Uitbreidingen van Fouriers 17 lijnen probleem

Jelte J. Zwetsloot
Begeleider Dr. W. Bosma

Radboud Universiteit Nijmegen

1 juli 2015

Inhoudsopgave

Hoofdstuk 1

Fouriers 17 lijnen probleem

In 1788 schreef de wiskundige Joseph Fourier (1768-1830) in een brief naar zijn toenmalige docent wiskunde C.L. Bonard het volgende. *"Hier is een probleem van een nogal bijzondere aard: het kwam in me op in relatie tot verschillende Euclidische stellingen die we besproken hebben. Orden 17 lijnen in hetzelfde vlak zodat er precies 101 snijpunten zijn. Het wordt aangenomen dat de lijnen zich uitstrekken tot in het oneindige en dat geen enkel snijpunt tot meer dan twee lijnen behoort."* Tegenwoordig staat dit probleem bekend als het 17 lijnen probleem van Fourier of simpelweg het 17 lijnen probleem. Fourier vroeg zich af of er met het gebruik van alleen groepen van parallelle lijnen een configuratie te vinden was die precies 101 snijpunten zou geven in het Euclidische vlak. Dit probleem is niet het meest ingewikkelde en er kwamen dan ook snel oplossingen uit. Neem bijvoorbeeld vier parallelle groepen met respectievelijk 2, 3, 4 en 8 lijnen. Dan heb je 17 lijnen in totaal en $2\times3+2\times4+2\times8+3\times4+3\times8+4\times8=101$ punten.

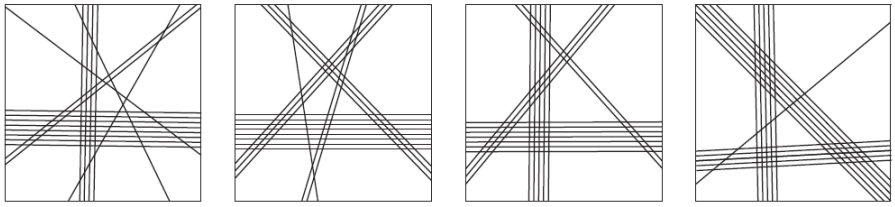
Fouriers 17 lijnen probleem is op twee manieren aan te pakken. De eerste manier is om tekenend naar een oplossing toe te werken. Begin met het tekenen van 1 lijn, voeg daar nog een lijn aan toe, enzovoorts, totdat er 17 lijnen op papier staan. Als de lijnen op een goede manier zijn toegevoegd, staan er ook 101 snijpunten op papier. Helaas is dat lang niet altijd het geval. Het is duidelijk dat deze methode nogal lang duurt.

De tweede manier is handiger. Hierbij wordt gebruik gemaakt van een numerieke aanpak. Het doel is om 17 lijnen te verdelen over een aantal parallelle groepen, ook wel families van parallelle lijnen genoemd, zodat er precies 101 snijpunten ontstaan. Het is duidelijk dat een lijn uit een bepaalde parallelle groep snijdt met alle lijnen die niet in die groep zitten. Per parallelle groep kan dan gemakkelijk het aantal snijpunten worden geteld. Het totaal aantal snijpunten van alle parallelle groepen geeft dan precies 202 punten, omdat elk punt twee keer wordt geteld. In wiskundige notatie ziet dat er als volgt uit:

$$\sum_{i=1}^{k} x_i = 17 \text{ en } \sum_{i=1}^{k} x_i(17 - x_i) = 202$$

met k het aantal families van parallelle lijnen en x_i het aantal lijnen in de i^e familie

Dit stelsel vergelijking kent een aantal oplossingen, vier oplossingen om precies te zijn. Deze oplossingen zijn van de vorm $\langle x_1, ..., x_k \rangle$ met wederom k het aantal families van parallelle lijnen en x_i het aantal lijnen in de i^e familie. Als er meerdere families zijn met gelijke aantallen lijnen, zeg l families met x_j lijnen, dan wordt dat afgekort tot x_j^l. De oplossingen voor Fouriers 17 lijnen probleem zijn te vinden in de afbeelding hieronder.



(a) $\langle 8, 4, 2, 1^3 \rangle$ (b) $\langle 8, 3^2, 2, 1 \rangle$ (c) $\langle 7, 5, 3, 2 \rangle$ (d) $\langle 6, 5^2, 1 \rangle$

Hiermee is het probleem van Fourier en de 17 lijnen opgelost. Gelukkig is het probleem op vele manieren uit te breiden tot bredere en ingewikkeldere problemen. In de volgende sectie wordt een korte inleidende omschrijving gegeven van de twee varianten op Fouriers 17 lijnen probleem die in deze scriptie behandeld zullen worden.

1.1 Uitbreidingen van het 17 lijnen probleem

De eerste uitbreiding van het probleem lijkt het meest op het oorspronkelijke probleem. Het verschil is dat nu gekeken zal worden naar een willekeurig aantal lijnen en een willekeurig aantal punten. De andere restricties blijven gelijk: lijnen mogen parallel aan elkaar zijn en door een snijpunt gaan precies twee lijnen. Ook het doel blijft gelijk: probeer een dusdanige configuratie van het aantal lijnen in het Euclidische vlak te vinden dat er precies zoveel snijpunten zijn als gevraagd werd. Om deze beschrijving wat concreter te krijgen wordt wederom de numerieke aanpak bekeken. Ga er van uit dat er een oplossing bestaat voor bepaalde n en m met n het aantal lijnen en m het aantal punten en k het aantal families van parallelle lijnen met x_i het aantal lijnen in de i^e familie. Dan geldt

$$\sum_{i=1}^{k} x_i = n \text{ en } \sum_{i=1}^{k} x_i(n - x_i) = 2m$$

of handiger:

$$\sum_{i=1}^{k} x_i = n \text{ en } \sum_{i=1}^{k} x_i^2 = n^2 - 2m$$

Op deze manier kan het probleem als volgt omschreven worden.

Probleem 1
Laten n het aantal lijnen en m het aantal punten zijn. Bestaan er positieve gehele getallen $x_1, ..., x_k$ zodanig dat $\sum_{i=1}^{k} x_i = n$ en $\sum_{i=1}^{k} x_i^2 = n^2 - 2m$? Oplossingen zijn dan van de vorm $\langle x_1, ..., x_k \rangle$.

Dit probleem heet de generalisatie van Fouriers 17 lijnen probleem en zal uitgebreid aan bod komen in Hoofdstuk 2.

Om de overgang te maken tot de tweede variant die uitgebreid bekeken zal worden, wordt eerst de aandacht gevestigd op een ander probleem dat gerelateerd is aan de generalisatie van Fouriers 17 lijnen probleem. Dit probleem heeft wederom als doel om een kloppende configuratie te vinden voor willekeurige aantallen lijnen en punten, maar ditmaal met andere restricties. Lijnen zijn nu niet parallel aan elkaar en door een snijpunt gaan minstens twee lijnen. De orde van een punt is het aantal lijnen dat door het punt heen gaat. Als deze orde groter dan twee is wordt het punt een multiplicatief punt genoemd, of makkelijker, een munt. In wiskundige notatie is het probleem dan als volgt te omschrijven.

Probleem 2
Laten n het aantal lijnen en m het aantal punten zijn. Bestaan er positieve gehele getallen $\lambda_1, ..., \lambda_m$ groter of gelijk aan twee zodat geldt $m = \binom{n}{2} - \sum_{i=1}^{m} [\binom{\lambda_i}{2} - 1]$? Hier zijn $\lambda_1, ..., \lambda_m$ de ordes van de punten $p_1, ..., p_m$. Oplossingen zijn van de vorm $\langle \lambda_1, ..., \lambda_m \rangle$.

Bovenstaand probleem geeft de aansluiting tot het grotere probleem dat bekeken zal worden, namelijk de generalisatie van Fouriers 17 lijnen probleem zonder restricties. Bij deze variant zal voor willekeurige n en m gekeken worden of er een configuratie bestaat met n lijnen en m punten, waarbij lijnen parallel aan elkaar mogen zijn en waarbij door elk snijpunt minstens twee punten gaan. Door de vorige twee problemen met elkaar te combineren, volgt de volgende beschrijving van het probleem.

Probleem 3
Laten n het aantal lijnen en m het aantal punten zijn. Bestaan er positieve gehele getallen μ_1,...,μ_k en positieve gehele getallen $\lambda_1, ..., \lambda_m$ groter of gelijk aan twee zodat geldt $m = \binom{n}{2} - \sum_{i=1}^{m}[\binom{\lambda_i}{2} - 1] - \sum_{j=1}^{k}\binom{\mu_j}{2}$? Hier zijn $\mu_1, ..., \mu_k$ het aantal lijnen per familie van parallelle lijnen en $\lambda_1, ..., \lambda_m$ de ordes van de punten $p_1, ..., p_m$. Oplossingen zijn van de vorm $[\langle\lambda_1, ..., \lambda_m\rangle,\langle\mu_1, ..., \mu_k\rangle]$.

De generalisatie van Fouriers 17 lijnen probleem zonder restricties, of simpeler, Fouriers 17 lijnen probleem zonder restricties, zal grondig bekeken worden in Hoofdstuk 3.

Merk op dat bij bovenstaande problemen niet per se alleen voldaan moet zijn aan de eisen die in de probleemomschrijving genoemd worden. Soms eist de vergelijking dat er aan extra restricties voldaan moet worden. Hoe deze restricties precies invloed hebben en wat in die zin hun betekenis is wordt uitgelegd in de desbetreffende hoofdstukken.

Hoofdstuk 2

Generalisatie van Fouriers 17 lijnen probleem

In dit hoofdstuk zal de focus liggen op Probleem 1: de Generalisatie van Fouriers 17 lijnen probleem. Hierbij stellen we de volgende vraag:

Vraag 2.1 Zij n,$m \in \mathbb{N}$. Bestaan er positieve gehele getallen $x_1, ..., x_k$ zodanig dat $\sum_{i=1}^{k} x_i = n$ en $\sum_{i=1}^{k} x_i^2 = n^2 - 2m$?

Het doel van dit hoofdstuk zal zijn om deze vraag te kunnen beantwoorden met ja of nee. Als het antwoord ja is, is er kennelijk een bepaalde configuratie in het Euclidische vlak met n lijnen en m punten waar de n lijnen over een bepaald aantal families van parallelle lijnen zijn verdeeld. Let wel op: het doel is NIET om een daadwerkelijke oplossing te vinden. We zijn alleen geïnteresseerd in de vraag of er een oplossing is.

Om antwoord te geven op deze vraag zal gekeken worden naar het algoritme van Woeginger [**1**] dat gedeeltelijk een antwoord levert op Vraag 2.1. Dit algoritme wordt grondig geanalyseerd en vervolgens aangepast om een nieuw algoritme te creëren dat wel in volledigheid antwoord geeft op Vraag 2.1.

2.1 Woegingers algoritme

Woeginger was geïnteresseerd in Probleem 1 en wist met gebruik van slimme lemma's een algoritme te ontwikkelen dat een gedeeltelijk antwoord geeft op Vraag 2.1. In zijn algoritme gebruikt Woeginger de positieve gehele getallen S en Q. De S van Woeginger is zoals de n hier gebruikt wordt, het aantal lijnen, en de Q stelt bij Woeginger $n^2 - 2m$ voor. Een paar (S, Q) heet *toelaatbaar* als er positieve gehele getallen $x_1, ..., x_k$ bestaan zodat $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$. $\langle x_1, ..., x_k \rangle$ heet dan een *certificaat* van (S, Q).

Algoritme van Woeginger
Laten S en Q willekeurige positieve gehele getallen zijn en laat dit de input zijn. (1) als $S \leq 8060$, werk dan het probleem uit met de hand. STOP. (2) als $Q < S$ of als $Q > S^2$ of als S en Q van verschillende pariteit zijn, output NO en STOP. (3) als $S \leq Q \leq S(S - 6\sqrt{S})$, output YES en STOP. (4) als $S \geq 8061$ en $S(S - 6\sqrt{S}) < Q \leq S^2$, vindt dan alle gehele getallen ξ die voldoen aan $\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi \leq \sqrt{Q}$. Los voor elke gevonden ξ het geval $(S - \xi, Q - \xi^2)$ recursief op. Output YES en STOP dan en slechts dan als tenminste één van deze uitkomsten YES is. Anders, output NO.

Als dit algoritme YES als output geeft, zijn er $x_1, ..., x_k$ zodanig dat $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$ en is (S, Q) dus toelaatbaar. Als dit algoritme NO als output geeft is (S, Q) niet toelaatbaar.

Het is een mooi algoritme, maar is nog niet optimaal dankzij stap 1. Voor S kleiner dan 8060 is het namelijk ongelooflijk veel werk om alle gevallen met de hand uit te gaan werken. Heel precies gezegd: met de restricties van stap 2 (die later worden uitgelegd) zijn er $\sum_{j=1}^{8060} \frac{j^3 - j^2}{2} = 527577291383545$ gevallen die handmatig uitgewerkt moeten worden. Kortom, dit moet makkelijker kunnen. Ondanks de ietwat vreemde grens van 8060, biedt het algoritme wel veel houvast. Het werkt namelijk prima voor alle S groter dan 8060. Daarom nemen we dit algoritme van Woeginger als basis voor het eigen algoritme. Om Woegingers algoritme aan te passen is het eerst nodig te weten waarom dit algoritme werkt.

2.2 Waarom werkt Woegingers algoritme

Het algoritme van Woeginger bestaat uit vier stappen, waarvan inmiddels duidelijk is dat de eerste stap een probleem vormt. Om dit probleem op te lossen gaan we Stappen 2, 3 en 4 bekijken. Te beginnen bij Stap 2.

Stap 2 *Als $Q < S$ of als $Q > S^2$ of als S en Q van verschillende pariteit zijn, output NO en STOP.*

Deze stap bestaat uit drie stukken. Elk stuk stelt een voorwaarde. Als aan die voorwaarde voldaan is, moet de output NO zijn. Hier staat dus dat er geen certificaat te vinden is voor S en Q die voldoen aan de één van de voorwaarden uit Stap 2. Hieronder staat dit bewezen voor elke voorwaarde.

Bewijs stap 2a Zij $Q, S \in \mathbb{N}$. Stel $Q < S$. Voor elke $k \in \mathbb{N}$ geldt voor elk k-tal positieve gehele getallen $x_1, ..., x_k$ dat $\sum_{i=1}^{k} x_i^2 \geq \sum_{i=1}^{k} x_i$. Stel nu

dat er een certificaat is voor (S, Q). Dan is er een $k \in \mathbb{N}$ zodat bepaalde positieve gehele getallen $x_1, ..., x_k$ dit certificaat vormen. Dan geldt $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$, maar $Q = \sum_{i=1}^{k} x_i^2 \geq \sum_{i=1}^{k} x_i = S > Q$: tegenspraak. Dus er is geen certificaat voor (S, Q), dus (S, Q) is niet toelaatbaar. □

Bewijs stap 2b Zij $Q, S \in \mathbb{N}$. Stel $Q > S^2$. Voor elke $k \in \mathbb{N}$ geldt voor elk k-tal positieve gehele getallen $x_1, ..., x_k$ dat $(\sum_{i=1}^{k} x_i)^2 \geq \sum_{i=1}^{k} x_i^2$. Stel nu dat er een certificaat is voor (S, Q). Dan is er een $k \in \mathbb{N}$ zodat bepaalde positieve gehele getallen $x_1, ..., x_k$ dit certificaat vormen. Dan geldt $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$, maar $S^2 = (\sum_{i=1}^{k} x_i)^2 \geq \sum_{i=1}^{k} x_i^2 = Q > S^2$: tegenspraak. Dus er is geen certificaat voor (S, Q), dus (S, Q) is niet toelaatbaar. □

Bewijs stap 2c Zij $Q, S \in \mathbb{N}$. Stel Q en S zijn van verschillende pariteit. Voor elke $k \in \mathbb{N}$ geldt voor elk k-tal positieve gehele getallen $x_1, ..., x_k$ dat $\sum_{i=1}^{k} x_i^2$ en $\sum_{i=1}^{k} x_i$ gelijke pariteit hebben. Stel nu dat er een certificaat is voor (S, Q). Dan is er een $k \in \mathbb{N}$ zodat bepaalde positieve gehele getallen $x_1, ..., x_k$ dit certificaat vormen. Dan geldt $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$, dus hebben Q en S gelijke pariteit, maar dat is in tegenspraak met de aanname. Dus er is geen certificaat voor (S, Q), dus (S, Q) is niet toelaatbaar. □

Hiermee is de volledige verantwoording voor Stap 2 gegeven.

Stap 3 *Als $S \leq Q \leq S(S - 6\sqrt{S})$, output YES en STOP.*

Deze stap is samen met de informatie uit de voorgaande stappen te vertalen naar het volgende lemma.

Lemma 1. *Laten S en Q twee positieve gehele getallen van gelijke pariteit zijn. Als $S \leq Q \leq S(S - 6\sqrt{S})$, dan is (S, Q) een toelaatbaar paar.*

Voor het precieze bewijs van dit lemma verwijzen we naar het artikel van Woeginger [**1**].

Stap 4 *Als $S \geq 8061$ en $S(S - 6\sqrt{S}) < Q \leq S^2$, vind dan alle gehele getallen ξ die voldoen aan $\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi \leq \sqrt{Q}$. Los voor elke gevonden ξ het geval $(S - \xi, Q - \xi^2)$ recursief op. Output YES dan en slechts dan als tenminste één van deze uitkomsten YES is.*

Wiskundig gezien is dit de meest complexe stap uit het algoritme, maar intuïtief is het aardig te begrijpen: er wordt gezocht naar gehele getallen ξ binnen een bereik dat wordt bepaald door de input (S, Q). Beschouw daarna voor elke gevonden ξ het paar $(S - \xi, Q - \xi^2)$ als nieuwe input voor hetzelfde algoritme. Blijf doorgaan met dit proces totdat er een toelaatbaar paar wordt gevonden óf totdat er geen opties meer zijn om bij langs te gaan.

Je kunt je hier verschillende dingen bij afvragen, zoals waarom hiermee nou wordt opgelost dat (S, Q) toelaatbaar is of niet, je kijkt immers al snel naar $(S-\xi, Q-\xi^2)$ en niet meer naar (S, Q), of waarom de begrenzing van ξ gekozen is zoals hij hier staat. Antwoorden op al dit soort vragen worden geleverd door Lemma 2 en Lemma 3 hieronder. Lemma 3 is inclusief bewijs overgenomen van Woeginger. Na het geven van de lemma's en de bijbehorende bewijzen volgt een uitleg die Stap 4 van volledige verantwoording voorziet.

Lemma 2. *a) Zij (S, Q) een toelaatbaar paar. Zij dan $\langle x_1, ..., x_k\rangle$ een certificaat voor (S, Q) en zij $\xi := max_{1\leq i\leq k} x_i$. Dan is ook $(S-\xi, Q-\xi^2)$ toelaatbaar.*
b) Zij (S, Q) een toelaatbaar paar en zij y een positief geheel getal. Dan is $(S+y, Q+y^2)$ ook een toelaatbaar paar.

Bewijs. a) (S, Q) is toelaatbaar met certificaat $\langle x_1, ..., x_k\rangle$ en $\xi := \max_{1\leq i\leq k} x_i$. Zij dan $j \in \{1, ..., k\}$ zodat $\xi = x_j$. Dan is $\sum_{\substack{i=1\\ i\neq j}}^{k} x_i = S - \xi$ en $\sum_{\substack{i=1\\ i\neq j}}^{k} x_i^2 = Q - \xi^2$, dus $\langle x_1, ..., x_{j-1}, x_{j+1}, ..., x_k\rangle$ is een certificaat voor $(S-\xi, Q-\xi^2)$, dus $(S-\xi, Q-\xi^2)$ is toelaatbaar.
b) (S, Q) is toelaatbaar. Dan zijn er positieve gehele getallen $x_1, ..., x_k$ voor bepaalde $k \in \mathbb{N}$ zodat $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$. Neem nu $x_{k+1} = y$. Dan geldt $\sum_{i=1}^{k+1} x_i = S + y$ en $\sum_{i=1}^{k+1} x_i^2 = Q + y^2$, dus $\langle x_1, ..., x_{k+1}\rangle$ is een certificaat voor $(S+y, Q+y^2)$, dus $(S+y, Q+y^2)$ is toelaatbaar. □

Lemma 3. *a) Zij (S, Q) een toelaatbaar paar dat voldoet aan $S(S-6\sqrt{S}) < Q \leq S^2$. Zij $\langle x_1, ..., x_k\rangle$ een certificaat voor (S, Q) met $\sum_{i=1}^{k} x_i = S$ en $\sum_{i=1}^{k} x_i^2 = Q$, en zij $\xi := max_{1\leq i\leq k} x_i$. Dan voldoet ξ aan*

$$\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi \leq \sqrt{Q}.$$

b) Als $S \geq 8061$, dan zijn er maximaal vijf waardes die ξ aan kan nemen en al deze waardes zijn groter of gelijk aan $S - 4\sqrt{S}$.

Bewijs. a) De bovengrens uit het lemma ($\xi \leq \sqrt{Q}$) volgt meteen omdat $\xi^2 \leq Q$ wegens de definitie van ξ. Neem voor de ondergrens ($\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi$) aan dat $\xi \leq \frac{S}{2}$. Dan is $Q = \sum_{i=1}^{k} x_i^2 \leq 2(\frac{S}{2})^2 = \frac{S^2}{2}$. Maar de voorwaarden van het lemma eisen dat $\frac{S^2}{2} < S(S - 6\sqrt{S}) \leq Q$: tegenspraak. Dus geldt

$$\frac{1}{2}S < \xi.$$

Nu, merk op dat $(S-\xi, Q-\xi^2)$ een toelaatbaar paar is. Met Stap 2b volgt dan dat $Q - \xi^2 \leq (S-\xi)^2$. De twee wortels van de onderliggende kwadratische vergelijking zijn $\xi_1 = \frac{1}{2}(S - \sqrt{2Q - S^2})$ en $\xi_2 = \frac{1}{2}(S + \sqrt{2Q - S^2})$, waarbij de ongelijkheid geldt dan en slechts dan als $\xi < \xi_1$ of $\xi > \xi_2$. Omdat $\xi < \xi_1$ in tegenspraak is met $\frac{1}{2}S < \xi$, moet wel gelden dat $\xi > \xi_2 = \frac{1}{2}(S + \sqrt{2Q - S^2})$.

b) Voor dit deel van het bewijs gaan we een schatting geven voor de afstand tussen de onder- en bovengrens uit *Lemma 3a* voor het geval $S \geq 8061$. Neem voor S een vast geheel getal groter of gelijk aan 8061 en bekijk het verschil tussen de boven- en ondergrens in termen van Q:

$$\Delta_S(Q) := \sqrt{Q} - \frac{1}{2}(S + \sqrt{2Q - S^2}).$$

De waarde van Q loopt van $S(S - 6\sqrt{S})$ tot S^2. De eerste afgeleide van $\Delta_S(Q)$ geeft $\frac{1}{2}(1/\sqrt{Q} - 1/\sqrt{2Q - S^2}) < 0$, en daarom is de functie $\Delta_S(Q)$ strikt dalend voor $S(S - 6\sqrt{S}) < Q \leq S^2$. Dus voor vaste S wordt het verschil $\Delta_S(Q)$ gemaximaliseerd voor $Q = S(S - 6\sqrt{S})$ waar het de waarde

$$\Delta^*(S) := \sqrt{S^2 - 6S^{3/2}} - \frac{1}{2}(S + \sqrt{S^2 - 12S^{3/2}})$$

aanneemt. Met standaard calculus is nu te laten zien dat deze functie $\Delta^*(S)$ strikt dalend is voor $S \geq 145$ en dat $\lim_{S\to\infty}\Delta^*(S) = 4\frac{1}{2}$. Hieruit volgt dat voor $S \geq 8061$ en $S(S - 6\sqrt{S}) < Q \leq S^2$ het grootst mogelijke verschil tussen de onder- en bovengrens uit *Lemma 3a* begrensd wordt door

$$\Delta^*(8061) = 4.9999669 < 5.$$

Dit betekent dat er maximaal vijf gehele getallen bestaan tussen deze onder- en bovengrens, oftewel, dat ξ maximaal vijf waardes aan kan nemen. Tot slot merken we op dat $S^2 - 12S^{3/2} > (S - 8\sqrt{S})^2$ geldt voor alle $S \geq 8061$ en dat $\frac{1}{2}(S + \sqrt{2Q - S^2}) > \frac{1}{2}(S + (S - 8\sqrt{S})) = S - 4\sqrt{S}$. Samen met *Lemma 3a* geeft dit $\xi \geq S - 4\sqrt{S}$ en dit completeert het bewijs. □

Aan dit bewijs dienen twee opmerkingen te worden toegevoegd. In het bewijs voor deel a) van het lemma zegt Woeginger dat $Q = \sum_{i=1}^{k} x_i^2 \leq 2(\frac{S}{2})^2 = \frac{S^2}{2}$ geldt, maar hij geeft hier nergens uitleg voor. Dit doet hij niet omdat deze stap in het algemeen gemaakt kan worden met gebruik van een paar handige regels. Het bewijs voor deze stap geven we nu niet, maar we mogen ervan uitgaan dat deze stap geldig is.

Als tweede dient opgemerkt te worden dat Woeginger in het bewijs voor deel a) van het lemma zegt dat $\frac{S^2}{2} < S(S - 6\sqrt{S}) \leq Q$ voortkomt uit de eisen die gesteld worden in het lemma. Nu is het zo dat $S(S - 6\sqrt{S}) \leq Q$ inderdaad geëist wordt, maar $\frac{S^2}{2} < S(S - 6\sqrt{S}$ is hier niet per se waar. Deze ongelijkheid geldt namelijk alleen als $S > 144$. Deze grens maakt voor de rest van het bewijs gelukkig niet uit, waardoor het bewijs blijft kloppen. Het enige dat veranderd is dat Lemma 3 alleen geldt voor $S > 144$. Voor de aanpassing van het algoritme zal dit van belang zijn. Dit komt in de volgende sectie aan bod.

Met het bewijs (en de twee opmerkingen) is goed te begrijpen wat er precies gebeurt in Stap 4 en kan de intuïtieve uitleg van eerder aangevuld worden met

verantwoording. Voor de duidelijkheid halen we Stap 4 er nog een keertje bij.

Als $S \geq 8061$ *en* $S(S - 6\sqrt{S}) < Q \leq S^2$*, vind dan alle gehele getallen* ξ *die voldoen aan* $\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi \leq \sqrt{Q}$*. Los voor elke gevonden* ξ *het geval* $(S - \xi, Q - \xi^2)$ *recursief op. Output YES dan en slechts dan als tenminste één van deze uitkomsten YES is.*

Ten eerste wordt er dus gezocht naar gehele getallen ξ waarvoor geldt $\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi \leq \sqrt{Q}$. Dit bereik waarbinnen ξ gezocht wordt komt voort uit Lemma 3a. Namelijk, stel dat (S, Q) toelaatbaar is, dan is er een certificaat $\langle x_1, ..., x_k \rangle$ voor (S, Q) en dus ook een een ξ zodat $\xi = \max_{1 \leq i \leq k} x_i$. Als (S, Q) dan inderdaad een toelaatbaar paar is, volgt met lemma 2a dat $(S - \xi, Q - \xi^2)$ ook een toelaatbaar paar is. Terug naar het algoritme: voor elke gevonden ξ wordt het paar $(S - \xi, Q - \xi^2)$ als nieuwe input genomen. Dit proces wordt herhaald totdat er één van de volgende twee uitkomsten is. De eerste uitkomst is dat het algoritme door middel van deze recursie een paar (V, W) vindt dat toelaatbaar is. Dan zijn er in het proces positieve gehele getallen $\xi_1, ..., \xi_m$ gevonden zodat $V + \sum_{i=1}^{m} \xi_i = S$ en $W + \sum_{i=1}^{m} \xi_i^2 = Q$ voor bepaalde $m \in \mathbb{N}$. Met het meermaals toepassen van lemma 2b is hieruit af te leiden dat (S, Q) dan ook een toelaatbaar paar is, vandaar dat het algoritme in dit geval output YES geeft. De tweede uitkomst is dat het algoritme geen toelaatbaar paar (V, Q) weet te vinden zoals eerder beschreven en zal het algoritme output NO geven.

De enige vraag die nu nog blijft staan is de vraag of dit een effectieve aanpak is. Het antwoord van Woeginger is ja en dit komt door de afbakening van het aantal opties voor ξ tot 5 én het feit dat $\xi \geq S - 4\sqrt{S}$, wat het resultaat is van Lemma 3b. Woeginger geeft een bewijs voor deze uitspraak. Dankzij dit bewijs is het duidelijk dat het algoritme altijd zal termineren.

Nu bekend is hoe en waarom het algoritme werkt kan er gekeken worden naar opties om Woegingers algoritme te versimpelen. Dat wordt gedaan in de volgende sectie.

2.3 Woegingers algoritme aanpassen

Wat veranderd moet worden aan Woegingers algoritme is de begrenzing $S \leq 8060$ in Stap 1. Na het analyseren van Woegingers algoritme in de vorige sectie is het duidelijk geworden waar deze begrenzing vandaan komt. Voor $S \geq 8061$ is het aantal opties voor ξ namelijk af te bakenen tot 5 in Stap 4, wat belangrijk is voor Woegingers bewijs dat het algoritme altijd zal termineren. Woeginger zegt echter nergens dat de grens $S \leq 8060$ essentieel is voor de terminatie en dat doet vermoeden dat Woeginger de grens voor S puur zo gekozen heeft omdat het mooi uitkwam. Dat blijkt ook zo te zijn. Door gebruik te maken van de grenzen die worden gebruikt in Lemma 3 krijgen we het volgende resultaat.

Lemma 4. *De grens* $S \leq 8060$ *voor* S *in Stap 1 van Woegingers algoritme kan vervangen worden door* $S \leq 144$ *zonder dat dit problemen oplevert voor de*

terminatie van het algoritme.

Bewijs. Woeginger heeft in het bewijs van *Lemma 3b* aangetoond dat de functie $\Delta_S(Q) = \sqrt{Q} - \frac{1}{2}(S + \sqrt{2Q - S^2})$ maximaal is voor $Q = S(S - 6\sqrt{S})$. Het invullen van $Q = S(S - 6\sqrt{S})$ geeft de strikt dalende functie $\Delta^*(S) = \sqrt{S^2 - 6S^{3/2}} - \frac{1}{2}(S + \sqrt{S^2 - 12S^{3/2}})$. Met de functie $\Delta^*(S)$ wordt het aantal opties voor de bepaling van ξ berekend. Uit het bewijs voor *Lemma 3a* halen we dat $S = 145$ de minimale waarde van S is die gebruikt kan worden in de rest van het bewijs van heel *Lemma 3*. (Als $S \leq 144$, dan geldt $\frac{1}{2}S^2 < S(S - 6\sqrt{S})$ niet en wordt er geen tegenspraak gevonden, terwijl die tegenspraak wel nodig is. Voor $S > 144$ geldt deze tegenspraak wel en kan het bewijs verder gaan.) Invullen van $S = 145$ geeft $\Delta^*(145) = 25.9463853 < 26$. Dit betekent dat het algoritme voor $145 \leq S \leq 8060$ meer dan 5 maar hoogstens 26 opties voor ξ langs zal lopen. Voor de terminatie van het algoritme maakt dit niet uit, omdat 26 nog steeds een klein eindig getal is. □

Met dit lemma kan Stap 1 vernieuwd worden:

Stap 1*: als $S \leq 144$, werk dan het probleem uit met de hand. STOP.

Ook al is 144 een heel stuk kleiner dan 8060, nog steeds is er hetzelfde probleem: er zijn erg veel gevallen om met de hand uit te werken. Gelukkig is dit enigszins op te lossen. De methode van het zoeken naar een ξ zoals dat gebeurt in Stap 4 kunnen we hier wederom toepassen, maar dan is een kleine aanpassing van die methode wel nodig. Hier hebben we namelijk geen mooi interval om in te zoeken, omdat S te klein is. Daarom gebruiken we de ietwat lelijke oplossing om simpelweg voor alle positieve gehele getallen $\zeta < S$ de paren $(S - \zeta, Q - \zeta^2)$ bij langs te gaan. Hier moet niet vergeten worden rekening te houden met het handige Lemma 1 en Stap 2. Als namelijk geldt dat $S \leq Q \leq S(S - 6\sqrt{S})$, dat $Q < S$, dat $Q > S^2$ of dat S en Q van verschillende pariteit zijn, dan is al duidelijk of (S, Q) wel of niet toelaatbaar is. Merk tot slot op dat $S \leq S(S - 6\sqrt{S})$ alleen geldt voor $S \geq 38$, dus $S \leq Q \leq S(S - 6\sqrt{S})$ is alleen mogelijk als $S \geq 38$. Daarom wordt de laatste extra toevoeging gedaan om voor $S \leq 37$ alle mogelijke certificaten te maken, op deze manier alle mogelijke toelaatbare paren (S, Q) te vinden en deze in een database Z te stoppen, zodat

$$Z := \{(S, Q) | (S, Q) \text{ is toelaatbaar en } S \leq 37\}.$$

Dit gaat als volgt in zijn werk. Neem een $S \leq 37$. Vind alle partities voor deze S. Een partitie van S is van de vorm $\langle x_1, ..., x_k \rangle$ zodat $\sum_{i=1}^{k} x_i = S$. Door hierbij $Q = \sum_{i=1}^{k} x_i^2$ te kiezen, is $\langle x_1, ..., x_k \rangle$ het certificaat van (S, Q). Zo is vanuit een partitie voor een bepaalde S simpel een certificaat te maken waar een toelaatbaar paar (S, Q) uit voortkomt. Door dit te doen voor elke partitie van elke $S \leq 37$ kan Z geconstrueerd worden. Het blijkt zo te zijn dat $\sum_{n=1}^{37} p(n) = 120769$ het aantal partities is voor $S \leq 37$, maar $\#Z = 5493$. Dat is een redelijk groot verschil en wordt veroorzaakt door het feit dat uit verschillende partities voor S gelijke paren (S, Q) kunnen ontstaan door het gebruik

van bovenstaande methode. Het is duidelijk dat deze dubbele paren (S, Q) vrij vaak ontstaan voor $S \leq 37$. De aantallen zijn overigens gevonden door gebruik van Python, waar dit gemakkelijk in berekend kan worden.

Hiermee hebben we alle ingrediënten die nodig zijn om een algoritme te krijgen dat volledig antwoord geeft op Vraag 2.1. Door al deze bovenstaande resultaten samen te voegen, wordt het algoritme dat we gezocht hebben als volgt.

Algoritme van Zwetsloot

Zij S en Q willekeurige positieve gehele getallen en laat dit de input zijn.

(1) als $S \leq 37$, controleer dan of (S, Q) in de database Z zit. Zo ja, output YES en STOP. Zo nee, output NO en STOP.

(2) als $Q < S$ of als $Q > S^2$ of als S en Q van verschillende pariteit zijn, output NO en STOP.

(3) als $S \leq Q \leq S(S - 6\sqrt{S})$, output YES en STOP.

(4) als $S \leq 144$, los dan voor elke $\zeta < S$ het geval $S - \zeta, Q - \zeta^2)$ recursief op. Output YES en STOP dan en slechts dan als tenminste één van deze uitkomsten YES is. Anders, output NO.

(5) als $S \geq 145$ en $S(S - 6\sqrt{S}) < Q \leq S^2$, vind dan alle gehele getallen ξ die voldoen aan $\frac{1}{2}(S + \sqrt{2Q - S^2}) \leq \xi \leq \sqrt{Q}$. Los voor elke gevonden ξ het geval $(S - \xi, Q - \xi^2)$ recursief op. Output YES dan en slechts dan als tenminste één van deze uitkomsten YES is. Anders, output NO.

2.4 Afsluitende woorden rondom Woeginger

Zwetsloots algoritme geeft volledig antwoord op Vraag 2.1 zoals gewenst, maar kan nog steeds mooier of beter gemaakt worden. Zo zou het handiger zijn om de database Z uit te breiden tot $Z^* := \{(S, Q) | (S, Q)$ is toelaatbaar en $S \leq 144\}$, wat in principe makkelijk gedaan wordt, maar ook veel tijd kost. Er zijn namelijk $\sum_{n=1}^{144} p(n) = 229407862322$ partities voor $S \leq 144$, dat is aanzienlijk meer dan 120769 voor $S \leq 37$. Tegelijkertijd kost het ook veel tijd om stap 4 van Zwetsloots algoritme uit te moeten voeren. Een ander punt van kritiek is dat het mooier is om te werken zonder databases en dat alles wordt aangetoond aan de hand van lemma's, zoals Woeginger dat deed voor $S \geq 8061$. Het is de vraag of deze lemma's gevonden kunnen worden, maar in ieder geval is het duidelijk dat het algoritme mooier kan. Tegelijkertijd is het ook duidelijk dat Zwetsloots algoritme een volledig antwoord geeft op Vraag 2.1 en in die zin een goede verbetering in ten opzichte van Woegingers algoritme.

Als laatste is het interessant op te merken dat verschillende oplossingen in het Euclidische vlak met de aanpak van Woeginger kennelijk weergegeven kunnen

worden door hetzelfde paar (S, Q). We zagen immers dat verschillende partities voor S gelijke paren (S, Q) kunnen opleveren, terwijl verschillende partities verschillende certificaten vormen en daarmee horen bij verschillende oplossingen. Voor Vraag 2.1 maakt dit niet uit, maar als we op zoek willen gaan naar verschillende oplossingen moeten we kennelijk met meer precisie te werk gaan.

Hoofdstuk 3

Gerelateerde 17 lijn problemen

In dit hoofdstuk zal de focus liggen op Probleem 3: Fouriers 17 lijnen probleem zonder restricties. Voor het gemak herhalen we dit probleem nog eens.

Probleem 3
Laten n het aantal lijnen en m het aantal punten zijn. Bestaan er positieve gehele getallen μ_1,...,μ_k en positieve gehele getallen $\lambda_1, ..., \lambda_m$ groter of gelijk aan twee zodat geldt (3.1) geldt? $$m = \binom{n}{2} - \sum_{i=1}^{m}[\binom{\lambda_i}{2} - 1] - \sum_{j=1}^{k}\binom{\mu_j}{2} \quad (3.1)$$ Hier zijn $\mu_1, ..., \mu_k$ het aantal lijnen per familie van parallelle lijnen en $\lambda_1, ..., \lambda_m$ de ordes van de punten $p_1, ..., p_m$. Oplossingen zijn van de vorm $[\langle\lambda_1, ..., \lambda_m\rangle,\langle\mu_1, ..., \mu_k\rangle]$.

Bij dit probleem komen meer aspecten kijken dan bij de Generalisatie van Fouriers 17 lijnen probleem, zoals de munten. Hierdoor is het ingewikkelder om een algemeen algoritme te geven zoals in Hoofdstuk 2. Daarom pakken we Probleem 3 anders aan dan Probleem 1. Het hele hoofdstuk wordt uitgegaan van vaste n,$m \in \mathbb{N}$, waarbij we voor de vorm het meest ingaan op het geval $n = 17$ en $m = 101$, de getallen uit Fouriers 17 lijnen probleem. De volgende drie vragen zullen centraal staan.

Vraag 3.1
Zij $n, m \in \mathbb{N}$. Zijn er positieve gehele getallen $\mu_1, ..., \mu_k$ en positieve gehele getallen $\lambda_1, ..., \lambda_m$ groter of gelijk aan twee zodat geldt $m = \binom{n}{2} - \sum_{i=1}^{m}[\binom{\lambda_i}{2} - 1] - \sum_{j=1}^{k}\binom{\mu_j}{2}$?

Vraag 3.2
Hoeveel oplossingen zijn er voor Fouriers 17 lijnen probleem zonder restricties met 17 lijnen en 101 punten en wat zijn deze oplossingen?

Vraag 3.3
Hoe kunnen voor vaste n,$m \in \mathbb{N}$ alle oplossingen voor Fouriers 17 lijnen probleem zonder restricties gevonden worden?

Vraag 3.1 is hetzelfde als Vraag 2.1, maar dan voor Probleem 3 in plaats van voor Probleem 1. Deze vraag zal helpen met het begrijpen waarom er hier een andere aanpak moet worden gebruikt. Verder lijken vragen 3.2 en 3.3 erg op elkaar. Je zou kunnen zeggen dat antwoord geven op vraag 3.2 ook meteen een antwoord geeft op vraag 3.1. Door dit in detail te bekijken blijkt het toch complexer te liggen wat zal blijken uit de volgende secties. Voordat we aan het beantwoorden van deze vragen beginnen, volgt er eerst een korte toelichting op de notatie voor dit hoofdstuk.

3.1 Korte toelichting op notatie

Als we kijken naar vergelijking (3.1) vallen wat zaken op. Als eerste merken we op dat elke i met $\lambda_i = 2$ een term gelijk aan 0 oplevert in de som. Omdat het fijn is een zo kort mogelijke notatie te hebben voor oplossingen, wordt ervoor gekozen om alleen ordes groter of gelijk aan 3 mee te nemen in de notatie. Dit zal geen verandering aanbrengen in de sommering, aangezien we alleen termen gelijk aan 0 weglaten. Aangezien we weten dat er m snijpunten in totaal moeten zijn, weten we met de nieuwe notatie nog steeds hoeveel snijpunten er zijn van orde 2. Dit verandert voor vergelijking (3.1) alleen dat we de ordes van de snijpunten niet meer sommeren over m maar over een $l \leq m$. l is dan het aantal munten, punten waar meer dan twee lijnen doorheen gaan.

Als tweede merken we op dat in vergelijking (3.1) moet gelden dat $\mu_i \geq 2$, omdat we anders kans hebben op termen als $\binom{1}{2}$ wat gedefinieerd is als 0. Daarom worden in de notatie alleen de ordes van parallelle families meegenomen die bestaan uit meer dan één lijn. Dit levert geen problemen op voor de sommering, omdat we wederom alleen termen gelijk aan 0 weglaten. Omdat bekend is dat er n lijnen in totaal moeten zijn, kunnen we ook hier uit de nieuwe notatie afleiden hoeveel niet-parallelle lijnen er precies zijn. In vergelijking (3.1) zou strikt genomen de sommering niet meer over alle k parallelle families mogen gaan, omdat er een goede kans is op een losse niet-parallelle lijn, maar dat laten we nu achterwege. Het getal k beschouwen we nu als het aantal families van parallelle lijnen met minstens twee lijnen. Het totale aantal families kan dan waar nodig nog steeds bepaald worden met gebruik van deze k.
Om tot slot de notatie nog wat logischer en overzichtelijker te maken wordt geëist dat $\lambda_i \leq \lambda_{i+1}$ en $\mu_j \leq \mu_{j+1}$ voor alle i, j, zodat λ_1 de grootste orde van de munten is en μ_1 de grootste orde van de parallelle families.

Samenvattend: als we het vanaf nu hebben over berekeningen aan de ordes

of over een oplossing $[\langle\lambda_1, ..., \lambda_m\rangle, \langle\mu_1, ..., \mu_k\rangle]$ zullen we er automatisch vanuit gaan dat moet gelden $\lambda_i \geq 3$, $\mu_j \geq 2$, $\lambda_i \leq \lambda_{i+1}$ en $\mu_j \leq \mu_{j+1}$ voor alle i, j. k is nu het aantal families van parallelle lijnen met minstens twee lijnen en l is het aantal munten.

3.2 Vraag 3.1

Vraag 3.1
Zij $n, m \in \mathbb{N}$. Zijn er positieve gehele getallen $\mu_1, ..., \mu_k$ en positieve gehele getallen $\lambda_1, ..., \lambda_m$ groter of gelijk aan twee zodat geldt $m = \binom{n}{2} - \sum_{i=1}^{m}[\binom{\lambda_i}{2} - 1] - \sum_{j=1}^{k}\binom{\mu_j}{2}$?

Aan deze vraag worden niet te veel woorden gewijd vanwege de simpele conclusie die hier getrokken gaat worden. We zagen dat in hoofdstuk 2 antwoord te geven was op een soortgelijke vraag door gebruik te maken van het algoritme van Woeginger. Voor Probleem 3 is er voorlopig nog niet een dergelijk algoritme gevonden en ook geen andere methode om hier in het algemeen antwoord op te geven. Wel zijn er net als bij Probleem 1 bepaalde grenzen aan te geven waarbinnen n, m sowieso moeten vallen, wat in de volgende sectie beter aan bod zal komen. Eigenlijk is er niet op een simpele manier een antwoord te geven op deze vraag, behalve door simpelweg op zoek te gaan naar deze positieve gehele getallen $\mu_1, ..., \mu_k$ en positieve gehele getallen $\lambda_1, ..., \lambda_m$ groter of gelijk aan twee. Doordat er enige begrenzing is zal je uiteindelijk uitkomen op een antwoord, maar dit is wel uitermate inefficiënt.

Als Vraag 3.1 zo'n onhandig maar makkelijk antwoord heeft, waarom stellen we de vraag dan? Dat is omdat het logisch gezien in het hele plaatje nodig is. Het is raar om op zoek te gaan naar oplossingen als je niet zeker weet of er wel een oplossing is. Dit is namelijk niet logisch gezien we op zoek zijn naar enige efficiëntie bij het oplossen van dit probleem. Toch zullen we zo te werk gaan, maar gelukkig niet zonder resultaat. De volgende secties zullen meer duidelijkheid geven.

3.3 Vraag 3.2

Vraag 3.2
Hoeveel oplossingen zijn er voor Fouriers 17 lijnen probleem zonder restricties met 17 lijnen en 101 punten en wat zijn deze oplossingen?

Voor het beantwoorden van vraag 3.2 willen we alle oplossingen vinden voor Fouriers 17 lijnen probleem zonder restricties met 17 lijnen en 101 punten. Dit komt volgens de probleemomschrijving en extra toelichting overeen met het vinden van alle positieve gehele getallen $\lambda_1, ..., \lambda_l$ en $\mu_1, ..., \mu_k$ met $\lambda_i \geq 3$ en $\mu_j \geq 2$ voor alle i, j zodat $101 = \binom{17}{2} - \sum_{i=1}^{l}[\binom{\lambda_i}{2} - 1] - \sum_{j=1}^{k}\binom{\mu_j}{2} = 136 -$

$\sum_{i=1}^{l}[\binom{\lambda_i}{2} - 1] - \sum_{j=1}^{k}\binom{\mu_j}{2}$, dus zodat

$$35 = \sum_{i=1}^{l}[\binom{\lambda_i}{2} - 1] + \sum_{j=1}^{k}\binom{\mu_j}{2}$$

Het vinden van al deze $\lambda_1, ..., \lambda_l$ en $\mu_1, ..., \mu_k$ is al eens eerder gedaan, namelijk door Daniel Lichtblau en Wacharin Wichiramala [**2**] en Gerald L. Alexanderson en John E. Wetzel [**3**]. Zij gebruikten de volgende aanpak.

Ten eerste maken ze onderscheid tussen de soorten oplossingen van Fouriers 17 lijnen probleem zonder restricties. Het is logisch in te zien dat de oplossingen van Probleem 1 en Probleem 2 ook oplossingen zijn van Probleem 3. Probleem 1 is immers Probleem 3 waarbij voor elke i geldt $\lambda_i = 2$, oftewel, Probleem 3 zonder munten is equivalent aan Probleem 1. Net zo is Probleem 2 gelijk aan Probleem 3 waarbij voor elke j geldt $\mu_j = 1$, dus dat er geen lijnen parallel aan elkaar zijn. Deze twee gevallen vormen dan ook de eerste twee soorten oplossingen voor Probleem 3. Als derde soort worden oplossingen genomen waarvoor geldt dat er minstens één i is zodat $\lambda_i \geq 3$ en minstens één j zodat $\mu_j \geq 2$. Deze drie soorten samen geven dan alle oplossingen op Probleem 3. Voor Probleem 1 met 17 lijnen en 101 punten is al bekend dat er precies 4 oplossingen zijn. Voor Probleem 2 blijken dat er 20 te zijn [**3**]. Nu willen we de focus leggen op het derde geval waar het aantal oplossingen voor bepaald moet worden.

Om het aantal oplossingen met minstens één familie van meerdere parallelle lijnen en minstens één munt te bepalen, bekijken we eerst de grenzen voor alle getallen. De ondergrens voor de ordes van de munten is 3 en de ondergrens van de families parallelle lijnen is 2. De bovengrenzen kunnen gehaald worden uit vergelijking (3.1) en blijken 8 te zijn voor zowel de munten als de families van parallelle lijnen. Er geldt immers $\binom{9}{2} - 1 = 35$, dan zouden er geen families van parallelle lijnen meer kunnen zijn, en er geldt $\binom{9}{2} = 36$, wat groter is dan 35 en dus ook niet kan volgens vergelijking (3.1). Nu hebben we dus

$$3 \leq \lambda_i \leq 8 \text{ en } 2 \leq \mu_j \leq 8 \text{ voor alle } i, j$$

Deze begrenzing geeft ons de mogelijke waardes 2, 5, 9, 14, 20 en 27 voor $\binom{\lambda_i}{2} - 1$ en de mogelijke waardes 1, 3, 6, 10, 15, 21 en 28 voor $\binom{\mu_j}{2}$.

Als tweede krijgen we de volgende logische eis waaraan voldaan moet zijn:

$$\sum_{j=1}^{k} \mu_j \leq 17$$

Het aantal lijnen dat over de parallelle families verdeeld is moet onder de 17 blijven, zodat het met niet-parallelle lijnen kan worden aangevuld tot 17. Gaat het over de 17 heen, dan hebben we kennelijk meer dan 17 lijnen en zitten we in een totaal ander geval te werken.

Tot slot herkennen we de eis dat er per specifiek geval een extra maximum is voor de orde van de munten. Deze eis wordt gegeven door

$$\lambda_1 \leq k + (17 - \sum_{j=1}^{k} \mu_j)$$

met k, het aantal families van parallelle lijnen, en $17 - \sum_{j=1}^{k} \mu_j$, het aantal niet-parallelle lijnen. Dit is in te zien met het logische feit dat parallelle lijnen elkaar niet snijden.

Nu al deze grenzen vastgesteld zijn kan het doel beter uitgelegd worden: het vinden van alle partities van 35 waarbij gebruikt gemaakt wordt van de getallen 2, 5, 9, 14, 20, 27 en 1, 3, 6, 10, 15, 21, 28 én waarbij voldaan wordt aan de andere eisen. Uit een gevonden partitie kan een bepaalde oplossing $[\langle\lambda_1, ..., \lambda_l\rangle, \langle\mu_1, ..., \mu_k\rangle]$ worden afgeleid. Hoe deze partities precies bepaald kunnen worden is precies te lezen in het artikel van Lichtblau en Wichiramala [**2**]. Er blijken precies 900 van deze partities te zijn. Met de informatie die we al hadden geeft dat in totaal 900+20+4=924 oplossingen voor Fouriers 17 lijnen probleem zonder restricties van 17 lijnen en 101 punten.

Is hiermee dan volledig antwoord gegeven op vraag 3.2? Numeriek gezien wel: alle mogelijke partities met gegeven eisen zijn gevonden. Maar er valt nog meer over te zeggen. Tijdens de bepaling van de 900 oplossingen door Lichtblau, Wichiramala, Alexanderson en Wetzel kwamen ze in eerste instantie uit op 901 oplossingen. Ze waren namelijk vergeten $\lambda_1 \leq k + (17 - \sum_{j=1}^{k} \mu_j)$ als eis toe te voegen aan hun programma, wat een oplossing gaf met een munt van hogere orde dan lijnen beschikbaar waren. Numeriek gezien was er inderdaad een oplossing gevonden, maar het tekenen van die oplossing zou laten zien dat dat niet zou kloppen. Gelukkig werd deze fout op een gegeven moment ontdekt, maar je kan je afvragen of alle andere 900 opties dan wel correct zijn. Er is namelijk nergens te vinden dat iemand ooit geprobeerd heeft deze 900 opties na te tekenen in het echte vlak én er is een gebrek aan een wiskundig bewijs van de werkelijke correctheid van deze numerieke oplossingen. Hoe kunnen we dan zeker weten dat we hiermee volledig antwoord hebben gegeven? Zouden er wellicht nog meer verborgen eisen zijn? Volgens Alexander en Wetzel zijn op deze wijze in ieder geval voor de situatie $n = 17$ en $m = 101$ alle oplossingen gevonden, maar perfecte argumentatie geven ze hier niet voor. Bij het beantwoorden van vraag 3.3 zullen we iets dieper ingaan op dit probleem.

3.4 Vraag 3.3

Vraag 3.3
Hoe kunnen voor vaste n,$m \in \mathbb{N}$ alle oplossingen voor Fouriers 17 lijnen probleem zonder restricties gevonden worden?

Een heel concreet antwoord op deze vraag zullen we niet geven. In plaats daarvan zullen we drie omschrijvingen geven van methodes hoe je dit aan zou kunnen pakken. Bij één van deze methodes gebruiken we de methode die bij vraag 3.2 wordt gebruikt als houvast. De reden dat er drie methodes gegeven worden, is omdat er nog geen duidelijke beste methode is. Elke methode heeft duidelijk zo zijn eigen voor- en nadelen. Later zal er meer tijd besteed moeten worden om tot een daadwerkelijke methode te komen die altijd een goed antwoord kan geven op vraag 3.3. Voor nu zullen we de drie methodes los van elkaar omschrijven en enkele voor- en nadelen noemen per methode.

Methode 1: genereer algoritme
Bij de eerste methode wordt er gebruik gemaakt van een genereer algoritme. Het doel van dit algoritme is (logischerwijs) om alle mogelijke configuraties voor bepaalde n te genereren. Uit al deze mogelijke configuraties met n lijnen kunnen dan precies de configuraties met m snijpunten worden gehaald. In die zin wordt er een database opgebouwd van alle mogelijke configuraties. Op die manier geeft dit algoritme niet zo zeer een concreet antwoord op vraag 3.2 voor het gaat om vaste n en m, maar kan dit algoritme voor alle $n, m \in \mathbb{N}$ een antwoord geven, indien het algoritme maar lang genoeg gedraaid heeft. Dit klinkt allemaal leuk en aardig, maar hoe moet dit algoritme nou te werk gaan?

Het generen gebeurt vanuit het lege vlak, het vlak zonder lijnen en snijpunten, door telkens een nieuwe lijn toe te voegen totdat er n lijnen zijn. Noem voor het gemak een willekeurige configuratie met n lijnen een n-configuratie. Dan is de bedoeling dat het algoritme aan een reeds gegenereerde n-configuratie op elke mogelijke manier een lijn toevoegt. Stel dat er p manieren zijn om dit te doen, dan krijgen we p verschillende $(n+1)$-configuraties vanuit de n-configuratie in kwestie. Het gaat hier om *mogelijke manieren* omdat het toevoegen van een lijn simpelweg op meerdere manieren kan. Zo kan je een lijn toevoegen aan een bepaalde parallelle familie, zo kan je een lijn laten snijden met al bestaande snijpunten om een munt van hogere orde te krijgen, enz. Er zijn veel keuzes die gemaakt moeten worden. Daarom is het nodig dat dit algoritme gebruik gaat maken van een stappenplan.

Het stappenplan is het belangrijkste gedeelte van het algoritme, aangezien hier het meeste rekenwerk wordt gedaan. Daarom is het belangrijk dat een dergelijk stappenplan extreem duidelijk verwoord moet worden en gaten laat voor eventuele fouten. De optimale vorm van dit stappenplan is nog niet duidelijk. Een voorbeeld hoe dit aangepakt kan worden is om het stappenplan het uiterlijk van een soort vragenlijst te maken, zoals hieronder. Let op dat dit een zeer simplistisch voorbeeld is.

Voorbeeld stappenplan

Er wordt een $(n+1)^e$ lijn toegevoegd aan een bepaalde n-configuratie.

1. Wordt deze lijn toegevoegd aan een parallelle familie? Zoja, kies een familie die nog niet aan bod is gekomen.

2. Gaat deze lijn door snijpunten die al bestaan of niet? Zoja, kies punten die voor de familie in kwestie nog niet aan bod zijn gekomen.

3. Sla de verkregen configuratie op als $(n+1)$-configuratie.

4. Zijn alle families en punten al eens aan bod gekomen? Zo nee, doorloop het stappenplan nogmaals.

5. Zijn alle n-configuraties aan bod gekomen? Zo nee, kies een n-configuratie die nog niet aan bod is gekomen en doorloop het stappenplan nogmaals. Zo ja, kies een $(n+1)$-configuratie en doorloop het stappenplan voor het toevoegen van een $(n+2)^2$ lijn aan een $(n+1)$-configuratie.

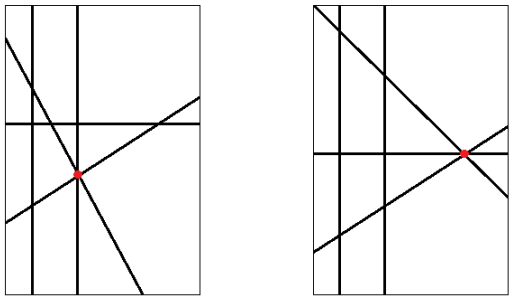
Naar alle waarschijnlijkheid zijn er meer vragen nodig en zijn er heel erg veel details waarmee rekening gehouden moet worden. Zo mag een nieuwe lijn niet door twee punten gaan die ook al op een andere lijn liggen, dan bestond de nieuwe lijn namelijk al. Dit geeft een schets van hoe belangrijk en complex dit gedeelte zal zijn voor het algoritme.

Naast dit stappenplan is het ook belangrijk hoe het algoritme alle gegevens bijhoudt en welke gegevens dit dan precies moeten zijn. Het is duidelijk dat het aantal lijnen, het aantal snijpunten, de ordes van de families van parallelle lijnen en de ordes van de munten in ieder geval ergens opgeslagen moeten worden. Een optie die (zeer waarschijnlijk) werkt, is het definiëren van elke lijn door een lineaire functie, bijvoorbeeld $y_i = ax + b$ met y_i de i^e lijn die wordt toegevoegd. Dan worden de families van parallelle groepen gedefinieerd door de a, de snijpunten worden gevonden door alle snijpunten tussen $y_1, ..., y_n$ te berekenen en de ordes van de snijpunten worden gevonden door slim te tellen hoe vaak elk snijpunt gevonden is. Het voordeel van deze manier is dat alle nodige gegevens bekend zijn en het redelijk duidelijk is hoe de configuratie op papier getekend kan worden. Het nadeel is dat er zeer veel informatie wordt bijgehouden, waardoor het algoritme elke keer lang de tijd nodig zal hebben om met nieuwe configuraties te komen.

Een andere manier om de gegevens bij te houden is door elke lijn te nummeren, door elke familie van parallelle lijnen een verzameling van genummerde lijnen te laten zijn en door van elk snijpunt de specifieke lijnen te noemen die er doorheen gaan. Op deze manier wordt de noodzakelijke informatie bijgehouden en is er geen sprake van overbodige informatie. Het probleem is wel dat het lastiger is te controleren of er allemaal kloppende configuraties gegenereerd zullen worden. Zo kunnen er wellicht configuraties ontstaan met een kromme lijn en het is niet meteen duidelijk vanuit de bijgehouden informatie of dat wel of niet het geval is. In principe moet dit voorkomen kunnen worden door het correct definiëren van het stappenplan. Het kan ook voorkomen worden door een correct wiskundig bewijs voor deze aanpak. Feit blijft dat deze aanpak niet

volledig vrij lijkt van dergelijke complicaties.

Een laatste detail waar we bij stil moeten staan is de vraag wanneer configuraties gelijk aan elkaar zijn. We willen namelijk niet dat het algoritme allerlei configuraties op gaat slaan als *verschillend* terwijl het eigenlijk allemaal gelijke configuraties zijn. Als het duidelijk is wanneer configuraties verschillend zijn, kan er gemakkelijk een stap aan het stappenplan worden toegevoegd die ervoor zorgt dat er geen dubbele configuraties opgeslagen worden. Waarom is het dan lastig dit onderscheid tussen configuraties te zien? Als voorbeeld bekijken we het geval met certificaat $[\langle 3\rangle,\langle 2,1^3\rangle]$ in de afbeelding hieronder. Het is duidelijk dat het certificaat voor beide oplossingen hetzelfde is. Aan de andere kant, wanneer we elke lijn nummeren en we op die manier de notatie willen weergeven, krijgen we twee verschillende notaties. Zijn deze configuraties daarom dan ook verschillend of blijven ze hetzelfde? Dit is een vraag waar voorlopig niemand zich duidelijk over heeft uitgesproken, waar het is het wel waard deze kwestie beter te bekijken.



Al met al ziet deze methode eruit alsof het veel tijd kost de berekeningen te doen, er wordt immers een gigantische database gemaakt, maar lijkt het alsof je met genoeg precisie ook genoeg zekerheid krijgt om te kunnen zeggen dat er werkelijk juiste configuraties gevonden kunnen worden.

Methode 2: verifieer algoritme

Deze methode maakt gebruik van de oplossing die werd gegeven voor vraag 3.2. Het doel is om voor vaste n en m alle numerieke oplossingen oftewel certificaten te genereren, zoals ook werd gedaan voor vraag 3.2, en een algoritme te geven om deze certificaten te verifiëren, dus een algoritme dat kan zeggen of een bepaald certificaat klopt en dat eventueel een afbeelding van de configuratie kan geven.

Het genereren van de numerieke oplossingen zou soepel moeten gaan, we weten immers al hoe dit moet. Het is ingewikkeld precies te zeggen hoe een dergelijk algoritme te werk zou moeten gaan om deze oplossingen te controleren op juistheid. Dit algoritme zou namelijk een wiskundig bewijs moeten vormen. Anders gezegd, als er een wiskundig bewijs gegeven wordt voor wanneer een numerieke oplossing wel of niet juist is, hebben we een verifieer algoritme. Voor nu valt er verder weinig te zeggen over deze methode, behalve dan dat

er bepaalde zaken zijn waar sowieso rekening mee gehouden moet worden. De meest opvallende van deze zaken is het probleem dat we ook al tegenkwamen bij methode 1: uit één certificaat kunnen meerdere configuraties getekend worden. Ook hier is het dus belangrijk dat de vraag beantwoord wordt wanneer configuraties verschillend of gelijk zijn.

Wat zouden de voordelen zijn van deze methode? Dat is simpel. Zodra er een verifieer algoritme is, kost het naar alle waarschijnlijkheid weinig tijd om dit algoritme te doorlopen. Het genereren van de numerieke oplossingen kost weliswaar wat meer tijd, maar het hele proces wordt gemakkelijker te volgen en zal ons een duidelijk antwoord geven. Het grote nadeel is natuurlijk dat er een wiskundig bewijs moet worden gevonden zodat in het algemeen met zekerheid dingen gezegd kunnen worden over de numerieke oplossingen. Het vinden van dit bewijs kan nog behoorlijk ingewikkeld zijn.

Methode 3: ombouw algoritme

Voor de laatste methode kijken we weer even naar hoofdstuk 2. Voor vaste n en m' kunnen we $S = n$ en $Q = n^2 - 2m'$ definiëren en bekijken of (S, Q) al dan niet toelaatbaar is. Is (S, Q) toelaatbaar, dan bestaat er een configuratie zonder munten met n lijnen en m snijpunten. Stel dat we een dergelijke configuratie met n lijnen en m' snijpunten hebben, dan kunnen we het ombouw algoritme erop loslaten. Het idee van het ombouw algoritme is dat het een configuratie bestaande uit n lijnen die verdeeld zijn over een aantal parallelle families kan ombouwen tot een configuratie waarin munten wel zijn toegestaan door te schuiven met lijnen en snijpunten. Merk meteen op dat het aantal lijnen gelijk zal blijven en alleen het aantal snijpunten kan veranderen. De vaste m' die we hebben genomen om Q mee te berekenen is dan ook niet de m waar naar gevraagd wordt in vraag 3.3. Dit schuiven met lijnen en snijpunten kan gebeuren totdat er m punten zijn.

Intuïtief is deze aanpak vrij duidelijk, maar zoals gebruikelijk komt er meer bij kijken. Deze methode begint met het controleren voor de vaste n en bepaalde m' of er een oplossing is voor Probleem 1. De m' die we hier kiezen moet groter zijn dan de vaste m waar we uiteindelijk het antwoord voor willen hebben. Wat de betekenis is van het verschil tussen m en m' komt later aan bod. Stel nu dat we voor n en m' inderdaad toelaatbaar paar (S, Q) vinden. Dan is er dus een configuratie. Die configuratie moet dan nog wel gevonden worden, maar dat zou niet heel erg moeilijk moeten zijn. De makkelijkste manier om dit te doen is, waarschijnlijk, door de aanpak van vraag 3.2 te gebruiken waar we als extra eis aan toevoegen dat $\lambda_i = 2$ voor alle i. Dit geeft een certificaat voor een configuratie met n lijnen, m' snijpunten, k families van parallelle lijnen en 0 munten. Hier kan eigenlijk niets fout gaan. (Merk op dat een numerieke oplossing voor Probleem 1 altijd een oplossing in het Euclidische vlak geeft, doordat families van parallelle lijnen zonder munten eenduidig te tekenen zijn.)

Dan komt de volgende stap: het ombouwen van deze configuratie tot een configuratie waar wel munten zijn toegestaan. Hier zijn twee opties. De eerste optie is dat je een lijn gaat verschuiven, waarmee de snijpunten van die lijn ook zullen veranderen en verschuiven. Hier moet dan nog een keuze worden

gemaakt of je een niet-parallelle lijn verschuift of een lijn die wel parallel is aan andere lijnen. Dit zal namelijk verschillende resultaten opleveren. Zolang de niet-parallelle lijn niet-parallel blijft, maar het aantal snijpunten wel verandert, zal het aantal snijpunten sowieso afnemen. Wanneer een parallelle lijn door verschuiven niet-parallel wordt, zal deze lijn ook met alle lijnen snijden die daarvoor nog parallel aan de lijn waren, wat juist nieuwe snijpunten op zal leveren. De tweede optie is dat je een snijpunt gaat verschuiven, waarbij de lijnen die door die snijpunt mee bewegen en dus ook verschoven worden. Wederom moet hier rekening gehouden worden met de oriëntatie van de lijnen. In beide gevallen is het duidelijk dat er een nieuwe configuratie tevoorschijn zal komen. Dit proces wordt herhaald totdat de juiste oplossing gevonden is of totdat het uitgesloten kan worden dat er een oplossing is.

Een voordeel van deze methode is dat er gebruik gemaakt wordt van kennis die er al is, namelijk Woegingers algoritme. Op deze manier is hier snel een goede uitgangspositie gevonden. Ook worden op deze manier snel en gemakkelijk allerlei verschillende nieuwe configuraties gevonden. Het grote nadeel is dat het waarschijnlijk veel tijd kan kosten om een juiste oplossing te vinden en wellicht nog meer tijd om het bestaan van een oplossing uit te sluiten. Mocht het algoritme zodanig gemaakt kunnen worden dat dit enigszins efficiënt kan worden gedaan, dan lijkt dit een erg goede methode te zijn.

3.5 Afsluitende woorden

Terugkijkend op de vragen van hoofdstuk 3 is ten eerste te concluderen dat er meer ingewikkelde zaken bij komen kijken dan men intuïtief van te voren denkt. Het belangrijkste hiervan is dat de connectie tussen numerieke oplossing en daadwerkelijk oplossing tot op heden vaag is gebleven. Het is duidelijk dat het onpraktisch is om voor bijvoorbeeld 1000 lijnen en 4000 snijpunten alle mogelijke opties met de hand uit te gaan tekenen om een zeker bewijs te hebben voor de (on)juistheid van een oplossing, dus willen we de numerieke aanpak graag blijven gebruiken. Om dit te mogen doen moet er dus nog wel het een en ander gebeuren, zoals te zien is uit de omschrijvingen van methode 1, 2 en 3.

Een grote uitdaging die nog niet genoemd is, is het vertalen van omschrijvingen van de methodes naar daadwerkelijke algoritmes of programma's die perfect werken. Als voorbeeld nemen we methode 3 waarmee we al een beetje geëxperimenteerd hebben. Eén van de resultaten bevatte een configuratie met een snijpunt dat zodanig verschoven was dat er een kromme lijn ontstond, terwijl dat natuurlijk niet de bedoeling is. Het vinden van de juiste code kan zorgen voor veel onverwachte problemen. Het is dus waard om te benadrukken dat het een grote klus gaat zijn dit in precisie te programmeren.

Het ultieme doel is natuurlijk om een mooie en werkende methode te krijgen om vraag 3.2 mee op te lossen. Dit lijkt niet onmogelijk, er is simpelweg nog meer tijd en nadenkwerk nodig. Het is dan ook aan te moedigen dat er hier verder naar gekeken moet worden om dit doel te verwerkelijken.

Referenties

1. Gerhard J. Woeginger, Seventeen lines and one-hundred-and-one points, *Theoretical Computer Science* **321** (2004) 415-421.

2. Daniel Lichtblau and Wacharin Wichiramala, Perplexed Computations, `http://www.maa.org/pubs/mm_supplements/index.html`

3. Gerald L. Alexanderson and John E. Wetzel, Perplexities Related to Fourier's 17 Line Probleem, *Mathematics Magazine* **85** (2012) 3-12.