

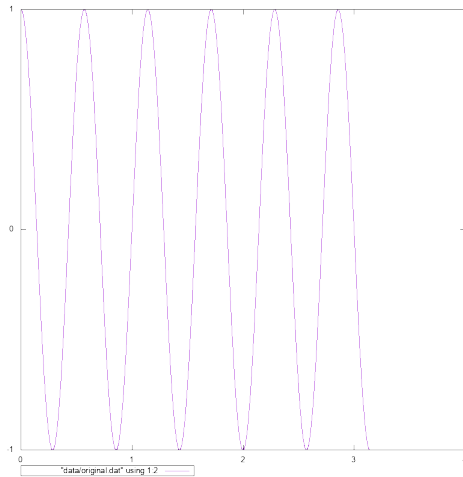
Data Fitting - Oefening 5

Ruben Van Assche

8 december 2017

1 Originele functie

In de opgave moest de functie $\cos(11x)$ op het interval $[0, 2\pi]$. Als eerste kijken we eens naar een plot van deze functie:



1.1 Nulpunten

Vervolgens kijken we eens naar de nulpunten van deze functie. We weten dat $\cos(x)$ als nulpunt $\frac{\pi}{2}$ heeft op het interval $[0, 2\pi]$. $\cos(11x)$ heeft 11 nulpunten op het interval $[0, 2\pi]$.

$$\cos(11x) = 0$$

$$11x = \frac{1}{2}\pi + k\pi$$

$$x = \frac{\pi}{22} + \frac{k\pi}{11}$$

Hierbij is $k = 0, \dots, 10$. Wanneer we dit verder trekken verkrijgen we volgende nulpunten:

$$\begin{array}{ccc} \frac{\pi}{22} & \frac{\pi}{22} + \frac{\pi}{11} & \frac{\pi}{22} + \frac{2\pi}{11} \\ \frac{\pi}{22} + \frac{3\pi}{11} & \frac{\pi}{22} + \frac{4\pi}{11} & \frac{\pi}{22} + \frac{5\pi}{11} \\ \frac{\pi}{22} + \frac{6\pi}{11} & \frac{\pi}{22} + \frac{7\pi}{11} & \frac{\pi}{22} + \frac{8\pi}{11} \\ \frac{\pi}{22} + \frac{9\pi}{11} & \frac{\pi}{22} + \frac{10\pi}{11} & \frac{\pi}{22} + \frac{11\pi}{11} \end{array}$$

Of decimaal:

$$\begin{array}{ccc} 0,1427996661 & 0,4283989982 & 0,7139983304 \\ 0,9995976625 & 1,2851969947 & 1,5707963268 \\ 1,8563956589 & 2,1419949911 & 2,4275943232 \\ 2,7131936554 & 2,9987929875 & \end{array}$$

1.2 Extrema

De extrema van de $\cos(x)$ functie zijn -1 en 1. Een aanpassing hiervan kan enkel wanneer een transformatie van de vorm $r * \cos(x)$ of $\cos(x) + r$ plaatsvindt. Gezien deze niet gebeurt bij de transformatie van $\cos(x)$ naar $\cos(11x)$ zullen de extrema hetzelfde blijven -1 en 1.

2 Trigonometrische Interpolant

Voor het volgende deel van de opgave moest een trigonometrische interpolant bepaald worden. De gegeven punten hiervoor zijn:

$$\begin{array}{ccc} \mathbf{x} & 0 & \pi/2 & \pi \\ \mathbf{y} & 1 & 0 & -1 \end{array}$$

De bedoeling is om een trigonometrische interpolant van de vorm:

$$\frac{a_0}{2} + a_1 \cos(x) + b_1 \sin(x)$$

Hiervoor worden volgende formules gebruikt:

$$a_j = \frac{2}{n} \sum_{k=0}^{n-1} y_k \cos(jx_k)$$

$$b_j = \frac{2}{n} \sum_{k=0}^{n-1} y_k \sin(jx_k)$$

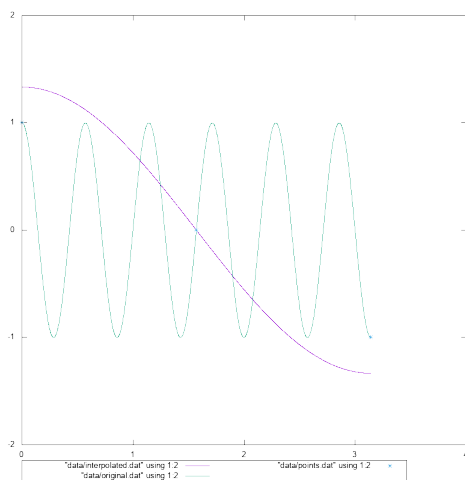
Wanneer we deze formules invullen met de gegevens komen we volgende coëfficiënten uit:

	a	b
0	0	0
1	1.33333333	0

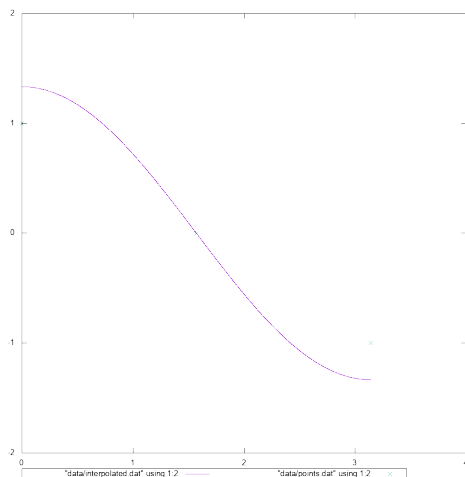
En dus is de trigonometrische interpolant:

$$1.33333\cos(x)$$

Vervolgens plotten we de interpolant met de originele functie:



Dit komt natuurlijk niet in de buurt van een goede interpolant, in feite is het zelfs geen interpolant. Wanneer we de gegeven punten en de interpolant plotten zien we dat deze niet door de originele punten gaat:



3 Discrepantie

Er is duidelijk iets mis met de interpolatie. Deze gaat slechts door één punt $(\frac{\pi}{2}, 0)$ inplaats van alle en volgt totaal niet de vorm van de originele functie.

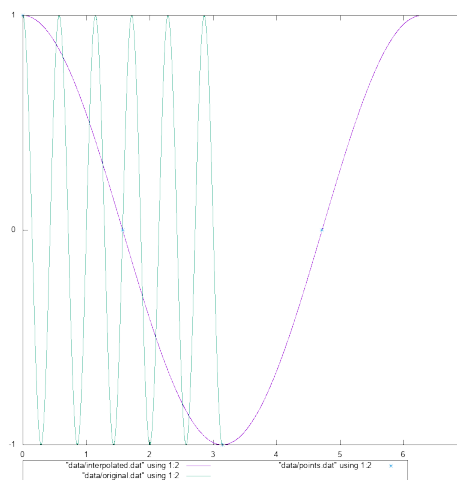
3.1 Oplossing

3.1.1 Algoritme

Een eerste idee kan natuurlijk zijn dat er ergens een foutje is geslopen in het vertalen van de theorie naar de code, nu is dit zeer snel gecheckt. Wanneer de gegevens van de voorbeelden in de theorie worden ingevuld in het algoritme. Dan zien we dat deze dezelfde resultaten als in de theorie opleveren. Dus kunnen we feitelijk stellen dat het algoritme klopt.

3.1.2 Periodiciteit

Wat belangrijk is voor de trigonometrische interpolant is dat de datapunten gelijk verdeeld zijn over het interval 2π anders krijgen we dus een fout zoals deze. Dit zien we door een extra punt $(\frac{3}{2})\pi, 0)$ toe te voegen. Dan verkrijgen we volgende plot:



Dit begint er al beter uit te zien! De interpolant gaat nu door de 4 punten en haar extrema blijven ook binnen het berekende interval $[-1, 1]$ van de originele functie.

Nu kunnen we niet altijd zomaar een extra punt toevoegen. Een verstandiger idee is om de punten te transformeren van het interval $[0, \frac{3}{2}\pi]$

naar $[0, 2\pi]$. Hierbij is de eindwaarde van het originele interval $\frac{3}{2}\pi$ in plaats van π . Dit omdat het laatste punt niet gelijk mag zijn aan het eerste punt. Vervolgens gebruiken we volgende formule:

$$x = \frac{2\pi}{d-c}(s-c)$$

Hierbij schalen we de originele punten s van $[c, d]$ naar $[0, 2\pi]$. In dit geval is dus $c = 0$ en $d = \frac{3}{2}\pi$. Wat volgende transformatie oplevert:

$$x = \frac{2\pi}{0 - \frac{3}{2}\pi}(s-0) = \frac{4}{3}s$$

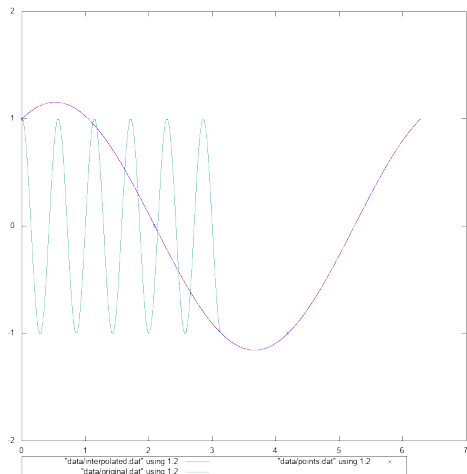
Dit levert dan volgende punten op:

$$\begin{array}{cc} \mathbf{x} & 0 \quad \frac{2}{3}\pi \quad \frac{4}{3}\pi \\ \mathbf{y} & 1 \quad 0 \quad -1 \end{array}$$

Wanneer we vervolgens de interpolant berekenen krijgen we volgende functie:

$$\cos(x) + 0.577350\sin(x)$$

Dit geplot geeft volgend resultaat:



3.1.3 De vorm

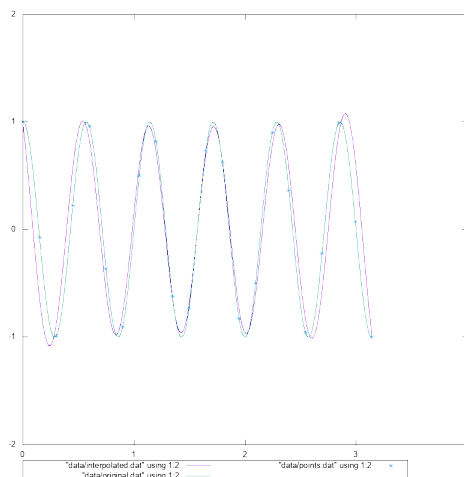
Er is nog steeds iets mis met de interpolatie, de vorm is niet juist! Nu is dit niet geheel onlogisch: gezien er geen transformatie van $\cos(x)$ naar $\cos(rx)$ gebeurt. Dit is dan ook niet mogelijk gezien die r afhangt van de keuze van m (in dit voorbeeld $= 1$).

We gaan het aantal gegeven punten (n) aan de interpolatie verhogen zodat we uiteindelijk de waarde (m) kunnen verhogen. Dit is niet een echte oplossing gezien dat we extra punten toevoegen terwijl we waarschijnlijk enkel de gegeven punten hebben gekregen.

Momenteel is $n = 3$ en $m = 1$. Volgende condities zijn nodig voor een interpolatie: $2m + 1 = n$ en $2m = n$. Uiteindelijk willen we 11 nulpunten dus hebben we een functie van de vorm $\cos(11x)$ nodig. Wat dan weer vereist dat $m = 11$ en dus $n = 22$ of $n = 23$. We kijken eerst naar het geval met $n = 22$ dit levert volgende interpolant op:

$$\begin{aligned} &0.090909\cos(1x) + 0.090909\cos(3x) \\ &+ 0.090909\cos(5x) + 0.090909\cos(7x) \\ &+ 0.090909\cos(9x) + 0.522727\cos(11x) \end{aligned} \quad (1)$$

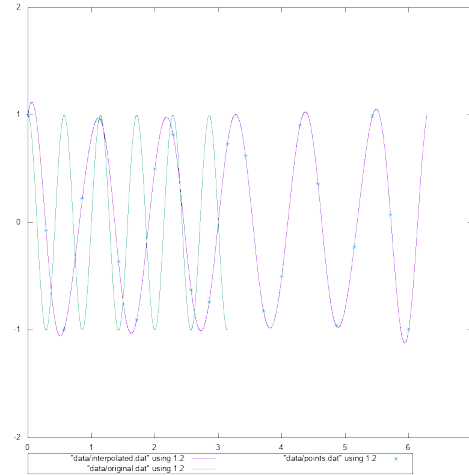
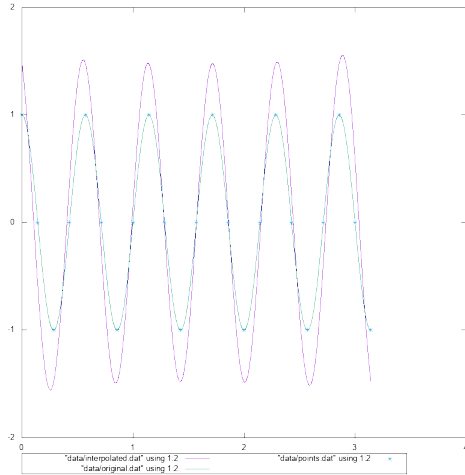
Wanneer we deze interpolant plotten zien we dat deze de vorm volgt van de originele functie!



Maar dit is nog geen goede interpolant want de functie gaat niet door alle originele datapunten en haar beeld ligt niet in $[-1, 1]$. We kijken naar de interpolant met $n = 23$ welke volgende functie oplevert:

$$\begin{aligned} &0.086957\cos(1x) + 0.086957\cos(3x) \\ &+ 0.086957\cos(5x) + 0.086957\cos(7x) \\ &+ 0.086957\cos(9x) + 1.043478\cos(11x) \end{aligned} \quad (2)$$

Wanneer we de plot bekijken zien we dat het interval van het beeld een stuk geworden dan de versie met $n = 22$.



3.1.4 Een combinatie

We hebben voor deze discrepantie 2 oplossingen gezien, uiteindelijk zal de oplossing een combinatie van beide zijn. Bij $n = 22$ zullen we de punten transformeren met $x = \frac{42}{22}s$. Dit ratio $\frac{42}{22}$ halen we uit volgende formule:

$$\frac{(n-1) * 2}{n}$$

Uiteindelijk bekomen we als interpolant:

$$\begin{aligned} &0.710858\cos(6x) + 0.615962\sin(6x) \\ &+ 0.174734\cos(7x) \\ &+ 0.112295\sin(7x) \\ &+ 0.119972\cos(8x) \\ &+ 0.054790\sin(8x) \\ &+ 0.101026\cos(9x) \\ &+ 0.029664\sin(9x) \\ &+ 0.093146\cos(10x) \\ &+ 0.013392\sin(10x) \\ &+ 0.045455\cos(11x) \end{aligned}$$

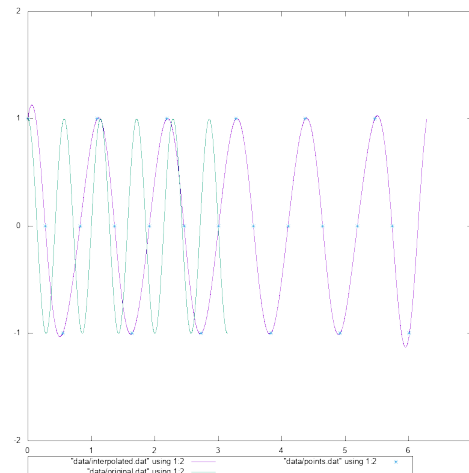
En dit geeft een mooie plot:

Dit process herhalen we ook bij $n = 23$ waarbij de punten getransformeerd worden met $x = \frac{44}{23}s$:

$$\begin{aligned} &0.680593\cos(6x) + 0.635630\sin(6x) \\ &+ 0.173311\cos(7x) \\ &+ 0.122336\sin(7x) \\ &+ 0.118872\cos(8x) \\ &+ 0.061595\sin(8x) \\ &+ 0.099528\cos(9x) \\ &+ 0.035372\sin(9x) \\ &+ 0.090881\cos(10x) \\ &+ 0.018885\sin(10x) \\ &+ 0.087365\cos(11x) \\ &+ 0.005976\sin(11x) \end{aligned} \quad (4)$$

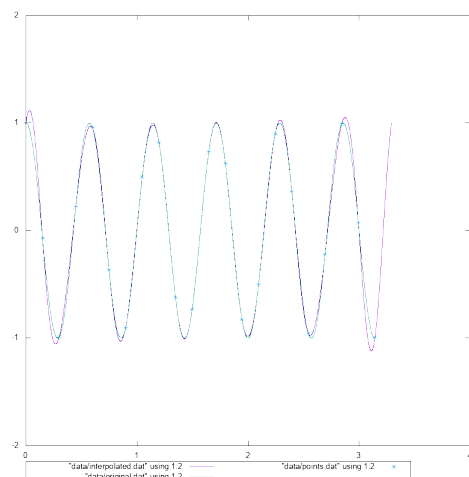
Dit levert dan volgende plot op:

(3)

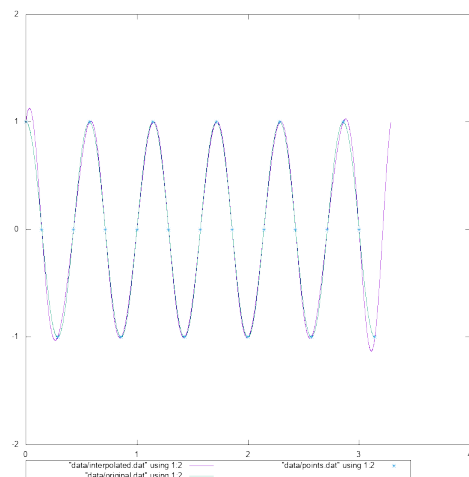


Wat snel opvalt is dat nu het beeld van de interpolant wel veel meer in het interval $[-1, 1]$ ligt t.o.v. $n = 22$.

Als laatste herschalen we $p(x)$, de interpolant terug naar $[0, \pi]$. Dit doen we door $p(x) * (\frac{22}{42})$ uit te voeren bij $n = 22$ en $p(x) * (\frac{23}{44})$ bij $n = 23$. Voor $n = 22$ levert dit volgende plot op:



Als laatste kijken we naar een herschaalde plot van $n = 23$:



Dit is waarschijnlijk het beste wat we in de buurt gaan komen van een interpolant die $\cos(11x)$ benadert. De functie loopt door de gegeven punten, volgt de vorm en haar beeld blijft ongeveer in het interval $[-1, 1]$

4 Code

Nog een klein woordje over de code: normaal schrijf ik veel stukken zodat elke mogelijke oplossing telkens bij het runnen wordt gegenereerd. Deze keer heb ik vaak zoveel liggen aanpassen dat het me interessanter leek om sommige regels soms niet te laten compileren. Ik heb steeds bij deze regels een stuk commentaar geschreven over wat ze doen indien ze toch worden gecompileerd.