Universiteit Utrecht

Department
of Mathematics

Numerieke Wiskunde

door

G. Sleijpen en P. Zegeling

Juli, 2000

Voorwoord

Wat is Numerieke Wiskunde?
De mathematische problemen die men in de fysische en technische praktijk tegenkomt [1] zijn gewoonlijk niet met analytische middelen oplosbaar. Van bijvoorbeeld de functie e^{-t^2} kan de primitieve niet gegeven worden in termen van bekende functies als e-machten, logaritmen, sinussen, ... Toch is het voor statistische toepassingen van belang numerieke waarden te hebben voor $\int_{-\infty}^{x} e^{-t^2}\,dt$. In dit soort gevallen is men aangewezen op numerieke berekeningen. Numeriek wiskundigen ontwerpen en analyseren algoritmen, waarmee met behulp van de computer, op een *efficiënte* manier de voornoemde problemen *nauwkeurig* en *betrouwbaar* opgelost kunnen worden. Numerieke uitkomsten zijn altijd met *fouten* belast (rekenfouten en benaderingsfouten). Met een nauwkeurige uitkomst bedoelen we een uitkomst waarvan de fout kleiner is dan een opgegeven waarde. Uiteraard zijn de fouten zelden exact te bepalen (dan zouden de exacte resultaten bekend zijn), maar kunnen op zijn best geschat worden. Die schattingen moeten dan wel gegeven kunnen worden en betrouwbaar zijn. De problemen zijn vaak zeer rekenintensief en om een algoritme te ontwerpen waarmee een nauwkeurig betrouwbare resultaten efficiënt, d.w.z. binnen redelijke (korte) tijd verkregen kunnen worden moeten de nodige wiskundige inspanningen geleverd worden.

Waarom numerieke wiskunde?
Voor diverse mathematische problemen zijn er standaard routines [2] en programmapakketten te krijgen [3]. De belangrijkste zijn ook nog ingebouwd in een gebruikersvriendelijke omgeving [4]. Helaas is er niet één algoritme, of één routine die bijvoorbeeld alle differentiaalvergelijkingen betrouwbaar kan oplossen. De ene routine is geschikt voor het ene type differentiaalvergelijking, terwijl de andere beter werkt voor een ander type. In iedere routine moet uiteraard het mathematisch probleem gespecificeerd worden maar ook nog extra parameters. Kortom je moet verstand hebben van de theorie achter het mathematisch probleem, maar ook van de numerieke theorie. De situatie doet denken aan autorijden: een auto is een handig vervoermiddel. Een automaat (denk aan MATLAB) maakt autorijden gemakkelijker, maar ook voor zo'n auto moet het rijden geleerd worden. Als universitair geschoolde wil je ook van de goedgebaande wegen af. Dan is het prettig als je ook verstand hebt van het mechaniek.

Inhoud van deze cursus.
In deze cursus maken we kennis met de fundamentele begrippen en een aantal fundamentele gedachten uit de Numerieke Wiskunde. De begrippen en gedachten worden telkens geformuleerd in een praktische context. Vaak betreft dat die van gewone differentiaalvergelijkingen (DV's): we gebruiken de problemen die optreden bij het numeriek oplossen van DV's als motivatie en als inspiratiebron. Vaak leidt dit tot resultaten en inzichten die ook nuttig zijn buiten de context van DV's. Resultaten en inzichten worden telkens getoetst aan de praktijk. Vaak worden ze ook geïnterpreteerd naar hun betekenis voor het numeriek oplossen van DV's.

Een elementaire calculus cursus geeft voldoende voorkennis: Taylor reeksen van functies van één veranderlijke zijn de meest geavanceerde objecten die we zullen gebruiken.

Structuur van de cursus.
De cursus is verdeeld in drie delen. Ieder deel vormt een verdieping van de voorgaande delen. Onderwerpen kunnen dus vaker aan de orde komen. Delen bestaan uit meerdere

hoofdstukken. Ieder hoofstuk sluit af met praktische toepassingen waarin alle inzichten uit het hoofdstuk gecombineerd worden.

Doel van deze cursus:
(1) het aanleren van de numerieke basisbegrippen (fouten, foutvoortplanting, gestructureerde fout, convergentie, nauwkeurigheid, stabiliteit, efficiëntie);
(2) het aanleren van basistechnieken (Euler methoden, Lagrange interpolatie, trapeziumregel, Newtons methode, extrapolatie, ...); vertrouwd maken met de wiskundige aanpak die tot die technieken leidt; bijbrengen van voldoende vaardigheid in het uitvoeren van kleine aanpassingen aan standaard technieken om eigen problemen die niet geheel standaard zijn toch ook te kunnen oplossen;
(3) het geven van zelfvertrouwen in het omgaan met standaardpakketten: weten dat je ze kan gebruiken, maar ook dat ze beperkingen hebben; het bijbrengen van een kritische houding t.o.v. numerieke resultaten; handigheid bijbrengen in het opsporen van de oorzaak van eventuele numerieke problemen; leren resultaten naar waarde kunnen schatten; laten inzien dat het maken van een 'fool-proof' code erg moeilijk is.

Om dit doel te bereiken vragen we van de cursisten een aktievere inzet dan gebruikelijk is bij andere cursussen. De docent speelt in feite alleen een begeleidende rol. Al het werk wordt door de cursist zelf gedaan.

Werkwijze:
Het gebruik van de computer speelt een dominante rol in deze cursus. De cursist brengt tijdens de "contact uren" de hele tijd door achter de computer (geen hoorcollege, geen werkcollege). Numerieke resultaten die door de cursist zelf met behulp van de computer bepaald worden roepen vragen op en nodigen uit tot theoretische beschouwingen. Inzichten worden onmiddellijk aan de praktijk getoetst. De resultaten roepen weer nieuwe vragen op, etc.. Het formaliseren van de inzichten (omzetten in correcte wiskunde) gebeurt thuis.

In het diktaat gebruiken we het volgende sjabloon:

Vraag. Hierin formuleren we een vraag die we willen beantwoorden. Idealiter is de "vraag" slechts een expliciete verwoording van wat de cursist zichzelf al afvroeg naar aanleiding van resultaten in eerdere experimenten: de vragen vormen de rode draad door de cursus. Vragen zijn genummerd.

Experiment. Hierin geven we suggesties om al experimenterend inzicht te krijgen in het probleem dat in **Vraag** geopperd is. De experimenten zullen ook nieuwe vragen oproepen. De experimenten worden uitgevoerd met MATLAB. Een groot deel van het benodigd programmatuurwerk is al uitgevoerd (dit is geen cursus programmeren). De routine die nodig is voor vraag X is te vinden in MATLAB m-file `X.m` (waarbij X het nummer van de vraag). In de heading van de m-file staan ("uitgecommentarieerd", d.w.z. achter %) de aanwijzigingen die betrekking hebben op details van het experiment (geschikte parameter keuze, te volgen MATLAB instructies, etc.).

Verklaring. In de onderdelen "**Verklaring**" staan wat suggesties/vragen die, naast de experimenteel verkregen resultaten, zouden kunnen helpen om te begrijpen wat de oplossing is van het in **Vraag** geopperde probleem. Bij deze onderdelen is het niet de bedoeling om formele bewijzen te leveren: het gaat hier puur om inzicht.

Opgave. *Van een aantal inzichten is het nuttig om ze te formaliseren. Aanwijzingen om dat te doen staan in dit onderdeel. De bedoeling is om dit soort werk thuis te doen.*

Aan het eind van ieder hoofdstuk staat een samenvatting van de belangrijkste resulaten.

De cursus wordt afgesloten met een werkstuk.

Referenties

[1] F. Beukers, Modellen en Computers, College Diktaat, Mathematisch Instituut, Universiteit Utrecht, Utrecht, 1994.

[2] W.H. Press, S.A. Teukolsky, W.T. Vettering, and B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, Cambridge, 2nd. ed., 1992.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, LAPackUsers' Guide – Release 2.0, SIAM,Philadelpha, 1995. `http://www.netlib.org/lapack`

[4] The Student Edition of MATLAB, version 4, User's Guide, Prentice-Hall, 1995

Dankwoord

De ontwikkeling van het materiaal in deze cursus heeft plaats gevonden binnen het K&S-project “Introductie IT in het wiskunde onderwijs”.

Inhoudsopgave

Fouten

In alle numerieke berekeningen worden fouten gemaakt.
Als een grootheid berekend wordt uit een mathematische benadering spelen **benaderingsfouten**, ook wel *approximatie fouten* genoemd, een rol. Als we bijvoorbeeld $f'(a)$ berekenen volgens $\widetilde{f'(a)} := \frac{1}{h}(f(a+h) - f(a))$, met zeg $h = 0.0001$, dan is $f'(a) - \widetilde{f'(a)}$ de benaderingsfout.
Hoewel de computer met een behoorlijk aantal decimalen werkt is het aantal decimalen eindig en hebben we gewoonlijk ook nog te maken met **afrondfouten**, ook wel *evaluatie fouten* genoemd: $-0.1\ldots\ 10^{-12}$ is de afrondfout in de benadering $3.141\,592\,653\,589\,8$ van π.
Het numeriek resultaat is een benadering van de exacte waarde en is gewoonlijk belast met zowel benaderingsfouten als afrondfouten. Als we van een benaderende grootheid geen betrouwbare schatting kunnen geven voor (een bovengrens van) de fout dan heeft de benadering geen enkele betekenis (immers ook 7 benadert π). Foutschattingen en een analyse van de betrouwbaarheid van de foutschatting zijn daarom van cruciaal belang.

In de praktijk zal de fout niet exact bekend zijn. Immers als de fout bekend is, dan kan de exacte grootheid gemakkelijk bepaald worden. Het begrip "fout" speelt echter een grote rol in theoretische beschouwingen, bovendien kan de fout nogal eens redelijk betrouwbaar geschat worden.

Als we een benaderende waarde $\widetilde{a}$ voor een grootheid a hebben verstaan we onder
de fout: $a - \widetilde{a}$
de absolute fout: $|a - \widetilde{a}|$
de relatieve fout: $|(a - \widetilde{a})/a|$
de onbetrouwbaarheid: een bovengrens voor de absolute fout
de relatieve onbetrouwbaarheid: een bovengrens voor de relatieve fout.

Als de fout of de onbetrouwbaarheid in %'en wordt uitgedrukt wordt automatisch de relatieve fout of onbetrouwbaarheid bedoeld. In plaats van "fout" gebruiken we vaak "onnauwkeurigheid". Het woord "absoluut" wordt nogal eens weggelaten als er geen misverstand dreigt.

Met bijvoorbeeld $3.141\,6 \pm 0.002\,3$ voor π bedoelen we dat $3.141\,6$ een benaderende waarde is van π met een onbetrouwbaarheid van $0.002\,3$. Verder schrijven we bijvoorbeeld $2.008\,6\,10^1$ voor e^3 als we $2.008\,6\,10^1 \pm 0.000\,05\,10^1$ bedoelen: zondere verdere specificatie van de onbetrouwbaarheid is de onbetrouwbaarheid de helft van het laatst gegeven decimaal.

Experiment. Welke conventies hanteert MATLAB? (raadpleeg `help format` om de representatie van getallen in MATLAB te veranderen.)

Als bijvoorbeeld de benadering $2.712\,8$ van e een onbetrouwbaarheid $0.000\,05$ heeft dan zeggen we ook wel dat "de benadering correct is in 5 decimalen". Deze uitdrukking moet niet altijd naar de letter genomen worden, immers $1.000\,0$ is een benadering van $0.999\,999\,999$ met een onbetrouwbaarheid van $0.000\,05$, terwijl in feite geen enkel decimaal correct is. Met "correct in 5 decimalen" bedoelen we dus "correct afgerond op 5 decimalen".

Als je resultaten van een berekening presenteert aan andere mensen moet je natuurlijk de betrouwbaarheid van de resultaten vermelden, maar voorkom dat je de mensen vermoeit met teveel decimalen: rond resultaten af op een relevant aantal decimalen, geef de onbetrouwbaarheid in maar 1 of 2 (interessante) decimalen.

Experiment. Van de funktie $f(x) = \tan(x)$ willen we de waarde in een punt a weten. We beschikken over de benadering $\widetilde{a}$ die belast is met een onbetrouwbaarheid van 1%. Wat is de onbetrouwbaarheid in $f(\widetilde{a})$ voor het geval $\widetilde{a} = 1.4$ en voor het geval $\widetilde{a} = 3.0$? (Je kan dit uitzoeken met MATLAB, maar ook met een pocket calculator).

Verklaring. De relatieve onbetrouwbaarheden in $f(a)$ zijn (veel) groter dan in a voor deze waarden van a. Kan je dat verklaren?

Opgave. *Schrijf $\widetilde{a} = a+h$ en gebruik een 1-ste orde Taylor reeks om een uitdrukking te krijgen voor de relatieve fout in $f(\widetilde{a})$ in termen van de relatieve fout in $\widetilde{a}$. Hoe pakt dit uit in geval $f = \tan$ en $\widetilde{a} = 1.4$, $\widetilde{a} = 3.0$ beide met een onbetrouwbaarheid van 1%?*

Met $a \approx b$ geven we aan dat de benadering in relatieve zin is: $|(a-b)/b| \ll 1$. Als a en b funkties zijn en b heeft geen of maar een paar nulpunten dan schrijven we ook $a \approx b$ als $|[a(t)-b(t)]/b(t)| \ll 1$ voor alle t behalve voor de t in kleine intervalletjes rond de nulpunten van b.

Hoofdstuk I

Kennismaking met de Numerieke Wiskunde

door G. Sleijpen

SLEUTELBEGRIPPEN: *benaderingsfout, foutvoortplanting, gestruktureerde fout, foutschatting, betrouwbaarheid foutschatting, efficiëntie, stabiliteit.*

1 Numeriek oplossen van differentiaalvergelijkingen

Problemen die je in de praktijk tegenkomt kunnen nogal eens gemodelleerd worden als een differentiaalvergelijking (DV) of een stelsel DV's [1]. Gewoonlijk zijn die DV's niet analytisch oplosbaar. Middels analytische technieken [1, Hoofdstuk 2] kan je deelaspecten analyseren die enig inzicht kunnen geven in het "kwalitatieve" gedrag van de oplossing: via bijvoorbeeld linearisatie kan je iets te weten komen over de oplossing in de buurt van evenwichten, soms zijn er asymptotische uitspraken mogelijk, etc.. Helaas zijn die analytische technieken niet altijd toepasbaar of geven ze een onvolledig beeld. Bovendien is men in de praktijk geïnteresseerd in "kwantitatieve" resultaten: al in een ontwerpfase van bijvoorbeeld een gebouw zal men zeker willen weten of de constructie stormen en aardbevingen zal doorstaan; een raket wordt pas gelanceerd als de baan in grote nauwkeurigheid berekend is, etc.. In dit soort gevallen is men aangewezen op numerieke berekeningen. De belangrijke vraag is dan of de numerieke resultaten betrouwbaar zijn en hoe je kan achterhalen hoe betrouwbaar ze zijn. Verder vraag je je af of er één numerieke methode is waarmee iedere DV nauwkeurig opgelost kan worden en als dat niet zo is waar het van afhangt welke methode voor welk type DV het meest geschikt is. Zijn er standaard pakketten? Hoe roep je ze aan? Werken ze altijd? Als dat niet het geval is, hoe pas je ze aan voor je eigen probleem?

1.1 Differentiaalvergelijkingen

Gewone differentiaalvergelijkingen, het type DV's dat we in deze cursus bekijken, zijn van de vorm

$$u'(t) = f(t, u(t)), \quad t \in [t_0, t_0 + T]. \tag{1.1.a}$$

Hierin is f een gegeven functie en u een onbekende functie. u moet bepaald worden. De functies u en f zijn reëelwaardig of vectorwaardig: u is een C^1-afbeelding van het 'integratie' interval $[t_0, t_0+T]$ naar $\mathbb{R}^d$, f is gedefinieerd op een relevant deel van $\mathbb{R}\times\mathbb{R}^d$ en is ook $\mathbb{R}^d$-waardig. De oplossing u hangt ook nog af van een *beginvoorwaarde*

$$u(t_0) = u_0, \tag{1.1.b}$$

met u_0 een bekende vector.
In toepassingen staat t vaak voor de tijd, maar er zijn ook toepassingen waarbij t een andere interpretatie heeft. Voor het gemak noemen we in deze cursus t de tijd: t_0 is dan het aanvangstijdstip, T is de tijdsspanne waarover de oplossing van de DV gevolgd wordt, $t_0 + T$ is de eindtijd. Voor het schrijfgemak nemen we verder $t_0 = 0$.

De DV's die we hieronder beschrijven zullen we gebruiken om onze numerieke inzichten te testen. Andere voorbeelden zijn te vinden in diverse tekstboeken over gewone DV's (zie bijvoorbeeld [1]). De tekst hieronder bevat wat achtergrondinformatie bij de DV's die je voor de cursus niet per se hoeft te lezen.[1]

Voorbeeld 1A. [`Linear`] We bekijken eerst een eenvoudige DV:

$$\begin{cases} u'(t) = \lambda(t)u(t) + g(t), & t \in [0, 20] \\ u(0) = u_0, \end{cases} \tag{1.2}$$

hierin zijn u, g en λ reëelwaardige functies ($d = 1$), g en λ zijn gegeven functies gedefinieerd op $[0, 20]$, u_0 is een bekende scalair.

De DV (1.2) is eendimensionaal en lineair. Als λ constant is en g niet al te ingewikkeld is kan u analytisch bepaald worden. DV (1.2) is niet direct een voorbeeld om de behoefte aan een numerieke methode te illustreren. Maar juist omdat de DV zo eenvoudig is, is ze geschikt voor numerieke experimenten. Als een numerieke methode faalt voor deze eenvoudige DV dan zal hij het zeker laten afweten voor gecompliceerdere. Verder kunnen we, als we de exacte oplossing hebben, controleren of onze uitspraken betreffende de fout in een numeriek resultaat correct zijn. In een experimenteel stadium is het wel prettig om de exacte oplossing ter beschikking te hebben. We kiezen g in onze experimenten daarom als volgt:

$$g(t) \equiv w'(t) - \lambda\, w(t), \qquad u_0 \equiv w(0) \quad \text{met} \quad w(t) \equiv \cos(t). \tag{1.2.c}$$

Dan is de oplossing van (1.2) precies gelijk aan w: $u = w$.
We kunnen overigens iedere DV modificeren om iedere gewenste oplossing te forceren: met $g(t) \equiv w'(t) - f(t, w(t))$ is w de oplossing van $u'(t) = f(t, u(t)) + g(t)$, $u(0) = w(0)$.

In onze testen kiezen we λ constant (o.a. $\lambda = 0.3$, $\lambda = -1$ en $\lambda = -1000$) of voor een $\varepsilon > 0$, $\varepsilon \ll 1$,

$$\lambda(t) = \frac{-2}{\varepsilon + (\sin\frac{1}{4}t)^2}. \tag{1.2.d}$$

[1] Ieder voorbeeld heeft een naam die tussen rechte haakjes vermeld staat achter het nummer. Deze naam correspondeert met de naam van de MATLAB file waarin de DV gedefinieerd is. Zo staat het voorbeeld "Linear" in de file `Linear.m`. Als je een indruk wilt hebben over de oplossing kan je met de MATLAB functie `odeExact` een plaatje proberen te maken van de exacte oplossing. Zo levert `odeExact('Linear')` een plaatje van de exacte oplossing van de DV in voorbeeld 1A. Voor $d = 2$ en voor $d = 3$ krijg je een faseplaatje, voor andere dimensies worden alle coördinaten uitgezet tegen de tijd.

Voorbeeld 1B. De lineaire eendimensionale DV (1.2) is ook van het type (1.1) als λ en g complexwaardige functies zijn. De oplossing u is dan ook complexwaardig. Door alle grootheden te splitsen in reële en imaginaire delen kan de DV geschreven worden als een lineaire tweedimensionale reële DV. Voor de analyse is het overigens handiger om met de complexe grootheden te werken.

In het algemeen is het zo dat iedere 'complexe' d-dimensionale DV geschreven kan worden als een reële $2d$-dimensionale DV.

Voorbeeld 1C. `[Flame]` De volgende eendimensionale DV is een model voor het verloop van de temperatuur in een chemische reactie. κ en μ zijn constanten, $u(t)$ is de temperatuur op tijdstip t.

$$\begin{cases} u' = \mu\,(2-u)\,e^{\kappa(1-\frac{1}{u})} \quad \text{op} \quad [0,1] \\ u(0) = 1. \end{cases} \tag{1.3}$$

Voorbeeld 1D. `[Immuun]` Antigenen ("ziekmakers") worden in ons lichaam gebonden door antilichamen en vervolgens onschadelijk gemaakt. De DV in dit voorbeeld modelleert de verhouding in concentraties tussen deze twee. $x(t)$ is de concentratie antigenen in het lichaam op tijdstip t, $y(t)$ is de concentratie antilichamen. Het verloop van de concentraties kan, onder zekere aanname, beschreven worden door

$$\begin{cases} x' = 2(1+\kappa x-\kappa y)x \\ y' = (-1+3\kappa x-\kappa y-\vartheta\kappa xy)y \end{cases}. \tag{1.4}$$

Hierin zijn κ en ϑ nog twee nader te specificeren parameters.

De DV (1.4) is tweedimensionaal en met

$$u = \begin{bmatrix} x \\ y \end{bmatrix}, \quad f(t,u) = \begin{bmatrix} 2(1+\kappa x-\kappa y)x \\ (-1+3\kappa x-\kappa y-\vartheta\kappa xy)y \end{bmatrix}$$

zien we dat deze DV van het type is als in (1.1.a).

Voorbeeld 1E. `[vdPol]` De *van der Pol vergelijking* speelt een rol in de medische biologie bij het modelleren van de hartslag. De vergelijking bevat niet alleen eerste orde afgeleiden (u'), maar ook een hogere orde (u''). Met een eenvoudige truc kan de hogere orde afgeleide omgeschreven worden naar een afgeleide van de eerste orde. De DV wordt hierdoor wel hoger dimensionaal.

Voor zekere reële constanten μ, β en ε en u_0, w_0 wordt de van der Pol vergelijking gegeven door

$$\begin{cases} \varepsilon u'' = \mu(u'+\beta) - u - (u'+\beta)^3 \quad \text{op} \quad [0,T] \\ u(0) = u_0, \quad u'(0) = w_0. \end{cases} \tag{1.5}$$

Met $v \equiv u' + \beta$ is (1.5) equivalent met

$$\begin{cases} u' = v - \beta \\ v' = \frac{1}{\varepsilon}(\mu v - u + v^3) \end{cases} \tag{1.6.a}$$

met beginvoorwaarde

$$\begin{cases} u(0) = u_0 \\ v(0) = w_0 + \beta. \end{cases} \tag{1.6.b}$$

Deze DV is tweedimensionaal en weer van het type als in (1.1).

In de tests nemen we $\mu = 1$, $\beta = 0.57$ en $\varepsilon = 1/19$, $u_0 = 0$, $w_0 = -\beta$.

Voorbeeld 1F. [`Emigration`] De DV $\gamma\, u' = (\alpha - u)\, u$ is een eenvoudig model om de bevolkingsgroei te modelleren (*logistische groei*). Hierin zij α en γ positieve constanten en stelt $u(t)$ de gemiddelde bevolkingsdichtheid van een bepaalde populatie dieren voor op tijdstip t. Hierbij is gemiddeld over een zeker (leef)gebied. Als een klein deel van de populatie wegtrekt (emigratie) dan leidt een verfijning van het model (onder zekere aanname), waarin met dit kleine effect rekening wordt gehouden, tot $\gamma\, u' = (\alpha - u)\, u + \varepsilon u''$, met $\varepsilon < 0$, $|\varepsilon| \ll 1$. Deze DV kan, met $v = u'$ geschreven worden als

$$\begin{cases} u' = v \\ v' = \frac{1}{\varepsilon}[\gamma\, v - (\alpha - u)\, u]. \end{cases} \tag{1.7}$$

Voorbeeld 1G. [`Body3`] De volgende DV modelleert de baan die een kunstmaan K beschrijft die zich vrij (geen raketaandrijving) beweegt in een stelsel van een planeet A (aarde) met een satelliet M (maan). De massa van M is $\mu = 0.012\,277\,471$ maal de massa van A. Zwaartekrachtinvloeden van andere hemellichamen (en van de kunstmaan) verwaarlozen we. Verder nemen we aan dat de kunstmaan zich alleen maar in het vlak van planeet en satelliet beweegt. We kiezen de oorsprong van dit '(x, y)-vlak' in het zwaartepunt van het stelsel. We leggen de x-as door het centrum van A (en het centrum van M): het (x, y)-assenstelsel roteert dus met A en M mee. A ligt op positie $(\mu, 0)$ en M op $(\nu, 0) \equiv (1 - \mu, 0)$. Als op tijdstip t de kunstmaan positie $(x(t), y(t))$ heeft dan geldt

$$\begin{cases} x'' = x + 2y' - \nu \dfrac{x + \mu}{d_A} - \mu \dfrac{x - \nu}{d_M}, \\[2ex] y'' = y - 2x' - \nu \dfrac{y}{d_A} - \mu \dfrac{y}{d_M}, \end{cases} \tag{1.8.a}$$

$$\text{waarbij} \quad d_A \equiv ((x + \mu)^2 + y^2)^{\frac{3}{2}} \quad \text{en} \quad d_M \equiv ((x - \nu)^2 + y^2)^{\frac{3}{2}}.$$

Als u de snelheid is van K in de x-richting, $u = x'$, v de snelheid in de y-richting, $v = y'$, dan leidt dit tot een vierdimensionale DV.

Het is niet de bedoeling dat de kunstmaan in de kosmos verdwijnt of te pletter slaat op A of op M: we zijn dus geïnteresseerd in een gesloten baan, d.w.z. in een gesloten oplossingskromme $t \to (x(t), y(t))$. Zo'n baan wordt een *Arenstorf baan* genoemd. De beginwaarden

$$\begin{cases} x(0) = 0.994, & u'(0) = 0, \\ y(0) = 0, & v'(0) = -2.001\,585\,106\,379\,082\,522\,405\,378\,622\,24\ldots, \end{cases} \tag{1.8.b}$$

passen bij een Arenstorf baan met periode $T = 17.065\,216\,560\,157\,962\,558\,891\,720\,624\,9$. Het berekenen van een Arenstorf baan vergt hoge precisie en de startwaarden moeten in een groot aantal cijfers bekend zijn.

1.2 Kennismaking met numerieke oplosmethoden

SLEUTELBEGRIPPEN: *Betrouwbaarheid numeriek antwoord, efficiëntie*

In deze paragraaf proberen we te achterhalen of we blindelings mogen vertrouwen op de oplossingen die de computer ons levert.

Vraag 2A. Kunnen computers DV's eigenlijk wel oplossen? Differentiëren is een limietoperatie. Computers kunnen alleen met getallen uit een eindige collectie werken. Krijg je wel de oplossing te zien?

Experiment. Los (1.2) (`Linear`) met λ als in (1.2.d) en $\varepsilon = 0.02$ op in MATLAB met de routine '`odeEF`'.[2] Plot ook de exacte oplossing.[3] Bekijk de figuren die MATLAB produceert. Is het numeriek antwoord in alle tijdstippen berekend? (Zoom in als je meer details wilt zien). Valt het numerieke antwoord samen met het exacte antwoord in relevante tijdstippen? Welke zijn de relevante tijdstippen?

Verklaring. De afgeleide $u'(t)$ is benaderd door $\frac{1}{h}[u(t+h) - u(t)]$ of een ander differentie-quotiënt. Verder is er alleen gerekend in een beperkte set van tijdstippen t_n (is $t_n = nh$ voor $n = 0, 1, 2, \ldots$?). Door te benaderen introduceer je fouten.

Vraag 2B. Wat gebeurt er als je in meer tijdstippen rekent, d.w.z. als je de afstand h tussen de rekenpunten verandert?

Experiment. Verklein h eens met, zeg, een factor 3.[4] Wat gebeurt er met de fout? Verklein h eens met een factor 10 (bereken nu niet de exacte oplossing).[5] Wat gebeurt er met de rekentijd? Wat gebeurt er als je h vergroot?

Verklaring. Met een grotere h zijn er minder rekenpunten en dus zal het rekenwerk ook wel minder zijn. Echter voor grotere h is de benadering 'grover', d.w.z. de benaderingsfouten zijn groter. h wordt de **stapgrootte** genoemd.

Vraag 2C. Zijn de antwoorden altijd betrouwbaar? Of hangt de betrouwbaarheid samen met de stapgrootte? Wordt de fout kleiner als je h verkleint? Kan je aan de resultaten zien of ze betrouwbaar zijn?

Experiment. We experimenteren met twee routines. De eerste `odeEF` is door onszelf gecodeerd en berust op een eenvoudige methode die we in de volgende paragrafen uitgebreid zullen bestuderen. De tweede routine `ode23` is zeer geavanceerd en behoort tot de ODE Toolbox van MATLAB.

Los (1.5) (`vdPol`) eens op met $h = 0.03$ en `odeEF` en plot een faseplaatje van de numerieke oplossing.[6] (Laat ook eens de tussenresultaten plotten[7]). Vertrouw je het antwoord? Probeer ook eens $h = 0.01$. Zou je h nog kleiner moeten nemen? Los de DV ook eens op met `ode23`. Welke oplossing is de nauwkeurigste denk je? Hoe kan je daar enige zekerheid over krijgen?

Met `odeEF` kan je een hogere nauwkeurigheid krijgen door h te verkleinen. De professionele code `ode23` kiest zelf een "optimale" stapgrootte (en past die aan tijdens het

[2] 'Run' hiertoe de m-file `Vraag2.m`. Kijk wel eerst of de goede DV gekozen is: `ODE='Linear2'` en de goede oplosmethode: `method='odeEF'`. Check ook eerst of de function file `Linear2.m` de DV goed gedefinieerd is. Een uitgebreidere handleiding bij de MATLAB files staat in de file `READ.ME`.

[3] D.w.z, neem `PlotExactSolution=1`.

[4] h heet `dt` in `Vraag2.m` en in de andere '`Vraag`-files'.

[5] Neem `PlotExactSolution=0`.

[6] Neem `J=[1,2]`.

[7] `ShowSolutionProcess=1`

rekenproces). Je kan dus zelf h niet aan passen. Je kan wel opgeven hoe nauwkeurig je je antwoord wil middels de parameter `RelTol`. De suggestie is dat `ode23` het probleem oplost met een relatieve onbetrouwbaarheid `RelTol`. De default waarde is `RelTol=1.e-3`. Is de relatieve onbetrouwbaarheid inderdaad 0.001? Neem ook eens een hogere “relatieve tolerantie” (d.w.z. met kleinere `RelTol`).

Verklaring. Gaat de rest van het eerste hoofdstuk over.

> **Conclusie.** *Je kan er niet blindelings op vertrouwen dat de computer correcte antwoorden produceert.*

Bovenstaande experimenten roepen een aantal vragen op.
Hoe weet je of de h voldoende klein is, of daaraan gerelateerd, wanneer kan je het numeriek antwoord vertrouwen? Hoe betrouwbaar is dat antwoord precies? Hoe kan je een betrouwbaar antwoord te krijgen en veel rekenwerk vermijden? Hoe hangt dit samen met de numerieke methode en met de DV? Deze vragen staan centraal in dit hoofdstuk.

1.3 Convergentie

Sleutelbegrippen: *lokale fout, cumulatie lokale fouten, globale fout, convergentie, gestructureerde globale fout, foutschatten, betrouwbaarheid foutschatting*

In programmapakketten worden geavanceerde methoden gebruikt. Om te begrijpen wat de problemen zijn waarmee je bij het numeriek oplossen van DV's te maken krijgt, bekijken we eerst een eenvoudige methode, die overigens voor talloze DV's nog steeds een van de efficiëntste is.

Een computer kan niet differentiëren. Omdat voor kleine $h > 0$

$$u'(t) \approx \frac{u(t+h) - u(t)}{h}, \tag{3.1}$$

kunnen we, als we de waarde van u in $t+h$ en in t nauwkeurig kennen, $u'(t)$ toch nauwkeurig berekenen. Helaas kennen we die functiewaarden niet. Maar we hebben ook nog de DV en we kunnen kijken waar een combinatie van (1.1.a) met (3.1) toe leidt. Voor een $h > 0$ die voldoende klein is geldt

$$u(t+h) \approx u(t) + h\,u'(t) = u(t) + h\,f(t, u(t)).$$

Als we in een tijdstip t_n een nauwkeurige benadering, zeg u_n, van $u(t_n)$ tot onze beschikking hebben dan kunnen we u in het tijdstip $t_n + h$ nauwkeurig benaderen door $u_n + hf(t_n, u_n)$: de grootheden u_n en t_n zijn bekend, en $f(t_n, u_n)$ kan uitgerekend worden. Er kan dus gemakkelijk een benadering $u_{n+1} \equiv u_n + hf(t_n, u_n)$ van $u(t_n + h)$ berekend worden. We kunnen vervolgens het spelletje herhalen: we kunnen nu op dezelfde manier in het tijdstip $t_n + 2\,h$ de oplossing schatten, etc.. Beginnen we met deze aanpak in het tijdstip $t_0 = 0$ dan weten we dat $u(0) = u_0$ (zie (1.1.b)) en komen we tot het volgend rekenproces:

$$u_{n+1} = u_n + hf(t_n, u_n), \qquad t_{n+1} = t_n + h, \quad n = 0, 1, 2, \ldots, \tag{3.2}$$

waarin u_n een benadering is voor $u(t_n)$. Dit rekenschema heet de **Euler forward** (EF) methode.

Vraag 3A. De oplossing van de DV's hebben we grafisch gerepresenteerd. Hoe kan je de EF methode grafisch interpreteren?

Experiment. Plot voor (1.2) (`Linear`) met $\lambda = -1$ en g als in (1.2.c),[8] de grafiek van de exacte oplossing en plot, voor niet al te kleine h, de numerieke oplossing. Verbind de punten (t_n, u_n) met lijnstukjes.[9] Plot ook de exacte oplossing van de DV (1.2.a) door (t_1, u_1), d.w.z. plot de grafiek van die u waarvoor $u' = \lambda\, u + g$ en $u(t_1) = u_1$. Doe dat ook voor de exacte oplossingen van de DV door achtereenvolgens (t_2, u_2),..., (t_6, u_6).[10] Bekijk de grafische resultaten in de buurt van de eerste paar t_n (zoom in).

Verklaring. EF volgt telkens over korte stukjes (stukjes met, langs de t-as, lengte h) de raaklijn en stapt daarbij telkens over op een andere 'naburige' oplossing.

Vraag 3B. Convergeert EF? Wat betekent convergentie hier?

Experiment. Onderzoek door h te verkleinen voor de testvoorbeelden (1.2) (`Linear`) met $\lambda = -1$ en (1.4) (`Immuun`) met $\kappa = 1.3$ en $\vartheta = 0.06$ of EF convergeert en als er convergentie is, in welke zin dat gebeurt.[11]

Verklaring. Er geldt

$$u(t_{n+1}) = u(t_n) + h\, f(t_n, u(t_n)) + \tau_n \quad \text{waarbij} \quad \tau_n \equiv \tfrac{1}{2}\, (h)^2\, u''(\xi_n) \tag{3.3}$$

voor geschikte ξ_n tussen t_n en t_{n+1}.
EF kan je zien als een methode waarin in ieder tijdstip t_n het **defect** τ_n weggemoffeld wordt (vergelijk (3.3) en (3.2)). De defecten τ_n gaan naar 0 als $h \to 0$ (waarom?). Het is duidelijk dat de defecten de nauwkeurigheid van het numeriek resultaat beïnvloeden, het is niet duidelijk hoe ze dat doen. Het is evenmin duidelijk of er niet nog andere factoren een rol spelen.

Bij convergentiebeschouwingen moet je je realiseren dat voor iedere t de n in de vergelijking $t_n = t$ afhangt van h.

Opgave. *Bewijs* (3.3). *Hint: gebruik een Taylor ontwikkeling.*

We onderzoeken eerst hoe de defecten de numerieke nauwkeurigheid beïnvloeden.
Als h klein is, dan is

$$\tau_n \approx \tfrac{1}{2}\, h^2\, u''(t_n). \tag{3.4}$$

Als we voor kleine h de stapgrootte h halveren dan worden de defecten met ongeveer een factor 4 kleiner, immers $\frac{1}{2}(\frac{1}{2}h)^2 u''(t) = \frac{1}{4}[\frac{1}{2}h^2 u''(t)]$. Geldt dat ook voor de fout in het tijdstip t? Deze **fout** is gegeven door

$$e_n \equiv u(t_n) - u_n \quad \text{met } n \text{ zodat} \quad t_n = t. \tag{3.5}$$

Vraag 3C. Hoeveel kleiner wordt de fout als je h halveert?

Experiment. Plot de exacte oplossing. Los (1.2) (`Linear`) met $\lambda = -1$ op met EF met stapgrootte h, $h/2$, $h/4$, ... Wat gebeurt er met de fout?

Verklaring. Defecten cumuleren.
De defecten worden weliswaar een factor vier kleiner, maar het aantal stappen om in

[8] Deze DV is gedefinieerd in `Linear.m`.
[9] Gebeurt vanzelf met `Vraag3A.m`.
[10] Neem `PlotExactNearbySol=1`.
[11] De tweede DV is gedefinieerd in `Immuun.m`.

eenzelfde tijdstip te komen verdubbelt.
De fout in het tijdstip t (waarvoor t/h geheel is) lijkt evenredig met h.

Opgave. *Bekijk een tijdstip $t \in [0, T]$ waarvoor t/h geheel is. Vergelijk de fout in twee stappen EF vanuit het tijdstip t met stapgrootte $h/2$ met een stap EF ook vanuit het tijdstip t maar nu met stapgrootte h.*

Vraag 3D. Is de fout e_n een som van defecten in voorgaande tijdstippen of hebben nog andere factoren een invloed op de fout?

Experiment a. Los weer (1.2) (`Linear`) op, maar neem nu eens $\lambda = 0.3$. Kijk met name naar het verschil tussen de numerieke oplossing voor $\lambda = -1$ en $\lambda = 0.3$ in $t = 20$. Hoe kan je dat verschil verklaren? Plot ook weer eens de exacte oplossingen door (t_n, u_n) voor $n = 1, \ldots, 6$ en kijk naar de resultaten in $t = 20$.
Tussen haakjes: halveert hier de fout nog steeds als je h halveert?

Verklaring. De DV zelf kan fouten vergroten.
Met EF maak je telkens foutjes (nl. de defecten) die de DV kan vergroten.
Als u de oplossing is van (1.1.a-b) en $\widetilde{u}$ voldoet aan (1.1.a), maar nu met $\widetilde{u}(0) = u_0 + \varepsilon$ voor een $\varepsilon \neq 0$, dan geeft $e(t) \equiv u(t) - \widetilde{u}(t)$ aan hoe de DV de verstoring ε van de beginwaarde voortplant.

$$e(t + h) \approx e^{h\,\lambda(t)}\, e(t). \tag{3.6}$$

Over een tijdstraject(je) ter lengte h vergroot of verkleint de DV de fout, d.w.z. het effect van de verstoring, met ongeveer een factor $e^{h\,\lambda(t)}$.

Opgave. *Bekijk* (1.2) *met onverstoorde en verstoorde beginwaarde.*

a. *Bewijs dat $e'(t) = \lambda(t)e(t)$ voor alle $t \in [0, T]$ en $e(0) = -\varepsilon$.*

Stel nu dat $\lambda(t) = \lambda_0$ voor alle t (λ is constant)

b. *Bewijs dat $|e(t)| = |\varepsilon|\, e^{t\lambda_0}$, in geval $\lambda \in \mathbb{R}$. Spoort dit met je observaties in je experiment? In het geval dat λ_0 complex is en niet reëel (zie Voorbeeld 1B) geldt $|e(t)| = |\varepsilon|\, e^{tRe(\lambda_0)}$.*

c. *Bewijs dat $e(t + h) = e^{h\lambda_0}\, e(t)$ (zie* (3.6)*).*

Experiment b. Neem nu eens $\lambda = -125$ en $h \geq 0.02$. Is het resultaat ook zo beroerd voor kleinere h? Halveert de fout dan weer bij halverende h?
Vergroot de DV fouten? Plot ook weer eens de exacte oplossingen door de eerste paar numerieke resultaten en zoom in $t = 0$?

Verklaring. EF kan fouten vergroten.
Voor EF toegepast op (1.2) geldt

$$e_{n+1} = (1 + h\,\lambda_n)\, e_n + \tau_n \quad \text{met} \quad \lambda_n \equiv \lambda(t_n). \tag{3.7}$$

Over een tijdstraject ter lengte h vergroot of verkleint EF de 'oude' fout met een factor $1 + h\,\lambda_n$. (Hoe groot was deze factor in de beroerde gevallen?) Verder komt er in dat traject nog een 'nieuwe' fout bij, nl. het defect.

Opgave. *Bekijk EF toegepast op* (1.2).

a. *Bewijs* (3.7).

b. *Als $|h\,\lambda_n| \ll 1$ (denk aan $|h\,\lambda_n| \leq 0.2$) dan geldt $1 + h\,\lambda_n \approx e^{h\,\lambda_n}$. Bewijs dit en vergelijk dit resultaat met het resultaat in onderdeel c. van de vorige opgave.*

Conclusie.
a. *Defecten cumuleren.*
b. *De DV zelf kan fouten vergroten.*
c. *EF kan fouten vergroten.*

Als $|h\,\lambda_n| \not\ll 1$ dan springt EF anders om met 'oude' fouten dan de DV. Dit is gewoonlijk fataal als ook nog $|1+h\,\lambda_n| > 1$. We zullen deze situatie uitgebreid bestuderen in §1.4. Voorlopig concentreren we ons op de situatie waarin h zo klein is dat EF de fouten op min of meer dezelfde manier voortplant als de DV zelf. In dat geval is de conclusie in 3C correct.

Vraag 3E. Kan je het inzicht uit 3C gebruiken om de fout te schatten? Kan je dit inzicht gebruiken om de stapgrootte zo aan te passen dat je een fout krijgt die kleiner is dan bv. 10^{-3}?

Experiment. Los weer (1.2) (`Linear`) met $\lambda = -1$ op met EF voor $h = 0.2$ en $h = 0.1$ en schat de fout. Bepaal ook de exacte oplossing. Hoe goed is je schatting? Met welke waarde voor h verwacht je een fout te krijgen die in alle (relevante) tijdstippen kleiner is dan 10^{-3}. Kijk of je verwachting uitkomt.

Vraag 3F. Geldt die evenredigheid van de fout met h ook voor grotere h (met h toch nog zo klein dat EF fouten niet beroerder voortplant dan de DV zelf)? Geldt die evenredigheid in alle (relevante) tijdstippen?

Experiment. Los weer (1.2) (`Linear`) met $\lambda = -1$ op met EF maar nu met $h = 1$ en $h = 0.5$.

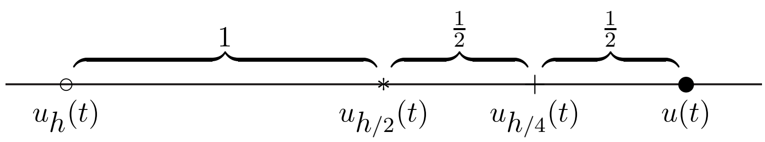
Verklaring. Als h niet klein is (of preciezer, als u'' veel varieert op $[t, t+h]$) dan is (3.4) niet correct. Soms is de fout "per ongeluk" ≈ 0.

Als in een numerieke oplosmethode van DV's de fouten min of meer evenredig zijn met h^p voor alle h die klein genoeg zijn dan is de methode van **orde** p: EF is een methode van orde 1.

Vraag 3G. Als je de fout wilt schatten (en het exacte antwoord niet hebt), hoe weet je of h voldoende klein is?

Experiment. De EF oplossing die verkregen is met stapgrootte h noemen we u_h. Dus in $t_n = n\,h$ is $u_h(t_n) = u_n$. Wat kan je voor het tijdstip t (met t/h geheel) zeggen over de verhouding tussen $u_h(t) - u_{h/2}(t)$ en $u_{h/2}(t) - u_{h/4}(t)$? Hoe hangt die verhouding samen met h? Gebruik de resultaten uit de experimenten in 3C en 3F.

Verklaring. Als voor een zekere c geldt $u_h(t) - u_{h/2}(t) \approx c\,h$ dan kan je de resultaten als volgt weergeven op de getallenrechte. De getallen geven verhoudingen weer.



Vraag 3H. Zijn die inzichten ook correct voor ingewikkeldere DV's?

Experiment a. Pas EF toe op de niet-lineaire DV (1.3) (`Flame`) met $\mu = \frac{1}{4}$ en $\kappa = 2$.[12] Hoe gedraagt de fout zich?

[12] in `Flame.m`.

Verklaring. Als EF fouten niet beroerder voortplant dan de DV zelf dan gedragen de fouten zich min of meer als de oplossing van een lineaire DV en wel als de oplossing van

$$\begin{cases} e'(t) = \lambda(t)e(t) + g(t) \quad \text{met} \quad \lambda(t) \approx \frac{\partial f}{\partial u}(t, u(t)) \quad \text{en} \quad g \approx \frac{1}{2}h\, u''(t) \\ e(0) = 0. \end{cases} \tag{3.8}$$

Opgave. *Bekijk de eendimensionale versie van* (1.1).

a. *Bewijs dat, voor het geval dat* $\varepsilon \equiv e(t)$ *klein is, geldt*

$$f(t, u(t) + \varepsilon) = f(t, u(t)) + \lambda(t)\varepsilon, \quad \textit{waarbij} \quad \lambda(t) \approx \frac{\partial f}{\partial u}(t, \widetilde{u}(t))$$

en $\widetilde{u}(t)$ *een geschikte waarde is tussen* $u(t)$ *en* $u(t) + \varepsilon$.

b. *Geef de versie van* (3.6) *en van* (3.7) *voor de 1-d. versie van* (1.1).

Experiment b. Hoe zit het met meerdimensionale DV's? Pas EF eens toe op (1.4) (`Immuun`) met $\kappa = 1.3$ en $\vartheta = 0.06$ en $T = 5$. $h = 0.1$ lijkt een geschikte 'start' voor het halveringsproces. Kijk naar de oplossingen als functie van t maar ook naar het faseplaatje. Doe hetzelfde eens voor $T = 40$.

Verklaring. Bovenstaande beschouwingen kunnen coördinaatsgewijs gehouden worden.

Experiment c. Pas EF weer toe op (1.2) (`Linear`) met $\lambda = -500$.

Verklaring. Ook hier is het antwoord ja, maar h moet wel erg klein gekozen worden. De verklaring zit in de grootte van de factor $|1 + h\,\lambda_n|$ in (3.7). Je kan je hier afvragen hoe groot h mag/moet zijn. We bewaren deze vraag tot §1.4.

Zijn de inzichten die we voor EF opgedaan hebben ook van toepassing voor andere oplosmethoden? We bekijken ondermeer de Trapeziumregel. Deze methode ontstaat door in iedere t_n het defect τ_n^{trap} in de relatie

$$\begin{cases} u(t_{n+1}) = u(t_n) + \frac{1}{2}h\,[f(t_n, u(t_n)) + f(t_{n+1}, u(t_{n+1}))] + \tau_n^{\text{trap}} \\ \text{met} \quad \tau_n^{\text{trap}} = -\frac{1}{12}h^3\, u'''(\xi_n) \end{cases} \tag{3.9}$$

weg te poetsen.

Vraag 3I. Hoe verandert de fout in de Trapeziumregel als je h verandert? Zijn er methoden die nog nauwkeuriger zijn?

Experiment a. Pas de Trapeziumregel toe op (1.4) (`Immuun`) met $\kappa = 1.3$ en $\vartheta = 0.06$ en kijk of je verwachting uitkomt.[13]

Conclusie. *De Trapeziumregel is van orde 2.*

Opgave. *Bewijs* (3.9).

Hint: bedenk dat $f(t, u(t)) = u'(t)$ *en ontwikkel de voorkomende functies rond* t_n.

Experiment b. Welke orde heeft RK4?[14] De orde is lastig af te lezen uit de plaatjes. Hoe komt dat?

Opgave 3J. *Bewijs dat* (3.9) *ontstaat door de Trapeziumregel voor numeriek integreren toe te passen op de integraal in de relatie* $u(t + h) = u(t) + \int_t^{t+h} u'(s)\, \mathrm{d}s$.

[13] De Trapeziumregel roep je aan met `odeIT`.

[14] `odeRK` maakt gebruik van RK4. Het heeft hier geen zin om de exacte oplossing te laten plotten omdat de "exacte" oplossing berekend wordt met `odeRK`: neem `PlotExactSolution=0`.

Opgave 3K. *Hieronder een schets van het bewijs dat de EF "convergeert". Details worden als opgave aan de lezer over gelaten.*

Neem $d = 1$. Bewijs de volgende uitspraken.

a. $f(t_n, u(t_n)) - f(t_n, u_n) = \frac{\partial f}{\partial u}(t_n, \widetilde{u}_n)\, e_n$ *voor zekere* $\widetilde{u}_n$ *tussen* $u(t_n)$ *en* u_n.

b. $e_0 = 0, \quad e_{n+1} = e_n + h\,\frac{\partial f}{\partial u}(t_n, \widetilde{u}_n)\, e_n + \tau_n = \Big(1 + h\,\frac{\partial f}{\partial u}(t_n, \widetilde{u}_n)\Big)\, e_n + \tau_n.$

c. *Als* $\left|\frac{\partial f}{\partial u}(t_n, \widetilde{u}_n)\right| \leq M$ *voor alle* n *dan geldt*

$$|e_{n+1}| \leq (1 + h\,M)|e_n| + |\tau_n| \leq \textstyle\sum_{j=0}^{n}(1 + h\,M)^{n-j}|\tau_j|$$

d. *Omdat* $1 + h\,M \leq e^{h\,M}$ *is* $(1 + h\,M)^i \leq e^{ih\,M}$ *en geldt*

$$|e_n| \leq e^{n\,h\,M}\textstyle\sum_{j<n}|\tau_j| \leq e^{TM}\sum_{j<n}|\tau_j| \quad \text{voor alle} \quad t_n \equiv 0 + n\,h \leq T.$$

e. $\sum_{j<n}|\tau_j| \leq n\,\max_{j<n}|\tau_j| \leq n\,h^2\,\frac{1}{2}\,\max_\xi|u''(\xi)| \leq \Big(T\,\frac{1}{2}\,\max|u''(\xi)|\Big)\,h.$

f. *EF convergeert als de functies* $|u''|$ *en* $|\frac{\partial f}{\partial u}|$ *begrensd zijn.*

1.4 Stabiliteit

SLEUTELBEGRIPPEN: *Stabiliteit, groeifactor fout, foutvoortplanting, stijve DV, impliciet versus expliciet, efficiëntie*

De **stabiliteitsanalyse** in de Numerieke Wiskunde houdt zich bezig met de vraag hoe kleine fouten (zowel benaderingsfouten als afrondfouten), die gemaakt zijn in de afzonderlijke rekenstappen van een rekenproces, doorwerken in het eindantwoord.

> *Een rekenmethode is* **instabiel** *als kleine fouten in de afzonderlijke rekenstappen een grote invloed hebben op het eindantwoord, anders wordt de methode* **stabiel** *genoemd.*

In deze formulering zijn de begrippen 'instabiel' en 'stabiel' nogal subjectief: wat groot is en wat klein is hangt erg af van de toepassing. Een relatieve fout van bijvoorbeeld 0.1 % in de weersvoorspelling voor het weer van overmorgen ($300 \pm 0.3°$ K) is klein terwijl een fout van 0.1% in de berekening van de baan van een raket naar Mars fataal is. Voor mathematische toepassingen wordt in de numerieke wiskunde het stabiliteitsbegrip dat in een bepaalde context gehanteerd wordt telkens gepreciseerd. Het precieze begrip wordt dan versierd met bijwoorden als 'sterk stabiel', 'asymptotisch stabiel', etc.. Het heuristisch begrip speelt echter telkens op de achtergrond een rol. Merk op dat de heuristiek die aan de numerieke stabiliteitsbegrippen ten grondslag ligt precies dezelfde is als voor de stabiliteitsbegrippen in de mathematische analyse. Zo is een stabiel evenwicht van een DV een evenwicht dat de oplossing van de DV opzoekt als diezelfde evenwichtsoplossing licht verstoord wordt (kleine verstoringen hebben uiteindelijk een klein effect).

De oorzaak van een grote amplificatie van kleine fouten in de afzonderlijke stappen van een rekenproces hoeft overigens niet altijd te liggen bij de numerieke methode: zie bv. experiment a in 3D. Als bijvoorbeeld het beginwaarde probleem van een autonome DV een instabiel evenwicht als exacte oplossing heeft dan zal dat probleem met geen enkele numerieke methode stabiel opgelost kunnen worden (waarom niet?). Door een mathematische herformulering (in dit voorbeeld, door de tijd 'om te draaien', d.w.z. door t door $-t$ te vervangen) kan het evenwicht wel toegankelijk worden voor een stabiele numerieke berekening (zie dat in).

Vraag 4A. In experiment c in 3H hebben we gezien dat we voor een acceptabel numeriek resultaat de h wel eens erg klein moeten nemen. Zeker voor meerdimensionale DV's geven kleine h aanleiding tot lange rekentijden. Om efficiëntie redenen is het gewenst h zo groot mogelijk te kiezen. Hoe groot kan h maximaal zijn in experiment c in 3H? Hoe kritisch is die maximale waarde?

Experiment a. Probeer voor (1.2) (`Linear`) met $\lambda = -500$ zo goed mogelijk de maximale waarde van de h te vinden.

Verklaring. Kijk eens naar (3.7) en naar de waarde van de factor $|1 + \lambda h|$ waarmee EF voor deze DV oude fouten voortplant.

Experiment b. Voor $\lambda = -400$ en $h = 0.005$ krijgen we een nauwkeurige numerieke oplossing. Voor $h = 0.005\,01$ is $|1{+}h\,\lambda| = 1.004$. Wat denk je: is de numerieke oplossing ook nauwkeurig?

Verklaring. Hoeveel stappen heb je nodig om in $T = 20$ te komen?

Experiment c. Met $\lambda = 0.1$ en $h = 0.04$ is $|1 + \lambda h| = 1.004$. Dit is precies de waarde in ons vorige experiment waarvoor we slechte resultaten kregen. Hoe pakt dat hier uit?

Verklaring. Hoeveel stappen heb je nu nodig?

> **Conclusie. a.** *Als* $|h\,\lambda| \ll 1$ *(denk aan* $|h\,\lambda| \leq 0.2$*) dan plant EF de fouten voort op dezelfde manier als de DV dat doet.*
> **b.** *Als* $|h\,\lambda_0| \not\ll 1$ *dan kan je alleen een nauwkeurige numerieke oplossing krijgen als (vrijwel) exact voldaan is aan de* stabiliteitseis $|1 + h\,\lambda_0| \leq 1$.

Vraag 4B. Bovenstaande conclusie geldt voor de eendimensionale lineaire DV (1.2) voor $\lambda(t) = \lambda_0$ is constant. Wat hebben we eraan voor gecompliceerdere DV's? Blijft dan nog iets van de conclusie overeind? Kunnen we gemakkelijk zien of we met instabiliteiten te maken hebben?

Experiment a. Pas EF weer toe op (1.2) maar neem nu een variabele λ: neem λ als in (1.2.d) met $\varepsilon = 0.02$.[15] Voor welke h heb je geen stabiliteitsproblemen? Bekijk voor $h = 0.04$ ook eens de grafiek van $h\lambda(t)$.[16] Kan je de instabiliteit verklaren? Kan je, aan de hand van de grafiek van $h\lambda$ schatten voor welke h je van de instabiliteit af bent?

Verklaring. Op trajecten waar $|1 + h\,\lambda(t)| > 1$ blaast de fout op.

Experiment b. We kijken eens wat er van onze inzichten overblijven voor niet-lineare DV's. Pas EF toe op (1.3) (`Flame`) met $\mu = \frac{1}{4}$ en $\kappa = 20$. Probeer de "kritische" h te vinden. Plot ook eens de grafiek van $h\,\lambda(t)$, nu met $\lambda(t) \equiv \frac{\partial f}{\partial u}(t, u(t))$.[17]

Verklaring. De fouten gedragen zich min of meer als de oplossing van de lineaire DV (zie (3.8)).

Experiment c. Los (1.6) (`vdPol`) op met EF en stapgrootte $h = 0.04$. Plot een faseplaatje. Voor welke waarde van h raak je de wiggels kwijt? Bekijk voor deze h ook $h\,\lambda_i(t_n)$ met $\lambda_1(t_n)$ en $\lambda_2(t_n)$ de eigenwaarden van de Jacobi matrix $\frac{\partial f}{\partial u}(t_n, u_n)$.[18]

Verklaring. In de meerdimensionale situatie hangt de stabiliteit samen met $h\,\lambda(t)$ met $\lambda(t)$ eigenwaarde van de Jacobi matrix $h\,\frac{\partial f}{\partial u}(t, u(t))$.

[15] DV staat in `Linear2`.

[16] Neem `ShowEigenvaluesJacobian = 1;`. Na het volgende experiment zul je begrijpen waarom we de naam 'ShowEigenvaluesJacobian' gekozen hebben voor het plotten van de grafiek van $h\lambda(t)$.

[17] Neem `ShowEigenvaluesJacobian = 1;`.

[18] Neem `ShowEigenvaluesJacobian = 1;`.

> **Conclusie.** *Ook voor niet-lineaire (meerdimensionale) DV's manifesteert instabiliteit zich in de vorm van "wiggels" in de numerieke oplossing. In zo'n geval moet de stapgrootte h kleiner genomen worden. In geval van instabiliteit krijgen we niet een onnauwkeurige oplossing die er toch "redelijk" uitziet: door de "wiggels" zien we dat we met instabiliteiten te maken hebben.*

Vraag 4C. Voor welk type DV's kunnen we stabiliteitsproblemen verwachten met EF?
Experiment. Los (1.2) (`Linear`) met $\lambda = -40$ op met EF en $h = 0.05$. Plot ook de exacte oplossingen door (t_n, u_n) voor de eerste paar n en kijk eens hoe die exacte oplossingen zich gedragen: zoom in bij $t = 0$. Neem nu eens $h = 0.07$ en $T = 0.5$. Zoom in, nu rond de hele exacte oplossing. Kan je grafisch zien waarom EF instabiel is?

> **Conclusie.** *Als de exacte oplossing 'langzaam' varieert terwijl naburige oplossingen zeer 'snel' variëren dan hebben we te maken met een zogenaamde* **stijve** *DV. In zo'n geval bepalen bij EF de naburige snel variërende oplossingen de stapgrootte.*

Langzaam variërend wil voor de numerieke methode zeggen dat de defecten τ_n in absolute waarde klein zijn voor niet al te kleine stapgrootte h (denk aan $|\tau_n| \leq 10^{-3}$ voor $h = 0.02$). Voor EF mag dan $|u''|$ niet al te groot zijn: immers voor EF is $\tau_n = \frac{1}{2}h^2 u''(\xi_n)$.

Stijve DV's komen in de praktijk nogal eens voor. Ze ontstaan typisch als aan een goed model, d.w.z. aan een DV die het praktijk probleem al behoorlijk goed modelleert, ter perfectionering hogere orde afgeleiden met een klein gewicht worden toegevoegd. Je kan hierbij denken aan de (0-de orde) "DV" $u(t) - g(t) = 0$ waaraan de term $\varepsilon u'(t)$ wordt toegevoegd: $\varepsilon u' + u - g = 0$. Dus $u' = -\frac{1}{\varepsilon}(u - g)$ met $u(0) = g(0)$. Voor $0 < \varepsilon \ll 1$ is $u \approx g$ terwijl naburige oplossingen van de vorm $w = u + c\,e^{-t/\varepsilon}$ zijn. Hier is c een zekere constante. Omdat $w''(0) = u''(0) + \frac{c}{\varepsilon^2}$ zal $|w''(0)|$ groot zijn (tenzij $|c| \ll \varepsilon$). In de praktijk zijn de modellen (DV's) natuurlijk wel wat ingewikkelder.

Vraag 4D. Kunnen we voor stijve DV's met een nauwkeurige numerieke methode met grotere stapgroottes toe?
Experiment. Vergelijk de grootte van de stapgrootte die EF vereist met die voor RK4 voor de DV (1.2) met $\lambda = -500$ en weer $T = 20$ (zie experiment c in 4A). Hoeveel groter kan de stapgrootte met RK4 zijn? Zie je deze factor ook terug bij ingewikkeldere DV's? Bekijk ook eens (1.3) (zie experiment b in 4B).

Vraag 4E. We hebben hierboven gezien dat een nauwkeurigere methode voor stijve DV's ook wel eens een heel kleine stapgrootte kan vergen. Je kan je afvragen of er überhaupt wel methoden bestaan waarmee je ook met een grotere stapgrootte nauwkeurige resultaten kan behalen?

We proberen de Euler backward methode (EB). EB is gebaseerd op de 'terugwaartse' differentie (vergelijk dit met (3.1))

$$u'(t) \approx \frac{u(t) - u(t-h)}{h}, \tag{4.1}$$

Dit leidt tot de relatie

$$u(t_n) = u(t_{n-1}) + h\,f(t_n, u(t_n)) + \tau_n^{\text{EB}} \quad \text{waarbij} \quad \tau_n^{\text{EB}} = -\tfrac{1}{2}\,h^2\,u''(\xi_n). \tag{4.2}$$

Wegpoetsen van de defecten τ_n^{EB} geeft EB.

Experiment a. Gebruik de routine '`odeEB`' om de DV uit experiment a in 4A (`Linear2`) op te lossen. Met welke h (zo groot mogelijk) krijg je nog redelijke oplossingen? Vergelijk dit met de situatie voor EF. Hoe pakt het uit voor de DV (`vdPol`) in experiment c in 4B? Neem ook eens in de DV uit experiment a in 4A (`Linear2`) $\varepsilon = 10^{-6}$ en $h = 0.1$. Hoe doet `ode23` het voor deze DV met achtereenvolgens $\varepsilon = 10^{-4}$, $\varepsilon = 10^{-5}$, $\varepsilon = 10^{-6}$?

Verklaring. Probeer EB eens grafisch te interpreteren.
Verder geldt voor de fouten

$$e_n = \tfrac{1}{1-h\,\lambda_n}(e_{n-1} + \tau_n^{\mathrm{EB}}). \tag{4.3}$$

Opgave. *Bewijs* (4.3).

Experiment b. Wat verwacht je van EB voor stijve niet-lineaire DV's? Pas EB ook eens toe op (1.7) (`Emigration`) met $\alpha = 2$, $\gamma = 1$, $\varepsilon = -0.01$. Wat is de "optimale" stapgrootte? Hoe ligt dat voor EF? Moet je de "optimale" stapgrootte voor EB en EF nog herzien als je $\varepsilon = -0.001$ neemt?

Verklaring. De fouten gedragen zich min of meer als de oplossing van een lineaire DV (zie (3.8)).

Vraag 4F. Waarom werken we niet altijd met EB?

Experiment a. Probeer eens een stap EB uit te voeren "met de hand" voor (1.3).

Experiment b. Los (1.4) (`Immuun`) eens op met EB met $h = 0.1$. Hoe kan je verklaren dat je zo'n oplossing krijgt? Plot ook $h\,\lambda_i(t_n)$ als functie van t_n.

Verklaring. Kijk eens naar de foutvoortplantingsfactor $\frac{1}{1-h\,\lambda_n}$ van EB voor de λ's die we hier tegen kwamen.

Experiment c. Los (1.4) (`Immuun`) op met EF en met EB in beide gevallen met een betrouwbaarheid van 0.01 over het hele interval $[0, T]$, $T = 40$. Welke methode is het efficiëntst hier?

Verklaring. Hint: Is stabiliteit een probleem voor EF in dit voorbeeld?

> **Conclusie. a.** *Per stap is EB, zeker voor niet-lineaire en meerdimensionale DV's veel 'duurder' dan EF. Het gebruik van EB betaalt alleen af als EB met veel minder stappen toe kan.*
> **b.** *Ook EB kan stabiliteitsproblemen hebben. Die treden met name op als er grotere positieve eigenwaarden zijn.*
> **c.** *Als nauwkeurigheid de bepalende factor is en niet stabiliteit dan is EF veel efficiënter dan EB.*

In experiment c hierboven hebben we gezien dat het niet eenvoudig is om de EB stappen uit te voeren. De code `odeEB` speelt dat toch voor elkaar. In een volgend hoofdstuk zullen we zien hoe dat mogelijk is.

Vraag 4G. Zijn er methoden die een hogere orde nauwkeurigheid hebben dan EB (en EF) en waarmee je ook stijve DV's met wat grotere stapgrootte nauwkeurig kan oplossen?

Experiment. Pas de trapeziumregel toe op (1.2) (`Linear2`) met λ als in (1.2.d) en $\varepsilon = 10^{-6}$ en $h = 0.1$ (zie experiment c in 4E). Vergelijk de nauwkeurigheid die je nu krijgt met die die je krijgt met EB.

Opgave. *Leid een uitdrukking voor de foutvoortplanting met de Trapeziumregel af analoog aan (4.3).*

Opgave 4H. *EF komt tot stand via een eerste orde Taylor reeks. We onderzoeken in deze opgave waar een tweede orde Taylor reeks toe leidt.*

a. *Laat zien dat*

$$u(t_{n+1}) = u(t_n) + h\, f(t_n, u(t_n)) + \tfrac{1}{2}\, h^2\, f(t_n, u(t_n))\, \frac{\partial f}{\partial u}(t_n, u(t_n)) + \tau_n$$

geldt en geef een uitdrukking voor τ_n.

b. *Van welke orde is de resulterende methode?*

c. *Analyseer de stabiliteit van de methode.*

d. *Vergelijk de voor– en nadelen van deze methode met die van EF en van de Trapeziumregel.*

1.5 Toepassingen

In onderstaande "praktijk problemen" moet je proberen de inzichten, die je je in de vorige paragrafen eigen gemaakt, hebt toe te passen.

Opdracht 5A. Onderzoek welke numerieke methode, EF of EB, het efficiëntst (in ze kort mogelijke rekentijd) de DV (1.3) (`Flame`) met $\mu = \frac{1}{4}$ en $\kappa = 20$ nauwkeurig oplost (met een betrouwbaarheid van 0.01). Doet EB het veel beter dan EF? Waarom wel/niet?

Opdracht 5B. Los de DV (1.8) (`Body3`) op met `odeRK` met $h = 0.007$. We krijgen geen Arenstorf baan (gesloten kromme). Hoe komt dat? Is RK4 bij deze stapgrootte nog niet nauwkeurig genoeg of heeft RK4 problemen met de stabiliteit (zoom eens in rond de positie $(-\mu, 0)$ van A)? Probeer ook eens $h = 0.0068$. Voor welke h krijg je wel een Arenstorf baan? Kan je ook met een praktische stapgrootte met EF een Arenstorf baan krijgen? Verwacht je dat het beter gaat met EB?

Probeer ook eens `ode23`. Heb je een hogere `RelTol` nodig? Kan je zien hoe `ode23` het klaar speelt om zo efficiënt een redelijk nauwkeurige oplossing te krijgen? Plot de coördinaten van de oplossing ook eens als functie van de tijd en probeer te achterhalen met welke stapgrootte `ode23` werkt. Is die overal hetzelfde? Is er een 'fysische' verklaring waarom in sommige stukken van de baan 'nauwkeuriger' rekenwerk vereist is?

1.6 Samenvatting

Bij DV's wordt de fout in het numeriek resultaat bepaald door

1. *de wijze waarop de DV fouten voortplant,*
2. *de fouten die in iedere stap van het rekenproces geïntroduceerd worden,*
3. *de wijze waarop de numerieke oplosmethode fouten voortplant.*

In §§1.6.1-1.6.3 analyseren we deze aspecten voor de eendimensionale DV

$$\begin{cases} u'(t) = \lambda(t)u(t) + g(t), \quad t \in [0, T] \\ u(0) = u_0. \end{cases} \tag{6.1}$$

Hierin zijn λ en g gegeven reëel (eventueel complex) waardige functies en u_0 is een gegeven scalair. u is de exacte oplossing van (6.1). Daarna geven we in §1.6.4 aan hoe de resultaten er voor niet-lineaire – en meerdimensionale DV's er uit zien.

1.6.1 Foutvoortplanting door de DV

Een verstoring van bijvoorbeeld de beginvoorwaarde leidt tot een andere oplossing. In sommige gevallen kan die oplossing flink afwijken van de beoogde oplossing. Dat kan zelfs gebeuren voor heel kleine verstoringen. In zo'n geval mogen we niet verwachten dat we de DV numeriek nauwkeurig kunnen oplossen: immers in een numeriek proces zijn fouten onvermijdelijk (en fouten worden niet alleen in begin gemaakt!).

Om te achterhalen wat we op zijn minst aan fouten in de numerieke resultaten mogen verwachten analyseren we het effect van een verstoring ε op de beginvoorwaarde in (6.1).

Als $\widetilde{u}$ de 'verstoorde' oplossing is van (6.1), d.w.z. de oplossing van

$$\begin{cases} u'(t) = \lambda(t)u(t) + g(t), \quad t \in [0, T] \\ u(0) = u_0 + \varepsilon, \end{cases} \tag{6.2}$$

dan geeft

$$e(t) \equiv u(t) - \widetilde{u}(t) \tag{6.3}$$

aan hoe de verstoring van de beginvoorwaarde doorwerkt op de oplossing u in het tijdstip t. Trekken we (6.2) af van (6.1) dan zien we dat

$$\begin{cases} e'(t) = u'(t) - \widetilde{u}'(t) = \lambda(t)\, e(t), \\ e(0) = -\varepsilon. \end{cases} \tag{6.4}$$

Dus

$$e(t) = -c(t)\varepsilon, \quad \text{met} \quad c(t) \equiv \exp\left(\int_0^t \lambda(s)\, ds\right) : \tag{6.5}$$

$c(t)$ geeft aan hoe de verstoring door de DV voortgeplant wordt.
Met $\lambda_{\min} \equiv \min_t \mathrm{Re}(\lambda(t))$ en $\lambda_{\max} \equiv \max_t \mathrm{Re}(\lambda(t))$ geldt $e^{\lambda_{\min} T} \leq |c(t)| \leq e^{\lambda_{\max} T}$. Als $\lambda_{\min} > 0$ dan wordt de verstoring vergrotend voortgeplant, als $\lambda_{\max} < 0$ is dat verkleinend. Met bv $\lambda_{\min} = 2$ en $T = 20$ is $e^{\lambda_{\min} T} = 2.4\, 10^{17}$!

Een zelfde soort uitspraak geldt met betrekking tot verstoringen gemaakt op latere tijdstippen. Afhankelijk van de functie λ kan de voortplanting van een verstoring op sommige delen van $[0, T]$ groeiend zijn en op andere delen krimpend.

Als $\lambda(t)$ over een tijdstraject ter lengte h vrijwel constant is, dan geldt

$$e(t + h) \approx e^{h\, \lambda(t)}\, e(t) : \tag{6.6}$$

over een tijdstraject ter lengte h wordt de oude fout, d.w.z. het effect van een eerdere verstoring, door de DV voortgeplant met een factor $e^{h\, \lambda(t)}$. We mogen niet verwachten dat een numerieke oplosmethode fouten gunstiger zal voortplanten.

1.6.2 Lokale fouten in het rekenproces

Voor $h > 0$ voldoende klein geldt $u(t+h) \approx u(t) + h\,u'(t)$. Verder is $u'(t) = f(t, u(t))$. Dit leidt tot het volgend rekenproces:

$$u_{n+1} = u_n + hf(t_n, u_n), \qquad t_0 = 0, \quad t_{n+1} = t_n + h, \quad n = 0, 1, 2, \ldots, \tag{6.7}$$

waarin u_n een benadering is voor $u(t_n)$. Dit rekenschema heet de **Euler forward** (EF) methode. Voor de exacte oplossing in het tijdstip $t_{n+1} = t_n + h$ geldt:

$$u(t_{n+1}) = u(t_n) + h\,f(t_n, u(t_n)) + \tau_n, \quad \text{met} \quad \tau_n \equiv \tfrac{1}{2}\,h^2\,u''(\xi_n) \tag{6.8}$$

voor een zekere ξ_n in $[t_n, t_{n+1}]$ (gebruik een eerste orde Taylor benadering). De EF methode ontstaat dus door in ieder tijdstip t_n het **defect** τ_n weg te poetsen[19] en de exacte oplossing in t_n te vervangen door een benaderende.

In de numerieke literatuur schrijft men gewoonlijk f_n in plaats van $f(t_n, u_n)$. Deze notatie maakt de formules beknopter en we zullen deze notatie ook gebruiken. De afhankelijkheid van h wordt wel onder de tafel geveegd (maar dat was ook al in de notatie t_n gebeurd!). We schrijven λ_n in plaats van $\lambda(t_n)$.

De benadering u_{n+1} is gebaseerd op de benadering u_n van $u(t_n)$, die al met een zekere fout belast was. Bovendien wordt er nog een nieuwe fout gemaakt door het defect weg te poetsen. De **fout** (ook wel **globale fout** genoemd) $u(t_{n+1}) - u_{n+1}$ in het tijdstip t_{n+1} hangt dus af van de fout op het voorgaande tijdstip t_n en van een nieuwe fout, nl. het defect: EF maakt nieuwe fouten, maar "haalt ook nog eens oude koeien uit de sloot". Dit doet overigens iedere numerieke methode voor het oplossen van DV's. Noteren we de fout in t_n met e_n,

$$e_n \equiv u(t_n) - u_n,$$

dan geldt (trek (6.7) af van (6.8))

$$e_0 = 0, \qquad e_{n+1} = (1 + h\,\lambda_n)\,e_n + \tau_n, \quad n = 0, 1, 2, \ldots. \tag{6.9}$$

Als $\lambda = 0$, d.w.z., als $u' = g$ voor zekere $g = g(t)$, dan volgt uit (6.9) dat $e_n = \sum_{j<n} \tau_j$: de fout op een tijdstip is een cumulatie van defecten van voorgaande tijdstippen. Merk op dat een cumulatie van defecten een orde h groter kan zijn dan de defecten afzonderlijk:

$$|\sum_{j<n} \tau_j| \leq \sum_{j<n} |\tau_j| \leq n \max_j |\tau_j| \leq \frac{T}{h} \max_j |\tau_j|.$$

Onder omstandigheden kunnen de ongelijkheden gelijkheden zijn.
In het algemeen is $\lambda \neq 0$ en hebben we niet simpel weg te maken met een cumulatie van defecten maar ook nog met de *manier* waarop oude fouten worden voortgeplant. Afhankelijk van de grootte in absolute waarde van $1 + h\,\lambda_n$ worden oude fouten groeiend of krimpend voortgeplant. Deze factor wordt daarom de **foutvoortplantings factor** voor EF genoemd. Merk op dat

$$1 + h\,\lambda_n \approx e^{h\,\lambda_n} \quad \text{is als} \quad |h\,\lambda_n| \ll 1 \qquad (\text{denk aan } |h\,\lambda_n| \leq 0.2);$$

[19] De grootheid $\delta_n \equiv \tau_n/h$ noemt men ook wel **lokale discretisatiefout**. In deze visie wordt $(u(t+h) - u(t))/h$ gezien als benadering voor $u'(t)$. Ga na dat $(u(t+h) - u(t))/h - \delta(t) = u'(t) = f(t, u(t))$ tot EF leidt.

in dit geval planten EF en de DV zelf fouten op min of meer dezelfde manier voort (zie (6.6)).

Als λ begrensd is onafhankelijk van h dan kunnen we de fout majoreren door een constante maal h.

Stelling 1.1 *Als $|\lambda|$ begrensd is op $[0, T]$ dan is er een $C > 0$ zodat voor iedere h, die voldoende klein is, en iedere n waarvoor $t_n \leq T$, geldt dat*

$$|u(t_n) - u_n| \leq C \sum_{j<n} |\tau_j| \leq C \left(\tfrac{1}{2} T \max_{\xi \leq T} |u''(\xi)| \right) h. \tag{6.10}$$

Door h naar 0 te laten gaan volgt **convergentie**.

Stelling 1.2 *Als λ begrensd is dan convergeert EF* **uniform** *op $[0, T]$, d.w.z.,*

$$\sup_n |u(t_n) - u_n| \to 0 \quad \textit{voor} \quad h \to 0$$

Hierbij is het supremum genomen over alle n waarvoor $t_n \leq T$.

Schatting (6.10) is niet erg praktisch: in de praktijk kennen we C niet (zie §1.6.3) en u'' niet. Gelukkig vertoont de fout, voor h voldoende klein, een "h-gedrag": er is een functie w, die niet afhangt van h en n, zodanig dat

$$u(t_n) - u_n \approx h\, w(t_n), \tag{6.11}$$

voor h voldoende klein (zie stelling 1.4). Omdat de fout zo fraai gestructureerd is kunnen we aan de hand van de numerieke oplossingen zelf de fout schatten:

Stelling 1.3 *Als $\tau \in [0, T]$ en h is zo dat $\ell h = \tau$ en voldoende klein is, dan geldt*

$$u(\tau) - u_{2\ell}^{(2)} \approx \tfrac{1}{2} \left[u(\tau) - u_\ell^{(1)} \right], \tag{6.12}$$

hierin is $u_n^{(1)}$ de EF benadering met stapgrootte h en $u_n^{(2)}$ die met stapgrootte $\frac{1}{2}h$.

De fout halveert dus min of meer als de stapgrootte halveert. De fout moet uiteraard wel gemeten worden in geschikte punten.

Stelling 1.4 *Als f voldoende glad is ($f \in C^2$), dan is er een functie w en een $C > 0$ zodanig dat voor iedere h die voldoende klein is, geldt*

$$|u(t_n) - u_n - h\, w(t_n)| \leq C h^2, \quad \textit{voor alle } n \textit{ waarvoor} \quad t_n \in [0, T]. \tag{6.13}$$

In plaats van "er is een C zodat voor iedere voldoende kleine h (6.13) geldt" zeggen we

$$u(t_n) - u_n = h\, w(t_n) + \mathcal{O}(h^2) \quad \text{uniform op} \quad [0, T]. \tag{6.14}$$

Het woord "uniform" wijst erop dat dezelfde C gebruikt moet worden voor alle h, voldoende klein, en *alle* n. In woorden luidt (6.14):

"de fout $u(t_n) - u_n$ is gelijk aan $h\, w(t_n)$ op een **orde** h^2-term na.

1.6.3 Stabiliteit

Hoe groot moeten we de stapgrootte h kiezen om een numerieke oplossing te krijgen met een door ons gewenste nauwkeurigheid? Om efficiëntie redenen willen we h zo groot mogelijk nemen. De uitspraken in de stellingen 1.3 en 1.4 geven een hanteerbare foutschatting, maar vertellen niet wat "voldoende klein" is.

Schatting (6.10) geldt voor $C = e^{MT}$ waarbij $M = \max_t |\lambda(t)|$. De constante e^{MT} is vaak waanzinnig groot, ook in situaties waarin de DV zelf verstoringen dempend voortplant. De constante C is tot stand gekomen door de foutvoortplantingsfactoren $|1 + h\,\lambda_n|$ af te schatten door $1 + h\,M \leq e^{h\,M}$. In het algemeen is die schatting te pessimistisch. Als $|h\,\lambda_n|$ klein is voor alle n dan is $1 + h\,\lambda_n \approx e^{h\,\lambda_n}$ en plant EF de fout op min of meer dezelfde manier voort als de DV zelf. In dit geval is C min of meer gelijk aan de factor waarmee de DV zelf verstoringen kan vergroten en beter mogen we niet verwachten. Als $|1 + h\,\lambda_n| \leq 1$ voor (vrijwel) alle n dan worden oude fouten (vrijwel) niet vergroot. In dit geval is $C = 1$ ($C \approx 1$). De veel te grote C komen we tegen als $|h\,\lambda_n| \not\ll 1$ en $|1 + h\,\lambda_n| > 1$ voor een groot aantal tijstippen t_n (voor zeer veel n).

Voor een nauwkeurige numerieke oplossing moeten de defecten voldoende klein zijn en de amplificatie van de fout (door de foutvoortplanting) moet binnen de perken blijven. Zoals we net gezien hebben is dit laatste het geval als

(i) $|h\,\lambda_n| \ll 1$ voor alle n (denk aan $|h\,\lambda_n| \leq 0.2$) of (ii) $|1 + h\,\lambda_n| \leq 1$ voor (vrijwel) alle n.

Voorwaarde (ii) is een *stabiliteitseis*: oude fouten mogen niet vergroot worden. In zo'n geval is $C = 1$ in (6.10) en wordt de nauwkeurigheid in de numerieke oplossing gemajoreerd door een cumulatie van defecten. Als (i) of (ii) geldt dan wordt de fout in essentie volledig bepaald door Taylor resttermen en "voldoende klein" in de stellingen 1.3 en 1.4 betekent dat hogere orde Taylor resttermen verwaarloosbaar zijn ten opzichte van de laagste.

Als $-M \leq \lambda(t) \leq 0$ voor alle t dan geldt (ii) als $h \leq 2/M$ en voor dit soort h hangt verder het foutgedrag nog maar alleen af van Taylor resttermen. Voor bv. $M = 5$ is $2/M = 0.4$ en vormt de stabiliteitseis (ii) niet echt een extra beperking. Voor grote M (denk aan $M \geq 10^4$) kan dat wel het geval zijn. Andere numerieke oplosmethoden met betere stabiliteitseigenschappen kunnen dan uitkomst bieden. Een variant op de afleiding van EF leidt tot zo'n methode.

Voor kleine h geldt ook $u(t-h) \approx u(t) - h\,u'(t)$ of, equivalent hiermee, $u(t) \approx u(t-h) + h\,f(t, u(t))$. Dit leidt tot de **Euler Backward** (EB) methode:

$$\begin{cases} u_n = u_{n-1} + h\,f(t_n, u_n), \\ u(t_n) = u(t_{n-1}) + h\,f(t_n, u(t_n)) + \tau_n \quad \text{met} \quad \tau_n \equiv -\frac{1}{2}\,h^2\,u''(\xi_n). \end{cases} \tag{6.15}$$

De eerste relatie is het rekenschema. Voor de fout $e_n \equiv u(t_n) - u_n$ geldt

$$e_n = (1 - h\,\lambda_n)^{-1}\,(e_{n-1} + \tau_n)\,. \tag{6.16}$$

Fouten in EB worden niet vergroot als

$$\left|(1 - h\,\lambda_n)^{-1}\right| \leq 1 \quad \text{voor alle} \quad n. \tag{6.17}$$

In dat geval kan de fout gemajoreerd worden door $\sum_{j<n} |\tau_j|$.

Als bv. $\mathrm{Re}(\lambda(t)) \leq 0$ voor alle t, dan geldt (6.17) voor iedere h en de nauwkeurigheid in de numerieke oplossing wordt volledig bepaald door de defecten, ongeacht hoe groot $|\lambda(t)|$ is. Als $\mathrm{Re}(\lambda(t)) \leq 0$ dan hoeven we ons met EB alleen om de grootte van de defecten te bekommeren, terwijl we met EF ook nog rekening moeten houden met de stabiliteit. Voor zogenaamde **stijve** DV's, DV's waarbij $\mathrm{Re}(\lambda(t))$ negatief is maar absoluut groot en $|u''|$ niet al te groot, kunnen we met EB met veel grotere stappen werken dan met EF. Helaas kleeft er ook een nadeel aan EB. In iedere stap moet u_n uit de vergelijking $u_n - h\,f(t_n, u_n) = u_{n-1}$ *opgelost* worden, terwijl in EF u_n simpel volgt door *evaluatie* van $f(t_{n-1}, u_{n-1})$. EF wordt daarom een **expliciete** methode genoemd, en EB noemt men een **impliciete** methode.

1.6.4 Niet-lineaire en meerdimensionale DV's

Vervangen we in geval de eendim. DV niet-lineair is de functie $\lambda(t)$ in de uitspraken betreffende fouten door

$$\lambda(t) = \frac{\partial f}{\partial u}(t, u(t))$$

dan vinden we in essentie de correcte uitspraken voor niet-lineaire DV's. Dit laat zich als volgt verklaren. Stellen we voor het niet-lineaire geval relaties op voor de fout dan lopen we aan tegen $f(t, u(t)) - f(t, u(t) + e)$ hetgeen min of meer gelijk is aan $\frac{\partial f}{\partial u}(t, u(t))e$.

Het rekenschema voor meerdimensionale problemen is exact gelijk aan dat voor een-dimensionale. De discussie betreffende nauwkeurigheid en stabiliteit loopt ook niet wezenlijk anders. In plaats van scalairen komen in de discussie vectoren en matrices voor. De grootheden u_n^* en u_n zijn dan d-dimensionale vectoren terwijl voor $\frac{\partial f}{\partial u}$ de Jacobi matrix $(\frac{\partial f_j}{\partial u_i})$ gelezen moet worden: $\frac{\partial f_j}{\partial u_i}$ is het (i, j)-de element van die matrix. De absolute waarde $|v|$ van de vector $v = (v_1, \ldots, v_d)^T$ moet men, voor 'n geschikte "norm", lezen als de norm van de vector: bv. $(|v| =) \|(v_1, \ldots, v_d)^T\| = \max_{i \leq d} |v_i|$. Evenzo wordt de absolute waarde van een matrix de norm van die matrix. Normen van matrices zijn lastiger te interpreteren dan normen van vectoren. Omdat normen van matrices wellicht geen vertrouwde objecten zijn werken we dit niet verder uit. We volstaan met de opmerking dat de norm van een matrix gerelateerd kan worden aan de eigenwaarden van de matrix. Ruwweg gesproken betekent de stabiliteitsvoorwaarde (ii) voor het meerdimensionale geval dat $|1 + h\,\lambda| \leq 1$ moet zijn voor alle eigenwaarden λ van de Jacobi matrix $(\frac{\partial f_j}{\partial u_i}(t, u))$ (en dat voor alle relevante (t, u)).

Referenties

[1] F. Beukers, Modellen en Computers, College Diktaat, Mathematisch Instituut, Universiteit Utrecht, Utrecht, 1994.

[2] L.F. Shampine Numerical Solution of Ordinary Differential Equations, Chapman & Hal, New York, 1994.

[3] L.F. Shampine and M.W. Reichelt, *The* Matlab *ODE Suite*, SIAM J. Sc. Comput., 18 (1997), pp. 1–22.

[4] *The Student Edition of MATLAB, version 4, User's Guide*, Prentice-Hall, 1995

Hoofdstuk II

Benaderingsschema's door P. Zegeling

2 Interpolatie

Sleutelbegrippen: *Taylorontwikkeling, Lagrange interpolatie*

2.1 Inleiding

Tijdens het eerste deel van de cursus hebben we kennisgemaakt met enkele basisbegrippen uit de numerieke wiskunde (toegepast op DVs). Deel II van de cursus behandelt numerieke onderwerpen die nodig zijn voor het analyseren en verder uitwerken van de technieken uit Deel I. Bijvoorbeeld: hoe kan Euler-Backward gebruikt worden voor niet-lineaire DVs. Of, hoe kunnen we op een natuurlijke manier de nauwkeurigheid verhogen van een numerieke methode. Naar aanleiding van deel I zou je je bijvoorbeeld ook de volgende vraag kunnen stellen:
• hoe kom je aan nauwkeurige oplossingswaarden in tijdstippen $t \neq t_n$ als je de oplossing alleen als benadering kent in de discrete verzameling $t_0, t_1, \ldots, t_N$ zoals we in het geval van numerieke oplossingen van DVs hebben gezien?
Je zou natuurlijk Euler-Forward (of een andere methode) kunnen toepassen met een twee keer zo kleine stapgrootte, maar dit levert je (na minstens twee keer zoveel rekenwerk) slechts een benadering die precies op de helft van twee vorige benaderingen ligt. Wat echter als je geinteresseerd bent in een andere tussenliggende waarde?
Dit is typisch een situatie waarbij je te maken krijgt met interpolatie. Dit is niet het enige geval waarbij dit belangrijke numerieke onderwerp om de hoek komt kijken. Interpolatie is onder meer van belang in de volgende gevallen:
• het invullen van 'gaten' in tabellen met data die het resultaat zijn van fysische experimenten of observaties (zie Voorbeeld1C verderop)
• functies die weliswaar bekend zijn, maar waarvoor het simpelweg teveel tijd (en/of geheugen) kost om ze direkt te berekenen of bepalen. Hiervoor kan het vaak gunstiger zijn om de waarde via interpolatie uit een kleinere hoeveelheid functiewaarden te bepalen.
• interpolatie wordt gebruikt door CAD systemen om krommen en oppervlakken te tekenen uit bestanden met coordinaatposities die interactief gespecificieerd zijn door een gebruiker met behulp van 'computer-graphics' programmatuur.
• interpolatie is de basis van de meeste numerieke methoden voor het oplossen van DVs en het benaderen van integralen (zie volgende hoofdstuk). Door middel van interpolatie worden de onbekende grootheden ('functies') eerst benaderd met reeds bekende

waarden in specifieke punten. Deze geinterpoleerde waarden worden dan 'ingevuld' in de vergelijking of de integraal, die vervolgens kan worden opgelost c.q. uitgewerkt in termen van reeds bekende waarden.
Nog enkele specifieke voorbeelden:

Voorbeeld 1A. Een integraal die gerelateerd is aan de 'complete elliptische integraal' wordt gedefinieerd door

$$K(x) = \int_0^{\pi/2} \frac{\mathrm{d}\xi}{(1 - \sin(x^2)(\sin(\xi))^2)^{1/2}}. \tag{1.1}$$

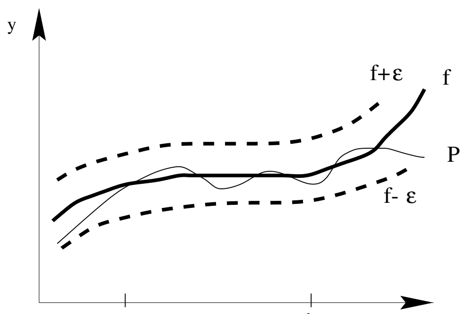
Indien voor enkele waarden van x in een tabel $K(x)$ bekend is, hoe kunnen we voor tussenliggende waarden een benadering van K krijgen (zonder de integraal te hoeven benaderen)?

Voorbeeld 1B. Een dergelijke vraag kan ook gesteld worden voor de Bessel functie van de eerste soort (en nulde orde):

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin(t))\, \mathrm{d}t. \tag{1.2}$$

Voorbeeld 1C. Veelal worden grootheden in experimenten gemeten voor eindig veel waarden van het argument. Bijvoorbeeld worden bevolkingsaantallen om de zoveel jaar geteld. Men is vaak geinteresseerd in de waarden in tussenliggende jaren of maanden. Hiervoor kan interpolatie o.a. gebruikt worden. Of indien men bijvoorbeeld de bevolking in 2010 wil 'voorspellen' (in dat geval spreken we natuurlijk van extrapolatie).

Voor interpolatie kan gekozen worden uit een verscheidenheid van 'simpelere' functies: polynomen, stuksgewijze polynomen, sommen van sinussen en cosinussen, sommen van exponentiele functies, en rationale functies (ratio's van polynoomfuncties). We zullen ons in Deel II bezighouden met interpolatie gebaseerd op polynomen. Hiervoor zijn enkele redenen aan te dragen. Ten eerste bestaan er belangrijke wiskundige stellingen die het benaderen van functies door polynomen rechtvaardigen. Ten tweede zijn polynomen gemakkelijk algebraisch te manipuleren en zijn ze efficient te gebruiken in computerberekeningen. Verder is de theorie betreffende polynomen relatief 'simpel' en kan eenvoudig bewezen worden dat polynoominterpolatie 'uniek' is.



FIGUUR 1. *De stelling van Weierstrass*

Het is dus niet verrassend dat één van de meest gebruikte klassen van functies die de verzameling van reële getallen op zichzelf afbeeldt de klasse van algebraische polynomen is; d.w.z. functies van de vorm

$$P_n(x) = a_0 + a_1 x + \ldots + a_n x^n, \qquad n \geq 0, \quad a_i \in \mathbb{R}. \tag{1.3}$$

Een belangrijke reden voor hun bruikbaarheid in de approximatietheorie is het feit dat ze continue functies uniform benaderen. Anders geformuleerd: gegeven een functie, die gedefinieerd en continu is op een gesloten interval, dan bestaat er een polynoom dat 'dicht bij' die functie ligt. Dit resultaat kan wat preciezer worden geformuleerd m.b.v. de stelling van Weierstrass (zie ook figuur 1):

Stelling 2.1 *Als f gedefinieerd en continu is op $[a, b]$, en $\varepsilon > 0$ is gegeven, dan bestaat er een polynoom P, gedefinieerd op $[a, b]$ met de eigenschap dat*

$$|f(x) - P(x)| < \varepsilon \qquad \forall x \in [a, b]. \tag{1.4}$$

Voor een bewijs wordt naar de literatuur verwezen.
Andere belangrijke redenen voor het beschouwen van polynomen bij het benaderen van functies zijn o.a. dat de afgeleide en de primitieve van een polynoom eenvoudig te bepalen zijn en dat het ook weer polynomen oplevert. Bovengenoemde stelling van Weierstrass is vooral belangrijk uit een theoretisch oogpunt. Voor praktische berekeningen is dit resultaat niet echt bruikbaar. Het zegt immers alleen iets over de existentie en niets over de constructie van een benadering. Hoe moeten we dan verder?
Laten we eerst eens het probleem beschouwen waarbij een polynoom van een bepaalde graad 'dicht bij' een gegeven functie ligt in een enkel punt.
Een polynoom P komt met een functie f in het punt x_0 overeen als $P(x_0) = f(x_0)$. Als we ook nog wensen dat het polynoom en de functie dezelfde 'richting' hebben in $(x_0, f(x_0))$ dan moet ook gelden $P'(x_0) = f'(x_0)$. Dit idee doorvoerend geeft dat het n-de graads polynoom dat het 'best' de functie f in de buurt van x_0 benadert zoveel mogelijk afgeleiden gemeenschappelijk heeft in x_0 met die van f. Maar dit is precies de conditie waar het n-de Taylorpolynoom van f in x_0 aan voldoet(!):

$$P_n(x) = f(x_0) + f'(x_0)(x - x_0) + \ldots + f^{(n)}(x_0)\frac{(x - x_0)^n}{n!}. \tag{1.5}$$

Zoals bekend heeft deze benadering van f als foutterm (restterm):

$$R_n(x) \equiv f(x) - P_n(x) = f^{(n)}(\xi(x))\frac{(x - x_0)^{n+1}}{(n + 1)!} \tag{1.6}$$

voor zekere $\xi(x)$ tussen x en x_0.

2.2 Taylor benadering

N.a.v. het bovenstaande lijkt de volgende vraag gerechtvaardigd:

Vraag 2A. Is het mogelijk om met behulp van een Taylorontwikkeling een nauwkeurige benadering te verkrijgen van een gegeven functie?

Experiment. Bekijk de functie $f(x) = e^x$ rond $x = 0$. [20] Gezocht is de functiewaarde $f(1)$. Verbetert de benadering door het toevoegen van meer termen in de benadering? Wanneer wordt de fout in $x = 1$ kleiner dan, zeg, 10^{-8}? Probeer ook eens $x = -5$.

Verklaring. Het is duidelijk dat, om een enigszins nauwkeurige oplossing te krijgen, we meerdere termen in de ontwikkeling moeten meenemen.

[20] Gebruik de file `taylexp.m` om enkele experimenten met een Taylorbenadering van oplopende graad uit te voeren.

Opgave. *Bepaal m.b.v. de restterm het kleinste gehele getal n dat nodig is om $f(x) = e^x$ in $x = 1$ te benaderen met een nauwkeurigheid van 10^{-8}, gebruikmakend van het n-de Taylorpolynoom rond $x_0 = 0$. Vergelijk je resultaat met het computerexperiment.*

Vraag 2B. Gezien het 'succes' in de vorige opgave is de volgende vraag gerechtvaardigd:
Is deze benaderingsmethode van functiewaarden nu algemeen bruikbaar (of zit er ergens nog een addertje onder het gras?)?

Experiment. Doe hetzelfde experiment als eerder, maar nu met de functie $f(x) = 1/x$. [21] De functie wordt bekeken rond $x_0 = 1$ en gezocht is de waarde $f(3)$. Wat gebeurt er met de fout bij het ophogen van het aantal termen van de Taylorreeks? Probeer ook eens enkele andere waarden van x.

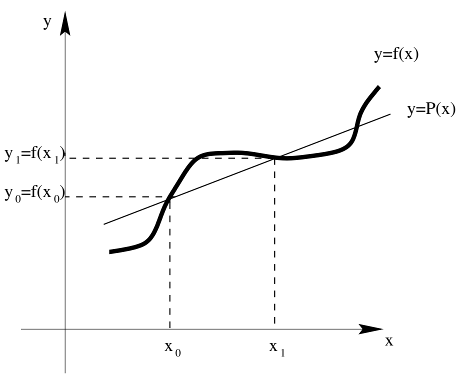
Verklaring. Een Taylorontwikkeling is alleen lokaal geldig en dus i.h.a. niet bruikbaar voor het benaderen van 'willekeurige' functiewaarden.

Opgave. *Onderzoek de foutterm in de Taylorreeks van $f(x) = 1/x$ rond $x_0 = 1$ en verklaar hiermee het experimentele resultaat. Welke waarde van n is nodig om $f(1.25)$ met een nauwkeurigheid van 10^{-8} te benaderen?*

2.3 Lineaire Interpolatie

Aangezien Taylorpolynomen de eigenschap hebben om voor de benadering alle informatie te halen uit een enkel punt x_0, zullen de fenomenen zoals boven waargenomen zeer vaak kunnen optreden. Dit zal i.h.a. de bruikbaarheid van Taylorbenaderingen beperken tot situaties waarbij we benaderingen nodig hebben in punten die dicht bij het punt x_0 liggen. Voor algemenere toepassingen zou het dus misschien nuttig kunnen zijn om methoden te bekijken waarbij informatie wordt gebruikt in verscheidene punten. Het gebruik van Taylorpolynomen in de numerieke wiskunde is in de eerste plaats vooral gericht op het analyseren en afleiden van numerieke technieken en dus niet zozeer op het benaderen van functiewaarden zelf.

Vraag 3A. In het voorgaande experiment hebben we gezien dat het gebruik van één functiewaarde i.h.a. niet genoeg is om een andere 'onbekende' functiewaarde te benaderen. Zou het mogelijk zijn om de benadering iets 'minder lokaal' te maken en tegelijkertijd de interpolatiemethode toch relatief simpel te houden?



FIGUUR 2. *Lineaire interpolatie*

[21]gebruik `tayl1x.m`.

Geeft het gebruik van bijvoorbeeld twee functiewaarden (zie figuur 2) een (duidelijke) verbetering t.o.v. één referentiewaarde?
Indien we twee functiewaarden gebruiken ligt het voor de hand om de functie te gaan benaderen met een lineair polynoom $P_1(x)$ (rechte lijn). Hiermee zijn we beland bij het begrip 'lineaire interpolatie'. De twee bijbehorende x-waarden heten steunpunten van de interpolatie.

Experiment. Benader [22] de functie $f(x) = \sin(x)$ voor $x \in [0, 1]$. Speel met enkele x-waarden om een indruk te krijgen van de fout bij lineaire interpolatie. Herhaal het experiment uit 1.2, maar nu voor $f(x) = \frac{1}{x}$ m.b.v. lineaire interpolatie. Geeft lineaire interpolatie voor deze functie een verbetering t.o.v. het alsmaar uitbreiden van een Taylorontwikkeling?

Verklaring. Zie verderop.

Opgave. *Leidt een uitdrukking af voor het lineaire interpolatiepolynoom $P_1(x)$ in de 'steunpunten' x_0 en x_1 in termen van $f(x_0)$ en $f(x_1)$.*

Vraag 3B. Hoe hangt de fout (bijv. de maximale waarde van de absolute fout over een bepaald interval) af van de gekozen intervallengte h? Gelden hier net zulke regels als die we gezien hebben voor DVs in deel I?

Experiment. Speel met de lengte h. [23] Doe dit eerst voor een vast tussenpunt m (zie Matlab-file). Halveer de staplengte h enkele malen. Welke factor (zelf berekenen!) tussen fracties van opeenvolgende maximale opeenvolgende fouten zie je verschijnen? Welke afhankelijkheid van de staplengte suggereert dit?

Verklaring. Bij het verkleinen van de staplengte neemt de fout af met $\mathcal{O}(h^2)$ (voor h klein genoeg). Verder hangt de fout af van de tweede afgeleide van f in een onbekend tussenpunt.

Opgave. *Laat zien dat de fout bij lineaire interpolatie in een punt x voldoet aan*

$$f(x) - P_1(x) = \frac{1}{2}(x - x_0)(x - x_1)f''(\xi(x)). \tag{3.1}$$

Hierbij is $\xi = \xi(x)$ een zeker tussenpunt liggend tussen de kleinste en de grootste waarde van x_0, x_1, en x.

Opgave. *Stel dat er een tabel is samengesteld voor de functie $f(x) = e^x$ op het interval $[0, 1]$ en dat het verschil tussen opeenvolgende x-waarden (de stapgrootte) gelijk aan h is. Hoe groot (klein) moet h zijn om met lineaire interpolatie een absolute fout van hoogstens 10^{-6} te krijgen?*

Vraag 3C. Gelden bovenstaande beschouwingen over de fout bij lineaire interpolatie ook op andere intervallen dan $[0, 1]$?

Experiment. Bekijk bijvoorbeeld eens het interval [-0.2,0.2] voor $f(x) = \sin(x)$. Zien we bij halveren weer een factor 4 terug (neem nu $m = 0$)? [24]

Verklaring. In $x = 0$ is de tweede afgeleide van de sinus gelijk aan 0. De laatste factor in de foutuitdrukking speelt nu dus een bijzondere rol (ξ komt zeer dicht bij 0 te liggen voor kleine h).

Opgave. *Bewijs dat de maximale absolute fout E in dit laatste geval voldoet aan*

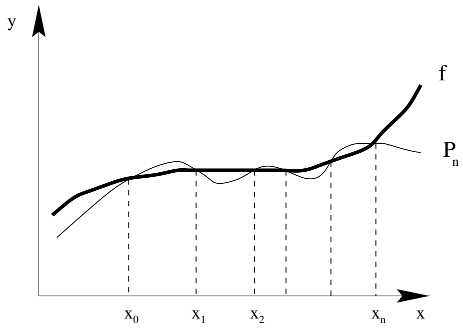
$$\lim_{\delta \to 0} \frac{E}{\delta^3} = \frac{1}{9\sqrt{3}}|f'''(q)| \tag{3.2}$$

[22] Maak gebruik van de file `intpol.m`.
[23] Gebruik de file `opgfou1.m` voor de functie $f(x) = \sin(x)$.
[24] Gebruik `opgfou1.m`.

waarbij verondersteld wordt dat $f^{''}(q) = 0$, $f^{'''}(q) \neq 0$ met steunpunten q en $q+\delta$ voor lineaire interpolatie. Verklaar hiermee de factor in het experiment.

2.4 Lagrange Interpolatie

Als natuurlijke generalisatie van lineaire interpolatie kan men de benadering beschouwen van een functie door een n-de graads polynoom P_n, zodanig dat de waarden van f en P_n in een vooraf gegeven aantal (verschillende) punten $x_0, \ldots, x_n$ overeenstemmen. In dat geval spreken we van n-de orde interpolatie (zie figuur 3). Het vermoeden zal bestaan dat de nauwkeurigheid toeneemt net als de bijbehorende hoeveelheid rekenwerk (# f-evaluaties). Om te zien of dit vermoeden juist is gaan we eerst weer enkele experimenten uitvoeren.



FIGUUR 3. *Interpolatie in $n+1$ punten*

Vraag 4A. We willen de nauwkeurigheid verhogen ('sneller dan $\mathcal{O}(h^2)$ zoals bij lineaire interpolatie'). Zouden we dit kunnen bewerkstelligen door interpolatie toe te passen op meer dan twee punten?

Experiment. Neem eerst weer $f(x) = \sin(x)$. [25] Kies eerst drie steunpunten en controleer of de fout inderdaad kleiner is geworden. Doe vervolgens hetzelfde met 4, 5 en 6 steunpunten.

Verklaring. De functie wordt steeds meer 'ingeklemd' door hogere graadspolynomen hetgeen de nauwkeurigheid overeenkomstig verhoogt.

Opgave. *Laat zien dat het n-de graads polynoom*

$$P_n(x) = \sum_{k=0}^{n} f(x_k) L_{k\,n}(x),$$

waarin

$$L_{k\,n}(x) = \frac{(x-x_0)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)}$$

de functie $f(x)$ interpoleert in de $n+1$ steunpunten $x_0, \ldots, x_n$. Dit is de zogenaamde Lagrange formulering (er zijn veel manieren om één en hetzelfde polynoom te beschrijven; dit is er één van). Er kan bewezen worden (hoeft hier niet) m.b.v. stellingen uit de Analyse, dat de fout bij Lagrange interpolatie in het algemene geval voldoet aan

$$f(x) - P_n(x) = (x-x_0)(x-x_1)\ldots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}, \tag{4.1}$$

[25] Gebruik de Matlabfile: `intpol.m`.

waarbij ξ een onbekend punt is liggend tussen $\min(x_0, \ldots, x_n, x)$ *en* $\max(x_0, \ldots, x_n, x)$.

Vraag 4B. Heeft het zin om de graad van P_n zeer hoog te nemen? Of een stap verder: treedt er 'convergentie' op indien we de graad van P_n 'naar oneindig' laten gaan, waarbij we de steunpuntenverdeling uniform houden?

Experiment. Hoog het aantal steunpunten op in je experimenten (voor $\sin(x)$) en probeer hiermee een antwoord op de eerste vraag te krijgen. Neem in je beschouwingen ook de 'kosten' mee die hoger worden naarmate je meer steunpunten meeneemt in de berekeningen.
Neem vervolgens de 'extreem' gladde functie $f(x) = 1/(1+x^2)$ en pas hetzelfde experiment nogmaals toe. [26] Wat gebeurt er met de benadering (let vooral op de randen)?

Verklaring. Indien we het produkt $(x-x_0)(x-x_1)\ldots(x-x_n)$ bekijken in de foutuitdrukking van interpolatie voor toenemende n, dan nemen de extremen van deze produktfunctie toe aan de randen. Bovendien spelen de afgeleiden van de betreffende functie een belangrijke rol in de foutuitdrukking.

Opgave. *Onderzoek en teken deze produktfunctie voor* $n = 1, \ldots, 6$.

Vraag 4C. We hebben gezien dat de fout naar de randen toe sterk toeneemt, als het aantal steunpunten en dus de graad van polynoombenadering in interpolatie op een uniforme puntenverdeling toeneemt. Zou het zin hebben de steunpunten zodanig te kiezen dat er een verdichting optreedt naar de rand toe om dit 'opblaaseffect' tegen te gaan?

Experiment. [27] Hoog de orde op net als in het vorige experiment. Zie je een verbetering?

Verklaring. Er kan bewezen worden (maar dat doen we hier niet), dat voor de speciale puntenverdeling $x_i = \frac{a+b}{2} - \frac{b-a}{2}\cos(\frac{(2i+1)\pi}{2(n+1)})$ $(i = 0, \ldots, n)$ geldt:

$$\max_{x\in[a,b]} |f(x) - P_n(x)| = o\left(\frac{\log(n)}{n^{k-1}}\right), \quad \text{als} \quad f \in \mathcal{C}^k. \tag{4.2}$$

Deze steunpunten x_i zijn nulpunten van zogenaamde Chebyshev-polynomen. In feite wordt door deze keuze van steunpunten $\max_{x\in[a,b]} |(x-x_0)(x-x_1)\ldots(x-x_n)|$ geminimaliseerd in een zekere norm.

2.5 Toepassingen

We gaan nu de opgedane kennis proberen toe te passen op enkele 'praktijk'-problemen.

Opdracht 5A. Bereken de waarde van de elliptische integraal K uit (1.1) in $x = 3.5$ m.b.v. een 2de-graads interpolatiepolynoom.

x	1	4	6
$K(x)$	1.5709	1.5727	1.5751

N.B. de 'werkelijke' waarde is 1.572 25.

Opdracht 5B. Bereken de waarde van de Bessel-functie uit (1.2) voor $x = 1.5$. Doe dit voor meerdere keuzes van n in Lagrange-interpolatie. Vergelijk de numerieke resultaten met elkaar (indien mogelijk ook voor verschillende keuzes van steunpunten).

[26] Gebruik `rungeu.m`.
[27] Gebruik de file `rungec.m` waarin een speciale keuze is gemaakt voor de steunpuntenverdeling.

x	1.0	1.3	1.6	1.9	2.2
$J_0(x)$	0.765 197 7	0.620 086 0	0.455 402 2	0.281 818 6	0.110 362 3

NB. de werkelijke waarde is 0.511 827 7.

Opdracht 5C. De bevolking van de Verenigde Staten van Amerika wordt eens in de 10 jaar 'geteld'. Tussen 1930 en 1980 heeft dit de volgende tabel opgeleverd. De aantalen staan in miljoenen.

jaar	1930	1940	1950	1960	1970	1980
populatie V.S.	123.203	131.669	150.697	179.323	203.212	226.505

Schat m.b.v. Lagrange-interpolatie de bevolking in de jaren 1920, 1965 en 2000. In 1920 waren er ongeveer 105711000 mensen in de V.S. Hoe nauwkeurig zijn de geschatte waarden voor 1965 en 2000? (zoek *eventueel* op internet naar de 'juiste' waarden) Kies verschillende datasets uit de tabel. Zie je verschillen? Verklaring?

Opdracht 5D. Maak gebruik van Matlab om een tabel functiewaarden te genereren van de Gamma-functie:

$$\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t)dt \tag{5.1}$$

voor $x = 3.11, 3.12, \ldots, 3.17$. Probeer met een kleine aanpassing van lineaire interpolatie de waarde x te vinden waarvoor $\Gamma(x) = 2.3$. Dit proces heet 'inverse interpolatie'.

Opdracht 5E. De volgende tabel (van een 'gladde' functie) bevat twee type-fouten:

x	0.0	0.1	0.2	0.3	0.4	0.5
$f(x)$	1.000	1.161	1.248	1.268	1.224	1.125
x	0.6	0.7	0.8	0.9	1.0	
$f(x)$	0.976	0.783	0.552	0.289	0.001	

Corrigeer de type-fouten en geef een expliciete uitdrukking voor deze functie.

2.6 Samenvatting

• Interpolatie met Taylorpolynomen: laat f een functie zijn met $N+1$ continue afgeleiden op een zeker interval dat de punten x_0 en x bevat. Dan geldt:

$$f(x) = \sum_{p=0}^{N} (x - x_0)^p \frac{f^{(p)}(x_0)}{p!} + (x - x_0)^{N+1} \frac{f^{(N+1)}(\xi)}{(N+1)!} \tag{6.1}$$

voor zekere $\xi = \xi(x)$ tussen x_0 en x. Deze formule is vooral van theoretisch belang voor het onderzoeken van numerieke methoden. In de praktijk kunnen we met deze vorm van interpolatie niet veel beginnen; immers de Taylor-formule is lokaal geldig, d.w.z. voor punten x 'ver' van x_0 kan de fout behoorlijk oplopen of zelfs exploderen....
• *Lineaire interpolatie.* Laat a en b twee punten in $\mathbb{R}$ zijn met $a < b$, en f een functie die gedefinieerd is op $[a, b]$. Laat verder x_0 en x_1 twee verschillende punten zijn in $[a, b]$. Het **lineaire interpolatiepolynoom** van f op de **steunpunten** x_0 en x_1 is dan gedefinieerd als het hoogstens eerstegraads polynoom P_1 dat in x_0 en x_1 dezelfde waarde aanneemt als f.

Men gaat snel na dat het bedoelde polynoom gegeven wordt door

$$P_1(x) = f(x_0) + \frac{x - x_0}{x_1 - x_0}(f(x_1) - f(x_0)) = \frac{x_1 - x}{x_1 - x_0} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1). \quad (6.2)$$

Grafisch komt dit er op neer dat men op $[\,a,\, b\,]$ de grafiek van f benadert door de rechte lijn door de punten $(x_0,\, f(x_0))$ en $(x_1,\, f(x_1))$.

Stelling 2.2 *Laat x_0 en x_1 punten in $[\,a,\, b\,]$ zijn, $x_0 \neq x_1$. Zij $f \in C[\,a,\, b\,]$, en laat f' en f'' bestaan op $]\,a,\, b\,[$. Dan geldt voor het lineaire interpolatiepolynoom P_1 van f op de steunpunten x_0 en x_1, en voor elke $x \in [\,a,\, b\,]$: er is een $\xi \in \,]\,x_0,\, x_1,\, x\,[$ zo dat*

$$f(x) - P_1(x) = \frac{1}{2}(x - x_0)(x - x_1) f''(\xi). \quad (6.3)$$

• *Lagrange interpolatie.* Als natuurlijke generalisatie van lineaire interpolatie kan men de approximatie beschouwen van een functie f door een hoogstens n-de graads polynoom P_n, zodanig dat de waarden van f en P_n in een vooraf gegeven $n+1$-tal verschillende punten $x_0, \ldots, x_n$ overeenstemmen. Men spreekt dan van **n-de orde Lagrange interpolatie**.

Een polynoom P_n dat aan deze eisen voldoet is gemakkelijk aan te geven:

$$P_n(x) = \sum_{k=0}^{n} f(x_k) L_{k\,n}(x) \quad (6.4)$$

waarin

$$L_{k\,n}(x) = \frac{(x - x_0) \ldots (x - x_{k-1})(x - x_{k+1}) \ldots (x - x_n)}{(x_k - x_0) \ldots (x_k - x_{k-1})(x_k - x_{k+1}) \ldots (x_k - x_n)}. \quad (6.5)$$

Het polynoom P_n is het **Lagrange interpolatiepolynoom** van f op de steunpunten $x_0, \ldots, x_n$. Het rechterlid van (1.4) is de *Lagrange representatie* van P_n (elk polynoom heeft talloze representaties!).

Stelling 2.3 *Laat $x_0, \ldots, x_n$ verschillende punten zijn in $[\,a,\, b\,]$. Laat $f \in C[\,a,\, b\,]$, en laat $f', f'', \ldots, f^{(n+1)}$ bestaan op $]\,a,\, b\,[$. Dan geldt voor het bijbehorende interpolatiepolynoom P_n en voor elke $x \in [\,a,\, b\,]$: er is een $\xi \in \,]\,x_0, x_1, \ldots, x_n, x\,[$ zo dat*

$$f(x) - P_n(x) = (x - x_0) \cdot (x - x_1) \cdot \ldots \cdot (x - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!}. \quad (6.6)$$

• Chebyshev-verdeling van steunpunten: Aangezien het ophogen van de orde $(n \to \infty)$ in sommige gevallen, zelfs voor gladde functies, tot divergentie kan leiden (zie bijvoorbeeld de functie $f(x) = 1/(1 + x^2)$), kan het handig zijn om een niet-uniforme puntenverdeling te kiezen. Er kan bewezen worden, dat voor de speciale puntenverdeling $x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos(\frac{(2i+1)\pi}{2(n+1)})$ $(i = 0, \ldots, n\ )$ geldt:

$$\max_{x \in [a,b]} |f(x) - P_n(x)| = o\left(\frac{\log(n)}{n^{k-1}}\right), \quad \text{als} \quad f \in \mathcal{C}^k. \quad (6.7)$$

Deze steunpunten zijn nulpunten van zogenaamde Chebyshev-polynomen. In feite wordt door deze keuze van steunpunten $\max_{x \in [a,b]} |(x - x_0) \cdot (x - x_1) \cdot \ldots \cdot (x - x_n)|$ geminimaliseerd.

3 Integratie

SLEUTELBEGRIPPEN: *Kwadratuurformules, gerepeteerde versie*

3.1 Inleiding

In dit onderdeel gaan we kijken naar numerieke technieken voor het bepalen van de integraal

$$\int_a^b f(x)\,\mathrm{d}x \tag{1.1}$$

voor gegeven beginpunt a en eindpunt b. De waarde van deze integraal komt, zoals bekend, overeen met de oppervlakte onder de grafiek van f tussen a en b. Uit de analyse weten we dat het bepalen van de oppervlakte onder een kromme samenhangt met het vinden van een primitieve van de bijbehorende functie. Voor verschillende interessante situaties hebben we truukjes geleerd om aan zo'n primitieve functie te komen. Helaas geldt dit niet voor algemene functies $f(x)$... voor de meeste integralen bestaan geen expliciete analytische uitdrukkingen! Daarom is het gerechtvaardigd om methoden te bestuderen voor het numeriek benaderen van integralen van de vorm (1.1).
Er is, om aan te sluiten op Deel I, nog een tweede argument om dit te gaan doen. Realiseer je dat een gewone DV te herschrijven is naar een integraalvergelijking en door het 'nauwkeuriger' numeriek berekenen van die integraal zouden we wel eens een nauwkeurigere methode voor het oplossen van de DV kunnen vinden!

Opgave. *Euler-forward kan ook op een andere manier, dan beschreven in Deel I, afgeleid worden. Herschijf hiertoe de DV* (1.1) *naar de integraalvergelijking:*

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, u(\tau))d\tau. \tag{1.2}$$

Benader vervolgens de integraal met de 'rectangular-rule'

$$\int_{t_n}^{t_{n+1}} f(\tau, u(\tau))d\tau \approx (t_{n+1} - t_n) f(t_n, u(t_n)). \tag{1.3}$$

Probeer deze benadering te interpreteren (maak ook een tekening). Zie in dat dit weer Euler-forward oplevert met bijbehorende (lokale) fout zoals eerder beschreven.

Vraag 1A. Hoe goed is deze benadering voor (1.1) eigenlijk?

Experiment. Experimenteer met [28] toegepast op $f(x) = \cos(x)$ en $[a, b] = [0, \pi/2]$. Probeer uit te vinden hoe snel de fout naar nul gaat bij halvering van het interval (splits hiertoe steeds het interval in twee keer zo kleine stukjes; dit heet 'repeteren').

Verklaring. De fout is $\mathcal{O}(h)$ voor $h \to 0$, waarbij $h = (b - a)/k$ en k het aantal gebruikte intervallen is.

Opgave. *Laat zien dat:*

$$\int_a^b f(x)\,\mathrm{d}x = h \sum_{i=0}^{k} f(a + (i-1)h) + \frac{b-a}{2} h f'(\xi) \tag{1.4}$$

Deze methode (die dus samenhangt met Euler-Forward voor DV-en) is blijkbaar niet erg nauwkeurig. Voordat we nauwkeurigere numerieke methoden gaan afleiden en

[28] de file `rectang.m`

bestuderen, volgen hier eerst enkele voorbeelden van integralen die daarbij aan bod komen.

Voorbeeld 1B.

$$\int_{-1}^{1} \exp(x)\,\mathrm{d}x. \tag{1.5}$$

We kennen weliswaar de waarde van deze integraal ($= \exp(1) - \exp(-1)$), maar het is handig om dit voorbeeld te gebruiken voor het testen van de nauwkeurigheid van een numerieke integratietechniek.

Voorbeeld 1C.

$$\int_{0}^{1} \exp(-x^2)\,\mathrm{d}x. \tag{1.6}$$

De integrand speelt een zo belangrijke rol in de wiskunde dat er een speciale naam aan toegekend is: de Gauss-functie (of in een andere context: $\frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\,\mathrm{d}t = \mathrm{erf}(x)$; de 'error-functie'). Er is geen primitieve functie van de integrand bekend!

Voorbeeld 1D. De 0-de orde Besselfunctie $J_0(x)$ (zie ook het vorige hoofdstuk over interpolatie) wordt gedefinieerd door de integraal

$$\frac{1}{\pi} \int_{0}^{\pi} \cos(x \sin(\vartheta))\,\mathrm{d}\vartheta. \tag{1.7}$$

Voorbeeld 1E.

$$\int_{0}^{1} \sqrt{x} \cos(x)\,\mathrm{d}x. \tag{1.8}$$

De integrand heeft een singulariteit in zijn eerste afgeleide in $x = 0$. De exacte waarde van deze integraal is 0.531 202 683 084 509 3. Wat voor effect zou de singulariteit hebben op de numerieke oplossing, aangezien er in de foutuitdrukking, zoals we al meerdere keren gezien hebben, vaak afgeleiden van de gebruikte functie voorkomen?

Voorbeeld 1F. De relatief 'milde' integrand in

$$\int_{0}^{\pi} \frac{1}{1 + \cos^2(x)}\,\mathrm{d}x. \tag{1.9}$$

is vanwege zijn periodieke karakter een interessante testfunctie.

Voorbeeld 1G. Een numerieke integratiemethode moet gebruikt worden voor de volgende situatie: stel dat een race-auto in 84 seconden een volledige ronde rijdt op een circuit en dat de snelheid van de bolide steeds in intervallen van 6 seconden wordt gemeten m.b.v. radar. Als deze snelheden nu gegeven worden (in m/sec) door de volgende tabel

tijd	0	6	12	18	24	30	36	42
snelheid	124	134	148	156	147	133	121	109
tijd	48	54	60	66	72	78	84	
snelheid	99	85	78	89	104	116	123	

dan is de opdracht: bereken de lengte van het circuit.

Voorbeeld 1H. Een voorbeeldje uit de vloeistofmechanica:
Een deeltje met massa m beweegt door een vloeistof en ondervindt een visceuze weerstand R, die een functie is van de snelheid v. De relatie tussen de weerstand R, snelheid v, en de tijd t wordt gegeven door de vergelijking

$$t = \int_{v(t_0)}^{v(t)} \frac{m}{R(u)}\,\mathrm{d}u. \tag{1.10}$$

Stel dat voor een bepaalde vloeistof $R(v) = -v\sqrt{v}$, waarbij R in $Newtons$ en v in m/sec worden uitgedrukt. Gevraagd is de tijd t die het deeltje nodig heeft om afgeremd te worden tot $v = 5\,\mathrm{m/sec}$, als gegeven is dat $m = 10\,\mathrm{kg}$ en $v(0) = 10\,\mathrm{m/sec}$.

Vraag 1I. Hoe (nauwkeurig) integreert Matlab zelf model (1.1) eigenlijk?

Experiment. Gebruik [29] voor het oplossen van (1.5) en (1.6). [30] Geeft Matlab betrouwbare oplossingen?

Verklaring. Het gedrag van de fout in numerieke integratie hangt af van de eigenschappen van de integrand en de integratiegrenzen. We kunnen dus niet klakkeloos numerieke resultaten accepteren. Nader onderzoek is dus nodig!

3.2 Kwadratuur

Het numeriek bepalen van een integraal wordt ook vaak aangegeven met de term kwadratuur [31]. Specifieke benaderingsformules heten dan ook kwadratuur-regels.

Vraag 2A. Gegeven de grafiek van een functie $f(x)$ op een interval $[a, b]$. Hoe kunnen we aan een simpele kwadratuurformule komen? Een idee zou kunnen zijn om eerst de functie f te benaderen met een lineaire functie p op $[a, b]$ (zie vorige hoofdstuk) en vervolgens de integraal van f te benaderen met die van p.

Experiment. Bepaal hiermee een numerieke oplossing van (1.5). [32] Is de oplossing nauwkeurig?

Verklaring. De oppervlakte onder de grafiek van f wordt benaderd met de oppervlakte van het trapezium begrensd door a, b, $f(b)$ en $f(a)$ ($\rightarrow$ '**trapeziumregel**').

Vraag 2B. Hoe kunnen we een nauwkeurigere benadering krijgen zonder van het bovenstaande principe af te stappen? Wat te denken van het verdelen van het interval $[a, b]$ in k stukjes ter lengte h en op elk stukje de integraal te benaderen m.b.v. de trapeziumregel (dit heet: de '**gerepeteerde trapeziumregel**')?

Experiment. [33] Maak een tabel waarin je de volgende gegevens verwerkt: de 'stapgrootte' h, de benadering (die we met $T(h)$ zullen aangeven), en de fout $R(h) \equiv \int_{-1}^{1} \exp(x)\,\mathrm{d}x - T(h)$. Kies bijvoorbeeld een rij stapgrootten die steeds een factor 2

[29] `quad.m` (standaard beschikbaar, gebruik evt. help quad)

[30] De integrand kan worden aangepast in de file `integr.m`.

[31] De naam kwadratuur verwijst, in oude artikelen, naar het vinden van een vierkant waarvan de oppervlakte gelijk is aan de oppervlakte onder de kromme.

[32] Gebruik hiervoor `trapez.m` (aanroep: `trapez`; zie ook help trapez) met $k = 1$ (d.w.z. gebruik het hele interval $[a, b]$ 'in 1 keer').

[33] Doe enkele experimenten met `trapez.m` door met $k > 1$ (en dus h) te spelen.

afneemt. Wordt de benadering nauwkeuriger bij afnemende h? Welke factor neemt de fout af bij halveren van h? Hadden we dit kunnen verwachten?

Verklaring. De grafiek van f wordt met toenemende k steeds beter benaderd door de op deze wijze geconstrueerde stuksgewijs lineaire functie (en dus ook de oppervlakte onder de grafiek).

Opgave. *Laat zien dat voor de fout ('restterm') bij de trapeziumregel, op hogere orde termen na, geldt:*

$$R(h) \approx -\frac{h^2}{12}[f'(b) - f'(a)]. \tag{2.1}$$

Vraag 2C. Is het mogelijk om tijdens het halveren van de intervalletjes de betrouwbaarheid van de verkregen benadering te bepalen? D.w.z. kunnen we de fout schatten tijdens het rekenproces?

Experiment. [34] Bereken eens voor een paar waarden van h de grootheid $[T(2h) - T(h)]\,/\,[T(h) - T(h/2)]$, de 'echte' fout $R(h)$, en de grootheid $\frac{1}{3}\,[T(h) - T(h/2)]$.

Verklaring. Als de fout bij halveren ongeveer $4\times$ zo klein wordt (zie (2.1), dan is $[T(2h) - T(h)]\,/\,[T(h) - T(h/2)] \approx 4$, en is $R(h) \approx \frac{1}{3}\,[T(h) - T(h/2)]$.

Opgave. *Probeer hetgeen in de 'Verklaring' gesteld wordt theoretisch te onderbouwen. NB. Het begrip 'foutschatten' wordt in Deel III uitvoeriger behandeld.*

Vraag 2D. We hebben gezien dat de fout afhangt van afgeleiden van de integrand. Wat gebeurt er met de gerepeteerde trapeziumbenadering als de integrand een 'singulariteit' vertoont in de eerste afgeleide?

Experiment. [35] Wat gebeurt er met de factor $R(h/2)/R(h)$?

Verklaring. De afgeleide van de integrand is onbegrensd in $x = 0$.

Vraag 2E. Hoe kunnen we dit nare gedrag bij 0 omzeilen?

Experiment. Pas de substitutie $u = \sqrt{x}$ toe op (1.8) en gebruik de gerepeteerde trapeziumregel op dit 'nieuwe' probleem. Gaat het nu wel goed? D.w.z. vind je nu wel de verwachte factor in $R(h/2)/R(h)$ terug?

Verklaring. Door de transformatie van de integrand raak je de singulariteit in de afgeleiden kwijt, waardoor er weer het $\mathcal{O}(h^2)$ gedrag optreedt.

Vraag 2F. Zou het voor bepaalde integranden juist ook eens bijzonder mee kunnen vallen?

Experiment. Doe een experiment met de gerepeteerde trapeziumregel voor toenemende k toegepast op (1.9). Wat zie je nu met de foutfactor gebeuren?

Verklaring. De integrand is periodiek op $[0, \pi]$ en dus. . . ?

Vraag 2G. Zou de benadering van een integraal slechter worden, indien we de integrand zouden benaderen met een stuksgewijs constante functie (oftewel: verdeel het interval in n stukjes en neem het middelpunt van elk deelinterval met bijbehorende functiewaarde → **gerepeteerde midpuntregel**)?

[34] Maak gebruik van de file `trapez.m`

[35] Pas `trapez.m` toe op (1.8) voor toenemende waarden van n.

Experiment. [36] Halveer h enkele malen. Maak weer een tabel met de fout. Hoe neemt de fout af bij halverende h?

Verklaring. Ondanks het feit dat de benadering 'simpeler' is geworden, zien we nog steeds $\mathcal{O}(h^2)$-gedrag in de fout!

Opgave. *Hoe ziet de midpuntbenadering van* (1.1) *er in een formule uit? Bepaal de fout* $R(h)$ *voor de gerepeteerde midpuntregel.*

Terugkomend op het oplossen van DVs: herschrijf DV (1.1) *uit Deel I weer naar een integraalvergelijking en pas de midpuntregel toe om de integraal te benaderen. Hoe ziet deze numerieke methode ('de midpunt-methode' voor DVs) en de bijbehorende fout er uit?*

Vraag 2H. N.a.v. het succes van de midpuntregel ('stuksgewijs constante' benadering van de integrand) ten opzichte van de trapeziumregel ('stuksgewijs lineaire' benadering van de integrand) is het interessant om te onderzoeken wat het effect is, als we op elk deelinterval de functie benaderen met een kwadratisch polynoom. Dit levert de '**Simpsonregel**' op.

Experiment. [37] Wat gebeurt er met de fout bij het halveren van de h?

Verklaring. De fout bij de gerepeteerde Simpson-regel is $\mathcal{O}(h^4)$.

Opgave. *Leidt de gerepeteerde Simpsonregel af voor* (1.1).

Vraag 2I. Hoe kunnen we de nauwkeurigheid verhogen zonder al teveel'f-evaluaties' uit te voeren? In het vorige hoofdstuk hebben we gezien dat het kiezen van 'slimme' (Chebyshev) punten op de x-as een aanzienlijke verbetering kan opleveren voor de numerieke benadering. Zouden we dit ook voor elkaar kunnen krijgen in het geval van numerieke integratie?

Experiment. [38] Vergelijk deze benadering met de trapeziumbenaderingen uit het eerste experiment. Hoeveel functiewaarden heb je nodig voor de trapeziumregel voor ongeveer dezelfde nauwkeurigheid?

Verklaring. Zie verderop in de cursus.

Opgave. *Probeer d.m.v. het bestuderen van de Matlab-file deze kwadratuurformule op papier te krijgen.*

3.3 Toepassingen

In het onderstaande proberen we de vergaarde inzichten m.b.t. numerieke integratie toe te passen op enkele "praktijkproblemen".

Opdracht 3A. Bepaal met 'grote' nauwkeurigheid zowel $J_0(x)$ voor $x = 1, 2, 3$ (zie (1.2)) als de integraal uit (1.6). Controleer in dit laatste geval je berekening door Matlab zelf die waarde uit te laten rekenen m.b.v. een 'standaard-function'.

Opdracht 3B. Bepaal de lengte van het race-circuit uit Voorbeeld1G met een kwadratuurregel. Welke komt hiervoor in aanmerking? [39]

Opdracht 3C. Gebruik de Simpsonregel met stapgrootte $h = 0.25$ om de gevraagde tijd te bepalen in (1.10). Doe dit ook met de trapeziumregel met dezelfde stapgrootte. Vergelijk deze benaderingen met de werkelijke waarde.

[36] Doe enkele experimenten met `midp.m` (evt. in Matlab: help midp) toegepast op (1.5).

[37] Doe enkele experimenten met `simp.m` (evt: help simp) toegepast op (1.5).

[38] Pas `gaussl.m` (evt. help gaussl) toe op (1.5) met verschillend aantal 'steunpunten' (speel met de parameter k).

[39] Lees hiervoor de file `circuit.m`. De tabelwaarden zijn gegeven in de file `race.dat`.

3.4 Samenvatting

3.4.1 Rechthoeksregel

De kwadratuurformule die samenhangt met Euler-Forward na het herschrijven van een DV naar een integraalvergelijking luidt:

$$\int_a^b f(x)\,\mathrm{d}x \approx f(a)(b-a) \tag{4.1}$$

en in gerepeteerde vorm

$$\int_a^b f(x)\,\mathrm{d}x = h\sum_{i=0}^{n-1} f(x_i) + \frac{b-a}{2}hf'(\xi). \tag{4.2}$$

Deze kwadratuurregel is slechts exact voor polynomen van graad 0.

3.4.2 Trapeziumregel

Laat a en b twee punten in $\mathbb{R}$ zijn, $a<b$, en f een continue functie op $[a,b]$. Laat p het lineaire interpolatiepolynoom van f op a en b zijn:

$$p(x) = \frac{b-x}{b-a}f(a) + \frac{x-a}{b-a}f(b).$$

We gebruiken nu $\int_a^b p(x)\,\mathrm{d}x$ als benadering voor $\int_a^b f(x)\,\mathrm{d}x$:

$$\int_a^b p(x)\,\mathrm{d}x = \frac{b-a}{2}(f(a)+f(b)). \tag{4.3}$$

De benadering $\frac{b-a}{2}(f(a)+f(b))$ voor $\int_a^b f(x)\,\mathrm{d}x$ heet de **trapeziumregel**: het oppervlak onder de grafiek van f tussen a en b wordt benaderd met het oppervlak van het trapezium met als hoekpunten $(a,0)$, $(b,0)$, $(b,\ f(b))$ en $(a,\ f(a))$. De trapeziumregel is exact voor polynomen van graad ≤ 1.

Stelling 3.1 *Zij $f \in C^2[\,a,\,b\,]$. Dan is er een $\eta \in [\,a,\,b\,]$ zo dat*

$$\int_a^b f(x)\,\mathrm{d}x - \frac{b-a}{2}(f(a)+f(b)) = -\frac{(b-a)^3}{12}f''(\eta). \tag{4.4}$$

Met $nh = b-a$ vinden we zo:

$$\int_a^b f(x)\,\mathrm{d}x = T(h) - \frac{b-a}{12}h^2 f''(\xi), \tag{4.5}$$

waarin

$$T(h) = h[\frac{1}{2}f(x_0) + f(x_1) + \ldots + f(x_{n-1}) + \frac{1}{2}f(x_n)]. \tag{4.6}$$

De aldus verkregen kwadratuurformule $T(h)$ heet de n maal **gerepeteerde trapeziumregel**.

3.4.3 Midpuntregel

Door nulde orde interpolatie van f in het punt $m = \frac{1}{2}(a+b)$ op het segment $[a,\, b]$ vinden we de kwadratuurformule voor $\int_a^b f(x)\,\mathrm{d}x$:

$$(b-a)f(m). \tag{4.7}$$

Er geldt:

$$\int_a^b f(x)\,\mathrm{d}x - (b-a)f(m) = \frac{(b-a)^3}{24} f''(\xi), \tag{4.8}$$

voor zekere $\xi \in [a,\, b]$.

Voor de n keer **gerepeteerde** **midpunt** **regel** vindt men dan, met $h = (b-a)/n$, $m_i = a + (i + \frac{1}{2})h$:

$$\int_a^b f(x)\,\mathrm{d}x = h[f(m_0) + \ldots + f(m_{n-1})] + \frac{b-a}{24} h^2 f''(\xi). \tag{4.9}$$

De midpuntregel is exact voor polynomen van graad ≤ 1.

3.4.4 Simpsonregel

We kiezen nu als steunpunten $x_0 = a$, $x_1 = \frac{1}{2}(a+b)$, $x_2 = b$, en benaderen f met het tweedegraads interpolatiepolynoom op x_0, x_1, x_2. Dan vinden we als kwadratuurformule voor $\int_a^b f(x)\,\mathrm{d}x$:

$$\frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)], \tag{4.10}$$

waarbij $h = \frac{b-a}{2}$. Dit is de **regel** **van** **Simpson**.

Voor de n keer gerepeteerde Simpson regel vindt men nu met $h = (b-a)/2n$, $x_i = a + ih$

$$\begin{aligned} \int_a^b f(x)\,\mathrm{d}x \quad &= \tfrac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots + \\ &\quad +4f(x_{2n-1}) + f(x_{2n})] - \tfrac{b-a}{180} h^4 f^{(4)}(\xi). \end{aligned} \tag{4.11}$$

De regel van Simpson is exact voor polynomen van graad ≤ 3!

3.4.5 Algemenere kwadratuurformules

Stelling 3.2 *Voor de restterm R_n van de Newton-Cotes kwadratuurformules voor het benaderen van $\int_a^b f(x)\,\mathrm{d}x$ geldt: als n even is en $f \in C^{n+2}[a,\, b]$, dan*

$$R_n = C_n h^{n+3} f^{(n+2)}(\xi), \tag{4.12}$$

$$C_n = \frac{1}{(n+2)!} \int_0^n t^2(t-1)(t-2)\ldots(t-n)\,\mathrm{d}t, \quad h = \frac{b-a}{n}; \tag{4.13}$$

NB. Wanneer men een kwadratuurformule heeft voor een zekere integratie interval dan kan men er daaruit een afleiden voor een ander integratie interval:

Stelling 3.3 *Zij* $\sum_0^n w_i f(x_i)$ *een kwadratuurformule voor* $\int_a^b f(x)\,\mathrm{d}x$*, en laat gelden*

$$\int_a^b f(x)\,\mathrm{d}x = \sum_0^n w_i f(x_i) + \alpha f^{(k)}(\xi), \quad \xi \in [\,a,\, b\,] \tag{4.14}$$

voor alle k *maal continu differentieerbare functies* f*. Dan geldt*

$$\int_c^d g(y)\,\mathrm{d}y = \frac{d-c}{b-a}\sum_0^n w_i g(y_i) + \alpha \left(\frac{d-c}{b-a}\right)^{k+1} g^{(k)}(\xi), \quad \xi \in [\,c,\, d\,] \tag{4.15}$$

met $y_i = \frac{d-c}{b-a}x_i + \frac{bc-ad}{b-a}$*, voor alle* k *maal continu differentieerbare functies* g*.*

4 Nulpuntsbepaling

SLEUTELBEGRIPPEN: *Halveringsmethode, Successieve substitutie, Newton-Raphson, Koorden-Newton*

4.1 Inleiding

Het vinden van één of meer wortels van een vergelijking (of nulpunten van een functie)

$$f(x) = 0 \tag{1.1}$$

is een van de meest voorkomende problemen in de toegepaste wiskunde en in vele andere toepassingsgebieden. In de meeste gevallen zijn expliciete oplossingen niet voorhanden en moeten we tevreden zijn met de mogelijkheid om een benadering voor de wortel te vinden met een zekere van tevoren gespecificeerde nauwkeurigheid. De numerieke oplossingen worden vaak verkregen met behulp van iteratieve methoden. Met een iteratieve methode wordt uitgaande van een geschatte startwaarde, die natuurlijk bij voorkeur zo dicht mogelijk bij het gezochte nulpunt 'moet' liggen, een rij getallen gegenereerd, met uiteraard de bedoeling dat deze rij getallen convergeert naar het gevraagde nulpunt. Iedere volgende iteratiewaarde wordt op basis van zijn voorganger(s) volgens een bepaalde iteratieve formule berekend. In dit hoofdstuk zullen we voornamelijk de situatie beschouwen van één of twee voorgangers, dus $x_{i+1} = \phi(x_i)$, of $= \phi(x_{i-1}, x_i)$. Als deze rij convergeert, dan bereiken we nooit de exacte waarde, maar kunnen we deze onder 'gunstige' omstandigheden wel zo dicht mogelijk benaderen. Bij het gebruik van iteratieve methoden voor nulpunten moet je je direkt de volgende zaken gaan afvragen. Bijvoorbeeld:

Vraag 1A. wat is, numeriek gezien, een nulpunt eigenlijk? Probeer n.a.v. deze vraag eens de functie $f(x) = (1 - x)^6$ te plotten met Matlab. [40]

Of de vraag: hoeveel nulpunten zijn er, en naar welk van die nulpunten convergeert de iteratieve methode? Convergeert de methode wel en, zo ja, hoe snel? Hoe moet er gestart worden met het proces (startwaarde) en wanneer moet je stoppen (stopcriterium)? En, uiteindelijk, natuurlijk de belangrijkste vraag: hoe nauwkeurig is de berekende oplossing?
Al deze vragen zien we straks terug bij de experimenten.
Terugverwijzend naar Deel I wordt het nut van numerieke methoden voor vergelijkingen als (1.1) op een andere manier duidelijk, indien we bijvoorbeeld Euler-Backward willen toepassen op een niet-lineaire DV.

Opgave. *Schrijf nog eens Euler-Backward uit voor de niet-lineaire DV* (1.3) *uit Deel I en zie in dat we een numerieke methode nodig hebben voor het oplossen van een niet-lineaire vergelijking om überhaupt van t_n naar t_{n+1} te kunnen komen.*

Nu volgen eerst enkele voorbeelden van niet-lineaire vergelijkingen en 'praktijkproblemen' die we tegen zullen komen.

Voorbeeld 1B.

$$f(x) = x^3 + 3x - 4 = 0. \tag{1.2}$$

[40] gebruik de file `nulpvb.m`.

Deze vergelijking heeft de unieke wortel $x = 1$. Dit is een simpel testprobleem voor numerieke methoden, waarbij we het gedrag van de fout tijdens het iteratieproces kunnen bijhouden.

Voorbeeld 1C.

$$f(x) = x^3 - 3x + 2 = 0. \tag{1.3}$$

Deze vergelijking lijkt op de voorgaande, maar heeft twee oplossingen $x_1 = -2$ en $x_2 = 1$, waarbij x_2 een zogenaamde 'dubbele' wortel is. Het is op dit moment nog onduidelijk naar welk van de twee nulpunten een iteratief proces convergeert en welke invloed het speciale karakter van x_2 heeft?

Voorbeeld 1D.

$$f(x) = x + \exp(-Bx^2)\cos(x) = 0. \tag{1.4}$$

waarbij $B > 0$ een parameter is. Deze vergelijking heeft een unieke oplossing, waarvan de exacte waarde onbekend is. Het ophogen van de parameter B verhoogt de 'moeilijkheidsgraad' van het probleem navenant.

Voorbeeld 1E. Een interessant 'praktijk'probleem voor nulpuntsbepaling treedt op bij het berekenen van annuiteiten: een bedrag P_1 aan Euro's wordt op een rekening gezet aan het begin van de jaren $1, 2, \ldots, N_1$. Het wordt jaarlijks vergroot met een factor r (bijv. $r = 0.05$ betekent een rentevoet van 5%). Aan het begin van de jaren $N_1 + 1, \ldots, N_1 + N_2$ wordt een bedrag van P_2 Euro opgenomen van de rekening. Na de laatste transactie staat er precies 0 Euro op de rekening. De relatie die dit proces beschrijft wordt gegeven door de volgende vergelijking:

$$P_1[(1+r)^{N_1} - 1] = P_2[1 - (1+r)^{-N_2}]. \tag{1.5}$$

Als gegeven is dat $N_1 = 30$ jaar, $N_2 = 20$ jaar, $P_1 = 2000$ Euro en $P_2 = 8000$ Euro, wat is de rente r dan? Om dit te berekenen moeten we een niet-lineaire vergelijking oplossen.

Voorbeeld 1F. Beschouw het probleem van het richten van een kanon om een doel te raken op afstand d. Het kanon geeft een beginsnelheid V_0 mee aan de kanonskogel onder een hoek ϑ. Dit wordt beschreven door

$$f(\vartheta) = \frac{2V_0^2\cos(\vartheta)\sin(\vartheta)}{g} - d = 0. \tag{1.6}$$

Gevraagd is de hoek ϑ te bepalen bij gegeven V_0 en d ($g = 9.8\,\mathrm{m/s^2}$).

Voorbeeld 1G. Een pijp ter lengte L heeft een dwarsdoorsnede in de vorm van een halve cirkel met straal r. Als deze gevuld is met water tot een afstand h vanaf de bovenkant gerekend, geldt voor het volume V van het water:

$$V = L(0.5\pi r^2 - r^2\arcsin(h/r) - h\sqrt{r^2 - h^2}). \tag{1.7}$$

Stel dat $L = 10$ meter, $r = 1$ meter, and $V = 12.4$ kubieke meter. Gevraagd is: bepaal de diepte van het water in de pijp tot op een nauwkeurigheid van 0.01 meter.

Voorbeeld 1H. Een object valt verticaal door de lucht naar beneden en ondervindt zowel wrijving als zwaartekracht. Stel dat men het object met massa m laat vallen van een hoogte S_0 en dat de hoogte van het object na t seconden wordt gegeven door de volgende formule:

$$S(t) = S_0 + \frac{mg}{k}t - \frac{m^2g}{k^2}(1 - e^{-kt/m}), \tag{1.8}$$

waarbij g de versnelling t.g.v. de zwaartekracht is en k coefficient van de luchtweerstand weergeeft in kg sec/m.
Stel dat $S_0 = 300$ meter, $m = 0.25$ kg en $k = 0.1$ kg sec/m.
De vraag is: bepaal met een nauwkeurigheid van 0.01 sec, de tijdsduur die het object nodig heeft om de grond te raken.

Voorbeeld 1I. De groeifactor in grote populaties (vergelijk met Voorbeeld5C!) kan worden gemodelleerd (over niet al te lange tijdperioden) door aan te nemen dat de populatie continu in de tijd groeit met de factor die evenredig is met het aantal individuen die er op dat moment zijn. Als we met $N(t)$ het aantal individuen op tijdstip t aangeven en met λ de constante geboorte factor van de populatie, dan voldoet de populatie aan de volgende DV

$$\frac{dN(t)}{dt} = \lambda N(t) + v, \tag{1.9}$$

waarbij v de constante immigratie factor beschrijft.
De oplossing van deze DV is

$$N(t) = N_0 e^{\lambda t} + \frac{v}{\lambda}(e^{\lambda t} - 1). \tag{1.10}$$

Stel nu dat een zekere populatie aanvankelijk 1 miljoen individuen telt, dat er, bijv., 435 000 individuen immigreren in het eerste jaar, en dat er 1 564 000 individuen aanwezig zijn aan het einde van het eerste jaar.
De vraag is: wat is de geboortefactor van deze populatie? Hiervoor moet eerst een niet-lineaire vergelijking voor λ uitgewerkt worden!

4.2 Matlab

Vraag 2A. Hoe vindt Matlab zelf nulpunten van een functie en zijn deze uitkomsten betrouwbaar?

Experiment. Probeer beide nulpunten van vergelijking (1.3) [41] te vinden. Speel met verschillende startwaarden (start ook eens 'zeer dicht' bij de wortel(s)) .

Verklaring. Om één of andere reden kan Matlab het nulpunt x_2 niet vinden! Op dit moment is het onduidelijk waarom. We zullen dus numerieke methoden moeten gaan onderzoeken om in te kunnen zien welke criteria van belang zijn.

4.3 Halveringsmethode

Als een expliciete uitdrukking voor een nulpunt (noem deze α) niet bestaat, zou je kunnen zoeken naar een interval, waar de functie in beide eindpunten een tegengesteld teken heeft!

Idee. We kunnen in dit geval gebruik maken van de tussenwaardestelling. Door twee x-waarden x_0 en x_1 te kiezen waarvoor $f(x_0)f(x_1) < 0$, weten we dan dat er in ieder

[41] De functie wordt gedefinieerd in `f.m`; gebruik de Matlab-function `fzero.m` (evt. `help fzero`)

geval één punt $\alpha \in [x_0, x_1]$ moet zijn waarvoor $f(\alpha) = 0$! Op basis van deze gedachte zouden we een iteratieve methode kunnen uitwerken die ons een nulpunt van f oplevert.

Vraag 3A. Is het mogelijk met dit idee een algoritme te verzinnen dat convergeert naar een α met $f(\alpha) = 0$?

Experiment. [42]

Verklaring. Deze techniek heet de **halveringsmethode** (of **bisectie**) en convergeert naar de wortel van (1.2).

Opgave. *Beschrijf het algoritme in een zogenaamd 'stroomdiagram'.*

Vraag 3B. Hoe 'snel' convergeert de halveringsmethode?

Experiment. Herhaal het vorige experiment, maar let nu op het aantal iteraties dat nodig is bij een voorgeschreven 'nauwkeurigheid' (of 'tolerantie').

Verklaring. Bij iedere iteratieslag kom je, door het halveren van het interval, hooguit 2 keer zo dicht bij de exacte oplossing.

Opgave. *Controleer dat*

$$|x_i - \alpha| \leq \frac{1}{2}|x_{i-1} - \alpha| \tag{3.1}$$

en dus

$$|x_i - \alpha| \leq \frac{|x_1 - x_0|}{2^i}, \quad \textit{voor} \quad n \geq 1. \tag{3.2}$$

D.w.z. de fout wordt per iteratieslag ongeveer ('maximaal') gehalveerd. Vanwege deze eigenschap wordt de halveringsmethode een lineair convergent proces genoemd.

Vraag 3C. Aangezien vergelijking (1.3) bijzonder veel op (1.2) lijkt qua moeilijkheidsgraad, zou je ongeveer hetzelfde gedrag van de halveringsmethode verwachten. We gaan proberen het nulpunt $\alpha = 1$ te vinden.

Experiment. [43] Probeer, indien nodig, meerdere startwaarden (kies als default 'precisie' 0.001 om de verschillende methoden later goed met elkaar te kunnen vergelijken).

Verklaring. De halveringsmethode kan het 'dubbele' nulpunt $\alpha = 1$ niet detecteren, en convergeert dus, of helemaal niet, of naar het andere nulpunt $\alpha = -2$.

We zien dus dat deze methode niet altijd het gewenste resultaat oplevert. Bovendien is de convergentie 'slechts' lineair. Verder is het niet duidelijk hoe we dit principe zouden moeten uitbreiden naar stelsels van niet-lineaire vergelijkingen. Redenen genoeg voor ons om wellicht een andere weg in te slaan...

4.4 Successieve substitutie

Idee. We kunnen vergelijking (1.1) herschrijven naar de vorm

$$x = \phi(x) \tag{4.1}$$

en vervolgens via het iteratieve proces (dit heet de methode van successieve substitutie)

$$x_{i+1} = \phi(x_i) \tag{4.2}$$

[42] Pas de file `halverng.m` toe op vergelijking (1.2). Kijk ook eens in de file en probeer er achter te komen hoe deze methode werkt.

[43] Gebruik `halverng.m`.

en een gekozen startwaarde x_0 een nulpunt van f proberen te vinden. Dit principe is gebaseerd op de zogenaamde 'vaste-punt-theorie' (vaste punten van ϕ, d.w.z. oplossingen van (4.1), komen dan overeen met wortels van (1.1)). Het moge vooraf al duidelijk zijn, dat niet iedere keuze van ϕ tot een goed resultaat hoeft te leiden, ook al zijn oplossingen van (1.1) per definitie vaste punten van $\phi(x)$.

Opgave. *Laat met een simpel voorbeeldje zien dat $f(x) = 0$ altijd te herschrijven is naar $x = \phi(x)$.*

Vraag 4A. Laten we eerst eens bekijken hoe een dergelijk proces werkt op het eerste testprobleem (1.2). Convergeert het, en zo ja, hoe snel, en naar welke wortel?

Opgave. *Check eerst dat de functie $\phi_1(x) = \frac{4}{x^2+3}$ een kandidaat is voor de functie ϕ uit (4.1).*

Experiment. [44] Doe enkele experimenten met verschillende startwaarden. 'Meet' ook de convergentiesnelheid; d.w.z. welke $p > 0$ voldoet (ongeveer) aan $|x_{i+1}-\alpha|/|x_i-\alpha|^p \approx$ constante? Hoe wordt de 'precisie' hier trouwens gemeten? Heb je andere (betere?) suggesties?

Verklaring. De functie ϕ_1 heeft blijkbaar zodanige eigenschappen dat er voor iedere startwaarde convergentie optreedt. Overigens is de convergentie, net als bij de halveringsmethode, lineair (d.w.z. $p = 1$).

Opgave. *Laat zien dat de fout voldoet aan de volgende formule:*

$$x_{i+1} - \alpha = \phi'(\xi_i)(x_i - \alpha) \approx \phi'(\alpha)(x_i - \alpha) \quad \textit{voor } i \textit{ 'ver genoeg'}$$

voor zekere $\xi_i \in [\alpha, x_i]$. We zien dus dat de afgeleide van ϕ, en i.h.b. de waarde ervan in α, een cruciale rol speelt in dit iteratieve proces. Check dat in de bovenstaande situatie de ϕ' in absolute waarde altijd kleiner is dan $8/9 < 1$. Dus?

Misschien hebben we geluk gehad met deze keuze van ϕ. We hadden net zo goed, zonder voorkennis, een andere functie kunnen kiezen!

Opgave. *Laat zien dat vaste punten van alle volgende functies ook nulpunten van f uit (1.2) zijn:*

$$\phi_2(x) = \frac{4 - x^3}{3},$$

$$\phi_3(x) = \frac{4 - 3x}{x^2},$$

$$\phi_4(x) = x^3 + 4x - 4,$$

$$\phi_5(x) = (4 - 3x)^{1/3},$$

$$\phi_6(x) = \frac{2x^3 + 2x + 4}{3x^2 + 3}.$$

Vraag 4B. Welke functie uit dit rijtje moeten we nemen om (snellere) convergentie te verkrijgen?

Experiment. Doe enkele experimenten met al deze functies om het nulpunt van (1.2) zo snel en nauwkeurig mogelijk te benaderen [45]. Noteer alle bijzonderheden die je

[44] Gebruik de file `xisphix.m` met bijbehorende file `phi.m` (hierin kan de functie $\phi(x)$ desgewenst aangepast worden).

[45] Pas hiertoe, per geval, de file `phi.m` aan.

tegenkomt (convergentie, divergentie, snelheid van convergentie, afhankelijkheid van startwaarde x_0, etcetera).

Verklaring. De afgeleide van ϕ is de overheersende factor in alle experimenten. Deze grootheid kan de methode 'verpesten', maar ook aanzienlijk 'verbeteren'!

Opgave.

a. *Bereken ϕ' en i.h.b. $\phi'(\alpha)$ voor alle gebruikte functies ϕ_j, $\quad j = 1, \ldots, 6$. Wat voor gevolgen heeft dit voor de 'convergentie-snelheid? Teken ook in een $(x, \phi(x))$-diagram het iteratie proces voor iedere ϕ_j.*

b. *Laat ook zien dat als $|\phi'(\alpha)| > 1$, er* nooit *convergentie naar α kan optreden.*

Uit het voorafgaande concluderen we dat er keuzes voor ϕ mogelijk zijn, waarvoor de convergentiesnelheid veel groter is dan misschien verwacht!

Vraag 4C. Is dit toeval of zit er een structuur in? Oftewel, kunnen we een dergelijke ϕ ook construeren voor het algemene geval (1.1)?

Opgave. *Ontwikkel een aantal termen van een Taylorreeks voor de functie $\phi(x)$ in x_i rond het nulpunt α. Laat vervolgens zien dat voor de keuze $\phi(x) = x - f(x)/f'(x)$ (welke ϕ_j uit het voorbeeld voldoet hieraan?) geldt: $\phi'(\alpha) = 0$ en dus:*

$$x_{i+1} - \alpha = \frac{\phi''(\xi_i)}{2}(x_i - \alpha)^2. \tag{4.3}$$

Geef een uitdrukking voor $\phi''(\xi_i)$ in termen van f. De fout voor de onderliggende iteratie-methode neemt dan, in principe (wanneer misschien niet bijvoorbeeld?), **kwadratisch** *af per iteratieslag! Een nadere analyse van deze speciale methode is dan ook gewenst.*

4.5 Newton-Raphson

De methode van **Newton**–**Raphson** (NR) voor (1.1) wordt gegeven door de iteratie (zie laatste opgave van het vorige hoofdstuk):

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i \leq 0, \qquad x_0 \text{ gegeven.} \tag{5.1}$$

Vraag 5A. Kunnen we NR ook op een andere manier (dus niet in termen van ϕ), bijvoorbeeld grafisch interpreteren? Het gebruik van de afgeleide van f suggereert een relatie met de raaklijn in verschillende x-waarden.

Experiment. [46]

Verklaring. In iedere iteratieslag maakt Newton-Raphson gebruik van de raaklijn aan de grafiek van f in de oude x-waarde om het volgende iteratiepunt te bereiken.

Opgave. *Laat zien dat, om van x_0 naar x_1 te komen, we de raaklijn in x_0 kunnen nemen, en dat vervolgens het nulpunt van deze lineaire functie gelijk is aan de x_1 uit NR. Door vervolgens verder te gaan met x_1 als nieuwe 'startwaarde' en het proces te herhalen krijgen we het iteratieve proces horend bij NR. Ga dit na.*

[46] Gebruik de file `newton.m` (voor een zelfgekozen f (bijv. $\sin(x)$; zie de files `f.m` en `facc.m`) om enkele iteratieslagen van NR te zien. Desgewenst kan m.b.v. `graffun.m` en `graffacc.m` de grafiek van $f(x)$, resp. $f'(x)$, getekend worden.

Vraag 5B. Zouden we, zonder het nulpunt α te kennen, een schatting kunnen geven van de onderweg gemaakte fout? We hebben immers gezien dat voor NR i.h.a. geldt:

$$\frac{x_{i+1}-\alpha}{(x_i-\alpha)^2} \approx \frac{f''(\alpha)}{2f'(\alpha)} \equiv C \qquad \text{voor } i \text{ groot genoeg}$$

Derhalve zouden we de grootheid C kunnen schatten met $\frac{f''(x_{i+1})}{2f'(x_{i+1})}$ en daarna

$$x_{i+1}-\alpha \quad (= \text{de fout in slag } i+1) \approx C(x_i - x_{i+1})^2,$$

waarbij x_i en x_{i+1} berekend zijn met NR.

Experiment. [47]

Verklaring. De boven gegeven schatting voor de fout in NR geldt inderdaad; tenminste voor deze vergelijking.

Vraag 5C. In het NR-proces komt de afgeleide van f in de noemer voor. Wat gebeurt er als $f'(\alpha) = 0$? Hebben we dan nog steeds convergentie, laat staan kwadratische convergentie?

Experiment. [48] Noteer per iteratieslag de grootheid $(x_{i+1}-\alpha)/(x_i-\alpha)^p$ voor $p = 1$ en $p = 2$.

Verklaring. Voor $f'(\alpha) = 0$ wordt NR een lineair convergent proces!

Opgave. *Laat zien dat voor het geval $f(\alpha) = f'(\alpha) = 0$ geldt:*

$$\lim_{i\to\infty} \frac{x_{i+1}-\alpha}{x_i-\alpha} = \frac{1}{2}. \tag{5.2}$$

Vraag 5D. Zou in speciale situaties het NR-proces i.p.v. een langzamere (lineair) juist een nog sneller (sneller dan kwadratisch) convergentie kunnen opleveren?

Experiment. [49] Is het proces nu lineair of kwadratisch convergent? Plot deze functie en probeer uit te vissen wat er rond $x = 1$ aan de hand is.

Verklaring. In $x = 1$ heeft deze speciale f een buigpunt, terwijl de afgeleide daar niet gelijk aan nul is. Deze eigenschap versnelt het NR-proces.

Opgave. *Laat zien dat geldt: als $f(\alpha) = f''(\alpha) = 0$ en $f'(\alpha) \neq 0$, dan convergeert Newton-Raphson* ***kubisch*** *($\Longrightarrow$ convergentie-orde is 3) naar het nulpunt α.*
(hint: werk o.a. $\phi''(\alpha)$ en $\phi'''(\alpha)$ uit voor $\phi(x) \equiv x - f(x)/f'(x)$.)

Vraag 5E. In de vorige experimenten hebben we de invloed van eigenschappen van de functie f op het convergentie-gedrag van NR onderzocht, wanneer we dicht genoeg bij het nulpunt α zitten. MAAR: hoe komen we dichtbij het nulpunt zonder kennis vooraf? Oftewel, welke invloed heeft de startwaarde x_0 op het iteratieproces?

Opgave. *Onderzoek de functie f uit* (1.4) *en ga na wat de invloed is van de parameter B op de 'vorm' van de grafiek van f.*

Experiment. [50]

---

[47] Doe enkele experimenten met `newton.m` toegepast op (1.2). Vergelijk de geschatte fout met de werkelijke fout(de functie $f''/(2f)$ staat in de functionfile `f2f.m`).

[48] Pas `newton.m` toe op vergelijking (1.3) en probeer het nulpunt $\alpha = 1$ te benaderen.

[49] Pas `newton.m` toe op $f(x) = \tan(x-1) = 0$ en onderzoek het gedrag van de fout in de buurt van $\alpha = 1$.

[50] Gebruik `newton.m` om het nulpunt van (1.4) te vinden voor verschillende startwaarden (dichtbij α, ver weg van α en probeer ook eens $x_0 = 0$). Doe dit voor enkele waarden van B (bijv. $B = 1, 5, 10, 25, 50$).

Verklaring. Numerieke methoden voor het berekenen van nulpunten zijn i.h.a. zeer gevoelig voor de keuze van de startwaarde.

Opgave. *Laat zien dat voor NR geldt: als* $f(\alpha) = 0$ *voor* $f \in \mathcal{C}^2$, $f(x) \neq 0$ *en* $f''(x)f(x) \geq 0 \quad \forall x \in [a, b]$, *dan convergeert NR monotoon naar* α voor iedere startwaarde $x_0 \in [\alpha, b]$.

Vraag 5F. Zou het mogelijk zijn om een methode te bedenken waarbij we de afgeleide f' niet nodig hebben en toch de voordelen van NR, zoals het meer dan lineair convergente karakter van het proces, gedeeltelijk kunnen behouden?

4.6 Koorden-Newton

Idee. Om de laatste vraag te beantwoorden zouden we bijvoorbeeld de afgeleide $f'(x_i)$ kunnen benaderen door een differentiequotiënt.

Opgave 6A. *Laat m.b.v. Taylorontwikkelingen zien dat*

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} + E. \tag{6.1}$$

Bepaal ook de foutterm E.

Vraag 6B. Indien we $f'(x_i)$ in NR vervangen door $(f(x_i) - f(x_{i-1}))/(x_i - x_{i-1})$ krijgen we de zogenaamde **koorden**–**Newton** methode (KN). Wat gebeurt er met het i.h.a. kwadratisch convergente karakter van de methode? Treedt er überhaupt wel convergentie op naar het nulpunt?

Experiment. [51]

Verklaring. De methode KN verliest het kwadratisch convergente karakter van NR door de extra benadering die we maken, maar is nog steeds 'sneller dan' lineair convergent!

Opgave. *Verklaar de naam Koorden-Newton met een plaatje.*
NB. de orde van convergentie van KN blijkt $\frac{1}{2}(1 + \sqrt{5})$ *te zijn. Voor een bewijs (via Fibonacci-getallen) verwijzen we naar de literatuur.*

4.7 Toepassing

Onderstaande 'praktijkproblemen' kunnen alleen met numerieke methoden worden opgelost.

Opdracht 7A. Bereken de oplossing van het beschreven model uit Voorbeeld1E m.b.v. Koorden-Newton of Newton-Raphson.

Opdracht 7B. Bepaal de diepte van het water in de pijp uit Voorbeeld1G. Kies zelf een 'geschikte' methode.

Opdracht 7C. Bereken voor een aantal zelfgekozen waarden van V_0 en d de benodigde hoek om d te bereiken (zie Voorbeeld1F). Houd hierbij rekening met het feit dat niet iedere combinatie een (unieke) oplossing heeft!

[51] Probeer met de file `koordnwt.m` toegepast op (1.2) de convergentie-orde p te schatten in $(x_{i+1} - \alpha)/(x_i - \alpha)^p$. Merk op dat we nu 2 startwaarden x_0 en x_1 nodig hebben!

Opdracht 7D. Bedenk zelf een functie ϕ voor de methode van successieve substitutie om het probleem uit Voorbeeld1H op te lossen (en los het probleem ook daadwerkelijk op).

Opdracht 7E. Gebruik Newton-Raphson voor het bepalen van de geboortefactor uit Voorbeeld1I met de aldaar gegeven parameters.

4.8 Samenvatting

4.8.1 Halveringsmethode

• De bisectiemethode is gebaseerd op de tussenwaardestelling:

Stelling 4.1 *Als $f(x)$ een continue functie is op het interval $[a, b]$ en $f(a) \neq f(b)$, dan geldt voor elk getal y tussen $f(a)$ en $f(b)$, dat er een getal x bestaat met $x \in [a, b]$ waarvoor $f(x) = y$. (lees in ons geval $y = 0$)*

Zoek eerst een interval $[a, b]$ waarvoor $f(a)$ en $f(b)$ tegengesteld teken hebben. Neem het middelpunt m van $[a, b]$ en controleer of $f(m)$ hetzelfde teken heeft als $f(a)$ of $f(b)$. In het eerste geval ga je verder met het interval $[m, b]$, in het tweede met $[a, m]$. Hierna neem je het middelpunt van het volgende (kleinere) interval en je herhaalt bovenstaande procedure. Dit levert je het bisectie-algoritme op voor het bepalen van nulpunten.
Het voordeel van deze methode is dat, indien $f(x_0)f(x_1) < 0$, er altijd convergentie naar een nulpunt van f plaatsvindt. De vraag is alleen, in het geval van meerdere nulpunten, welk nulpunt uiteindelijk berekend wordt...
Verder hoeft 'de fout' niet monotoon naar nul te gaan. Oftewel, zonder stopcriterium, zou bijv. iteratieslag 12 een betere benadering kunnen geven dan iteratieslag 14! De methode convergeert 'lineair' naar 'het' nulpunt. Het is helaas niet mogelijk om de halveringsmethode te beschrijven voor stelsels van niet-lineaire vergelijkingen.

4.8.2 Successieve substitutie

Een andere techniek is gebaseerd op het volgende principe:
Herschrijf $f(x) = 0$ naar $x = \phi(x)$ (dit is altijd mogelijk; vaak zelfs meerdere mogelijkheden!). Oplossingen van $x = \phi(x)$ worden 'vaste punten' van ϕ genoemd. Een iteratieve methode voor het bepalen van een vast punt van ϕ, en dus een nulpunt van f, wordt gegeven door

$$x_{i+1} = \phi(x_i), \quad i = 0, 1, \ldots, \qquad x_0 \text{ gekozen.} \tag{8.1}$$

Het is mogelijk om enkele nuttige Lemma's en Stellingen te formuleren:

Lemma 4.2 *Als $\phi(x)$ continu is op $[a, b]$ en $\phi(x) \in [a, b]$ voor $x \in [a, b]$, dan heeft de vergelijking $x = \phi(x)$ minstens een oplossing $\alpha \in [a, b]$.*
Bewijs. m.b.v. de tussenwaardestelling. □

Lemma 4.3 *Als $\phi(x)$ continu is op $[a, b]$ en $\phi(x) \in [a, b]$ voor $x \in [a, b]$, en er bestaat een constante k*

$$0 < k < 1 \quad met \quad |\phi(x) - \phi(y)| \leq k\,|x - y| \quad \forall x, y \in [a, b],$$

dan heeft de vergelijking $x = \phi(x)$ precies een oplossing $\alpha \in [a, b]$ en de rij $x_{i+1} = \phi(x_i)$ $(i \geq 0)$ convergeert voor iedere startwaarde $x_0 \in [a, b]$ naar α.

Bewijs. Stel ook β oplossing van $x = \phi(x)$. .... Er volgt een tegenspraak door gebruik te maken van $|\phi(x) - \phi(y)| \leq k\,|x - y| \quad \forall x, y \in [a, b]$. Convergentie volgt door diezelfde relatie meerdere malen toe te passen (volledige inductie). $\square$

Lemma 4.4 *Als $\phi(x) \in C^1[a, b]$, dan geldt met $K \equiv max_{x\in[a,b]}|\phi'(x)|$ en gebruikmakend van de middelwaardestelling: $|\phi(x) - \phi(y)| \leq K|x - y| \quad \forall x, y \in [a, b]$. Indien nu deze K kleiner dan 1 is, dan kan dit laatste Lemma toegepast worden.*

Een stelling over 'globale' convergentie:

Stelling 4.5 *Als $\phi(x) \in C^1[a, b]$, $\phi(x) \in [a, b]$ voor $x \in [a, b]$ en*

$$K \equiv \max_{x\in[a,b]} |\phi'(x)| < 1,$$

dan geldt:

i) $x = \phi(x)$ *heeft een unieke oplossing* $\alpha \in [a, b]$

ii) $\forall x_0 \in [a, b]$ *convergeert het proces* $x_{i+1} = \phi(x_i)$ *naar* α, *d.w.z.* $\lim_{i\to\infty} x_i = \alpha$.

iii) *een afschatting voor de fout:* $|x_{i+1} - \alpha| \leq K^i |x_i - \alpha|$

iv) $\displaystyle\lim_{i\to\infty} \frac{x_{i+1} - \alpha}{x_i - \alpha} = \phi'(\alpha)$.

NB. Als $\phi'(\alpha) \neq 0$ en $K < 1$, dan noemen we het proces **lineair convergent** en $\phi'(\alpha)$ is de zogenaamde **asymptotische convergentiefactor**.

Een stelling over 'lokale' convergentie:

Stelling 4.6 *Als $\phi'(\alpha)$ bestaat en $|\phi'(\alpha)| < 1$, dan is er een $\delta > 0$ zo dat voor elke startwaarde x_0 met $|x_0 - \alpha| < \delta$ (x_0 "ligt voldoende dicht" bij α) geldt:*

$$\lim_{i\to\infty} x_i = \alpha \quad en \quad \lim_{i\to\infty} \frac{x_{i+1} - \alpha}{x_i - \alpha} = \phi'(\alpha).$$

Bewijs. Opgave of zie literatuur. $\square$

'Divergentie'. *Als $|\phi'(\alpha)| > 1$, dan zal het proces $x_{i+1} = \phi(x_i)$ nooit naar de oplossing α kunnen convergeren.*

Een stelling over 'snellere' convergentie (van orde p):

Stelling 4.7 *Als 'x_0 voldoende dicht bij α' gekozen wordt, $\phi(x)$ p-keer continu differentieerbaar is $\forall x$ rond α (voor zekere $p \geq 2$), en er geldt $\phi'(\alpha) = \phi''(\alpha) = \ldots = \phi^{(p-1)}(\alpha) = 0$ met $\phi^{(p)}(\alpha) \neq 0$, dan heeft het iteratieproces $x_{i+1} = \phi(x_i) \quad (i \geq 0)$ orde van convergentie p en er geldt:*

$$\lim_{i\to\infty} \frac{x_{i+1} - \alpha}{(x_i - \alpha)^p} = \frac{\phi^{(p)}(\alpha)}{p!}.$$

Voor $p = 1$, $p = 2$, $p = 3$ heet het proces respectievelijk **lineair**, **kwadratisch** en **kubisch** convergent.

De analyse van de methode der successieve substitutie hangt sterk samen met de theorie van 'vaste-punten'. In een vector-vorm kan deze methode ook voor stelsels van niet-lineaire vergelijkingen geformuleerd worden. Bij een 'slimme' keuze van de afbeelding ϕ kan snellere convergentie (zie bovengenoemde stelling) bewerkstelligd worden. *Echter*, het is niet altijd eenvoudig om de 'juiste' ϕ te bepalen.

4.8.3 Newton-Raphson

Voor de speciale keuze $\phi(x) = x - \frac{f(x)}{f'(x)}$ in de methode de successieve substitutie krijgt men de methode van **Newton–Raphson**.
Als $f'(\alpha) \neq 0$ en $f''(\alpha) \neq 0$, dan heeft dit proces convergentie-orde 2 met asymptotische convergentiefactor $\frac{1}{2}\frac{f''(\alpha)}{f'(\alpha)}$.
Een stelling over 'globale' convergentie van Newton-Raphson:

Stelling 4.8 *Als* $f \in C^2$*,* $f(\alpha) = 0$*,* $f(x) \neq 0$ *en* $f''(x)f(x) \geq 0 \quad \forall x \in (\alpha, b]$*, dan convergeert Newton-Raphson* monotoon *naar* α *voor iedere startwaarde* $x_0 \in (\alpha, b]$*.*

NB. een dergelijk resultaat geldt natuurlijk ook voor het interval $[a, \alpha)$.
Deze methode (en varianten ervan) is in de praktijk de meest populaire methode voor het oplossen van (stelsels) niet-lineaire vergelijkingen. Een nadeel is echter het berekenen van f' en de Jacobiaan in het geval van stelsels.

4.8.4 Koorden-Newton

Deze methode is gebaseerd op Newton-Raphson en benadert de daarin voorkomende afgeleide f' alsvolgt:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}. \tag{8.2}$$

Door deze extra benadering verliest Koorden-Newton het kwadratisch-convergente karakter van Newton-Raphson. De orde van convergentie wordt (afleiding via Fibonacci-getallen, zie literatuur): $p = \frac{1}{2}(1 + \sqrt{5})$, dus nog steeds 'sneller' dan lineair convergent!
Voor deze methode hebben we geen f' nodig tijdens de iteraties, maar wel twee startwaarden. Het is onduidelijk hoe Koorden-Newton moet worden toegepast op stelsels van niet-lineaire vergelijkingen (dit kan slechts voor zeer speciale gevallen). Bovenstaande stelling over globale convergentie voor NR geldt ook voor Koorden-Newton.

4.8.5 Slotopmerkingen

Voor alle behandelde methoden geldt: het kiezen van de juiste startwaarde(n) is essentieel voor de convergentie en nauwkeurigheid van het benaderingsproces. Verder speelt het stopcriterium in alle gevallen een belangrijke rol; bij een verkeerde interpretatie van het numerieke resultaat kunnen onjuiste conclusies getrokken worden over de nauwkeurigheid van de oplossing.

Opgave 8A. *(terugkoppeling naar Euler-Backward)*
Beschrijf het toepassen van Newton-Raphson en die van de methode van successieve substitutie op de niet-lineaire DV (1.3) *uit Deel I. Bespreek de voor- en nadelen van beide methoden. Wanneer kunnen we beter Euler-Forward gebruiken, i.p.v. Euler-Backward gecombineerd met een nulpuntsmethode?*

Hoofdstuk III

Gestruktureerde fouten door G. Sleijpen

5 Differentiëren

Sleutelbegrippen: *Differenties, stabiliteit, maximale nauwkeurigheid.*

5.1 Waarom numeriek differentiëren?

Van een funktie f willen we, in een aantal punten, de waarde van de eerste afgeleide (en van hogere afgeleiden) kennen. Als f gedefinieerd is door middel van een analytische uitdrukking (een formule) kan je voor de afgeleide(n) uitdrukkingen afleiden waarmee je de waarde in de gespecificeerde punten kunt uitrekenen. Het komt echter nogal eens voor dat van f alleen maar de waarde in een aantal punten bekend is (als f getabelleerd is) of dat f gedefinieerd is als de oplossing van een differentiaalvergelijking (zie Hoofdstuk I). Men is dan aangewezen op numerieke methoden. Maar ook als f gedefinieerd is via een analytische uitdrukking kan het efficiënter zijn om numerieke methoden te gebruiken. Immers, de analytische uitdrukkingen hebben bij differentieren, zeker in geval van hogere afgeleiden, de neiging om gekompliceerd te worden. Bovendien is in de praktijk het aantal analytische uitdrukkingen van afgeleiden dat bepaald moet worden nogal eens enorm groot en is het niet aantrekkelijk om al die uitdrukkingen “met de hand” te bepalen. Het is veel werk en men loopt het risico fouten te maken.[52] Zo moet je bijvoorbeeld voor de Jacobiaan van een $\mathbb{R}^d$-waardige funktie van d-veranderlijken d^2 partiële afgeleiden bepalen. De Jacobiaan heeft men nodig in het Newton-Raphson proces (zie §4.5 in Hoofdstuk II) voor $\mathbb{R}^d$-waardige funkties van d veranderlijken.

In deze sektie bespreken we een voor de hand liggende methode om afgeleiden numeriek te berekenen. We bekijken een reeel-waardige funktie f op een interval $[a, b]$ en een rij punten $x_0, \ldots, x_n$ in $[a, b]$. We besteden speciale aandacht aan de rij $\ldots, c - h, c, c + h, \ldots$, waarbij c in (a, b) en $h > 0$.

De afgeleide van de funktie f in het punt c is gedefinieerd door

$$f'(c) = \lim_{h \to 0} \frac{f(c + h) - f(c)}{h}. \tag{1.1}$$

Vraag 1A. We zouden $f'(c)$ kunnen benaderen door $\frac{f(c+h)-f(c)}{h}$. We mogen verwachten dat de benadering beter is voor kleinere h. Hoe pakt dat uit in de praktijk? Als we

[52] Programma's als Mathematica en Maple zijn ontwikkeld om snel en foutloos formules voor afgeleiden en voor primitieven van funkties te bepalen.

$f'(c)$ nauwkeurig kunnen benaderen kunnen we dezelfde procedure toepassen om $f''(c)$ te benaderen, etc.. Werkt dit? Gaat het weer beter voor kleinere h? Hangt het af van de gladheid van f?

Experiment. Bekijk de funktie $f(x) = \exp(x)$. Van deze funktie kunnen we de afgeleiden gemakkelijk berekenen en kunnen we controleren hoe nauwkeurig de numerieke differentiatie procedure is. Bovendien is deze funktie zeer glad (f is analytisch) en komen eventuele ongewenste resultaten niet van een gebrek aan gladheid van f. Bereken voor verschillende grootten van h de eerste zes afgeleiden van f, plot de resultaten en de fout erin.[53]

Conclusie. De voor de hand liggende aanpak om te differentiëren lijkt, zekere voor kleine h en hogere afgeleiden onnauwkeurig.

Hoe kunnen we nauwkeurig numeriek differentiëren? Hoe kunnen we inzicht krijgen in de fout die hierbij gemaakt wordt?

5.2 Eerste afgeleide

SLEUTELBEGRIPPEN: *Differentie quotiënt, voorwaartse –, terugwaartse – en centrale differentie.*

Als f gedefinieerd is in de punten $x_0, x_1, \ldots, x_n$ dan is het, met het oog op de definitie in (1.1), verleidelijk om de afgeleide in x_i te benaderen door

$$f[x_i, x_{i+1}] \equiv \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}. \tag{2.1}$$

De waarde $f[x_i, x_{i+1}]$ is de **eerste differentie** of **differentie quotiënt** van f in x_i en x_{i+1}. Als x_{i+1} dicht bij x_i ligt dan verwachten we dat het differentie quotiënt de afgeleide van f in x_i goed benadert.

Vraag 2A. Hoe goed benadert $f[x_i, x_{i+1}]$ de waarde $f'(x_i)$? Is de benadering niet beter in bijvoorbeeld het gemiddelde $\mu_i \equiv \frac{1}{2}(x_i + x_{i+1})$ van x_i en x_{i+1}? Wat betekent 'beter'?
Is $f[x_i, x_{i+1}] = f'(\xi)$ voor zekere ξ tussen x_i en x_{i+1}?

Experiment. Bekijk, voor $h > 0$, de punten x_i van de vorm $x_i = a + ih$. Plot in hetzelfde figuur de grafiek van f' en $f[x_i, x_i+h]$ als funktie van x_i. Is $f[x_i, x_{i+1}]$ een f'-waarde? Zo ja, in welk punt? Hoe hangt $f[x_i, x_{i+1}] - f'(\nu_i)$ af van h, voor bijvoorbeeld $\nu_i = x_i$ of $\nu_i = \mu_i$? Welke ν_i is het geschikst?[54]

We zeggen dat we 'exactheid' hebben als $f[x_i, x_{i+1}] = f'(\mu_i)$. Neem eens voor f een polynoom van graad $\leq \ell$ $(= 0, 1, 2, \ldots)$. Wat is de maximale ℓ waarvoor je exactheid hebt? Hangt dat nog af van μ_i?

Verklaring. Schets de grafiek van een funktie f. Teken in deze grafiek de raaklijn in $(x_i, f(x_i))$ aan f en de koorde door $(x_i, f(x_i))$ en $(x_{i+1}, f(x_{i+1}))$. Zie in dat de koorde parallel loopt aan de raaklijn door $(\xi, f(\xi))$ aan f voor een zekere ξ tussen x_i en x_{i+1}.

Opgave. *Bewijs dat voor voldoende gladde funkties f en voor h (voldoende klein) geldt*

$$\frac{f(c+h) - f(c)}{h} = f'(c) + \tfrac{1}{2}hf''(\xi) \quad \text{voor zekere} \quad \xi \in [c, c+h] \tag{2.2}$$

[53] Gebruik het programma `vraag1a.m`.
[54] Je kan `vraag2a.m` gebruiken.

en

$$\frac{f(c+h)-f(c-h)}{2h} = f'(c) + \tfrac{1}{6}h^2 f'''(\xi) \quad \textit{voor zekere} \quad \xi \in [c-h, c+h] \tag{2.3}$$

Hint: Taylor de voorkomende termen rond c.

Conclusie.
$f[x_i, x_{i+1}] = f'(\xi)$ voor zekere $\xi \in (x_i, x_{i+1})$.
$f[x_i, x_{i+1}] = f'(\frac{1}{2}(x_i + x_{i+1})) + \mathcal{O}(h^2)$ voor $h \to 0$ en $x_i = c - h$, $x_{i+1} = c + h$.
$f[x_i, x_{i+1}] = f'(\frac{1}{2}(x_i + x_{i+1}))$ als f een polynoom is van graad ≤ 2.

De **symmetrische** differentie

$$\frac{f(c+h)-f(c-h)}{2h} \tag{2.4}$$

vormt, voor $h > 0$, in het algemeen een nauwkeurigere benadering voor $f'(c)$ dan de **terugwaartse** differentie of **voorwaartse** differentie,

$$\frac{f(c)-f(c-h)}{h}, \quad \text{respektievelijk} \quad \frac{f(c+h)-f(c)}{h}. \tag{2.5}$$

Toch hebben deze onnauwkeurigere differenties ook toepassingen zoals we gezien hebben in Hoofdstuk I. Merk op dat of een differentie 'symmetrisch', 'terugwaarts' of 'voorwaarts' is afhangt van het punt waarin de differentie als benadering van f' gezien wordt.

5.3 Hogere afgeleiden

SLEUTELBEGRIPPEN: *Gedeelde en ongedeelde differenties.*

Als $f[x_i, x_{i+1}]$ een benadering is van f' in het punt $\mu_i \equiv \frac{1}{2}(x_i + x_{i+1})$ kunnen we dan, door differenties te vormen van de $f[x_j, x_{j+1}]$, nauwkeurige benaderingen vormen van hogere afgeleiden van f? In deze aanpak stapelen we wel fout op fout. Komt er desondanks iets bruikbaars uit?

Merk op dat
$\frac{1}{2}(x_{i+1} + x_{i+2}) - \frac{1}{2}(x_i + x_{i+1}) = \frac{1}{2}(x_{i+2} - x_i)$. We bekijken daarom de **tweede gedeelde differenties**

$$f[x_i, x_{i+1}, x_{i+2}] \equiv 2\,\frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}. \tag{3.1}$$

Vraag 3A. Kan je $f[x_i, x_{i+1}, x_{i+2}]$ zien als benadering voor f'' in het gemiddelde

$$\tfrac{1}{2}[\tfrac{1}{2}(x_{i+1} + x_{i+2}) + \tfrac{1}{2}(x_i + x_{i+1})] = \tfrac{1}{4}(x_i + 2x_{i+1} + x_{i+2}) \tag{3.2}$$

van de μ_i en μ_{i+1}? Of is $f[x_i, x_{i+1}, x_{i+2}]$ niet een betere benadering voor f'' in het echte gemiddelde $\frac{1}{3}(x_i + x_{i+1} + x_{i+2})$? Wat betekent 'beter' hier?
Is $f[x_i, x_{i+1}, x_{i+2}] = f''(\xi)$ voor zekere ξ? Waar moeten we de ξ zoeken?

Experiment. Ga te werk als in het experiment van Vraag 2A. Kan je de vragen beantwoorden voor equidistant verdeelde steunpunten? Bekijk ook steunpunten x_i die sterk ongelijk verdeeld liggen.

Verklaring. Een grafische interpretatie is hier wat lastiger te geven, maar uitschrijven in Taylorreeksen rond, bv. x_i, leidt tot een korrekte verklaring.

Opgave. *Bewijs dat*

$$f[c-h,c,c+h] = \frac{f(c+h) - 2f(c) + f(c-h)}{h^2} \tag{3.3}$$

en

$$f[c-h,c,c+h] = f''(c) + \frac{1}{12}h^2 f^{(4)}(\xi) \quad \textit{voor zekere } \xi \textit{ tussen } c-h \textit{ en } c+h.$$

De differentie in (3.3) *is de* **symmetrische** *differentie voor* $f''(c)$.

In Vraag 3A hebben we gezien dat de tweede differentie $f[x_i, x_{i+1}, x_{i+2}]$ gezien kan worden als een benadering voor f'' in het gemiddelde $\frac{1}{3}(x_i+x_{i+1}+x_{i+2})$. Dit gemiddelde is onafhankelijk van de 'volgorde' van x_i, x_{i+1}, x_{i+2}, terwijl het gewogen gemiddelde (3.2), dat wellicht meer voor de hand ligt, dat wel is: $\frac{1}{4}(x_i + 2x_{i+1} + x_{i+2})$ is in het algemeen niet $\frac{1}{4}(x_{i+1}+2x_i+x_{i+2})$. De eerste differentie $f[x_i, x_{i+1}]$ hangt ook niet af van de volgorde (Ga na dat $f[x_i, x_{i+1}] = f[x_{i+1}, x_i]$). Hoe zit het met de $f[x_i, x_{i+1}, x_{i+2}]$: is $f[x_i, x_{i+1}, x_{i+2}] = f[x_{\pi(i)}, x_{\pi(i+1)}, x_{\pi(i+2)}]$ voor iedere permutatie π van $(i, i+1, i+2)$?

Vraag 3B. Hangt de waarde van $f[x_i, x_{i+1}, x_{i+2}]$ af van de volgorde van de betrokken x_j's?

Experiment. Het is voldoende om naar de situatie voor $i = 0$ te kijken. Waarom? Vergelijk de waarde $f[x_{\pi(0)}, x_{\pi(1)}, x_{\pi(2)}]$ voor verschillende permutaties π van $(0, 1, 2)$. Probeer ook verschillende distributies van de steunpunten x_0, x_1, x_2.[55]

Verklaring. Schrijf alles uit in termen van $f(x_0)$, $f(x_1)$ en $f(x_2)$.

Het idee om tweede differenties te gebruiken om tweede afgeleiden te benaderen lijkt te werken. Generalisatie ligt nu voor de hand.

We definiëren recursief de k-**de gedeelde**[56] **differenties** door

$$f[x_0, x_1, \ldots, x_k] \equiv k\,\frac{f[x_1, x_2, \ldots, x_k] - f[x_0, x_1, \ldots, x_{k-1}]}{x_k - x_0}. \tag{3.4}$$

Hierin hebben we, om de formules enigszins beknopt te houden, $i = 0$ genomen. Verder hebben we al gebruikt dat $f[x_0, \ldots, x_{k-1}]$ de afgeleide $f^{(k-1)}$ benadert in het gemiddelde $\frac{1}{k}(x_0 + x_1 + \ldots + x_{k-1})$ van de betrokken punten $x_0, \ldots, x_{k-1}$. Verder is gebruikt dat $\frac{1}{k}(x_1 + x_2 + \ldots + x_k) - \frac{1}{k}(x_0 + x_1 + \ldots + x_{k-1}) = \frac{1}{k}(x_k - x_0)$.

Vraag 3C. Is $f[x_0, \ldots, x_k]$ een $f^{(k)}$-waarde? In welke zin benadert $f[x_0, \ldots, x_k]$ de funktie $f^{(k)}$ in $\frac{1}{k+1}(x_0 + \ldots + x_k)$? Is $f[x_0, \ldots, x_k]$ afhankelijk van de volgorde van de x_j?

[55] Gebruik `vraag3b.m`.

[56] In de literatuur komt men ook het begrip '*ongedeelde* differentie' tegen. In de recursieve definitie van ongedeelde differentie ontbreekt de 2 in (3.1) en de k in (3.4). Verder valt de definitie samen met die voor gedeelde differentie. Ongedeelde differenties benaderen dus niet $f^{(k)}$ maar $\frac{1}{k!}f^{(k)}$ in de gemiddelde x_j-waarde.

Experiment. Experimenteer met de funktie $f(x) = \exp(x)$.[57]

Verklaring. Een verklaring is geheel analoog aan de verklaring in Vraag 3A en kan induktief gegeven worden. Het valt echter niet mee om een beknopt bewijs te vinden (zie §6.5).

Conclusie. $f[x_0, \ldots, x_k]$ is onafhankelijk van de volgorde van de x_j.
Als $f \in C^{(k)}([a, b])$ dan

$$f[x_0, \ldots, x_k] = f^{(k)}(\xi) \quad \text{voor zekere } \xi \text{ tussen } x_0, \ldots, x_k. \tag{3.5}$$

De waarde ξ ligt dicht bij het gemiddelde $\frac{x_0+\ldots+x_k}{k+1}$: voor $f \in C^{(k+2)}([a, b])$ geldt

$$f[x_0, \ldots, x_k] = f^{(k)}\left(\frac{x_0 + \ldots + x_k}{k+1}\right) + \mathcal{O}(h^2) \quad \text{voor} \quad h \to 0,$$

hierbij is $h \equiv \max\{|x_j - x_\ell| \mid j, \ell = 0, \ldots, k\}$.

In bovenstaande definities en beschouwingen hebben we stilzwijgend aangenomen dat de x_i onderling verschillend zijn. Voor het vormen van de differenties is er geen extra gladheid van f vereist: de differentie quotiënten kunnen gevormd worden voor iedere funktie f waarvoor de waarden $f(x_0)$, ..., $f(x_n)$ beschikbaar zijn. Met het oog op (3.5), kunnen we de differenties ook definiëren voor rijen punten x_0, ..., x_n waarin zekere x_i-waarden samenvallen. In zo'n geval zijn sommige differenties de waarden van echte afgeleiden die dan wel moeten bestaan en voor het berekenen moeten de waarden van die afgeleide beschikbaar moeten zijn.

Stel dat de x_i met gelijke waarde na elkaar genummerd zijn en stel dat f in het punt x_i d-maal continu differentieerbaar is in het geval dat $x_i = x_{i+d}$: d.w.z., voor relevante i en d geldt

$$x_i = x_{i+j} \quad \text{voor} \quad j = 1, \ldots, d \quad \text{en} \quad f \text{ is } C^{(d)} \text{ in } x_i \quad \text{als} \quad x_i = x_{i+d}. \tag{3.6}$$

Definieer dan

$$f[x_i, x_{i+1}, \ldots, x_{i+d}] \equiv f^{(d)}(x_i) \quad \text{als} \quad x_i = x_{i+d}. \tag{3.7}$$

Als f- en f'-waarden bekend zijn dan kunnen we deze aanpak volgen om hogere afgeleiden te berekenen. Merk op dat f- en f'-waarden bekend zijn als bijvoorbeeld f gedefinieerd is door een differentiaalvergelijking en die vergelijking numeriek opgelost is (zie Hoofdstuk I).

Opgave 3D. *Ga na dat definitie* (3.7), *in het geval* (3.6) *geldt, consistent is met de definities* (3.1) *en* (3.4), *d.w.z.,*

$$f[x_i, \ldots, x_{i+d}] = \lim_{y_j \to x_{i+j}} f[y_0, \ldots, y_d].$$

[57]Gebruik `vraag3c.m`.Voor de afhankelijkheid van de volgorde, gebruik `vraag3b.m`.

5.4 Betrouwbaar de fout schatten, minimale onbetrouwbaarheid

SLEUTELBEGRIPPEN: *Stabiliteit numeriek differentiëren, de fout schatten, vertrouwensgetallen, de fout betrouwbaar schatten, maximale nauwkeurigheid, ongestruktureerde fouten.*

De aanpak in §§5.2–5.3 om waarden van de afgeleide van f te bepalen met behulp van waarden van f in zekere punten, lijkt doeltreffend. Maar de vraag blijft of we de afgeleide met iedere gewenste nauwkeurigheid kunnen bepalen. Verder kan men zich afvragen hoe efficiënt achterhaald kan worden of de gewenste nauwkeurigheid ook behaald is?

In deze sektie bekijken we alleen rijen punten x_i die *equidistant verdeeld liggen*, d.w.z., die voor zekere stapgrootte $h > 0$, van de vorm $x_i \equiv a + ih$ zijn. We benaderen f' in x_i door $f[x_i - h, x_i + h] = f[x_{i-1}, x_{i+1}]$. Merk op dat nu $\mu_i = \frac{1}{2}(x_{i-1} + x_{i+1}) = x_i$. Verder is de afstand tussen de twee steunpunten x_{i-1} en x_{i+1} gelijk aan $2h$.

Vraag 4A. Kan je al rekenend, door met verschillende stapgroottes te werken, de fout schatten in de benaderende waarden van f'?

Experiment. Bekijk op $[-2, +1]$ de funktie $f(x) = 1/(1+8x^2)$. Bepaal, voor $h = 0.2$, de waarden van f in de punten x_i. Gebruik die waarden om f' te benaderen in de μ_i. Halveer h eens. Hoe neemt de fout af als je h halveert? Kan je de inzichten gebruiken om de fout te schatten? Met welke waarde voor h behaal je een nauwkeurigheid 10^{-4}? Doe hetzelfde eens voor $h = 0.6$. Hoe neemt de fout nu af als je h halveert? Hangt dit nog af van het punt μ_i?

Verklaring. De fout in de benadering van f' in μ is ongeveer gelijk aan $h^2 C$ met $C \approx \frac{1}{6} f'''(\mu)$ (zie (2.3)), mits h zo klein is dat de 'hoger orde Taylorreeks termen' verwaarloosbaar zijn.

De situatie is tamelijk analoog aan die in Hoofdstuk I, §1.3 (zie Vraag 3G). Dit hoeft ons niet te verbazen. De numerieke methoden die we in dat hoofdstuk gebruikt hebben zijn immers gebaseerd zijn op het soort differentie quotiënten dat we nu onder de loep hebben.

Vraag 4B. Kan je f' in iedere gewenste nauwkeurigheid berekenen?

Experiment. Met welke h denk je dat je in het voorbeeld in het vorige experiment een nauwkeurigheid haalt van 10^{-8}? Bepaal eerst h zonder de computer te gebruiken, maar met de numerieke resultaten van het experiment in de vorige vraag en met het feit dat de fout ongeveer Ch^2 is. Kontroleer nu je voorspelling. Met welke h denk je een nauwkeurigheid van 10^{-12} te halen? Klopt die voorspelling? Pakt het beter uit als f een gladdere funktie is dan $\exp(x)$?

Verklaring. Voor kleinere waarden van h spelen afrondfouten een grote rol. De funktiewaarden $f(x_i)$ worden in een groot, maar beperkt, aantal cijfers nauwkeurig berekend. In het differentie quotiënt wordt door h gedeeld. Hierdoor worden de afrondfouten vergroot.

Conclusie. De fout in de benadering $f[x_{i-1}, x_{i+1}]$ van $f'(x_i)$ is gelijk aan $h^2 C$ waarbij $C \approx \frac{1}{6} f'''(\mu_i)$ (zie (2.3)), mits

- h voldoende klein is (zodat hogere orde Taylorreeks termen verwaarloosbaar zijn)

en

- h niet zo klein is dat rekenfouten niet verwaarloosbaar zijn.

Opgave. *Bewijs dat $\overline{\varepsilon}/h$ de onzekerheid (d.w.z. een bovengrens) is ten gevolge van afrondfouten op f. Hierin is $\overline{\varepsilon}$ de onzekerheid in de f-waarden.*

Vraag 4C. Hoe kan je al rekenend achterhalen of aan de twee 'mitsen' voldaan is? Hoe schat je de fout?

Experiment. Bepaal in de punten $\mu_i = c$ voor de stapgroottes $4h$, $2h$ en h, de quotiënten van opeenvolgende differenties:

$$V(c) \equiv \frac{f[c-2h, c+2h] - f[c-h, c+h]}{f[c-h, c+h] - f[c-\frac{1}{2}h, c+\frac{1}{2}h]}. \tag{4.1}$$

Kijk naar de waarden van $V(c)$ voor grotere waarden van h ($h = 0.2$) en ook voor kleinere ($h = 10^{-5}$). Als de fout in $f[c-h, f+h]$ evenredig is met h^2 is $V(c) = 4$ (Waarom?). Naarmate $V(c)$ meer afwijkt van vier zal aan die evenredigheid slechter voldaan zijn. Zie je dat terug voor grotere en voor kleinere waarde van h?

Als $V(c) \approx 4$ lijkt $\frac{1}{3}\left(f[c-h, c+h] - f[c-\frac{1}{2}h, c+\frac{1}{2}h]\right)$ een betrouwbare schatting voor de fout in de benadering $f[c-\frac{1}{2}h, c+\frac{1}{2}h]$ van $f'(c)$ (Waarom?). Zie je dat terug in de experimentele resultaten?

Verklaring. Zie de verklaring in Vraag 3G van Hoofdstuk I. Teken ook voor de huidige situatie de relevante grootheden op een getallenrechte; bedenk hierbij dat de fout nu evenredig is (of evenredig behoort te zijn) met h^2.

Conclusie. Als de zogenaamde **vertrouwensgetallen** $V(c)$ dicht in de buurt liggen van vier, dan kunnen we er enigszins op vertrouwen dat aan de mitsen in de conclusie van Vraag 4B is voldaan en lijkt $\frac{1}{3}\left(f[c-h, c+h] - f[c-\frac{1}{2}h, c+\frac{1}{2}h]\right)$ een betrouwbare schatting voor de fout in de benadering $f[c-\frac{1}{2}h, c+\frac{1}{2}h]$ van $f'(c)$.

We hebben gezien dat de effekten van afrondfouten verhinderen dat de f'-waarden in iedere gewenste nauwkeurigheid bepaald kan worden. Wat is de grootste nauwkeurigheid die haalbaar is?

Vraag 4D. Wat is de kleinste onbetrouwbaarheid waarmee we met $f[x_i, x_{i+1}]$ de waarde $f'(\mu_i)$ kunnen bepalen?

Experiment. Bepaal experimenteel de h waarvoor je min of meer de nauwkeurigste benaderingen hebt.[58]

Verklaring. De fout (of, preciezer, de onbetrouwbaarheid) bestaat uit twee componenten: de benaderingsfout en de effekten van de afrondfouten (zie de sektie "Fouten" op p.1). De benaderingsfout is, voor h voldoende klein, $\approx \frac{1}{6}h^2 f''(\mu_i)$. De effekten van de afrondfouten kunnen gemajoreerd worden door $\overline{\varepsilon}/h$, hierin is $2\overline{\varepsilon}$ de onbetrouwbaarheid in de f-waarden en hebben we andere bronnen van afrondfouten verwaarloosd. De totale fout wordt redelijk gemajoreerd door de funktie

$$\phi(h) \equiv \tfrac{1}{6}h^2|f''(\mu_i)| + \frac{2\overline{\varepsilon}}{h}. \tag{4.2}$$

Als funktie van h, met $h > 0$, heeft deze funktie een minimum > 0.

Opgave. *Stel dat $f^\star(x_i) = f(x_i) + \varepsilon(x_i)$ met ε zo dat $|\varepsilon(x_i)| \leq \overline{\varepsilon}$ voor iedere x_i.*
Ga na dat dan $|f^\star[x_i, x_{i+1}] - f[x_i.x_{i+1}]| \leq 2\,\overline{\varepsilon}/h$.
Ga na dat ϕ in (4.2) een redelijke majorant is voor $|f^\star[x_i, x_{i+1}] - f'(\mu_i)|$. Waarom 'redelijk'?

[58] Gebruik `vraag4c.m`.

Bepaal de minimale waarde van ϕ voor de funktie f uit het experiment. Voor welke waarde van h wordt dat minimum aangenomen? Komen deze waarden overeen met die uit het experiment?

Vraag 4E. De funktie ϕ in (4.2) is een majorant van de fout. Is de schatting veel te grof? Als de schatting te grof is, kunnen we van te voren weten voor welke waarden van h (en voor welke f en x_i) dat het geval zal zijn?

Experiment. Kies een punt c. Plot de absolute waarde van de fout in $f[c-h, c+h]$ voor diverse waarden van h rond h_0, waarbij h_0 de optimale stapgrootte is uit het vorige experiment.

Verklaring. Afrondfouten zijn, in tegenstelling tot benaderingsfouten, ongestruktureerd, dat wil zeggen, ze hebben geen mooie struktuur waarmee je voorspellingen kan doen.[59]

Vraag 4F. Kunnen we met een andere strategie uit de $f(x_i)$-waarden de f'-waarden berekenen die nauwkeuriger zijn dan de minimale waarde van de majorant ϕ uit (4.2)?

Experiment. Probeer $f'(c)$ te benaderen door

$$f^\bullet(c) \equiv \frac{f(c-2h) - 8f(c-h) + 8f(c+h) - f(c+2h)}{12h}. \tag{4.3}$$

Probeer verschillende h (halveer h) en zoek uit hoe de fout van h afhangt in geval h 'voldoende klein' is (om Taylorreeks resttermen te mogen verwaarlozen), maar ook weer niet 'te klein' (zodat afrondfouten het beeld niet vertroebelen)? Wat is de kleinste onbetrouwbaarheid die haalbaar is?

Verklaring. De approximatie fout in $f^\bullet(c)$ is, t.o.v. die in $f[c-h, c+h]$, al behoorlijk klein voor niet al te kleine h. Omdat met grotere h gewerkt kan worden, worden de afrondfouten minder opgeblazen.

Opgave. *Bewijs dat $f^\bullet(c) - f'(c) = -\frac{1}{30}h^4 f^{(5)}(\xi)$ voor zekere $\xi \in (c-2h, c+2h)$.*
Zie in dat $|(f^\star)^\bullet(c) - f'(c)|$ redelijk gemajoreerd wordt door $\frac{1}{30}h^4|f^{(5)}(c)| + 18\,\overline{\varepsilon}/(12\,h)$.
Bepaal het minimum van deze majorant voor de funktie uit het experiment.

Conclusie. Met hogere orde schema's als in (4.3) (d.w.z. schema's met een hogere orde approximatie fout) kunnen de vervelende effecten van afrondfouten beperkt worden.

Als afrondfouten geen rol spelen en f glad genoeg is, dan benaderen de gedeelde differenties $f[x_i, \ldots, x_{i+k}]$ de waarden $f'((x_i + \ldots + x_{i+k})/(k+1))$ met nauwkeurigheid $\mathcal{O}(h^2)$. De orde h^2 nauwkeurigheid geldt voor iedere k. Hangen de effecten van afrondfouten ook op zo'n 'milde' manier af van k, of zijn die effekten veel groter voor grotere k?

Vraag 4G. Hoe nauwkeurig kunnen we $f^{(6)}(x_{i+3})$ bepalen met $f[x_i, \ldots, x_{i+6}]$?

Experiment. [60]

Verklaring. De effekten van afrondfouten in f-waarden worden gemajoreerd door $2^k\overline{\varepsilon}/h^k$.

[59] In `ruis.m` wordt het 9de graads polynoom $p = x(x-1)(x-2)\cdot\ldots\cdot(x-8)$ uitgeschreven in termen van machten van x en vervolgens geevalueerd in de buurt van $x = 8$. Bekijk de grafiek voor de door de computer berekende p op het interval $[8-\delta, 8+\delta]$ voor verschillende δ ($\delta = 10^{-6}, 10^{-9}, 10^{-10}, 10^{-11}, \ldots$). Bekijk ook eens de grafiek van de funktie exp. Voor beide funkties hebben we te maken met afrondfouten. De effekten zijn echter niet gelijk. Kan je dit verklaren?

[60] Gebruik `vraag3c.m`.

Opgave. *Laat zien dat de majorant in de 'Verklaring' correct is.*

Conclusie. Numeriek differentiëren is instabiel.
Zeker met de voor-de-hand-liggende differenties en voor hogere afgeleiden kunnen afrondfouten het behalen van een grote nauwkeurigheid onmogelijk maken.

Fouten op f-waarden hebben een 'lokaal' effect: de fout $\varepsilon(x_i)$ in het punt x_i werkt alleen door in $f[x_j,\ldots,x_{j+k}]$ voor $j = i-k, i-k+1,\ldots,i$ (Waarom?).

Vraag 4H. Hoe zou je het lokale doorwerken van fouten en het sterk doorwerken van fouten in de differenties kunnen gebruiken om fouten op te sporen?

Experiment. Verstoor in een punt x_{i_0} de waarde $f(x_{i_0})$ met $\varepsilon(x_{i_0}) = 10^{-\ell}$ voor $\ell = 4$. Verder verstoor je geen f-waarden. Bekijk de gedeelde differenties (voor $k < 6$). In welke differenties zie je de effekten van de verstoring? Zie je dat ook al voor kleinere k en voor grotere ℓ?

5.5 Samenvatting

We bekijken een funktie f die gedefinieerd is op $[a,b]$.
Definieer recursief, voor $k = 2,3,\ldots$, de k-**de gedeelde differentie**

$$f[x_i,x_{i+1},\ldots,x_{i+k}] \equiv k\,\frac{f[x_{i+1},x_{i+2},\ldots,x_{i+k}] - f[x_i,x_{i+1},\ldots,x_{i+k-1}]}{x_{i+k}-x_i},$$

waarbij

$$f[x_i,x_{i+1}] \equiv \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}.$$

Stelling 5.1 *Als π een permutatie is van $\{0,1,\ldots,k\}$ dan*

$$f[x_0,x_1,\ldots,x_k] = f[x_{\pi(0)},x_{\pi(1)},\ldots,x_{\pi(k)}].$$

Verder geldt, als $f \in C^{(k)}([a,b])$, dat

$$f[x_0,\ldots,x_k] = f^{(k)}(\xi) \quad \text{voor zekere } \xi \text{ tussen} \quad x_0,\ldots,x_k, \tag{5.1}$$

en als $f \in C^{(k+2)}([a,b])$ dan geldt, met $h \equiv \max\{|x_j - x_i| \mid i,j = 0,\ldots,k\}$, dat

$$f[x_0,\ldots,x_k] = f^{(k)}\left(\frac{x_0+\ldots+x_k}{k+1}\right) + \mathcal{O}(h^2) \qquad (h \to 0).$$

De laatste uitspraak in de stelling volgt uit het volgende lemma door voor f een Taylor polynoom benadering te nemen, de eerste en de tweede uitspraak bewijzen we later (zie de Stellingen 6.5 en 6.4). Het resultaat in (5.1) is de **gegeneraliseerde middelwaarde stelling** van de differentiaalrekening genoemd.

Lemma 5.2 *Als p een polynoom is van graad $\leq k+1$ dan $p[x_0,\ldots,x_k] = p^{(k)}\left(\frac{x_0+\ldots+x_k}{k+1}\right)$.*

Bewijs. De polynomen $p_0 \equiv 1$, $p_j(x) \equiv (x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_{j-1})$ vormen, voor $j = 1, \ldots, k+1$, een basis voor ruimte van alle polynomen van graad $\leq k+1$. Voor p_k en p_{k+1} gaat men, door uitschrijven, na dat de uitspraak klopt. Voor p_j met $j < k$ volgt de uitspraak middels een induktie argument. □

Stel nu dat de f-waarden met een fout ε belast zijn: we rekenen met $f^\star$ waarbij

$$f^\star(x) = f(x) + \varepsilon(x), \quad \text{voor alle} \quad x \in [a, b]. \tag{5.2}$$

Van $\varepsilon(x)$ weten we dat

$$|\varepsilon(x)| \leq \overline{\varepsilon} \quad \text{voor alle} \quad x \in [a, b] \tag{5.3}$$

en kennen we de waarde $\overline{\varepsilon}$: $\overline{\varepsilon}$ is de onbetrouwbaarheid in de f-waarden. Verder weten we niets (of wensen we niets te weten) over de 'numerieke ruis' ε.

Stelling 5.3 *Stel dat de x_i* equidistant *verdeeld liggen, d.w.z. voor zekere $h > 0$ geldt $x_i = x_0 + ih$ voor $i = 1, \ldots, k$. Dan geldt*

$$|f^\star[x_0, \ldots, x_k] - f[x_0, \ldots, x_k]| \leq \frac{2^k}{h^k} \overline{\varepsilon}. \tag{5.4}$$

De schatting is scherp *in de volgende zin: er is een funktie ε die aan* (5.3) *voldoet en waarvoor in* (5.4) *de ongelijkheid in feite een gelijkheid is.*

Betrouwbaar de fout schatten. De fout in de benadering $f^\star[x_0, \ldots, x_k]$ van $f^{(k)}(\mu)$, met $\mu \equiv \frac{x_0 + \ldots + x_k}{k+1}$, is opgebouwd uit twee componenten: de *benaderingsfout* en de *evaluatiefout*. De evaluatiefout is het effekt van afrondfouten. Deze fout is gewoonlijk *ongestruktureerd* en alleen een majorant (onbetrouwbaarheid) als in (5.4) kan ervoor gegeven worden. De benaderingsfout ontstaat door te werken met een de wiskundige benadering (in dit geval met $f[x_0, \ldots, x_k]$). De benaderingsfout heeft, in geval f voldoende glad is, een *struktuur*. De struktuur is bekend en dat kan uitgebuit worden om bijvoorbeeld de fout te schatten. De evaluatiefout moet dan wel verwaarloosbaar zijn ten opzichte van de benaderingsfout.

De benaderingsfout in $T(h) \equiv f^\star[x_0, \ldots, x_k]$ is, in geval de betrokken x_j equidistant verdeeld liggen, voor voldoend kleine h ongeveer evenredig met h^2 (dus ongeveer van de vorm Ch^2). Hierbij is h de stapgrootte, d.w.z. de afstand tussen twee opeenvolgende x_j (de evenredigheidsfactor C is een reëel getal onafhankelijk van h). Dit betekent bijvoorbeeld dat de benaderingsfout min of meer halveert als we h halveren en h voldoende klein is. Dus, als

(i) h voldoende klein is en

(ii) de evaluatiefout verwaarloosbaar is,

dan

$$T(\tfrac{1}{2}h) - f^{(k)}(\mu) \approx \tfrac{1}{3}\left(T(h) - T(\tfrac{1}{2}h)\right). \tag{5.5}$$

Merk op dat we voor deze schatting de preciese waarde van de evenredigheidsfactor niet hoeven te kennen; we hoeven alleen te weten dat er evenredigheid met h^2 is. Als de fout ongeveer evenredig is met h^2, dus als (i) en (ii) gelden, dan is

$$V(h) \equiv \frac{T(2h) - T(h)}{T(h) - T(\frac{1}{2}h)} \approx 4. \tag{5.6}$$

Met ander woorden, als $V(h)$ niet ongeveer 4 is (zeg > 4.4 of < 3.6) dan is h niet klein genoeg of zijn evaluatiefouten niet verwaarloosbaar. Als we mogen aannemen dat (i) en (ii) gelden dan is schatting (5.5) correct. We mogen enig vertrouwen hebben in de juistheid van deze aanname als (5.6) geldt. We noemen $V(h)$ dan ook het *vertrouwensgetal*.

Hoe nauwkeurig schatting (5.5) is, d.w.z. hoe dicht $\frac{1}{3}(T(h)-T(\frac{1}{2}h))$ bij de echte fout ligt, wordt bepaald door de relatieve grootte van de evaluatiefout en van de hogere order term (de $\mathcal{O}(h^3)$-term) in de benaderingsfout (relatief ten opzichte van Ch^2). Zijn deze bijdrage kleiner dan is de schatting nauwkeuriger. In dat geval ligt $V(h)$ ook dichter bij 4. $V(h)$ kan dus ook een indikatie geven over de nauwkeurigheid van schatting (5.5). Willen we alleen een indikatie hebben over de onbetrouwbaarheid in de benadering $T(h)$ van $f^{(k)}(\mu)$ dan hoeft de schatting niet zo vreselijk nauwkeurig te zijn: de mededeling dat de onbetrouwbaarheid bijvoorbeeld gelijk is aan $0.536\,10^{-4}$ is net zo informatief als dat de onbetrouwbaarheid $0.6\,10^{-4}$ is (of 10^{-4}).

Aan het schatten van de fout hangt een prijskaartje: we moeten niet alleen $T(\frac{1}{2}h)$ uitrekenen, maar ook $T(h)$ (voor schatting (5.5)) en, als we vertrouwen willen krijgen in de betrouwbaarheid van de schatting, ook nog $T(2h)$ (zie (5.6)).

Maximale nauwkeurigheid. Voor x_j's die equidistant verdeelde liggen met stapgrootte h wordt de fout (benaderingsfout plus evaluatiefout) in $T(h) \equiv f^*[x_0, \ldots, x_k]$ gemajoreerd door

$$\phi(h) \equiv Ch^2 + 2^k\overline{\varepsilon}/h^k.$$

Hierbij is aangenomen dat h zo klein is dat de $\mathcal{O}(h^3)$-term in de benaderingsfout verwaarloosbaar is (t.o.v. Ch^2). Gewoonlijk is dit een realistische aanname zelfs voor waarden van h die veel groter zijn (denk aan een factor 10 of 100) dan die waarvoor ϕ minimaal is. De schatting is scherp als $\overline{\varepsilon}$ een scherpe bovengrens is voor de fout in de $f^\star(x_j)$ (en de $\mathcal{O}(h^3)$-term verwaarloosbaar zijn). De majorant laat zien dat voor kleine h de evaluatiefout de totale fout domineert, terwijl voor grotere h de benaderingsfout dat doet.
Voor $h \to 0$ gaat de evaluatiefout naar ∞: *numeriek differentiëren is instabiel.*
De majorant is minimaal voor die h waarvoor $\phi'(h) = 0$. Dan is $2\,Ch^2 = k\,2^k\overline{\varepsilon}/h^k$. De benaderingsfout is, op een factor $2/k$ na, gelijk aan de evaluatiefout: beide componenten zijn in dezelfde orde van grootte. De optimale h wordt, voor $D \equiv \sqrt[k+2]{\frac{k\,2^{k-1}}{C}}$, gegeven door $h = D\,\sqrt[k+2]{\overline{\varepsilon}}$. Dit correspondeert met de ϕ-waarde $D'\,\overline{\varepsilon}^{\,2/(k+2)}$ waarbij $D' \equiv \frac{2+k}{k}\,C\,D^2$. Voor bijvoorbeeld $k = 2$ en $\overline{\varepsilon} = 10^{-12}$ heeft het geen zin h kleiner te nemen dan $D\,10^{-3}$ en zal, voor geen enkele stapgrootte de onbetrouwbaarheid kleiner zijn dan $D'\,10^{-6}$. De waarde van de constanten D en D' hangt natuurlijk af van f, maar voor een interpretatie van de bewering kan het geen kwaad om aan 1 te denken: hebben we f-waarden in 12 cijfers, dan moeten we vrezen dat we f''-waarden in slechts 6 cijfers kunnen berekenen. Voor grotere k moeten we vrezen dat we meer te lijden hebben van effekten van afrondfouten. Met de methode uit de vorige paragraaf om de fout te schatten (schatting (5.5) en 'vertrouwen' (5.6)) kunnen we dicht in de buurt komen van maximale nauwkeurigheid zonder de constante C (D en D') expliciet te bepalen.

Met een ander benaderingsschema, waarin bijvoorbeeld de benaderingsfout evenredig is met h^4 (zie (4.3)), zouden we een grotere nauwkeurigheid kunnen halen. In zo'n schema is, voor optimale h, de benaderingsfout weer in dezelfde orde van grootte als de evaluatiefout. Vanwege de h^4-evenredigheid, zal de benaderingsfout voor grotere h al klein zijn. Voor de grotere h is het effect van de afrondfouten kleiner.

6 Interpolatie II

6.1 Inleiding

In Hoofdstuk II, §2 hebben we gezien hoe we, middels *interpolatie*, de waarde $f(t)$ van een funktie f kunnen benaderen voor punten t die tussen punten x_i in liggen en waarvoor de waarden $f(x_i)$ beschikbaar zijn. In deze sektie gaan we wat dieper in op de interpolatietheorie.

In deze sektie is $a \leq t_0 < t_1 < \ldots < t_n < t_{n+1} = b$ en zijn de waarden van f (en eventueel van diens afgeleide(n)) in de punten t_i beschikbaar. De waarden $f(t_i)$ zouden bijvoorbeeld door meting verkregen kunnen zijn of zouden de numeriek verkregen oplossing van een differentiaalvergelijking kunnen zijn.[61]

Voorbeeld 1A. We besteden aandacht aan de volgende voorbeelden

$$f(x) = \exp(x), \tag{1.1}$$

$$f(x) = \frac{1}{1+\alpha x^2} \quad \text{met} \quad \alpha = 25, \tag{1.2}$$

$$f(x) = \frac{\cos(\gamma x) + \sin(x) - x - 1}{x} \quad \text{voor} \quad x \neq 0, \quad \text{met} \quad \gamma = 1.2, \tag{1.3}$$

en zijn geïnteresseerd in hoe interpolatie uitpakt op het interval $[-1, +1]$ en met name in de buurt van 0.

In de praktijk zullen we geen behoefte hebben om deze funkties via interpolatie te benaderen, maar de voorbeelden staan model voor achtereenvolgens, de situatie waarin f glad is en de afgeleiden niet groot zijn (de f in (1.1)), f glad is maar de afgeleiden, zekere de hogere in de buurt van 0 (de f in (1.2)), groot zijn, en tenslotte voor het berekenen van een limiet door middel van interpolatie: in (1.3) zijn we geïnteresseerd in $\lim_{x\to 0} f(x)$.

6.2 Stuksgewijze interpolatie

Sleutelbegrippen: *Stuksgewijze interpolatie*

Als de funktiewaarden $f(x_0), \ldots, f(x_k)$ beschikbaar zijn dan zouden we het k-de graads Lagrange interpolatiepolynoom kunnen vormen (zie §2.4). Echter, voor grotere k moet er nogal wat rekenwerk verzet worden. Bovendien hebben we in Vraag 4B van Hoofdstuk II gezien dat er geen garantie is dat een grotere k een nauwkeurig interpolatieresultaat oplevert. Maar waarom zouden we met grotere k moeten werken? We zouden ook al nauwkeurige resultaten kunnen boeken door te interpoleren in een paar punten x_i die in de 'buurt' liggen van het punt x waarvoor we $f(x)$ willen weten.

Vraag 2A. We willen $f(x)$ nauwkeurig benaderen met behulp van de beschikbare funktie waarden $f(t_i)$. Op welke punten geeft interpolatie het beste resultaat?

We formuleren die vraag wat preciezer.
We willen $k+1$ verschillende punten $x_0, \ldots, x_k$ kiezen uit de collektie van t_i's en $f(x)$ benaderen door de waarde $p(x)$ van het Lagrange interpolatiepolynoom p voor f op $(x_0, \ldots, x_k)$. Hoe groot moet k zijn? Welke punten t_i kunnen we het best kiezen? Hoe

[61] De programma's voor deze sektie staan in de folder `Romberg`.

hangt dat samen met de positie van x? Hangt dat nog af van f? Maakt het nog uit of de t_i equidistant verdeeld liggen, d.w.z., $t_i = t_0 + ih$ voor zekere stapgrootte $h > 0$?

Experiment. Onderzoek de vragen experimenteel voor de funkties in Voorbeeld 1A.[62]

Conclusie. Het lijkt een goede strategie om de x_j zo te kiezen dat x in het 'middelste' interval ligt, als de t_i min of meer equidistant verdeeld liggen, de beste strategie: met $x \in (t_\ell, t_{\ell+1})$ kies $x_0 = t_\ell$, $x_1 = t_{\ell+1}$, $x_2 = t_{\ell-1}$, $x_3 = t_{\ell+2}$, $x_4 = t_{\ell-2}$, ... (kies de x_i alternerend rond t_ℓ zodat $(t_\ell, t_{\ell+1})$ het 'middelste interval' is in de collektie van relevante intervalletjes (t_i, t_{i+1})).

Verklaring. De fout is van de vorm $A(x)\frac{f^{(k+1)}(\xi)}{(k+1)!}$ met $A(x) \equiv (x - x_0) \cdot \ldots \cdot (x - x_k)$. Als de t_i min of meer equidistant verdeeld liggen is de absoluut maximale waarde van het polynoom A op de betrokken intervalletjes (t_i, t_{i+1}) op het 'middelste' intervalletje (zie 'Conclusie') het kleinst. Als de $f^{(k+1)}$ niet al te sterk varieert is deze polynoomeigenschap bepalend.[63]

Merk op dat de vraag in Vraag 2A een andere is dan die in Vraag 4C in Hoofdstuk II. In deze laatste vraag hadden we de vrijheid de t_i te kiezen. Nu liggen de t_i's vast en kunnen we alleen uit deze collektie kiezen. We zijn ook nog vrij in het doorlopen van de volgorde. De volgorde is overigens voor de waarde $p(x)$ niet van belang (Waarom niet?).

MATLAB heeft ook een interpolatie procedure.

Vraag 2B. Op wat voor manier interpoleert MATLAB? Is de aanpak die MATLAB aanbiedt toereikend?

Experiment. Bekijk voor bijvoorbeeld de funktie f in (1.2) de MATLAB interpolatie resultaten voor de vier opties die MATLAB biedt (zie eventueel `help interp1`) en probeer uit de resultaten af te leiden wat MATLAB doet.[64] Kan je tekortkomingen zien van de verschillende MATLAB benaderingen?

Er zijn situaties waarin er behoefte is aan hogere graads interpolatie. Het is van te voren niet duidelijk met welke k we resultaten krijgen die voldoende nauwkeurig zijn. Het hoeft niet gunstig te zijn om met een grote k te werken. Afrondfouten kunnen ongewenste effekten hebben, de hoeveelheid rekenwerk neemt toe. Wellicht kunnen we k verhogen en met de resultaten voor opeenvolgende waarden van k schatten hoe groot de fout is. De vraag is hoe dat moet en of het efficiënt kan. Dit is het onderwerp van de volgende sektie.

6.3 Romberg schema en het schatten van de fout

SLEUTELBEGRIPPEN: *Romberg schema, de fout betrouwbaar schatten, schema van Horner*

We kijken eerst naar een manier om een waarde voor een polynoom interpolatie van graad k te berekenen uit waarden voor interpolatie van graad $k - 1$. Ga na dat de volgende relatie geldt.

Als $p_{i,j}$ het Lagrange interpolatiepolynoom is voor f op $(x_{i-j}, \ldots, x_i)$ dan is

$$p_{i,j+1}(x) = p_{i,j}(x) + \frac{x - x_i}{x_{i-1-j} - x_i}\left(p_{i-1,j}(x) - p_{i,j}(x)\right). \tag{3.1}$$

[62] Gebruik `vraag2a.m`.

[63] Je kan de grafiek van A plotten: kies `FOUT_POL = 1` in `vraag2a.m`.

[64] Gebruik `vraag2b.m`.

Hierbij is $p_{i,0}(x) = f(x_i)$, het constante polynoom met waarde $f(x_i)$.

De $p_{i,j}(x)$ kunnen dus rekursief berekend worden. We kiezen een x en we schrijven $T_{i,j}$ voor de waarde $p_{i,j}(x)$.[65] De rekursieve berekening kunnen we schematisch voorstellen in een zogenaamd **Romberg schema**:[66]

$$\begin{array}{llllll} x_0 \to & f(x_0) = T_{0,0} & & & & \\ & & \searrow & & & \\ x_1 \to & f(x_1) = T_{1,0} & \to & T_{1,1} & & \\ & & \searrow & & \searrow & \\ x_2 \to & f(x_2) = T_{2,0} & \to & T_{2,1} & \to & T_{2,2} \\ \vdots & \quad\vdots & & & & \ddots \end{array} \tag{3.2}$$

De lijntjes geven aan uit welke grootheden links van de lijntjes, de grootheid rechts ervan berekend wordt. Hieronder hebben we een deel uit dit schema gelicht om aan te geven hoe nieuwe kolommen gevormd worden.

$$\begin{array}{llllllll} x_{i-1-j} & \to & T_{i-1-j,0} & & & & & \\ \vdots & & \vdots & \ddots & & & & \\ & & & & \searrow & & & \\ x_{i-1} & \to & T_{i-1,0} & \cdots & \to & T_{i-1,j} & & \\ & & & & \searrow & & \searrow & \\ x_i & \to & T_{i,0} & \cdots & \to & T_{i,j} & \to & T_{i,j+1} \equiv T_{i,j} + \dfrac{x - x_i}{x_{i-1-j} - x_i}(T_{i-1,j} - T_{i,j}) \end{array}$$

Vraag 3A. Mogen we langs de diagonaal van het Romberg schema de beste resultaten verwachten (met andere woorden, is $|f(x) - T_{i,i}|$ kleiner dan $|f(x) - T_{i+j,i}|$)? Hangt dit nog af van de volgorde waarmee we de steunpunten $x_0, \ldots, x_k$ kiezen uit de $t_{i_0}, \ldots, t_{i_0+k}$? Wat is de geschikste i_0 (Zie Vraag 2A)? Is $T_{i,i} - T_{i-1,i-1}$ een betrouwbare foutschatting voor de fout $f(x) - T_{i,i}$ in $T_{i,i}$?

Experiment. Onderzoek deze vraag voor de funkties f in (1.1) en (1.2) (met $\alpha = 1.2$ en $\alpha = 25$).[67] Merk op dat $T_{k,k}$ niet afhangt van de volgorde waarmee de steunpunten gekozen zijn. Was dit te verwachten?

Vraag 3B. We mogen verwachten (Waarom?) dat $T_{i,j+1}$ een betere benadering is van $f(x)$ dan $T_{i,j}$ (Waarom?). Als dit inderdaad zo is, dan is

$$E_{i,j} \equiv \frac{x - x_i}{x_{i-1-j} - x_i}(T_{i-1,j} - T_{i,j}) \quad \text{voor} \quad i \geq 2 \tag{3.3}$$

een goede schatting voor de fout in $T_{i,j}$ (Waarom?). Hoe pakt dit uit in de praktijk? Hoe vergelijkt deze schatting zich met die in de vorige vraag? In welke situaties is de schatting beroerd?

Experiment. Onderzoek deze vraag voor de drie funkties in Voorbeeld 1A; kombineer de funktie in (1.3) met een exponentieel dalende rij als $x_n = 2^{-n}$.[68]

[65] In deze subsektie is x soms een constante en soms een variabele. Hopelijk blijkt de preciese rol van x uit de context.

[66] Romberg paste het schema toe om limieten te berekenen (zie §7 verderop). In de huidige context wordt het schema vaak het Aitken-Neville schema genoemd.

[67] Gebruik `vraag4a.m` en de drie keuzes van de volgorde die in deze m-file gemaakt zijn.

[68] Gebruik `vraag3b.m`. Het programma `Romberg.m` dat op de laatste regels van `vraag3b.m` wordt aangeroepen kan $E_{i,j}$, berekenen en de resultaten in een schema laten zien. Als de exacte waarde $f(x)$ bekend is laat het programma ook de exacte fouten laten zien. Type `help Romberg` voor een beschrijving van de opties.

Verklaring. Als we de leidende coëfficiënt van $p_{i,j}$ noteren met α_{ij}, d.w.z. $p_{i,j}(x) = \alpha_{i,j}x^j +$ *lager graads termen*, dan vertelt (3.1) ons dat

$$\alpha_{i,j+1} = \frac{\alpha_{i-1,j} - \alpha_{i,j}}{x_{i-1-j} - x_i}.$$

Verder is $\alpha_{i,0} = f(x_i)$. De leidende coëfficiënten zijn blijkbaar (zie (3.4) in §5.3) op dezelfde manier gerelateerd als de ongedeelde differenties. Dit impliceert dat

$$\alpha_{i,j} = \frac{f[x_{i-j}, \ldots, x_i]}{j!}$$

(Ga dit na met induktie). Het j-de graads polynoom $p_{i-1,j} - p_{i,j}$ is gelijk aan 0 op de j steunpunten $x_{i-j}, \ldots, x_{i-1}$ (Waarom?). Dus geldt (volgens de Hoofdstelling van de Algebra)

$$p_{i-1,j}(x) - p_{i,j}(x) = \beta(x - x_{i-j}) \cdot \ldots \cdot (x - x_{i-1}) \quad \text{voor 'n scalar } \beta. \tag{3.4}$$

Vergelijken we in (3.1) de leidende coëfficiënten, dan zien we dat $\frac{\beta}{x_{i-1-j} - x_i} = \alpha_{i,j+1}$ en leidt substitutie van (3.4) in (3.1) tot

$$p_{i,j+1}(x) = p_{i,j}(x) + (x - x_{i-j}) \cdot \ldots \cdot (x - x_i) \frac{f[x_{i-j-1}, \ldots, x_i]}{(j+1)!}. \tag{3.5}$$

Bovendien (zie (3.3) en (3.1))

$$E_{i,j} = (x - x_{i-j}) \cdot \ldots \cdot (x - x_i) \frac{f[x_{i-j-1}, \ldots, x_i]}{(j+1)!}. \tag{3.6}$$

Als f een funktie is die voldoende glad is, dan geldt volgens (3.5) dat $f[x_{i-j-1}, \ldots, x_i] = f^{(j+1)}(\xi)$ voor 'n ξ in de buurt van $\frac{x_{i-j-1} + x_{i-j} + \cdots + x_i}{j+2}$. Bovendien is

$$f(x) = p_{i,j}(x) + (x - x_{i-j}) \cdot \ldots \cdot (x - x_i) \frac{f^{(j+1)}(\xi')}{(j+1)!}, \tag{3.7}$$

voor 'n ξ' in de buurt van $\frac{x + x_{i-j} + \cdots + x_i}{j+2}$. Dit verklaart waarom $E_{i,j}$ een goed schatting is voor de fout in $p_{i,j}(x)$ in geval $f^{(j+1)}$ niet al te veel varieert in het relevante gebied.

Hoe kunnen we zien of $f^{(j+1)}$ niet al te sterk varieert op het relevante gebied? Omdat de gedeelde differenties waarden zijn van afgeleiden, zouden de differenties ons hier inzicht in kunnen verschaffen. De volgende aanpak maakt gebruik van deze gedachte zonder de gedeelde differenties *expliciet* te berekenen. Volgens (3.4) is, voor $i \geq 3$,

$$V_{i,j} \equiv \frac{p_{i-2,j}(x) - p_{i-1,j}(x)}{p_{i-1,j}(x) - p_{i,j}(x)} \approx q_{i,j}, \quad \text{waarbij } q_{i,j} \equiv \frac{x - x_{i-1-j}}{x - x_{i-1}} \cdot \frac{x_{i-2-j} - x_{i-1}}{x_{i-1-j} - x_i},$$

als de betrokken differenties niet al te veel verschillen (dus, als $\alpha_{i-1,j+1} \approx \alpha_{i,j+1}$). De tabel van $V_{i,j}$'s kan vrijwel zonder extra kosten uit het Romberg schema opgesteld worden en kan ons inzicht verschaffen over de variatie in de $f^{(j+1)}$-waarden: als deze funktie waarden sterk variëren is het waarschijnlijk dat de differenties dat ook doen en zullen de $V_{i,j}$'s behoorlijk afwijken van de $q_{i,j}$. Omgekeerd, als deze grootheden redelijk samenvallen zal $f^{(j+1)}$ wel redelijk constant zijn op het relevante gebied.

Vraag 3C. Kunnen de $V_{i,j}$ ons in de praktijk inderdaad enige zekerheid verschaffen in de vraag of de schattingen $E_{i,j}$ van de fout in de $p_{i,j}(x)$?

Experiment. Vergelijk, voor de voorbeelden die je in Vraag 3B bekeken hebt, de exacte fout met de geschatte fout en ga na of de kwaliteit van de foutschatting samenhangt met hoe goed $V_{i,j}$ de waarde $q_{i,j}$ benadert: druk hiertoe de "relatieve vertrouwensgetallen" $V_{i,j}/q_{i,j}$ af.[69]

Conclusie. Als $V_{i,j} \approx q_{i,j}$ mogen we met enige zekerheid aannemen dat $f^{(j+1)}$ niet al te sterk varieert op het relevante gebied en kunnen we vertrouwen hebben in de schatting $E_{i,j}$ van de fout in de benadering $p_{i,j}(x)$ van $f(x)$. De $V_{i,j}$ worden daarom **vertrouwensgetallen** genoemd.

De uitdrukking in (3.5) vertelt ons dat het Lagrange interpolatiepolynoom p voor f op $(x_0, \ldots, x_k)$ geschreven kan worden als

$$\begin{aligned} p(x) = f(x_0) + (x - x_0)\frac{f[x_0, x_1]}{1!} + (x - x_0)(x - x_1)\frac{f[x_0, x_1, x_2]}{2!} + \ldots \\ + (x - x_0) \cdot \ldots \cdot (x - x_{n-1})\frac{f[x_0, \ldots, x_k]}{k!}, \end{aligned} \tag{3.8}$$

(Ga dit na). Deze voorstelling levert een alternatieve manier om $p(x)$ rekursief te berekenen middels het zogenaamde **schema van Horner**: met $\alpha_i \equiv f[x_0, \ldots, x_i]/i!$, bereken

$$S_0 \equiv \alpha_k, \quad \text{en} \quad S_{j+1} = (x - x_j)S_j + \alpha_{k-j-1} \text{ voor } j = 0, \ldots k - 1 \tag{3.9}$$

Dan $p(x) = S_k$ (Ga dat dit correct is).

Vraag 3D. Is het schema van Horner efficiënter dan dat van Romberg? Hoe vergelijken deze aanpakken zich met die in §2.4 van Hoofdstuk II? Kan je uit de resultaten van het schema van Horner de fout schatten?

Experiment. Meet de tijden de nodig zijn voor Horner en Romberg [70] en verklaar het verschil. Beantwoord de andere twee vragen zonder experimenten.

Verklaring. Tel het aantal vermenigvuldigingen in beide schema's.

6.4 Interpoleren met behulp van afgeleiden

SLEUTELBEGRIPPEN: *Lagrange interpolatie, Hermite interpolatie, Taylor reeks.*

Door stuksgewijze Lagrange interpolatie kunnen we f-waarden redelijk nauwkeurig benaderen. De grafiek van zo'n stuksgewijs polynoom heeft echter niet altijd de gladheid die we wensen. Door bijvoorbeeld ook te eisen dat in de steunpunten de afgeleide van het interpolerend polynoom en de afgeleide van de funktie f samenvallen kunnen we een "gladdere" interpolant krijgen. Deze gedachte lijkt aantrekkelijk, temeer daar we ook de $f'(t_i)$ tot onze beschikking hebben in geval de $f(t_i)$ waarden verkregen zijn als de numerieke oplossing van een gewone differentiaalvergelijking (zie Hoofdstuk I).

[69]Gebruik `vraag3b.m`. Het programma `Romberg.m` dat op de laatste regels van `vraag3b.m` wordt aangeroepen kan ook $V_{i,j}$ en de relatieve vertrouwensgetallen $V_{i,j}/q_{i,j}$ berekenen en de resultaten in een schema laten zien. Type `help Romberg` voor een beschrijving van de opties.

[70]Gebruik `vraag3d.m`.

Een polynoom p is het Hermite interpolatiepolynoom voor f op de punten t_i en t_{i+1} als p van minimale graad, en

$$p(t_j) = f(t_j) \quad \text{en} \quad p'(t_j) = f'(t_j) \quad \text{voor} \quad j = i, i+1.$$

Zie in dat p van graad ten hoogste 3 is: p is een **kubisch Hermite interpolatie polynoom**. Door gelijkheid te eisen in meer punten krijgen we hogere graads Hermite interpolatie polynomen.

Vraag 4A. Is stuksgewijs kubisch Hermite interpolatie nauwkeuriger dan kubische Lagrange interpolatie?

Experiment. Kies de punten t_i equidistant. Zie je de hogere graad van gladheid bij Hermite interpolatie terug in de plots? Bekijk de fout in Hermite interpolatie. De fout lijkt, voor x tussen t_i en t_{i+1}), min of meer evenredig met $(x-t_i)^2(x-t_{i+1})^2$. Was zoiets te verwachten?[71]

Verklaring. De restterm bij kubische Hermite interpolatie is van de vorm

$$(x-t_i)^2(x-t_{i+1})^2\tfrac{f^{(4)}(\xi)}{4!} \quad \text{voor zekere} \quad \xi \in (t_i, t_{i+1}),$$

(zie verderop), die bij Lagrange

$$(x-t_{i-1})(x-t_i)(x-t_{i+1})(x-t_{i+2})\tfrac{f^{(4)}(\xi)}{4!} \quad \text{voor zekere} \quad \xi \in (t_{i-1}, t_{i+2}).$$

Opgave. *Schat de maximale waarde van beide resttermen op het interval* (t_i, t_{i+1}).

Soms zijn ook funktiewaarden voor tweede of derde afgeleiden beschikbaar. We formuleren een algemene interpolatietheorie.

Bekijk een rij $\mathbf{x} \equiv (x_0, x_1, \ldots, x_k)$ van punten x_i in $[a, b]$ die zo genummerd zijn dat gelijke x_i na elkaar staan:

$$x_i = x_{i+j} \quad \text{voor} \quad j = 1, \ldots, d \quad \text{als} \quad x_i = x_{i+d}. \tag{4.1}$$

We zeggen dat $f = g$ **op de rij x** als voor iedere $i = 0, \ldots, k$ geldt

$$f(x_i) = g(x_i) \qquad \text{en} \qquad f^{(j)}(x_i) = g^{(j)}(x_i) \quad \text{voor iedere } j \text{ waarvoor} \quad x_i = x_{i+j}.$$

De funkties f en g moeten in de relevante punten x_i voldoende vaak differentieerbaar zijn.

Een polynoom p **interpoleert** f **op x** als $p = f$ op die rij en de graad van $p \leq k$ is.

Als de punten x_i onderling verschillend zijn (zeg $x_0 < x_1 < \ldots < x_k$) dan spreken we over **Lagrange interpolatie**. Is het aantal punten in de rij $\mathbf{x}$ even, $k = 2m+1$, en is $x_0 = x_1 < x_2 = x_3 < \ldots < x_{2m} = x_{2m+1}$ dan spreken we over **Hermite interpolatie** (dus voor het Hermite interpolatie polynoom p geldt $p(x_{2i}) = f(x_{2i})$ en $p'(x_{2i}) = f'(x_{2i})$ voor $i = 0, \ldots, m$). Vallen alle punten x_i samen ($x_0 = x_1 = x_2 = \ldots$) dan is het interpolatiepolynoom p precies het **Taylor polynoom**.

Het interpolatiepolynoom voor f op $\mathbf{x}$ is uniek en kan geschreven worden als in (3.8) ook in geval steunpunten samenvallen. In geval van samenvallende steunpunten

[71] Gebruik `vraag2a.m`.

zijn de differenties gedefinieerd als in (3.7) in §5.3. Als $f \in C^{(k+1)}([a,b])$, dan is er voor iedere x een ξ tussen $x_0, \ldots, x_k, x$ (met $\xi \approx \frac{x_0+x_1+\ldots+x_k+x}{k+2}$) zodat

$$f(x) = p(x) + (x - x_0) \cdot \ldots \cdot (x - x_k) \frac{f^{(k+1)}(\xi)}{(k+1)!}.$$

De representatie in (3.8) is flexibeler dan die in (6.4) en (6.5) in Hoofdstuk II: de graad kan gemakkelijk verhoogd worden en de hoeveelheid werk om polynoomwaarden uit te rekenen is aanzienlijk minder. Bovendien hebben we nu één theorie voor verschillende type polynoom benadering. We hebben immers dezelfde soort uitdrukkingen als voor het Taylor polynoom. De resultaten voor een Taylor polynoom zijn speciale gevallen.

Uit dit resultaat volgen eenvoudig de resultaten in Stelling 5.1:
Nummeren we $x_0, \ldots, x_k$ in een andere volgorde, maar met in achtname van conventie (4.1), dan hebben we hetzelfde interpolatie polynoom, en dus ook dezelfde leidende coëfficiënt.
Omdat $f - p = 0$ op $\mathbf{x}$, vertelt de stelling van Rolle (Lemma 6.1) ons dat $f^{(k)}(\xi) - p^{(k)}(\xi) = 0$ voor zekere ξ tussen $x_0, \ldots, x_k$. Omdat p van graad k is, is de k-de afgeleide gelijk aan $k!$ maal de leidende coëfficiënt, dus gelijk aan $f[x_0, \ldots, x_k]$.

6.5 Samenvatting

De rij $\mathbf{x} \equiv (x_0, x_1, \ldots, x_n)$ van punten x_i in $[a,b]$ is zo genummerd dat

$$x_i = x_{i+j} \quad \text{voor} \quad j = 1, \ldots, d \quad \text{als} \quad x_i = x_{i+d}. \tag{5.1}$$

We zeggen dat $f = g$ **op de rij** $\mathbf{x}$ als voor iedere $i = 0, \ldots, n$ geldt

$$f(x_i) = g(x_i) \qquad \text{en} \qquad f^{(j)}(x_i) = g^{(j)}(x_i) \quad \text{voor iedere } j \text{ waarvoor} \quad x_i = x_{i+j}.$$

De funkties f en g moeten in de relevante punten x_i voldoende vaak differentieerbaar zijn.

Het volgende lemma generaliseert de stelling van Rolle ($f'(\xi) = 0$ voor zekere $\xi \in (c,d)$ als $f(c) = f(d) = 0$). Het lemma speelt een belangrijke rol bij het bepalen van de vorm van de restterm.

Lemma 6.1 (Stelling van Rolle) *Als f n maal continu differentieerbaar is en $f = 0$ op $(x_0, x_1, \ldots, x_n)$, dan $f^{(n)}(\xi) = 0$ voor zekere ξ tussen $x_0, \ldots, x_n$.*

Bewijs. We mogen aannemen dat $x_0 \leq x_1 \leq \ldots \leq x_n$. Volgens de standaard stelling van Rolle zijn er y_i ($i = 0, \ldots, n-1$) zodat $x_i \leq y_i \leq x_{i+1}$ ($i = 0, \ldots, n-1$) en $f' = 0$ op $(y_0, \ldots, y_{n-1})$ (ga na dat de bewering ook korrekt is in geval een aantal x_i-en samenvallen). Herhaald toepassen van deze observatie (op achtereenvolgense f', f'', $\ldots$) levert de uitspraak van het lemma. □

Een polynoom p **interpoleert** f **op** $\mathbf{x}$ als $p = f$ op die rij en de graad van $p \leq n$ is.

Stelling 6.2 *Het interpolatiepolynoom voor f op $\mathbf{x}$ is uniek.*

Bewijs. Als p en q van graad $\leq n$ zijn en beide zijn gelijk aan f op $\mathbf{x}$ dan is $p - q = 0$ op $\mathbf{x}$ en $p(x) - q(x) = r(x)(x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_n)$ voor zeker polynoom r (Hoofdstelling van de Algebra). Omdat $p - q$ van graad $\leq n$ is en $(x - x_0) \cdots (x - x_n)$ van graad $n + 1$ kan dit alleen als $r = 0$ en dus is $p = q$. □

Als de punten x_i onderling verschillend zijn (zeg $x_0 < x_1 < \ldots < x_n$) dan spreken we over **Lagrange interpolatie**. Is het aantal punten x_i even, $n = 2m+1$, en is $x_0 = x_1 < x_2 = x_3 < \ldots < x_{2m} = x_{2m+1}$ dan spreken we over **Hermite interpolatie** (dus voor het Hermite interpolatie polynoom p geldt $p(x_{2i}) = f(x_{2i})$ en $p'(x_{2i}) = f'(x_{2i})$ voor $i = 0, \ldots, m$). Vallen alle punten x_i samen ($x_0 = x_1 = x_2 = \ldots$) dan is het interpolatiepolynoom p precies het **Taylor polynoom**.

Stelling 6.3 *Het interpolatiepolynoom p voor f op $\mathbf{x}$ wordt gegeven door*

$$p(x) = f(x_0) + (x - x_0)\frac{f[x_0, x_1]}{1!} + (x - x_0)(x - x_1)\frac{f[x_0, x_1, x_2]}{2!} + \ldots$$
$$+ (x - x_0) \cdot \ldots \cdot (x - x_{n-1})\frac{f[x_0, \ldots, x_n]}{n!}.$$

Bewijs. Bekijk de polynomen q_0 en q_1 die f interpoleren op, respektievelijk $(x_0, \ldots, x_{n-1})$ en $(x_1, \ldots, x_n)$. Dan is

$$p = \frac{x - x_n}{x_0 - x_n} q_0 + \frac{x - x_0}{x_n - x_0} q_1; \tag{5.2}$$

immers het polynoom rechts interpoleert f op $(x_0, x_1, \ldots, x_n, x_{n-1})$ (ga dat na door invullen van de x_i's) en het interpolatiepolynoom is uniek (Stelling 6.2). Nemen we aan dat de stelling korrekt is voor q_0 en q_1 (induktie aanname), dan weten we dat de leidende coëfficiënt van q_0 en q_1 gelijk is aan, respektievelijk, $f[x_0, \ldots, x_{n-1}]/(n-1)!$ en $f[x_1, \ldots, x_n]/(n-1)!$. Relatie (5.2) vertelt ons dat de leidende coëfficiënt van p gelijk is aan

$$\frac{f[x_0, \ldots, x_{n-1}]}{(x_0 - x_n)(n-1)!} + \frac{f[x_1, \ldots, x_n]}{(x_n - x_0)(n-1)!} = n\,\frac{f[x_0, \ldots, x_{n-1}] - f[x_1, \ldots, x_n]}{x_0 - x_n}\,\frac{1}{n!}.$$

Per definitie (zie (3.4) in §5.3) is dit $f[x_0, \ldots, x_n]/n!$. □

Uit voorgaande stelling volgen eenvoudig twee resultaten (Stelling5.1 in §5.5) die al eerder genoemd zijn.

Nummeren we $x_0, \ldots, x_n$ in een andere volgorde met in achtname van conventie (5.1), dan hebben we hetzelfde interpolatie polynoom, met dezelfde leidende coëfficiënt. Dus

Stelling 6.4 $f[x_0, \ldots, x_k] = f[x_{\pi(0)}, \ldots, x_{\pi(k)}]$ *voor iedere toelaatbare permutatie π van* $0, \ldots, k$. □

Omdat $f - p = 0$ op $\mathbf{x}$ voor het interpolatiepolynoom p, vertelt de stelling van Rolle (Lemma 6.1) ons dat $f^{(n)}(\xi) - p^{(n)}(\xi) = 0$ voor zekere ξ tussen $x_0, \ldots, x_n$. Omdat p van graad n is, is de n-de afgeleide gelijk aan $n!$ maal de leidende coëfficiënt, dus gelijk aan $f[x_0, \ldots, x_n]$. Blijkbaar geldt

Stelling 6.5 $f[x_0, \ldots, x_n] = f^{(n)}(\xi)$ *voor zekere ξ tussen $x_0, \ldots, x_n$.* □

Stelling 6.6 *Beschouw het interpolatiepolynoom p voor f op $\mathbf{x}$. Dan geldt voor iedere x dat*

$$f(x) = p(x) + (x - x_0) \cdot \ldots \cdot (x - x_n)\frac{f[x_0, \ldots, x_n, x]}{(n+1)!},$$

en, als f voldoende glad is, dan is er een ξ tussen $x_0, \ldots, x_n, x$ ($\xi \approx \frac{x_0 + \ldots + x_n + x}{n+2}$) zodat

$$f(x) = p(x) + (x - x_0) \cdot \ldots \cdot (x - x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

Bewijs. Kies een x_{n+1} in $[a,b]$. Schrijf $y = x_{n+1}$. Volgens Stelling 6.3 is

$$p(x) + (x - x_0) \cdot \ldots \cdot (x - x_n) \frac{f[x_0, \ldots, x_n, y]}{(n+1)!}$$

het interpolatiepolynoom van f op $(x_0, \ldots, x_n, y)$. Omdat het polynoom ook in y interpoleert geldt

$$f(y) = p(y) + (y - x_0) \cdot \ldots \cdot (y - x_n) \frac{f[x_0, \ldots, x_n, y]}{(n+1)!},$$

We hebben al gezien dat $f[x_0, \ldots, x_n, y] = f^{(n+1)}(\xi)$ (zie Stelling 5.1). Met $y = x_{n+1} = x$ is hiermee het bewijs geleverd. $\square$

De eerste gelijkheid in Stelling 6.6 is van belang omdat de uitdrukking voor de resterm geen (hogere) afgeleide van f bevat en dus toepasbaar ook als f niet glad is.

Kanttekeningen. Newton (1642-1727) gebruikte interpolatie om integralen te berekenen. Een aantal resultaten in deze sektie stammen van hem en dragen zijn naam. Stelling 6.3 is bewezen volgens de methode van Newton. De eerste gelijkheid in Stelling 6.6 is de gelijkheid van Newton. Merk op dat de Newton represenatie in Stelling 6.3 van het interpolatiepolynoom ouder is dan de representatie van Lagrange (1736–1813) in H.II, §2.4 en in H.II, (6.4) en (6.5).

7 Extrapolatie

SLEUTELBEGRIPPEN: *Extrapolatie, Romberg schema*

7.1 Inleiding

In de vorige sektie (§6) hebben we gezien dat interpolatie gebruikt kan worden om limieten uit te rekenen. Gewoonlijk ligt het punt waarin de limiet berekend moet worden buiten het gebied waarop waarden van de funktie bekend zijn. In zo'n geval spreekt men over **extrapolatie**. De funktie in (1.3) in de vorige sektie, waarvoor een limietwaarde bepaald moest worden, was expliciet gegeven. Vaak is dat niet het geval. Die problemen die hiermee samenhangen en algemeen met extrapolatie bestuderen we in deze sektie. Het volgende "klassieke" voorbeeld illusteert mooi met wat voor soort problemen we te maken kunnen krijgen.

Voorbeeld 1A. [π] De omtrek van een cirkel met straal een is, voor grote n, bij benadering gelijk aan de omtrek van een ingeschreven regelmatige 2^n-hoek (zie dat in). Dit levert een methode om π rekursief nauwkeurig te berekenen: de halve omtrek γ_n van een ingeschreven regelmatige 2^n-hoek kan berekend worden uit γ_{n-1}:

$$\gamma_n = 2^n \beta_n \quad \text{waarbij} \quad \beta_n = \sqrt{\tfrac{1}{2}\left(1 - \sqrt{1-\beta_{n-1}^2}\right)}, \quad \beta_1 = 1 \tag{1.1}$$

(Dit kan je nagaan door een tekening van de situatie te maken en te werken met gelijkvormige driehoeken en de stelling van Pythagoras. Je kan ook de verdubbelingsformule voor de sinus gebruiken. We hebben het lijnstuk ter lengte twee gezien als de ingeschreven regelmatige twee-hoek. Dit lijnstuk heeft halve omtrek twee: $\gamma_1 = 2$). Verder is

$$\lim_{n\to\infty} \gamma_n = \pi.$$

Door n maar groot genoeg te nemen kunnen we, met γ_n, π in principe in iedere gewenste precisie berekenen. De oude grieken hebben zo een paar cijfers van π bepaald. Het is echter niet aantrekkelijk om n groot te nemen: het rekenwerk neemt toe en afrondfouten gaan een grote rol spelen.

Vraag 1A. Ga experimenteel na wat de maximale nauwkeurigheid is die we met MATLAB met (1.1) kunnen halen?[72] Voor welke n hebben we de nauwkeurigste benadering? Voor $n \geq 30$ is γ_n voor 100% onbetrouwbaar. Hoe komt dat?
Verklaring. Bepaal de grootste δ waarvoor MATLAB $1 - \delta^2 = 1$ levert.

Christiaan Huygens (1629–1695) gebruikte eerste orde extrapolatie (d.w.z., hij gebruikte twee opeenvolgende waarden, γ_n en γ_{n-1}) om de 'griekse' benadering van π te verbeteren.

Vraag 1B. Had Huygens succes?
γ_n is een funktie van n. Kunnen we uit $\gamma_1, \ldots, \gamma_7$ met extrapolatie de waarde van γ in "∞" schatten? Waarom lukt extrapoleren 'naar ∞' toe niet met polynomen? Werkt

[72] Gebruik `vraag1a.m`.

extrapolatie wel als we γ_n zien als de waarde van een funktie in het punt $x_n = 1/n$?[73] De scalair γ_n is de halve omtrek van een regelmatige 2^n-hoek. Het is dus niet zo'n gekke gedachte om γ_n te zien als de waarde van een funktie in $x_n = 2^{-n}$. Hoe pakt extrapolatie nu uit? Hoe pakt een en ander uit als we γ_n zien als de waarde in $x_n = 4^{-n}$? In hoeveel cijfers nauwkeurig berekende Huygens π (als we aannemen dat hij $\gamma_1 \ldots$, γ_7 gebruikte)? Wat is de maximale nauwkeurigheid die Huygens gehaald zou kunnen hebben als hij nog meer γ_n's gebruikt zou hebben en met MATLAB had kunnen rekenen (d.w.z., als hij in iedere arithmetische operatie een relatieve fout van $2.2\,10^{-16}$ gemaakt zou hebben)?

Met hoeveel γ_n's kunnen we hetzelfde resultaat bereiken als Huygens als we ook hogere orde extrapolatie toepassen, d.w.z, als we een Romberg schema opstellen?

Bovenstaand voorbeeld is zeer instruktief. Het laat zien dat extrapolatie een nuttige techniek kan zijn om limieten uit te rekenen zelfs als er niet expliciet een 'onderliggende' funktie gegeven is. Het laat ook zien dat het niet altijd onmiddellijk duidelijk is of er een 'onderliggende' funktie is en, als die er is, hoe die samenhangt met de steunpunten. Verder hebben we gezien dat extrapolatie gebruikt kan worden om stabiliteitsproblemen te omzeilen en de efficiëntie te verhogen. Limietprocessen spelen een grote rol op diverse plekken, en extrapolatie kan een bijzonder krachtige techniek zijn.

De afgeleide van een funktie in een punt is volgens de formele definitie een limietwaarde en zou middels extrapolatie bepaald kunnen worden. Numeriek differentiëren is instabiel (zie Vraag 4G in §5.4) en de vraag is hoe instabiliteiten doorwerken in de extrapolatieresultaten. We onderzoeken deze vraag in §7.4. De integraal van een funktie is ook een limietwaarde. Numeriek integreren kan nogal 'duur' zijn en in §7.5 zullen we zien dat extrapolatie de efficiëntie aanzienlijk kan vergroten.

In de rest van deze sektie nemen we aan dat de limiet voor $n \to \infty$ uitgerekend moet worden van een rij (γ_n). De γ_n's zijn de waarde van een (onbekende) funktie in x_n met $x_n \to 0$ (of we proberen de γ_n als zodanig te zien). De γ_n's zijn berekenbaar, maar het berekenen van γ_n voor grotere n is bezwaarlijk omdat dat bijvoorbeeld te duur is of te onnauwkeurig.

7.2 Steunpunten voor Romberg extrapolatie

SLEUTELBEGRIPPEN: *Lineaire extrapolatie, vertrouwensgetallen, Romberg extrapolatie, asymptotische reeks.*

In het voorbeeld 1A was het exacte antwoord bekend, we konden de fouten exact berekenen en zien of we de steunpunten effektief gesitueerd hadden. In realistische toepassingen is het exacte antwoord niet bekend en zullen we een andere strategie moeten volgen om de steunpunten te kiezen.

Vraag 2A. Is alleen de distributie van de steunpunten van belang of ook de exacte positie (d.w.z., geeft $x_n = \kappa y_n$ voor alle relevante n, voor iedere scalair $\kappa \neq 0$ hetzelfde resultaat)?[74]

Verklaring. $x_n = y_n$ vervangen door $x_n = \kappa y_n$ is een schaling. De aanpak is niet afhankelijk van de schaling.

[73] Gebruik `vraag1b.m`.

[74] Gebruik `vraag2a.m` voor je experimenten.

Kunnen we *van te voren* inzien hoe de steunpunten gedistribueerd moeten worden? Dit is een analytische vraag en zal analytisch beantwoord moeten worden.

Opgave 2B. *Zie in dat de halve omtrek γ_n van een ingeschreven regelmatige 2^n-hoek in de eenheids cirkel gelijk is aan $\gamma_n = 2^n \sin(\pi 2^{-n})$. Dus met $g(t) \equiv \frac{\sin(\pi t)}{t}$ is $\gamma_n = g(2^{-n})$. Hierbij is γ_n als in Voorbeeld 1A. Ga na dat $g(t) = \pi(1 - t^2\frac{\pi^2}{3!} + t^4\frac{\pi^4}{5!} - \ldots)$. Dus met $f(t) \equiv \pi - t\frac{\pi^3}{3!} + t^2\frac{\pi^5}{5!} - \ldots$ is $\gamma_n = f(4^{-n})$. Verder is, voor iedere n en voor x voldoende klein, $f(x)$ ongeveer gelijk aan de waarde in x van het n-de graads polynoom $\pi - t\frac{\pi^3}{3!} + t^2\frac{\pi^5}{5!} - \ldots + (-t^n)\frac{\pi^{2n+1}}{(2n+1)!}$. Verklaar nu waarom de keuze $x_n = 4^{-n}$ in Voorbeeld 1A zo effektief is.*

Het is niet altijd duidelijk hoe je analytisch te werk moet gaan. Het zou prettig zijn als de getallen γ_n je een aanwijzing zouden kunnen geven of er een effektieve distributie is en hoe die er uit zou kunnen zien.

Vraag 2C. Kan je uit de waarden γ_n een indruk krijgen betreffende de 'optimale' distributie van de steunpunten?

Idee. Als de x_n zo liggen dat de γ_n de waarde zijn in de x_n van een 'bijna' lineaire funktie, dan werkt lineaire extrapolatie uitstekend. De vraag is hoe we kunnen achterhalen of er zo'n bijna lineaire funktie is. De **vertrouwensgetallen** V_n (zie Vraag 3C in §6.3), gegeven door

$$V_n \equiv \frac{\gamma_{n-2} - \gamma_{n-1}}{\gamma_{n-1} - \gamma_n},$$

kunnen ons een aanwijzing geven. Merk op dat de V_n vrijwel kosteloos te berekenen zijn uit de beschikbare γ_i.

Stel dat $\gamma_n = f(x_n)$ voor een bijna lineare funktie f, d.w.z. voor zekere (onbekende) scalairen α en β is $f(t) = \alpha + \beta t + \mathcal{O}(t^2)$ voor $|t| \to 0$.[75] Dan is, voor n voldoende groot,

$$V_n = \frac{f(x_{n-2}) - f(x_{n-1})}{f(x_{n-1}) - f(x_n)} \approx \frac{x_{n-2} - x_{n-1}}{x_{n-1} - x_n}.$$ [76]

De vertrouwensgetallen V_n zijn dus min of meer onafhankelijk van α en β en dat is fijn want α en β zijn onbekend — α is de limiet en die moesten we uitrekenen.

In realistische situaties kennen we de bijna lineaire funktie niet (we weten zelfs niet of hij bestaat), maar we kunnen wel proberen de x_n zo te kiezen dat $x_{n-2} - x_{n-1} \approx V_n(x_{n-1} - x_n)$ voor $n = 2, 3, \ldots$ en $x_n \to 0$ als $n \to \infty$. Als dit lukt dan past de distributie bij een bijna lineaire funktie.

Als bijvoorbeeld $V_n \to 1/C$ voor $n \to \infty$ ($V_n \approx 1/C$ voor 'grote' n) en $|C| < 1$ dan hebben we met $x_n = C^n$ zo'n distributie (Ga dat na).

Zie ook §7.3 voor een alternatieve, maar vergelijkbare aanpak.

Experiment. Probeer voor de zes voorbeelden in `example2.m` een geschikte distributie te schatten uit de vertrouwensgetallen en kijk of je distributie voor lineaire extrapolatie effektief is.[77]

[75] De argumenten zijn ook al korrekt als $f = \alpha + \beta t + \mathrm{o}(t)$ voor $|t| \to 0$. Hier betekent $f = g + \mathrm{o}(t)$ voor $|t| \to 0$ dat $|f(t) - g(t)|/|t| \to 0$ voor $|t| \to 0$. Merk op dat $f = \alpha + \beta t + \mathrm{o}(t)$ als $f(t) = \alpha + \beta t + \mathcal{O}(t^2)$.

[76] In deze sektie bedoelen we met $\alpha_n \approx \beta_n$ voor grote n dat $\alpha_n/\beta_n \to 1$ voor $n \to \infty$.

[77] De voorbeelden worden aangeroepen in `vraag2c.m`. Probeer de distributie te schatten zonder te kijken naar de definitie van de γ_n's in `example2.m`.

Als je, als in Vraag 2C, de x_n zo gedistribueerd hebt dat lineaire extrapolatie effektief is, dan weet je nog niet of die distributie ook effektief is voor hogere graads extrapolatie.

Als je op grond van analytische argumenten (als bijvoorbeeld in Opgave 2B) een 'optimale' distributie gekozen hebt dan weet je niet of die in de praktijk ook al 'effektief' is. Waarom niet?

Vraag 2D. Kan je achteraf checken of de gekozen distributie 'optimaal' was.

Idee. Als de relatieve vertrouwensgetallen $V_{i,j}/q_{i,j}$ van het Romberg schema (zie Vraag 3C in §6.3) ongeveer 1 zijn dan is de fout schatting in het Romberg schema betrouwbaar en is de korrektie met de geschatte fout effektief.

Experiment. Ga voor de eerste vier voorbeelden in `example2.m` na hoe een en ander uitpakt.[78] De relatieve vertrouwensgetallen zijn niet altijd ongeveer 1. Kan je dat verklaren? Voor één voorbeeld is het verstandig om toch maar een andere distributie te kiezen? Voor welke voorbeeld is dat het geval? Welke distributie is dan beter? Kan je verklaren waarom dat die beter is (kijk eventueel in `example2.m` hoe de γ_n's gedefinieerd zijn en maak een reeks ontwikkeling voor de definiërende funktie)?

Opgave. *Bewijs dat de relatieve vertrouwensgetallen $V_{i,j}/q_{i,j}$ van het Romberg schema (zie Vraag 3C in §6.3) bij een 'optimale' distributie naar 1 convergeren voor $i \to \infty$.*

Vraag 2E. Is er altijd één distributie die effektief is ook voor hoger graads extrapolatie?

Experiment. Kijk eens naar het vijfde en zesde (en eventueel het zevende) voorbeeld in `example2.m`.[78]

Verklaring. Kijk in `example2.m` hoe de γ_n's gedefinieerd zijn en maak een reeks ontwikkeling voor de definiërende funktie. Probeer een verband te leggen tussen de kolommen in het Romberg schema waarvoor de relatieve vertrouwensgetallen ongeveer 1 zijn en de reeks.

Experiment. Kijk weer eens naar het eerste voorbeeld en neem voor $x_n = 2^{-n}$ in plaats van $x_n = 4^{-n}$.[78] Sommige kolommen van het Romberg schema laten een effektieve verbetering zien ten opzichte van de resultaten in de dezelfde rij in de voorafgaande kolom. Welke kolommen zijn dat? Hoe komt dat?

Neem nu in het tweede voorbeeld in `example2.m` de `kappa=5` en kijk eens naar de resultaten in het Romberg schema voor de effektieve x_n ($x_n = 2^{-n}$). Waarom zijn niet alle Romberg 'korrekties' even effektief?

Conclusie. Extrapolatie met polynomen kan alleen werken als de funktie f die op de achtergrond een rol speelt goed te benaderen is met polynomen: voor de funktie f moeten er β_i's zijn zodat voor iedere n geldt

$$f(t) = \alpha + \beta_1 t + \ldots + \beta_n t^n + \mathcal{O}(t^{n+1}) \quad \text{voor} \quad t \to 0. \tag{2.1}$$

De funktie f die op de achtergrond een rol speelt is niet noodzakelijk de funktie waarmee de γ_n berekend zijn (zie bijvoorbeeld Opgave 2B). De reeks in (2.1) wordt een **asymptotische reeks** genoemd. De klasse van funkties met een asymptotische reeks is groter dan die van de funkties met een machtreeks.

Zijn er extrapolatiestrategiën waarmee je benaderingen effektief kunt verbeteren ook in geval er geen distributie is van de steunpunten die ook effektief is voor hoger graads extrapolatie? Als je lineair geëxtrapoleerd hebt krijg je, bij een geschikte distributie

[78] Gebruik `vraag2d.m`.

van de steunpunten, betere benaderingen. Je zou voor de nieuwe resultaten op zoek kunnen gaan naar een nieuwe distributie van de steunpunten die weer past bij lineaire extrapolatie en vervolgens zou je weer lineair kunnen extrapoleren, etc..

Vraag 2F. Werkt herhaald lineair extrapoleren en is het effektief?

Experiment. Kijk voor de voorbeelden vijf en zes in `example2.m` of herhalen van lineair extrapoleren effektief is.[78] Een keer herhalen is voldoende. Vergelijk het resultaat met het Romberg schema op de eerste set steunpunten.

Opgave. *Stel dat, voor $x_n = 4^{-n}$, $\gamma_n = f(x_n)$ voor een funktie f van de vorm*

$$f(t) = \alpha + \beta t + \gamma t^q.$$

Hierin zijn α, β en γ en q (onbekende) reële getallen en is $q > 1$.

a. *Ga na dat de waarde $T_{i,1} \equiv p_{i,1}(0)$ in 0 van het Lagrange interpolatiepolynoom $p_{i,1}$ voor f op (x_{i-1}, x_i) gelijk is aan $T_{i,1} = \alpha + \widetilde{\gamma} 4^{-iq}$ waarbij $\widetilde{\gamma} \equiv \gamma \frac{4-4^q}{4-1}$.*

b. *Laat zien dat $T_{i,1} = g(y_i)$ voor zekere lineaire funktie g met $g(0) = \alpha$ en zekere y_i. Hoe kies je de y_i? Lukt dit als de q onbekend is? Kan je q schatten uit de $V_{i,1}$, de vertrouwensgetallen bij de $T_{i,1}$?*

c. *Ga na dat, in geval $q = 2$, lineaire extrapolatie van g op (y_{i-1}, y_i) hetzelfde oplevert als tweede graads extrapolatie voor f op (x_{i-2}, x_{i-1}, x_i).*
Zie in dat herhaald lineair interpoleren in geval f is als in (2.1) *hetzelde is als een Romberg schema toepassen.*

d. *Zie in dat herhaald lineair extrapolatie mogelijk is, als f is als in* (2.1), *maar met niet gehele stijgende exponenten.*

Is iedere distributie stabiel?
Dit is een vraag naar het effekt van fouten op de γ_n's. Als we werken met $\gamma_n^\star = \gamma_n + \delta_n$ dan is de fout in $T_{i,j}$ tengevolge van de perturbaties δ_n op γ_n gelijk aan $(\Delta p)_{i,j}(0)$ waarbij $(\Delta p)_{i,j}$ het j-de graads polynoom is met waarde δ_k in x_k voor $k = i-j, \ldots, i$: $(\Delta p)_{i,j}$ is het "interpolatie polynoom" voor δ op $(x_{i-j}, \ldots, x_i)$. Hierbij is $T_{i,j}$ de waarde op plaats (i, j) in het Romberg schema voor de γ_n bij de gekozen steunpunten rij van x_n. Schrijf $\Delta_{i,j} \equiv (\Delta p)_{i,j}(0)$ en de stabiliteitsvraag laat zich nu bijvoorbeeld als volgt vertalen:

Vraag 2G. Hoeveel groter is $|\Delta_{i,j}|$ dan $\max_{n \le i} |\delta_n|$ en hangt dat af van de distributie van de steunpunten?

Experiment. Onderzoek hoe veel groter $|\Delta_{i,j}|$ kan worden dan $\max_n |\delta_n|$. Voor dit onderzoek is het geen beperking om δ_n zo kiezen dat $\max |\delta_n| = 1$. Waarom niet? Ga eerst experimenteel na dat de grootste waarden voor $|\Delta_{i,j}|$ optreden als $\delta_n = (-1)^n$.[79] Hoe hangt de 'vergrotings factor' af van de distributie van de steunpunten? Is het gunstig om een snel dalende of een langzaam dalende rij x_n'en te hebben?

Is de foutvoortplanting gunstiger als de $|\delta_n|$ systematisch kleiner worden, bijvoorbeeld als $|\delta_n| \le \overline{\delta} x_n$?

Opgave. *Laat, voor het geval $x_n = 4^{-n}$ en $\max_n |\delta_n| \le 1$, zien dat er een $C > 0$ is zodat $|\Delta_{i,j}| \le C$ voor alle i, j. Geef een schatting voor C.*

Opgave 2H. *In deze opgave geven we preciezere inhoud aan de verklaring in Vraag 2E.*

[79] Gebruik `vraag2g.m`.

Als f is als in (2.1) en $p_{i,j}$ is het Lagrange interpolatiepolynoom voor f op $(x_{i-j}, \ldots, x_i)$, bewijs dat dan geldt

$$p_{i,j}(0) = f(0) + \mathcal{O}(x_{i-j}^{j+1}) \quad \text{voor} \quad i \to \infty.$$

Je mag hierbij gebruiken dat er een $C > 0$ is zodat voor alle i, j $|\Delta_{i,j}| \leq C$ als $\max_n |\delta_n| \leq 1$ (zie de opgave in Vraag 2G).

In het voorafgaande in deze sektie hebben we geleerd hoe we extrapolatie met het Romberg schema kunnen gebruiken om de limiet van een rij getallen uit te rekenen. De techniek is uiteraard ook toepasbaar op een rij vektoren van gelijke dimensie. We moeten dan gewoon koördinaatsgewijs te werk gaan. De voorbeeldjes in `example2.m` waarmee we gewerkt hebben zijn nogal kunstmatig. In de volgende drie sekties zullen we realistische voorbeelden bekijken en zal blijken dat extrapolatie en met name Romberg extrapolatie een zeer krachtig numerieke techniek is.

7.3 Richardson extrapolatie

SLEUTELBEGRIPPEN: *Richardson extrapolatie, lineaire convergentie, convergentiefactor.*

In deze sektie kijken we naar een toepassing van lineaire extrapolatie. Voor deze toepassing hebben we overigens niet het volledig Romberg mechanisme nodig (zie Opgave 3B.a). We kijken naar lineair convergente rijen. De vertrouwensgetallen kunnen ons enige zekerheid geven of de rij lineair convergeert. Met dit soort rijen en hun relatie met vertrouwensgetallen zijn we overigens al vertrouwd (zie H.I,§1.3,Vraag3G, H.II,§**??**, en in dit hoofdstuk, §5.4). Hier plaatsen we alleen alles in een wat abstrakter kader en geven we een nieuwe toepassing.

De γ_n convergeren **lineair** naar α met **konvergentiefactor** C als $|C| < 1$ en

$$\gamma_{n+1} - \alpha \approx C(\gamma_n - \alpha) \quad \text{voor grote } n: \tag{3.1}$$[76]

de fout $\gamma_n - \alpha$ wordt bij iedere stap met ongeveer een faktor C gereduceerd (zie H.II§4.4). De α is onbekend (die willen we nu net berekenen) evenals de C. Echter als we weten dat de γ_n linear convergeren dan kunnen we de convergentiefactor schatten uit de vertrouwensgetallen, immers

$$V_n = \frac{\gamma_{n-2} - \gamma_{n-1}}{\gamma_{n-1} - \gamma_n} = \frac{(\gamma_{n-2} - \alpha) - (\gamma_{n-1} - \alpha)}{(\gamma_{n-1} - \alpha) - (\gamma_n - \alpha)} \approx \frac{(\gamma_{n-2} - \alpha)(1 - C)}{(\gamma_{n-1} - \alpha)(1 - C)} \approx \frac{1}{C}.$$

Anderzijds, als $V_n \to C$ voor $n \to \infty$ dan convergeren de γ_n linear en wel met convergentie factor C.

Opgave 3A. *Bewijs de laatste bewering.*

Als $V_n \approx C$ voor een aantal n'en dan is het redelijk om aan te nemen dat de γ_n lineair convergeren. Bovendien passen de γ_n met $x_n \equiv C^n$ bij een bijna lineaire funktie en mogen we met lineaire extrapolatie

$$T_{n,1} = \gamma_n + \frac{0 - C^n}{C^{n-1} - C^n}(\gamma_{n-1} - \gamma_n) = \gamma_n - \frac{C}{1 - C}(\gamma_{n-1} - \gamma_n)$$

een betere benadering van α verwachten. We spreken in deze toepassing van lineaire extrapolatie over **Richardson** extrapolatie.

Opgave 3B.
a. *Bewijs rechtstreeks uit* (3.1) *dat*

$$\gamma_{n+1} - \alpha \approx \frac{C}{C-1}(\gamma_{n+1} - \gamma_n).$$

b. *Ga na dat, als* $x_n = Cx_{n-1}$ *en* $\gamma_n = f(x_n)$ *voor* $f(t) = \alpha + \beta t + o(t)$*, dan is* $\gamma_n - \alpha \approx \beta x_n = C\beta x_{n-1} \approx C(\gamma_{n-1} - \alpha)$.

Vraag 3C. In H.II,§4.4 hebben we gezien dat successieve substitutie processen nogal eens lineair convergeren. In welke mate kan Richardson extrapolatie deze processen verbeteren?

Experiment. Kies x_0 bereken $\gamma_1 = \phi(x_0)$ en $\gamma_2 = \phi(\gamma_1)$. Pas Richardson extrapolatie toe op $\gamma_0 = x_0$, γ_1 en γ_2. Noem het resultaat x_1:

$$V = \frac{\gamma_0 - \gamma_1}{\gamma_1 - \gamma_2}, \qquad x_1 = \gamma_2 - \frac{\gamma_1 - \gamma_2}{V-1} = \frac{V\gamma_2 - \gamma_1}{V-1}.$$

Bereken $\gamma_1 = \phi(x_1)$, $\gamma_2 = \phi(\gamma_1)$. Extrapoleer weer op nieuwe $\gamma_0 = x_1$, γ_1 en γ_2, etc.. Dit proces heet het **Stephenson-Aitken** proces. Hoe snel convergeren de op deze manier geproduceerde x_n naar het vaste punt α van ϕ? Neem voor ϕ de funktie ϕ_1 in Vraag 4A van §4.4 in Hoofdstuk II. Bereken het vaste punt α met een betrouwbaarheid van 10^{-9}.[80] Probeer het vaste punt ook eens te berekenen met een betrouwbaarheid van 10^{-10}.[81] Hoe verklaar je de resultaten? Wat zijn de voor en nadelen van dit proces in vergelijking met die van het Newton proces?

Als het successieve substitutie proces divergeert dan zouden we door terugwaarts te extrapoleren dichter bij het vaste punt kunnen komen. Dus, als $|V| < 1$ dan bereken x_1 volgens

$$V = \frac{\gamma_0 - \gamma_1}{\gamma_1 - \gamma_2}, \qquad x_1 = \gamma_0 - \frac{1}{\frac{1}{V} - 1}(\gamma_1 - \gamma_0) = \frac{\gamma_0 - V\gamma_1}{1 - V}.$$

Etc.. Hoe pakt dit uit? Leidt dit tot een convergent proces? Probeer dit proces eens voor de varianten $\phi_2, \ldots, \phi_6$ van van ϕ in Vraag 4A van §4.4 in Hoofdstuk II die daar niet tot convergentie leidden.[82]

7.4 Numeriek differentiëren

Sleutelbegrippen: *Stabiliteit numeriek differentiëren*

In §5.4 hebben we gezien dat we stabiliteitsproblemen bij numeriek differentiëren gedeeltelijk kunnen omzeilen door met hogere orde differentie schema's te werken (zie vraag 4F in §5.4). De benaderingsfout in de gedeelde differentie benadering voor $g^{(k)}(c)$ op een equidistant verdeelde steunpunten rij is evenredig met h^2 (zie §5.3). Hierbij is h de afstand is tussen opeenvolgende steunpunten. Romberg extrapolatie (in ieder geval lineaire extrapolatie) lijkt toepasbaar en lijkt een mogelijkheid te bieden om met hogere orde (in h) differenties te werken zonder ze expliciet op te stellen.

[80] Gebruik `vraag3d.m`. Kies `tol=1.e-9` en `select=1`. In `example3.m` zijn verschillende funkties ϕ gedefinieerd. Met `select` kies je er een uit `example3.m`.

[81] Kies `tol=1.e-10`.

[82] Kies `select=2, ...`; zie `example3.m`. Het Stephenson-Aitken proces in `vraag3d.m` extrapoleert automatisch terugwaarts als $|V| < 1$.

Vraag 4A. Is Romberg extrapolatie op gedeelde differenties mogelijk en leidt het tot nauwkeurigere benaderingen voor de afgeleiden?

Experiment. Bekijk de funktie $g(x) = \exp(x)$. Zelfs voor zo'n mooie gladde funktie hadden we al problemen om $g^{(k)}(c)$ voor $k = 6$ (en $c = 0.1$) nauwkeurig uit te rekenen met gedeelte differenties (zie Vraag 4G in §5.3). Bepaal $g(x_i)$ voor x_i van de vorm $c + jh$ voor k even en $c + \frac{1}{2}h + jh$ voor k oneven en relevante j in $\mathbb{Z}$ (d.w.z. die j waarvoor $x_i \in [a, b]$): voor deze keuze van x_i is c telkens een geschikt "gemiddelde" van x_i waarden (zie Vraag 3C in §5.3). Kies een rijtje van opeenvolgende waarden voor h, bijvoorbeeld $h = 0.4, 0.2, 0.1, 0.05$. Bepaal, voor $g^{(k)}(c)$, de k-de gedeelde differentie voor $h_n = h(n)$ en noem deze differentie γ_n. Is Romberg extrapolatie mogelijk voor $g^{(k)}(c)$?[83] Kan je aan de vertrouwensgetallen zien waarom de keuze '$x_n = h_n^2$' geschikt is voor de steunpunten 'x_n' die bij de extrapolatie op de γ_n horen (zie §7.2: excuses voor het dubbele gebruik van de notatie x_n)? In welke andere punten $d \in [a, b]$ is, voor $g^{(k)}(d)$, op grond van de bepaalde differenties, ook nog Romberg extrapolatie mogelijk?[84] Als we de h's kleiner nemen worden de resultaten niet noodzakelijk beter. Hoe komt dat? Zie je dat ook terug in de vertrouwensgetallen? Is het de moeite waard om te werken met een alternatieve rij als $h = 0.4 \cdot [1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \ldots]$? Probeer een rijtje van h's te vinden die een zo 'optimaal' nauwkeurige benadering voor $g^{(k)}(c)$ levert. Hoeveel beter is het 'beste' Romberg resultaat dan het 'beste' resultaat zonder Romberg extrapolatie? Hoe groot is het effekt van afrondfouten in de g-waarden op de resultaten in het Romberg schema?[85] Is er een verband met de maximale nauwkeurigheid die je kon behalen?

Opgave.

a. *Bewijs dat er α, β_1, β_2, ... zijn zodat*

$$g[c - \tfrac{1}{2}h, c + \tfrac{1}{2}h] - g'(c) = \alpha + \beta_1 h^2 + \beta_2 h^4 + \ldots$$

als g voldoende glad is en h voldoende klein is.

b. *Verklaar waarom voor $g'(c)$ de keuze $x_n = h_n^2$ voor de steunpunten x_n voor Romberg extrapolatie op $\gamma_n \equiv g[c - \frac{1}{2}h_n, c + \frac{1}{2}h_n]$ effektief is.*

Soortgelijke resultaten gelden ook voor de k-de afgeleide en k-de gedeelde differenties. Het bewijs is echter veel meer gepriegel en dat proberen we hier dan ook niet uit te voeren.

De getallen op de diagonaal van het Romberg schema convergeren gewoonlijk zeer snel. Om deze reden wordt de fout in $T_{n,n}$ nogal eens geschat door

$$|T_{n,n} - \alpha| \leq |T_{n-1,n-1} - T_{n,n}|. \tag{4.1}$$

Vraag 4B. Is de schatting van de fout in (4.1) betrouwbaar in geval van numeriek differentiëren?

Experiment. Bekijk (de fout in) het Romberg schema voor de berekening van $g^{(k)}(c)$ met $g(x) = \exp(x)$, $c = 0.1$ en $k = 6$ en stapgrootte h achtereenvolgens 0.4, 0.2, 0.1, 0.05, 0.025, 0.0125 (zie Vraag 4A).[86]

Verklaring. De effecten van de afrondfouten worden groter bij numeriek differentiëren naarmate je 'dieper' in het Romberg schema komt. De afrondfouten bederven de convergentie.

[83] Gebruik `vraag4a.m`. Kijk naar de (relatieve) vertrouwensgetallen in het Romberg schema om te zien of de foutschatting in het Romberg schema te vertrouwen is. Vergelijk de schatting van de fout met de echte fout die we in dit speciale geval exact kunnen bepalen.

[84] Zie de door `vraag4a.m` geproduceerde figuren.

[85] Neem de optie `FoutAmpl=1` in `vraag4a.m`.

[86] Gebruik `vraag4a.m`.

7.5 Numeriek integreren

SLEUTELBEGRIPPEN: *Gerepeteerde trapezium regel, efficiëntie numeriek integreren, Romberg rij, Bulirsch rij, glijdende stap integratie.*

De gerepeteerde trapezium regel $T(h)$, met

$$T(h) \equiv h\left(\tfrac{1}{2}g(a) + [g(a+h) + g(a+2h) + \ldots + g(b-h)] + \tfrac{1}{2}g(b)\right),$$

wordt gebruikt om $\int_a^b g(t)\,dt$ numeriek te benaderen. In Vraag 2B in §3.2 van Hoofdstuk II hebben we gezien dat de fout hierin (in geval van een 'beschaafde' integrand g) evenredig is met h^2. Lineaire extrapolatie is mogelijk en we mogen hopen dat ook hogere orde extrapolatie effektief is. Overigens is $T(h)$ is alleen gedefinieerd voor specifieke waarde van h: de lengte $b-a$ van het integratie-interval $[a,b]$ moet een geheel veelvoud zijn van h. Het is dus niet duidelijk of $T(h)$ van de vorm $T(h) = \alpha + \beta_1 h^2 + \beta_2 h^4 + \ldots$ is, met $\alpha = \int g(t)\,dt$.

Vraag 5A. Is Romberg extrapolatie mogelijk? Biedt Romberg extrapolatie voordelen? Zo ja, welke zijn dat dan?

Experiment. Benader

$$\int_0^1 e^t\,dt \tag{5.1}$$

met de 2^n-maal gerepeteerde trapezium regel $T(h_n)$ voor $n = 0, 1, 2, \ldots$. Hierbij is $h_n = 2^{-n}$. Kijk naar de vertrouwensgetallen en bepaal aan de hand van deze getallen een rij steunpunten x_n in termen van de h_n waarvoor Romberg extrapolatie effektief is.[87]

Voor dit speciale geval kan je de exacte waarde van de integraal analytisch bepalen. Je kan dus ook de fout in de trapezium regel exact uitrekenen. Hoe goed is de foutschatting waarop lineaire interpolatie gebaseerd is? Is er een verband tussen de kwaliteit van de foutschatting en vertrouwensgetallen?

Is Romberg extrapolatie ook effektief? Kijk weer of de kwaliteit van de foutschatting gerelateerd is aan de (relatieve) vertrouwensgetallen.

Wat zijn de voordelen van Romberg extrapolatie hier? Hoeveel funktiewaarden $g(t_i)$ moeten er bepaald worden voor de gerepeteerde trapezium regel om een numerieke benadering voor de integraal te krijgen met een onbetrouwbaarheid $\leq 10^{-10}$. Hoeveel funktiewaarden heb je nodig als je lineair extrapoleert (tweede kolom van het Romberg schema)? Is de foutschatting voor deze geëxtrapoleerde resultaten betrouwbaar? Hoe pakt een en ander uit als je gebruik maakt van het hele Romberg schema?

Romberg extrapolatie is, in termen van de aantallen funktie-evaluaties, efficiënter dan niet extrapoleren, en ook dan alleen maar lineair extrapoleren. Zie je dat terug in de rekentijd?

Beantwoord de vragen ook eens voor

$$\frac{1}{\pi}\int_0^\pi \cos(y\sin(0.7\,t))\,dt \quad \text{voor} \quad y = [1:0.1:10] \tag{5.2}$$

Voor deze situatie is het niet doenlijk om voor alle $y(k) = 1 + 0.1\,k$ de bijbehorende Romberg schema's bekijken: het is echter al informatief genoeg om alleen naar de

[87] Gebruik `vraag5a.m`.

foutschattingen en vertrouwensgetallen te kijken voor $y = 1$.[88]
Kijk ook eens naar

$$\tfrac{1}{\pi}\int_0^{\pi} \cos(y\sin(t))\,dt \quad \text{voor} \quad y = [1 : 0.1 : 10].$$

Romberg extrapolatie lijkt hier overbodig. Is dit alleen het geval voor $y(1)$? Wat is je verklaring?
Onderzoek ook

$$\int_0^1 e^{\sqrt{t}}\,dt \quad \text{en} \quad \int_0^1 e^{\sqrt{1+t}}\,dt.$$

Kan je het verschillend gedrag verklaren?

Conclusie. Romberg extrapolatie kan, voor een gladde integrand, de efficiëntie van de 2^n-maal gerepeteerde trapezium regel aanzienlijk verbeteren.

Romberg extrapolatie leidt niet tot essentiële verbetering in geval de integrand g $(b-a)$-periodiek is.

Romberg extrapolatie werkt niet in geval $g'(a)$ of $g'(b)$ singulier zijn.

Opgave 5B. *Bewijs dat lineaire extrapolatie op de gerepeteerde trapezium regel resultaten met stapgroottes $h_n = 2^{-n}(b-a)$ tot precies dezelfde resultaten leidt als de regel van Simpson voor deze stapgroottes.*

In bovenstaande experimenten hebben we geen instabiliteiten gezien. We hebben al gezien dat het Romberg schema fouten op de γ_n niet al te zeer vergroot (zie Vraag 2G). Dus Romberg extrapolatie voor de gerepeteerde trapezium regel is stabiel als numeriek integreren met de gerepeteerde trapezium regel stabiel is met betrekking tot fouten op de g-waarden.

Opgave 5C. *Als $g^\star(t) = g(t) + \varepsilon(t)$ met $|\varepsilon(t)| \le \bar{\varepsilon}$ voor alle x, dan geldt*

$$\left|\int_a^b g(t)\,dt - \int_a^b g^\star(t)\,dt\right| \le \bar{\varepsilon} \quad \textit{en} \quad |T(h) - T^\star(h)| \le \bar{\varepsilon}$$

Bewijs dit en zie in dat numeriek integreren met de gerepeteerde trapezium regel stabiel is met betrekking tot fouten op de g-waarden.

De winst in efficiëntie in het Romberg schema komt uit de "breedte" en niet uit de "hoogte". Met een andere rij stapgroottes (bv. $h_n = (b-a)/n$ in plaats van $h_n = (b-a)/2^n$) kunnen we ook brede Romberg schema's tegen kosten die aanzienlijk lager zijn. Als de Romberg extrapolatie resultaten (de $T(n,n)$) ook nauwkeurig zijn, kunnen we de efficiëntie nog meer verhogen.

Vraag 5D. Kunnen we numerieke integratie met de gerepeteerde trapezium regel en Romberg extrapolatie verbeteren door met andere rijen van stapgroottes te werken?

Experiment. Hoe pakt een en ander uit voor de rij $h_n = (b-a)/n$ en de integralen in (5.1) en (5.2)?

De rij $((b-a)/n)$ van stapgroottes zakt wellicht te langzaam. Bulirsch stelde daarom de rij

$$h_0 \equiv b-a,\ h_1 \equiv \tfrac{1}{2}h_0,\ h_2 \equiv \tfrac{1}{3}h_0, \quad h_{2j+1} \equiv 2^{-j}h_1,\ h_{2j+2} \equiv 2^{-j}h_2, \quad j = 1, 2, \ldots$$

[88] Dit doet `vraag5a.m` automatisch: de schema's worden alleen voor $y(1)$ vertoont, de berekening van het (n,n)-de element (n maximaal) in het Romberg schema wordt wel uitgevoerd voor alle $y(k)$.

voor. Kijk ook eens hoe Romberg extrapolatie uitpakt voor deze **Bulirsch rij**. Tel zorgvuldig hoeveel funkties je moet evalueren in iedere aanpak; bedenk daarbij dat je wellicht waarden $T(h_j)$ voor $j < n$ kunt gebruiken in de berekening van $T(h_n)$.

Conclusie. De efficiëntie kan vergroot worden door met een rij staproottes te werken die minder snel naar nul daalt dan $h_n = 2^{-n}(b-a)$.

MATLAB heeft twee ingebouwde integratie routines `quad.m` en `quad8.m`. De eerste routine gebruikt de regel van Simpson en schat daarin de fout door de stapgrootte te halveren (de fout in de Simpson regel neemt met een factor 16 af bij stapgroottehalvering). Als de geschatte fout kleiner is dan de verlangde toleratie dan wordt het resultaat aanvaard. In het andere geval wordt het integratie-interval in twee gelijke delen gesplitst en op iedere van de twee deelintervallen wordt de aanpak (Simpson + foutschatten + acceptatie/verwerping) met gehalveerde toleratie herhaald, etc.. Er wordt niet met één stapgrootte gewerkt. De stapgrootte die uiteindelijk tot een aanvaardbaar resultaat leidt verschilt per plek. De stapgrootte kan in sommige delen groot zijn en in andere klein. Deze automatische integratie aanpak wordt **glijdende stap integratie** genoemd.

Vraag 5E. Haalt MATLAB met `quad.m` altijd de gewenste precisie?

Experiment. Integreer eens de funktie $g(t) \equiv e^{t/4} + e^{1-400t^2}$ over het interval $[-1, 4]$ met `quad.m` met achtereenvolgens tolerantie 10^{-3}, 10^{-4} en 10^{-5}.[89] Wordt de toleratie gehaald?[90] Kan je verklaren wat er gebeurd?

Vraag 5F. Zijn de ingebouwde MATLAB routines efficiënter dan de gerepeteerde trapezium regel met Romberg extrapolatie? Hangt dit nog af van het type integrand? Voor welke type integrand is "Romberg" beter?

Experiment. Vergelijk de twee methoden met elkaar voor de berekening van achtereenvolgens

$$\int_0^1 e^t\,dt, \quad \tfrac{1}{\pi}\int_0^\pi \cos(0.7\sin(t))dt \quad \text{en} \quad \int_{-1}^4 e^{t/4} + e^{1-400t^2}dt.$$

Probeer voor beide methoden een nauwkeurigheid van 10^{-10} te halen met zo weinig mogelijk funktie-evaluaties (kies eventueel een toleratie groter dan 10^{-10} als `quad.m` voor die grotere tolerantie al een voldoend nauwkeurig resultaat geeft. Kies voor de 'Romberg' aanpak het aantal stapgrootte halvering zo klein mogelijk. Let er op of de foutschattingen te vertrouwen zijn. Vergelijk eventueel het 'Romberg' resultaat met het `quad.m` resultaat).[91]

Verklaring. In de 'Romberg' aanpak vereisen de gerepeteerde trapezium regels een uniforme verdeling van de steunpunten. Als de grafiek van de funktie een piek heeft, dan bepaalt die piek de stapgrootte. De stapgrootte zal klein zijn. In de glijdende stap aanpak zal de stapgrootte voor de piek ook klein zijn, maar buiten de piek kan die veel groter zijn.

[89] Je kan `vraag5d.m` gebruiken. De funktie f is gedefinieerd in `example5.m` en wordt met `select=6` gekozen in `vraag5d.m`. In `figure 1` staan de funtiewaarden geplot die MATLAB in zijn automatische integratie procedure gebruikt heeft. Je kan hierin zien waarde de stapgrootte groter zijn en waar kleiner.

[90] Vergelijk het `quad.m` resultaat met dat van de gerepeteerde trapezium regel met Romberg extrapolatie: dit gebeurt automatisch in `vraag5d.m`.

[91] Je kan weer `vraag5d.m` gebruiken. Kies `trace=0` om een enigszins redelijke tijdschatting voor beide methoden te maken.

Conclusie. Als de grafiek van de funktie geen lokale pieken heeft is de 'Romberg' aanpak zeer efficiënt. Voor een funktie met een grafiek met één of een paar pieken kan de glijdende stap aanpak aantrekkelijk zijn.

7.6 Samenvatting

In deze sektie is (γ_n) een rij getallen die naar α convergeert. De γ_n zijn (voor de kleinere n) beschikbaar of ze zijn efficiënt te berekenen. De limiet α moet bepaald worden.

7.6.1 Lineaire convergentie.

De rij (γ_n) convergeert **lineair** naar α als er een getal C is, $|C| < 1$, zodat

$$\gamma_{n+1} - \alpha \approx C\,(\gamma_n - \alpha) \quad \text{voor grote } n.^{92} \tag{6.1}$$

Dus in geval van lineaire convergentie neemt de fout op den duur ongeveer af met een factor C: C wordt de **convergentie** **factor** genoemd.

De **vertrouwensgetallen** V_n, gedefinieerd door

$$V_n \equiv \frac{\gamma_{n-2} - \gamma_{n-1}}{\gamma_{n-1} - \gamma_n} \quad \text{voor} \quad n \geq 2, \tag{6.2}$$

kunnen ons vertellen of de rij (γ_n) lineair convergeert:

Stelling 7.1 *Voor* $|C| < 1$ *zijn de volgende twee uitspraken equivalent:*
(a) *de rij* (γ_n) *convergeert lineair met convergentiefactor* C*;*
(b) *de rij* (V_n) *convergeert naar* $\frac{1}{C}$.
Als (a) *geldt dan*

$$\gamma_n - \alpha \approx \frac{C}{1-C}\,(\gamma_{n-1} - \gamma_n) \quad \textit{voor grote } n. \qquad \square \tag{6.3}$$

De foutschatting in (6.3) kan ook gebruikt worden om de benadering γ_n van α te verbeteren:

$$\alpha \approx T_{n,1} \equiv \gamma_n - \frac{C}{1-C}\,(\gamma_{n-1} - \gamma_n) \quad \text{voor grote } n. \tag{6.4}$$

7.6.2 Romberg extrapolatie

Stel dat er een funktie f is en een rij (x_n) zodat $\gamma_n = f(x_n)$ voor iedere n en stel dat er scalairen $\beta_1, \ldots, \beta_{m-1}$ zijn zodat

$$f(t) = \alpha + \beta_1 t + \beta_2 t^2 + \cdots + \beta_{m-1} t^{m-1} + \mathcal{O}(t^m) \quad \text{voor} \quad t \to 0. \tag{6.5}$$

Als de rij (x_n) bekend is dan is Romberg extrapolatie effektief tot en met de m-de kolom (d.w.z. $m-1$ extrapolatie slagen zijn effektief: zie Stelling 7.2). De funktie f en de scalairen β_i hoeven hiervoor niet expliciet bekend zijn: we hoeven alleen te weten dat ze er zijn, niet welke waarde ze hebben. Vaak kan het bestaan van zo'n f op grond van analytische argumenten aangetoond worden (zie bijvoorbeeld §7.6.3).

[92] Met $\alpha_n \approx \beta_n$ voor grote n bedoelen we dat $\lim_{n\to\infty} \alpha_n/\beta_n = 1$.

Stelling 7.2 *Voor $|C| < 1$ zijn de volgende twee uitspraken equivalent:*
(a) *de rij (γ_n) convergeert lineair naar α met convergentiefactor C;*
(b) *de rij (x_n) convergeert lineair naar 0 met convergentiefactor C.*

Laat $T_{n,j}$ het (n,j)-de getal zijn in het Romberg schema voor f op $(x_0, x_1, \ldots)$ in $x = 0$: $T_{n,j} = p_{n,j}(0)$ waarbij $p_{n,j}$ het interpolatiepolynoom is voor f op $(x_{n-j}, \ldots, x_n)$. Als (b) *geldt dan convergeert, voor iedere $j = 0, \ldots, m-1$, de rij $(T_{n,j})$ lineair naar α met convergentiefactor C^{j+1}.* □

Als de γ_n lineair convergeren dan kan de convergentiefactor C geschat worden uit de vertrouwensgetallen voor (γ_n) (zie (6.2) en Stelling 7.1). Hieruit kunnen de effektieve steunpunten x_n voor de Romberg extrapolatie geschat worden: $x_n = C^n$. Iedere volgende kolom in het Romberg schema convergeert lineair maar met een convergentiefactor die een factor C kleiner is dan die van de voorafgaande kolom.

Vaak weten we niet
(i) of er een f is met de geschikte eigenschap (6.5),
(ii) of de x_n-en al voldoende klein zijn en
(iii) of de effecten van afrondfouten het geschikte convergentiegedrag bedorven hebben.
Met behulp van de vertrouwensgetallen kunnen we checken of het gedrag van de getallen in de $j+1$-ste kolom van het Romberg schema past bij lineaire convergentie met factor C^{j+1}.

In geval de γ_n lineair convergeren en (6.5) geldt, dan kunnen we het Romberg schema ook vormen door op iedere kolom lineaire extrapolatie toe te passen als in (6.4) met uiteraard de bijbehorende convergentiefactor C^{j+1} (zie Stelling 7.2):

$$T_{n,0} = \gamma_n, \qquad T_{n,j+1} \equiv T_{n,j} - \frac{C^{j+1}}{1 - C^{j+1}} \left(T_{n-1,j} - T_{n,j}\right). \tag{6.6}$$

Als (6.5) geldt is Romberg extrapolatie effektief ook als de x_n niet lineair convergeren. In geval de x_n niet lineair convergeren is het lastig om, als in de vorige alinea, Romberg extrapolatie te zien als herhaald lineaire extrapolatie. Met name voor deze situatie hadden we behoefte aan de ingewikkeldere Romberg aanpak en konden we niet volstaan met lineaire extrapolatie.

Als (6.5) geldt voor iedere m dan zeggen we dat f een **asymptotische reeks** (in t) heeft. In zo'n geval convergeren de diagonaal elementen $T_{n,n}$ van het Romberg schema **super-lineair** naar α, d.w.z.

$$\lim_{n\to} \frac{T_{n+1,n+1} - \alpha}{T_{n,n} - \alpha} = 0.$$

Super-lineaire convergentie wil dus zeggen dat de fout steeds 'sneller' afneemt.

Lineaire extrapolatie is ook effectief als in de "reeks" voor f gebroken machten voorkomen ($f(t) = \alpha + \beta_1 t + \beta_2 t^{1+\delta_2} + \cdots$), terwijl hogere graads Romberg extrapolatie dat dan niet is. Herhaald lineaire extrapolatie is wel effektief. Voor zo'n situatie is het lastig om voor iedere kolom de geschikte steunpunten x_n te vinden. Vaak echter geven de vertrouwensgetallen de gewenste informatie.

Romberg extrapolatie is stabiel tenzij de steunpunten x_n toevallig heel beroerd liggen:

Stelling 7.3 *Stel dat (x_n) lineair naar 0 convergeert met convergentiefactor C. Dan is Romberg extrapolatie stabiel: als $|\gamma_n^* - \gamma_n| \leq \overline{\varepsilon}$ voor iedere $n \leq m$ dan*

$$|T^\star_{n,j} - T_{n,j}| \leq \frac{1+C}{1-C} \cdot \frac{1+C^2}{1-C^2} \cdot \ldots \cdot \frac{1+C^j}{1-C^j} \cdot \overline{\varepsilon} \leq D \cdot \overline{\varepsilon} \quad \text{voor iedere } n, j \leq m. \tag{6.7}$$

Hierbij is $D \equiv \exp\left(\frac{2C}{(1-C)^2}\right)$.

Bewijs. Door (6.6) te gebruiken volgt dat

$$\begin{aligned} |T^\star_{n,j+1} - T_{n,j+1}| &\leq \frac{1}{1-C^{j+1}} |T^\star_{n,j} - T_{n,j}| + \frac{C^{j+1}}{1-C^{j+1}} |T^\star_{n-1,j} - T_{n-1,j}| \\ &\leq \frac{1+C^{j+1}}{1-C^{j+1}} \max_{i \leq n} |T^\star_{i,j} - T_{i,j}|. \end{aligned}$$

Nu levert een induktie argument de eerste ongelijkheid in (6.7). De tweede ongelijkheid volgt uit de schatting

$$\frac{1+h}{1-h} = 1 + \frac{2h}{1-h} \leq \exp(\tfrac{2h}{1-h}) \quad \text{voor} \quad 0 \leq h < 1. \qquad \square$$

Voor, bijvoorbeeld, $C = \frac{1}{4}$ is $D = e^{8/9} \leq e$. De fouten in de 'input', d.w.z. de fouten in de betrokken γ_n worden dan met hoogstens een factor e vergroot door de extrapolatie. Uiteraard zullen in het Romberg resultaat grote fouten zitten als de input met grote fouten belast zijn, maar die fouten zijn dan niet te wijten aan de extrapolatieprocedure.

Ook als voor een specifieke situatie het Romberg schema niet effektief is dan kan het in het algemeen geen kwaad om toch Romberg extrapolatie toe te passen: geen van de fouten, de benaderingsfout nog de afrondfout, wordt in het algemeen in volgende kolom veel groter, immers

$$|T^\star_{n,j} - \alpha| \leq \frac{1}{1-C^j} |T^\star_{n,j-1} - \alpha| + C^j |T^\star_{n-1,j-1} - \alpha|.$$

Dit argument wordt ook gebruikt om te betogen dat de convergentie langs de diagonaal van het Romberg schema sneller is dat de convergentie in wat voor kolom dan ook.

Vaak schat men de fout in het Romberg schema door

$$|T_{n,n} - \alpha| \leq |T_{n-1,n-1} - T_{n,n}|.$$

Vaak is deze schatting correct, maar de wiskundige rechtvaardiging is niet heel sterk. Betrouwbaarder is het om de fout in de kolommen te schatten en eerst te checken of de relevante vertrouwensgetallen passen bij het correcte foutgedrag. Als bijvoorbeeld in de $j+1$-ste kolom een of een paar relevante vertrouwensgetallen passen bij lineaire convergentie met convergentiefactor C^{j+1} dan lijkt het veilig de fout te schatten door

$$|T_{n,j} - \alpha| \leq \frac{C^{j+1}}{1-C^{j+1}} |T_{n-1,j} - T_{n,j}|.$$

Overigens is deze aanpak en geen enkele andere, die gebaseerd is op een 'eindige' hoeveelheid informatie, "waterdicht". Er is immers geen enkele garantie dat, als bijvoorbeeld de eerste 100 γ_n gelijk zijn aan 0, de limiet dat ook is. Met de aanpak met behulp van vertrouwensgetallen vergroten we alleen de waarschijnlijkheid dat we correcte schattingen uitvoeren.

7.6.3 Voorbeelden

We bespreken nu een tweetal voorbeelden waarin we op grond van analytische argumenten weten dat er een funktie f is als in (6.5) waarvoor $\gamma_n = f(x_n)$.

Numeriek differentiëren. Met behulp van Taylor reeksen volgt de volgende stelling.

Stelling 7.4 *Voor een funktie g die voldoende glad is in c (d.w.z. waarvoor de relevante afgeleiden bestaan) geldt*

$$T(h) \equiv \frac{g(c+h)-g(c-h)}{2h} = g'(c) + \frac{g^{(3)}(c)}{3!}h^2 + \frac{g^{(5)}(c)}{5!}h^4 + \ldots . \qquad \square$$

De funktie T heeft dus een asymptotische ontwikkeling in h^2. Dus als $\gamma_n \equiv T(h_n)$ voor een zekere rij (h_n) van stapgroottes dan is Romberg extrapolatie effektief voor $x_n = h_n^2$: $\gamma_n = f(x_n)$ voor f als in (6.5) met $\alpha = g'(c)$, $\beta_i = \frac{g^{(2i+1)}(c)}{(2i+1)!}$.

Voor grote h kunnen we al een nauwkeurige afgeleide berekenen met Romberg extrapolatie. Voor grote h zijn de invloeden van afrondfouten gering. Dit verklaart waarom we met behulp van Romberg extrapolatie nauwkeurig kunnen differentiëren.

Numeriek integreren. Voor een funktie g benaderen we, voor $h = (b-a)/n$, de integraal $\int_a^b g(t)\,dt$ met de n maal gerepeteerde trapezium regel

$$T(h) \equiv h\left(\tfrac{1}{2}g(a) + [g(a+h) + g(a+2h) + \ldots + g(b-h)] + \tfrac{1}{2}g(b)\right).$$

Stelling 7.5 *Er zijn constanten c_i zodat voor iedere g die voldoende glad is en voor iedere toegelate h geldt*

$$\int_a^b g(t)dt = T(h) + c_1 h^2[g^{(1)}(b) - g^{(1)}(a)] + c_2 h^4[g^{(3)}(b) - g^{(3)}(a)] + \ldots . \qquad (6.8)$$

Bewijs. Laten we eerst even aannemen dat (6.8) correct is voor het speciale geval waarin $a = 0$ en $b = h$ (dus $n = 1$): er constanten $c_1, c_2, \ldots$ zijn zodat

$$\begin{aligned}\textstyle\int_0^h g(t)dt = \tfrac{1}{2}h[g(h)+g(0)] + \\ c_1 h^2[g^{(1)}(h) - g^{(1)}(0)] + c_2 h^4[g^{(3)}(h) - g^{(3)}(0)] + \ldots .\end{aligned} \qquad (6.9)$$

Door het integratie-interval te verschuiven, volgt hier eenvoudig uit dat

$$\int_{t_i}^{t_i+h} g(t)dt = \tfrac{1}{2}h[g(t_i+h) + g(t_i)] + c_1 h^2[g^{(1)}(t_i+h) - g^{(1)}(t_i)] + \ldots .$$

Omdat, voor $h = (b-a)/n$, $\int_a^b g(t)dt = \int_a^{a+h} g(t)dt + \int_{a+h}^{a+2h} g(t)dt + \ldots$ volgt (6.8).

Rest ons het speciale geval (6.9) te bewijzen. Het is geen beperking $h = 1$ te nemen: vervang anders g door $\widetilde{g}(s) \equiv g(sh) = g(t)$.
Door partiële integratie volgt dat $\int_0^1 g(t)\,dt = \frac{1}{2}[g(1) + g(0)] + \int_0^1 (t - \frac{1}{2})g'(t)\,dt$. Door partiële integratie te herhalen volgt de rest van het bewijs. De (constanten in de) polynomen moeten wel met zorg gekozen worden: de polynomen $p_1 \equiv t - \frac{1}{2}$, p_2, $p_3, \ldots$, met p_i van graad i, $p'_{i+1} = p_i$ moeten zo zijn dat $p_i(0) = p_i(1)$ voor iedere $i > 1$. Dan

volgt dat $p_i(t) = (-1)^i p(1-t)$ voor idere $i \geq 1$ en i.h.b., $p_{2i+1}(0) = p_{2i+1}(1) = 0$ voor alle i. Dus

$$\int_0^1 p_{2i}(t) g^{(2i)}(t)\,dt = -\int_0^1 p_{2i+1} g^{(2i+1)}(t)\,dt =$$
$$= -p_{2i+2}(0)[g^{(2i+1)}(1) - g^{(2i+1)}(0)] + \int_0^1 p_{2i+2}(t) g^{(2i+2)}(t)\,dt. \qquad \square$$

De constanten c_i zijn uiteraard onafhankelijk van h, maar ook van g en van a en b: de afhankelijkheid van g, a en b zit volledig in de termen $[g^{(2i-1)}(b) - g^{(2i-1)}(a)]$. De reeks in (6.8) wordt de **Euler-MacLaurin** reeks genoemd. De polynomen in het bewijs zijn de zogenaamde Bernoulli polynomen. Door in (6.9) voor $g(t) = \exp(t)$ te nemen zien we dat de Euler-MacLaurin coëfficiënten c_i de coëfficiënten zijn van de Tayler reeks van $\phi(t) \equiv -\frac{t}{e^t-1}$.

Het resultaat in (6.9) laat zien dat, met $x_n = h_n^2$, Romberg extrapolatie op $T(h_n)$ effektief is als de integrand g in a en in b voldoende glad is, immers: $T(h_n) = f(x_n)$ voor f als in (6.5) met $\alpha = \int_a^b g(t)\,dt$, $\beta_i = -c_i[g^{(2i-1)}(b) - g^{(2i-1)}(a)]$. Dit is overigens precies de situatie waarvoor W. Romberg in 1955 zijn schema introduceerde. Het verband met extrapolatie in de algemene context (Aitken-Neville schema; zie §6.3) is pas later gelegd door R. Bulirsch (in 1964; [**?**]).

Als $g^{(2i+1)}(b) = g^{(2i+1)}(a)$ voor iedere $i = 0, 1, 2, \ldots$ dan is de gerepeteerde trapezium regel *zonder* Romberg extrapolatie net zo effektief als met Romberg extrapolatie. De oneven afgeleiden van de integrand g vallen in a en b samen als bijvoorbeeld g $(b-a)$-periodiek is.

Als bijvoorbeeld $g'(a)$ singulier is dan is Romberg extrapolatie met $x_n = h_n^2$ niet effektief. In zo'n geval kan een andere keuze voor de Romberg steunpunten x_n uitkomst bieden (denk aan $x_n = h_n\sqrt{h_n}$). Helaas loopt men vaak met een andere keuze voor de steunpunten aan tegen problemen in de volgende kolommen van het Romberg schema. Het is daarom handiger om de singulariteit uit de integrand met analytische middelen weg te transformeren en pas dan Romberg extrapolatie toe te passen.

Omdat $T(h) = \frac{1}{2}T(2h) + h[f(a+h) + f(a+3h) + \ldots]$ kost het opstellen van het Romberg schema op $T(h_0), \ldots, T(h_n)$ met $h_0 = b - a$ en $h_j = 2^{-j}h_0$ vrijwel evenveel als het evalueren van T_n. De extra kosten zijn wat optellingen en delingen.

De convergentie langs de diagonaal van het Romberg schema is gewoonlijk zeer snel en het is effektiever om een 'breed' Romberg schema te maken dan om met kleine stapgrootte h te werken. Deze gedachte ligt grondslag aan het idee van de **Bulirsch rij** waarin $h_0 = b - a$, $h_1 = \frac{1}{2}h_0$, $h_2 = \frac{1}{3}h_0$ en verder $h_{2i+1} = \frac{1}{2}h_{2i-1}$, $h_{2i+2} = \frac{1}{2}h_{2i}$. Het blijkt dat vaak het resultaat $T_{n,n}$ voor de Bulirsch rij nauwkeuriger is dan het resultaat $T_{n-1,n-1}$ voor de **Romberg rij** waarin $h_i \equiv 2^{-i}(b-a)$. Voor het berekenen van $T_{n-1,n-1}$ voor de Romberg rij moeten $2^{n-1}+1$ funktiewaarden geëvalueerd worden, terwijl voor $T_{n,n}$ voor de Bulirsch rij dat aantal slechts $2 \cdot 2^i + 1$ is als $n = 2i$ en $3 \cdot 2^i + 1$ als $n = 2i+1$. Dus om bijvoorbeeld $T_{9,9}$ voor de Romberg rij te berekenen moeten er 513 funktiewaarden berekend worden terwijl $T_{10,10}$ voor de Bulirsch rij de evaluatie van slechts 65 funktiewaarden kost.