



Universiteit Utrecht

WISKUNDE BACHELORSCRIPTIE

**16e-eeuwse numerieke oplossingsmethodes voor
polynomen**

Stein Meereboer

Begeleid door
Dr. Steven Wepster

16 januari 2020

Inleiding

In 1596 wordt Vanden Cirkel gepubliceerd, de auteur van Vanden Cirkel is Ludolph van Ceulen. Van Ceulen was een rekenmeester en schermleeraar geboren in 1540 te Hildesheim in Duitsland. Rond 1562 vestigt Ludolph zich in Delft. Op 56-jarige leeftijd publiceert hij zijn boek Vanden Cirkel. Vanden cirkel is een resultaat van 10 jaar onderzoek naar cirkels en ingeschreven veelhoeken [11].

Een beroemd resultaat in het boek is de benadering van π , de irrationale constante wordt benaderd met een precisie van 20 decimalen. Later wordt deze benadering door Ludolph verbeterd tot 35 decimalen. Dit is niet het enige wat in Vanden Cirkel wordt besproken, veel meetkundige constructies in de cirkel worden in de eerste hoofdstukken besproken. Er staan sinus, tanges en secanten tabellen voor landmeters in het boek, werk over de interestrekening, en in hoofdstuk 14 bespreekt Ludolph hoe we de lengtes van zijdes van ingeschreven regelmatige n -hoeken in een cirkel van diameter twee berekenen. Het blijkt dat we deze lengtes kunnen vinden door nulpunten van polynomen te berekenen. In de loop van hoofdstuk 14 legt Ludolph netjes uit hoe we voor enkele n -hoeken de polynomen verkrijgen. Niet alleen de polynomen worden gegeven, maar ook de gevraagde oplossingen van de polynomen. In hoofdstuk 14 geeft Ludolph deze oplossingen tot een precisie van wel 33 decimalen! Achter aan hoofdstuk 14 geeft Ludolph een overzicht van de lengte van alle zijdes van de ingeschreven n -hoeken voor $1 \leq n \leq 80$ met een precisie van 14 decimalen.

Deze bachelorscriptie zal zich bezighouden met hoofdstuk 14 uit Vanden Cirkel. Het meest interessante van hoofdstuk 14 is dat Ludolph niet vermeldt hoe hij de oplossingen van de polynomen met zulke hoge precisie berekent. De polynomen die Van Ceulen opstelt in hoofdstuk 14 worden snel van enorm hoge graad. Zo vinden we de zijde van een 34-hoek door een polynoom van graad 16 op te lossen, en de zijde van een 118-hoek door een polynoom van graad 60 op te lossen. Het oplossen van deze polynomen is zeker geen triviale taak, echter besteedt Van Ceulen hier geen aandacht aan. Hij laat het in het midden hoe hij deze polynomen van enorme graad met hoge precisie heeft opgelost. Tot op heden weten we niet hoe hij dit heeft gedaan.

Het doel van deze bachelorscriptie is om erachter te komen hoe Ludolph deze polynomen heeft opgelost. Dit doel is wellicht te hoog gegrepen voor een korte bachelor scriptie vandaar ook de titel van de bachelorscriptie: 16e-eeuwse numerieke oplossingsmethodes voor polynomen. We verzamelen mogelijke numerieke methoden die Van Ceulen gebruikt heeft met als achterliggende gedachte het oplossen van de polynomen uit hoofdstuk 14. In deze scriptie wordt een overzicht gegeven van enkele numerieke oplossingsmethodes voor polynomen die bekend waren in de 16e eeuw. Deze methoden worden niet alleen gegeven, ook de juistheid van de methoden zal met moderne wiskunde bewezen worden. Zo kan deze scriptie gezien worden als een klein handboek voor de 16e -eeuwse wiskundige om polynomen op te lossen, in de zin dat we bekende methodes voor het oplossen van polynomen uit de 16e eeuw bespreken.

Niet alle waardes die Van Ceulen geeft in hoofdstuk 14 zijn correct. Er worden ook enkele fouten gemaakt. Sommige hiervan zijn overduidelijk drukfouten terwijl enkele ook geïnterpreteerd kunnen worden als rekenfouten van Van Ceulen. De rekenfouten geven ons houvast. Als een numerieke methode dezelfde fouten produceert, dan zou de methode wellicht de methode kunnen zijn die Van Ceulen gebruikt heeft. Helaas komt het niet veel voor dat Ludolph overduidelijke rekenfouten maakt en is onze enige houvast een handje vol mogelijke rekenfouten. Elk resultaat dat we krijgen is een stap in de goede richting. Als we enkele numerieke methoden kunnen uitsluiten, dan hebben we al meer kennis dan voordat deze scriptie geschreven is.

De scriptie begint met een bespreking van hoofdstuk 14. We willen laten zien hoe Ludolph aan de polynomen is gekomen, en bovendien welk van de vele oplossingen van de polynomen we moeten beschouwen voor de lengtes van de zijdes van de ingeschreven n -hoeken. Daarnaast kijken we naar de oplossingen die Van Ceulen geeft in hoofdstuk 14. In dit hoofdstuk nemen we de gemaakte fouten onder de loep. Er wordt een overzicht gegeven van welke fouten wellicht rekenfouten zijn. Nadat we deze basis hebben gelegd, worden er numerieke methodes besproken die Van Ceulen wellicht gebruikt heeft. Hier proberen we ook na te rekenen of deze numerieke methoden dezelfde fouten geven als in hoofdstuk 14. Ten slotte leggen we de besproken numerieke methoden naast elkaar en geven aan de hand hiervan een advies over de waarschijnlijkheid van het gebruik van Van Ceulen van deze methoden.

Verder wil ik graag Steven Wepster bedanken voor zijn begeleiding tijdens het schrijven van mijn scriptie. Voor het corrigeren van inhoudelijke fouten, schrijf en taalfouten in vorige versies van de scriptie. Voor het in de goede richting sturen in de literatuur. Maar ook voor de vele kopjes koffie en thee.

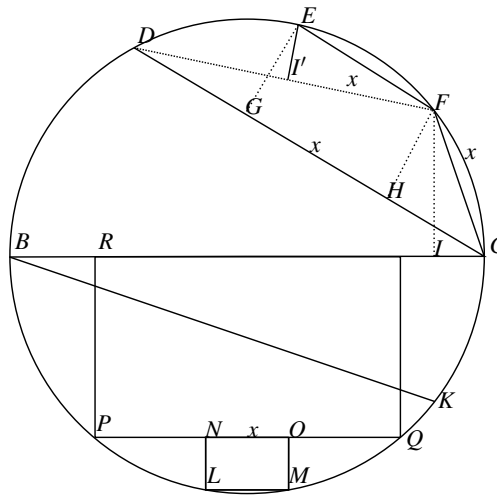
Inhoudsopgave

1	De polynomen van van Ceulen	5
1.1	Lengte van een zijde van een ingeschreven 9-hoek in een cirkel van diameter 2	5
1.2	Lengte van een zijde van een ingeschreven 18-hoek in een cirkel van diameter 2	6
1.3	Lengte van een zijde van een ingeschreven 34-hoek in een cirkel van diameter 2	7
1.4	Oplossingen van de polynomen	9
1.4.1	Vermedigvuldiginpolynomen	10
1.4.2	Geneste wortels	12
1.5	Opmerkingen over de oplossingen	14
2	Oplossingen	15
2.1	Beschouwing van de fouten	18
2.1.1	Tabel 1	18
2.2	Hoe verder	18
3	Interpolatie methode van Chuquet	20
3.1	Het algoritme	20
3.2	Werking algoritme	20
3.2.1	Iteratie van de interpolatie	22
3.3	Toegepast op het polynoom van de ingeschreven 18-hoek in een cirkel van diameter 2	23
3.4	Interpolatie via Python	25
3.4.1	Van breuken naar decimalen	26
4	De vastepunt iteratie van Chelebi	28
4.1	Het algoritme	28
4.2	Werking algoritme	28
4.3	Voorbeeld	29
5	Vieta's iteratieve methode	32
5.1	Het algoritme	32
5.2	Werking algoritme	33
5.2.1	Van de Newton-Raphson naar Vieta's methode	35
5.3	Voorbeeld van Vieta's iteratieve methode op de ingeschreven regelmatige 18-hoek	36
5.4	Fouten reproduceren uit tabel 2.1.1	37
5.4.1	De ingeschreven regelmatige 27-hoek in cirkel van diameter 2	37
5.4.2	De ingeschreven regelmatige 35-hoek in cirkel van diameter 2	38
5.4.3	De ingeschreven regelmatige 40-hoek in cirkel van diameter 2	38
5.4.4	De ingeschreven regelmatige 55-hoek in cirkel van diameter 2	38
5.4.5	De ingeschreven regelmatige 63-hoek in cirkel van diameter 2	39
5.4.6	De ingeschreven regelmatige 70-hoek in cirkel van diameter 2	39
5.4.7	Conclusie	40
5.5	Opmerkingen algoritme	40
5.5.1	Aanpassingen op het algoritme	40
5.5.2	Snelheid	41
5.5.3	Fouten in tabel 2	41
6	Vergelijken van de besproken methoden	42
6.1	Rekenfouten	42
6.1.1	Rekenfouten bij de interpolatie van Chuquet	42
6.1.2	Rekenfouten bij de iteratieve methode van Vieta	42
6.2	Snelheid algoritme	43
6.2.1	Snelheid iteratieve methode van Vieta	43
6.2.2	Snelheid interpolatie van Chuquet	43

7	Slotwoord	45
7.1	Advies voor vervolgonderzoek	45
7.1.1	Chuquet	45
7.1.2	Vieta	46
7.1.3	Literatuur	46

1 De polynomen van van Ceulen

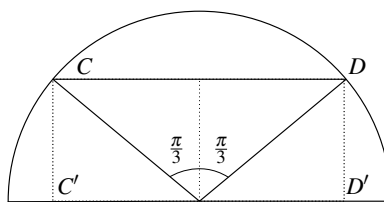
Hoofdstuk 14 van Vanden Cirkel van Ludolph van Ceulen heeft als doel het benaderen van de omtrek van ingeschreven regelmatige n -hoeken in een cirkel van diameter twee. Dit doet Ludolph door de lengte van één zijde van de regelmatige n -hoek te benaderen. De lengte van een zijde wordt gegeven door een nulpunt van een polynoom. Het doel van deze paragraaf is, om te laten zien hoe Ludolph deze polynomen verkrijgt. De uitleg die we geven is gelijk aan de uitleg die van Ceulen geeft in Hoofdstuk 14 uit Vanden Cirkel, echter is de uitleg meer uitgebreid dan de originele tekst, daar we hebben geprobeerd elke stap te onderbouwen.



Figuur 1: Cirkel met ingeschreven 9- & 18-hoeken

1.1 Lengte van een zijde van een ingeschreven 9-hoek in een cirkel van diameter 2

In het bovenstaande figuur is DC een zijde van een regelmatige 3-hoek, en EF een zijde van een regelmatige 9-hoek. Ons doel is de lengte van EF te vinden. De lengte van DC vinden we door DC te projecteren.



Figuur 2: Projectie van CD

Dan vinden we dat $CD = \cos(\frac{\pi}{2} - \frac{\pi}{3}) - \cos(\frac{\pi}{2} + \frac{\pi}{3}) = \cos(\frac{2\pi}{3}) - \cos(\frac{\pi}{3}) = \frac{\sqrt{3}}{2} - \frac{-\sqrt{3}}{2} = \sqrt{3}$. Laat EF , FC en GH allen gelijk aan x , en $EG \perp DC$ en $FH \perp DC$. Dan vinden we dat $HC = \sqrt{\frac{3}{4} - \frac{x}{2}}$, want

$$DC = DG + GH + HC = 2HC + GH.$$

Hieruit volgt dat $2HC = \sqrt{3} - x$, en dus dat $HC = \sqrt{\frac{3}{4} - \frac{x}{2}}$. Zo vinden we ook dat

$$HD = DC - HC = \sqrt{3} - (\sqrt{\frac{3}{4} - \frac{x}{2}}) = \sqrt{\frac{3}{4} + \frac{x}{2}}.$$

Nu kunnen we met behulp van Pythagoras de lengte van HF^2 berekenen. Uit Pythagoras volgt dat

$$FC^2 - HC^2 = x^2 - \left(\sqrt{\frac{3}{4}} - \frac{x}{2}\right)^2 = x^2 - \left(\frac{3}{4} - \sqrt{\frac{3}{4}x^2} + \frac{x^2}{4}\right) = \frac{3x^2}{4} + \sqrt{\frac{3}{4}x^2} - \frac{3}{4}.$$

Nu we de lengte HF weten kunnen we hiermee, opnieuw met Pythagoras, de lengte DF berekenen. Dit gaat als volgt

$$DF^2 = HD^2 + HF^2 = \left(\sqrt{\frac{3}{4}} + \frac{x}{2}\right)^2 + \frac{3x^2}{4} + \sqrt{\frac{3}{4}x^2} - \frac{3}{4} = \sqrt{\frac{3}{4}x^2} + \frac{3}{4} + \frac{x^2}{4} + \frac{3x^2}{4} + \sqrt{\frac{3}{4}x^2} - \frac{3}{4} = x^2 + \sqrt{3x^2}.$$

Ons doel nu is om een andere uitdrukking van DF te vinden in termen van x . Laat I het punt op BC zodat $FI \perp BC$. De driehoek $\triangle CBF$ is gelijkvormig met $\triangle FCI$. Volgens Thales is de hoek $\angle BFC$ recht, ook geldt dat $\angle FCI$ recht is, en dus dat de hoeken $\angle BFC$ en $\angle FCI$ gelijk aan elkaar zijn. Ook geldt dat $\angle FCB = \angle FCI$. Twee gelijken hoeken maakt een gelijkvormige driehoek, dus de driehoek $\triangle CBF$ is gelijkvormig met $\triangle FCI$. Hieruit volgt dat $\frac{FI}{FC} = \frac{BF}{CB}$, dus we vinden dat

$$\frac{FI}{x} = \frac{\sqrt{4-x^2}}{2},$$

waaruit volgt dat

$$FI = \frac{\sqrt{4x^2 - x^4}}{2}.$$

Teken het punt I' zodat EI' loodrecht op DF staat. Dan vinden we dat $\angle DFI' = 2\angle DEF$, en dus dat $DF = 2DI'$. Omdat de hoek $\angle EI'F$ recht is, $\angle CIF$ recht is, en de gelijkheid $\angle I'FE = \angle IFC$ geldt, zijn de driehoeken $\triangle EI'F$ en $\triangle FIC$ gelijkvormig. Omdat $EF = FC$ volgt dat $FI = FI'$, dus we vinden dat $DF = 2FI$. Eerder hebben we gevonden dat $DF = \sqrt{x^2 + \sqrt{3x^2}}$. Bovendien geldt ook dat $DF = \sqrt{4x^2 - x^4}$, dus de uitdrukkingen $\sqrt{x^2 + \sqrt{3x^2}}$ en $\sqrt{4x^2 - x^4}$ zijn gelijk aan elkaar. Door beiden kanten te kwadrateren volgt

$$x^2 + \sqrt{3x^2} = 4x^2 - x^4,$$

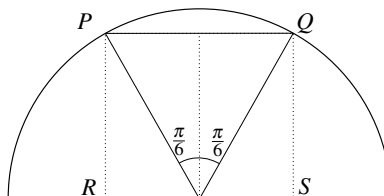
dus $\sqrt{3x^2} = 3x^2 - x^4$. Hier uit volgt dat $3x - x^3 = \sqrt{3}$. Om de lengte van de zijde van een 9-hoek te berekenen hoeven we nu alleen de vergelijking

$$3x - x^3 = \sqrt{3} \tag{1}$$

op te lossen.

1.2 Lengte van een zijde van een ingeschreven 18-hoek in een cirkel van diameter 2

In figuur 1 is LM een zijde van een ingeschreven 18-hoek. Laat de lengte van LM gelijk aan x . PQ is een zijde van een ingeschreven 6 hoek. De lengte van PQ vinden we weer door te projecteren.



Figuur 3: Projectie van PQ

Dan vinden we dat $PQ = \cos(\frac{\pi}{2} - \frac{\pi}{6}) - \cos(\frac{\pi}{2} + \frac{\pi}{6}) = \cos(\frac{2\pi}{6}) - \cos(\frac{\pi}{6}) = \frac{1}{2} - \frac{-1}{2} = 1$. De lengte van RP is gelijk aan $\sin(\frac{\pi}{6}) = \sqrt{\frac{3}{4}}$. Dan vinden we dat de lengte PN gelijk is aan $\frac{1}{2} - \frac{x}{2}$ en de lengte NQ gelijk aan

$\frac{1}{2} + \frac{x}{2}$. De segmenten PL en MQ hebben lengte x . Dan vinden we met behulp van Pythagoras de lengte van NL^2 en LQ^2 . Dit gaat als volgt

$$NL^2 = PL^2 - PN^2 = x^2 - \left(\frac{1}{2} - \frac{x}{2}\right)^2 = x^2 - \left(\frac{1}{4} - \frac{x}{2} + \frac{x^2}{4}\right) = \frac{3x^2}{4} + \frac{x}{2} - \frac{1}{4}$$

$$LQ^2 = NL^2 + NQ^2 = \frac{3x^2}{4} + \frac{x}{2} - \frac{1}{4} + \left(\frac{1}{2} + \frac{x}{2}\right)^2 = \frac{3x^2}{4} + \frac{x}{2} - \frac{1}{4} + \frac{1}{4} + \frac{x}{2} + \frac{x^2}{4} = x^2 + x.$$

Laat de lengte van KC even lang als LQ , dus de lengte van KC is gelijk aan $\sqrt{x^2 + x}$. Volgens de stelling van Thales is de hoek $\angle BKC$ recht. Daarom volgt uit Pythagoras dat $BK^2 = BK^2 - KC^2 = 4 - x^2 - x$. Met behulp hiervan kunnen we de lengte van LM vinden. Noem $\angle CBK = \beta$, we zien dat $\frac{BK}{BC} = \frac{BK}{2} = \cos(\beta)$. Omdat $LM = MQ$ en $LQ = KC$ volgt dat $\sin\left(\frac{\beta}{2}\right) = \frac{LM}{BC} = \frac{LM}{2}$, dus we vinden dat $LM = 2\sin\left(\frac{\beta}{2}\right)$, ook wel $LM^2 = 4\sin^2\left(\frac{\beta}{2}\right)$. We weten tevens dat $\cos(\beta) = 1 - 2\sin^2\left(\frac{\beta}{2}\right)$, hieruit volgt dat

$$\left(2\sin^2\left(\frac{\beta}{2}\right)\right) = 1 - \cos(\beta),$$

ook wel $\frac{LM^2}{2} = 1 - \cos(\beta)$. Ten slotte vinden we dat $LM^2 = 2 - 2\cos(\beta) = 2 - \sqrt{4 - x^2 - x}$. We vinden dus dat $x = \sqrt{2 - \sqrt{x^2 - x}}$. Door beiden kanten te kwadrateren volgt er dat $x^2 = 2 - \sqrt{x^2 - x}$, hieruit volgt dat $x^2 - 2 = -\sqrt{x^2 - x}$ dus $4 - 4x^2 + x^4 = x^2 - x$, ook wel $3x^2 - x^4 = x$. Omdat x de lengte van LM is en deze ongelijk aan nul is, vinden we dat $3x - x^3 = 1$. Om de lengte van LM te vinden moeten we de vergelijking

$$3x - x^3 = 1 \tag{2}$$

oplossen.

We hebben nu de lengtes van een 9-hoek en een 18-hoek gevonden in termen van polynomen. In beiden gevallen berekenen we de lengte van een $3n$ -hoek met behulp van de lengte van een zijde van een n -hoek. Het volgende blijkt waar te zijn.

Stelling 1.1. Zij x de lengte van de zijde van een ingeschreven $3n$ -hoek in een cirkel met diameter 2. Dan is de lengte van de zijde van een ingeschreven regelmatige n -hoek gelijk aan $3x - x^3$.

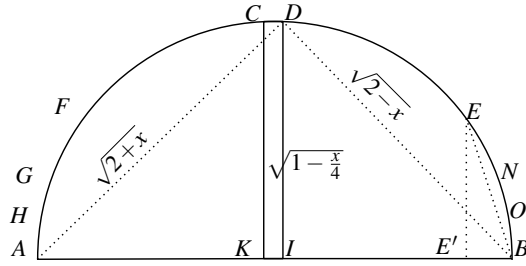
In de vorige gevallen zijn de lengtes van de n -hoeken makkelijk te berekenen, hierdoor is het verkregen polynoom een derdegraads polynoom. Deze lagere-graads polynomen kunnen we zonder veel moeite algebraïsch oplossen. Met behulp de stelling kunnen we bijvoorbeeld de lengte van een 27-hoek berekenen. Dit doen we op de volgende manier: Zij x de lengte van een zijde van een ingeschreven 27-hoek, dan vinden we dat de volgende gelijkheid geldt:

$$3x - x^3 = \text{zijde 9-hoek}.$$

In hoofdstuk 14 vermeld Ludolph dat deze stelling bekend was voor hem. Ten slotte wil ik nog kijken naar de lengte van een 34-hoek, hierbij verkrijgen we niet een polynoom van graad 3, maar een polynoom van graad 16.

1.3 Lengte van een zijde van een ingeschreven 34-hoek in een cirkel van diameter 2

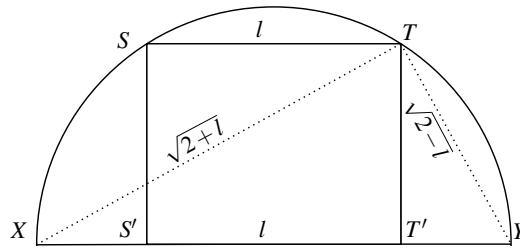
In het onderstaande figuur is het segment CD een zijde van een 34-hoek. Noem de lengte van CD x . De cirkelboog $AFCD$ is negen keer zo groot als CD , en DEB is achtmaal zo groot als de cirkelboog CD . CD is een zijde van een ingeschreven 34 hoek, dan wordt de helft van een volledige cirkelboog opgevuld met 17 zijdes van de 34 hoek. Links vanaf CD passen er 8 zijdes van de 34-hoek tot A . Er passen ook 8 zijdes van een 34-hoek rechts van CD tot B . Hieruit volgt dat de cirkelboog $AFCD$ 9 keer zo groot is als CD , en de cirkelboog DEB 8 keer zo groot als CD . Het punt E ligt op het midden van de cirkelboog DB , het punt N precies op het midden van de cirkelboog EB , het punt O precies op het midden van de cirkelboog NB en F ligt op het midden van de cirkelboog AC . Hieruit volgt dat OB even lang is als CD .



Figuur 4: Cirkel met ingeschreven 34-hoek

Een belangrijk resultaat waarmee we de lengte CD zullen vinden is het volgende resultaat: Zij de punten A, B, C en D gedefinieerd zoals hierboven, en de lengte van CD x . Dan is de lengte van AD gelijk aan $\sqrt{2+x}$. Dit resultaat zullen we vaak toepassen, we kunnen een algemeen resultaat formuleren als een stelling.

Stelling 1.2. Gegeven een cirkel van diameter 2 en punten X, Y gegeven zodat de lengte XY gelijk is aan de diameter van de cirkel. Gegeven zijn ook de punten S en T zodat $XS = TY$. Bovendien is de cirkelboog $XSTY$ gelijk aan de cirkelboog XY . Zij de lengte ST gelijk aan l , dan is de lengte XT gelijk aan $\sqrt{2+l}$ en de lengte TY gelijk aan $\sqrt{2-l}$.



Figuur 5: Een voorbeeld van punten XY en ST op een cirkel van diameter 2.

Bewijs. De driehoeken $\triangle XT'T$, $\triangle XTY$ en $\triangle TYT'$ zijn gelijkvormig. Alle driehoeken hebben een rechte hoek, respectievelijk $\angle XT'T$, $\angle XTY$ en $\angle TYT'$. Verder bevatten $\triangle XT'T$ en $\triangle XTY$ beide $\angle T'XT$, en $\triangle XTY$ en $\triangle TYT'$ bevatten beide $\angle T'YT$. Twee gelijke hoeken maken gelijkvormige driehoeken, dus de driehoeken $\triangle XT'T$, $\triangle XTY$ en $\triangle TYT'$ zijn gelijkvormig. Uit gelijkvormigheid volgt dat

$$\frac{T'Y}{T'T} = \frac{T'T}{XT'} \text{ dus } T'T^2 = T'X \cdot T'Y.$$

We vinden dat $T'Y = 1 - \frac{l}{2}$ en, dat $XT' = 1 + \frac{l}{2}$. Hieruit volgt dat $T'T = \sqrt{\left(1 - \frac{l}{2}\right) \left(1 + \frac{l}{2}\right)} = \sqrt{1 - \frac{l^2}{4}}$.

Dan volgt uit Pythagoras dat

$$XT^2 = XT'^2 + T'T^2 = 1 - \frac{l^2}{4} + 1 + l + \frac{l^2}{4} = 2 + l$$

en

$$TY^2 = YT'^2 + T'T^2 = 1 - \frac{l^2}{4} + 1 - l + \frac{l^2}{4} = 2 - l.$$

Hiermee hebben we bewezen dat de lengte XT gelijk is aan $\sqrt{2+l}$ en, dat $TY = \sqrt{2-l}$. □

Nu kunnen we deze stelling toepassen op de punten op onze cirkel. Omdat $AB = 2$, $AC = BD$ en $CD = x$, volgt dat $AD = \sqrt{2+x}$ en $DB = \sqrt{2-x}$.

Merk op dat de cirkelboog FC 4 keer zo groot is als de cirkelboog CD , omdat F op het midden van AC ligt, dit geldt ook voor de cirkelboog DE , dus de cirkelboog FE is negen keer zo groot als CD . We hebben ook gevonden dat de cirkelboog AD negen keer zo groot is als CD . Hieruit volgt dat $AD = FE$, dus $FE = \sqrt{2+x}$.

Het punt F ligt op het midden van de cirkelboog AC . Het punt E ligt op het midden van de cirkelboog DB . Omdat $AC = DB$, volgt dat $AF = EB$. De punten A en F voldoen aan de voorwaarden van **Stelling 1.2**. Omdat $FE = \sqrt{2+x}$, volgt uit **Stelling 1.2** dat de lengte van AE gegeven wordt door $\sqrt{2+\sqrt{2+x}}$.

Plaats nu G op het midden van de cirkelboog AF . Dan is de cirkelboog GN 13 keer zo groot als de cirkelboog CD . GF is twee keer zo groot als CD evenals EN , en FE is 9 keer zo groot als CD , dus GN is $9+4$ keer zo groot. De cirkelboog AE is ook 13 keer zo groot als de cirkelboog CD , want AB is 17 keer zo groot als CD en de cirkelboog AE is de cirkelboog AB min de cirkelboog BE . De cirkelboog BE is vier keer zo groot als de cirkelboog CD . Hieruit volgt dat de cirkelboog AE is 13 keer zo groot als de cirkelboog CD . We concluderen dat de cirkelbogen van AE en GN gelijk aan elkaar zijn waaruit volgt dat $AE = GN$. Dus we vinden dat $GN = \sqrt{2+\sqrt{2+x}}$.

Het punt G ligt op het midden van de cirkelboog AF . Het punt N ligt op het midden van de cirkelboog EB . Bovendien geldt dat $AF = EB$. Hieruit volgt dat $AG = NB$. Daarom kunnen we **Stelling 1.2** opnieuw toepassen. Omdat de lengte van GN gelijk is aan $2 + \sqrt{2+\sqrt{2+x}}$ volgt uit **Stelling 1.2** dat de lengte van AN gelijk is aan $\sqrt{2+\sqrt{2+\sqrt{2+x}}}$.

Het punt H ligt op het midden van de cirkelboog AG . Het punt O ligt op het midden van de cirkelboog NB . Omdat GF twee keer zo groot is als CD evenals NB geldt dat $AG = CD = OB$. Daarom is de cirkelboog HO 15 keer zo groot als CD . Ook geldt dat de cirkelboog NB 15 keer zo groot is als CD , omdat deze cirkelboog de cirkelboog AB is min de cirkelboog NB . De cirkelboog NB is twee keer zo groot als CD , dus de cirkelboog NB is $17-2 = 15$ keer zo groot als de cirkelboog CD . Hieruit volgt dat $AN = HO$. We concluderen dat $HO = \sqrt{2+\sqrt{2+\sqrt{2+x}}}$. Omdat $AH = OB$ volgt uit **Stelling 1.2** dat

$$OB = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{2 + x}}}} \quad (3)$$

Omdat OB even lang is als CD volgt dat $CD = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{2 + x}}}}$. Om de lengte van CD te vinden moeten we de vergelijking (3) = x oplossen. Als we deze vergelijking omschrijven naar een veelterm volgt:

$$2 - \sqrt{2 + \sqrt{2 + \sqrt{2 + x}}} = x^2, \quad \text{dus} \quad 2 - x^2 + x^4 = \sqrt{2 + \sqrt{2 + x}}.$$

Wanneer we weer de uitdrukking kwadrateren, volgt dat $x^8 - 8x^6 + 20x^4 - 16x^2 + 2 = \sqrt{2+x}$. Het resulterende polynoom dat we willen oplossen om de lengte CD te vinden is:

$$x^{16} - 16x^{14} + 104x^{12} - 352x^{10} + 660x^8 - 672x^6 + 336x^4 - 64x^2 + 2 - x = 0. \quad (4)$$

1.4 Oplossingen van de polynomen

Van Ceulen is hier niet volledig. Hij vertelt ons niet welk van de vele oplossingen van de gegeven polynomen we moeten beschouwen voor de lengte van de zijde van de n -hoeken. Zo heeft het polynoom verkregen bij (2) drie oplossingen, en het polynoom verkregen bij (4) 16 oplossingen. Het doel van deze paragraaf is om te laten zien dat er een manier is om te zien welk van de vele nulpunten, de gevraagde lengte is van de ingeschreven veelhoek. Hiervoor vertellen we eerst iets meer over de manier hoe Van Ceulen tot zijn vergelijkingen komt. Het blijkt dat we deze vergelijkingen op twee manieren kunnen classificeren. De eerste hiervan zijn geneste wortels, de tweede noem ik vermedigvuldiginopolynomen. Een voorbeeld van een vermedigvuldiginopolynomen is het polynoom dat we zagen bij **Stelling 1.1**, het blijkt dat er meer van dit soort

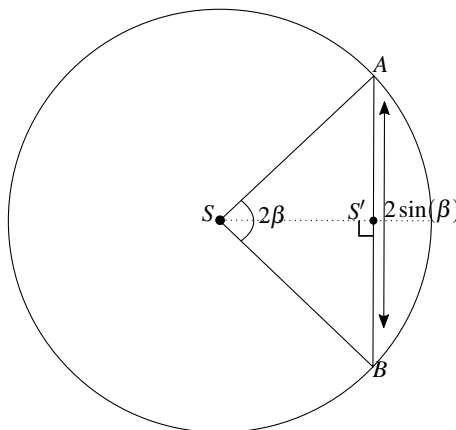
polynomen zijn voor het vermenigvuldigen van hoeken. Voor de oplossingen van de vermedigvuldiginspolynomen zijn we altijd op zoek naar de kleinste positieve oplossing van de vergelijking. Bij de geneste wortels hebben we een makkelijke test. We zullen eerst de vermedigvuldiginspolynomen bespreken.

1.4.1 Vermedigvuldiginspolynomen

Doorgaans hoofdstuk 14 gebruikt Van Ceulen de volgende vermenigvuldigs polynomen.

1. Zij x de lengte van de zijde van een ingeschreven $3n$ -hoek in een cirkel met diameter 2. Dan is de lengte van de zijde van een ingeschreven regelmatige n -hoek gelijk aan $3x - x^3$.
2. Zij x de lengte van de zijde van een ingeschreven $5n$ -hoek in een cirkel met diameter 2. Dan is de lengte van de zijde van een ingeschreven regelmatige n -hoek gelijk aan $5x - 5x^3 + x^5$.
3. Zij x de lengte van de zijde van een ingeschreven $7n$ -hoek in een cirkel met diameter 2. Dan is de lengte van de zijde van een ingeschreven regelmatige n -hoek gelijk aan $7x - 14x^3 + 7x^5 - x^7$.
4. Zij x de lengte van de zijde van een ingeschreven $9n$ -hoek in een cirkel met diameter 2. Dan is de lengte van de zijde van een ingeschreven regelmatige n -hoek gelijk aan $9x - 30x^3 + 27x^5 - 9x^7 + x^9$.

We zullen alle polynomen individueel bespreken. Hierbij bespreken we wat alle positieve oplossingen zijn en wat deze oplossingen meetkundig betekenen. Als we de meetkundige betekenis van de oplossingen weten, dan weten we ook welke oplossing de lengte van een gevraagde zijde is. Hier geven we alleen de resultaten, we geven geen bewijs voor de resultaten. De resultaten zijn overgenomen uit [9, p. 445]. Eerst bewijzen we een hulplemma, met behulp van dit lemma kunnen we de oplossingen meetkundig interpreteren.



Figuur 6: Een voorbeeld van punten AB en hoek β op een cirkel van diameter 2.

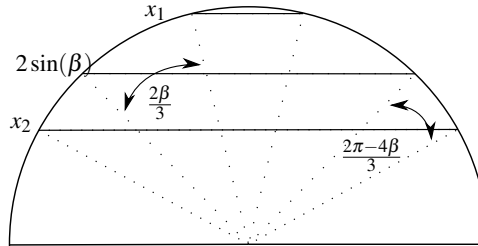
Lemma 1.3. Zij S het middelpunt van een cirkel van diameter twee. Laat AB een koorde zodat de hoek $\angle ASB$ gelijk is aan 2β . Dan is de lengte van de koorde AB gelijk aan $2 \sin(\beta)$.

Bewijs. We plaatsen S' precies zodat $SS' \perp AB$. De lijn $S'S$ snijdt de hoek $\angle ASB$ precies door de helft zodat $\angle S'SA = \angle S'SB = \beta$. Hieruit volgt dat $AS' = S'B = \frac{AB}{2}$. We vinden dat $AS' = \sin(\beta)$. Hieruit volgt direct dat $AB = 2 \sin(\beta)$. Hiermee hebben we het lemma bewezen. \square

We zullen nu de oplossingen van de vermenigvuldigs polynomen bespreken.

1. Laat $\beta \in [0, \frac{\pi}{2})$. Dan heeft de vergelijking $3x - x^3 = 2 \sin(\beta)$ twee positieve oplossingen. De eerste oplossing wordt gegeven door $x_1 = 2 \sin(\frac{\beta}{3})$, de tweede oplossing wordt gegeven door $x_2 = 2 \sin(\frac{\pi - \beta}{3})$.

Als we nu aannemen dat $2\sin(\beta)$ de lengte is van een n -hoek. Dan weten we met behulp van **Lemma 1.3** dat x_1 de lengte is van een $3n$ -hoek. De tweede positieve oplossing van dit polynoom is de lengte van een koorde horend bij een hoek die twee maal het verschil is tussen $\frac{\beta}{3}$ en $\frac{2\pi}{3}$. Omdat $\beta \in [0, \frac{\pi}{2})$, zal $\frac{\beta}{3} < \frac{\pi-\beta}{3} \leq \pi$. Omdat de sinus monotoon stijgend is op $[0, \pi]$, zal $2\sin(\frac{\beta}{3}) < 2\sin(\frac{\pi-\beta}{3})$. Als gevolg hiervan wordt de lengte van de $3n$ -hoek gegeven door de kleinste positieve oplossing van het polynoom $3x - x^3 = 2\sin(\beta)$.



Figuur 7: Een meetkundige interpretatie van de positieve oplossingen van $3x - x^3 = 2\sin(\beta)$.

2. Laat $\beta \in [0, \pi)$. Dan heeft het polynoom $5x - 5x^3 + x^5 = 2\sin(\beta)$ maximaal drie positieve oplossingen. Neem aan dat $2\sin(\beta)$ de lengte is van een n -hoek. De eerste oplossing wordt gegeven door $x_1 = 2\sin(\frac{\beta}{5})$ en de tweede oplossing wordt gegeven door $x_2 = 2\sin(\frac{\pi-\beta}{5})$. De derde oplossing bestaat als $\frac{2\pi+\beta}{5} \leq \pi$, als dit het geval is dan wordt de derde oplossing gegeven door $2\sin(\frac{2\pi+\beta}{5})$. We hebben dat

$$0 < \frac{\beta}{5} \leq \frac{\pi-\beta}{5} \leq \frac{2\pi+\beta}{5} \leq \pi,$$

em omdat de sinus monotoon stijgend is op $[0, \pi]$, volgt opnieuw een rij ongelijkheden:

$$0 < 2\sin(\frac{\beta}{5}) \leq 2\sin(\frac{\pi-\beta}{5}) \leq 2\sin(\frac{2\pi+\beta}{5}) \leq 1.$$

Volgens **Lemma 1.3** is $x_1 = 2\sin(\frac{\beta}{5})$ de lengte van de $5n$ -hoek, dus we vinden dat de lengte van een zijde van de de $5n$ -hoek gelijk is aan het kleinste positieve oplossing van de vergelijking $5x - 5x^3 + x^5 = 2\sin(\beta)$.

3. Laat $\beta \in [0, \pi)$, dan heeft de vergelijking $7x - 14x^3 + 7x^5 - x^7 = 2\sin(\beta)$ vier positieve oplossingen. De positieve oplossingen $x_1, x_2, x_3, x_4 \in \mathbb{R} > 0$ worden gegeven door

$$\begin{aligned} x_1 &= 2\sin(\frac{\beta}{7}) \\ x_2 &= 2\sin(\frac{\pi-\beta}{7}) \\ x_3 &= 2\sin(\frac{2\pi+\beta}{7}) \\ x_4 &= 2\sin(\frac{3\pi-\beta}{7}) \end{aligned}$$

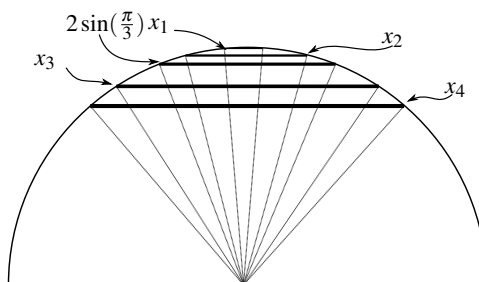
Omdat $\beta \in [0, \frac{\pi}{2})$, verkrijgen we de volgende rij ongelijkheden:

$$0 < \frac{\beta}{7} \leq \frac{\pi-\beta}{7} \leq \frac{2\pi+\beta}{7} \leq \frac{3\pi-\beta}{7} \leq \pi.$$

Hieruit volgt opnieuw een rij ongelijkheden in de sinus

$$0 < \sin(\frac{\beta}{7}) \leq \sin(\frac{\pi-\beta}{7}) \leq \sin(\frac{2\pi+\beta}{7}) \leq \sin(\frac{3\pi-\beta}{7}).$$

Volgens **Lemma 1.3** is $x_1 = 2\sin(\frac{\beta}{7})$ de lengte van de $7n$ -hoek, dus we vinden dat de lengte van een zijde van de de $7n$ -hoek gelijk is aan de kleinste oplossing van de vergelijking $7x - 14x^3 + 7x^5 - x^7 = 2\sin(\beta)$. Hieronder een meetkundige interpretatie van de oplossingen in het geval dat $\beta = \frac{\pi}{3}$.



Figuur 8: Een meetkundige interpretatie van de positieve oplossingen van $7x - 14x^3 + 7x^5 - x^7 = 2 \sin(\frac{\pi}{3})$.

4. Laat $\beta \in [0, \pi)$, dan heeft de vergelijking $9x - 30x^3 + 27x^5 - 9x^7 + x^9 = 2 \sin(\beta)$ vijf positieve oplossingen. Laat $2 \sin(\beta)$ de lengte van een n -hoek. Met dezelfde aanpak als voorheen vinden we dat het kleinste positieve nulpunt van de bovenstaande vergelijking gegeven is door $2 \sin(\frac{\beta}{9})$, waardoor we de lengte van de $9n$ -hoek vinden als de kleinste positieve oplossing van de vergelijking $9x - 30x^3 + 27x^5 - 9x^7 + x^9 = 2 \sin(\beta)$.

Nu weten we welke oplossingen we moeten beschouwen voor de gegeven vermenigvuldigs polynomen van Van Ceulen. In elk geval moeten we de kleinste positieve oplossing van het polynoom beschouwen. Ten slotte worden niet alle polynomen gegeven door de vermenigvuldigs polynomen. Andere komen voort uit geneste wortels of combinaties van geneste wortels en vermenigvuldigs polynomen. Het volgende gedeelte van deze paragraaf zullen we kijken naar de oplossingen van de geneste wortels. We bespreken hier opnieuw welke van de oplossingen we moeten beschouwen voor de lengtes van de ingeschreven veelhoeken.

1.4.2 Geneste wortels

De polynomen die voortkomen uit de geneste wortels hebben over het algemeen meer oplossingen dan we hopen. Zo hebben we gevonden dat (4) 16 oplossingen heeft, terwijl we naar een specifieke oplossing op zoek zijn. Daarom willen we een manier hebben om te weten welke van deze oplossingen, de gevraagde oplossing is. Om dit te bespreken, zullen we eerst laten zien wat we bedoelen met een geneste wortel. De geneste wortels hebben in het algemeen de gedaante:

$$\sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{\dots + \sqrt{2 \pm x}}}}} := f(x). \tag{5}$$

Hier is het teken in de laatste wortel vast, dit verschilt afhankelijk van de zijde die we willen vinden. De vergelijkingen die Ludolph in de loop van hoofdstuk 14 oplost zijn $f(x) = x$ of $f(x) = p(x)$, waar $p(x)$ een van de vermenigvuldigs polynomen is. Het blijkt dat de vergelijking $f(x) = x$ maar één oplossing heeft. Als we weten dat deze vergelijking maar één oplossing heeft, dan kunnen we gemakkelijk aanwijzen of een oplossing van het polynoom ook een oplossing is van de gevraagde geneste wortel. We controleren of de oplossing van het polynoom ook een oplossing is van de geneste wortel. In het geval dat we $f(x) = p(x)$ oplossen kunnen we nog steeds aanwijzen welke oplossing de oplossing is waar we naar op zoek zijn.

Om te bewijzen dat $f(x) = x$ maar één oplossing heeft, laat ik zien dat er minimaal één oplossing is, en dat deze oplossing uniek is. We laten eerst zien dat er een oplossing is. Om het makkelijker voor ons te maken nemen we aan dat $x \in [0, 1]$. Dit kunnen aannemen omdat Ludolph de geneste wortel voor het eerst gebruikt om de zijde van de ingeschreven 9-hoek te berekenen. De lengte van deze zijde is kleiner dan 1 en alle zijdes voor de hoeken groter dan de ingeschreven 9-hoek zullen dus ook kleiner dan 1 zijn. We kunnen dus aannemen dat $x \in [0, 1]$. Om te bewijzen dat er een oplossing is, definiëren we de functie $g : [0, 1] \rightarrow \mathbb{R}$, zodat

$$g(x) = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{\dots + \sqrt{2 \pm x} - x}}}.$$

We willen laten zien dat g een nulpunt heeft op $[0, 1]$. Dan heeft $f(x) = x$ een oplossing. We laten zien dat g een nulpunt heeft met behulp van de middelwaardstelling. We vinden dat

$$g(0) = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{\dots + \sqrt{2 \pm 0}}}} - 0} = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{\dots + \sqrt{2}}}}} > 0.$$

Bovendien geldt dat

$$g(1) = \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{\dots + \sqrt{2 \pm 1}}}} - 1} < \sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{\dots + \sqrt{2 - 1}}}} - 1} < \sqrt{2 - \sqrt{2}} - 1 < 0.$$

Omdat g een continue functie is volgt uit de middelwaardstelling dat g de waarde 0 aan neemt op het interval $[0, 1]$. We concluderen dat $f(x) = x$ een oplossing heeft op het interval $[0, 1]$. Nu hebben we het bestaan van een nulpunt aangetoond, ten slotte bewijzen we dat er g maar één nulpunt op dit interval heeft.

Stel dat $g(x) = f(x) - x$ meer dan één nulpunt heeft op het interval $[0, 1]$. Dan bestaan er $\xi, \eta \in [0, 1]$ met $\xi \neq \eta$ zodat $f(\xi) - \xi = g(\xi) = g(\eta) = f(\eta) - \eta$. Neem zonder verlies van algemeenheid aan dat $\xi > \eta$. We vinden dat

$$f(\xi) - f(\eta) = \xi - \eta.$$

Door beiden kanten in de vorige gelijkheid door $\xi - \eta$ te delen volgt:

$$\frac{f(\xi) - f(\eta)}{\xi - \eta} = 1.$$

Omdat we aangenomen hebben dat $\eta, \xi \in [0, 1]$ en $\xi > \eta$ volgt dat $0 < \xi - \eta < 1$. Daarom vinden we dat $f(\xi) - f(\eta) > 1$. Als $f(\xi) - f(\eta) > 1$, dan moet zeker ook gelden dat

$$\sup_{\xi, \eta \in [0, 1], \xi > \eta} f(\xi) - f(\eta) > 1.$$

We vinden dat $\sqrt{2 - \sqrt{2}} > f(y) > 0$ voor alle $y \in [0, 1]$. Hieruit volgt dat

$$\sqrt{2 - \sqrt{2}} \geq \sup_{\eta \in [0, 1]} \sqrt{2 - \sqrt{2}} - f(\eta) \geq \sup_{\xi, \eta \in [0, 1], \xi > \eta} f(\xi) - f(\eta) > 1.$$

Echter geldt er dat $\sqrt{2 - \sqrt{2}} < 1$, dit is in tegenspraak is met het gevonden resultaat. De aanname dat er meerdere nulpunten zijn leidt tot een tegenspraak, waaruit volgt dat er maar één nulpunt op het interval $[0, 1]$ ligt, dus we hebben laten zien dat vergelijking $f(x) = x$ een oplossing heeft. Bovendien is er maar één oplossing. Als we de een oplossing vinden van het polynoom voorgebracht uit $f(x) = x$, waarvoor geldt $x \in [0, 1]$ en deze voldoet aan de vergelijking $f(x) = x$. Dan weten we dat deze oplossing de gevraagde oplossing is.

Ten slotte bespreken we nog de oplossingen van $f(x) = p(x)$, waar $p(x)$ een van de verdubbelingspolynomen is. In dit geval is het vinden van oplossingen gelukkig makkelijk met de gevonden resultaten. We hebben gevonden dat $0 < f(x) < 1$. Omdat $0 < f(x) < 2$, volgt uit de resultaten van de vermedigvuldigin-polynomen dat de gevraagde oplossing de kleinste positieve oplossing van de vergelijking $f(x) = p(x)$ is.

We weten nu dus dat voor elk polynoom die Van Ceulen geeft in hoofdstuk 14 welke oplossing we willen beschouwen voor de lengte van de zijde van een ingeschreven n -hoek. Voor de verdubbelingspolynomen en combinaties met de geneste wortels moeten we de kleinste positieve oplossing beschouwen. Voor enkel de geneste wortels bestaat er maar één oplossing. Als een oplossing van het polynoom ook een oplossing is van de geneste wortel dan is het de lengte van de gevraagde zijde.

1.5 Opmerkingen over de oplossingen

We willen kort nog iets opmerken over de oplossingen die Van Ceulen geeft in hoofdstuk 14. Interessant om op te merken is dat bij Ludolph **Lemma 1.3** al bekend was. Ludolph gebruikt deze gelijkheid bijvoorbeeld op [1, p. 130]. Er is dus een gemakkelijke manier om de lengte van een zijde van een regelmatige veelhoek te vinden. We berekenen welke hoek bij de zijde van de regelmatige veelhoek, ookwel een koorde van de cirkel, hoort en noemen deze hoek 2β . Om de lengte van de bijbehorende koorde te vinden zoeken we in ons sinus tabel de waarde van $2\sin(\beta)$ op, en we hebben de lengte van de zijde van de regelmatige veelhoek. We lossen hier de problemen uit hoofdstuk 14 op, op een veel gemakkelijker en bovendien snellere manier. Als er zo'n makkelijke manier is, waarom lost Van Ceulen de problemen uit hoofdstuk 14 dan niet op deze manier op. De belangrijkste reden hiervoor is, dat er volgens [5] in de 16e eeuw de sinus tabellen alleen bekend tot 11 decimalen. Van Ceulen wil een hogere precisie, waarom zou hij anders zoveel tijd investeren als er een veel gemakkelijker methode bekend is. Daarom lost hij het bovenstaande probleem op een andere manier op.

Wat we hier ook uit kunnen halen, is dat we met de door Van Ceulen gegeven waarden een sinus tabel kunnen maken met een hogere precisie dan bekend in de tijd van Van Ceulen. Van Ceulen geeft bijvoorbeeld voor de lengte van de ingeschreven 9 hoek in een cirkel van diameter 2 de correcte waarde op 33 decimalen: 0.684040286651337466088199229364518, alleen het laatste decimaal is incorrect, deze had een 9 moeten zijn in plaats van een 8. Omdat we weten dat deze waarde gelijk is aan twee maal de sinus van $\frac{\pi}{9}$, weten we ook de waarde van $\sin(\frac{\pi}{9})$ met een precisie van 32 decimalen! Over het algemeen geeft Van Ceulen de waarde van alle lengtes van ingeschreven n -hoeken in een cirkel van diameter 2 tot de ingeschreven 80 hoek met een precisie van 14 decimalen. We zouden dus voor een aantal specifieke waardes van de sinus een betere sinustabel op kunnen stellen met een precisie van 14 decimalen.

2 Oplossingen

In hoofdstuk 14 wordt uitgelegd hoe we de lengte van een zijde van een ingeschreven n -hoek kunnen berekenen met behulp van polynomen. Bovendien worden de oplossingen van de opgestelde polynomen gegeven. Zo geeft Ludolph een oplossing voor het opgestelde polynoom van de ingeschreven 9-hoek. We willen het volgende vergelijking oplossen: $3x - x^3 = 1$, een benadering van een oplossing gegeven door Ludolph is $x = 0.684040286651337$. Een oplossing van dit polynoom met een precisie van 14 decimalen is 0.684040286651337 . In dit geval geeft Ludolph dus een benadering van 14 correcte decimalen! Voor het oplossen van derdegraads polynomen van de vorm $x^3 + px = q$ was in de 16e eeuw al een algemene methode bekend. Niccoló Fontana Tartaglia gaf in 1535 voor deze vergelijking algemene oplossingen, [8]. Alhoewel deze oplossingen in hier niet begrepen zouden worden door van Ceulen. Niccoló geeft hier de oplossing

$$x = \frac{1}{\sqrt[3]{2}} \left(\sqrt[3]{\sqrt{-3} - 1} + \sqrt[3]{-\sqrt{-3} - 1} \right).$$

Hier hebben we in de derdemachts wortel complexe getallen. In 1535 was er nog niets bekend over complexe getallen. Deze oplossing zorgt echter wel voor een oplossing die van Ceulen zelf had kunnen vinden omdat de getallen $\sqrt[3]{\sqrt{-3} - 1}$ en $\sqrt[3]{-\sqrt{-3} - 1}$ elkaars complex geconjugeerde zijn. Er geldt namelijk dat $\sqrt{-3} - 1 = -\sqrt{-3} - 1$, en complex geconjugeerden blijven behouden onder worteltrekking. Hieruit volgt dat de gevonden oplossing een reëel getal is.

Wonderbaarlijk genoeg geeft Ludolph ook benaderingen voor de hogere-graads vergelijkingen, zo geeft hij voor de ingeschreven 34-hoek in een cirkel van diameter 2 de benadering $x = 0.1845367189266039904794$. Dit is een benadering van het kleinste nulpunt van het polynoom

$$x^{16} - 16x^{14} + 104x^{12} - 352x^{10} + 660x^8 - 672x^6 + 336x^4 - 64x^2 - x + 2.$$

De werkelijke oplossing op 23 decimalen precies wordt gegeven door $0.18453671892660399047930$, alleen het laatste decimaal is dus incorrect. De 4 aan het eind van de benadering van Van Ceulen, had een 3 moeten zijn. Dit is niet het enige polynoom waarvoor Van Ceulen zo'n precieze benadering geeft, dit doet hij voor alle ingeschreven regelmatige n -hoeken in een cirkel van diameter 2. Van de regelmatige ingeschreven 2-hoek tot en met de ingeschreven 80-hoek. Verder geeft hij een aantal specifieke oplossingen voor hogere hoeken tot en met de 675-hoek.

Onder een aantal voorwaarden kunnen we zesdegraads polynomen terugbrengen naar derdegraads polynomen. Dit geldt ook voor vierdegraads polynomen. In het algemeen is het zeker niet gemakkelijk om algebraïsch oplossingen van hogere graads polynomen te vinden. We vinden polynomen leuk, tot een bepaalde graad. Dit is een van de redenen dat we geïntrigeerd zijn door de oplossingen die Van Ceulen geeft. Merkwaardig aan de oplossingen die van Ceulen geeft is dat Van Ceulen niet vermeldt hoe hij deze oplossingen heeft berekend. Hij moet een numerieke methode hebben gebruikt om de wortels van de polynomen te benaderen. Om de benaderingen van de nulpunten te verkrijgen, kom je er bijna niet onder uit om herhaald polynomen te evalueren. Dit alleen al is een tijdrovende klus, aangezien de graad van de polynomen zo hoog als graad 60 kunnen worden. Bereken maar eens een 60-ste macht van een niet triviaal getal zonder een computer.

Echter zijn niet alle de benaderingen in Vanden Cirkel correct. Sommige fouten in Vanden Cirkel zijn hoogstwaarschijnlijk drukfouten, terwijl andere meer op rekenfouten lijken. Achteraan hoofdstuk 14 wordt een tabel gegeven met de lengtes van de ingeschreven n -hoeken in een cirkel met diameter $2 \cdot 10^{14}$, met de opmerking eronder: "Godt alleen de eer". Hieronder een overzicht van de oplossingen gegeven door Van Ceulen en oplossingen berekend uit het tabel van Van Ceulen op pagina 79 van [1], met daarnaast de met Mathematica berekende oplossing. Dit tabel is overgenomen uit het onderzoek van Marloes Bazelier en Marloes Turk: Controle van Van Ceulens antwoorden [3, p. 4]. Daaronder een overzicht van waarden van de lengtes van de ingeschreven n -hoek in de cirkel van diameter 2 die van Ceulen doorgaans in hoofdstuk 14 geeft. Dit tabel is ook overgenomen uit [3, p. 3]. Tabel 2 is in chronologische volgorde van hoofdstuk 14. Tevens staat bij beiden tabellen of de benadering van Ludolph overeenkomt met de door mathematica berekende waarde, de foute decimalen zijn dik gedrukt.

Hoeken	L. van Ceulen	Mathematica	Juist	Hoeken	L. van Ceulen	Mathematica	Juist
3	1.73205080756887	1.732050807568877	✓	42	0.14946018717284	0.149460187172849	✓
4	1.41421356237309	1.414213562373095	✓	43	0.14599062932181	0.145990629321815	✓
5	1.17557050458494	1.175570504584946	✓	44	0.14267836639846	0.142678366398465	✓
6	1.00000000000000	1.000000000000000	✓	45	0.13951294748825	0.139512947488251	✓
7	0.86776747823511	0.867767478235116	✓	46	0.13648482672934	0.136484826729342	✓
8	0.76536686473017	0.7653668647301795	✓	47	0.13358526749024	0.133585267490243	✓
9	0.68404028665133	0.684040286651337	✓	48	0.13080625846028	0.130806258460286	✓
10	0.61803398874989	0.618033988749895	✓	49	0.12814043996142	0.128140439961426	✓
11	0.56346511368285	0.563465113682859	✓	50	0.12558103905862	0.125581039058627	✓
12	0.51763809020504	0.517638090205042	✓	51	0.12312181226788	0.123121812267886	✓
13	0.47863132857511	0.478631328575116	✓	52	0.12075699484457	0.120756994844572	✓
14	0.44504186791262	0.445041867912629	✓	53	0.11848125578742	0.118481255787429	✓
15	0.41582338163551	0.415823381635519	✓	54	0.11628965785095	0.116289657820952	✗
16	0.39018064403225	0.390180644032257	✓	55	0.11417762161551	0.114177621725536	✗
17	0.36749903563314	0.367499035633141	✓	56	0.11214089447438	0.112140894474384	✓
18	0.34729635533386	0.347296355333861	✓	57	0.11017552071173	0.110175520711731	✓
19	0.32918918056146	0.329189180561468	✓	58	0.10827781717083	0.108277817170835	✓
20	0.31286893008046	0.312868930080462	✓	59	0.10644434968435	0.106444349684357	✓
21	0.29808453235234	0.298084532352349	✓	60	0.10467191248588	0.104671912485888	✓
22	0.28462967654657	0.2846296765465703	✓	61	0.10295750954069	0.102957509540693	✓
23	0.27233329819249	0.272333298192493	✓	62	1.01298337677425	0.1012983376774254	✓
24	0.26105238444010	0.261052384440103	✓	63	0.99691771321395	0.0996917713213943	✗
25	0.25066646712860	0.250666467128608	✓	64	0.09813534865483	0.098135348654836	✓
26	0.24107336051064	0.241073360510646	✓	65	0.09662759051014	0.0966267590510141	✗
27	0.23218582850460	0.232185828250460	✗	66	0.09516383164748	0.095163831647484	✓
28	0.22392895220661	0.223928952206616	✓	67	0.09374452493988	0.093744524939880	✓
29	0.21623803684788	0.216238036847884	✓	68	0.09236691729147	0.092366917291479	✓
30	0.20905692653530	0.209056926535307	✓	69	0.09102919826592	0.091029198265927	✓
31	0.20233664397486	0.202336643974864	✓	70	0.08972966970063	0.089729660701029	✗
32	0.19603428065912	0.196034280659121	✓	71	0.08846669345075	0.088466693450757	✓
33	0.19011208660836	0.190112086608365	✓	72	0.08723877473067	0.087238774730672	✓
34	0.18453671892660	0.184536718926604	✓	73	0.08604446600906	0.086044466009060	✓
35	0.17927861780608	0.179278617806867	✗	74	0.08488240639229	0.084882406392296	✓
36	0.17431148549531	0.174311485495316	✓	75	0.08375130745839	0.083751307458399	✓
37	0.16961184895101	0.169611848951018	✓	76	0.08264994849762	0.082649948497626	✓
38	0.16515869094466	0.165158690944665	✓	77	0.08157717212317	0.081577172123171	✓
39	0.16093313743345	0.160933137433452	✓	78	0.08053188021883	0.080531880218830	✓
40	0.15691819145568	0.156918191455690	✗	79	0.07951303019385	0.079513030193851	✓
41	0.15309850567299	0.153098505672991	✓	80	0.07851963151813	0.078519631518137	✓

Tabel 1: Controle van Van Ceulens antwoorden uit het tabel achteraan hoofdstuk 14

Hoeken	L. van Ceulen	Mathematica	Juist
9	0.684040286651337466088199	0.6840402866513374660881992	✓
27	0.232185828250460459351333	0.2321858282504604593513330	✓
45	0.139512947488250601551917	0.1395129474882506015519177	✓
75	0.083751307458399259105670	0.0837513074583992591056707	✓
135	0.046537912748112644543202	0.0465379127481126445432026	✓
225	0.027924360678290543243822	0.0279243606782905432438228	✓
675	0.009308389071322324827845	0.0093083890713223248278454	✓
9	0.684040286651337466088199229364518	0.6840402866513374660881992293645192	✗
18	0.3472963553338606977034332535	0.34729635533386069770343325354	✓
14	0.445041867912628808577805	0.4450418679126288085778051	✓
7	0.867767478235116240951536	0.8677674782351162409515367	✓
22	0.28462967654657028088	0.284629676546570280888	✓
11	0.56346511368285939	0.563465113682859395	✓
26	0.241073360510646106698135	0.2410733605106461066981354	✓
13	0.4786313285751155	0.47863132857511553	✓
34	0.1845367189266039904794	0.18453671892660399047930	✗
17	0.3674990356331406631488	0.36749903563314066314882	✓
38	0.165158690944664649200687	0.1651586909446646492006879	✓
19	0.329189180561467	0.3291891805614678	✓
46	0.1364848267293419518423	0.13648482672934195184238	✓
23	0.27233329819249318152	0.272333298192493181521	✓
92	0.068282220371935790	0.06828222037193579057	✓
25	0.25066646712860849074623	0.250666467128608490746238	✓
27	0.2321858282504604593513330468	0.2321858282504604593513330467	✓
58	0.1082778171708350	0.10827781717083505	✓
29	0.216238036847883	0.2162380368478835	✓
62	0.10129833767742542	0.101298337677425425	✓
31	0.2023366439748643	0.20233664397486436	✓
74	0.08488240639229661	0.0848824063922966118	✓
37	0.1696118489510183	0.16961184895101838	✓
82	0.076605467380070	0.07660546738007070	✓
41	0.15309850567299	0.153098505672991	✓
86	0.0730440461153177	0.073044046115317670	✓
43	0.145990629321815	0.1459906293218151	✓
45	0.139512947488250601551917670388286657	0.1395129474882506015519176703882866572	✓
94	0.06682995401534914	0.0668299540153491418	✓
47	0.1335852674902431	0.13358526749024311	✓
49	0.12814043996142	0.128140439961426	✓
106	0.0592666556451194	0.059266655645119481	✓
118	0.053241042875548	0.05324104287554953	✗
59	0.10644434968435	0.106444349684357	✓
122	0.051495827309977	0.05149582730997711	✓
61	0.10295750954069	0.102957509540693	✓
134	0.0468851472065208	0.046885147206520887	✓
67	0.0937445249398801	0.093744524939880125	✓
142	0.04424417430637379	0.0442441743063737952	✓
71	0.088466693450757	0.08846669345075771	✓
146	0.043032194872444396	0.04303219487244439675	✓
73	0.0860444660090609	0.086044466009060949	✓
75	0.08375130745839925910567070314546536644	0.083751307458399259105670703115465366447	✗
135	0.04653791274811264454320260447491	0.0465379127481126445432026044749027	✗
158	0.0397643753301403386543	0.039764375330140338654358	✓

Tabel 2: Controle van Van Ceulens antwoorden doorgaans in hoofdstuk 14

2.1 Beschouwing van de fouten

We gaan kijken naar de fouten die gemaakt zijn door ofwel Van Ceulen ofwel de drukker. Het is belangrijk om in te zien dat wanneer een fout in het midden van de decimalen staat en de rest van de benadering wel goed is, dit waarschijnlijk een drukfout is. Als dit niet het geval zou zijn, dan had vanaf het foute decimaal gerekend met een foute waarde, en waren de volgende decimalen waarschijnlijk ook fout geweest. Met dit gegeven gaan we naar de fouten in tabel 1 en 2 kijken.

2.1.1 Tabel 1

Er staan in totaal acht fouten in tabel 1. Bij de 27-hoek mist de tiende decimaal, maar de resterende decimalen zijn correct. Dit doet denken dat de fout bij de 27-hoek een drukfout is. Zo hebben we de rest van de waardes ook gecontroleerd. Hieronder een overzicht van de fouten waarvan we verwachten dat ze drukfouten zijn. Deze incorrecte benaderingen hebben een fout in het midden van de benadering, maar zijn voor de resterende decimalen correct.

Hoeken	L. van Ceulen	Mathematica
27	0.23218582850460	0.232185828250460
54	0.11628965785095	0.116289657820952
65	0.09662759051014	0.0966267590510141
75	0.08375130745839925910567070314546536644	0.083751307458399259105670703115465366447

Tabel 3: Fouten in Hoofdstuk 14 van Vanden Cirkel die vermoedelijk drukfouten zijn

De fout bij de benadering van de 55-hoek lijkt erg op een rekenfout. Er zijn 3 decimalen fout. De negen, tiende en dertiende decimaal zijn fout. Omdat alle decimalen die fout zijn zo dicht op elkaar staan doet dit denken dat er een rekenfout gemaakt is bij de berekening van de zijde van de 55-hoek. Het laatste decimaal van de benadering van de 9-hoek is incorrect. Ludolph geeft dat deze gelijk is aan 8, echter had dit decimaal gelijk aan 9 moeten zijn. Een rekenfout is hier niet uit te sluiten. De fouten uit tabel 2 zijn voornamelijk fouten bij het laatste decimaal. Dit in tegenstelling tot enkele fouten uit tabel 1 waar het voorkomt dat wel 4 decimalen incorrect zijn. Hieronder een overzicht van de fouten waarvan we denken dat het geen drukfouten zijn, en dus rekenfouten gemaakt door Van Ceulen. Bij de foute benadering van de 70-hoek

Hoeken	L. van Ceulen	Mathematica
35	0.17927861780608	0.179278617806867
40	0.15691819145568	0.156918191455690
55	0.11417762161551	0.114177621725536
63	0.99691771321395	0.0996917713213943
70	0.08972966970063	0.089729660701029
9	0.684040286651337466088199229364518	0.6840402866513374660881992293645192
34	0.1845367189266039904794	0.18453671892660399047930
118	0.053241042875548	0.05324104287554953
135	0.04653791274811264454320260447491	0.0465379127481126445432026044749027

Tabel 4: Fouten in Hoofdstuk 14 van Vanden Cirkel die vermoedelijk rekenfouten zijn

gebeurt er ook nog iets opmerkelijks. Hier is het paar decimalen *61* steeds één te laag ten opzichte van het correcte paar decimalen *72*. Dit zou wellicht ook een drukfout kunnen zijn omdat de drukken het paar loodjes één te laag heeft gepakt, maar omdat het laatste decimaal ook fout is lijkt deze fout ook veel op een rekenfout.

2.2 Hoe verder

Nu we een idee hebben welke fouten van Ceulen heeft gemaakt kunnen we proberen de volgende vraag te beantwoorden: Hoe heeft van Ceulen deze benaderingen berekend? We zullen een overzicht geven van numerieke methoden die van Ceulen mogelijk gebruikt heeft. Hier geven we hoe de wortels van de polynomen met de methodes berekend kunnen worden, maar we zullen ook enkele zaken bewijzen wat

betreft de werking van de methoden. Hierbij willen we nagaan of de methoden wellicht ook de foute waardes reproduceren die voorkomen in tabel 1 en 2. Dan is het nog belangrijk om af te vragen of het onderzoeken hiervan mogelijk naar resultaten leidt. Heeft Van Ceulen wel een bekende methode gebruikt of is het een eigen methode die hij niet wil prijsgeven aan het publiek? Hierover zegt Steven Wepster het volgende: *"Nu is Van Ceulen er de man niet naar om de lezer moedwillig informatie te onthouden. Zelfs indien hij om commerciële redenen zijn methodes liever niet prijs gaf, dan zou hij hier nog wel nadrukkelijk de aandacht vestigen op de marktwaarde ervan. Er rest slechts een conclusie: blijkbaar vond hij het niet de moeite waard om er iets over te zeggen"* [10, p. 7]. We zijn dus op zoek naar een bekende numerieke methode uit de 16e eeuw.

3 Interpolatie methode van Chuquet

Nicolas Chuquet was een Franse wiskundige in de 15e eeuw. In het boek *Geschiede der Elementarmathematik* van Johannes Tropicke wordt het onderstaande algoritme genoemd voor het vinden van nulpunten van polynomen van willekeurige graad. Het onderstaande algoritme is vertaald uit [4, p. 505], met waar nodig verduidelijking toegevoegd. Het algoritme leggen we uit, en de aannames die we maken zullen we later bewijzen. Vervolgens wordt een voorbeeld uitgewerkt, we zullen dit algoritme toepassen op het polynoom voor de ingeschreven 18-hoek in de cirkel van diameter 2. Ten slotte zullen we kijken naar de convergentiesnelheid van de interpolatiemethode.

3.1 Het algoritme

Zij $f : \mathbb{R} \rightarrow \mathbb{R}$ een continue functie. We zijn op zoek naar een numerieke benadering van een nulpunt van f . Neem aan dat f een niet triviaal nulpunt heeft. Zij $\frac{a'}{b'} < \frac{a''}{b''}$ rationale getallen zodat f een nulpunt heeft tussen $\frac{a'}{b'}$ en $\frac{a''}{b''}$.

Omdat we op zoek zijn positief nulpunt van een opgesteld polynoom door Van Ceulen en de nulpunten van de polynomen tussen de 0 en 2 liggen, nemen we aan dat $0 \leq a'$ en $0 < a'', b', b''$. Bovendien nemen we aan dat $\gcd(a', b') = 1$, en $\gcd(a'', b'') = 1$.

Dan vinden we als benadering van een nulpunt van f het punt m_1 dat gegeven wordt door $m_1 := \frac{a' + a''}{b' + b''}$.

Bovendien voldoet m_1 aan de volgende ongelijkheden: $\frac{a'}{b'} < m_1 = \frac{a' + a''}{b' + b''} < \frac{a''}{b''}$.

Reken vervolgens $f(m_1)$ uit. Ligt het nulpunt van f tussen m_1 en $\frac{a''}{b''}$ dan maken we de rij m_i . De getallen m_i definiëren we door $m_i = \frac{a' + i \cdot a''}{b' + i \cdot b''}$ en m_1 zoals boven gegeven. Het nulpunt van f ligt in een van de intervallen $[m_i, m_{i+1}]$. Stop het algoritme als het nulpunt in een van deze intervallen ligt.

Ligt het nulpunt tussen $\frac{a'}{b'}$ en m_1 dan maken we de rij n_i . De getallen n_i definiëren we door $n_i = \frac{i \cdot a' + a''}{i \cdot b' + b''}$ en $n_1 = m_1$. Het nulpunt van f ligt in een van de intervallen $[n_{i+1}, n_i]$. Stop het algoritme als het nulpunt in een van deze intervallen ligt.

Merk op dat $\frac{a'}{b'} < \dots < n_2 < n_1$ en $m_1 < m_2 < \dots < \frac{a''}{b''}$. We vinden als benadering van een nulpunt van f de getallen x_0 zodat $x_0 \in [m_i, m_{i+1}]$ ofwel $x_0 \in [n_{i+1}, n_i]$. We noemen $[m_i, m_{i+1}]$ ofwel $[n_{i+1}, n_i]$ het resulterende interval na de interpolatie van Chuquet. Hiermee hebben we een benadering gevonden van het nulpunt van de functie f . Om vervolgens een betere benadering te vinden van het nulpunt van f kunnen we dit algoritme weer toepassen op de breuken m_i, m_{i+1} ofwel n_{i+1}, n_i . Het itereren over het resulterende interval zal een betere schatting van het nulpunt geven.

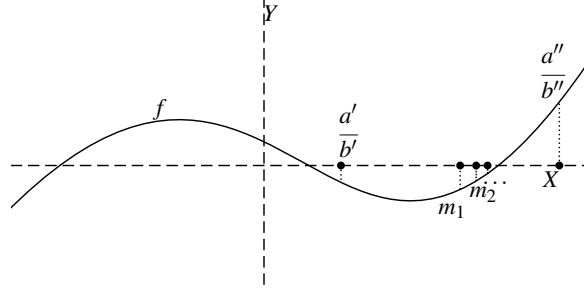
3.2 Werking algoritme

Eerst zullen we de correctheid van het algoritme bewijzen. We zullen ook alle gemaakte aannames bewijzen. Daarna zullen er nog enkele zaken bewezen worden over de iteratie. Hierbij is het belangrijkste resultaat dat er een nulpunt in het resulterende interval ligt.

Er geldt $\frac{a'}{b'} < m_1 = \frac{a' + a''}{b' + b''}$ want we hebben aangenomen dat $\frac{a'}{b'} < \frac{a''}{b''}$. Hieruit volgt dat $a'b'' < a''b'$ dus

$a'(b' + b'') < (a' + a'')(b')$ ook wel $\frac{a'}{b'} < \frac{a' + a''}{b' + b''}$. Uit $\frac{a'}{b'} < \frac{a''}{b''}$ volgt ook dat $(a' + a'')b'' < (b' + b'')a''$.

Hiermee hebben we bewezen dat $\frac{a'}{b'} < m_1 < \frac{a''}{b''}$.



Figuur 9: Voorbeeld van de interpolatie van Chuquet

Merk op dat er een nulpunt in een van de intervallen $\left[\frac{a'}{b'}, m_1\right]$ of $\left[m_1, \frac{a''}{b''}\right]$ ligt aangezien er een nulpunt in $\left[\frac{a'}{b'}, \frac{a''}{b''}\right]$ ligt. We willen bewijzen dat $m_1 < m_2 < \dots < \frac{a''}{b''}$. Hiervoor willen we bewijzen dat $m_i < \frac{a''}{b''}$ voor

alle natuurlijke getallen i , en dat $m_i < m_{i+1}$ voor alle natuurlijke getallen i . Eerst bewijzen we dat $m_i < \frac{a''}{b''}$. Merk weer op dat $a'b'' < a''b'$, daarom geldt ook dat $a'b'' + ia''b'' < a''b' + a''ib''$, en dus ddat

$$(a' + ia'')b'' < a''(b' + ib'')$$

Omdat $0 < (a' + ia'')b'' < a''(b' + ib'')$ vinden we dat wanneer we beiden kanten van de ongelijkheid delen door b'' en $(b' + ib'')$ $\frac{a' + ia''}{b' + ib''} < \frac{a''}{b''}$. Hiermee hebben we bewezen dat $m_i = \frac{a' + i \cdot a''}{b' + i \cdot b''} < \frac{a''}{b''}$.

Ten slotte bewijzen we dat $m_i < m_{i+1}$ voor alle natuurlijke getallen i . Zij i een natuurlijk getal. We hebben $a'b'' < a''b'$ omdat we aannemen dat $0 < \frac{a'}{b'} < \frac{a''}{b''}$. Om ons gewenste resultaat

$$\frac{a' + ia''}{b + ib''} < \frac{a' + (i+1)a''}{b' + (1+i)b''}$$

te verkrijgen, tellen we aan beide kanten van deze ongelijkheid de term

$$i(1+i)a''b'' + a'b' + ia'b'' + ia''b.$$

Hieruit volgt dat

$$a'b' + (1+i)a'b'' + ia'b'' + i(1+i)a''b'' < a'b' + ia'b'' + (1+i)a''b' + i(1+i)a''b''.$$

Als we nu aan beide kanten van de ongelijkheid de termen factoriseren volgt dat

$$(a' + ia'')(b' + (1+i)b'') < (b + ib'')(a' + (i+1)a'').$$

Omdat $0 < (a' + ia'')(b' + (1+i)b'') < (b + ib'')(a' + (i+1)a'')$, volgt wanneer we door $(b' + (1+i)b'')$ en $(b + ib'')$ delen dat $\frac{a' + ia''}{b + ib''} < \frac{a' + (i+1)a''}{b' + (1+i)b''}$. Hiermee hebben we laten zien dat

$$\frac{a' + ia''}{b + ib''} < \frac{a' + (i+1)a''}{b' + (1+i)b''}$$

Hiermee hebben we bewezen dat $m_i < m_{i+1}$. Omdat $m_i < m_{i+1}$ en $m_i < \frac{a''}{b''}$ voor alle natuurlijke getallen i , daarom kunnen we concluderen dat

$$m_1 < m_2 < \dots < \frac{a''}{b''}.$$

Op een analoge manier wordt bewezen dat $\frac{a'}{b'} < \dots < n_2 < n_1$. Dit wordt niet behandeld omdat we dan twee keer hetzelfde bewijs op schrijven. Ten slotte willen we nog bewijzen dat er een nulpunt van f in één van

de intervallen $[n_i, n_{i+1}]$ en $[m_i, m_{i+1}]$ ligt. Dit bewijzen we uit het ongerijmde. Stel namelijk dat er in geen van de intervallen zit een nulpunt van f ligt. We hebben aangenomen dat de functie f een nulpunt heeft in het interval $\left[\frac{a'}{b'}, \frac{a''}{b''}\right]$. Noem het reële getal x het nulpunt van f waarvan ze het bestaan hebben aangenomen, zodat $x \in \left[\frac{a'}{b'}, \frac{a''}{b''}\right]$. Stel dat x tussen m_1 en $\frac{a''}{b''}$ ligt. Omdat x in geen van de intervallen ligt volgt

$$x \in \left[m_1, \frac{a''}{b''}\right] \setminus \bigcup_{i \in \mathbb{N}} [m_i, m_{i+1}] = \left[m_1, \frac{a''}{b''}\right] \setminus \lim_{n \rightarrow \infty} [m_1, m_n]$$

Echter geldt dat $\lim_{n \rightarrow \infty} m_n = \lim_{n \rightarrow \infty} \frac{a' + n \cdot a''}{b' + n \cdot b''} = \frac{a''}{b''}$. Hieruit volgt dat $x \in \emptyset$, wat in tegenspraak is met het bestaan van het nulpunt x , dus x moet in een van de intervallen $[m_i, m_{i+1}]$ zitten.

Stel dat x tussen $\frac{a'}{b'}$ en m_1 ligt. Omdat x in geen van de intervallen ligt volgt

$$x \in \left[\frac{a'}{b'}, m_1\right] \setminus \bigcup_{i \in \mathbb{N}} [n_{i+1}, n_i] = \left[\frac{a'}{b'}, m_1\right] \setminus \lim_{j \rightarrow \infty} [n_j, m_1]$$

Echter geldt er dat $\lim_{j \rightarrow \infty} n_j = \lim_{j \rightarrow \infty} \frac{j \cdot a' + a''}{j \cdot b' + b''} = \frac{a'}{b'}$. Hieruit volgt dat $x \in \emptyset$, wat in tegenspraak is met het bestaan van het nulpunt x , dus x moet in een van de intervallen $[n_i, n_{i+1}]$ zitten. Omdat we hebben aangenomen dat $x \in \left[\frac{a'}{b'}, m_1\right] \cup \left[m_1, \frac{a''}{b''}\right]$, moet er gelden $x \in \left[m_1, \frac{a''}{b''}\right]$ ofwel $x \in \left[\frac{a'}{b'}, m_1\right]$, maar beide mogelijkheden leiden tot een tegenspraak, dus onze aanname is incorrect. Hieruit volgt dat x in een van de intervallen $[m_i, m_{i+1}]$, $[n_i, n_{i+1}]$ ligt.

3.2.1 Iteratie van de interpolatie

We hebben bewezen dat er een nulpunt van f in het interval $[m_i, m_{i+1}]$ ofwel $[n_i, n_{i+1}]$ ligt. Na één keer itereren is de precisie van ons antwoord misschien nog niet hoog genoeg. In het volgende gedeelte willen we een aantal zaken opmerken over de iteratie. Het belangrijkste resultaat dat we bewijzen is dat de interpolatie naar een nulpunt van de gevraagde functie convergeert. Om te kunnen praten over het itereren van de interpolatie willen we notatie invoeren. Aan de hand van deze notatie wordt bewijzen dat deze interpolatie naar een nulpunt convergeert.

Definitie 3.1. We noteren met $C([a, b])$ het resulterende interval na de interpolatie van Chuquet. Zij n een natuurlijk getal, noteer het resulterende interval na de n -de iteratie met $C^n([a, b])$. Om te kunnen praten over de linker en rechter kant van het interval noteren wij voor $C^n([a, b]) = [x_n, y_n]$. Dan noteren we de linker kant van het interval $C^n([a, b])_L := x_n$ en we noteren $C^n([a, b])_R := y_n$ voor de rechter kant van het interval.

Zonder verlies van algemeenheid kunnen we aannemen dat de breuken x_n en y_n aan de voorwaarden van het algoritme van Chuquet voldoen. De noemers van x_n en y_n blijven positief. Ze zijn namelijk een som van positieve gehele getallen. Bovendien kunnen we veronderstellen dat de breuken vereenvoudigd zijn. Daarom is $C^n([a, b])$ goed gedefinieerd. Als gevolg hiervan kunnen we de interpolatie van Chuquet zonder problemen itereren. Om ook te kunnen praten over "m₁" definiëren we het getal "m₁" voor een willekeurige iteratie.

Definitie 3.2. Definieer de functie $m: \mathbb{R} \supseteq \bar{I} \rightarrow \mathbb{R}$, van de begrensde gesloten intervallen deelverzamelingen van de reële getallen naar de reële getallen. Op de volgende manier: Zij p, q en n, m reële getallen, dan wordt het interval afgebeeld op de reële getallen door

$$\left[\frac{p}{q}, \frac{s}{t}\right] \mapsto \frac{p+s}{q+t}.$$

Merk op dat $m([a, b])$ precies overeenkomt met het eerder gedefinieerde getal m_1 . Nu we de functie m hebben gedefinieerd kunnen wij over het getal "m₁" spreken na een aantal iteraties. Zie dat $m(C^n([a, b]))$ overeen komt met het getal "m₁" na n iteraties. Nu deze notatie is ingevoerd, kunnen we gemakkelijk bewijzen dat de interpolatie naar een nulpunt convergeert. Dit wordt bewezen in het volgende lemma.

Lemma 3.1. Zij $f : \mathbb{R} \rightarrow \mathbb{R}$ een continue functie en $\frac{a'}{b'} < \frac{a''}{b''}$ rationale getallen zodat f een nulpunt heeft tussen $\frac{a'}{b'}$ en $\frac{a''}{b''}$. Met $\gcd(a', b') = 1$, en $\gcd(a'', b'') = 1$, en b', b'' beiden groter dan 0. Dan convergeert de rij van intervallen $\left(C^n \left(\left[\frac{a'}{b'}, \frac{a''}{b''} \right] \right) \right)_{n \in \mathbb{N}}$ naar een nulpunt van f . Ook wel

$$\lim_{n \rightarrow \infty} C^n \left(\left[\frac{a'}{b'}, \frac{a''}{b''} \right] \right) = \{y\} \quad \text{en} \quad f(y) = 0.$$

Bewijs. Voor ons gemak noteren we $C^n \left(\left[\frac{a'}{b'}, \frac{a''}{b''} \right] \right) = [x_n, y_n]$ zoals in **Definitie 3.2**. We bewijzen eerst dat voor alle natuurlijke getallen n geldt, dat $[x_{n+1}, y_{n+1}] \subsetneq [x_n, y_n]$. Per definitie weten we dat

$$[x_{n+1}, y_{n+1}] \subset [x_n, m([x_n, y_n])] \quad \text{ofwel} \quad \subset [m([x_n, y_n]), y_n].$$

We hebben bewezen dat $x_n < m(x_n, y_n) < y_n$. Hieruit volgt dat $[x_n, m([x_n, y_n])] \subsetneq [x_n, y_n]$ en ook dat $[m([x_n, y_n]), y_n] \subsetneq [x_n, y_n]$. Omdat $[x_{n+1}, y_{n+1}] \subset [m([x_n, y_n]), y_n]$ ofwel $[x_{n+1}, y_{n+1}] \subset [x_n, m([x_n, y_n])]$ vinden we ook dat $[x_{n+1}, y_{n+1}] \subsetneq [x_n, y_n]$, dus we hebben bewezen dat $[x_{n+1}, y_{n+1}] \subsetneq [x_n, y_n]$ voor alle natuurlijke getallen n . Hiermee kunnen we in ieder geval bewijzen dat de rij van intervallen $([x_n, y_n])_{n \in \mathbb{N}}$ naar een interval convergeert.

Noteer de diameter van het resulterende interval $[x_n, y_n]$ met $D_n := y_n - x_n$. Omdat een volgend resulterend interval strikt bevat zit in de vorige, hebben we dat $D_n > D_{n+1} \cdots \geq 0$. Hier hebben we een strikt dalende rij die van onder begrensd is. Hieruit volgt dat het limiet van deze rij naar de grootste ondergrens convergeert. We gaan nu bewijzen dat de rij D_n daadwerkelijk naar 0 convergeert. Merk ten eerste op dat $D_n > 0$. Omdat alle resulterende intervallen strikt bevat zijn in het vorige resulterende interval volgt dat er een reëel getal $0 < \rho_n < 1$ bestaat zodat $D_{p+1} = \rho_n D_n$. Dan vinden wij dat $D_n = \prod_{i=0}^n \rho_i D_0$. Omdat voor alle natuurlijke getallen i geldt dat $0 < \rho_i < 1$, volgt dat

$$\lim_{n \rightarrow \infty} D_n = \prod_{i=0}^n \rho_i D_0 \rightarrow 0.$$

Merk tevens op dat in elk resulterend interval een nulpunt van f zit. Omdat de diameter naar 0 convergeert en het interval niet leeg concluderen we dat:

$$\lim_{n \rightarrow \infty} C^n \left(\left[\frac{a'}{b'}, \frac{a''}{b''} \right] \right) = \{y\} \quad \text{en} \quad f(y) = 0.$$

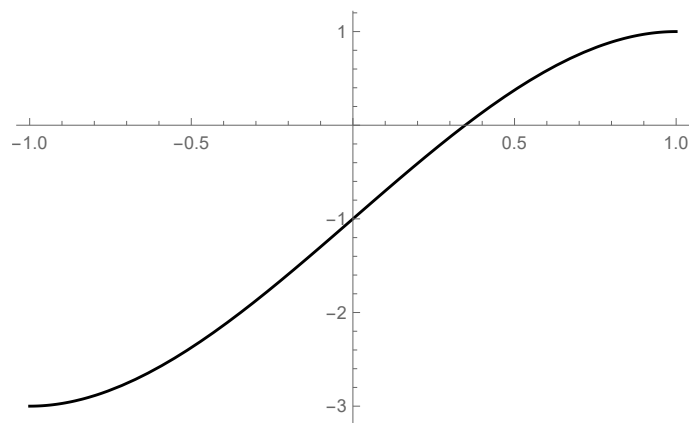
□

3.3 Toegepast op het polynoom van de ingeschreven 18-hoek in een cirkel van diameter 2

We willen graag zien hoe dit algoritme in de praktijk werkt. Daarom zullen we een voorbeeld uitwerken. In paragraaf 2 van hoofdstuk 1 hebben we gevonden dat we de lengte van een zijde van een ingeschreven 18 hoek in een cirkel van diameter 2 kunnen vinden door het kleinste positieve nulpunt van het polynoom

$$p(x) = 3x - x^3 - 1$$

te vinden. Op dit polynoom zullen we de interpolatie methode toepassen. Zij $c \in \mathbb{R}$ het kleinste nulpunt van p . Om het algoritme te laten werken moeten we een manier vinden om te controleren of punten links van c liggen ofwel rechts van c liggen. Dit doen we op de volgende manier. Stel dat we een interval $[a, b]$ hebben gevonden dat voldoet aan de eisen van het algoritme van Chuquet. Neem aan dat er maar één nulpunt van p in dit interval ligt. In figuur 10 zien we dat voor punten kleiner dan c in de omgeving van c geldt dat de



Figuur 10: $3x - x^3 - 1$ op het interval $[-1, 1]$

functiewaarde kleiner dan nul is wegens continuïteit. Voor punten groter dan c in de omgeving van c geldt dat de functiewaarde groter is dan nul. Met dit gegeven kunnen ons interval $[a, b]$ zo nodig aanpassen zodat voor alle $x \in [a, c]$ geldt dat $p(x) < 0$ en voor alle $y \in [c, b]$ geldt dat $p(y) > 0$. We zullen het algoritme dan ook stoppen als er een tekenverandering is. Nu hebben we een makkelijke manier om te controleren of een nulpunt in een van de intervallen ligt. Deze methode zullen we toepassen in ons voorbeeld.

Eerste iteratie

We passen het interpolatie algoritme van Chuquet toe op de beginwaarden $\frac{0}{1}$ en $\frac{1}{2}$. Dan volgt dat $m_1 = \frac{0+1}{1+2} = \frac{1}{3}$. Vervolgens rekenen we $p(\frac{1}{3})$ uit, dit is gelijk aan $3(\frac{1}{3}) - (\frac{1}{3})^3 - 1$.

We vinden dat $3(\frac{1}{3}) - (\frac{1}{3})^3 - 1 = -(\frac{1}{3})^3 < 0$. Omdat $p(\frac{1}{3}) < 0$ ligt het nulpunt van p tussen $\frac{1}{3}$ en $\frac{1}{2}$. Dan maken we de rij m_i tot dat $f(m_i) > 0$. In tabel 5 de bijbehorende waarden van m_i en het teken van $f(m_i)$.

Tabel 5: De rij m_i

i	m_i	$\text{sign}(f(m_i))$
1	$\frac{1}{3}$	-
2	$\frac{2}{5}$	+

Tweede iteratie

We passen het algoritme van Chuquet opnieuw toe, maar nu met beginwaarden $\frac{1}{3}$ en $\frac{2}{5}$. We vinden nu $m_1 = \frac{1+2}{3+5} = \frac{3}{8}$. Er geldt dat $p(\frac{3}{8}) = \frac{37}{512} > 0$ dus het nulpunt van p ligt tussen $\frac{1}{3}$ en $\frac{3}{8}$. Nu maken we de rij n_i tot $f(n_i) < 0$. Hiernaast in tabel 6 de waarden van n_i en het teken van $f(n_i)$. We concluderen dat er een nulpunt van f tussen $\frac{9}{26}$ en $\frac{8}{23}$ ligt.

Tabel 6: De rij n_i

i	n_i	$\text{sign}(f(n_i))$
1	$\frac{3}{8}$	+
2	$\frac{4}{11}$	+
3	$\frac{5}{14}$	+
4	$\frac{6}{17}$	+
5	$\frac{7}{20}$	+
6	$\frac{8}{23}$	+
7	$\frac{9}{26}$	-

Derde iteratie

Ten slotte passen we het algoritme nog een keer toe op $\frac{9}{26}$ en $\frac{8}{23}$. Nadat we dit opnieuw toepassen vinden we dat het nulpunt tussen $\frac{25}{72}$ en $\frac{33}{95}$ ligt. De decimale expansie van deze breuken zijn

$$\frac{25}{72} \approx 0.347222222222 \quad \text{en} \quad \frac{33}{95} \approx 0.3473684210526.$$

We hadden eerder gevonden dat het juiste nulpunt tot 14 decimalen gegeven wordt door 0.34729635533386. Na een iteratie van 10 stappen hebben we dus een precisie van 3 decimalen. Deze schatting is evident afhankelijk van onze beginwaarden, we kunnen ons afvragen hoeveel iteraties we moeten doen om een precisie van 14 decimalen te krijgen bij willekeurige beginschattingen. In het volgende gedeelte zullen we de convergentiesnelheid uittesten.

3.4 Interpolatie via Python

We hebben nu een algoritme gegeven en zijn juistheid nagegaan, en we hebben een voorbeeld ter verduidelijking uitgewerkt. De vraag is of van Ceulen mogelijk dit algoritme heeft gebruikt. Dit vraagstuk gaan we onderzoeken met een implementatie. Een resultaat kan bijvoorbeeld zijn dat er veel iteraties nodig zijn om de precisie van Van Ceulen te verkrijgen of dat er juist weinig iteraties nodig zijn.

De methode van Chuquet hebben we geïmplementeerd en geschreven in de programmeertaal Python. De input van de code is een functie en een interval zodat de functie een nulpunt in het interval heeft. Deze code zal ons het aantal iteraties teruggeven van de interpolatie. De code stopt als er een interval gevonden is zodat het verschil tussen de randpunten kleiner is dan 10^{-15} . Hier doen we ook de aanname dat er een manier is om te testen of punten kleiner dan het nulpunt zijn ofwel groter dan het nulpunt. In de uitgewerkte voorbeelden blijkt het mogelijk te zijn om dit te controleren door naar het teken te kijken, net zoals wij in het bovenstaande voorbeeld hebben gedaan.

Data: Gegeven zijn de breuken $\frac{a'}{b'}$, $\frac{a''}{b''}$ en een polynoom p , zodat er een nulpunt c van p ligt tussen $\frac{a'}{b'}$ en $\frac{a''}{b''}$.

Result: Aantal iteraties van de interpolatie van Chuquet

```

while  $\frac{a''}{b''} - \frac{a'}{b'} < 10^{-15}$  do
    Maak de waarde  $m_1 = m(\lfloor \frac{a'}{b'}, \frac{a''}{b''} \rfloor)$ 
    if  $c \in [\frac{a'}{b'}, m_1]$  then
        Maak de rij  $n_i$  tot  $n_{i+1} < c$ ;
        Sla het getal  $i + 1$  op;
        Definieer nu  $\frac{a'}{b'} := n_{i+1}$  en  $\frac{a''}{b''} := n_i$ 
    else
        Maak de rij  $m_i$  tot  $m_i > c$ ;
        Sla het getal  $i + 1$  op Definieer nu  $\frac{a'}{b'} := m_i$  en  $\frac{a''}{b''} := m_{i+1}$ 
    end
    Aantal iteraties = Aantal iteraties + ( $i + 1$ )
end
Print het aantal iteraties

```

We gaan dit algoritme testen met een aantal verschillende beginwaarden op twee verkregen polynomen. Eerst zullen we de code testen op het derdegraads polynoom van de zijde van de ingeschreven 18-hoek in een cirkel van diameter 2. Daarna zullen we het polynoom van de zijde van een ingeschreven 34 hoek in een cirkel van diameter 2 testen.

Het polynoom van de zijde van de ingeschreven 18-hoek in een cirkel van diameter 2 is gegeven door $p(x) = 3x - x^3 - 1$. Uit tabel 1 lezen we af dat het nulpunt is ongeveer gelijk aan 0.34. We zullen verschillende beginwaarden geven, en het resulterende aantal nodige iteraties. Voor de beginwaarden $\frac{0}{1}$ en $\frac{1}{2}$ volgt dat we 53 iteraties nodig hebben. Als we nu betere beginwaarden gebruiken, namelijk $\frac{1}{3}$ en $\frac{2}{3}$, hebben we 50 iteraties nodig.

Nu zullen we de interpolatie testen op het polynoom van de zijde van de ingeschreven 34-hoek. Deze is gegeven door

$$p(x) = x^{16} - 16x^{14} + 104x^{12} - 352x^{10} + 660x^8 - 672x^6 + 336x^4 - 64x^2 - x + 2. \quad (6)$$

Uit tabel 1 lezen we af dat het nulpunt ongeveer gelijk is aan 0.184. Een goede schatting voor het begininterval is bijvoorbeeld $[\frac{18}{100}, \frac{19}{100}]$. Met deze beginwaarden testen we de interpolatie met de code. Deze geeft terug dat er 45 iteraties nodig zijn voor een precisie van 14 decimalen. Een betere schatting wordt gegeven door het begininterval $[\frac{184}{1000}, \frac{185}{1000}]$. Wanneer we de code testen met deze beginwaarden krijgen we terug dat er 43 iteraties nodig zijn om een precisie van 14 decimalen te verkrijgen. Interessant is om op te merken dat de beginwaarden weinig invloed hebben op het aantal iteraties. In beide gevallen hebben we

een beginschatting van een orde 10 beter, maar het aantal iteraties neemt af met ongeveer $\sqrt{10}$.

Deze hoeveelheid iteraties is onrealistisch om uit te werken zonder computer. Hiervoor hebben we twee argumenten. Het belangrijkste argument hiervoor is: In het algoritme moet men bij elke stap dat het controleren dat $n_i < c$ ofwel $m_i > c$. De meest voordehandliggende manier om op te merken dat in dit geval $p(x) > 0$ voor $x \in (o - \delta, o)$ met δ een klein getal, en $p(y) < 0$ voor $y \in (o, o + \delta)$. Om te controleren dat $n_i < c$, testen we of $f(n_i) > 0$, en om te controleren dat $m_i > c$, testen we of $f(m_i) < 0$, maar om $f(n_i)$ ofwel $f(m_i)$ uit te rekenen moeten we hier een 16-de graads polynoom evalueren. Het uitrekenen van hoge machten kost veel tijd, zelfs met een enorm goede schatting van $[\frac{184}{1000}, \frac{185}{1000}]$ moet er 43 keer een 16-de graads polynoom geëvalueerd worden. Dit is een erg langdurige procedure. Hiervoor moeten we nog wel opmerken dat er methodes zijn om het evalueren van hogere-graads polynomen makkelijker te maken door het polynoom te herschrijven ofwel door schattingen te geven van de waarden met behulp van een analoog van de formule van Taylor die bekend was in 1600. Als het polynoom herschreven is moeten er nog steeds 60 breuken met elkaar vermenigvuldigd worden, wat niet prettig is. Ook zou het analoog van de formule van Taylor hier helaas niet van pas komen omdat we evaluaties exact willen hebben.

Echter, stel nu dat van Ceulen een slimigheid had om erachter te komen of n_i links van het nulpunt ligt of dat m_i rechts van het nulpunt ligt. Dan is het bovenstaande argument ongeldig, echter zou hij dan nog steeds de polynomen op elke stap moeten evalueren om te controleren of

$$\frac{a''}{b''} - \frac{a'}{b'} < 10^{-15} \quad (7)$$

Stel dat Ludolph begon met de enorm goede schatting van $[\frac{184}{1000}, \frac{185}{1000}]$. Dan moet hij 43 keer itereren, hierbij moet hij 27 keer controleren of (7) waar is. Om dit te controleren moet hij de verkregen breuken uitschrijven in decimale notatie of de breuken van elkaar aftrekken. Het uitrekenen van de decimale expansie van een breuk is een langdurig proces. Om te beargumenteren waarom dit een langdurig proces is geef ik het canonicke algoritme om van een breuk de decimale expansie te berekenen.

3.4.1 Van breuken naar decimalen

Zij $\frac{p}{q}$ een breuk, met p en q gehele getallen. Om het gemakkelijker voor ons te maken nemen we aan dat $0 < \frac{p}{q} \leq 2$. Dit kunnen we prima aannemen, omdat alle zijden van de ingeschreven n -hoeken in een cirkel van diameter 2 hieraan voldoen. We weten dat er een decimale expansie van $\frac{p}{q}$ bestaat in de vorm:

$$a_0, a_1 a_2 \dots = \sum_{i=0}^{\infty} a_i \cdot 10^{-i} = \frac{p}{q}$$

zodat $0 \leq a_i \leq 9$ voor alle natuurlijke getallen i . Dan is de vraag: Hoe vinden we a_i ?

We definiëren de getallen a_i inductief. Hiervoor definiëren we twee rijen, namelijk a_i en x_i . Eerst definiëren we a_0, x_0 en x_1 . We zeggen dat

$$a_0 = \lfloor \frac{p}{q} \rfloor, \quad x_0 = 0 \quad \text{en} \quad x_1 = \frac{p}{q} - a_0.$$

Vervolgens definiëren we x_{i+1} , zodat $x_{i+1} = x_i - a_i \cdot 10^{-i}$ en we definiëren $a_{i+1} = \lfloor 10^i \cdot x_i \rfloor$. Om het algoritme duidelijk te maken wil ik twee voorbeelden uitwerken. We rekenen de decimale expansie van de breuken $\frac{1}{4}$ en $\frac{8}{7}$ uit.

Eerst rekenen we de decimale expansie van $\frac{1}{4}$ uit. We vinden dat $0 < \frac{1}{4} < 1$ dus $a_0 = \lfloor \frac{1}{4} \rfloor = 0$. Dan vinden we dat

$$a_1 = \lfloor \frac{10}{4} \rfloor = \lfloor 2 + \frac{1}{2} \rfloor = 2.$$

Hiermee vinden we dat $x_1 = \frac{1}{4} - \frac{2}{10} = \frac{1}{20}$. Dan vinden we dat $a_2 = \lfloor \frac{100}{20} \rfloor = 5$. Hieruit volgt dat $x_2 = \frac{1}{20} - \frac{5}{100} = 0$. Dan vinden we dat $a_3 = \lfloor 0 \rfloor$ en dus $x_3 = 0 - 0 = 0$. Hieruit volgt dat $x_i = a_i$ voor alle $i \geq 3$. We concluderen dat de decimale expansie van $\frac{1}{4}$ wordt gegeven door $\frac{1}{4} = 0.2500\dots$

Ten slotte werken we de decimale expansie van $\frac{8}{7}$ uit. We vinden dat $a_0 = \lfloor \frac{8}{7} \rfloor = \lfloor 1 + \frac{1}{7} \rfloor = 1$, dus we vinden $x_1 = \frac{8}{7} - 1 = \frac{1}{7}$. Hieruit volgt dat $a_1 = \lfloor \frac{10}{7} \rfloor = 1$, dus $x_2 = \frac{1}{7} - \frac{1}{10} = \frac{3}{70}$. Dan volgt dat $a_2 = \lfloor \frac{300}{70} \rfloor = 4$, dus we vinden dat $x_3 = \frac{3}{70} - \frac{4}{100} = \frac{1}{350}$. Dit proces is recht toe recht aan en op deze manier vinden we dat $\frac{8}{7} = 1.1428571428571\dots$

Wat we hier mee duidelijk willen maken is dat het gaan van breuken naar decimalen ook geen kort proces is. Bovendien is dit proces ook nog eens inductief waardoor gemaakte fouten invloed kunnen hebben op de volgende schatting. Om volledig te zijn zouden we graag willen na gaan of dit algoritme dezelfde fouten geeft als deze van Van Ceulen, maar hier hebben we niet genoeg informatie voor. Als we na willen gaan of het algoritme dezelfde fouten geeft als van Ceulen. Dan zouden we na kunnen gaan: als we een incorrecte a_n geven en de resterende decimalen hiermee berekenen, komt dan de uitgerekende waarde overeen met Van Ceulens incorrecte antwoord. Bijvoorbeeld geeft Ludoplph de incorrecte schatting van de ingeschreven 55-hoek 0.114177621**61551**. Hier zijn de dikgedrukte decimalen incorrect. We zouden het bovenstaande algoritme kunnen laten werken op de incorrecte waarde 0.114177621**6** zodat deze de resterende decimalen uit rekt. Echter hebben we hiervoor ook de gevraagde breuk voor nodig. Omdat we niet weten welke beginschattingen Van Ceulen gebruikte kunnen alleen zeggen dat de door Van Ceulen gevonden breuk heel dicht bij het nulpunt ligt. Echter kunnen we wel concluderen dat dit proces minder lastig is dan de interpolatie van Chuquet. Het is een hoop rekenwerk maar in verhouding met het evalueren van polynomen van hoge graad is dit een gemakkelijke taak.

Het duurt ook lang om na te gaan of het resulterende interval een precisie heeft van 10^{-14} . Zij $[x_n, y_n]$ het resulterende interval. Om te controleren of het resulterende interval een precisie van 10^{-14} heeft willen we controleren of $y_n - x_n < 10^{-15}$. Dit kan een langdurig proces zijn. Bijvoorbeeld na 20 iteraties verkrijgen we bijvoorbeeld het interval $[\frac{302916518647773}{1641497260869000}, \frac{138119428633250}{748465830739000}]$. Als van Ceulen dit gedaan zou hebben door de breuken van elkaar af te trekken zou dit ook niet leuk geweest zijn. Je wilt namelijk niet met de hand controleren of de noemers van deze grote breuken gemeenschappelijke delers hebben, dus hij zou waarschijnlijk de tellers en noemers met elkaar moeten vermenigvuldigen zodat we de breuken van elkaar af kunnen trekken. Dit is ook geen prettig proces als dit meer dan 40 keer gecontroleerd moet worden.

Al met al zijn dit geen positieve resultaten voor het gebruik van de interpolatie van Chuquet. Het benaderen van nulpunten blijkt zeer lang te duren en is niet foutgevoelig. Bovendien lijkt het dat fouten hier alleen voorkomen door het omschrijven van breuken naar decimale notatie, echter is dit proces gemakkelijk in verhouding met het algoritme. Op dit moment vermoed ik dus niet dat dit algoritme gebruikt is.

4 De vastepunt iteratie van Chelebi

In 1498 gebruikte de Arabier Mariam Chelebi een iteratieve methode om nulpunten van functies te berekenen. De methode die hij gebruikte is een speciaal geval van de vastepunt iteratie. In het bijzonder gebruikte Chelebi deze iteratieve methode om de sinus van 1° te berekenen. De vastepunt iteratie wordt in andere vormen ook gebruikt door andere Arabische wiskundigen. Al-Habas al-Hasib gebruikte deze iteratieve methode bijvoorbeeld voor astronomische doeleinde in de 9e eeuw. Hieronder de vastepunt iteratie, vertaald uit [4, p. 506]. Dit is de moderne interpretatie van het algoritme. We zullen eerst kijken naar de algemene vastepunt iteratie, daarna zullen we kijken naar de vastepunt iteratie van Mariam Chelebi. Nadat we de vastepunt iteratie geïntroduceerd hebben, gaan we de correctheid van het algoritme bewijzen, dit bewijs neem ik over uit [7, p. 3]. Vervolgens passen we de vastepunt iteratie van Chelebi toe op drie polynomen van Van Ceulen.

4.1 Het algoritme

Zij $f : [a, b] \rightarrow \mathbb{R}$ een continue functie. Neem aan dat de functie f één nulpunt heeft in het interval $[a, b]$. Zij $g : [a, b] \rightarrow \mathbb{R}$ ook een continue functie met de eigenschap dat $f(x) = 0$ dan en slechts dan als $x = g(x)$.

Zij $x_0 \in [a, b]$ willekeurig. Dan convergeert de reeks $x_{i+1} = g(x_i)$ naar de oplossing $g(x) = x$ wanneer $|g'(x)| < 1$ voor alle getallen $x \in [a, b]$.

De kunst hier is om een functie g te vinden deze aan de bovenstaande voorwaarden voldoet. Er zijn veel keuzes voor g . De functie $g(x) = f(x) + x$ voldoet aan de voorwaarde $f(x) = 0$ dan en slechts dan $g(x) = 0$, maar we zouden ook $g(x) = x - 10 \cdot f(x)$ kunnen kiezen. Op bladzijde 290 van [8] wordt gezegd dat Mariam Chelebi de volgende keuze maakt voor g .

Zij $f : \mathbb{R} \rightarrow \mathbb{R}$ een polynoom. Schrijf $f(x) = \sum_{i=0}^n a_i \cdot x^i$, neem aan dat $a_1 \neq 0$. Dan kiezen we als functie g de functie

$$g(x) = \frac{a_1 x - \sum_{i=0}^n a_i \cdot x^i}{a_1} = \frac{a_1 x - f(x)}{a_1}.$$

Deze functie voldoet in ieder geval aan de eis $f(x) = 0$ dan en slechts dan als $g(x) = x$.

Dit bewijzen we kort. Neem aan dat $f(x) = 0$ dan zien we dat $g(x) = \frac{a_1 x - f(x)}{a_1} = \frac{a_1 x - 0}{a_1} = x$. Uit

$f(x) = 0$, volgt dat $g(x) = x$. Stel nu dat $g(x) = x$, dan volgt dat $x = \frac{a_1 x - f(x)}{a_1}$. Hieruit volgt dat $a_1 x = a_1 x - f(x)$. Uit $g(x) = x$ volgt dat $-f(x) = 0$, waaruit wij concluderen dat $f(x) = 0$. Hiermee is bewezen dat $f(x) = 0$ dan en slechts dan als $g(x) = x$.

De rest van de paragraaf zullen we kijken naar de vastepunt iteratie van Chelebi. Eerst zullen we bewijzen dat de vastepunt iteratie naar een nulpunt van de functie f convergeert.

4.2 Werking algoritme

We bewijzen de werking van de vastepunt iteratie met behulp van de middelwaardestelling. Het bewijs is een aangepast bewijs uit [7, p. 4] We willen bewijzen dat de rij $x_{k+1} = g(x_k)$ naar een punt x convergeert dat voldoet aan $f(x) = 0$ dan en slechts dan als $g(x) = x$. Neem aan dat $|g'(x)| < 1$ voor alle getallen $x \in [a, b]$ en zij k een natuurlijk getal. Omdat g een continue functie is volgt uit de middelwaarde stelling dat er een c tussen x_0 en x_1 bestaat zodat

$$g'(c)(x_{k+1} - x_k) = g(x_{k+1}) - g(x_k).$$

Omdat $x_{k+1} = g(x_k)$ en dus ook $x_{k+2} = g(x_{k+1})$ volgt dat

$$|g'(c)| |x_{k+1} - x_k| = |g(x_{k+1}) - g(x_k)| = |x_{k+2} - x_{k+1}|.$$

We hebben aangenomen dat $|g'(c)| < 1$ voor alle c tussen a en b , dus we vinden dat

$$|x_{k+1} - x_k| > |x_{k+2} - x_{k+1}|.$$

Hieruit volgt dat dat $|x_1 - x_0| > |x_2 - x_1| > |x_3 - x_2| > \dots \geq 0$. De rij $|x_{k+1} - x_k|$ is een van onder begrensde strikt dalende rij. Deze rij convergeert naar zijn grootste ondergrens. Nu we weten dat deze rij convergeert, zullen we bewijzen dat de rij naar 0 convergeert. Zij K het getal zodat $K = \sup_{x \in [a,b]} g'(x)$. Omdat voor alle $x \in [a, b]$ geldt dat $g'(x) < 1$, volgt dat $K < 1$. Hieruit volgt dat

$$|x_{k+1} - x_k| < |g'(c)| |x_k - x_{k-1}| < K \cdot |x_k - x_{k-1}| < \dots < K^k |x_1 - x_0|.$$

Omdat $K < 1$, volgt dat $|x_{k+1} - x_k| \rightarrow 0$ als $k \rightarrow \infty$. Uit de uitspraak $|x_{k+1} - x_k| \rightarrow 0$ volgt dat $|g(x_k) - x_k| \rightarrow 0$ dus $\lim_{n \rightarrow \infty} g(x_k) = \lim_{n \rightarrow \infty} x_k$. Omdat f een continue functie is, concluderen we dat $f(\lim_{n \rightarrow \infty} x_k) = \lim_{n \rightarrow \infty} f(x_k) = 0$. Hiermee hebben we bewezen dat de gegeven rij x_k naar een nulpunt van f convergeert.

4.3 Voorbeeld

We zullen een voorbeeld van de vastepunt iteratie uitwerken. We gaan de lengte van een zijde van de ingeschreven regelmatige 27-hoek in een cirkel van diameter twee berekenen. Deze lengte vinden wij door de kleinste positieve wortel van het volgende polynoom te vinden:

$$3x - x^3 = \text{lengte zijde 9-hoek.}$$

Noem de lengte van een zijde van de ingeschreven 9-hoek q , de waarde van q nemen we over uit tabel 2.

Dan zien we dat de functie $g(x)$ gegeven wordt door $g(x) = \frac{q + x^3}{3}$. We beginnen met de beginschatting

$x_0 = 0.2$. Dan volgt dat $x_1 = \frac{q - (0.2)^3}{3} = 0.2306800955504458$. Nu berekenen we x_2 door $x_2 = g(x_1)$ te berekenen, dit gaat als volgt

$$x_2 = g(x_1) = \frac{q + x_1^3}{3} = 0.23210517909180953.$$

We implementeren de vastepunt iteratie in Python. De implementatie in Python zal recht toe recht aan dit iteratieve proces uitvoeren. Dit proces is niet zo spannend, dus we verklappen dat $x_{10} = 0.232185828250460$. We hebben dus 10 iteraties nodig met de beginwaarde 0.2 om de waarde uit tabel 2 te verkrijgen. Deze waarde komt overeen met de correcte waarde van tabel 1. Hieronder een overzicht van de waarden van x_i voor i tussen 0 en 10.

x_i	Waarde x_i
x_0	0.2
x_1	0.2306800955504458
x_2	0.23210517909180953
x_3	0.23218148194347069
x_4	0.23218559394431168
x_5	0.2321858156189681
x_6	0.23218582756949346
x_7	0.23218582821374933
x_8	0.23218582824848133
x_9	0.23218582825035375
x_{10}	0.23218582825045467
x_{11}	0.2321858282504601

Tabel 7: Vastepunt iteratie toegepast op het polynoom van de ingeschreven regelmatige 27-hoek

In dit geval was het toevallig zo dat de vastepunt iteratie met deze keuze voor g convergeert naar het nulpunt. Ook nog eens heel snel, we hebben binnen 10 iteraties een precisie van 15 decimalen, in dit geval werkt deze methode heel goed! Dit blijkt echter niet altijd het geval. We zullen proberen met de vastepunt iteratie een wortel van (4) te vinden.

De vergelijking (4) is gegeven door:

$$x^{16} - 16x^{14} + 104x^{12} - 352x^{10} + 660x^8 - 672x^6 + 336x^4 - 64x^2 + 2 - x = 0.$$

Opnieuw gaan we de vastepunt iteratie van Chelebi toepassen. Hierbij maken we dezelfde keuze als Chelebi voor de functie $g(x)$. We kiezen namelijk $g(x) = \frac{a_1x - f(x)}{a_1} = x - f(x)$. Hieruit volgt dat de afgeleide van g gegeven wordt door $g'(x) = 1 - f'(x)$. We hebben gevonden dat een voorwaarde voor convergentie is dat $|g'(x)| < 1$ voor alle waarden $x \in [a, b]$. Als we $|g'(x)|$ evalueren in het nulpunt y van $f(x)$, dan zien we dat $|g'(y)| > 1$. Omdat er een waarde $y \in [a, b]$ bestaat zodat $g'(y) > 1$ kunnen we geen convergentie garanderen. Dit blijkt ook als we de vastepunt iteratie narekenen tot 1000 iteraties. Om een beeld te geven hoe deze er uit ziet, laten we de eerste 11 iteraties zien.

x_i	Waarde x_i
x_0	0.1
x_1	1.989442003723929
x_2	1.6987228393554688
x_3	-0.785387247800827
x_4	0.7621250587456414
x_5	-0.12619091796563753
x_6	1.291721238450667
x_7	-1.9791102315994067
x_8	-0.95037841796875
x_9	-1.041427014498589
x_{10}	-1.2939927959469957
x_{11}	-1.9608255429902783

Tabel 8: Vastepuntiteratie toegepast op het polynoom van de ingeschreven regelmatige 34-hoek

Het zelfs proces divergeert overduidelijk. Na 154 iteraties zien we dat de waarde van x_{154} gelijk is aan $-5.57 \cdot 10^{82}$. Bij de 155-ste iteratie is de waarde van x_{155} zelfs te groot voor het geheugen van Python.

Een ander voorbeeld van een polynoom dat Van Ceulen geeft waarvan wij geen convergentie kunnen garanderen is het polynoom voor de ingeschreven regelmatige 86-hoek in een cirkel van diameter 2. Van Ceulen geeft op bladzijde 76 van Vanden Cirkel dat hij deze zijde vindt met behulp van het kleinste nulpunt van het volgende polynoom:

$$\begin{aligned} & 2 + x - 484x^2 + 19481x^4 - 311696x^6 + 2643850x^8 - \\ & 13748020x^{10} + 47805615x^{12} - 117675360x^{14} + 213286590x^{16} - \\ & 292746300x^{18} + 310465155x^{20} - 258048960x^{22} + 169695240x^{24} - \\ & 88763664x^{26} + 36984860x^{28} - 12243264x^{30} + 3196578x^{32} - \\ & 649572x^{34} + 100529x^{36} - 11440x^{38} + 902x^{40} - 44x^{42} + x^{44} \end{aligned}$$

Omdat de coëfficiënt van de eerstegraads term gelijk is aan 1, vinden we opnieuw dat $g(x) = x - f(x)$. Hieruit volgt opnieuw dat $g'(x) = 1 - f'(x)$. Noem y het nulpunt van f , we nemen deze over uit tabel 2. Wanneer we g' evalueren zien we dat $|g'(y)| \approx 40 > 1$. We kunnen dus geen convergentie garanderen. Dit blijkt ook als we de vastepunt iteratie narekenen. Na 6 iteraties wordt de waarde van x_6 te groot voor het geheugen van Python. Daarom concluderen we dat de vastepunt iteratie niet werkt voor het polynoom van de ingeschreven 86-hoek.

Deze twee voorbeelden laten zien dat we de vastepunt iteratie van Chelebi niet altijd kunnen toepassen. Dit kan namelijk alleen onder gunstige omstandigheden. De voorbeelden die we hebben gekozen zijn polynomen van hoge graad, deze zijn in het algemeen lastiger op te lossen in verhouding met de lagere graads

polynomen. In het bijzonder zijn we geïnteresseerd in hoe Van Ceulen de hogeregraads polynomen met hoge precisie heeft opgelost. We kunnen in ieder geval uitsluiten dat Ludolph bij deze hogere graads polynomen de vastepunt iteratie van Chelebi heeft gebruikt.

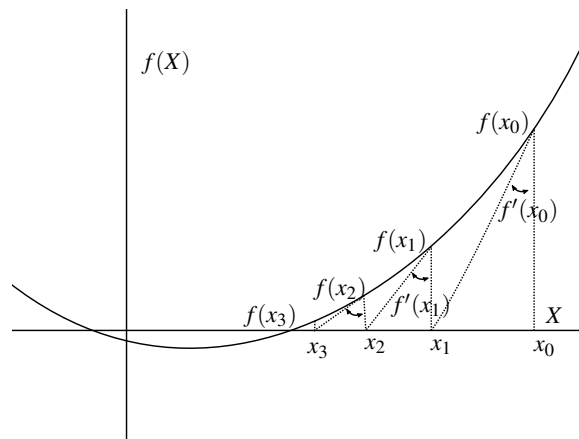
5 Vieta's iteratieve methode

De bekende Newton-Raphson methode is gebaseerd op Vieta's iteratieve methode. De Newton-Raphson methode is een iteratieve methode waarmee onder enkele voorwaarde een nulpunt van continue functies benaderd kan worden. Ik zal de Newton-Raphson methode kort introduceren.

Zij $f : [a, b] \rightarrow \mathbb{R}$ een continue functie en $x_0 \in [a, b]$ een schatting van een nulpunt van f . Een volgende schatting wordt iteratief gegeven door

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Onder enkele voorwaarden convergeert de rij x_n naar een nulpunt van de functie f . Hieronder een meetkundige interpretatie van de Newton-Raphson methode.



Figuur 11: Voorbeeld van de Newton-Raphson

Vieta's iteratieve methode verschijnt het eerst in 1646 in het boek Opera Mathematica van Franois Vieta. Opera Mathematica verschijnt 43 jaar na de dood van Vieta, en is een verzameling van zijn werk. Deze methode is net als de iteratieve methode van Al-Kashi een vastepunt iteratie. Alhoewel deze methode nu bekend staat als de Newton-Raphson methode zou het volgens de wiskundige John Wallis uit de 17e eeuw beter bekend moeten zijn als Vieta's iteratieve methode. Wallis zegt: "*de Newton-Raphson is in zekere zin een verbeterde versie van de procedure uitgelegd door Vieta en versimpeld door Oughtred...*". vertaald uit [5, p. 66]. Toch verschillen de methode van Vieta en Newton.

De nadruk van dit hoofdstuk ligt op Vieta's iteratieve methode. We zullen eerst dit algoritme introduceren, vervolgens willen we de correctheid van het algoritme bewijzen. Echter moeten we dit doen met behulp van de Newton-Raphson methode, omdat ik geen directe bewijzen in de literatuur kon vinden van Vieta's iteratieve methode. Na de introductie van het algoritme bewijzen we dus de correctheid van de Newton-Raphson methode. Nadat we dit hebben bewezen, kunnen we gemakkelijk de correctheid van Vieta's iteratieve methode bewijzen. Hierna zullen we inzoomen op de fouten die van Ceuelen maakt in tabel 1 en proberen om deze te reproduceren met behulp van Vieta's iteratieve methode. Hieronder de procedure van Vieta vertaald uit [5, p. 66].

5.1 Het algoritme

Zij $f : \mathbb{R} \rightarrow \mathbb{R}$ een polynoom en N een reel getal. We willen de vergelijking $f(x) = N$ oplossen. Stel dat x een oplossing is van de vergelijking $f(x) = N$. Omdat we hebben aangenomen dat x een reel getal is, kunnen we x op de volgende manier schrijven met behulp van zijn decimale expansie:

$$x = a_0 \cdot 10^k + a_1 \cdot 10^{k-1} + a_2 \cdot 10^{k-2} + \dots$$

met $0 \leq a_i \leq 9$ natuurlijke getallen, en $.$ Laet het getal x_1 een schatting van x zodat x_1 te schrijven is als

$$x_1 = a_0 \cdot 10^k + a_1 \cdot 10^{k-1} + a_2 \cdot 10^{k-2} + \dots + a_l \cdot 10^{k-l},$$

waar l een natuurlijk getal is. We zien x_1 als schatting van x waarbij de eerste l decimalen overeenkomen met deze van x . We vinden nu a_{l+1} met behulp van de volgende hulpwaarde:

$$g_k(x_l) = f(x_l + 10^{k-l-1}) - f(x_l) - 10^{(k-l-1)n},$$

waarbij n de graad van het polynoom f is. Dan is de waarde van a_{l+1} gelijk aan de absolute waarde van

$$\left\lfloor \frac{f(x_l) - N}{g_k(x_l)} \right\rfloor \pmod{10}. \quad (8)$$

Hiermee vinden we de nieuwe schatting x_2

$$x_2 = \sum_{i=1}^l a_i \cdot 10^{k-i} + a_{l+1} \cdot 10^{k-l-1} = x_1 + a_{l+1} \cdot 10^{k-l-1}.$$

We definiëren in het algemeen een volgende schatting met behulp van de vorige schatting op te volgende manier:

$$x_{n+1} = x_n + a_{l+n} \cdot 10^{k-l-n}.$$

5.2 Werking algoritme

Zoals we eerder hebben vermeld is de Newton-Raphson methode gebaseerd op Vieta's iteratieve methode. Alhoewel Vieta deze methode gebruikt in de Opera Mathematica, geeft hij geen bewijs over de werking hiervan. Daarom nemen we een omweg om de juistheid van deze methode te bewijzen, eerst zullen we kijken naar de Newton-Raphson iteratieve methode. We zullen bewijzen dat de Newton-Raphson methode onder enkele voorwaarden convergeert, nadat we dit hebben gedaan zullen we de methode aanpassen zodat deze overeenkomt met Vieta's iteratieve methode. Op deze manier lopen we terug door de geschiedenis. Het onderstaande bewijs van de Newton Raphson methode is een aangepast bewijs van [7, p. 5]. Het doel van dit bewijs is om te laten zien waarom Vieta's iteratieve methode benaderingen geeft voor de wortels van onze polynomen. We gebruiken hier moderne wiskunde, we maken bijvoorbeeld gebruik van afgeleiden van functies. Dit concept was nog niet bekend bij Vieta. Daarom moet dit gedeelte gezien worden als een moderne interpretatie van Vieta's iteratieve methode.

Zij $f : \mathbb{R} \rightarrow \mathbb{R}$ een polynoom en stel dat we de vergelijking $f(x) = N$ op willen lossen. Neem aan dat $f'(x) \neq 0$ en neem bovendien zonder verlies van algemeenheid aan dat $N = 0$. Als dit niet zo is kunnen we een nieuw polynoom $\tilde{f}(x)$ definiëren zodat $\tilde{f}(x) = f(x) - N$, hieruit volgt dat als $f(x) = N$ dan en slechts dan als $\tilde{f}(x) = 0$.

Zij x een nulpunt van f en x_0 een schatting van x . Definieer h als het reële getal zodat geldt dat $x = x_0 + h$. Dan vinden we met behulp van de formule van Taylor, dat

$$0 = f(x) = f(x_0 + h) = f(x_0) + hf'(x_0) + \mathcal{O}(h^2).$$

Omdat h klein is negeren we de hogere orde termen. Dan vinden we dat $hf'(x_0) \approx -f(x_0)$, dus

$$h \approx \frac{-f(x_0)}{f'(x_0)} := \hat{h}.$$

Dan vinden we een volgende schatting met behulp van $x_1 = x_0 + \hat{h}$. Mijn claim is dat de reeks

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

naar het nulpunt x convergeert. We zullen laten zien dat deze methode convergeert onder een aantal voorwaarden.

Bewijs. Om te bewijzen dat deze reeks naar een nulpunt convergeert willen we bewijzen dat $|x - x_k| \rightarrow 0$ als $k \rightarrow \infty$. Neem aan dat onze eerst benadering van het punt x op dezelfde manier gegeven is als in het

algoritme van Vieta. Dan geldt dat $x > x_k$ voor alle k . Definieer de fout van de benadering x_k als $e_k = x - x_k$. Wanneer we nu de tweede order Taylor uitvoeren rond x vinden we:

$$0 = f(x) = f(x_k + e_k) = f(x_k) + e_k f'(x_k) + \frac{e_k^2}{2} f''(\xi_k).$$

voor een bepaalde $\xi_k \in (x, x_k)$.

De eerste eis voor convergente is dat $f'(x_k) \neq 0$ voor alle k . Neem nu aan dat $f'(x_k) \neq 0$, we kunnen dan de bovenstaande uitdrukking door $f'(x_k)$ delen. Hieruit volgt dat

$$0 = \frac{f(x_k)}{f'(x_k)} + e_k + \frac{e_k^2}{2} \frac{f''(\xi_k)}{f'(x_k)}.$$

Omdat we x_{k+1} hebben gedefinieerd als $\frac{f(x_k)}{f'(x_k)}$, volgt dat

$$\frac{f(x_k)}{f'(x_k)} + e_k = x_k - \frac{f(x_k)}{f'(x_k)} - x = x_{k+1} - x = \frac{-e_k^2}{2} \frac{f''(\xi_k)}{f'(x_k)}.$$

Hieruit volgt dat $|x_{k+1} - x| = \left| \frac{e_k^2}{2} \frac{f''(\xi_k)}{f'(x_k)} \right| = |x_k - x| \left| \frac{e_k}{2} \frac{f''(\xi_k)}{f'(x_k)} \right|$.

Een tweede eis voor convergentie kunnen we als volgt formuleren:

$$\sup_{x \in [x - e_0, x + e_0]} \left| \frac{e_0}{2} \frac{f''(x)}{f'(x)} \right| < 1 \quad (9)$$

Neem aan dat aan voorwaarde (9) voldaan is. Met behulp van deze tweede voorwaarde kunnen we laten zien dat de reeks x_k naar de waarde x convergeert. We willen laten zien dat de rij $(|x_k - x|)_{k \in \mathbb{N}}$ een strikt dalende rij is met als grootste ondergrens 0. Eerst bewijzen we dat de rij een strikt dalende rij is, dit doen we met behulp van inductie.

Eerst bewijzen we de inductie basis, we laten zien dat $|x_1 - x| < |x_0 - x|$. Uit voorwaarde (9) volgt dat

$$|x_1 - x| = |x_0 - x| \cdot \left| \frac{e_0}{2} \frac{f''(\xi_k)}{f'(x_k)} \right| < |x_0 - x| \cdot \sup_{x \in [x - e_0, x + e_0]} \left| \frac{e_0}{2} \frac{f''(x)}{f'(x)} \right| < |x_0 - x|.$$

Dus we hebben laten zien dat $|x_1 - x| < |x_0 - x|$. Nu we de inductiebasis hebben, bewijzen we als $|x_k - x| < |x_i - x|$ voor alle $k < i$ dat dan ook

$$|x_{k+1} - x| < |x_k - x|.$$

Neem aan dat de uitspraak $|x_i - x| < |x_k - x|$ bewezen is voor alle $1 \leq i \leq k$. Dan willen we bewijzen dat

$$e_{k+1} = |x_{k+1} - x| < |x_k - x| = e_k.$$

We bewijzen eerst dat

$$\sup_{x \in [x - e_k, x + e_k]} \left| \frac{f''(x)}{f'(x)} \right| \leq \sup_{x \in [x - e_0, x + e_0]} \left| \frac{f''(x)}{f'(x)} \right|.$$

Merk ten eerste op dat $e_k < \infty$, dus het interval $[x - e_k, x + e_k]$ is compact. Omdat f een polynoom is, komt het supremum van $\frac{f''(x)}{f'(x)}$ overeen $[x - e_k, x + e_k]$ met het maximum. Hieruit volgt dat er een $y \in [x - e_k, x + e_k]$ bestaat zodat $\sup_{x \in [x - e_k, x + e_k]} \left| \frac{f''(x)}{f'(x)} \right| = \left| \frac{f''(y)}{f'(y)} \right|$. We hebben aangenomen dat $e_k < e_0$, hieruit volgt nu dat $y \in [x - e_0, x + e_0]$. Hieruit volgt vervolgens dat

$$\sup_{x \in [x - e_k, x + e_k]} \left| \frac{f''(x)}{f'(x)} \right| = \left| \frac{f''(y)}{f'(y)} \right| \leq \sup_{x \in [x - e_0, x + e_0]} \left| \frac{f''(x)}{f'(x)} \right|.$$

Met behulp van deze uitspraak kunnen we bewijzen dat $|x_{k+1} - x| < |x_k - x|$. We vinden dat

$$\begin{aligned} |x_{k+1} - x| &= |x_k - x| \cdot \left| \frac{e_k f''(\xi_k)}{2 f'(x_k)} \right| < |x_k - x| \cdot \sup_{x \in [x - e_k, x + e_k]} \left| \frac{e_0 f''(x)}{2 f'(x)} \right| \\ &< |x_k - x| \cdot \sup_{x \in [x - e_0, x + e_0]} \left| \frac{e_0 f''(x)}{2 f'(x)} \right| < |x_k - x|. \end{aligned}$$

We concluderen dat als $|x_k - x| < |x_i - x|$ voor alle $k < i$ dat dan ook $|x_{k+1} - x| < |x_k - x|$. Met behulp van inductie concluderen we nu dat de uitspraak $|x_{k+1} - x| < |x_k - x|$ waar is voor alle natuurlijke getallen k . Hiermee hebben we met inductie bewezen dat $|x_0 - x| > |x_1 - x| > \dots$ we vinden dat

$$|x_0 - x| > |x_1 - x| > \dots > 0.$$

Bovendien is deze rij van onderen begrensd door 0, dus de rij $|x_k - x|$ is een strikt dalende rij die van onderen begrensd is. Daarom convergeert deze rij naar zijn grootste ondergrens, dus er bestaat een $c \geq 0$ zodat $|x - x_k| \rightarrow c$ als $k \rightarrow \infty$. Hiermee hebben we bewezen dat, onder enkele voorwaarden, de Euler-Rahpson methode convergeert.

Hiermee hebben we nog niet bewezen dat $x_k \rightarrow x$ voor $k \rightarrow \infty$. We weten namelijk alleen nog maar dat er een $c \geq 0$ bestaat zodat $|x - x_k| \rightarrow c$ als $k \rightarrow \infty$. We zullen nu laten zien dat de grootste ondergrens gelijk is aan 0. Dit doen we als volgt. Noem

$$\sup_{x \in [x - e_0, x + e_0]} \left| \frac{e_0 f''(x)}{2 f'(x)} \right| := \rho.$$

Dan vinden we dat voor elk natuurlijk getal k geldt dat $|x_k - x| \leq \rho^k |x_0 - x|$. Omdat $\rho < 1$, vinden dat $|x_k - x| \rightarrow 0$, wanneer $k \rightarrow \infty$. Hiermee hebben we bewezen dat $|x_k - x| \rightarrow 0$ waaruit volgt dat onze rij naar een nulpunt van het gegeven polynoom f convergeert.

De cruciale voorwaarde hier is de voorwaarde (9), merk op dat deze van e_0 afhangt en dat deze uitdrukking voor kleinere e_0 kleiner wordt. We hebben e_0 gedefinieerd als $x - x_0$. Als we e_0 kleiner kiezen, dan vinden we dat $[x - e_0, x + e_0]$. Omdat f een polynoom is, en daarom continu is nemen f'' en f' maxima en minima op een interval $[x - x_0, x + x_0]$ aan, aan de hand hiervan zouden we e_0 kunnen proberen te kiezen zodat (9). \square

5.2.1 Van de Newton-Rahpson naar Vieta's methode

Nu we een idee hebben van de Newton-Rahpson methode gaan we deze aanpassen zodat hij overeenkomt met de iteratieve methode van Vieta. We zijn nog steeds op zoek naar een nulpunt van een polynoom f . We zijn nu in de setting die we hebben omschreven in paragraaf 1 van dit hoofdstuk. Zij x een nulpunt van f , gegeven door

$$x = a_0 \cdot 10^k + a_1 \cdot 10^{k-1} + a_2 \cdot 10^{k-2} + \dots$$

Bovendien hebben we ook een schatting van dit nulpunt, dit punt noemen we x_1 . De decimale expansie van het punt x_1 komt tot l cijfers overeen met het punt x_0 . Daarom geven we het punt x_1 op de volgende manier:

$$x_1 = a_0 \cdot 10^k + a_1 \cdot 10^{k-1} + a_2 \cdot 10^{k-2} + \dots + a_l \cdot 10^{k-l}.$$

We definiëren het getal h zodat $h = x - x_1$. Omdat de eerste l cijfers van de decimale expansie van x_1 overeenkomen met de cijfers van x zal h de resterende decimale expansie zijn van x . Dit kunnen we ook wel opschrijven als:

$$h = x - x_1 = a_{l+1} \cdot 10^{k-l-1} + a_{l+2} \cdot 10^{k-l-2} + \dots$$

Het doel is nu om de waarde van a_{l+1} te vinden. We hebben bij de Newton-Rahpson methode gevonden dat we de waarde van h kunnen schatten door $h \approx \frac{-f(x_1)}{f'(x_1)}$. Nu gaan we de waarde van $f'(x_1)$ benaderen. Eerder hebben we aangenomen dat $f'(x_1) \neq 0$. In tegenstelling tot de voorwaarde in het bovenstaande bewijs willen

we dat de benadering van $f'(x_1)$ ongelijk is aan 0. We vinden dat $f'(x_1) \approx \frac{f(x_1 + 10^{k-l-1}) - f(x_1)}{10^{k-l-1}}$. Vieta geeft de waarde:

$$f'(x_1) \approx \frac{f(x_1 + 10^{k-l-1}) - f(x_1) - 10^{(k-l-1)n}}{10^{k-l-1}}.$$

Waarbij n de graad van het polynoom f is. Waarom Vieta deze waarde geeft, is niet bekend voor mij. Vieta was niet op de hoogte van afgeleiden en zeker niet van Taylors formule. Hoe Vieta op deze methode is gekomen, is zeker een interessante vraag. Daarmee wil ik ook benadrukken dat deze afleiding een moderne interpretatie is van Vieta's iteratieve methode. Wanneer we de schatting invullen volgt

$$h \approx \frac{-f(x_1)}{\frac{f(x_1 + 10^{k-l-1}) - f(x_1) - 10^{(k-l-1)n}}{10^{k-l-1}}}.$$

We vinden dus dat $h \cdot 10^{k-l-1} \approx \frac{-f(x_1)}{f(x_1 + 10^{k-l-1}) - f(x_1) - 10^{(k-l-1)n}}$. Merk nu op dat

$$h = x - x_1 = a_{l+1} \cdot 10^{k-l-1} + a_{l+1} \cdot 10^{k-l-2} + \dots$$

Daarom vinden we de schatting

$$h \cdot 10^{-k+l+1} = a_{l+1} + a_{l+2} \cdot 10^{-1} + \dots \approx \frac{-f(x_1)}{f(x_1 + 10^{k-l-1}) - f(x_1) - 10^{(k-l-1)n}}.$$

Omdat $9 \geq a_{l+1} \geq 0$, en a_{l+1} een geheel getal vinden we dat

$$a_{l+1} = \left\lfloor \frac{-f(x_1)}{f(x_1 + 10^{k-l-1}) - f(x_1) - 10^{(k-l-1)n}} \right\rfloor \pmod{10}.$$

Hiermee definiëren we het iteratieve proces $x_{m+1} = x_k + a_{l+m} \cdot 10^{k-l-m}$. Dit iteratieve proces komt precies overeen met het iteratieve proces van Vieta. Merk op dat het grote verschil hier is dat we $f'(x_l)$ schatten en we hier hopen dat a_{l+1} hier niet door beïnvloed wordt. Dit maakt het proces enigszins ambigu, omdat we tweemaal een schatting doen, met de Taylor en met de afgeleide.

Ten slotte zetten we de Newton-Raphson methode en Vieta's iteratieve methode nog naast elkaar om te laten zien wat de verschillen zijn. Hiervoor gaan we ook nog terug naar de vastepunt iteratie. Zij $f: \mathbb{R} \rightarrow \mathbb{R}$ het polynoom waarvan we een wortel willen berekenen. De Newton-Raphson methode is een vastepunt iteratie met functie $g(x) = x - \frac{f(x)}{f'(x)}$. We hebben een volgende schatting gedefinieerd door $x_{k+1} = g(x_k)$. Vieta's iteratieve methode is in zekere zin een aangepaste versie van de Newton-Raphson methode dit geldt ook andersom. Het verschil zit vooral in hoe het iteratieve proces gedefinieerd is. Bij Vieta's iteratieve methode verkrijgen bij elke stap van de iteratie één nieuwe decimaal. Dit verkrijgen we door de nieuwe schatting te geven door $x_{m+1} = x_k + a_{l+m} \cdot 10^{k-l-m}$. Dus de Newton-Raphson methode is een speciale vaste punt iteratie, en Vieta's iteratieve methode is een Newton-Raphson iteratie waarbij we elke iteratie stap één decimaal verkrijgen.

In het volgende deel van dit hoofdstuk zullen we het algoritme laten werken op de polynomen van Van Ceulen. We gaan een voorbeeld uitwerken, en we zullen dit proces programmeren en daarmee proberen de fouten die in Vanden Cirkel voorkomen te reproduceren.

5.3 Voorbeeld van Vieta's iteratieve methode op de ingeschreven regelmatige 18-hoek

Vieta's iteratieve methode wordt duidelijker als we een voorbeeld uitwerken. We passen de iteratieve methode van Vieta toe op (2). Hier hebben we het polynoom $f(x) = 3x - x^3$, we willen de volgende vergelijking oplossen: $f(x) = 1$. Ten eerste willen we een schatting vinden van de wortel van $3x - x^3 = 1$, de schatting wordt gegeven door $x = 3 \cdot 10^{-1}$. Dus we hebben $k = -1$ en $a_0 = 3$. We bepalen nu de hulpwaarde $g_{-1}(x_0)$.

$$g_{-1}(x_0) = f(0.3 + 10^{-2}) - f(0.3) - 10^{(-2) \cdot 3} = f(0.3 + 10^{-2}) - f(0.3) - 10^{-6}$$

Dan vinden we dat a_1 gegeven wordt door

$$a_1 = \left\lfloor \left\lfloor \frac{f(0.3) - 1}{f(0.3 + 10^{-2}) - f(0.3) - 10^{-6}} \right\rfloor \right\rfloor \bmod 10 = 4$$

Hieruit volgt dat de nieuwe schatting gegeven wordt door $x_1 = 3 \cdot 10^{-1} + 4 \cdot 10^{-2}$. Hiermee rekenen we a_2 uit:

$$a_2 = \left\lfloor \left\lfloor \frac{f(0.34) - 1}{f(0.34 + 10^{-3}) - f(0.34) - 10^{-8}} \right\rfloor \right\rfloor \bmod 10 = 7$$

De volgende schatting x_2 is dus gelijk aan $3 \cdot 10^{-1} + 4 \cdot 10^{-2} + 7 \cdot 10^{-3} = 0.347$. Dit proces is recht toe recht aan, we rekenen a_i uit, en hiermee rekenen we a_{i+1} uit. Zoals in tabel 2 willen we graag de wortel op 14 decimalen precisie geven. Deze vinden we door hiervoor een programma te schrijven voor het algoritme van Vieta. Het programma heeft als input een polynoom, de wortel die we willen benaderen, een schatting van de wortel in de bovenstaande vorm en de gevraagde precisie. De output van het programma is de schatting met behulp van de iteratieve methode van Vieta op de gevraagde precisie. Het programma geeft in ons geval terug dat $x_{14} = 0.34729635533386$. Deze waarde komt overeen met de door Mathematica berekende waarde. Dus we kunnen concluderen dat deze methode in dit geval goede resultaten geeft. Ter controle willen we ook graag deze methode uittesten voor enkele hogere graads polynomen.

We passen Vieta's algoritme eerst toe op het 36-graads polynoom (4), het verkregen polynoom van de ingeschreven regelmatige 34-hoek. Vieta's algoritme geeft bij de beginschatting 0.1 de door Mathematica berekende waarde uit tabel 1 terug.

Ten slotte passen we ter controle Vieta's iteratieve methode toe op het polynoom (8) van graad 44. Het kleinste nulpunt van dit polynoom geeft de zijde van de ingeschreven regelmatige 86-hoek in een cirkel van diameter twee. Wanneer we het algoritme van Vieta laten werken op de beginschatting 0.07 vinden we dat het algoritme dezelfde waarde teruggeeft als Mathematica in tabel 2.

Kan dit algoritme ook de fouten reproduceren die we tegenkomen in tabel 2? Deze vraag zullen we in het volgende gedeelte beantwoorden.

5.4 Fouten reproduceren uit tabel 2.1.1

In paragraaf 2.1 hebben we de fouten die Van Ceulen mogelijk gemaakt heeft beschouwd. We zullen narekenen of de mogelijke fouten van Van Ceulen ook door het algoritme van Vieta veroorzaakt kunnen worden. We zullen de fouten op volgorde van kleinste regelmatige veelhoek, naar grootste regelmatige veelhoek afgaan. Bovendien gebruiken we het programma uit het bovenstaande voorbeeld om de schattingen te geven.

5.4.1 De ingeschreven regelmatige 27-hoek in cirkel van diameter 2

In tabel 2 is er een fout bij de benadering van de 27-hoek. Van Ceulen geeft 0.23218582850460 als lengte voor de lengte van een zijde van een ingeschreven regelmatige 27 hoek in een cirkel van diameter 2. Op bladzijde 75 van [1] schrijft Van Ceulen dat hij deze lengte uitrekent met behulp van de eerder berekende lengte van de ingeschreven regelmatige 9 hoek. Dit gaat met behulp van stelling 1.1, die zegt dat de lengte van de 27-hoek gegeven wordt door x , waarbij x het kleinste nulpunt is van $3x - x^3 = \text{lengte zijde 9-hoek}$. De lengte van de ingeschreven regelmatige 9-hoek wordt correct met een enorme precisie van 33 decimalen gegeven. Ter controle kunnen we het iteratieve proces van Vieta op dit polynoom laten werken met beginschatting 0.2. Dit geeft de goede waarde van mathematica uit tabel 2. Nu geven we de foutieve beginschatting 0.2321858285, en we laten Vieta's iteratieve proces de overige decimalen laat. Het algoritme geeft de waarde 0.23218582854949 terug. Opnieuw ter controle, als we wel de goede schatting 0.2321858282 geven en de code hiermee de resterende decimalen laten uitrekenen geeft deze de correcte waarde 0.232185828250460 terug.

5.4.2 De ingeschreven regelmatige 35-hoek in cirkel van diameter 2

Bij de benadering van de ingeschreven regelmatige 35-hoek in een cirkel van diameter 2 zijn de laatste twee decimalen die van Ceulen geeft incorrect. Van Ceulen geeft 0.17927861780608, terwijl deze waarde 0.179278617806867 moet zijn. We willen controleren of deze methode deze fout ook geeft. Op bladzijde 76 van [1] zegt van Ceulen dat hij de waarde uitrekent met behulp van de 7 hoek en 5 hoek. Dit doet ons denken dat hij deze lengte uitrekent met een van de volgende formules. Zij x de lengte van de ingeschreven regelmatige 35-hoek in een cirkel van diameter 2, dan vinden we dat x gelijk is aan het kleinste nulpunt van een van de volgende polynomen

$$5x - 5x^3 + x^5 = \text{lengte zijde 7-hoek} \quad (10)$$

$$7x - 14x^3 + 7x^5 - x^7 = \text{lengte zijde 5-hoek} \quad (11)$$

We kijken eerst naar het polynoom (10). De lengte van de ingeschreven regelmatige 7-hoek in een cirkel van diameter 2 wordt correct gegeven met precisie van 24 decimalen in [1] op bladzijde 73. Ter controle geeft het algoritme van Vieta laten werken op de beginschatting 0.1, dan geeft deze dezelfde waarde als Mathematica in tabel 2 terug. Evenals als we Vieta verder laten rekenen met de goede beginschatting 0.1792786178068. Al we nu mathematica verder laten werken met de foutieve beginschatting 0.1792786178060, dan geeft deze de waarde 0.17927861780606 terug. Deze komt niet overeen met de foutieve benadering van Van Ceulen.

Ten slotte kijken we naar (11). We berekenen het kleinste nulpunt met behulp van de lengte van de ingeschreven regelmatige 5-hoek. Deze lengte geeft van Ceulen exact op bladzijde 144 van [1]. Hij geeft deze lengte als $\sqrt{2\frac{1}{2}} + \sqrt{1\frac{1}{4}}$. Ter controle passen we Vieta's algoritme toe op beginwaarde 0.1, dan geeft deze de correcte waarde uit tabel 2 terug, evenals met de correcte beginschatting 0.1792786178068. Maar wanneer we de resterende decimalen uitrekenen met de foutieve beginschatting 0.1792786178060 geeft het algoritme opnieuw 0.17927861780606 terug. Deze waarde komt niet overeen met de waarde die van Ceulen geeft in tabel 2.

5.4.3 De ingeschreven regelmatige 40-hoek in cirkel van diameter 2

Het laatst gegeven decimaal in de benadering van de 40-hoek is één te laag, dit kan wellicht een rekenfout zijn. Helaas specificeert van Ceulen in Vanden Cirkel niet hoe hij de lengte van de ingeschreven regelmatige 40-hoek berekent. De meest voordehandliggende manier is door op te merken dat de lengte van een zijde van de ingeschreven regelmatige 40-hoek in een cirkel van diameter 2 voldoet aan $5x - 5x^3 + x^5 = \text{lengte zijde 8-hoek}$. Bovendien wordt de lengte van de ingeschreven regelmatige 40-hoek gegeven door het kleinste nulpunt van dit polynoom. De lengte van de ingeschreven regelmatige 8-hoek wordt alleen in tabel 1 gegeven door 0.76536686473017. Wanneer we op de beginschatting 0.1 Vieta's iteratieve algoritme laten werken, vinden we dat deze de waarde 0.156918191455687 terug geeft. De door Van Ceulen gegeven waarde is 0.15691819145568 en de door Mathematica correcte waarde is 0.156918191455690. Deze waarde komt overeen met de door Van Ceulen gegeven waarde!

5.4.4 De ingeschreven regelmatige 55-hoek in cirkel van diameter 2

Bij de benadering van de lengte van de ingeschreven regelmatige 55-hoek geeft tabel 2 3 foute decimalen in de door van Ceulen gegeven waarde. Omdat het zo veel fouten zijn, lijkt het dat deze komen door rekenfouten. Daarom gaan we deze fout bestuderen. Van Ceulen geeft de waarde 0.11417762161551 terwijl deze waarde 0.114177621725536 had moeten zijn. Op bladzijde 76 van [1] vermeldt van Ceulen dat hij de lengte van de ingeschreven regelmatige 55-hoek berekent met behulp van de ingeschreven regelmatige 10 en 11-hoek. We gaan er hier vanuit dat Van Ceulen de volgende vergelijking heeft gebruikt om de lengte van de ingeschreven regelmatige 55-hoek, in een cirkel van diameter 2 te vinden. Zij x de lengte van de ingeschreven regelmatige 55-hoek dan geldt

$$5x - 5x^3 + x^5 = \text{lengte zijde 11-hoek.}$$

Bovendien wordt de lengte x gegeven door de kleinste positieve oplossing van dit polynoom. Op bladzijde 73 van [1] wordt de lengte van de zijde van een ingeschreven regelmatige 11-hoek gegeven door

0.56346511368285939, deze waarde is correct. We laten het algoritme van Vieta op deze waardes werken. Met beginschatting 0.1 geeft het algoritme dezelfde waarde als in tabel 2. We zullen nu narekenen of de fouten dezelfde resultaten als tabel 2 geven. Hier gaan we alle mogelijke permutaties na. Het zou bijvoorbeeld kunnen dat de eerste fout in de benadering, de hier dikgedrukte 6, een drukfout is. Terwijl de volgende foute decimaal in de benadering, de dikgedrukte 1, een rekenfout is. Hieronder een de beginwaarden met daarnaast de schatting uitgerekend door het algoritme van Vieta.

Beginschatting	Schatting door iteratieve methode van Vieta
0.1141776216	0.11417762162559
0.11417762161	0.11417762161559
0.11417762171	0.11417762171553

Tabel 9: Resultaten van Vieta's iteratieve algoritme met foute beginschattingen bij de ingeschreven regelmatige 55-hoek

Wat hier erg interessant aan is, is dat Vieta's iteratieve methode met de foute beginschattingen de twee dertiende en de veertiende decimaal, de twee vijven achter elkaar in alle gevallen schuingedrukt, overeenkomen met het juiste antwoord van mathematica en de foute benadering van van Ceulen.

5.4.5 De ingeschreven regelmatige 63-hoek in cirkel van diameter 2

De foute waarde bij de 63 hoek in tabel 2 is mogelijk ook een rekenfout. We zullen nagaan of Vieta's methode deze ook geeft. Hier is het laatste decimaal incorrect. Van Ceulen geeft 0.99691771321395 terwijl deze waarde 0.0996917713213943 had moeten zijn. Op bladzijde 77 van [1] geeft Van Ceulen aan dat hij deze waarde heeft gevonden met behulp van de 7 en 9-hoek. Hij doelt hier op de vergelijking: Zij x de lengte van de ingeschreven regelmatige 63 hoek, dan geldt dat

$$7x - 14x^3 + 7x^5 - x^7 = \text{lengte zijde 9-hoek.}$$

Bovendien wordt x weer gegeven door de kleinste positieve oplossing van deze polynoom. De lengte van de zijde van de ingeschreven regelmatige 9-hoek in een cirkel van diameter 2 hebben we bij de berekening van de 27-hoek gebruikt, en zullen we hier opnieuw gebruiken. Wanneer we de beginschatting 0.09 geven geeft Vieta's iteratieve methode dezelfde waarde als mathematica in tabel 2 terug.

5.4.6 De ingeschreven regelmatige 70-hoek in cirkel van diameter 2

Ten slotte gaan we kijken naar de fout die wordt gemaakt bij de ingeschreven regelmatige 70-hoek. Van Ceulen geeft de benadering 0.08972966970063 terwijl deze waarde 0.089729660701029 had moeten zijn. In hoofdstuk 14 van Vanden Cirkel wordt niet vermeld hoe de waarde van de ingeschreven regelmatige 70-hoek wordt berekend. Toch zijn er twee voordehandliggende manieren waarop hij deze waarden berekend zou kunnen hebben. Dit zijn de volgende manieren: Zij x de lengte van de 70-hoek in cirkel van diameter 2. Dan voldoet x aan:

$$5x - 5x^3 + x^5 = \text{lengte zijde 14-hoek} \quad (12)$$

$$7x - 14x^3 + 7x^5 - x^7 = \text{lengte zijde 10-hoek} \quad (13)$$

Bovendien is de waarde van x de kleinste waarde die voldoet aan (12) ofwel (13). De waarde van de ingeschreven regelmatige 14-hoek wordt op bladzijde 73 van [1] gegeven door 0.445041867912628808577805, en de waarde van de regelmatige ingeschreven 10-hoek wordt gegeven door 0.61803398874989484 op bladzijde 82 van [1]. Laten we beginnen met (12). Wanneer we Vieta's iteratieve methode toepassen op de beginschatting 0.08, geeft deze de met mathematica berekende waarde uit tabel 2 terug. We zullen nu narekenen of de fouten dezelfde resultaten als tabel 2 geven. Hier gaan we opnieuw alle mogelijke permutaties na. Hieronder de resultaten samengevat in een tabel.

Beginschatting	Schatting door iteratieve methode van Vieta
0.089729669	0.089729669 298286
0.089729660700	0.089729660700 285
0.0897296607006	0.089729660700 629
0.089729669700	0.08972966 9700199
0.0897296697006	0.08972966 9700699

Tabel 10: Resultaten van Vieta's iteratieve algoritme met foute beginschattingen bij de ingeschreven regelmatige 70-hoek

Alle waardes komen niet overeen met de waardes die Van Ceulen geeft.

Het zou ook kunnen dat methode (13) gebruikt is. De beginschatting 0.08 geeft opnieuw de juiste benadering. We zullen alle mogelijke fouten weer narekenen. Hieronder de resultaten samengevat in een tabel. Opnieuw lijken de foute beginschattingen niet op de antwoorden van Van Ceulen.

Beginschatting	Schatting door iteratieve methode van Vieta
0.089729669	0.089729669 29823
0.089729660700	0.089729660700 000
0.0897296607006	0.089729660700 629
0,089729669700	0,08972966 970090
0.0897296697006	0.08972966 9700629

Tabel 11: Resultaten van Vieta's iteratieve algoritme met foute beginschattingen bij de ingeschreven regelmatige 70-hoek

5.4.7 Conclusie

Nu we alle beginschattingen uit tabel 2.1.1 hebben nagerekend met behulp van Vieta's iteratieve methode, wil ik graag terugkijken naar de verkregen resultaten. In tabel 2.1.1 hebben we een overzicht geplaatst van door van Ceulen gemaakte fouten die het meeste op rekenfouten lijken. Van deze fouten kwamen er 5 uit tabel 1. Drie van deze fouten hebben we niet kunnen reproduceren. Echter kunnen we wel één fout reproduceren en één fout deels. Respectievelijk de 40-hoek en de 55-hoek. Nu kunnen we ons afvragen of dit toeval is, of dat de oorzaak hiervan is dat Ludoph ook Vieta's iteratieve methode heeft gebruikt. Om dit te beargumenteren wil kijken naar de fouten die we hebben gereproduceerd. De enige fout die we reproduceert hebben, is de fout bij de 40-hoek. Echter hebben we de andere fouten op de 55-hoek na niet kunnen reproduceren. Bij de 55-hoek konden we de twee 55 steeds reproduceren. Ik verwacht dat dit meer te maken heeft de stabiliteit van de iteratieve methode van Vieta, dan dat Van Ceulen deze methode gebruikt heeft. Dat we de fout bij de 40-hoek precies konden reproduceren is wel een aanwijzing naar het gebruik van deze iteratieve methode. Het is wel heel toevallig dat alleen het laatste decimaal een incorrect is en overeenkomt met deze van Van Ceulen. Helaas hebben we de andere waardes niet kunnen reproduceren, dus nu concluderen dat Van Ceulen deze methode gebruikt heeft kan ook niet.

Samengevat: Voor mij is het reproduceren van de fout bij de ingeschreven regelmatige 40-hoek een duidelijke aanwijzing naar het gebruik van Vieta's iteratieve methode door Van Cuelen, maar we hebben ook niet genoeg bewijs om te zeggen dat deze methode daadwerkelijk gebruikt is.

5.5 Opmerkingen algoritme

5.5.1 Aanpassingen op het algoritme

Op [5, p. 67] wordt vermeld dat Vieta niet altijd dezelfde keuze maakt voor zijn iteratieve algoritme, zo gebruikt hij soms ook een andere manier op a_{l+1} te berekenen. Deze komt bijna overeen met de vorige

gedefiniëerde a_{l+1} , op de teller na. Hij kiest soms voor de waarde a_{l+1} :

$$\left\lfloor \frac{\frac{1}{2}(f(x_l) + f(x_l + 10^{k-l-1})) - N}{f(x_l + 10^{k-l-1}) - f(x_l) - 10^{(k-l-1)n}} \right\rfloor \pmod{10}.$$

Echter produceert dit iteratieve proces bij testen altijd incorrecte resultaten. Vaak zijn de geproduceerde waarden 1 te hoog of 1 te laag. In tegenstelling tot de waarden die het eerste genoemde proces produceert. Deze zijn nagenoeg altijd goed. Daarom heb ik niet gekeken naar deze mogelijkheid.

5.5.2 Snelheid

Ten slotte maken we nog een opmerking over de snelheid van het algoritme. Het mooie aan het algoritme van Vieta is dat we precies weten hoeveel iteraties nodig zijn voor een precisie van n decimalen gegeven een bepaalde beginschatting. Dit is een gegeven dat we niet hebben bij andere methoden. Bovendien is ook gegeven hoeveel rekenwerk er gedaan moet worden om een extra decimaal precisie te krijgen. Bij elke iteratiestap moet het getal a_{l+1} berekend worden, deze wordt gegeven door:

$$\left\lfloor \frac{f(x_l) - N}{f(x_l + 10^{k-l-1}) - f(x_l) - 10^{(k-l-1)n}} \right\rfloor \pmod{10}$$

Helaas kunnen we niet om het feit heen dat we per iteratie een polynoom moeten evalueren. Dit kan erg tijdrovend werk zijn aangezien de graden van de polynomen erg hoog worden. Een wiskundige uit 1670 zegt het volgende over het Vieta's algoritme. Vertaald uit [5, p. 68]. *"werk ongeschikt voor een Christen, dit is geschikter voor personen die de Italiaanse Alpen naar Italië kunnen verplaatsen..."*

5.5.3 Fouten in tabel 2

Ten slotte wil ik nog uitleggen waarom we niet naar de fouten uit tabel 2 hebben gekeken. Hiervoor heb twee redenen.

Om dit uit te leggen zal ik eerst uitleggen hoe we het algoritme hebben geïmplementeerd.

We schrijven het programma in Python. Gegeven een polynoom f waarvan we een wortel N willen berekenen. We interpreteren een lijst als een getal. Dit doen we op de volgende manier. Laat $[a_1, a_2, a_3, \dots, a_l]$ een lijst zodat voor alle natuurlijke getallen i geldt dat $0 \leq a_i \leq 9$ en $a_i \in \mathbb{N}$. We sturen de lijst $[a_1, a_2, a_3, \dots, a_l]$ naar een breuk zodat

$$[a_1, a_2, a_3, \dots, a_l] \mapsto \sum_{i=1}^l a_i \cdot 10^{1-i}.$$

Als voorbeeld kijken we naar de lijst $[1, 9, 0, 1, 2]$, deze zou naar het getal $1,901200\dots$ gestuurd moeten worden.

Zij $x = [a_1, a_2, a_3, \dots]$ de wortel van f zodat $f(x) = N$ en $x_l = [a_1, a_2, \dots, a_l]$ een schatting van de wortel van f . Met behulp van Vieta's iteratieve methode kunnen we a_{l+1} vinden. Dit vinden we door

$$a_{l+1} = \left\lfloor \frac{f(x_l) - N}{g_k(x_l)} \right\rfloor \pmod{10}$$

Nu komt het voor dat bij het berekenen van

$$[a_1, a_2, a_3, \dots, a_l] \mapsto \sum_{i=1}^l a_i \cdot 10^{1-i}$$

er fouten door Python gemaakt worden. Bijvoorbeeld wordt de lijst $[1, 7, 9]$ naar $1,7900000000000002$ gestuurd. Hier maakt Python een fout van $2 \cdot 10^{-17}$. Deze fout kan er voor zorgen dat de waarde van a_{l+1} fout is. In het algemeen maakt Python fouten bij het berekenen van orde 10^{-17} . In tabel 2 hebben we een precisie nodig van 10^{-14} , dus hebben deze fouten nog geen invloed gehad bij het berekenen van de waarden uit tabel 2. Als we echter een precisie van bijvoorbeeld 10^{-30} willen narekenen hebben deze fouten wel invloed. Daarom zijn we op dit moment nog niet in staat om deze fouten na te rekenen.

De tweede reden is, dat er weinig fouten gemaakt worden in Tabel 2. De fouten die gemaakt worden, zijn niet overduidelijk rekenfouten. Daarom heb ik Tabel 2 voor nu achterwege gelaten.

6 Vergelijken van de besproken methoden

De hoofdvraag die we deze scriptie willen beantwoorden, is de volgende vraag: Welke numerieke methode heeft Van Ceulen gebruikt in hoofdstuk 14? In dit hoofdstuk proberen we deze vraag te beantwoorden. Dit gaan we doen door de gevonden methoden te vergelijken. Aan de hand hiervan versimpelen we de bovenstaande vraag, de vraag die we dit hoofdstuk gaan beantwoorden is de volgende: Welk van de besproken numerieke methoden is het meest waarschijnlijk gebruikt? We gaan de methoden op de volgende twee criteria vergelijken:

1. Hoe kunnen er fouten voortkomen uit de methode?
2. Hoeveel rekenstappen hebben we nodig om een gevraagde precisie te verkrijgen?

Een methode lijkt waarschijnlijker gebruikt als deze de fouten in Vanden Cirkel verklaart, en als deze methode snel is. We zullen alleen de iteratieve methode van Vieta en de interpolatie van Chuquet vergelijken. We kunnen de vaste punt iteratie van Chelebi namelijk buiten beschouwing laten omdat we deze alleen kunnen toepassen onder gunstige omstandigheden. Deze vastepunt iteratie kunnen we bijvoorbeeld niet op (4) en andere hogeregraads vergelijkingen toepassen. De vaste punt iteratie van Chelebi kan dus niet verklaren hoe Van Ceulen enkele hogere graads vergelijkingen met zo'n hoge precisie heeft benaderd. Bovendien is het onwaarschijnlijk de vaste punt iteratie van Chelebi bekend was bij Van Ceulen. De Arabische wiskunde was namelijk nog niet bekend in Nederland en Van Ceulen kon geen Arabisch lezen. Daarom zullen we ons nu beperken tot de numerieke methode van Vieta en de interpolatie van Chuquet.

We gaan eerst de methoden vergelijken aan de hand van de rekenfouten. Daarna zullen we kijken naar de snelheid van de methoden. Tot slot geven we antwoord op de versimpelde vraag.

6.1 Rekenfouten

We bespreken de rekenfouten die kunnen ontstaan bij de interpolatie van Chuquet eerst, daarna de fouten die kunnen ontstaan bij de iteratieve methode van Vieta. Nadat we dit hebben gedaan, kunnen we, en gaan we, de methoden vergelijken.

6.1.1 Rekenfouten bij de interpolatie van Chuquet

- Bij de interpolatie van Chuquet zou er in theorie weinig kunnen gebeuren waardoor benaderingsfouten ontstaan. Dit komt omdat het algoritme zelfcorrigerend is, elke stap van een iteratie controleren we of er een nulpunt in het resulterende interval ligt. Als er geen nulpunt in het interval ligt, weten we dus dat er in een van de voorgaande stappen een fout is gemaakt. De interpolatie van Chuquet verklaart dus niet de grote fouten die bij de ingeschreven regelmatige 70 en 55 hoek gemaakt zijn.
- De interpolatie van Chuquet zou wel veel van de fouten bij de laatste decimalen kunnen verklaren als Van Ceulen een precisie van ongeveer 10^{-14} gekozen had. Echter zouden er meer fouten moeten zijn bij het laatste decimaal als van Ceulen deze keuze gemaakt zou hebben.
- Bij de interpolatie van Chuquet kunnen er ten slotte nog fouten gemaakt worden bij het omschrijven van breuken naar decimale notatie. Dit zou wellicht de grote fouten kunnen verklaren omdat dit een iteratief proces is. We hebben laten zien dat omschrijven van breuken naar de decimale expansie een langdurig proces is, maar zeker makkelijk in verhouding met het algoritme zelf. Daarom lijkt het mij onwaarschijnlijk dat de fouten hiervandaan komen.

6.1.2 Rekenfouten bij de iteratieve methode van Vieta

- Bij Vieta's iteratieve methode kan er meer fout gaan dan bij de interpolatie van Chuquet. Zoals we gezien hebben bij de ingeschreven regelmatige 40-hoek geeft de iteratieve methode van Vieta soms foute resultaten. Toen we bewezen hebben dat de methode werkt, hebben we de Newton-Rahpson methode aangepast. Hierbij was het grote verschil dat we voor een differentieerbare functie $f: \mathbb{R} \rightarrow \mathbb{R}$, de afgeleide $f'(x)$ hebben geschat door

$$f'(x) \approx \frac{f(x + 10^m) - f(x) + 10^m}{10^m}$$

De iteratieve methode van Vieta geeft soms dus foute resultaten.

- Bovendien is, in tegenstelling tot de interpolatie van Chuquet, de iteratieve methode van Vieta niet zelfcorrigerend. Als er een fout gemaakt wordt door de methode of als er een rekenfout gemaakt wordt, dan komen we hier niet achter.
- Omdat het een numerieke methode van Vieta een iteratief proces is, kunnen rekenfouten invloed hebben op de rest van de schatting. Zo kan de iteratieve methode van Vieta de grotere rekenfouten zoals deze bij de ingeschreven regelmatige 55-hoek verklaren.
- We konden de foute benadering van Van Ceulen bij de ingeschreven regelmatige 40-hoek reproduceren met de iteratieve methode van Vieta, en de fout bij de ingeschreven regelmatige 55-hoek deels.

Nu gaan we de numerieke methoden vergelijken op de mogelijkheid van foute resultaten als gevolg van de methode. De iteratieve methode van Vieta lijkt meer te kunnen verklaren dan de methode van Chuquet. We kunnen enkele fouten gemaakt in Vanden Cirkel reproduceren met behulp van de iteratieve methode van Vieta. Dit konden we niet doen met de interpolatie van Chuquet. Bovendien lijkt de interpolatie van Chuquet op geen enkele manier de foute resultaten te verklaren, terwijl de iteratieve methode van Vieta dit wel kan. De iteratieve methode van Vieta kan niet alleen de kleine fouten verklaren, maar ook de grotere fouten. Als we de methoden vergelijken op de fouten, lijkt de iteratieve methode van Vieta waarschijnlijker gebruikt dan de interpolatie van Chuquet.

6.2 Snelheid algoritme

Ten slotte kijken we naar de snelheden van de algoritmes. We doen hier de aannames dat de numerieke methoden uitgevoerd worden op de manier waarop ik de methoden geef. We zullen eerst kijken hoe lang een iteratiestap duurt bij beide methoden, hier nemen we alleen de onderlinge verhouding in beschouwing. Als we dit hebben gedaan kijken we hoeveel iteratiestappen we nodig hebben voor een bepaalde precisie. Aan de hand hiervan kunnen we beargumenteren welke methode sneller is.

Eerst herhaal ik kort welke rekenstappen we moeten zetten bij elke iteratie. Bij het algoritme van Chuquet maken we bij elke iteratiestap een nieuw getal waarvan we controleren of deze links of rechts van het nulpunt ligt. Een voordehandliggende manier om dit te controleren was het evalueren van polynomen. Bij het algoritme van Vieta moeten we bij elke stap twee keer een polynoom evalueren. We zien het evalueren van polynomen als het meest tijdrovend in beide iteratiestappen en alle andere rekenstappen als verwaarloosbaar in verhouding met het evalueren van polynomen. Met deze aanname, zou de iteratieve methode van Vieta per iteratiestap ongeveer twee keer zo lang duren als de interpolatie van Chuquet. Ik laat hier weg dat we eigenlijk nog meer moeten doen per iteratiestap bij de interpolatie van Chuquet, bijvoorbeeld controleren of het resulterende interval klein genoeg is of controleren waar het nulpunt ligt. Dit hoeven we niet elke iteratiestap te doen, maar ongeveer om de 8 iteraties. Met dit in ons achterhoofd gaan we kijken hoeveel iteratiestappen we nodig hebben voor een bepaalde precisie.

6.2.1 Snelheid iteratieve methode van Vieta

- Voor het algoritme van Vieta kunnen we makkelijk zeggen dat we 13 iteraties nodig hebben voor de precisie van 14 decimalen, aangenomen dat we een beginschatting geven met een precisie van 1 correct decimaal.
- Als we een precisie willen hebben van 32 decimalen, dan zouden we volgens dezelfde argumentatie 33 iteraties nodig hebben.

6.2.2 Snelheid interpolatie van Chuquet

- Bij de interpolatie van Chuquet hebben we meer iteraties voor een precisie van 14 decimalen. Uit de numerieke experimenten bleek dat 40 iteraties een heel gunstig aantal iteraties is voor een precisie van 14 decimalen.

- Ik vermoed dat de interpolatie van Chuquet exponentieel meer iteraties nodig heeft dan Vieta's iteratieve methode bij hogere precisie. Zoals we gezien hebben bij de numerieke experimenten van de interpolatie van Chuquet, nam het aantal iteraties af met ongeveer $\sqrt{10}$ bij een betere beginschatting van orde 10. Andersom, als onze eindschatting orde 10 beter moet zijn, bijvoorbeeld 15 decimalen in plaats van 14 decimalen precisie, hebben we $\sqrt{10}$ meer iteraties nodig. Volgens deze redenering zouden we ongeveer 600 iteraties nodig hebben om een precisie van 31 decimalen te verkrijgen. Dit is significant meer dan 31 iteraties.

Bij een precisie van 14 decimalen heeft de interpolatie van Chuquet onder gunstige omstandigheden 40 iteraties nodig. Dit zijn er ongeveer drie keer zo veel als de iteratieve methode van Vieta. De iteratieve methode van Vieta duurt ongeveer twee keer zo lang per iteratiestap, dus de iteratieve methode van Vieta zou ongeveer anderhalf keer zo snel zijn als de interpolatie van Chuquet.

Bij de precisie van 33 decimalen zou de interpolatie van Chuquet onder gunstige omstandigheden 600 iteraties nodig hebben. De iteratieve methode van Vieta heeft hiervoor 33 iteraties nodig. De iteratieve methode van Vieta zou hier 10 keer zo snel zijn.

In beide gevallen is de iteratieve methode van Vieta sneller. Hier hebben we de interpolatie van Chuquet ook nog eens enorm versimpeld; in de werkelijkheid zou je nog veel meer rekenstappen moeten zetten. Bovendien zijn we ook nog uitgegaan van zeer gunstige omstandigheden voor de interpolatie van Chuquet. We kunnen daarom concluderen dat de iteratieve methode van Vieta sneller is dan de interpolatie van Chuquet.

Ten slotte gaan we nu antwoord geven op de vraag: Welk van de gevonden numerieke methoden lijkt het meest waarschijnlijk gebruikt? Deze vraag kunnen we nu gemakkelijk beantwoorden. De iteratieve methode van Vieta is sneller dan de interpolatie van Chuquet. Ook verklaart de iteratieve methode van Vieta fouten die Van Ceulen maakt. Daarom is, van de drie besproken methoden, de iteratieve methode van Vieta het meest waarschijnlijk gebruikt.

7 Slotwoord

Tijdens het schrijven van mijn scriptie verbaast het mij keer op keer dat Van Ceulen de oplossingen van de polynomen met zo'n enorme precisie heeft benaderd. Je komt er niet onderuit om polynomen te evalueren, dit alleen al is gekkenwerk voor polynomen die Van Ceulen produceert, bijvoorbeeld polynomen van graad hoger dan 44. Dit laat alleen al zien wat voor rekenmeester Van Ceulen was. Wat men leert van het lezen van deze scriptie is hopelijk niet alleen dit.

Hopelijk krijgt men ook een idee van de werkwijze van Van Ceulen, hoe hij de polynomen opstelt. We kunnen ook meteen begrijpen waarom we op zoek zijn naar de kleinste positieve oplossing van de opgestelde polynomen. Hierna gingen we over op een opsomming van numerieke methoden uit de tijd van Van Ceulen. We hebben drie methoden onderzocht. Als we ons best doen begrijpen we ook de werking van de methoden, omdat de werking van de methoden bewezen worden. Ook krijgen we een beeld van de werking van de methoden op de polynomen die Van Ceulen opstelt. De werking van de vastepuntiteratie van Chelebi op sommige polynomen van Van Ceulen was minder wenselijk. De methode convergeert namelijk niet. Dit is niet het geval bij de interpolatie van Chuquet. Deze convergeert netjes, maar is daarentegen wel erg tijdrovend. De iteratieve methode van Vieta convergeert bijna altijd netjes naar het correcte nulpunt, en convergeert bovendien ook nog eens heel snel. Op één getest polynoom na gaf deze de correcte waarde, dit was het polynoom van de ingeschreven 40-hoek. Het resultaat van de iteratieve methode van Vieta komt precies overeen met de incorrect gegeven waarde van Van Ceulen. Wat we hier uit kunnen halen weet ik niet zo goed, maar het is zeker een aanwijzing naar het gebruik van Vieta's iteratieve methode door Van Ceulen!

Na de opsomming van de mogelijk gebruikte methoden, hebben we onszelf de vraag gesteld welke hiervan het meest waarschijnlijk gebruikt is. We konden gelijk de methode van Chelebi afstrepen, omdat deze toch niet bekend was bij Van Ceulen en niet convergeert voor de polynomen van Van Ceulen. Dan hebben we nog de iteratieve methode van Vieta en de interpolatie van Chuquet over. We merkten op dat de interpolatie van Chuquet niet tot fouten kon leiden, in tegenstelling tot de methode van Vieta. Bovendien is de methode van Vieta sneller dan die van Chuquet. Daarom is de methode van Vieta waarschijnlijker gebruikt dan de methode van Chuquet.

Hiermee wil ik ook een einde maken aan de scriptie. Het belangrijkste resultaat van mijn scriptie is dat ik denk dat de iteratieve methode van Vieta gebruikt is. Hiermee vermoed ik dus ook dat de interpolatie van Chuquet niet gebruikt is. Echter heb ik deze uitspraken niet kunnen bewijzen. Hopelijk kunnen we dit in de toekomst wel doen. Hieronder geef ik nog een opsomming van zaken waar men nog naar kan kijken om erachter te komen welke methode Van Ceulen gebruikt heeft. Tijdens het schrijven van de scriptie ben ik hier niet aan toe gekomen, maar het lijken mij wel zaken waarmee we erachter kunnen komen welke methode Van Ceulen gebruikt heeft.

7.1 Advies voor vervolgonderzoek

Ik heb geen sluitend antwoord kunnen geven op de vraag: Welke methode heeft van Ceulen gebruikt om de nulpunten te berekenen? Daarentegen heb ik er wel een beeld geschetst van de numerieke methoden in de 16e eeuw. Bovendien ben ik niet volledig geweest in mijn onderzoek. Dit omdat ik niet genoeg tijd heb en in sommige punten over het hoofd heb gezien. Daarom wil ik graag wijzen op punten die nog onderzocht of bewezen kunnen worden. De adviezen die ik geef, geef ik op dezelfde volgorde van de hoofdstukken van mijn scriptie.

7.1.1 Chuquet

Om de methode van Chuquet uit te sluiten met een hard argument zouden we willen analyseren hoe snel deze methode convergeert. Ik vermoed dat de methode significant langzamer convergeert ten opzichte van Vieta. Aan het bewijzen hiervan ben ik niet toegekomen, maar zou het naar mijn idee de makkelijkste stap zijn om uit te sluiten dat de methode van Chuquet gebruikt is. Dit hoeft niet eens analytisch bewezen te worden, met een aantal numerieke experimenten zou ook al een goed resultaat vergaakt kunnen worden. Met mijn implementatie van de interpolatie van Chuquet kon ik geen precisie van 33 decimalen verkrijgen, dus ik wist ook niet hoeveel iteraties je hier ongeveer voor nodig hebt. Met een betere implementatie

zouden we er wel achter kunnen komen hoeveel iteraties we nodig hebben voor zo'n grote precisie. Als het zo blijkt te zijn dat de methode van Chuquet snel convergeert, zouden we ook graag een manier willen bedenken hoe we kunnen narekenen of we de fouten van Van Ceulen kunnen reproduceren als we van breuken naar decimale notatie gaan, omdat hier wellicht fouten ontstaan. Ten slotte kan er nog nagedacht worden over hoe Ludolph dit algoritme slimmer had kunnen maken. Hier denk ik vooral aan een slimme manier om te controleren of een punt links of rechts van een nulpunt ligt. Dan zouden we ook graag willen weten hoe snel het algoritme dan zou zijn.

7.1.2 Vieta

Er zijn een aantal zaken waar ik niet aan toe ben gekomen te bespreken of bewijzen bij het algoritme van Vieta. Het belangrijkste van deze zaken is het bewijzen dat de iteratieve methode van Vieta convergeert voor alle polynomen die Ludolph geeft in hoofdstuk 14. Uit alle numerieke experimenten bleek het zo te zijn dat de methode van Vieta convergeert. Daarom zag ik het niet dus urgent om te bewijzen dat de methode voor alle gevallen convergeert. Als we vermoeden dat Ludolph deze methode gebruikt heeft, dan willen we ook bewijzen dat de methode convergeert om gegronde argumenten voor het gebruik van deze methode te hebben. Daarnaast zouden we ook graag na willen rekenen of de iteratieve methode van Vieta ook alle correcte waardes uit tabel 2 reproduceert. Dit lukte mij niet, we zouden graag een implementatie schrijven die wel in staat is om deze precisie te verkrijgen. Ten slotte zouden we de iteratieve methode van Vieta slimmer kunnen maken door de waardes van de polynomen te schatten. In [4, p. 501-504] bespreekt Trofke twee methoden die het berekenen van polynomen gemakkelijker maken. Het kan geen kwaad om hiernaar te kijken. Een van deze methoden is een exacte methode, de ander geeft een schatting van de waarde van de polynomen. Ik vermoed dat de schatting niet voor veel fouten zal zorgen omdat we bij de iteratieve methode van Vieta een waarde naar beneden afronden. Met deze methode van polynomen evalueren zouden we het algoritme van Vieta weer na kunnen rekenen. Deze zouden met heel wat *wishful thinking* dezelfde resultaten als hoofdstuk 14 kunnen reproduceren. Ten slotte zouden we ook nog kunnen kijken of het aangepaste algoritme, zie [5.6.1, 37], tot de resultaten van Van Ceulen leidt.

7.1.3 Literatuur

Ik weet niet of ik alle mogelijke numerieke methoden heb besproken die snel convergeren. Het lijkt mij onwaarschijnlijk dat ik alle mogelijke bekende methoden heb besproken. Daarom kan er gerust verder gekeken worden naar andere methoden, met in beschouwing genomen dat ik al heb gekeken naar de onderstaande bronnen.

Referenties

- [1] Ludolph van Ceulen, Vanden Cirkel, 1596, Hoofdstuk 14.
- [2] Sophie Verberne en Sonja Scheer, Onderzoek naar de konstige vergelijkingen van Van Ceulen, 14 juni 2006, pagina 5.
- [3] Controle van Van Ceulens antwoorden, Marloes Bazelier en Marloes Turk, 17 juli 2006, Tabel 2.2.
- [4] Johannes Tropfke, Geschichte der Elementarmathematik, Band 1 · Arithmetik und Algebra, Walter de Gruyter, Berlijn 1980, Hoofdstuk 3.
- [5] Hermand H.Goldstine, Springer-Verslag, New York Heidelberg Berlin 1977, Hoofdstuk 1 parragraaf 6.
- [6] François Vieta, Opera Mathematica, 1646.
- [7] Tritanstan van Leeuwen, Utrecht, Niet-lineaire vergelijkingen, Numerieke wiskunde 2018.
- [8] Herman Hankel, Hildesheim, Zur geschichte der mathematik in altetum und mittertalerm, Tweede druk 1965.
- [9] François Viéte, The Analytic Art, vertaald door T. Richard Witmer, The Kent State University Press 1983.
- [10] Steven Wepster, Van Ceulens Veelhoeken en Veeltermen, Nieuwe Wiskrant, 9 September 2008.
- [11] Wepster, S., & Rijnierse, M. (2010, 22 januari). Ludolph van Ceulen. Geraadpleegd van <http://web.science.uu.nl/ludolphvanceulen/site/biografie.php>