De Methode van Heun voor Numerieke Oplossingen van Differentiaalvergelijkingen

Ziaad Negmel-Din, Thijmen Batelaan

2 februari 2025

Numerical Recipes Project

Begeleiding: Gielens, S.P.N.

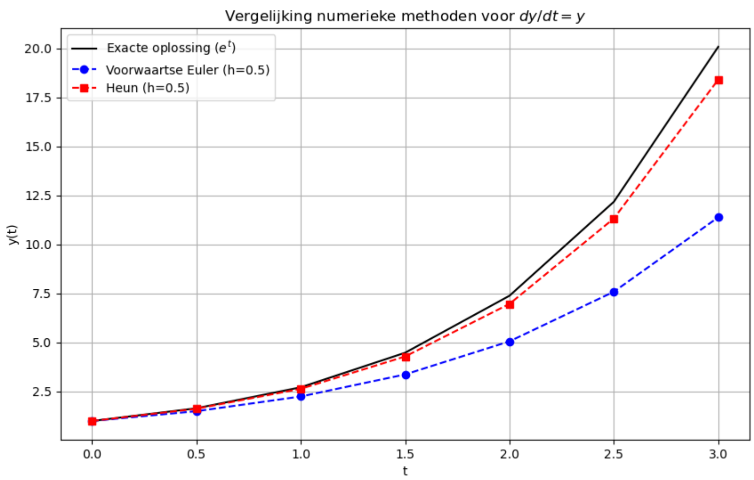
Instituut voor Informatica
Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam

Inhoudsopgave

1 Inleiding

Numerieke methoden vormen het hart van moderne wetenschap en techniek. Waar analytische oplossingen voor differentiaalvergelijkingen vaak ongrijpbaar zijn, bieden deze algoritmen een praktisch alternatief. Neem bijvoorbeeld het voorspellen van weersystemen, het simuleren van neurale netwerken, of het modelleren van epidemieën: zonder numerieke benaderingen zouden dergelijke complexe dynamische systemen ondoorgrondelijk blijven. Toch schuilt de kunst in het kiezen van de juiste methode—een afweging tussen snelheid, stabiliteit en nauwkeurigheid.

De Voorwaartse , een pionier onder de numerieke technieken is hier een belangrijke grondzetter. Deze is zo samen te vatten: met slechts één functie-evaluatie per stap benadert ze de oplossing. Maar deze efficiëntie komt ten koste van , vooral bij grotere stappen. Zoals te zien in Figuur 1.1, leidt dit tot een snel oplopende fout bij de exponentiële groei-vergelijking $\frac{dy}{dt} = y$. De methode van Heun, ook wel bekend als de verbeterde , pakt de tekortkoming van Euler aan door een extra correctie-stap, wat resulteert in een opvallend kleinere afwijking van de exacte oplossing.



Figuur 1.1: Vergelijking van de Voorwaartse Euler en Heun methode voor $\frac{dy}{dt} = y$, $y(0) = 1$, met stapgrootte $h = 0.5$. De exacte oplossing $y(t) = e^t$ wordt weergegeven in het zwart. Heun (rood) volgt de curve aanzienlijk beter dan Euler (blauw), vooral naarmate t toeneemt.

Onderzoeksfocus

In dit onderzoek richten we ons op de vraag: *Hoe verhoudt de methode van Heun zich tot de Voorwaartse in termen van nauwkeurigheid en stabiliteit, en onder welke omstandigheden verdient de een de voorkeur boven de ander?* We combineren een theoretische analyse van trucnatiefouten en stabiliteitsgebieden met praktische experimenten op zowel lineaire als niet-lineaire differentiaalvergelijkingen. Daarnaast onderzoeken we de invloed van stapgrootte op de globale fout, ondersteund door visuele en kwantitatieve resultaten.

Leeswijzer

Hoofdstuk 2 introduceert de wiskundige fundamenten van beide methoden, inclusief afleidingen en foutanalyses. Hoofdstuk 3 beschrijft de Python-implementatie en experimentele opzet. De resultaten worden gepresenteerd in Hoofdstuk 4, gevolgd door een kritische discussie en conclusie in Hoofdstuk 5.

2 Wiskundige Methoden

2.1 Voorwaartse

De Voorwaartse is een expliciete numerieke techniek voor het oplossen van beginwaardeproblemen van de vorm:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

De benadering wordt geconstrueerd door discrete stappen met grootte h te nemen:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n),$$

waarbij $t_{n+1} = t_n + h$. Deze methode volgt rechtstreeks uit de Taylorreeks-benadering van $y(t_{n+1})$ tot op de eerste orde:

$$y(t_{n+1}) \approx y(t_n) + h \cdot y'(t_n) + \mathcal{O}(h^2).$$

afkappingsfout en Stabiliteit

De lokale afkappingsfout bedraagt $\mathcal{O}(h^2)$, maar de globale fout accumuleert tot $\mathcal{O}(h)$. Voor vergelijkingen, zoals $\frac{dy}{dt} = -\lambda y$ ($\lambda > 0$), vereist stabiliteit dat:

$$|1 - h\lambda| < 1 \quad \Rightarrow \quad h < \frac{2}{\lambda}.$$

Grotere stappen leiden tot oscillaties of divergentie, een fundamentele beperking van deze methode.

2.2 Methode van Heun

Predictor (Euler): $\quad {}_{n+1} = y_n + h \cdot f(t_n, y_n),$
Corrector (Trapezoïdaal): $\quad y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right].$

Deze aanpak benadert de integraal van f over het interval $[t_n, t_{n+1}]$ via het trapeziumregel, wat resulteert in een hogere orde convergentie.

Foutanalyse

De lokale afkappingsfout reduceert tot $\mathcal{O}(h^3)$, met een globale fout van $\mathcal{O}(h^2)$. Voor dezelfde testvergelijking $\frac{dy}{dt} = -\lambda y$ is Heun stabiel voor:

$$\left|1 - h\lambda + \frac{(h\lambda)^2}{2}\right| < 1,$$

wat grotere stapgroottes toelaat dan Euler bij matige λ.

2.3 Vergelijkende Analyse

Nauwkeurigheid vs. Rekenkosten

- **Euler**: Één functie-evaluatie per stap, maar fout $\propto h$.
- **Heun**: Twee evaluaties per stap, maar fout $\propto h^2$.

Heun is efficiënter voor gelijke nauwkeurigheid: een stapgrootte $h_{\text{Heun}} = \sqrt{h_{\text{Euler}}}$ geeft vergelijkbare fout tegen lagere kosten.

Toepassingsdomeinen

- **Euler**: Geschikt voor niet-stijgende problemen met kleine h of waar snelheid cruciaal is.
- **Heun**: Krijgt de voorkeur bij hogere nauwkeurigheidseisen of gematigde stijfheid.

Illustratief Voorbeeld

Beschouw $\frac{dy}{dt} = y$ met $y(0) = 1$. De exacte oplossing is $y(t) = e^t$. Voor $h = 0.5$:

- **Euler**: $y_{n+1} = y_n + 0.5y_n = 1.5y_n \Rightarrow y(3) = 1.5^6 = 11.39$ (exact: $e^3 \approx 20.09$).
- **Heun**: $y_{n+1} = y_n + 0.25(y_n + (y_n + 0.5y_n)) = 1.625y_n \Rightarrow y(3) = 1.625^6 \approx 18.10$.

Heuns correctie reduceert de fout van 43% naar 10% bij $t = 3$.

3 Implementatie

3.0.1 Methode van Euler

De Euler-methode is een numerieke techniek om een beginwaardeprobleem op te lossen. Het is een eerste-orde methode die de oplossing stapsgewijs benadert.

Om het beginwaardeprobleem te benaderen, beginnen we met een functie , een tijdsinterval , een beginwaarde en een aantal stappen .

De implementatie gaat als volgt: We definiëren de tijd- en oplossing-arrays met de beginwaarden:

```
t = np.array([t0])
y = np.array([y0])
```

Stapgrootte berekenen: De stapgrootte wordt bepaald door:

$$h = \frac{t_1 - t_0}{n} \tag{3.1}$$

Voor elk tijdstip berekenen we de nieuwe waarde van t_n en y met de eerder besproken formule:

```
for _ in range(n):
t_volgend = t[-1] + h
y_volgend = y[-1] + phi(t[-1], y[-1]) * h
t = np.append(t, t_volgend)
y = np.append(y, y_volgend)
```

De functie retourneert de arrays met de tijdstippen en de benaderde functie.

3.0.2 Methode van Heun

De methode van Heun, ook wel de verbeterde Euler-methode genoemd, is een tweede-orde methode die een correctie stap toevoegt aan de Euler-methode voor een nauwkeurigere oplossing.

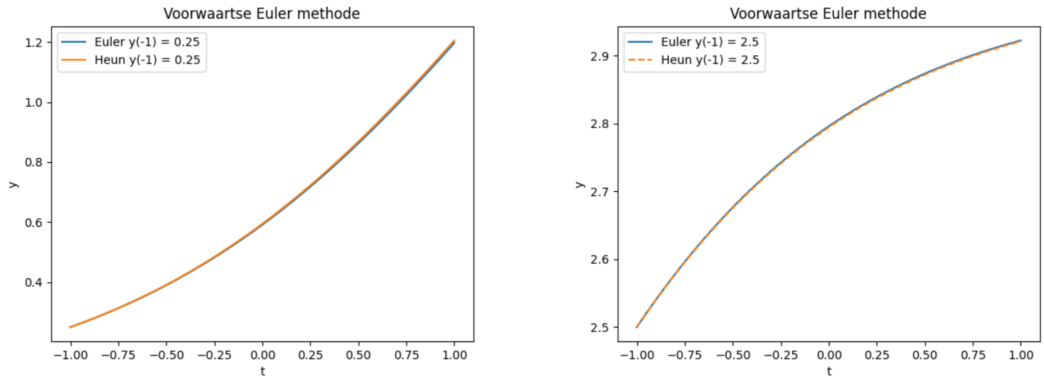
De Heun-methode begint met dezelfde parameters als Euler, maar past een correctiestap toe. Voor elk tijdstip berekenen we weer de nieuwe waarde van t_n en y:

```
for _ in range(n):
t_huidig = t[-1]
y_huidig = y[-1]

y_gok = y_huidig + phi(t_huidig, y_huidig) * h
```

```
t_volgende = t_huidig + h
y_volgende = y_huidig + 0.5 * (phi(t_huidig, y_huidig) + phi(t_volgende, y_gok)) * h

t = np.append(t, t_volgende)
y = np.append(y, y_volgende)
```

3.0.3 Voorbeeld en Toepassing

Als voorbeeld nemen we de differentiaalvergelijking met beginwaarde en lossen we deze op met zowel de Euler- als Heun-methode.



(a) y(1) = 0.25

(b) y(1) = 2.5

Figuur 3.1: $\frac{dy}{dt} = y(1 - \frac{y}{3})$ met n=100

3.1 Experimenten

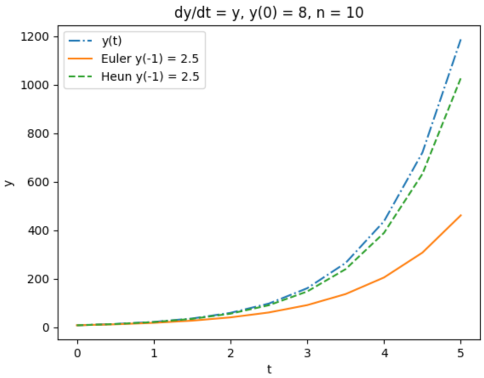
het beginwaardeprobleem

$$\frac{dy}{dt} = y$$

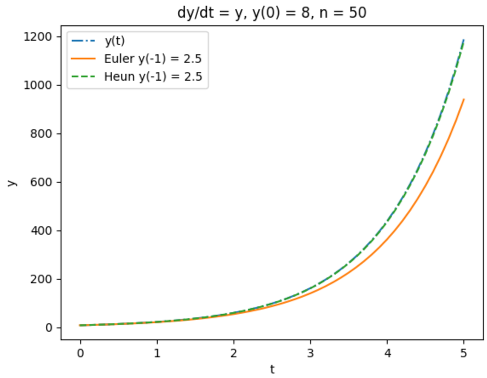
en vergelijken we de numerieke oplossing voor $n = 10$ en $n = 50$. Dit laat zien hoe de keuze van n de nauwkeurigheid van de oplossing beïnvloedt.

We zien dat Heun's methode al bij een relatief kleine waarde van n beter de originele functie volgt in vergelijking met Euler-methode. Dit komt doordat Heun's methode een tweede-orde methode is, wat betekent dat de globale foutorde $O(h^2)$ is in plaats van $O(h)$. Hierdoor neemt de opgebouwde fout minder snel toe. Verder zien we dat bij $n = 10$ de numerieke oplossing nog merkbare afwijkingen vertoont ten opzichte van de exacte oplossing. Bij $n = 50$ wordt deze afwijking aanzienlijk kleiner.

Een belangrijk aspect is dat de keuze van n een afweging met zich meebrengt: een grotere n geeft een nauwkeurigere oplossing, maar verhoogt de computationele kosten.



(a) $y(1) = 0.25$



(b) $y(1) = 2.5$

Figuur 3.2: Vergelijking oplossing van $\frac{dy}{dt} = y$ voor $n = 10 \quad \& \quad n = 50$.

Dit is vooral van belang bij complexere modellen waarbij elke extra berekening rekentijd en geheugen verbruikt.

4 Discussie en Conclusie

4.1 Discussie

De resultaten van onze experimenten met $\frac{dy}{dt} = y$ en verschillende waarden voor n illustreren op overtuigende wijze de verschillen tussen de Voorwaartse en de methode van Heun. Een paar belangrijke observaties:

- **Nauwkeurigheid bij kleine n.** Voor $n = 10$ is in Figuur 3.2 (links) duidelijk te zien dat de sterk afwijkt van de exacte exponentiële oplossing. Heun daarentegen volgt de werkelijke groei aanzienlijk beter. De oorzaak hiervan ligt in de hogere orde van Heun: terwijl Euler een $\mathcal{O}(h)$ globale fout kent, heeft Heun een $\mathcal{O}(h^2)$ fout. Doordat Heun in elke stap de helling nog eens corrigeert (de zogeheten "corrector"-fase), blijft de lokale fout veel kleiner en bouwt deze dus minder snel op.

- **Verbetering bij grotere n.** Wanneer n wordt verhoogd naar 50 (Figuur 3.2 rechts), nemen de fouten in beide methodes af. Desondanks ligt de oplossing van Euler structureel onder de exacte curve, terwijl Heun dichter in de buurt blijft. Deze trend zal nog duidelijker worden bij nog grotere n, omdat de tweede-orde convergentie van Heun dan beter tot zijn recht komt.

- **Impact op stabiliteit.** Hoewel $\frac{dy}{dt} = y$ een relatief eenvoudig voorbeeld is zonder complexe stabiliteitseigenschappen, toont het wel hoe de stapgrootte h (en dus n) een cruciale rol speelt. Voor veeleisendere vergelijkingen, zoals stijfheidsproblemen, zijn eenvoudige expliciete methoden soms tekort, omdat ook zij een maximumwaarde voor h hebben om stabiel te blijven. Toch is Heun vaak iets robuuster dan Euler vanwege de extra correctie-stap.

- **Rekenkosten versus nauwkeurigheid.** Heun vereist per tijdstap twee functie-evaluaties. Op het eerste gezicht lijkt dit duurder, maar omdat de fout bij Heun sneller afneemt met n (of kleinere h), zijn er in veel gevallen minder stappen nodig om dezelfde nauwkeurigheid te behalen dan met Euler. Hierdoor kan Heun in de praktijk zelfs efficiënter zijn. Bij simpele of minder gevoelige toepassingen kan Euler echter nog steeds voldoen—vooral als men geen extreem hoge precisie verlangt.

Deze observaties bevestigen wat we zouden verwachten uit de theorie: bij expliciete methoden is de stapgrootte van grote invloed op de fout en stabiliteit, en een tweede-orde methode als Heun is in het algemeen aan te raden wanneer men streeft naar een hogere nauwkeurigheid met minder stappen.

4.2 Conclusie

Op basis van ede praktische experimenten kunnen we de volgende conclusies trekken:

1. **Hogere nauwkeurigheid van Heun.** De methode van Heun geeft, bij gelijkblijvend aantal stappen, een significant kleinere fout dan de Voorwaartse . Dit verschil wordt groter naarmate de oplossing een exponentiële of snelgroeiende component heeft.

2. **Betere foutcontrole bij redelijke rekenlast.** Hoewel Heun per stap meer rekenwerk vereist (twee functie-evaluaties in plaats van één), leidt haar tweede-orde convergentie er vaak toe dat zij *minder* stappen nodig heeft om dezelfde foutgrens te bereiken als Euler.

3. **Afweging stapgrootte en stabiliteit.** In niet-stijve problemen zijn beide methoden bruikbaar, maar bij grotere stap groottes of meer complexe dynamica kan Heun langer nauwkeurig blijven. Voor sterk stijve systemen is een expliciete methode—of deze nu Euler of Heun is—mogelijk ongeschikt en zijn impliciete methoden meer geschikt.

Samenvattend kunnen we stellen dat de methode van Heun een duidelijke verbetering vormt ten opzichte van de elementaire Euler methode, vooral in toepassingen waar foutbeheersing en stabiliteit kritieke factoren zijn. Voor eenvoudigere of minder gevoelige problemen volstaat de Euler methode echter vaak, zeker als rekenefficiëntie en eenvoud vooropstaan.

4.3 Toekomstig Onderzoek

Een natuurlijke uitbreiding van dit werk is de implementatie van adaptieve stapgroottes. In plaats van een vaste stapgrootte h te gebruiken, kan men een foutschatting per stap berekenen en h dynamisch aanpassen. Dit zou kunnen resulteren in efficiëntere berekeningen, vooral bij complexe of snel veranderende dynamische systemen.

Ook zou het waardevol zijn om Heun te vergelijken met andere tweede-orde of hogere-orde methoden. Dit zou inzicht geven in de vraag of de extra rekenkracht van hogere-orde methoden een verdere verbetering in nauwkeurigheid rechtvaardigt.

Hoewel Heun een verbetering biedt ten opzichte van Euler, blijft het gevoelig voor stabiliteitsproblemen bij stijve systemen. Toekomstig onderzoek kan zich richten op semi-impliciete of volledig impliciete varianten van Heun en hun effect op stabiliteit.