BSc verslag TECHNISCHE WISKUNDE

"Geoptimaliseerde Chaos"
(Engelse titel: "Optimized Chaos")

Teun Michiel Louis Janssen

Technische Universiteit Delft

Begeleider

Dr. R.J. Fokkink

Overige commissieleden

Dr.ir. L.E. Meester

Dr. J.L.A. Dubbeldam

Dr.ir. M. Keijzer

Januari, 2012

Delft

Inhoudsopgave

Inleiding

Hoewel James Yorke niet de eerste was die het verschijnsel wat we nu kennen als chaos waarnam, was hij wel degene die het de naam chaos gaf. Yorke kreeg in 1972 het artikel "Deterministic Nonperiodic flow"van een collega. In dit stuk beschreef de meteoroloog Edward Lorenz een simpel systeem van vergelijkingen, gebaseerd op vloeistof dynamica met maar drie variabelen. Echter het gedrag van het systeem was niet te voorspellen door zijn sterke afhankelijkheid van de begincondities. De wetenschap had tot op dat moment eigenlijk geleerd deze chaos niet te zien. Men negeerde dit ongecontroleerde gedrag. Yorke zag in dit artikel de chaos, die tot op dat moment wiskundig onmogelijk werd geacht.

Yorke: "De eerste boodschap is dat wanorde bestaat. Wis- en natuurkundigen willen regelmaat ontdekken. De mensen zeggen: wat voor nut heeft wanorde. Maar de mensen moeten iets van die wanorde weten, willen ze er iets aan kunnen doen. Een automonteur die niets weet van olieklonters in de kleppen, is geen goede mecanicien."[6]

Vanuit deze gedachte zullen we ook gaan kijken naar het primal-dual Dikin-type algoritme. Dit is een algoritme waarmee een oplossing voor een lineair programmeerprobleem berekend kan worden. We zullen ons bezighouden met de vraag: vertoont het primal-dual Dikin-type algoritme chaotisch gedrag?
Hiervoor zullen we eerst gaan bekijken wat chaos in wiskundig opzicht inhoudt. Daarna zullen we naar een vergelijkbaar algoritme kijken, waarvan we weten dat deze chaos vertoont. Tot slot kijken we of we heuristisch chaos kunnen vinden in het primal-dual Dikin-type algoritme.

Chronologisch is dit project anders verlopen. Oorspronkelijk was de bedoeling om chaos in het Dikin proces, dat is voorgesteld door Kees Roos, te bestuderen. In dit proces bleek bij vaste parameterwaarden altijd een punt van periode 3 op te treden. Dit was het startpunt van dit project. Vervolgens was de vraag wat de consequentie is van chaos in het Dikin proces voor het primal-dual Dikin-type algoritme. De volgende stap was onderzoek naar dynamica van dergelijke algoritmen.

1 Yorke's periode drie

We zullen eerst gaan kijken naar een artikel geschreven door Yorke zelf. Dit artikel, "Period three implies Chaos", wordt gezien als een klassieker op het gebied van de chaostheorie. In dit artikel bewijst hij dat als een dynamisch systeem een element met periode 3 bevat, dat het dan elementen bevat van alle periodes bevat en dus chaotisch gedrag vertoont.

Een dynamisch systeem is een wiskundige formalisatie van een tijdsvariabel system. Hierbij wordt het gedrag van het systeem volledig bepaald door toestanden waarin het systeem in het verleden is geweest en de directe acties die de omgeving op dit systeem uitoefent. Het aantal toestanden uit het verleden, dat in de vergelijking gebruikt wordt om z'n toekomstige toestand te bepalen, wordt de orde van het dynamisch systeem genoemd. Als een dynamisch systeem bijvoorbeeld alleen de laatste toestand gebruikt is dit een systeem van orde één. Yorke gebruikt in z'n artikel als voorbeeld een discreet dynamisch systeem van orde een. De vraag waar York en ook wij ons mee bezig zullen houden is: gegeven een functie f en een beginwaarde x_0, wat gebeurt er uiteindelijk met de volgende rij iteraties

$$x_0, f(x_0), f(f(x_0)), f(f(f(x_0))), \ldots$$

In zijn artikel bekijkt Yorke het limietgedrag van een ecologisch model.

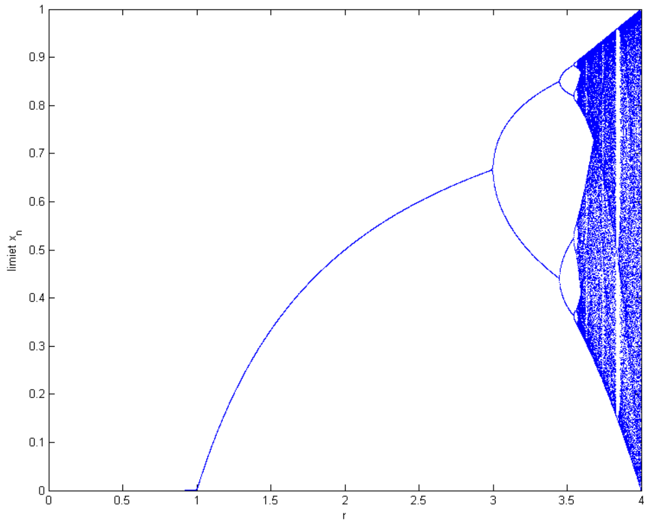
1.1 Het ecologische model van May

In Yorke's artikel gebruikt hij als voorbeeld een model voor het modelleren van insectenpopulatie met discrete generaties. De grootte van de nieuwe generatie is hierbij alleen afhankelijk van de grootte van de huidige. De bioloog Robert May, tevens een persoonlijke vriend van Yorke, bestudeerde dit model al in [5] en keek naar de verschillende limiet- populaties bij het variëren van de parameter r. Het model ziet er als volgt uit:

$$x_{n+1} = rx_n[1 - x_n/K]$$

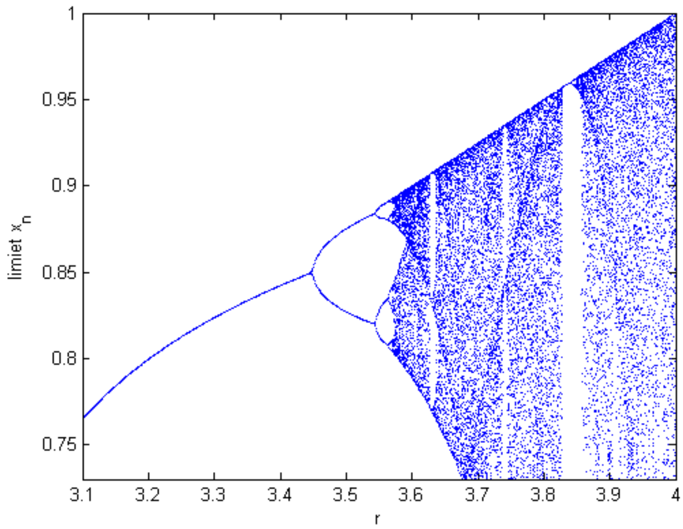
Hierbij stelt x_{n+1} de toekomstige populatie voor, x_n de huidige populatie, r de groeifactor en is $[1 - x_n/K]$ de term die de groei binnen de perken houdt. Voor kleine r sterft de soort uiteindelijk uit. Als je r groter neemt bereikt de populatie uiteindelijk een bepaalde limiet populatie, die groter wordt naarmate r groter genomen wordt. Echter neem je r nog groter, dan gaat dit discrete systeem chaos vertonen. De rij $x_0, f(x_0), f^2(x_0), \ldots$ convergeert dan niet meer.

Het model, dat May bestudeerde, is geïmplementeerd in Matlab. De code hiervan is te vinden in bijlage C.1. Hierbij hebben we $K = 1$ genomen en variëren we r tussen 0 en 4. We doen vervolgens per waarde van r n iteraties voor x_n, met als beginpunt x_0. Tot slot plotten we per r de laatste a waarden in een grafiek. Als we $n = 1000$, $x_0 = 0.1$ en $a = 100$ geeft dit geeft de volgende diagram. Dit soort diagrammen wordt ook wel een Feigenbaum diagram genoemd. De diagram is vernoemd naar de natuurkundige Mitchell Jay Feigenbaum, die dit diagram voor het eerst heeft onderzocht.
We zien hier dat de populatie 0 is en dus uitsterft voor $r < 1$. Voor r tussen 1 en ongeveer 3 gaat de populatie naar een limiet populatie. Tussen 3 en 3.4 krijgt de populatie een tweejarige cyclus; tussen 3.4 en 3.5 een vierjarige cyclus. Voor steeds grotere waarden van r krijgen we



Figuur 1: Limietwaarden van de populatie

steeds langere cycli tot op een gegeven moment er geen cyclus meer te zien is. Er is dan alleen "chaos". Maar als we heel goed kijken zien we dat bij ongeveer $r = 3.8$ de populatie een 3 jarige cyclus vertoont. De populatie neemt daar de waarden $0.149888\ldots$, $0.489172\ldots$ en $0.959299\ldots$ aan.



Figuur 2: Limietwaarden van de populatie

De 'structuur', die we zien tussen $r = 3\ldots4$, is typisch. Deze structuur zien we op vele niveaus. Als we bijvoorbeeld inzomen op het interval $[3.2, 3.5]$ zien we deze structuur weer verschijnen.

1.2 Chaos

Chaostheorie is een wetenschappelijk studiegebied, waarbinnen men het gedrag van bepaalde dynamische systemen bestudeert. Deze dynamische systemen zijn erg gevoelig voor begincondities. Een klein verschil in een bepaalde begincondities kan bij deze chaotische systemen al snel een groot verschil in de uitkomsten genereren. De dynamische systemen die de chaostheorie bestudeert zijn volledig deterministisch. Ze worden dus volledig bepaald door hun begincondities, er is dus geen enkel stochastisch element. De uitkomsten van een chaotische systeem zijn echter toch onmogelijk te voorspellen.

Een dynamisch systeem is niets anders dan een afbeelding van een ruimte naar zichzelf; $F : J \to J$. De metrische ruimte J wordt de faseruimte of toestandsruimte genoemd. Men bestudeert rijtjes $x_0, F(x_0), F^2(x_0), \ldots$. Deze rijtjes worden de banen van het dynamische systeem genoemd.
Er zijn verschillende definities voor een chaotisch dynamisch systeem. De definitie van chaos, die Devaney voorstelt [1], wordt echter het meest gebruikt. Devaney stelt dat een dynamisch systeem chaotisch is als hij voldoet aan de volgende drie condities:

1. Het dynamische systeem gevoelig voor zijn begincondities
2. Het dynamische systeem moet topologisch gemixt zijn (topological transitiviteit)
3. De periodieke punten van het systeem moeten een topologisch dichte verzameling vormen

De eerste conditie houdt in dat ieder punt in zo'n dynamisch systeem tot op willekeurige afstand benaderd kan worden door andere punten, die een volledig andere baan in de toekomst zullen volgen. Met andere worden een kleine afwijking van de huidige baan, zorgt voor significant ander gedrag in de toekomst.
In de tweede conditie staat dat een systeem topologisch gemixt moet zijn. Dit houdt in dat ieder deelgebied van zijn faseruimte uiteindelijk overlapt met iedere andere gegeven deelgebied van dezelfde faseruimte.
De laatste conditie wil zeggen dat ieder willekeurig punt in de ruimte willekeurig dicht benaderd kan worden door een verzameling van periodieke punten, een periodieke baan.

1.2.1 Definities en terminologie

Nu maken we het bovenstaande precies. Daarvoor hebben we echter eerst een aantal definities nodig. Bij deze definities laten we $F : J \to J$ met $J \subset \mathbb{R}^n$, $x \in J$ en $x = F^0(x)$ en $F^{n+1}(x) = F(F^n(x))$ voor $n = 0, 1, \cdots$.

Definitie (dynamisch systeem) Een paar (J, F) met $F : J \to J$ continu heet een dynamisch systeem. We nemen altijd J compact.

Definitie (baan) De baan van $x \in J$ is de verzameling gegeven door $\{F^n(x) : n \in \mathbb{N}\}$.

Definitie (periodiek punt) We noemen $p \in J$ een periodiek punt van periode $n\mathbb{N}$, als geldt $p = F^n(p)$ en $p \neq F^k(p)$ voor $1 \leq k < n$.

Definitie (uiteindelijk periodiek punt) We noemen q een uiteindelijk periodiek punt als voor een $m \in \mathbb{N}$ geldt, dat het punt $p = F^m(q)$ periodiek is . Met andere woorden de baan is eindig.

Merk op: omdat de functie F niet injectief hoeft te zijn kan een punt wel uiteindelijk periodiek zijn, maar zelf niet periodiek.

Definitie (limietpunt) Laat $S \subset \mathbb{R}$. Een punt $x \in \mathbb{R}$ is een limiet punt van S, als een rij punten $x_n \in S$ convergeert naar x.

Definitie (ω-limietverzameling) De ω-limietverzameling van een baan $x_0, F(x_0), F^2(x_0), \ldots$ is de verzameling van $y \in J$ waarvoor er een deelrij $(F^{n_k}(x))_{x=1}^{\infty}$ convergeert naar y.

N.B. in Figuur 2 zijn de ω-limieten geplot voor verschillende waarden van r.

Met deze definities kunnen we nu de woordelijke definitie van Devaney voor Chaos ook wiskundig formuleren:

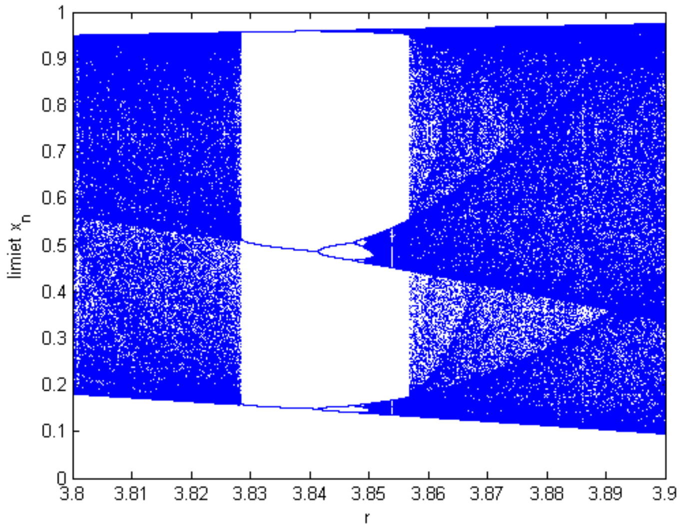
Definitie (chaos) Een system $F : J \to J$ met $x \in J$ en $x = F^0(x)$ en $F^{n+1}(x) = F(F^n(x))$ voor $n = 0, 1, \cdots$ is chaotisch als F voldoet aan de volgende drie condities:

1. Er bestaat een $\delta > 0$, zodat voor alle $x, y \in J$ met $x \neq y$ er een $N > 0$ bestaat zodat $\left|F^N(x) - F^N(y)\right| > \delta$. (gevoeligheid voor begin condities).
2. Er bestaat een $x \in J$ waarvoor de ω-limietverzameling gelijk is aan J (topologische transitiviteit)
3. De periodieke punten P liggen dicht in J. (topologisch dichte verzameling)

Er is bewezen dat wanneer een dynamisch systeem aan de eerste twee condities voldoet, dat het dan ook altijd aan de derde conditie voldoet [7].

1.3 Periode drie impliceert chaos

In het ecologische model van May viel Yorke iets op. Er was een klein interval van parameterwaarden r waarbij de ω-limiet bestaat uit 3 waarden, ongeacht de beginwaarde. Deze ω-limiet moet dan een punt zijn van periode 3. Als we nu inzomen op dit interval $[3.8, 3.9]$ zien we dit heel duidelijk.



Figuur 3: Periode 3 in het ecologische model

In zijn artikel laat Yorke zien dat een dynamisch systeem met een periodiek punt van periode 3, in feite punten van alle periodes bevat. N.B toch zien we die niet in het Feigenbaum diagram. Dit heeft te maken met kans 1. Yorke noemde dit verschijnsel van alle periodieke punten "chaos". Echter als we kijken naar de definitie van Devaney, is dit alleen een zwakkere vorm van onderdeel 3. Voordat we naar zijn stelling gaan, kijken we eerst naar de volgende twee lemma's.

Lemma 1.1 *Laat J een compact interval zijn en $F : J \to J$ een continu functie. Laat I een deelinterval zijn van J en $J \subseteq F(I)$. Dan bestaat er een punt $c \in I$ waarvoor geldt $F(c) = c$, oftewel c is een dekpunt.*

Bewijs Aangezien $J \subseteq F(I)$ bestaan er $\alpha, \beta \in I$ zodat $J = [F(\alpha), F(\beta)]$. Neem nu $G(x) = F(x) - x$. Dan is $G(\alpha) = F(\alpha) - \alpha < 0$ want $\alpha \in I \subseteq J$. Aan de andere kant $G(\beta) = F(\beta) - \beta > 0$. Nu volgt uit de tussenwaardestelling dat er een $c \in I$ bestaat zodanig dat $G(c) = 0$. Dit is het gewenste dekpunt. $\square$

Lemma 1.2 *Gegeven $F : J \to \mathbb{R}$ zodat $J \subseteq F(J)$. Dan bestaat er voor alle $k \in \mathbb{N}$ een interval $I_k \subseteq J$ zodat $F^k(I_k) = J$.*

Bewijs Dit bewijs gaat met inductie. Aangezien $J \subseteq F(J)$ bestaan er $\alpha, \beta \in J$ zodat $J = [F(\alpha), F(\beta)]$. Neem nu $I_1 = [\alpha, \beta]$ (eventueel kan $\alpha > \beta$ zijn).
Stel nu dat I_k is gedefinieerd. We bepalen nu I_{k+1}. Aangezien $F^{k+1}(I_k) = F(F^k(I_k)) = F(J)$, omvat dit J. Dus bestaan er $\alpha_k, \beta_k \in I_k$ zodat $J = [F^{k+1}(\alpha_k), F^{k+1}(\beta_k)]$. Neem nu $I_{k+1} = [\alpha_k, \beta_k]$ (eventueel kan $\alpha_k > \beta_k$ zijn). Dan $J = F^{k+1}(I_{k+1})$ vanwege samenhang, oftewel de tussenwaardestelling. $\square$

Nu gaan we kijken naar de hoofstelling van het bekende artikel van Li en Yorke: "Period three implies chaos".

Stelling 1.3 (Li en Yorke) *Laat J een interval zijn en laat $F : J \to J$ continu zijn. Neem nu aan dat er een punt $a \in J$ bestaat waarvoor geldt:*

$$F^3(a) \leq a < F(a) < F^2(a) \tag{1.1}$$

Dan geldt er:

1. *Voor iedere $k = 1, 2, \cdots$ is er een periodiek punt in J met periode k.*

2. *De dynamica op J is gerelateerd aan de deelverzameling van Σ_2 waarbij geen twee nullen na elkaar staan, met de shift operator T; een zogenaamnde subshift van eindig type.*

Bewijs van deel 1 van de stelling van Li en Yorke. Neem de punten $b = F(a)$, $c = F^2(a)$. We definiëren nu twee intervallen $I_0 = [a, b]$ en $I_1 = [b, c]$. Uit onze aannames van continuïteit en ons punt a volgt dat $I_1 \subset F(I_0)$ en dat $I_1 \cup I_0 \subset F(I_1)$.
We kijken nu eerst of F ook dekpunten bevat oftewel of F punten bevat van periode 1. F is continu en $I_1 \subseteq F(I_1)$, dus volgt uit Lemma 1.1 dat F een dekpunt heeft tussen b en c.
We gaan nu kijken naar een punt van periode 2. Omdat $I_1 \subseteq F(I_0)$ weten we uit Lemma 1.2, dat er een deelverzameling $A_1 \subseteq I_0$ bestaat, zodanig dat $F(A_1) = I_1$. Uit hetzelfde Lemma volgt ook dat omdat $I_0 \subseteq F(I_1)$ er een deelverzameling $A_2 \subseteq I_1$ bestaat zodanig dat $F(A_2) = A_1$. Nu geldt dus $F^2(A_2) = F(A_1) = I_1 \supseteq A_2$, oftewel $A_2 \subseteq F^2(A_2)$. Uit Lemma 1.1 volgt nu dat F^2 een dekpunt α_2 heeft in $A_1 \subset [a, c]$. Waarvoor geldt dat $\alpha_2 \in I_1$, $F(\alpha_2) \in I_0$ en $F^2(\alpha_2) = \alpha_2 \in I_1$. Merk op dat $I_0 \cap I_1 = \{b\}$ en dat b een punt is van periode 3 of meer. Dus $\alpha_2 \neq b$ en F heeft een punt van periode 2.

We zoeken nu naar punten met periode 3. Neem A_2 en A_1 weer zoals hiervoor. Omdat $A_2 \subset I_1$ en $I_1 \subset F(I_1)$ weten we weer uit Lemma 1.2 dat er een deelverzameling A_3 van I_1 bestaat zodanig dat $F(A_3) = A_2$. Dus $F^3(A_3) = F^2(A_2) = F(A_1) = I_1 \supseteq A_3$. Nu weten we uit Lemma 1.1 dat F^3 een dekpunt α_3 heeft in $A_3 \subset [a, c]$. Voor α_3 geldt dat $F(\alpha_3), \alpha_3 \in I_1$, $F^2(\alpha_3) \in I_0$ en $F^3(\alpha_3) = \alpha_3 \in I_1$. Dus heeft F een punt van periode 3.

Op eenzelfde wijze kunnen we nu ook een punt van periode k maken. Neem A_1 en A_2 net als hiervoor en definieer de deelintervallen A_i in I_1 voor $i = 3, 4, \ldots, k$ zodanig dat $F(A_i) = A_{i-1}$ met $A_i \subset I_1$ voor $i = 3, 4, \ldots, k$. Dit kan met behulp van Lemma 1.2. Nu geldt dus $F^k(A_k) = F^{k-1}(A_{k-1}) = \ldots = F^2(A_2) = F(A_1) = I_1 \supseteq A_k$. Nu weten we uit Lemma 1.1 dat F^k een dekpunt α_k heeft in A_k. Voor dit dekpunt geldt dat $F^{k-2}, \ldots, F(\alpha_k), \alpha_k \in I_1$, $F^{k-1}(\alpha_k) \in I_0$ en $F^k(\alpha_k) = \alpha_k \in I_1$. Dus F heeft een punt, dat zich $k-1$ iteraties in I_1 bevindt en 1 keer in I_0. Dus F heeft een punt van periode k. Hiermee is het gevraagde bewezen.

$\square$

Merk op dat aan deze stelling voldaan wordt als er een periodiek punt is met periode 3.
Voordat we naar onderdeel 2 gaan kijken moeten we eerst definiëren wat Σ_2 en T zijn. Σ_2 is de rijruimte met de symbolen 0 en 1. Elementen uit Σ_2 zijn oneindige rijen van nullen en enen. De subshift T plaats alle elementen van de rij één plek. De precieze definitie van Σ_2 en T zijn als volgt:

Definitie $\Sigma_2 = \{\mathbf{s} = (s_0 s_1 s_2 \ldots) | s_j = 0 \text{ of } 1\} = \{0, 1\}^{\mathbb{N}}$

Definitie De shift afbeelding $T : \Sigma_2 \to \Sigma_2$ wordt gegeven door $T(s_0 s_1 s_2 \ldots) = (s_1 s_2 s_3 \ldots)$

Bewijs van deel 2 van de stelling van Li en Yorke. We definiëren nu eerst de deelverzameling Σ_2' van Σ_2.

$$\Sigma_2' = \{\mathbf{s} = (s_0 s_1 s_2 \ldots) | s_j = 0 \text{ of } 1 \text{ en } s_j = 0 \Rightarrow s_{j+1} \neq 0\}$$

met $T : \Sigma_2' \to \Sigma_2'$ de subshift van eindig type. Σ_2' zijn dus de oneindige rijen nullen en enen waarbij nooit twee nullen naast elkaar staan. Uit deel 1 van de stelling weten we dat er een punt is in J van periode 3. Neem nu aan dat dit punt a is, dus $a = F^3(a) < F^2(a) = b < F(a) = c$. Dit mag zonder verlies van algemeenheid. Neem nu $I_0 = [a, b]$ en $I_1 = (b, c]$. Voor willekeurige $x \in [a, c]$ vervang de baan $x, F(x), F^2(x), \ldots$ door 0 of 1 al naar gelang $F^k(x) \in [a, b]$ of $F^k(x) \in [b, c]$. Op deze manier krijgen we een afbeelding van $H : [a, c] \to \{0, 1\}^{\mathbb{N}}$. Merk op dat F en T samen commutatief zijn ten opzichte van deze afbeelding H, oftewel: $T \circ H = H \circ F$.

Om de relatie tussen de dynamica op (J, F) en (Σ_2', T) verder weer te geven, tonen we aan dat het $H(J) \subset \Sigma_2'$ en uiteindelijk dat zelfs dat $H(J) = \Sigma_2'$. We laten nu eerst zien dat elk periodieke punt $(s) \in \Sigma_2'$ er ook een $x \in J$ bestaat zodanig dat $H(x) = \mathbf{s}$. Een periodiek punt in Σ_2' is een rij getallen $(s_0 s_1 s_2 \ldots)$ waarin een eindige rij getallen, zeg $(\tilde{s}_0 \tilde{s}_1 \ldots \tilde{s}_n)$ zich oneindig herhaald.
Gegeven een periodiek punt $(s) \in \Sigma_2'$ van periode n met $(\tilde{s}_0 \tilde{s}_1 \ldots \tilde{s}_n)$. Uit onze aannames van continuïteit en ons punt a volgt dat $I_1 \subset F(I_0)$ en dat $I_1 \cup I_0 \subset F(I_1)$. Definieer nu de volgende intervallen. Neem A_0 als deelverzameling van $I_{\tilde{s}_0}$ zodanig dat $F(A_0) = I_{\tilde{s}_n}$. Neem vervolgens A_1 als deelverzameling van $I_{\tilde{s}_1}$ zodanig dat $F(A_1) = A_0$. Neem vervolgens A_2 als deelverzameling van $I_{\tilde{s}_2}$ zodanig dat $F(A_2) = A_1$, enzovoort. Op deze manier nemen we A_i als deelverzameling van $I_{\tilde{s}_i}$ zodanig dat $F(A_i) = A_i - 1$ voor alle $i \in \{1, 2, \ldots, n\}$.
Op deze manier hebben we een rij intervallen gedefinieerd zodanig dat, $F^n(A_n) = F^{n-1}(A_{n-1}) = \ldots = F^2(A_2) = F(A_1) = I_{\tilde{s}_n} \supseteq A_k$ nu volgt uit Lemma 1.1 dat $F^k(A_k)$ een dekpunt α heeft. Voor dit dekpunt geldt $F^k(\alpha) = F^{k+n}(\alpha)$ en dat $F^i(\alpha) \in I_{\tilde{s}_n}$ voor alle $i \in \mathbb{Z}$. Voor α geldt dus dat $H(\alpha) = (s)$.

We laten nu zien dat voor elk element $(s) \in \Sigma_2'$ er een punt $x \in J$ bestaat zodanig dat $H(x) = (s)$. Met dezelfde inductie als we hierboven hebben gebruikt definiëren we nu de volgende intervallen:

$$J_{s_0 s_1 s_2 \ldots s_n} = \{x : F^k(x) \in I_{s_k} \text{ voor } 0 \leq k \leq n\}$$

Uit het punt van periode 3 en continuïteit volgt dat deze verzameling niet leeg is en gesloten. Verder door de manier waarop we $J_{s_0 s_1 s_2 \ldots s_n}$ hebben gedefinieerd, weten we dat

$$J_{s_0} \supset J_{s_0 s_1} \supset J_{s_0 s_1 s_2} \supset J_{s_0 s_1 s_2 s_3} \ldots \tag{1.2}$$

Dit volgt uit dezelfde redenatie, die we hierboven hebben gebruikt.
Neem nu:

$$S_n = \bigcap_{k=0}^{n} J_{s_0 s_1 s_2 \ldots s_k}$$

Omdat 1.2 geldt en omdat alle $J_{s_0 s_1 s_2 \ldots s_n}$ compact zijn volgt hier nu uit dat S_n ook niet leeg is voor $n \to \infty$. Dus $\overline{S} = \bigcap_{k=0}^{\infty} J_{s_0 s_1 s_2 \ldots s_k}$ is niet leeg. Er bestaat dus een punt $x \in \overline{S}$, waarvoor geldt dat $F^i(x) \in I_{s_i}$ voor alle $i \in \{0, 1, 2, \ldots\}$. Voor deze x geldt dan $H(x) = (s)$. Dus voor alle $\mathbf{s} \in \Sigma_2'$ er een punt $x \in J$ bestaat zodanig dat $H(x) = \mathbf{s}$. Dus $H(J) = \Sigma_2'$.

$\square$

1.4 De periodeverdubbelende weg naar chaos

Zoals we experimenteel gezien hebben in sectie 1.2 vertoont het dynamische systeem (J, F) eenvoudige dynamica, als $0 \leq r \leq 3$. Echter als r groter wordt, wordt de dynamica ingewikkelder en kan het bewezen worden, dat voor $r \geq 4$ het systeem chaotisch is [1] [51-52]. Nu is de vraag: wat gebeurt er tussen $r = 3$ en $r = 4$? Voor $r = 3$ vertoont het systeem simpele dynamica, maar voor $r \geq 4$ vertoont het systeem chaotisch gedrag.

Een gedeeltelijk antwoord wordt gegeven door de stelling van Sarkovskii. Sarkovskii introduceerde de volgende ordening van de natuurlijke getallen:

$$3 \triangleright 5 \triangleright 7 \triangleright \cdots \triangleright 2 \cdot 3 \triangleright 2 \cdot 5 \triangleright \cdots \triangleright 2^2 \cdot 3 \triangleright 2^2 \cdot 5 \triangleright \ldots$$

$$2^3 \cdot 3 \triangleright 2^3 \cdot 5 \triangleright \cdots \triangleright 2^3 \triangleright 2^2 \triangleright 2 \triangleright 1$$

Oftewel we beginnen met de oneven getallen vanaf 3, dan de oneven getallen vanaf 3 keer 2, dan keer 2^2, et cetera. Tot slot komen in de rij de even getallen van groot naar klein en dan 1.

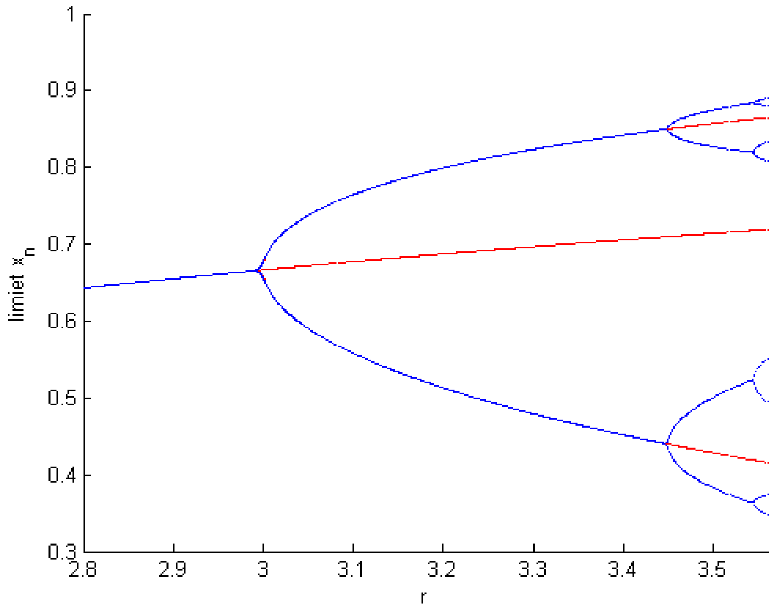
Stelling 1.4 (Sarkovskii) *Laat $F : J \to J$ en laat F een periodiek punt hebben met periode k. Als $k \triangleright l$ in de Sarkovskii ordering van de natuurlijke getallen, dan heeft F ook een periodiek punt van periode l.*

Het bewijs van de stelling van Sarkovskii is te vinden in [1][63-65]. Het bewijs lijkt veel op het bewijs van het eerste deel van de stelling van Li en Yorke.

Als we nu kijken naar het dynamische systeem (J, F) zien we de periodieke punten verschijnen in de tegenovergestelde volgorde als van de Sarkovskii ordening. We zien dus eerst alle punten van periode 2^j, voordat we uiteindelijk oneindig veel periodieke punten zien van verschillende ordes. Vanuit de bifurcatietheorie zijn er twee typische manieren, waarop deze periodieke punten kunnen verschijnen: in zadelknoopbifurcaties of in zogeheten periodeverdubbelingen.
In het geval van (J, F) zien we dat chaos ontstaat door een opeenvolging van periodeverdubbelingen. Dit is niet altijd de manier waarop dit gebeurt, maar het is de typische 'weg naar chaos'. Voor $r > 3$ convergeert F niet meer naar één punt, maar naar een punt van periode 2. Merk op dat volgens de stelling van Sarkovskii er ook een dekpunt is. Voor $r > 3.45$ convergeert F naar een punt van periode 4, maar ook dan zijn er punten van periode 2 en een dekpunt. Echter deze zijn niet aantrekkend.

Definitie (aantrekkend/afstotend) Een punt p van periode n is een aantrekkend (afstotent) periodiek punt, als geldt $|(F^n)'(p)| < 1$ ($|(F^n)'(p)| > 1$).

We hebben deze ontwikkeling van verdubbelende periodes voor grotere r geplot tot punten van periode 8. De aantrekkende periodieke punten zijn blauw; de afstotende rood.



Figuur 4: De periode verdubbelende weg naar chaos

In het vervolg van deze scriptie zullen we verder niet bewijzen dat een dynamisch systeem (of specifieker een optimalisatiealgoritme) chaotisch gedrag bevat, maar zullen we experimenteel bepalen of in het systeem de periodeverdubbelde weg naar chaos bevindt. Het daadwerkelijk bepalen of een systeem chaotisch is, vraagt een flinke analyse per probleem en zou buiten het doel van een bachelorproject vallen.

2 Lineair programmeren

Lineair programmeren is een methode voor het oplossen van optimaliseringsproblemen, waarbij de doelfunctie en de randvoorwaarden lineair zijn. Deze optimalisatieproblemen kregen voor het eerst echt aandacht vlak voor de tweede wereldoorlog. Lineair programmeren zelf werd voor het eerst ontwikkeld door Leonid Kantorovich in 1939. Hij gebruikte de theorie voor het optimaal verdelen van grondstoffen. Met de communistische plan politiek in die tijd, was dit een belangrijk vraagstuk. Ook op militair gebied werd de theorie toegepast om onder andere eigen uitgaven te minimaliseren en de verliezen bij de tegenstander te vergroten. Dit gebeurde ook aan de Amerikaanse zijde. Daar was George B. Danzig een van de eerste wiskundigen die zich met zogenaamde LP-problemen bezig hield. Danzig heeft onder andere het simplex algoritme voor ontwikkeld.

2.1 Affine Scaling Algoritme

Wij zullen ons echter niet bezighouden met het simplex algoritme, maar met zogeheten 'affine scaling algoritme' voorgesteld door VanderBei, Meketon en Freedman [2]. Dit type algoritme is een aanpassing van het algorithme dat door Dikin werd voorgesteld in 1967 en is een voorbeeld van een interne punt methode. Deze lost het volgende probleem op, ook wel het primale probleem genoemd:

$$\begin{aligned} \text{minimaliseer} \quad & c^T x \\ \text{onderhevig aan} \quad & Ax = b, \\ & x \geq 0 \end{aligned}$$

met $c, x \in \mathbb{R}^n$ kolom vectoren, A een matrix van $m \times n$ met rang m en $b\mathbb{R}^m$ een kolom vector. Verder nemen we aan dat het probleem een niet triviale oplossing heeft. Dit houdt in dat het polygoon $P = \{x : Ax = b, x \geq 0\}$ *niet leeg* is en dat $c^T x$ niet constant is op P.
Het affine scaling algoritme maakt bij het oplossen van het optimalisatieprobleem gebruik van het duale probleem. Deze is als volgt gedefinieerd:

$$\begin{aligned} \text{maximaliseer} \quad & b^T \lambda \\ \text{onderhevig aan} \quad & A^T \lambda \leq c, \end{aligned}$$

Waarbij $\lambda \in \mathbb{R}^m$ een kolomvector. Over het algemeen geldt dat $c^T x \geq b^T \lambda$ voor elke x in het primale polygoon en voor elke λ in het duale polygoon. Echter omdat dit een lineair optimalisatieprobleem is kunnen we de sterke dualiteitsstelling gebruiken. Deze zegt dat als $\tilde{x}$ de optimale oplossing is voor het primale probleem en $\tilde{\lambda}$ de optimale oplossing is voor het duale probleem, dat dan $c^T x = b^T \lambda$. LP-algoritmen, die gebruik maken van het duale probleem, zoeken dan ook altijd naar een x_0 en een λ_0 zodanig dat $c^T x_0 = b^T \lambda_0$. Het verschil $c^T x - b^T \lambda$ heet ook wel het dualiteitsgat.

Het affine scaling algoritme kiest vervolgens een stapgrootte t, $0 < t < 1$, en maakt vervolgens de primale reeks $\{x^i\}$ en de duale reeks $\{\lambda^i\}$ met $i = \{1, 2, 3, \ldots\}$. Voor het berekenen van x^{k+1} zoekt het algoritme eerst naar de optimale richting in het duale probleem en update vervolgens x_k in deze richting, waaruit x^{k+1} volgt. Het algoritme ziet er als volgt uit:

Affine Scaling Algoritme

Parameters

ϵ is de *accuracy* parameter

t is de stapgrootte

Een *stopping criterion*

Input

$x_0 > 0$ de begin vector zodanig dat $Ax_0 = b$

begin

$x = x_0$

while *stopping criterion* **do**

$D_k = diag(x_1^k, \ldots, x_n^k)$

$\lambda^k = (AD_k^2 A^T)^{-1} AD_k^2 c$

$s^k = c - A^T \lambda^k$

$x^{k+1} = x^k - t\frac{D_k^2 s^k}{\max_{i \in \{1,\ldots,n\}}(D_k s^k)_i}$

end

end

Hier is $\max_{i \in \{1,\ldots,n\}}(D_k s^k)_i$ de maximale pivot van de vector $D_k s^k$. De *stopping criterion* die we gebruiken is $x_k^T s_k < \epsilon, \epsilon > 0$. Dit zegt eigenlijk dat het algoritme stopt als het dualiteitsgat kleiner is dan ϵ.

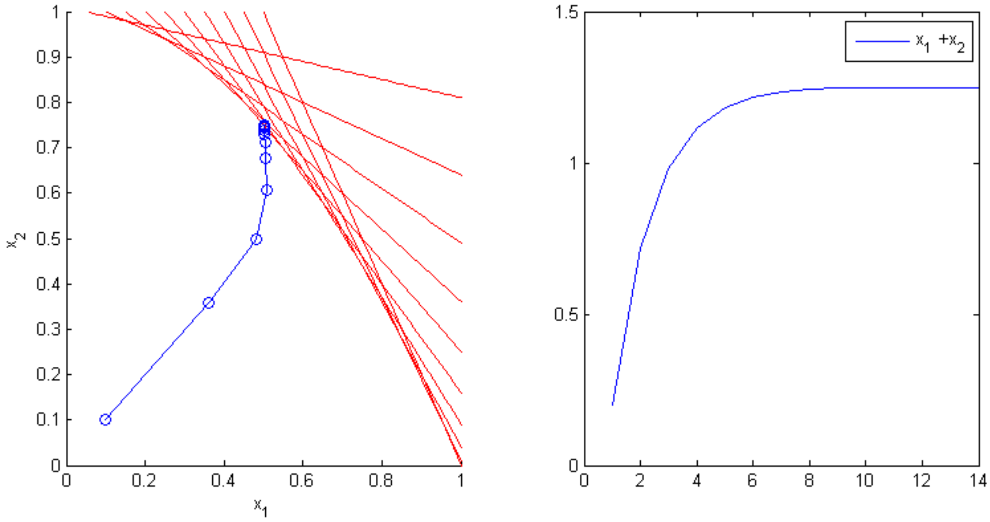
Om een illustratie te kunnen geven van de manier waarop dit algoritme convergeert, kijken we naar het volgende probleem:

$$
\begin{array}{rl}
\text{maximaliseer} & x_1 + x_2 \\
\text{onderhevig aan} & 2px_1 + x_2 \leq p^2 + 1 \\
 & x \geq 0 \\
 & \text{met } p = 0, 0.1, \ldots, 0.9, 1
\end{array}
$$

Nadat we het affine scaling algoritme hebben geprogrammeerd in Matlab, kunnen we dit probleem oplossen. Hiervoor moeten we het probleem herschrijven als een minimaliseringsprobleem en moeten we slackvariabelen toevoegen. Slackvariabelen worden geïntroduceerd in de vorm van de vector s, zodat alle ongelijkheden $A^T\lambda \leq c$ gelijkheden worden. Het duale probleem ziet er dan als volgt uit:

$$\begin{aligned} \text{maximaliseer} \quad & b^T\lambda \\ \text{onderhevig aan} \quad & A^T\lambda + s = c, \\ & \lambda \geq 0 \end{aligned}$$

Hieronder zie je een illustratie van de oplossing.



Figuur 5: Affine scaling algoritme voor

In dit geval is $t = 0.5$ en $\epsilon = 0.0001$. In de linkerplot zien we duidelijk hoe het algoritme stapsgewijs naar de oplossing convergeert met telkens een aangepaste richting. Dit gebeurt uiteindelijk in 14 iteraties.

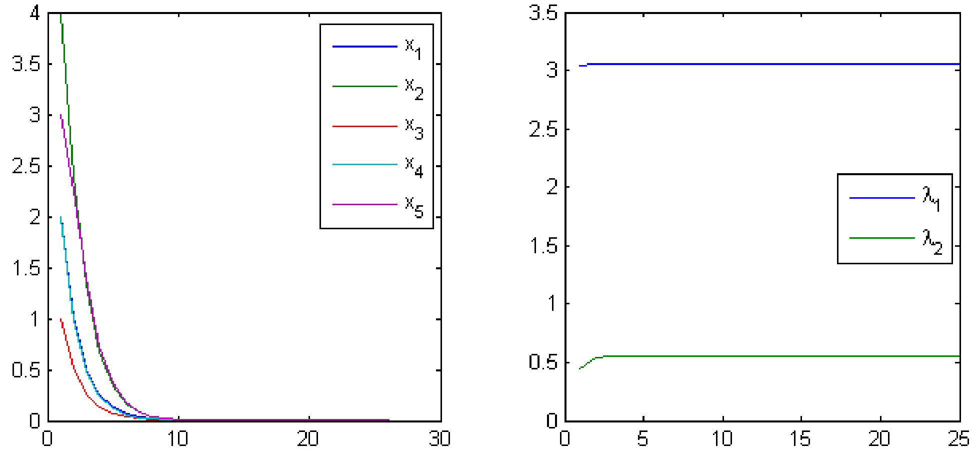
We gaan nu naar een volgend probleem kijken. Dit is het probleem dat Castillo en Barnes [4] bekijken. Castillo en Barnes tonen met dit voorbeeld aan, dat in het affine scaling algoritme chaos kan voorkomen. Deze chaos zit dan in $\{\lambda_n\}$, dus in de zoekrichting van het algoritme. De rij $\{x_n\}$ convergeert nog wel naar de optimale oplossing. Dit is geïmplementeerd in Matlab als controle op de correctheid van mijn code voor het affine scaling algoritme. Het probleem ziet er als volgt uit:

$$\begin{aligned} \text{minimaliseer} \quad & 10x_1 + 10x_2 + 5x_3 + x_4 + x_5 = 0 \\ \text{onderhevig aan} \quad & x_1 + 2x_2 - 3x_3 - 2x_4 - x_5 = 0 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

De duale van dit probleem ziet er als volgt uit:

$$\begin{array}{rl} \text{maximaliseer} & 0 \\ \text{onderhevig aan} & \lambda_1 - \lambda_2 \leq 10 \\ & 2\lambda_1 + 2\lambda_2 \leq 10 \\ & -3\lambda_1 - \lambda_2 \leq 5 \\ & -2\lambda_1 - \lambda_2 \leq 1 \\ & -\lambda_1 - \lambda_2 \leq 1 \end{array}$$

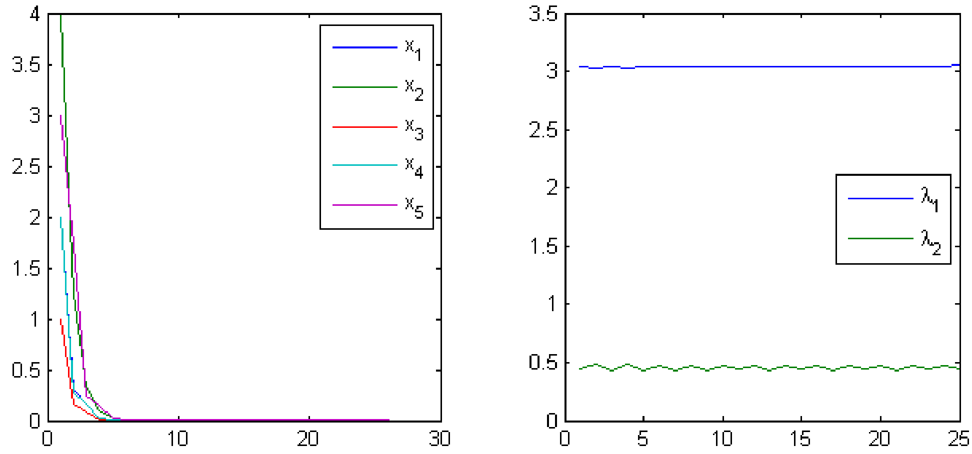
We zien meteen dat $x_1 = x_2 = x_3 = x_4 = x_5 = 0$ de optimale oplossing is voor dit probleem. Als we nu $x_0 = [2, 4, 1, 2, 3]^T$ nemen en $t = 0.5$ zien we dat het algoritme naar de oplossing convergeert.



Figuur 6: Affine scaling algoritme voor $t = 0.5$

Zowel de primale als de duale variabelen zien we convergeren.

Echter nemen we $t = 0.86$ dan krijgen we in de duale variabelen geen convergentie.



Figuur 7: Affine scaling algoritme voor $t = 0.86$

Als we kijken naar de duale variabelen voor $k = 17, \ldots, 22$ zien we het volgende:

$$\lambda_{17} = \begin{pmatrix} 3.0432890 \\ 0.4674003 \end{pmatrix}, \quad \lambda_{18} = \begin{pmatrix} 3.0440522 \\ 0.4754970 \end{pmatrix},$$
$$\lambda_{19} = \begin{pmatrix} 3.0433001 \\ 0.4675194 \end{pmatrix}, \quad \lambda_{20} = \begin{pmatrix} 3.0440708 \\ 0.4757432 \end{pmatrix},$$
$$\lambda_{21} = \begin{pmatrix} 3.0432944 \\ 0.4674515 \end{pmatrix}, \quad \lambda_{22} = \begin{pmatrix} 3.0440598 \\ 0.4755515 \end{pmatrix}.$$

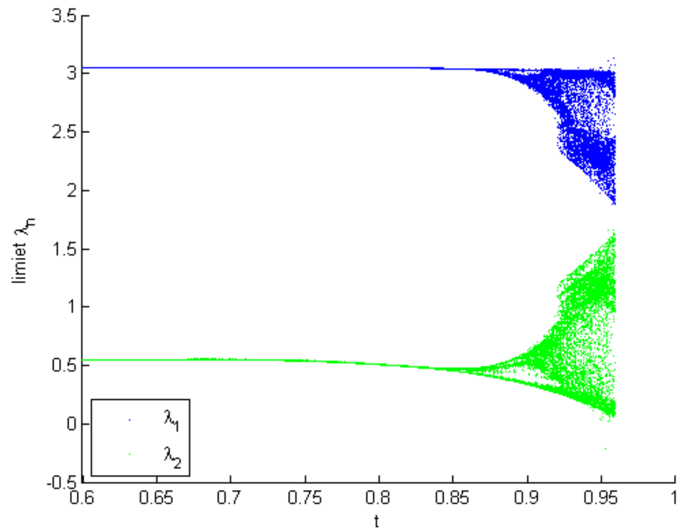
Voor oneven k hebben we dus een rij die lijkt te convergeren naar het punt $[3.0432968, 0.46748284]^T$ en voor even k naar $[3.0440618, 0.4756198]^T$. De rij $\{\lambda_n\}$ lijkt dus te convergeren naar een punt van periode 2. Dat dit zo is wordt bewezen in het artikel van Castillo en Barnes.

We kunnen nu nog verder kijken voor grotere waarden van t. We verwachten dat we convergentie zien naar punten met een grotere periode. Als we bijvoorbeeld kijken naar $t = 0.89$ zien we periode vier.

$$\lambda_{13} = \begin{pmatrix} 3.0318 \\ 0.3659 \end{pmatrix}, \quad \lambda_{14} = \begin{pmatrix} 2.9891 \\ 0.5019 \end{pmatrix}, \quad \lambda_{15} = \begin{pmatrix} 3.0314 \\ 0.3726 \end{pmatrix}, \quad \lambda_{16} = \begin{pmatrix} 2.9650 \\ 0.5469 \end{pmatrix},$$
$$\lambda_{17} = \begin{pmatrix} 3.0318 \\ 0.3659 \end{pmatrix}, \quad \lambda_{18} = \begin{pmatrix} 2.9891 \\ 0.5019 \end{pmatrix}, \quad \lambda_{19} = \begin{pmatrix} 3.0314 \\ 0.3725 \end{pmatrix}, \quad \lambda_{20} = \begin{pmatrix} 2.9650 \\ 0.5467 \end{pmatrix}.$$

We zien dus dat voor $t > \frac{2}{3}$ de rij $\{\lambda_k\}$ convergeert naar periodieke punten met verschillende periodes. Verder lijken de periodes te verdubbelen naar mate t groter wordt. Het lijkt dus op de situatie die we ook hadden bij het ecologische model van May.

We kunnen ook hier weer t laten lopen tussen 0 en 1 en vervolgens de limiet punten plotten om een Feigenbaum diagram te krijgen:



Figuur 8: Feigenbaum voor het Affine scaling algoritme

In het affine scaling algoritme komt dus chaos voor.

2.2 Primal-dual Dikin-type algoritme

Nu we gezien hebben dat in het affine scaling algoritme chaos voorkomt, zijn we benieuwd of we deze chaos ook in een ander algoritme kunnen vinden. Het algoritme waar we ons vanaf nu mee zullen bezig houden is het primal-dual Dikin-type algoritme beschreven door Jansen, Roos en Terlaky in [3]. Opmerkelijk genoeg was dit de originele opzeet van dit bachelor project.

Het primal-dual Dikin-type algoritme is ook een interne punt methode. Deze methode maakt gebruik van twee richtingen voor het zoeken naar de optimale oplossing; de affine scaling richting en de zogeheten centrerende richting. Hierbij zorgt de affine scaling richting naar de optimale oplossing convergeert en de centrerende richting zorgt ervoor dat de oplossing naar het analytische centrum van het uitvoerbare ('feasible') gebied convergeert. Door een goede combinatie te nemen van deze twee richtingen kan men ervoor zorgen dat de methode naar de optimale oplossing convergeert.

Verder bestaat voor het affine scaling algoritme geen bewijs dat het convergeert in polynomiale tijd. Voor het Primal-dual Dikin-type algoritme hebben Jansen, Roos en Terlaky wel kunnen bewijzen, dat deze conveert in $O(nL)$ met L een constante afhangend van de begin vectoren en de stapgrootte. Ter herinnering, het primal-dual Dikin-type algoritme lost het volgende (primale) probleem op:

$$\begin{aligned} \text{minimaliseer} \quad & c^T x \\ \text{onderhevig aan} \quad & Ax = b, \\ & x > 0 \end{aligned}$$

Het duale probleem, dat men gebruikt in het Primal-dual Dikin-type algoritme, wordt voor primal-dual Dikin-type algoritme geformuleerd met slackvariablen. Het duale probleem ziet er dan als volgt uit:

$$\begin{aligned} \text{maximaliseer} \quad & b^T y \\ \text{onderhevig aan} \quad & A^T y + s = c, \\ & s > 0 \end{aligned}$$

Waarbij aangenomen dat $s > 0$ en niet $s \geq 0$. Er is verder ook vereist dat $x > 0$. Dit moet gelden omdat er bij het algoritme coördinaatsgewijs gedeeld wordt door de vectoren s en x. Aangezien we hebben aangenomen dat de polytoop $P = \{x : Ax = b, x \geq 0\}$ niet leeg is, kunnen we altijd beginvectoren $x_0 \neq 0$ en $s_0 \neq 0$ kiezen. We gebruiken enkel inwendige punten in P. He blijkt dat deze vereiste $x, s > 0$ geen problemen oplevert. Merk verder op dat y uniek bepaald wordt door s, omdat $s = c - A^T y$ en $rang(A) = m$
Het idee van het algoritme is om te starten met x, y en s en dan aan te passen naar $x + \Delta x$, $y + \Delta Y$ en $s + \Delta s$ totdat een oplossing gevonden is. De aanpassingen Δx, Δy en Δs noemen we de zoekrichtingen. Om te voldoen aan de condities moet worden voldaan aan:

$$A\Delta x = 0, \quad A^T \Delta y + \Delta s = 0 \tag{2.1}$$

en:

$$\left\| X^{-1}\Delta x + S^{-1}\Delta s \right\| \leq 1 \tag{2.2}$$

met X en S de diagonaal matrices met op de diagonalen de elementen van x dan wel s en $\|\cdot\|$ de euclidische norm.
De eerste eis zorgt ervoor dat de oplossingen aan de conditie $Ax = b$ blijven voldoen. De tweede eis vervangt de condities $x > 0$ en $s > 0$ door een zogenaamnde Dikin ellipsoide. Dit is een idee van Dikin.

Het doel van het primal-dual Dikin-type algoritme is om in elke iteratie (een stap in een dynamisch stysteem) het dualiteitsgat $c^T x - b^T y$ te verkleinen tot in de limiet 0. Omdat we hier te maken hebben met een lineaire doelfunctie, moet weer gelden dat als x en y optimaal zijn, dat dan geldt dat $c^T x - b^T y = 0$. Het dualiteitsgat is dan minimaal. We kunnen het dualiteitsgat ook anders formuleren namelijk:

$$\begin{aligned} c^T(x + \Delta x) - b^T(y + \Delta y) &= c^T(x + \Delta x) - (Ax)^T(y + \Delta y) \\ &= c^T(x + \Delta x) - x^T A^T(y + \Delta y) \\ &= (c - A^T y)^T x + c^T \Delta x + s^T \Delta x - x^T A^T \Delta y \\ &= s^T x + (A^T y + s)^T \Delta x - x^T A^T \Delta y \\ &= s^T x + (y^T A + s^T)\Delta x + x^T \Delta s \\ &= s^T x + s^T \Delta x + x^T \Delta s \end{aligned}$$

Als we nu vereisen dat Δx en Δs orthogonaal zijn oftwel $\Delta x^T \Delta s = 0$ dan geldt het volgende:

$$c^T(x + \Delta x) - b^T(y + \Delta y) = (x + \Delta x)^T(s + \Delta s)$$

$x^T s$ is hierbij een constante. We kunnen nu onze zoekrichting bepalen door het minimaliseren van het dualiteitsgat $x^T s$, als we eisen dat Δx en Δs orthogonaal zijn. Wij krijgen dan het volgende probleem:

$$\begin{array}{rl} \text{minimaliseer} & x^T \Delta s + s^T \Delta x \\ \text{onderhevig aan} & A^T \Delta y + \Delta s = 0, \\ & \left\| X^{-1} \Delta x + S^{-1} \Delta s \right\| \leq 1 \\ & \Delta x^T \Delta s = 0 \end{array}$$

Nu geldt dat:

$$x^T \Delta s + s^T \Delta x = (xs)^T (X^{-1} \Delta x + S^{-1} \Delta s) \geq - \|xs\| \left\| X^{-1} \Delta x + S^{-1} \Delta s \right\| \geq - \|xs\|$$

Waarbij xs de puntsgewijze vermenigvuldiging van x en s. Voor deze ongelijkheid geldt gelijkheid dan en slechts dan als:

$$X^{-1} \Delta x + S^{-1} \Delta s = -\frac{xs}{\|xs\|}$$

We hebben nu een uitdrukking voor het minimum. Onze zoekrichting wordt daarom bepaald door het volgende stelsel van vergelijkingen:

$$\begin{aligned} A \Delta x &= 0 && (2.3) \\ A^T \Delta y + \Delta s &= 0 && (2.4) \\ X^{-1} \Delta x + S^{-1} \Delta s &= -\frac{xs}{\|xs\|} && (2.5) \end{aligned}$$

Als we 2.5 vermenigvuldigen met $\sqrt{x}\sqrt{s}$, dan krijgen we:

$$\begin{aligned} X^{-1} \sqrt{x}\sqrt{s} \Delta x + S^{-1} \sqrt{x}\sqrt{s} \Delta s &= -\sqrt{x}\sqrt{s} \frac{xs}{\|xs\|} \\ \frac{\sqrt{s}}{\sqrt{x}} \Delta x + \frac{\sqrt{x}}{\sqrt{s}} \Delta s &= -\frac{\sqrt{xs}^3}{\|xs\|} \\ d^{-1} \Delta x + d \Delta s &= p_v && (2.6) \\ & && (2.7) \end{aligned}$$

Met $d = \frac{\sqrt{x}}{\sqrt{s}}$ en $p_v = -\frac{(\sqrt{xs})^3}{\|xs^T\|}$. Omdat Δx en Δs orthogonaal zijn, zijn $d^{-1} \Delta x$ en $d \Delta s$ ook orthogonaal. Uit 2.3 volgt:

$$A \Delta x = A D d^{-1} \Delta x = 0$$

en uit 2.4 volgt:

$$A^T \Delta y + \Delta s = D A^T \Delta y + d \Delta s = 0$$

Daarom moet $d^{-1}\Delta x$ wel in de nulruimte zitten van AD en moet $d\Delta s$ in de rijruimte zitten van AD. Nu omdat $d^{-1}\Delta x + d\Delta s = p_v$ geldt dat:

$$\begin{aligned} \Delta x &= P_{null(AD)}(p_v) \\ \Delta s &= P_{row(AD)}(p_v) \end{aligned}$$

Waar $P_{null(AD)}(p_v)$ de projectie van p_v is in de nulruimte van AD en $P_{row(AD)}(p_v)$ de projectie in de rijruimte.

Nu we explicite uitdrukkingen hebben gevonden voor de zoekrichtingen Δx en Δs kunnen we het primal-dual Dikin-type algoritme beschrijven:

Primal-dual Dikin-type algoritme

Parameters
ϵ is de *accuracy* parameter
α is de stapgrootte
Input
$x_0 > 0, s_0 > 0$ de beginvectoren
zodanig dat $Ax_0 = b$ en $y_0 = A^{-T}c - s_0 > 0$
begin
$x = x_0, s = s_0$
while $x^T s > \epsilon$ **do**
$\Delta x = P_{null(AD)}(p_v)$
$\Delta s = P_{row(AD)}(p_v)$
$x = x + \alpha\Delta x$
$s = s + \alpha\Delta s$
end
end

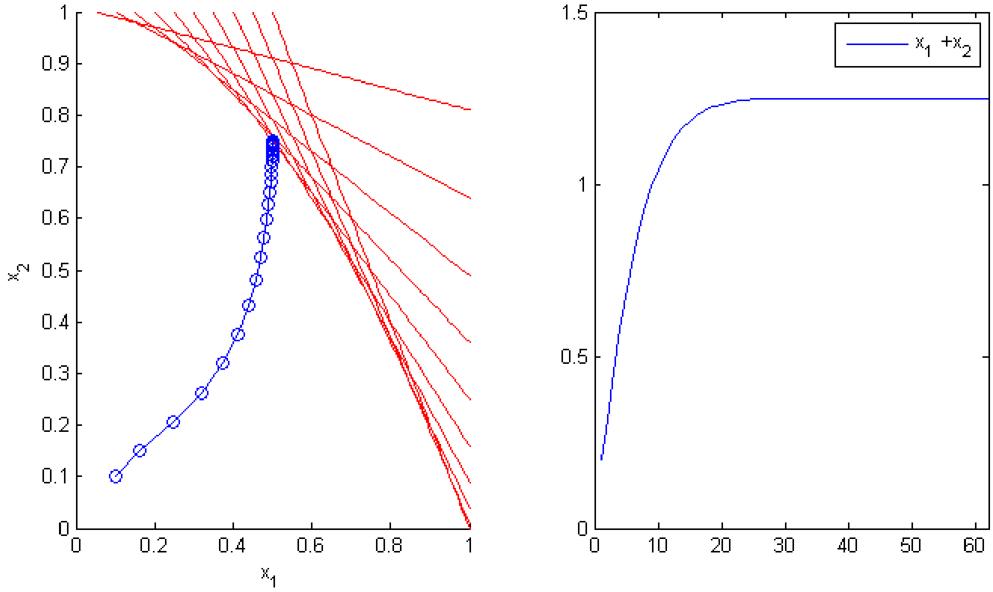
Voor $\alpha = 1/15\sqrt{15}$ hebben Janssen, Roos en Terlaky in [3], dat de methode een uitvoerbaar primal-dual paar (x, s) geeft zodat $x^T s \leq \epsilon$ in tijd $O(n \ln \frac{(x_0)^T s_0}{\epsilon})$. Verder noemen ze $x = x + \Delta x, s = s + \Delta s$ een *Dikin-stap* en α een dempingsfactor voor deze stappen.

2.3 Implementatie in Matlab

Het primal-dual Dikin-type algoritme is geïmplementeerd in Matlab en kan gevonden worden in de appendix C.4. We bekijken eerst als controle op correcte implementatie, net als by het affine scaling algortime, eerst het volgende LP-probleem:

$$\begin{array}{rl} \text{maximaliseer} & x_1 + x_2 \\ \text{onderhevig aan} & 2px_1 + x_2 \leq p^2 + 1 \\ & x \geq 0 \\ & \text{met } p = 0, 0.1, \ldots, 0.9, 1 \end{array}$$

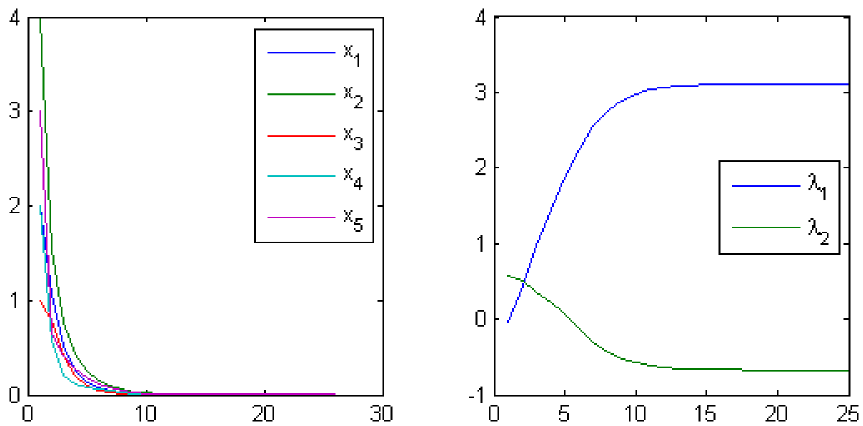
Nu nemen we net als bij het affine scaling algoritme $\alpha = 0.5$ en $\epsilon = 0.0001$. We krijgen dan de volgende oplossing:



Figuur 9: Primal-dual algoritme voor LP probleem

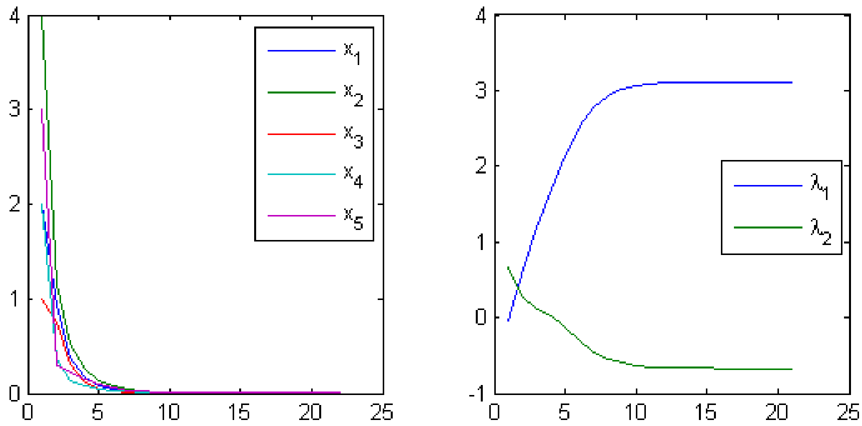
We zien duidelijk dat het algoritme een ander pad volgt dan het affine scaling algoritme. Verder heeft het primal-dual algoritme ook meer iteraties nodig, namelijk 62.

Maar nu is de vraag of het primal-dual algoritme net als het affine scaling algoritme chaos vertoont. Eerst kijken we naar het probleem van Castillo en Barnes. Dat geeft voor $\alpha = 0.86$ de volgende oplossing:



Figuur 10: Primal-dual algoritme voor probleem Castillos en Barnes $\alpha = 0.86$

We zien dus dat het algoritme in zowel de primale als de duale variabelen nog convergeert. Zelfs als we nu $\alpha = 0.999$ krijgen we nog convergentie.



Figuur 11: Primal-dual algoritme voor probleem Castillos en Barnes $\alpha = 0.999$

Het primal-dual algoritme vertoont dus geen chaos bij het probleem van Castillo en Barnes, maar dat sluit niet uit dat er geen andere problemen zijn waarvoor het primal-dual Dikin-type algortime wel chaos vertoont. Kees Roos heeft een vertaling gemaakt van het primal-dual Dikin-type algoritme naar een eenvoudig iteratief proces. Dit noemen we het Dikin proces. Het Dikin proces is afgeleid van het iteratieve proces dat het dualiteitsgat ondergaat tijdens het primal-dual Dikin-type algoritme. In hoofdstuk 3 zullen we hier naar gaan kijken.

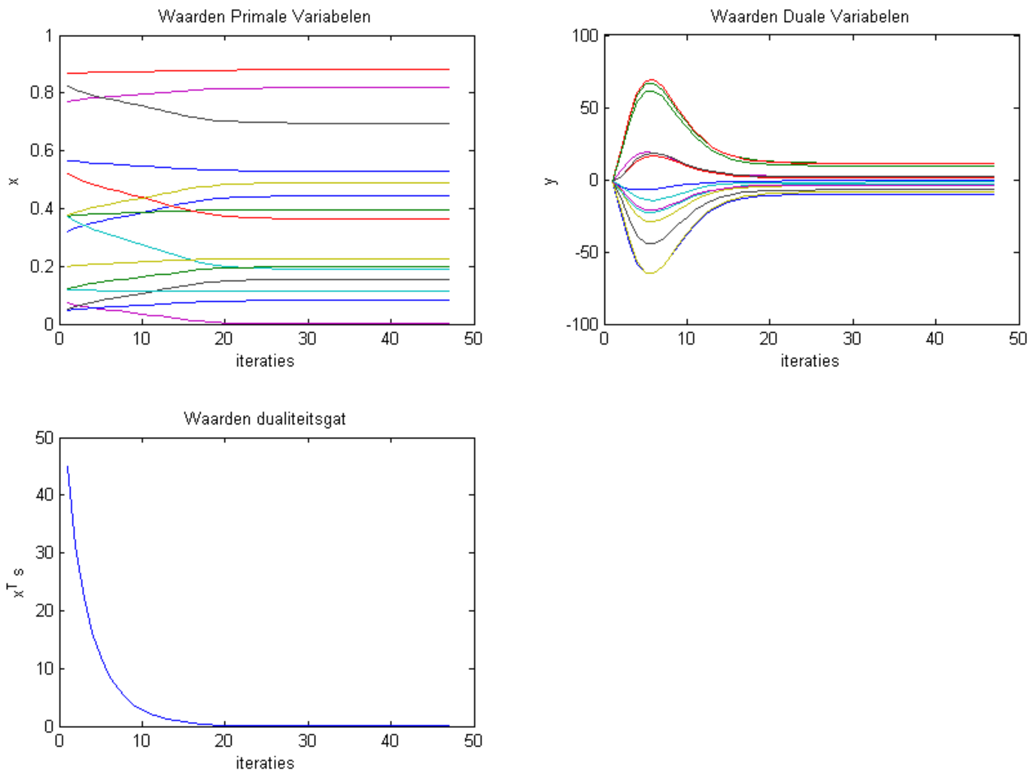
2.4 Willekeurige LP-problemen

Voordat we gaan kijken naar het Dikin proces, zullen we kijken of we zelf een LP-probleem kunnen vinden, waarvoor het algoritme chaos vertoont. We proberen dit door willekeurige LP-problemen te gebruiken met m variabelen en n randvoorwaarden. Ons probleem ziet er als volgt uit:

$$\begin{array}{rl} \text{minimaliseer} & c^T x \\ \text{onderhevig aan} & Ax = b, \\ & x \geq 0 \end{array}$$

waarbij c_i en a_{ij} getrokken worden uit een uniforme verdeling voor alle $i \in \{1, \ldots, m\}$ en $j \in \{1, \ldots, n\}$. Verder nemen we $b = A\tilde{x}$ waarbij $\tilde{x}$ de beginvector is, waarbij alle elementen x_i ook uit een uniforme verdeling zijn getrokken.

Door nu vele malen een probleem te simuleren en vervolgens naar de primale variabelen, de duale variabelen en het dualiteitsgat te kijken, hopen we chaotisch gedrag waar te nemen. Hieronder zien we de uitvoer van één simulatie.



Figuur 12: Waarden van het algoritme voor een willekeurig LP-probleem $\alpha = 0.95$

In iedere uitvoering van het probleem zien we geen chaotisch gedrag. Zelfs al nemen we waarden van α in de beurt van 1, dan nog nemen we in geen van de drie variabelen chaotisch gedrag waar. Het lijkt er dus op dat het primal-dual Dikin-type algoritme geen chaotisch gedrag vertoont of in ieder geval niet met kans 1!

3 Chaos in een Dikin proces

In het vorige hoofdstuk hebben we het primal-dual Dikin-type algoritme van Roos gezien. Kees Roos heeft voorgesteld, gemotiveerd door het artikel van Castillo en Barnes, om het primal-dual Dikin-type algoritme te analyseren op chaos. Hij heeft het algoritme herschreven om een analyse mogelijk te maken.

De vereenvoudiging maakt gebruik van de restrictie dat de vector $\Delta x \Delta s$ coördinaatsgewijs gelijk is aan nul. Dit is een gevolg van het feit dat $\Delta x = P_{null(AD)}(p_v)$ en $\Delta s = P_{row(AD)}(p_v)$. Verder nemen we bij de vereenvoudiging ook nog steeds aan dat $x\Delta s + s\Delta x = -(x^2 s^2)/\|xs\|$. De echte vereenvoudiging ten opzichte van het algoritme zit in de schaling die we uitvoeren op de vector $x^T s$.
We schrijven het algoritme nu als een iteratief proces. Het doel van de herschrijving is om de ontwikkeling van het dualiteitsgat $x^T s$ beter te kunnen analyseren. Merk op dat we in plaats van $x^T s$ naar de puntsgewijze vermenigvuldiging xs kijken.
Laat x_k gelijk zijn aan x na de $k$de iteratie. Dan weten we dat $x_{k+1} = x_k + \alpha \Delta x_k$ en $s_{k+1} = s_k + \alpha \Delta s_k$. Hierdoor geldt:

$$x_{k+1}s_{k+1} = (x_k + \alpha \Delta x_k)(s_k + \alpha \Delta s_k) = x_k s_k + \alpha(x_k \Delta s_k + s_k \Delta x_k) + \alpha^2 \Delta x_k \Delta s_k$$

Door onze twee vereenvoudigingen weten we nu dat:

$$x_{k+1}s_{k+1} = x_k s_k + \alpha(x_k \Delta s_k + s_k \Delta x_k) = x_k s_k - \alpha \frac{x_k^2 s_k^2}{\|x_k s_k\|} = x_k s_k \left(e - \alpha \frac{x_k s_k}{\|x_k s_k\|} \right) \qquad (3.1)$$

Waarbij e de vector is met alleen enen. Om nu te garanderen dat $x_{k+1} > 0$ en $s_{k+1} > 0$ moeten we zorgen dat $x_{k+1}s_{k+1} > 0$. Dit kunnen we garanderen als:

$$\alpha \leq \frac{\|x_k s_k\|}{\max x_k s_k}$$

De maximale α die we dus kunnen nemen is $\alpha_{\max} = \|x_k s_k\| / \max x_k s_k$. Nu nemen we θ de fractie α ten opzichte van $\alpha_{\max}$. Dus:

$$\theta = \frac{\alpha}{\alpha_{\max}} = \alpha \frac{\max x_k s_k}{\|x_k s_k\|}$$

3.1 wordt dan:

$$x_{k+1}s_{k+1} = x_k s_k \left(e - \theta \frac{x_k s_k}{\max x_k s_k} \right)$$

Als we nu $w_k = x_k s_k$ krijgen we:

$$w_{k+1} = w_k \left(e - \theta \frac{w_k}{\max w_k} \right)$$

Het proces is invariant onder schaling. Als we w_k bijvoorbeeld γ keer zo groot nemen, dan is w_{k+1} ook γ keer zo groot. Daarom normeren we het proces en eisen we dat alle coördinaten van w_k kleiner of gelijk zijn. Door nu de grootste coördinaat altijd gelijk aan 1 te nemen, krijgen we het volgende proces:

$$\bar{w}_{k+1} = \frac{\bar{v}_k}{\max \bar{v}_k}, \quad \bar{v}_k = \bar{w}_k(e - \theta \bar{w}_k)$$

Dit proces noemen we het Dikin proces.

3.1 Dikin Proces

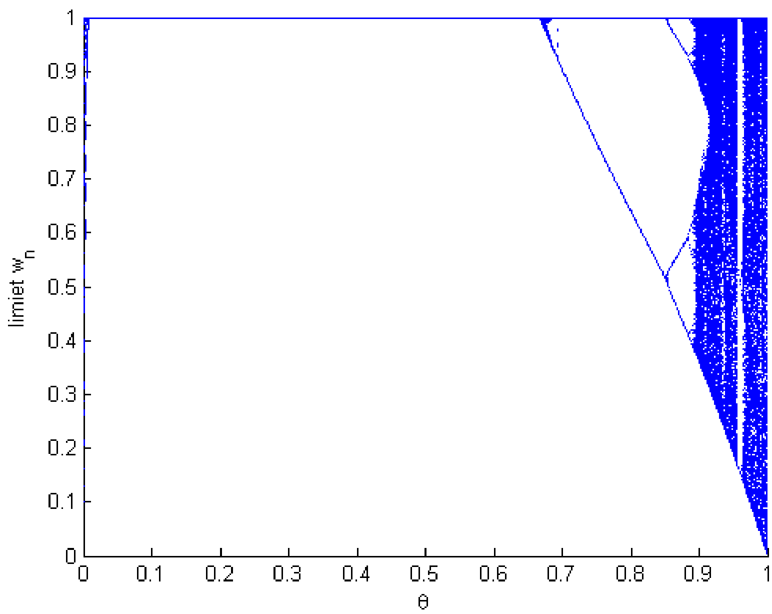
Nu kijken we of het Dikin proces chaotisch gedrag vertoont. Het Dikin proces is als volgt gedefinieerd:

$$\bar{v}_k = \bar{w}_k * (\bar{e} - \theta \bar{w}_k), \qquad \bar{w}_{k+1} = \frac{\bar{v}_k}{\max \bar{v}_k}, \qquad k = 1, 2, \dots \tag{3.2}$$

Hier zijn $\bar{w}_k$, $\bar{v}_k$ en $\bar{e}$ vectoren van lengte n en $*$ is de puntsgewijze vermenigvuldiging.

Het eerste deel van vergelijking (3.2) lijkt erg om het ecologische model, dat we eerder hebben besproken. Het tweede deel van de vergelijking normaliseert de vector waardoor hij maximaal 1 is in z'n grootste term.

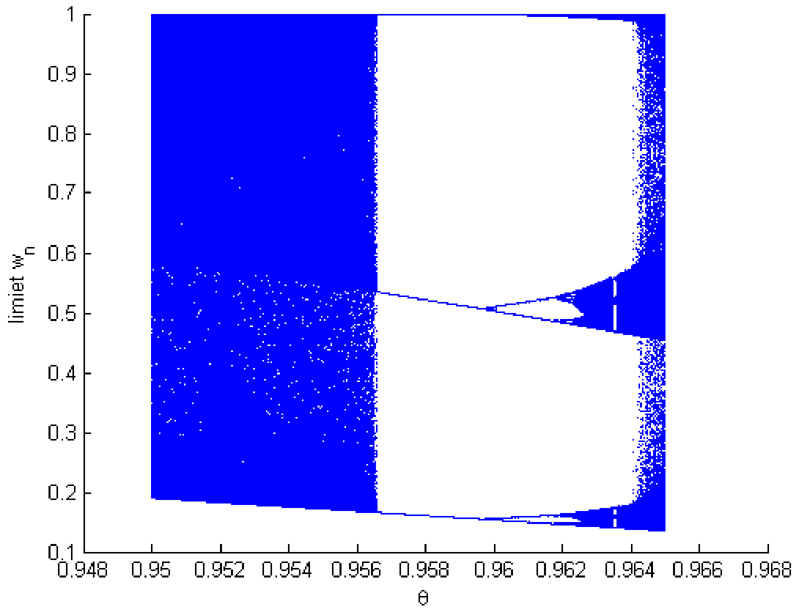
Het Dikin proces is gemodelleerd in matlab. De code is te vinden in de bijlagen. Hierbij hebben we $n = 12$ genomen en variëren we θ tussen 0 en 1. We kiezen een willekeurige waarde voor $\bar{w}_1$ tussen 0 en 1. We laten het proces lopen en tot slot plotten we de laatste a waarden per θ van $\bar{w}_k$ in een grafiek.



Figuur 13: Limiet waarden van $\bar{w}_k$

Bij een zeer kleine waarde van θ zien we dat het proces niet snel genoeg convergeert, dus dat de waarden weinig afwijken van de willekeurig gekozen waarde voor $\bar{w}_1$. Tot ongeveer $\theta \approx 0.66$ convergeert het proces naar één waarde. Vanaf dan gaat het proces periode 2 vertonen en vanaf $\theta \approx 0.85$ periode 4 of meer .

Als we nu verder kijken naar dit proces, zien we dat we dus door $\theta > 0.88$ te nemen we de periodeverdubbelende weg naar chaos krijgen. Verder zien we ook dat rond $\theta = 0.95$ er zich een interval bevind, waarbij de limiet waarden een periode 3 hebben. Om dit beter te kunnen zien maken we nu nog een simulatie met $\theta \in [0.95, 0.965]$.



Figuur 14: Limiet waarden van $\bar{w}_k$

3.2 Periode drie in het Dikin proces

We zien dus dat voor $\theta \approx 0.958$ de waarden van $\bar{v}_k$ lijken te convergeren naar 3 getallen, oftewel het Dikin proces convergeert voor $\theta \approx 0.958$ naar een punt van periode 3. We hebben in hoofdstuk 1 de stelling van Yorke gezien: periode drie impliceert chaos. Het is daarom ook interessant om deze punten verder te analyseren en te zien wat hier gebeurd.

We nemen ten eerste aan dat $n = 3$. Dit mag, want berekeningen met grotere n zouden doordat we hier te maken hebben met punten van periode 3 dezelfde berekeningen opleveren. Verder weten we dat een van de drie waarden gelijk is aan 1. Voor de andere 2 waarden nemen we x en y. Stel nu dat $\bar{V}_k = [1, x, y]$, dan geldt $\bar{V}_{k+1} = [x, y, 1]$ en $\bar{V}_{k+2} = [y, 1, x]$. Verder weten we ook dat:

$$\bar{V}_{k+1} = \begin{bmatrix} \frac{1(1-\theta 1)}{y(1-\theta y)} \\ \frac{x(1-\theta x)}{y(1-\theta y)} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1-\theta}{y-\theta y^2} \\ \frac{x-\theta x^2}{y-\theta y^2} \\ 1 \end{bmatrix}$$

Hieruit volgt dat $x = \frac{1-\theta}{y-\theta y^2)}$.

$$
\begin{aligned}
\bar{V}_{k+1} &= \begin{bmatrix} \frac{1-\theta}{y-\theta y^2} \\ \frac{\frac{1-\theta}{y-\theta y^2} - \theta \frac{(1-\theta)^2}{(y-\theta y^2)^2}}{y-\theta y^2} \\ 1 \end{bmatrix} \\
\bar{V}_{k+1} &= \begin{bmatrix} \frac{1-\theta}{y-\theta y^2} \\ \frac{1-\theta}{(y-\theta y^2)^2} - \theta \frac{(1-\theta)^2}{(y-\theta y^2)^3} \\ 1 \end{bmatrix}
\end{aligned}
$$

Om nu de punten van periode 3 te bepalen moeten we de volgende vergelijking oplossen:

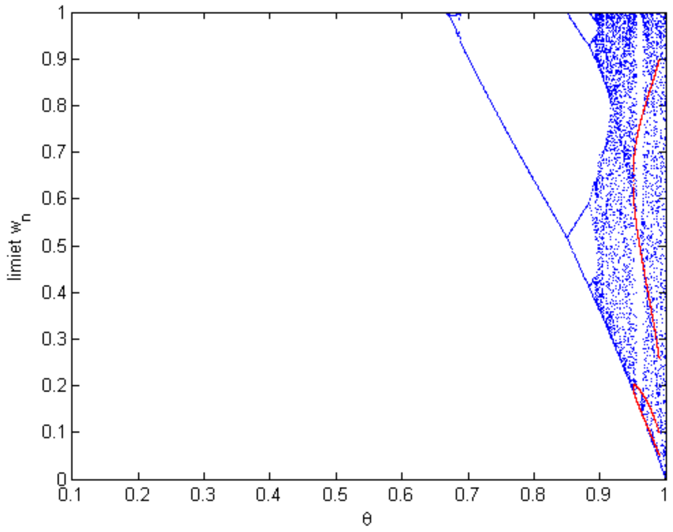
$$
\begin{aligned}
y &= \frac{1-\theta}{(y-\theta y^2)^2} - \theta \frac{(1-\theta)^2}{(y-\theta y^2)^3} \\
y(y-\theta y^2)^3 &= (y-\theta y^2)(1-\theta) - (1-\theta)^2 \\
y(y^3 - 3\theta y^4 + 3\theta^2 y^5 - \theta^3 y^6) &= (1-\theta)y - (\theta - \theta^2)y^2 - \theta^3 + 2\theta^2 - \theta \\
\theta^3 - 2\theta^2 + \theta - (1-\theta)y + (\theta - \theta^2)y^2 + y^4 - 3\theta y^5 + 3\theta^2 y^6 - \theta^3 y^7 &= 0
\end{aligned}
$$

Als we nu deze oplossen, krijgen we als oplossingen 1 en de oplossingen van de volgende vergelijking:

$$
\begin{aligned}
&\theta^3 z^6 + (-3\theta^2 + \theta^3)z^5 + (3\theta - 3\theta^2 + \theta^3)z^4 + (-1 + 3\theta - 3\theta^2 + \theta^3)z^3 \\
&\quad + (-1 + 3\theta - 3\theta^2 + \theta^3)z^2 + (-1 - 2\theta + 2\theta^2 + \theta^3)z + \theta - 2\theta^2 + \theta^3 = 0 \qquad (3.3)
\end{aligned}
$$

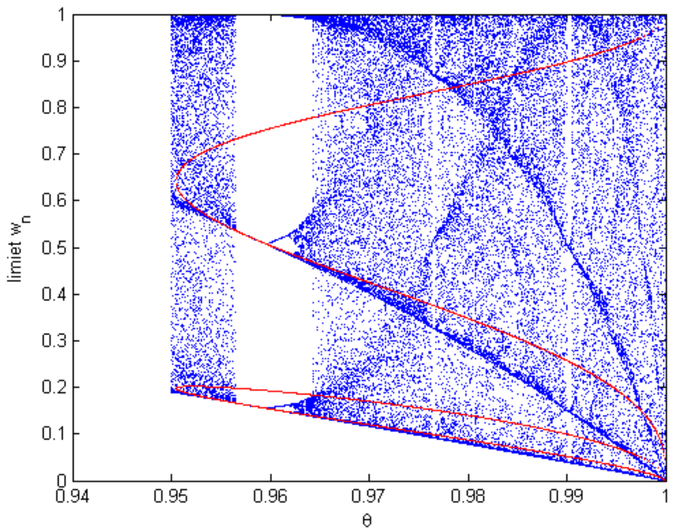
met $z \in \mathbb{C}$. Vergelijking 3.3 kan niet analytisch worden opgelost en geeft waarden in het complexe vlak. Wij zijn echter alleen geïnteresseerd in de reële oplossingen. Om deze te berekenen en zichtbaar te maken hebben we een programma in Matlab geschreven, dat eerst de waarden van z uit 3.3 berekend. Vervolgens zoekt dit programma alleen de reële oplossingen en plot deze in het Feigenbaum diagram van het Dikin proces. Het programma is te vinden in appendix C.5.

Het Feigenbaum diagram, dat het programma produceert, ziet er als volgt uit:



Figuur 15: Limiet waarden met punten van periode 3

We zien dat er geen punten van periode drie zien, wanneer $\theta < 0.95$. Interessant is het dus om beter te kijken naar $\theta \in [0.95, 1]$.



Figuur 16: Limiet waarden met punten van periode 3

We zien dat er vanaf ongeveer $\theta = 0.952$ we altijd twee punten hebben van periode 3. Één daarvan is op z'n hoogst aantrekkend, de ander altijd afstotend.

4 Conclusie en aanbevelingen

In dit project is het Dikin proces, dat is gedefinieerd door Kees Roos, onderzocht op dynamisch gedrag. Het blijkt dat het proces chaos vertoont en dat route naar chaos in het Feigenbaum diagram lijkt op de bekende route naar chaos in de kwadratische familie, via periodeverdubbeling. Ook is een punt van periode 3 aangetroffen bij parameter waarde $\theta \approx 0.96$.
Vervolgens is onderzocht of het primal-dual Dikin-type algoritme, waarvan dit proces is afgeleid, ook chaotisch gedrag vertoont. Dit blijkt verrassenderwijs niet het geval te zijn. Hiervoor is de volgende verklaring mogelijk. Dat dit komt doordat het Dikin proces een schaling uitvoert. Het chaotische gedrag zit dan als het ware in de richting waarin het optimum wordt benaderd. Welke verklaring de juiste is, kan nog onderzocht worden.

A Tussenwaardestelling

Stelling A.1 (Tussenwaardestelling) *Laat $F : [a, b] \to \Re$ continu zijn en s een getal waarvoor geldt $F(a) < s < F(b)$, dan bestaat er een getal $c \in [a, b]$ zodanig dat $F(c) = s$.*

Bewijs Laat L de set zijn van alle getallen x waarvoor geldt: $F(x) < s$. L is geen lege verzameling aangezien $F(a) < s$ en S is van bovenaf begrensd door $F(b)$. Door de compleetheid van de reële getallen volgt hieruit dat het supremum $c = \sup S$ bestaat. Nu gaan we bewijzen dat $F(c) = s$.
Neem nu eerst aan de $F(c) < s$, dan is $F(x) - s < 0$. Omdat F continu is, is er een $\delta > 0$ zodat $|F(x) - F(c)| < \epsilon$ als $|x - c| < \delta$. Neem nu $\epsilon = s - F(c)$, dan geldt dat $|F(x) - F(c)| < s - F(c)$, als $|x - c| < \delta$. Dus is $F(x) < F(c) + (s - F(c)) = s$ als $x \in [c - \delta, c + \delta]$. Dus zijn er getallen $x > c$ waarvoor geldt $F(x) < s$. Dit is in tegenspraak met onze definitie dat $c = \sup S$. Neem nu aan dat $F(c) > s$, dan is $F(x) - s > 0$. Omdat F continu is, is er een $\delta > 0$ zodat $|F(x) - F(c)| < \epsilon$ als $|x - c| < \delta$. We nemen nu $\epsilon = F(c) - s$, dan geldt dat $|F(x) - F(c)| < F(c) - s$, als $|x - c| < \delta$. Dus $F(x) > F(c) - (F(c) - s) = s$ als $x \in [c - \delta, c + \delta]$. Dus zijn er getallen $c - \delta < c$ waarvoor geldt dat ze een bovengrens zijn voor S. Echter we hebben aangenomen dat $c = \sup S$, dus dit is weer een tegenspraak. Hieruit volgt dus dat er een c is, waarvoor geldt $F(c) = s$. $\square$

Stelling A.2 (Tussenwaardestelling voor een interval) *Laat $F : [a, b] \to \Re$ continu zijn en $F([a, b])$ het beeld van $[a, b]$, dan is $F([a, b])$ ook een interval en bevat het óf $[F(a), F(b)]$ óf $[F(a), F(b)]$.*

Bewijs Zonder verval der algemeenheid kunnen we aannemen dat $F(a) < F(b)$. Stel dat $[F(a), F(b)]$ geen deelverzameling is van $F([a, b])$. Dan is er een $s \in [F(a), F(b)]$, waarvoor geldt dat $s \notin F([a, b])$. Echter omdat F continu is en $s \in [F(a), F(b)]$, weten we uit de Tussenwaardestelling dat er een $c \in [a, b]$ zodanig dat $F(c) = s$. Dus moet er gelden $s \in F([a, b])$. Dit is een tegenspraak met onze aanname dat $[F(a), F(b)]$ geen deelverzameling is van $F([a, b])$, dus moet $[F(a), F(b)] \subseteq F([a, b])$ $\square$

B Metriek op Σ_2

Om te laten zien dat Σ_2 een metrische ruimte is, moeten we laten zien dat er een metriek $d(\mathbf{s}, \mathbf{t})$ bestaat, die voor alle met $\mathbf{s}, \mathbf{t} \in \Sigma_2$ gedefinieerd is. We gebruiken hiervoor:

$$d(\mathbf{s}, \mathbf{t}) = \sum_{i=0}^{\infty} \frac{|s_i - t_i|}{2^i}$$

Stelling B.1 *d is een metriek op Σ_2.*

Bewijs Het is duidelijk dat $d(\mathbf{s}, \mathbf{t}) \geq 0$ voor alle $\mathbf{s}, \mathbf{t} \in \Sigma_2$ en dat daardoor ook geldt dat $d(\mathbf{s}, \mathbf{t}) = 0$ dan en slechts dan als $s_i = t_i$ voor alle i. Omdat verder geldt dat $|s_i - t_i| = |t_i - s_i|$, geldt ook dat $d(\mathbf{s}, \mathbf{t}) = d(\mathbf{t}, \mathbf{s})$. Tot slot neem $\mathbf{r}, \mathbf{s}, \mathbf{t} \in \Sigma_2$, dan geldt dat $|r_i - s_i| + |s_i - t_i| \geq |r_i - t_i|$ geldt ook dat $d(\mathbf{r}, \mathbf{s}) + d(\mathbf{s}, \mathbf{r}) \geq d(\mathbf{r}, \mathbf{t})$. Dus d is een metriek. $\square$

Merk op dat als $\mathbf{s}, \mathbf{t} \in \Sigma_2$ en wanneer $s_i = t_i$ voor $i = 0, 1, \ldots, n$, dat dan geldt:

$$\begin{aligned} d(\mathbf{s}, \mathbf{t}) &= \sum_{i=0}^{n} \frac{|s_i - s_i|}{2^i} + \sum_{i=n+1}^{\infty} \frac{|s_i - t_i|}{2^i} \\ &\leq \sum_{i=n+1}^{\infty} \frac{1}{2^i} \\ &= \frac{1}{2^n} \end{aligned}$$

Verder geldt ook dat als $s_j \neq t_j$ voor een $j \leq n$, dat dan:

$$d(\mathbf{s}, \mathbf{t}) \geq \frac{1}{2^j} \geq \frac{1}{2^n}$$

Dus als $d(\mathbf{s}, \mathbf{t}) < 1/2^n$, dan $s_i = t_i$ voor $i \leq n$.

C Matlab code

C.1 Model van May

Dit is de code gebruikt voor het ecologische model van May.

```
clear all
%number of iterations:
n=500;
%parameter K en stapgrootte voor r
K=1;
steps=400;
beginpoint=2.8;
endpoint=3.56;
interval=endpoint-beginpoint;
%beginwaarde
x0=0.1;

%a is the number of endpoints plotted by matlab
a=100;

%functie
for j=0:steps
    r(j+1)=beginpoint+interval*(1/steps)*j;
    x=zeros(1,n);
    x(1)=x0;
    for i=2:n
        x(i)=r(j+1)*x(i-1)*((1-x(i-1))/K);
    end
    X(:,j+1)=x(n-(a-1):n);
end

hold on
syms ro;
per=solve('ro*((ro*(x-x^2))-(ro*(x-x^2))^2)-x');
for i=0:steps
    per2=double(subs(per,ro,sym(r(i+1))));
    for k=1:length(per2)
        if (imag(per2(k))==0)
            plot(r(i+1),per2(k),'r.','MarkerSize',2)
        end
    end
end

plot(r,X,'b.','MarkerSize',3)
axis([beginpoint,endpoint,0.3,1])
xlabel('r')
ylabel(texlabel('limiet x_{n}'))
```

C.2 Periodieke punten in het model van May

Deze Matlab code plot de periodieke punten van periode 1,2 en 4.

```
hold on
syms ro;
per=solve('ro*((ro*(x-x^2))-(ro*(x-x^2))^2)-x');
for i=0:steps
    per2=double(subs(per,ro,sym(r(i+1))));
    for k=1:length(per2)
        if (imag(per2(k))==0)
            plot(r(i+1),per2(k),'r.','MarkerSize',2)
        end
    end
end

plot(r,X,'b.','MarkerSize',3)
axis([beginpoint,endpoint,0.3,1])
xlabel('r')
ylabel(texlabel('limiet x_{n}'))
```

C.3 Affine Scaling Algoritme

Dit is de code voor het affine scaling algoritme. Merk op dat als men in het probleem van Castillo en Barnes goed de periodicitiet van λ_1 en λ_2 wil laten zien, men een %-teken moet zetten voor `&&abs(x'*s)>eps'`.

```
%Affine Scaling Algorithm
eps=0.0001;
t=0.7;

x=x_0;
s=s_0;
i=0;
X=x_0;%nodig voor plot
S=s_0
L=[];
while (i<200)&&abs(x'*s)>eps
    D=diag(x);
    l=(A*D^2*A')^-1*A*D^2*c;
    s=c-A'*l;
    v=max(D*s);
    size(v);
    x=x-t*(D^2*s)./v;
    disp([c'*x, x'*s])
    i=i+1;
    X=[X,x];%nodig voor plot
    L=[L,l];
end
i
real(x)
y=linsolve(A',(c-s))

c'*x
b'*y

disp(['Waarde primal', '  Waarde Dual '])
disp( [c'*x , b'*y])
disp('Number of iterations')
disp(i)
```

C.3.1 LP-Probleem in 2D

Dit is de matlab code voor het volgende probleem:

$$\begin{array}{rl} \text{maximaliseer} & x_1 + x_2 \\ \text{onderhevig aan} & 2px_1 + x_2 \leq p^2 + 1 \\ & x \geq 0 \\ & \text{met } p = 0, 0.1, \ldots, 0.9, 1 \end{array}$$

```
clc
%Voorbeeld van een LP probleem in 2D
%http://en.wikipedia.org/wiki/Karmarkar's_algorithm
c=[-1,-1,zeros(1,11)]';
p=[0:0.1:1]'
A=[2*p.*ones(11,1),ones(11,1),eye(11)];
b=p.^2+1;

%De beginoplossing
x_0=[0.1,0.1,(b'-2*0.1*p'-0.1)]';
n=size(A);
m=n(1);
n=n(2);
y_0=-0.1*ones(11,1);
s_0=c-A'*y_0;

%Kies of Affine Scaling algoritme(Affine_ScalingvFinal) of
%Primal-Dual Dikin type algoritme(Primal_DualvFinal)
Affine_ScalingvFinal

%plot
%convergentie plotten
z=X'*-c;
subplot(1,2,1);
hold on
y=[0:0.01:1];
for p=0:0.1:1
    plot(y,p^2+1-2*p*y,'Color','red')
end
plot(X(1,:),X(2,:),'-o')
axis([0,1,0,1]);
xlabel('x_1')
ylabel('x_2')

subplot(1,2,2);plot([z])
axis([0,i,0,1.5])
legend('x_1 +x_2')
```

C.3.2 LP-Probleem van Castillo en Barnes

Dit is de Matlab code voor het probleem dat Castillo en Barnes voorstellen in [4]:

$$\begin{aligned} \text{minimaliseer} \quad & 10x_1 + 10x_2 + 5x_3 + x_4 + x_5 = 0 \\ \text{onderhevig aan} \quad & x_1 + 2x_2 - 3x_3 - 2x_4 - x_5 = 0 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

```
clc
%Probleem van Castillo & Barnes
c=[10,10,5,1,-1]';
A=[1,2,-3,-2,-1;-1,2,-1,-1,-1];
b=[0,0]';

%De beginoplossing
x_0=[2,4,1,2,3]';
n=size(A);
m=n(1);
n=n(2);
y_0=[0,0]';
s_0=c-A'*y_0;

%Kies of Affine Scaling algoritme(Affine_ScalingvFinal) of
%Primal-Dual Dikin type algoritme(Primal_DualvFinal)
Primal_DualvFinal

%plot
%convergentie plotten
z=X'*c;
[X',z];
subplot(1,2,1);plot([X'])
legend('x_1','x_2','x_3','x_4','x_5')
subplot(1,2,2);plot([L'])
legend('\lambda_1','\lambda_2','Location', 'East')
```

C.4 Primal-dual Dikin-type algoritme

Dit is de code voor het primal-dual Dikin-type algoritme. Merk op dat als men in het probleem van Castillo en Barnes of in de random problemen goed de periodicitiet van duale variabelen wil laten zien, men een %-teken moet zetten voor

```
%Primal-Dual Affine Scaling Algorithm
eps=0.0001;
%alp=1/(15*sqrt(n));
alp=0.95

x=x_0;
s=s_0;
i=0;
X=x_0;%nodig voor plot
Y=y_0;
S=s_0;
while ((abs(x'*s)>eps)&&(i<500))
    v=sqrt(x.*s);
    pv=-(v.^3)./sqrt(sum(v.^4));
    d=sqrt(x./s);
    P_AD=null(A*diag(d));
    Q_AD=rref((A*diag(d)))';
    dx=d.*((P_AD*P_AD')*pv);
    zero=((P_AD*P_AD')*v)'*((Q_AD*inv(Q_AD'*Q_AD)*Q_AD')*v); %should be zero
    ds=(1./d).*((Q_AD*inv(Q_AD'*Q_AD)*Q_AD')*pv);
    x=x+alp*dx;
    s=s+alp*ds;
    i=i+1;
    y=linsolve(A',(c-s)); %value of the dual problem
    X=[X,x];%nodig voor plot
    Y=[Y,y];
    S=[S,s];
end
real(x);


disp(['Waarde primal', '  Waarde Dual '])
disp( [c'*x , b'*y])
disp('Number of iterations')
disp(i)
```

C.4.1 Willekeurig gegenereerde LP-Problemen

Dit is de Matlab code voor de willekeurige LP-problemen.

$$\begin{array}{rl} \text{minimaliseer} & c^T x \\ \text{onderhevig aan} & Ax = b, \\ & x \geq 0 \end{array}$$

waarbij c_i, x_i en a_{ij} getrokken worden uit een uniforme verdeling voor alle $i \in \{1, \ldots, m\}$ en $j \in \{1, \ldots, n\}$ en $b = A\tilde{x}$ waarbij $\tilde{x}$ de beginvector is.

```
%De beginoplossing
x_0=rand(m,1);
b=A*x_0;

y_0=ones(n,1)*-1;
s_0=c-A'*y_0;

%Kies of Affine Scaling algoritme(Affine_ScalingvFinal) of
%Primal-Dual Dikin type algoritme(Primal_DualvFinal)
Primal_DualvFinal

%plot
%convergentie plotten
z=X'*c;
[X',z];
gap=sum(X.*S);
theta=max(X(:,1).*S(:,1));
gapnorm=X(:,1).*S(:,1)./theta;
for j=2:i
    theta=max(X(:,j).*S(:,j));
    gapnorm=[gapnorm, X(:,j).*S(:,j)./theta];
end
gapnorm=sum(gapnorm);

subplot(2,2,1);plot([X'])
title('Waarden Primale Variabelen')
xlabel('iteraties')
ylabel('x')
subplot(2,2,2);plot([Y'])
title('Waarden Duale Variabelen')
xlabel('iteraties')
ylabel('y')
subplot(2,2,3);plot([gap])
title('Waarden dualiteitsgat')
xlabel('iteraties')
```

```
ylabel('x^T s')
subplot(2,2,4);plot([gapnorm])
title('Waarden genormaliseerd dualiteitsgat')
xlabel('iteraties')
ylabel('x^T s /(max x_k s_k)')
```

C.5 Periode drie in het Dikin proces

De volgende Matlab code berekend voor het dikin proces de reële punten van periode 3 en plot deze als ze bestaan.

```
clc
clear all
format short
hold on


thetaBegin=0.95 %eerste waarde theta
thetaEnd=0.9999 %laatste waarde theta
steps=1000; %aantal stappen

for k=0:steps
    theta=thetaBegin+(thetaEnd-thetaBegin)/steps*k;

    %lost polynoom op
    pol=[theta^3,-3*theta^2+theta^3,3*theta-3*theta^2+theta^3];
    pol=[pol,-1+3*theta-3*theta^2+theta^3,-1+3*theta-3*theta^2+theta^3];
    pol=[pol,-1+2*theta-2*theta^2+theta^3,theta-2*theta^2+theta^3];
    s=roots(pol);

    %filtert rele oplossingen
    Y=[0,0,0];
    j=1;
    for i=1:length(s)
        if (imag(s(i))~=0)||(abs(s(i))>1)
           %do nothing
        else
            ys=s(i);
            xs=(1-theta)/(ys-theta*ys^2);
            zs=1;
            w=[xs,ys,zs];
            v=w.*(1-theta*w);
            w2=v./max(v);
            v=w2.*(1-theta*w2);
            w3=v./max(v);
            v=w3.*(1-theta*w3);
            w4=v./max(v);
            if (abs(sum(w-w4))<10^-10)
                Y(j,:)=w;
                j=j+1;
            else
                %do nothing
            end
        end
```

```
    end
    %plot oplossingen polynoom
    theta2=theta*ones(1,length(Y'));
    plot(theta2,Y','r.','MarkerSize',3)
end
```

Referenties

[1] Devaney, R.L.: An Introduction To Chaotic Dynamical Systems, second edition, Westview Press, Boulder Colorado, 2003

[2] Vanderbei, R.J.; Meketon, M.S.; Freedman, B.A.: A modification of Karmarkar's linear programming algorithm, Algorithmica, 1:395-407, 1986.

[3] Jansen, B.; Roos C.; Terlaky T.: A Polynomial Primal-Dual Dikin-Type Algorithm for Linear Programming, Mathematics of Operations Research, Vol. 21, No. 2 (May, 1996), 341-353

[4] Castillo, I., Barnes, E.R.: Chaotic Behavior of the Affine Scaling Algorithm for Linear Programming, SIAM Journal on Optimization(2001), 781-795

[5] Simple Mathematical Models with Very Complicated Dynamics, Nature 261 (1976), 459-467

[6] Gleick, J.: Chaos de derde wetenschappelijke revolutie, fourth edition, Uitgeverij Contact, The Netherlands, 59-78

[7] Banks, J., Brooks, J., Cairns, G., Davis, G., Stacey, P.: On Devaney's definition of chaos. American Mathematical Monthly 99, 4 (April 1992), 332-334