When using the equidistant points, there are large errors at the edges of the interpolation interval.

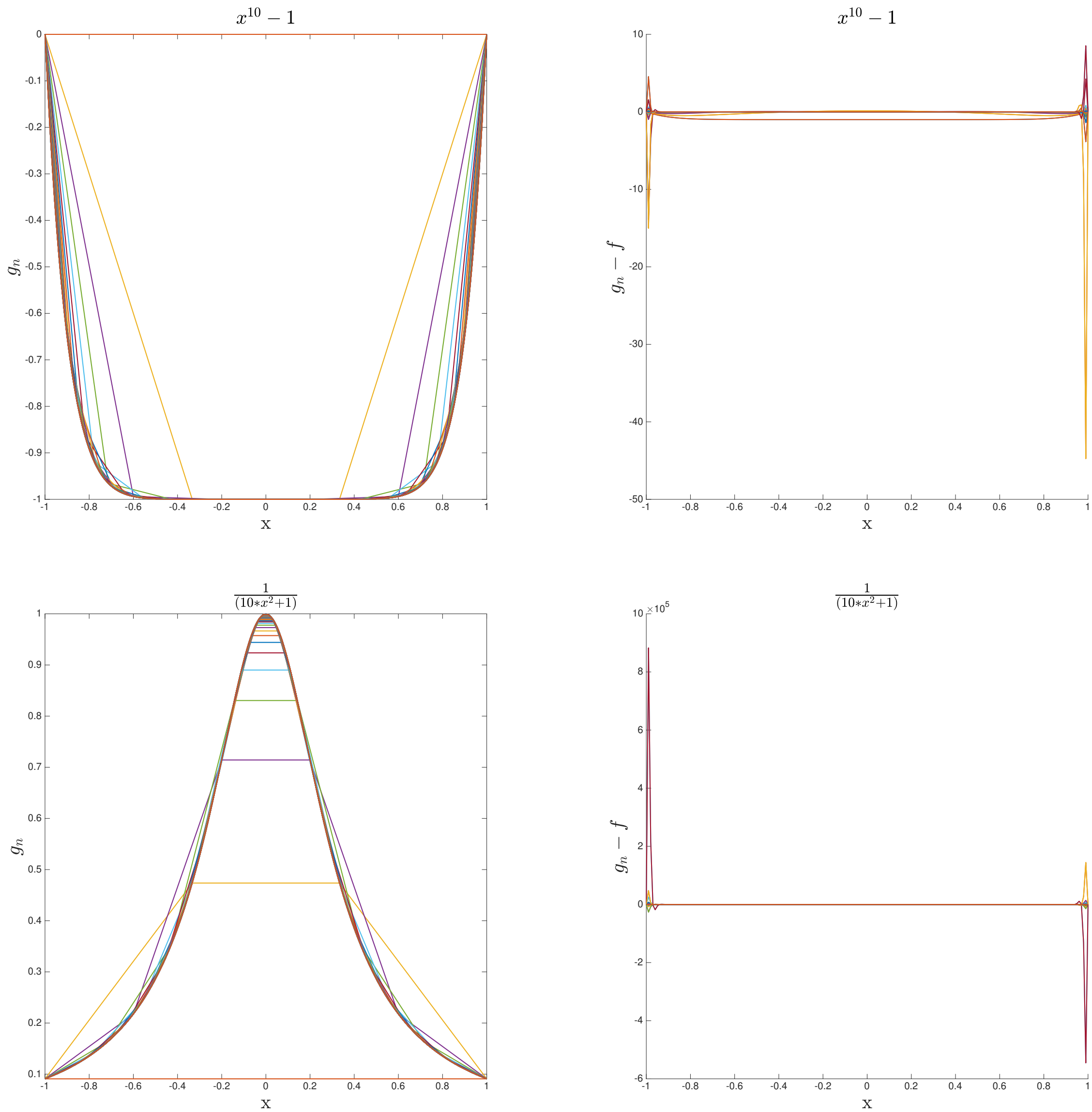


Figure 2: Equidistant

When using the Chebyshev zeros, the error keeps getting smaller and eventually the approximation converges to the original function in the entire interval.

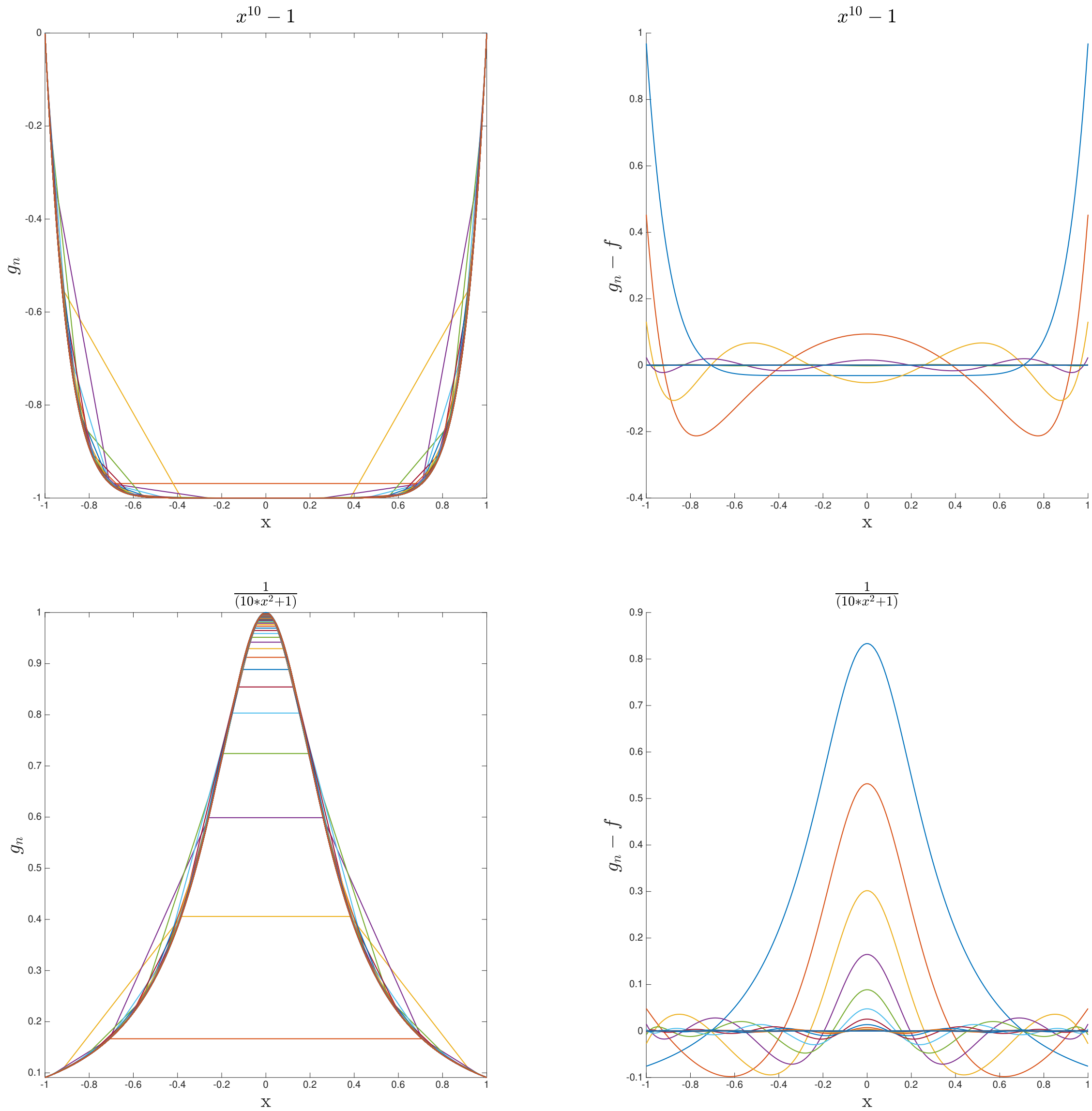


Figure 3: Checbyshev Zeros

3.2 Maak voor elke familie van punten en voor beide functies een grafiek van de foutwaarde uit 8.2 in functie van de graad n. Wat besluit je hieruit? (Welke functie is het moeilijkst te benaderen en waarom? Welke interpolatiepunten zijn het meest geschikt en waarom?) Function two is harder to approximate. Polynomials go to infinity outside of [-1,1]. Since the value stays at zero for x goes to infinity this function is hard to approximate using polynomials. The zeros of the Chebyshev functions are the best interpolation points, since these points lie closer to the edges of the interval [-1,1]. This means they are better suited to stop the going to infinity at the edges. When using equidistant points, the matrices M are very poorly conditioned, resulting in very large errors.

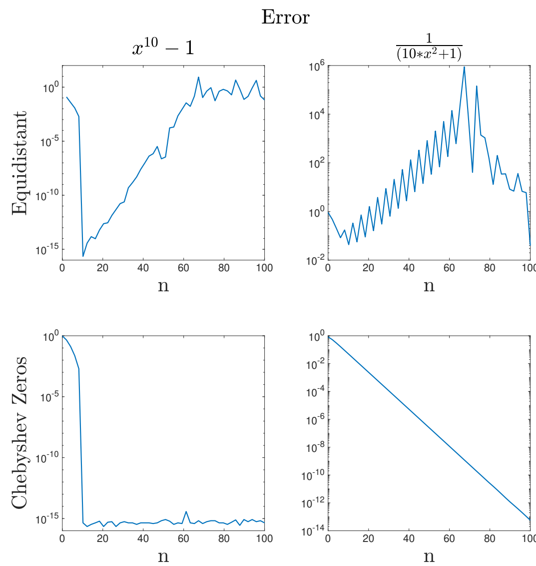


Figure 4: Error in function of n. Column one contains the error on function one, column two contains the error on function two.

3.3 Plot het verloop van het conditiegetal van M in functie van n voor beide puntenfamilies. Waarom is het conditiegetal van de matrix onafhankelijk van de functie f? The condition number of matrix M is not affected by the function since the function is not used in its creation. The condition number depends on the base polynomials and the interpolation points only.

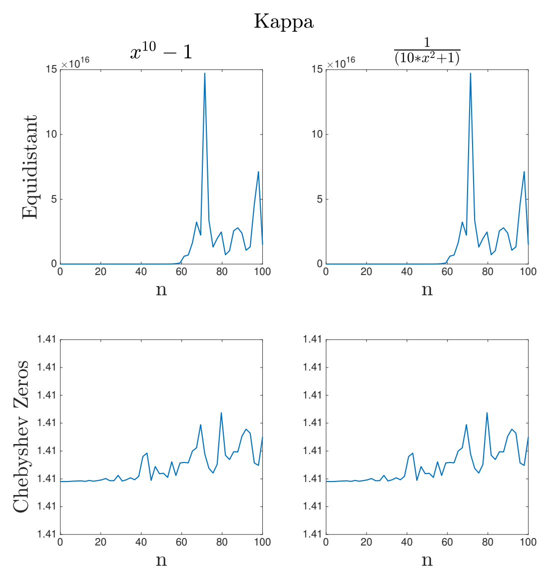


Figure 5: Kappa in function of n. Column one contains the error on function one, column two contains the error on function two.

4. We kunnen nu de resultaten van de continue kleinste kwadratenbenadering en de interpolatie vergelijken voor de functie $f_2(x)$ voor eenzelfde graad van de benaderende veelterm. Plot de maximale fout op eenzelfde figuur voor $n = 2, 4, ..., 100$. Komen de resultaten overeen? Welke methode is het nauwkeurigst voor eenzelfde graad en hoe komt dit? The interpolation with equidistant points is by far the worst method, so it is not discussed further here. The least squares and Chebyshev zeros interpolation converge at the same rate. They are nearly identical, apart from a constant factor. The least squares is most accurate. The least squares and Chebyshev zeros interpolation are functionally identical. But the interpolation has a worse condition.

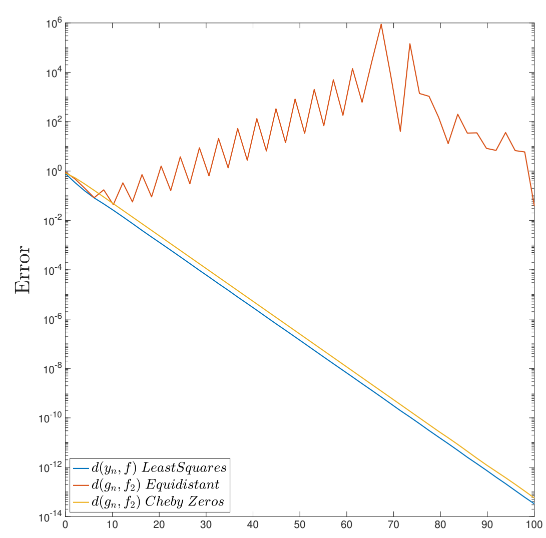


Figure 6: Error in function of n for Interpolation and Least Squares