

Jan Bouwe van den Berg

Afdeling Wiskunde
Vrije Universiteit Amsterdam
janbouwe.vanden.berg@vu.nl

Chris Groothedde

Afdeling Wiskunde
Vrije Universiteit Amsterdam
c.m.groothedde@vu.nl

Ray Sheombarsing

Afdeling Wiskunde
Vrije Universiteit Amsterdam
r.s.s.sheombarsing@vu.nl

Simulatie + Contractie = Bewijs

In hoeverre is het verantwoord om de resultaten van een simulatie van een probleem als benadering van een oplossing van het probleem te interpreteren? Een van de manieren om deze vraag te benaderen is door middel van zogenaamde computer-assisted (door de computer ondersteunde) bewijsmethoden. In dit artikel presenteren Jan Bouwe van den Berg, Chris Groothedde en Ray Sheombarsing een methode die, met als uitgangspunt een zorgvuldig geproduceerde numerieke simulatie, de mogelijkheid biedt om het probleem op een wiskundig exacte manier op te lossen met behulp van een computer.

Sinds het Manhattan project in de Tweede Wereldoorlog, maar vooral vanaf de opkomst van de op transistors gebaseerde computers in de jaren zestig zijn computersimulaties niet meer weg te denken uit de wiskunde. In de toegepaste wiskunde (en ver daarbuiten) vormen simulaties vaak het eindstadium van een oplossingsstrategie voor een gemodelleerd probleem. Daarbij bestaat het model bijvoorbeeld uit een groot stelsel gekoppelde (differentiaal) vergelijkingen of een niet-lineaire partiële differentiaalvergelijking, zoals de Navier-Stokes-vergelijking voor vloeistofstromingen. In de zuivere wiskunde dienen simulaties vooral om de wiskundige te helpen haar probleem te visualiseren en om inspiratie op te doen.

Een cruciale vraag die bij het gebruik van computersimulaties vrijwel altijd om de hoek komt kijken is hoe betrouwbaar deze resultaten zijn. Computer-assisted bewijzen zijn bij uitstek geschikt om die betrouwbaarheid onomstotelijk vast te stellen.

De problemen die wij met onze methode zullen behandelen zijn vrijwel allemaal afkomstig uit het vakgebied van de niet-lineaire dynamica, zoals het vinden van oplossingen van differentiaalvergelijkingen of banen van iteratieve afbeeldingen. In veel situaties kan het al lonen om slechts één

of enkele speciale oplossingen te vinden, zoals periodieke oplossingen of verbindende banen. Door middel van zogenaamde *forcing theorems* kan het bestaan van dit soort speciale oplossingen gebruikt worden om stellingen te bewijzen die uitspraken doen over globale eigenschappen van het systeem, zoals de aanwezigheid van chaos of het bestaan van speciale families van oplossingen. Een beroemd voorbeeld is de stelling “periode drie impliceert chaos” voor iteraties in één dimensie. We komen daar later op terug.

De problemen in de dynamische systemen zijn eigenlijk zonder uitzondering niet-lineair. Kwalitatieve, grove beschrijvingen van robuuste eigenschappen van dit soort systemen kunnen vaak gegeven worden met behulp van topologische en variationele methoden. Bovendien kan in sommige gevallen een ‘kleine’ parameter uitgebuit worden om oplossingen in limietgevallen via asymptotische analyses te beschrijven. Maar door het niet-lineaire karakter van de problemen kunnen bewijzen van stellingen die kwantitatieve uitspraken doen over dynamische systemen voor ‘gewone’ parameterwaardes in vrijwel geen enkel geval via analytische weg gevonden worden. Daarom wordt bij bestudering van dit soort problemen, die zowel in de wiskunde zelf als in toepassing erg vaak voor-

komen, gewoonlijk teruggevallen op numerieke simulaties. Tegenwoordig kunnen zulke simulaties, dankzij hoogstaande algoritmes, software en hardware vaak eenvoudig en snel ontwikkeld en ‘gedraaid’ worden. Echter, deze simulaties leveren in de meeste gevallen geen daadwerkelijk bewijs voor het bestaan van een oplossing of een garantie dat een numerieke methode een oplossing goed benadert.

Aan de basis van de methode die we in dit artikel beschrijven ligt het idee dat dit soort numerieke simulaties, ondanks deze beperkingen, toch gebruikt kunnen worden om een wiskundig exact bewijs te leveren. Om precies te zijn zullen wij laten zien dat in de buurt van een numeriek gesimuleerde ‘oplossing’ (dat wil zeggen, een uitkomst van een numerieke berekening waarvan we hopen/verwachten dat het een goede benadering van een oplossing is) een unieke wiskundig exacte oplossing bestaat. Sterker nog, onze methode vertelt ons expliciet hoe dicht de ware oplossing bij de gesimuleerde benadering ligt en hoe groot de omgeving is waarin deze uniek is.

De allereerste stap van de methode is om het probleem, of het nu afkomstig is uit bijvoorbeeld een stelsel gewone differentiaalvergelijkingen of bestaat uit een partiële differentiaalvergelijking met randvoorwaarden, te herformuleren als

$$F(x) = 0, \quad (1)$$

waarbij F een afbeelding is tussen, laten we zeggen, twee vectorruimten. Die vectorruimten zullen in het algemeen *oneindig* dimensionaal zijn, en dan kiezen we doorgaans Banachruimten, maar in eenvoudige gevallen kunnen ze ook eindig-dimensio-

naal zijn. Het is bijzonder lastig om met een computer gelijkheden zoals in (1) te controleren. In plaats daarvan zullen we daarom gebruik maken van een dekpuntstelling. Dit doen we door het probleem $F(x) = 0$ om te schrijven naar een dekpuntprobleem door middel van een variant van de methode van Newton. In de methode die we hier beschouwen, vervult de Banach contractiestelling de rol van dekpuntstelling, maar er bestaan ook aanpakken die werken met de dekpuntstellingen van Brouwer en Schauder, zie bijvoorbeeld [5].

We zullen vervolgens laten zien dat op een goed gekozen omgeving van de numeriek gesimuleerde oplossing (benadering) aan de voorwaarden van de dekpuntstelling van Banach is voldaan. Dan garandeert deze stelling namelijk het bestaan en de uniciteit van een (echte) oplossing. Via pen-en-papier-wiskunde zijn de voorwaarden van de Banach-contractiestelling te formuleren als een *eindig aantal ongelijkheden*, waarin de numerieke simulatie en analytische eigenschappen van het probleem verenigd worden. Om te laten zien dat er aan deze ongelijkheden voldaan wordt, zal er gewerkt moeten worden met de data die eerder door de computer geproduceerd zijn. Omdat gesimuleerde resultaten honderden of zelfs duizenden datapunten bevatten, is het praktisch om deze ongelijkheden met een computer te verifiëren. Het is cruciaal dat *ongelijkheden* (die contractie garanderen) met een computer veel makkelijker te controleren zijn dan gelijkheden zoals (1). Niettemin betekent het dat de computer op een *rigoureuze* manier (zonder afrondfouten, of althans met rigoureuze controle van de afrondfouten) met de data moet omgaan.

Al sinds de jaren tachtig worden dit soort *computer-assisted proofs* toegepast binnen het gebied van dynamische systemen, zowel voor discrete problemen (zoals iteraties van afbeeldingen) als voor continue problemen (zoals gewone en partiële differentiaalvergelijkingen). Een bekend voorbeeld is het eerste bewijs van het Feigenbaum-vermoeden over de universaliteit van de Feigenbaum-constante [8]. Ook binnen de continue dynamische systemen werd in de jaren negentig al gebruik gemaakt van *computer-assisted proofs*, hetgeen uitmondde in het bewijs van het bestaan van de chaotische attractor in het beroemde Lorenz-systeem [12, 13].

Net als bij de methode die we in dit artikel beschrijven berusten de bewijzen van bovengenoemde historische stellingen op het verifiëren van ongelijkheden. Om dit met een computer te kunnen doen wordt gebruik gemaakt van interval-aritmetiek. Interval-aritmetiek is een methode waarmee computersoftware op een geautomatiseerde manier afrondfouten kan bijhouden, zodat ongelijkheden op een exacte manier gecontroleerd kunnen worden. Het belang van afrondfouten moet wel enigszins gerelativeerd worden. In de praktijk vallen ze in het niet in vergelijking met de analytische schattingen die vooral bij oneindig-dimensionale problemen de crux (en het struikelblok) voor het oplossen van het probleem zijn. Niettemin is het essentieel dat naast het deel van het bewijs dat door een brein van vlees en bloed is geconstrueerd, ook het deel van het bewijs dat aan een brein van silicium is uitbesteed foutloos is. Daarom besteden we in de volgende paragraaf kort aandacht aan de vraag hoe we door toepassing van interval-aritmetiek de computer kunnen inschakelen om ongelijkheden wiskundig exact te controleren.

De laatste jaren groeit de interesse in *computer-assisted proofs* in de dynamische systemen snel. Dit komt vooral doordat de ontwikkelingen in algoritmen, software en hardware ervoor gezorgd hebben dat we niet langer een supercomputer en veel geduld nodig hebben om de berekeningen en bewijzen uit te voeren, maar met een laptop al een heel eind komen. Hierdoor kunnen nieuwe ideeën veel makkelijker uitgeprobeerd worden, en gaan de ontwikkelingen in het vakgebied overeenkomstig sneller. Aan het eind van dit artikel geven we enkele voorbeelden van problemen waar tegenwoordig de aandacht naar uitgaat (zie de laatste paragraaf).

Interval-aritmetiek

Een van de aspecten van een *computer-assisted proof* is de uitdaging dat computers niet zomaar kunnen werken met de reële getallen, maar zich moeten beperken tot een eindig aantal (binaire representeerbare) getallen. Deze eindige verzameling getallen, die we noteren als $\mathbb{F} \subset \mathbb{R}$, worden de *floating-point* getallen of *floats* genoemd.

Het grootste nadeel van deze getallen is dat er afrondfouten optreden wanneer er bewerkingen op worden uitgevoerd (zoals

optelling, vermenigvuldiging, et cetera). Dit betekent dat, zelfs in eindig veel dimensies, simulaties en numerieke berekeningen op basis van *floating-point* getallen afwijkingen met zich mee brengen. Om toch op een rigoureuze manier om te gaan met *floats* wordt gebruikt gemaakt van zogenaamde *interval-aritmetiek*. Interval-aritmetiek is een manier om, binnen een eindige verzameling van getallen \mathbb{F} (zoals de *floats*), door stelselmatig naar boven én naar beneden af te ronden toch op mathematisch gegarandeerde resultaten uit te komen.

Als voorbeeld bekijken we de operatie vermenigvuldiging $*: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. Doorgaans zal het niet zo zijn dat uit $a, b \in \mathbb{F}$ volgt dat ook $a * b \in \mathbb{F}$ geldt. In plaats van de gewone vermenigvuldiging gebruiken processoren dus een andere, numerieke, vermenigvuldiging $\otimes: \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{F}$. Deze numerieke vermenigvuldiging wordt door computerchips zo uitgevoerd dat deze de ware vermenigvuldiging benadert: $|a * b - a \otimes b| < \epsilon$, waarbij ϵ de 'machine precision' wordt genoemd; gewoonlijk geldt dat $\epsilon \approx 10^{-16}$. Met andere woorden, het getal $a \otimes b$ is een afronding van het getal $a * b$. Het is echter a priori niet bekend of dit getal naar boven of naar beneden wordt afgerond, waardoor er nog meer informatie verloren gaat.

Door systematisch naar boven én naar beneden af te ronden, kunnen we dit soort verlies van informatie tegengaan. In plaats van te rekenen met losse *floats*, zullen we rekenen met intervallen met eindpunten in \mathbb{F} :

$$\mathcal{I}_{\mathbb{F}} \stackrel{\text{def}}{=} \{[a, b] \subset \bar{\mathbb{R}}: a, b \in \bar{\mathbb{F}}\}.$$

Hierin is $\bar{\mathbb{R}} \stackrel{\text{def}}{=} \mathbb{R} \cup \{-\infty, +\infty\}$ de uitgebreide reële lijn en zijn $\bar{\mathbb{F}} \stackrel{\text{def}}{=} \mathbb{F} \cup \{-\infty, +\infty\}$ de uitgebreide *floating-point* getallen. Voor elke $x \in \mathbb{R}$ kunnen we dus een interval $[a, b] \in \mathcal{I}_{\mathbb{F}}$ vinden zodat $x \in [a, b]$.

Het idee is nu om een vermenigvuldiging $\otimes: \mathcal{I}_{\mathbb{F}} \times \mathcal{I}_{\mathbb{F}} \rightarrow \mathcal{I}_{\mathbb{F}}$ tussen intervallen te definiëren, zodanig dat voor elke I_1 en I_2 in $\mathcal{I}_{\mathbb{F}}$ geldt dat

$$\begin{aligned} \{x_1 * x_2 \in \bar{\mathbb{R}}: x_1 \in I_1 \text{ en } x_2 \in I_2\} \\ \subset I_1 \otimes I_2 \in \mathcal{I}_{\mathbb{F}}. \end{aligned}$$

In de praktijk kan de ondergrens van $I_1 \otimes I_2$ gevonden worden door al de uiteinden van I_1 en I_2 met elkaar te vermenigvuldigen, deze naar beneden af te ronden (zodat we in $\bar{\mathbb{F}}$ belanden) en vervolgens het kleinste resultaat te kiezen. Analoog

kan de bovengrens gevonden worden door naar boven af te ronden en het grootste resultaat te kiezen.

Op soortgelijke wijze kunnen we de andere elementaire operaties \oplus, \ominus, \otimes : $\mathcal{I}_{\mathbb{F}} \times \mathcal{I}_{\mathbb{F}} \rightarrow \mathcal{I}_{\mathbb{F}}$ definiëren. Het op deze wijze rekenen met intervallen noemen we interval aritmetiek. Minder elementaire functies zoals \exp , \log , \sin , et cetera, kunnen met behulp van bijvoorbeeld de stelling van Taylor zo geïmplementeerd worden dat ze aan de cruciale inclusie-eigenschap

$$\begin{aligned} \{f(x) : x \in I\} \subset f(I) \in \mathcal{I}_{\mathbb{F}} \\ \text{voor alle } I \in \mathcal{I}_{\mathbb{F}} \end{aligned} \quad (2)$$

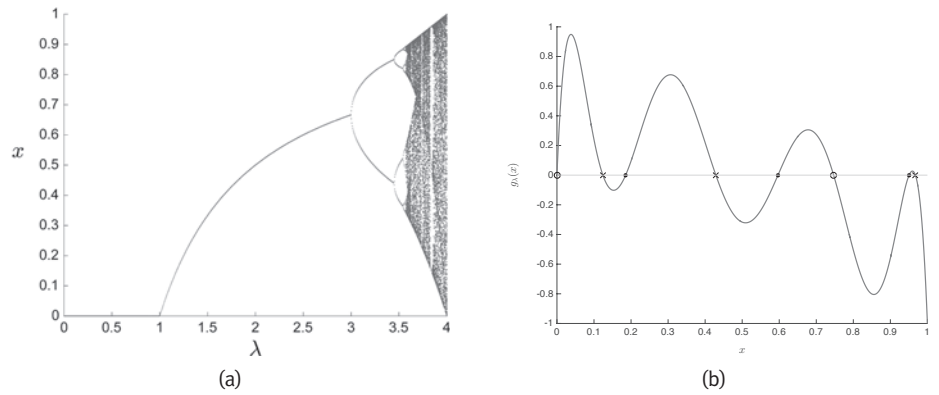
voldoen. Het is goed om te beseffen dat operaties in interval aritmetiek niet uniek gedefinieerd zijn, maar dat dit geen probleem is. Voor verschillende implementaties van interval aritmetiek kan $f(I)$ een verschillend resultaat opleveren, maar de inclusie (2) moet altijd gegarandeerd zijn. Op die manier kunnen we interval aritmetiek dus gebruiken om ongelijkheden te bewijzen.

Voor het gebruik van interval aritmetiek zijn verscheidene software-pakketten in omloop. Veel gebruikt zijn bijvoorbeeld de Intlab library voor Matlab en de Profil/BIAS library voor C++. Daarnaast bieden ook symbolische software pakketten als Maple en Mathematica ondersteuning voor interval aritmetiek.

Voorbeeld: de logistische afbeelding

Als voorbeeld bekijken we de in Figuur 1(a) gerepresenteerde logistische afbeelding $f_\lambda : [0, 1] \rightarrow [0, 1]$, gegeven door $f_\lambda(x) \stackrel{\text{def}}{=} \lambda x(1-x)$ met $\lambda \in [0, 4]$ een parameter. De dynamica $x_{k+1} = f_\lambda(x_k)$ gegenereerd door f_λ is relatief eenvoudig te beschrijven voor $\lambda \in [0, \lambda_c]$, waarbij $\lambda_c \approx 3,57$. Namelijk, alle banen convergeren naar een evenwichtstoestand of een periodieke baan. Voor $\lambda > \lambda_c$ is de dynamica niet meer eenvoudig te beschrijven: het systeem vertoont chaotisch gedrag. Maar hoe bewijs je dit?

Zoals besproken in de introductie bestaan er stellingen, *forcing theorems*, die het bestaan van speciale banen gebruiken om uitspraken te doen over het globaal gedrag van een dynamisch systeem. Een beroemd voorbeeld van een *forcing theorem* voor discrete dynamische systemen op de reële lijn is dat het bestaan van een periodieke baan van periode drie chaos impliceert [11]. Met andere woorden, we kunnen voor een



Figuur 1 (a) Het beroemde bifurcatiediagram van Feigenbaum. Het asymptotisch gedrag van de logistische afbeelding is in kaart gebracht (te zien op de verticale as) voor $\lambda \in [0, 4]$ met behulp van de computer. De numerieke simulatie doet vermoeden dat de dynamica chaotisch is voor $\lambda > \lambda_c \approx 3,57$. (b) De grafiek en nulpunten van g_λ voor $\lambda = 3,95$. De open cirkels zijn de twee evenwichtspunten van f_λ . De kruisjes en zwarte punten corresponderen met twee verschillende (niet triviale) periodieke banen van periode drie.

specifieke $\lambda > \lambda_c$ aantonen dat de dynamica van de logistische afbeelding chaotisch is door te bewijzen dat er zo'n periodieke baan van periode drie bestaat. Het is in het algemeen lastig (of zelfs onmogelijk) om een periodieke baan van een niet-lineair probleem met een pen-en-papieranalyse te vinden, en dit is typisch een situatie waarin numerieke simulaties handig zijn en *computer-assisted* bewijzen goed tot hun recht komen.

Een voor de hand liggende strategie om een periodieke baan van periode drie te vinden is het oplossen van de vergelijking $g_\lambda(x) = 0$, waarbij

$$g_\lambda(x) \stackrel{\text{def}}{=} f_\lambda(f_\lambda(f_\lambda(x))) - x,$$

zie Figuur 1(b). Om een eenvoudig voorbeeld van een *computer-assisted* bewijs te geven merken we op dat het probleem $g_\lambda(x) = 0$ kan worden opgelost met behulp van interval-aritmetiek én de tussenwaardstelling: door aan te tonen dat $g_\lambda(x_1) \in I_1$ en $g_\lambda(x_2) \in I_2$ voor zekere $x_1 < x_2$, waarbij $I_1 \subset [0, \infty]$ en $I_2 \subset [-\infty, 0]$ (of andersom), garandeert de tussenwaardstelling het bestaan van een nulpunt van g_λ in $[x_1, x_2]$. In dit speciale geval zijn die berekeningen zelfs met enig geduld nog wel met de hand te doen. Een nadeel van deze specifieke aanpak is dat de methode gebruik maakt van puur een-dimensionale argumenten die lastig te generaliseren zijn naar hoger dimensionale problemen.

Een alternatief voor de bovenstaande methode is om een nulpunt te vinden van de afbeelding $F_\lambda : [0, 1]^3 \rightarrow [0, 1]^3$ gegeven door

$$F_\lambda(x_1, x_2, x_3) \stackrel{\text{def}}{=} \begin{pmatrix} f_\lambda(x_1) - x_2 \\ f_\lambda(x_2) - x_3 \\ f_\lambda(x_3) - x_1 \end{pmatrix}. \quad (3)$$

In dit geval is het niet meer mogelijk om de tussenwaardstelling te gebruiken. In plaats daarvan combineren we het gebruik van interval-aritmetiek met de methode van Newton–Kantorovich (zie Stelling 1 verderop). Het grote voordeel van deze aanpak is dat deze makkelijk te generaliseren is naar willekeurige, en in het bijzonder oneindige, dimensies.

De methode van Newton–Kantorovich

We concentreren ons eerst op het eindig-dimensionale geval: we willen het bestaan van een nulpunt van een afbeelding $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ aantonen met behulp van de computer. De eerste stap bestaat uit het vinden van een goede numerieke benadering voor een nulpunt van F (bijvoorbeeld met de klassieke methode van Newton). De tweede stap bestaat uit een combinatie van analyse op papier en interval-aritmetiek. Het idee is om te bewijzen dat er een exact nulpunt van F bestaat in de buurt van de numerieke benadering door de condities van de stelling van Newton–Kantorovich te controleren.

In het Newton-algoritme wordt een nulpunt benaderd met het iteratieve proces $x_k = N(x_{k-1})$, $k \in \mathbb{N}$. Hierbij is $N : \mathbb{R}^n \rightarrow \mathbb{R}^n$ de klassieke Newton-afbeelding gegeven door

$$N(x) \stackrel{\text{def}}{=} x - [DF(x)]^{-1}F(x), \quad (4)$$

en $x_0 \in \mathbb{R}^n$ is een benadering (een gok) voor het nulpunt van F .

Een belangrijke aanname in de methode van Newton is dat de afgeleide DF van F in het exacte nulpunt bestaat, en continu en inverteerbaar is. Dit betekent in het bijzonder dat $DF(x)$ inverteerbaar is in een kleine omgeving van het exacte nulpunt. In deze omgeving geldt dat $F(x) = 0$ dan en slechts dan als $N(x) = x$. Met andere woorden, nulpunten van F zijn vaste punten van N en andersom.

De methode van Newton is in praktijk bijzonder effectief. Dit komt doordat de Newton-afbeelding een contractie is in een omgeving van het exacte nulpunt. Bovendien is de contractiesterkte van N 'oneindig' goed in de buurt van het exacte nulpunt. Om preciezer te zijn, een afbeelding $T: X \rightarrow X$ op een genormeerde vectorruimte X is een contractie op $U \subset X$ als er een constante $L \in (0, 1)$ bestaat zodat

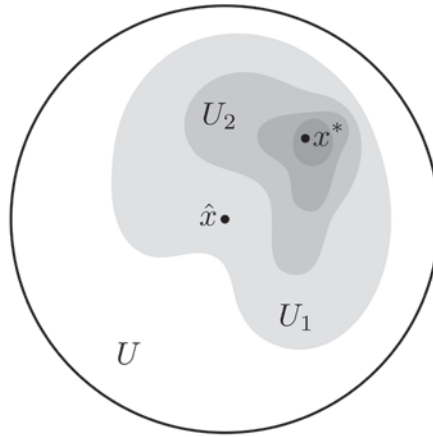
$$|T(x) - T(y)| \leq L|x - y| \quad \text{voor alle } x, y \in U,$$

waarbij $|\cdot|$ de norm op X is. De constante L wordt doorgaans een Lipschitz-constante voor T op U genoemd. In het geval dat T differentieerbaar is (en U convex) kan de middelwaardestelling worden gebruikt om een Lipschitz-constante te bepalen: kies L zodat $\|DT(x)\| \leq L$ voor alle $x \in U$, waarbij $\|DT(x)\|$ de operator-norm van de lineaire operator $DT(x)$ is. Hoe kleiner L , hoe sterker de contractie.

De dekpuntstelling van Banach stelt dat een contractie op een volledige ruimte een uniek dekpunt x^* heeft. Met andere woorden, als U volledig is en T een contractie zodat $T(U) \subset U$, dan bestaat er een uniek element $x^* \in U$ zodat $T(x^*) = x^*$. In het bijzonder geldt $\lim_{k \rightarrow \infty} T^k(x) = x^*$ voor elke startwaarde $x \in U$, zie Figuur 2.

Om nu terug te komen op de methode van Newton, die werkt zo goed omdat $DN(x^*) = 0$. Dit heeft als gevolg dat de Lipschitz-constante voor N willekeurig klein wordt als we maar dicht genoeg in de buurt van x^* zijn.

Stel nu dat we een goede numerieke benadering $\hat{x} \in \mathbb{R}^n$ voor een nulpunt van F hebben gevonden. Het idee is om deze numerieke benadering te gebruiken om te bewijzen dat er een uniek nulpunt van F bestaat in een omgeving van \hat{x} middels de Banach-contractiestelling. In het ideale geval zouden we een expliciete Lipschitz-constante voor N uitrekenen om aan te tonen dat de afbeelding een contractie is in een omgeving van \hat{x} . Echter, een dergelijke analyse van N is



Figuur 2 Een schematische weergave van de dekpuntstelling van Banach. We starten met een omgeving $U = \overline{B_r(\hat{x})}$ van een numerieke benadering \hat{x} . Omdat T een contractie is die U in zichzelf afbeeldt is $U_1 \stackrel{\text{def}}{=} T(U) \subset U$ een verzameling met een kleinere diameter. De verzameling U_1 wordt vervolgens door T afgebeeld op een nog 'kleinere' verzameling $U_2 \stackrel{\text{def}}{=} T(U_1) \subset U_1$, enzovoort. Uit het uniform kleiner worden van het rijtje verzamelingen U_k (en de volledigheid van U) volgt het bestaan van een uniek punt $x^* \in U$ (het dekpunt van T) zodat $\lim_{k \rightarrow \infty} T^k(x) = x^*$ voor elke $x \in U$.

voor niet-lineaire afbeeldingen F doorgaans moeilijk, omdat de analyse van de inverse $[DF(x)]^{-1}$ in (4) lastig is vanwege de afhankelijkheid in x .

Omdat de contractiesterkte van de Newton-afbeelding blijft toenemen naarmate we dicht in de buurt van het nulpunt komen, hebben we de vrijheid om kleine wijzigingen in N aan te brengen zonder de contractie-eigenschap te verliezen. In het bijzonder kunnen we de inverse $[DF(x)]^{-1}$ vervangen door een numerieke benadering $A \approx [DF(\hat{x})]^{-1}$ en daarmee de afhankelijkheid in x vermijden (hier evalueren we DF dus in een vast gekozen punt \hat{x}). Dit levert een nieuwe Newton-achtige afbeelding $T: \mathbb{R}^n \rightarrow \mathbb{R}^n$ op, gegeven door

$$T(x) \stackrel{\text{def}}{=} x - AF(x).$$

Onder de aanname dat de matrix A injectief is zullen de vaste punten van T precies de nulpunten van F zijn. De contractiesterkte van T is weliswaar zwakker dan die van N , maar het gebruik van deze afbeelding maakt het mogelijk om de analyse daadwerkelijk uit te voeren.

Zoals al eerder aangegeven is het doel nu om een omgeving $U = \overline{B_r(\hat{x})}$ te vinden (een gesloten bol om \hat{x} met straal $r > 0$) waarop T een contractie is. Hier hebben we twee ingrediënten voor nodig: de kwaliteit van de numerieke benadering \hat{x} en een Lipschitz-constante voor T . De kwaliteit van de numerieke benadering kan wor-

den bepaald middels een afchatting van het residu $|T(\hat{x}) - \hat{x}|$. De contractie sterkte kan worden bepaald door $\|DT(x)\|$ te begrenzen op $\overline{B_r(\hat{x})}$. Omdat we a priori niet weten voor welke straal r de afbeelding T een contractie is, houden we deze variabele in de analyse op papier. We maken vervolgens gebruik van de computer om r te bepalen.

De straal r moet namelijk enerzijds groot genoeg zijn om ervoor te zorgen dat het exacte nulpunt bevat is in $\overline{B_r(\hat{x})}$. Anderzijds moet de straal klein genoeg zijn om de contractiviteit van T te garanderen (zie Figuur 2). Door op zoek te gaan naar een zo klein mogelijke straal waarvoor T een contractie is kunnen we op een wiskundig exacte manier uitspraken doen over de kwaliteit van de numerieke simulatie. Aan de andere kant, door op zoek te gaan naar een zo groot mogelijke straal kunnen we uitspraken doen over de grootte van het gebied waarin de exacte oplossing uniek is.

De precieze formulering van de hierboven beschreven methode is samengevat in de volgende stelling en staat bekend als de geparametriseerde Newton-Kantorovich-methode (het bewijs is niet ingewikkeld, zie bijvoorbeeld [7]).

Stelling 1. *Laat T een differentieerbare afbeelding van een Banachruimte X naar zichzelf zijn, en laat $\hat{x} \in X$. Neem aan dat er een getal $Y > 0$ en een functie $Z: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ bestaan zodat*

$$|T(\hat{x}) - \hat{x}| \leq Y, \\ \sup_{v \in \overline{B_r(0)}} \|DT(\hat{x} + rv)\| \leq Z(r).$$

Als er een straal $\hat{r} > 0$ bestaat zodat

$$Y + Z(\hat{r})\hat{r} < \hat{r}, \quad (5)$$

dan is $T: \overline{B_{\hat{r}}(\hat{x})} \rightarrow \overline{B_{\hat{r}}(\hat{x})}$ een contractie.

Aangezien dekpunten van T corresponderen met nulpunten van F kunnen we dus aantonen dat F een uniek nulpunt in de buurt van de numerieke benadering \hat{x} heeft door de ongelijkheid te controleren. Het werk is nu 'verplaatst' naar het vinden van goede schattingen Y en $Z(r)$; in het bijzonder moet minstens gelden dat Z strikt kleiner dan 1 is.

We illustreren dit aan de hand van het voorbeeld uit de vorige paragraaf. We kiezen $\lambda = 3,95$ en proberen een nulpunt te vinden van F_λ , gegeven in (3). We gebruiken een benadering

$$\hat{x} = \begin{pmatrix} 0,19 \\ 0,60 \\ 0,95 \end{pmatrix}$$

van een periodieke baan van periode drie en een bijna-inverse

$$A \stackrel{\text{def}}{=} \begin{pmatrix} 0,47 & -0,62 & 0,17 \\ 0,17 & -1,54 & 0,43 \\ -0,13 & 0,17 & -0,33 \end{pmatrix}$$

De volgende stap is om de begrenzings Y en $Z(r)$ uit te rekenen.

Om het residu te begrenzen is het voldoende om in te zien dat

$$T(\hat{x}) - \hat{x} = -ADF_\lambda(\hat{x})$$

De bovenstaande term kan worden begrensd met behulp van interval aritmetiek. Dit levert $Y \stackrel{\text{def}}{=} 5,7 \cdot 10^{-3}$ als maat voor het residu (hoe kleiner hoe beter) van de numerieke benadering \hat{x} . Om de afschatting $Z(r)$ te bepalen, laat $v \in \mathbb{R}^3$ met $|v| \leq 1$, waarbij $|\cdot|$ de Euclidische norm is. Dan geeft de driehoeksongelijkheid

$$\begin{aligned} \|DT(\hat{x} + rv)\| &= \|I - ADF_\lambda(\hat{x} + rv)\| \\ &\leq \|(I - ADF_\lambda(\hat{x}))\| \\ &\quad + \|A[DF_\lambda(\hat{x} + rv) - DF_\lambda(\hat{x})]\|. \end{aligned} \quad (6)$$

De eerste term in (6) is opnieuw eenvoudig te begrenzen met behulp van interval-aritmetiek, en we vinden 0,06 als maat voor de kwaliteit (weer: hoe kleiner hoe beter) van de bijna-inverse A . De tweede term in (6) kunnen we niet direct afschatten met behulp van de computer, want die heeft een functionele afhankelijkheid van r . Er moet dus eerst enige analyse op papier worden uitgevoerd:

$$DF_\lambda(\hat{x} + rv) - DF_\lambda(\hat{x}) = -2\lambda r \begin{pmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{pmatrix}. \quad (7)$$

Omdat we hebben aangenomen dat $|v| \leq 1$, geldt

$$\left\| \begin{pmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{pmatrix} \right\| \leq 1.$$

We kunnen dus de tweede term in (6) afschatten met

$$\|A(DF_\lambda(\hat{x} + rv) - DF_\lambda(\hat{x}))\| \leq 2\lambda r \|A\|,$$

en met behulp van interval aritmetiek een begrenzing voor $\|A\|$ (en dus de bovenstaande term) uitrekenen. We vinden dat we $2\lambda\|A\|$ kunnen afschatten met 14,05. Door de bovenstaande berekeningen te combineren concluderen we dat $Z(r) \stackrel{\text{def}}{=} 14,05 r + 0,06$, en de ongelijkheid wordt

$$14,05 r^2 - 0,94 r + 5,7 \cdot 10^{-3} < 0. \quad (8)$$

Met interval-aritmetiek is dan eenvoudig aan te tonen dat voor elke r in het interval $[0,007, 0,06]$ aan (8) wordt voldaan, en daarmee is het bestaan van een unieke periodieke baan van periode drie in $\overline{B_r(\hat{x})}$ bewezen. Dus de logistische afbeelding is chaotisch voor $\lambda = 3,95$ op grond van de *forcing theorem*.

Oneindige dimensies

Zoals we gezien hebben in bovenstaand voorbeeld is het mogelijk om met behulp van Stelling 1 en interval-aritmetiek het bestaan van nulpunten van $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ aan te tonen. Om deze methode ook op differentiaalvergelijkingen toe te kunnen passen moeten we deze uitbreiden naar een *oneindig*-dimensionale context.

Als voorbeeld bekijken we het beginwaardeprobleem

$$\begin{cases} x'(t) = f(x(t)) & \text{met } t \in [0, 1], \\ x(0) = x_0. \end{cases}$$

Allereerst zullen we dit herschrijven als een probleem van de vorm $F(x) = 0$ op een oneindig-dimensionale ruimte. Een natuurlijke keuze is om de differentiaalvergelijking te integreren en dan de afbeelding $F: C([0, 1]) \rightarrow C([0, 1])$ te definiëren door

$$F(x)(t) \stackrel{\text{def}}{=} x(t) - x_0 - \int_0^t f(x(s)) ds. \quad (9)$$

Immers, $F(x)(0) = x(0) - x_0$ en $F(x)'(t) = x'(t) - f(x(t))$, en dus geldt dat $F(x) = 0$ dan en slechts dan als $x'(t) = f(t)$ én $x(0) = x_0$.

Een alternatieve strategie is om x als een machtreeks $x(t) = \sum_{k=0}^{\infty} a_k t^k$ te schrijven. Dit is een tweede natuurlijke keuze in het geval f reëel analytisch is. In dat geval kunnen we $f(x(t))$ representeren met Taylor-coëfficiënten $b_k(a_0, \dots, a_k)$, zodat

$$f(x(t)) = f\left(\sum_{k=0}^{\infty} a_k t^k\right) = \sum_{k=0}^{\infty} b_k(a_0, \dots, a_k) t^k.$$

Omdat $x'(t) = \sum_{k=0}^{\infty} (k+1) a_{k+1} t^k$, is het beginwaardeprobleem equivalent met het oneindige stelsel van algebraïsche vergelijkingen

$$(k+1) a_{k+1} = b_k(a_0, \dots, a_k), \quad k=0, 1, 2, \dots,$$

aangevuld met $a_0 = x_0$. In dit geval kunnen we F dus definiëren door

$$F: \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_k \\ \vdots \end{pmatrix} \mapsto \begin{pmatrix} a_0 - x_0 \\ a_1 - b_0(a_0) \\ 2a_2 - b_1(a_0, a_1) \\ 3a_3 - b_2(a_0, a_1, a_2) \\ \vdots \\ ka_k - b_{k-1}(a_0, \dots, a_{k-1}) \\ \vdots \end{pmatrix}. \quad (10)$$

Er zijn in de praktijk (door het kiezen van andere bases dan de machten t^k) nog vele andere keuzes voor dit soort rijtjesruimtes mogelijk. Zo wordt er ook vaak gebruikt gemaakt van de Fourier-, Legendre- of Chebyshev-basis. De precieze keuzes die worden gemaakt voor de afbeelding $F: X \rightarrow X'$ en de ruimten X en X' verschillen sterk per probleem en hangen vaak nauw samen met hoe de bijna-inverse A geconstrueerd wordt.

Al deze methoden hebben één ding gemeen: F is nu een afbeelding tussen *oneindig*-dimensionale ruimten. Op het eerste gezicht zou dat weinig moeten veranderen, want Stelling 1 is ook waar voor oneindig-dimensionale ruimten. Echter, omdat we nu te maken hebben met oneindig-dimensionale afbeeldingen is het doorgaans niet meer mogelijk om de functie F expliciet uit te rekenen. In het geval dat F een afbeelding is tussen rijtjesruimten is dit meteen duidelijk: F heeft immers oneindig veel termen. Ook het kiezen van de benadering A voor de inverse $DF(\hat{x})^{-1}$ is minder makkelijk.

Niettemin, in veel gevallen kunnen we F schrijven als $F = F^n + F^\infty$, waarbij F^n alleen afhangt van een n -dimensionale deelruimte van X en afbeeldt op een n -dimensionale deelruimte van X' . Hierbij willen we F^n zodanig kiezen dat het al het ‘interessante’ gedrag van F bevat. Als verder de staart F^∞ voldoende ‘klein’ is, dan kunnen we dus een bijna-nulpunt \hat{x} van F vinden simpelweg door een bijna-nulpunt van de eindig-dimensionale afbeelding F^n te berekenen. Voor de verdere analyse van het probleem willen we de oneindig-dimensionale component F^∞ zo kiezen dat deze analytisch ‘makkelijk’ te bestuderen is. Het maken van een juiste keuze is doorgaans moeilijk wanneer F niet-lineaire termen bevat. Er moet in alle niet-lineaire gevallen (dus eigenlijk in alle interessante gevallen) dus goed nagedacht worden over de gevolgen van de keuzes die gemaakt worden voor de functie F , de ruimtes X en X' , en de projecties op de eindige dimensionale deelruimtes (waarin de numerieke simulatie wordt uitgevoerd).

De lineaire operator $A: X' \rightarrow X$, die dienst doet als bijna-inverse van $DF(\hat{x})$, heeft in het oneindig-dimensionale geval ook een aparte behandeling nodig. Ook die moet worden gesplitst in een eindig-dimensionaal deel A^n (een matrix) en een oneindig-dimensionaal 'staartstuk' A^∞ . Aangezien F^n geacht wordt al het interessante gedrag van F te bevatten, kunnen we voor de matrix A^n een numerieke benadering voor de inverse van de Jacobiaan $DF^n(\hat{x})$ kiezen. Het oneindig-dimensionale deel A^∞ is (hopelijk) klein, maar mag niet een soort nulafbeelding zijn, want A moet wel injectief gekozen worden om correspondentie tussen dekpunten van T en nulpunten van F te garanderen.

In het voorbeeld van de afbeelding gedefinieerd in (10) is de term ka_k dominant voor grote k . Daarom kunnen we proberen om $DF(\hat{x})$ te benaderen door

$$DF(\hat{x}) \approx \left(\begin{array}{c|c} DF^n(\hat{x}) & 0 \\ \hline 0 & \begin{matrix} n \\ n+1 \\ \ddots \end{matrix} \end{array} \right). \quad (11)$$

In dit geval kunnen we A als volgt construeren. Eerst berekenen we (numeriek) een bijna-inverse voor $DF^n(\hat{x})$, namelijk $A^n \approx DF^n(\hat{x})^{-1}$. Vervolgens kiezen we

$$A \stackrel{\text{def}}{=} \left(\begin{array}{c|c} A^n & 0 \\ \hline 0 & \begin{matrix} \frac{1}{n} \\ \frac{1}{n+1} \\ \ddots \end{matrix} \end{array} \right).$$

Er geldt dan dat $ADF(\hat{x})$ een goede benadering van de identiteit is (hoe goed, dat moet expliciet worden afgeschat).

Grote delen van de eindig-dimensionale analyse uit de vorige paragraaf kunnen nu worden gegeneraliseerd naar het oneindig-dimensionale geval. Elk van de af te schatten termen omvat nu wel twee aparte analyses, namelijk die van het n -dimensionale gedeelte (dat net als in het eindig-dimensionale geval interval aritmetisch berekend kan worden) en die van het staartstuk. Dat laatste gedeelte is oneindig-dimensionaal (dus ongeschikt voor een directe computerberekening) en moet met behulp van een combinatie van functionaal-analytische ingrediënten en interval-aritmetiek afgeschat worden.

De kern van het probleem zit hem echter toch vooral in de benadering van de (oneindig-dimensionale) afgeleide $DF(\hat{x})$ door een eenvoudigere lineaire operator

(waarvan A dan weer een bijna inverse is). Het vinden van een scherpe afschatting van het verschil tussen $DF(\hat{x})$ en zijn benadering is het lastigste gedeelte van de hele analyse. Bovendien is het helaas niet voor elk probleem zo eenvoudig om het probleem op een diagonaal dominante manier te formuleren (zoals in ons voorbeeld). En zelfs als dat wel kan, is het nog maar de vraag of er een eindig-dimensionale projectie van het probleem gevonden kan worden die groot genoeg is om het essentiële gedrag van het oneindig-dimensionale niet-lineaire probleem te 'vangen' (zodat we schattingen kunnen vinden die aan de voorwaarden van Stelling 1 voldoen), maar die anderzijds klein genoeg is om mee te kunnen rekenen op een computer. Tot slot van dit artikel beschrijven we nog enkele toepassingen waarin het gelukt is om deze uitdagingen het hoofd te bieden.

Toepassingen

In deze laatste sectie geven we twee (omwille van de leesbaarheid relatief eenvoudige) voorbeelden van toepassingen. Er is een heel scala aan problemen uit de dynamische systemen waar momenteel aan gewerkt wordt: evolutionaire partiële differentiaalvergelijkingen [1], delay vergelijkingen [9], sterk indefiniete problemen [6], verbindende banen [2,10], domein decompositie [4], driedimensionale periodieke patronen, en natuurlijk het ontwikkelen van algemeen bruikbare software.

Periodieke banen in het Lorenz-systeem

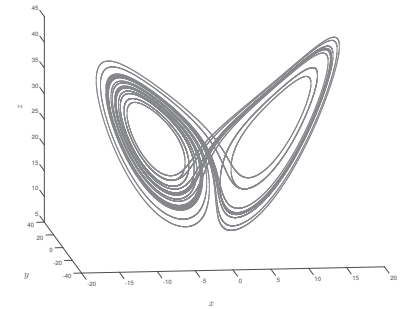
In dit voorbeeld zoeken we periodieke banen in het klassieke Lorenz systeem:

$$\begin{cases} \dot{x} \\ \dot{y} \\ \dot{z} \end{cases} = \begin{cases} 10(y-x) \\ x(28-z) - y \\ xy - \frac{8}{3}z \end{cases}, \quad t \in [0, L], \quad (12)$$

$$x(0) = x(L), y(0) = y(L), z(0) = z(L).$$

Zowel de periodieke baan $t \rightarrow (x(t), y(t), z(t))$ als de bijbehorende periode $L > 0$ zijn onbekend (dus het bepalen van de periode is onderdeel van het probleem).

Als $t \rightarrow (x(t), y(t), z(t))$ een oplossing is van (12), dan beschrijft de 'vershoven' afbeelding $t \rightarrow (x(t+\tau), y(t+\tau), z(t+\tau))$ dezelfde periodieke baan (voor elke $\tau \in \mathbb{R}$), en is dus ook een oplossing. Door deze translatie-symmetrie is de oplossing *niet* lokaal uniek, dus een contractiestelling is in deze formulering niet toepasbaar. We



Figuur 3 Een gevalideerde periodieke baan van periode $L \approx 25,03$ in het Lorenz systeem.

voegen daarom een extra conditie aan (12) toe, een zogenaamde fase-conditie, om de tijd-parametrisatie van de periodieke baan uniek vast te leggen. We eisen daartoe dat de baan op $t = 0$ in een vooraf gespecificeerd vlak in de fase-ruimte \mathbb{R}^3 ligt.

Aangezien het gaat om een periodieke oplossing ligt het voor de hand om een startformulering $F(x) = 0$, vergelijk (1), te kiezen in termen van Fourier-coëfficiënten:

$$\begin{aligned} x(t) &= \sum_{k \in \mathbb{Z}} a_k e^{ik\omega t}, \\ y(t) &= \sum_{k \in \mathbb{Z}} b_k e^{ik\omega t}, \\ z(t) &= \sum_{k \in \mathbb{Z}} c_k e^{ik\omega t}, \end{aligned}$$

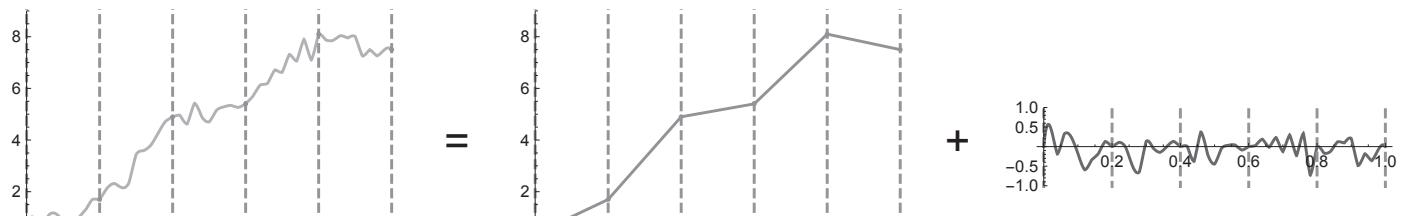
waarbij $\omega = 2\pi/L$. Het invullen van de bovenstaande expansies in (12) en de fase-conditie levert een oneindig stelsel van algebraïsche vergelijkingen op voor de onbekende rijtjes coëfficiënten a, b, c en de frequentie ω . We schrijven dat stelsel hier niet expliciet op, maar het is in hoge mate analoog aan (10). We passen nu de in de vorige paragraaf ontwikkelde methode toe op dit probleem en kunnen daarmee bijvoorbeeld bewijzen dat de kromme (het resultaat van een simulatie) in Figuur 3 gegarandeerd een periodieke baan van (12) representeert.

Een radiaal symmetrische PDV

We kunnen de methode ook toepassen om oplossingen van partiële differentiaalvergelijkingen (PDVs) te vinden. Als voorbeeld kijken we naar gelokaliseerde oplossingen van de stationaire niet-lineaire Schrödinger vergelijking

$$\Delta \psi(x) - \psi(x) + \psi(x)^3 = 0, \quad (13)$$

waarbij $\psi: \mathbb{R}^3 \rightarrow \mathbb{R}$. Wanneer we voor deze PDV op zoek gaan naar een radiaal symmetrische oplossing schrijven we $\psi(x) = f(|x|)$ voor een $f: [0, \infty) \rightarrow \mathbb{R}$ met



Figuur 4 Een afbeelding op $[0, 1]$ kan worden opgedeeld in een spline deel (met 6 roosterpunten in dit voorbeeld) en een restant.

de eigenschap dat $\lim_{r \rightarrow \infty} f(r) = 0$. Dit reduceert de PDV (13) tot

$$f''(r) + \frac{2}{r}f'(r) - f(r) + f(r)^3 = 0, \quad (14)$$

een singuliere niet-autonome gewone differentiaalvergelijking (GDV) op $(0, \infty)$. Numeriek rekenen op een oneindig domein is lastig, maar door deze GDV te transformeren naar het interval $(0, 1)$ en de methode van Green toe te passen, kan aangetoond worden dat (14) een begrensde oplossing heeft wanneer er een functie $w \in C[0, 1]$ bestaat die voor alle $t \in [0, 1]$ aan de volgende integraalvergelijking voldoet:

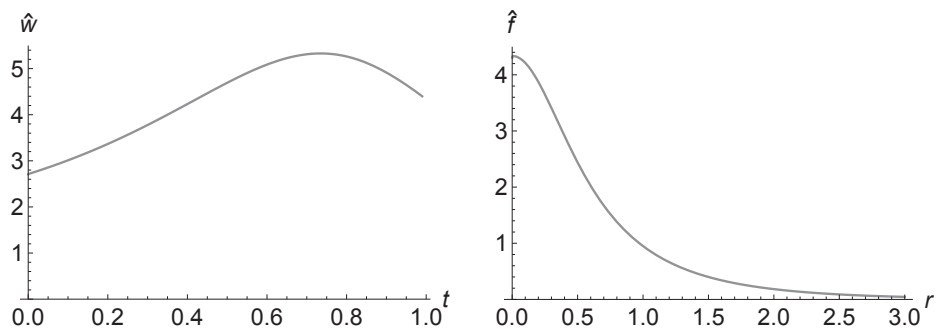
$$F(w) \stackrel{\text{def}}{=} w(t) - \frac{1}{2} \frac{1+t}{t^2} \int_0^t \frac{s^3(1-s)^3}{\log(s)^2} w(s)^3 ds - \frac{1}{2} \frac{1}{1-t} \int_t^1 \frac{s(1+s)(1-s)^4}{\log(s)^2} w(s)^3 ds = 0. \quad (15)$$

De relatie tussen de oplossing w van de integraalvergelijking (15) en de oplos-

sing f van de GDV wordt gegeven door $f(r) = \frac{1}{r} e^{-r} (1 - e^{-r}) w(e^{-r})$, zie Figuur 5. We merken op dat (15) op hoofdlijnen dezelfde vorm heeft als de formulering (9).

Voor dit probleem kiezen we als eindig-dimensionale benadering \hat{w} van de oplossing een stuksgewijs lineaire *spline*. Elke functie op $C[0, 1]$ kan worden opgedeeld in zo'n *spline*-deel met m roosterpunten, en een restant, waarbij het restant kleiner wordt naarmate het aantal roosterpunten m toeneemt, zie Figuur 4. De numeriek

gevonden simulatie \hat{w} , waarbij voor een uniform rooster is gekozen met $m = 1000$ punten, is afgebeeld in Figuur 5, evenals de bijbehorende functie \hat{f} . Het bewijs voor het bestaan van een (echte) oplossing van $F(w) = 0$ vlakbij \hat{w} , en daarmee van een radiaal-symmetrische gelocaliseerde oplossing van de PDV (13), volgt nu, na enige zorgvuldige analyse en met behulp van een computationeel gedeelte [3], uit de methode zoals beschreven in de vorige paragraaf. \diamond



Figuur 5 Links de functie $\hat{w}(t)$ op $[0, 1]$. Rechts de corresponderende functie $\hat{f}(r)$ op $(0, \infty)$.

Referenties

- 1 G. Arioli en H. Koch, Integration of dissipative partial differential equations: a case study, *SIAM J. Appl. Dyn. Syst.* 9(3) (2010), 1119–1133.
- 2 J.B. van den Berg, A. Deschenes, J.P. Lessard en J.D. Mireles-James, Stationary Coexistence of Hexagons and Rolls via Rigorous Computations, *SIAM J. Appl. Dyn. Syst.* 14(2) (2015), 942–979.
- 3 J.B. van den Berg, C.M. Groothedde en J.F. Williams, Rigorous computation of a radially symmetric localized solution in a Ginzburg–Landau problem, *SIAM J. Appl. Dyn. Syst.* 14 (2015), no. 1, 423–447.
- 4 J.B. van den Berg en R.S.S. Sheombarsing, Rigorous numerics for ODEs using Chebyshev series and domain decomposition, 2015. Preprint.
- 5 CAPD: Computer assisted proofs in dynamics, a package for rigorous numerics, <http://capd.ii.uj.edu.pl>.
- 6 R. Castelli, M. Gameiro en J.-P. Lessard, Rigorous numerics for ill-posed PDEs: periodic orbits in the Boussinesq equation, 2015. Preprint.
- 7 S. Day, J.-P. Lessard en K. Mischaikow, Validated continuation for equilibria of PDEs, *SIAM J. Numer. Anal.* 45(4) (2007), 1398–1424.
- 8 O.E. Lanford, A computer-assisted proof of the Feigenbaum conjectures, *Bull. Amer. Math. Soc.* 6 (1982), 427–434.
- 9 J.P. Lessard, Recent advances about the uniqueness of the slowly oscillating periodic solutions of Wright's equation, *J. Differential Equations* 248(5) (2010), 992–1016.
- 10 J.P. Lessard, J.D. Mireles-James en C. Reinhardt, Computer assisted proof of transverse saddle-to-saddle connecting orbits for first order vector fields, *J. Dynamics and Differential Equations* 26(2) (2014), 267–313.
- 11 T. Li en J.A. Yorke, Period three implies chaos, *The American Mathematical Monthly* 82 (1975), no. 10, 985–992.
- 12 K. Mischaikow en M. Mrozek, Chaos in the Lorenz equations: a computer assisted proof, *Bull. Amer. Math. Soc.* 32(1) (1995), 66–72.
- 13 W. Tucker, The Lorenz attractor exists, *C. R. Acad. Sci. Paris* 328(12) (1999), 1197–1202.