

Het 'oplossen' van differentiaalvergelijkingen met behulp van RK2 methoden



Christiaan Stringa 5701503 *Begeleider:* Dr. Tristan van Leeuwen

> Bachelorscriptie 17 Januari 2020

Samenvatting

Differentiaalvergelijkingen komen in de praktijk vaak voor. Helaas hebben de meeste van deze differentiaalvergelijkingen geen exacte oplossing. Om deze reden zijn numerieke methoden nodig om de oplossingen te benaderen. Echter moeten de numerieke methoden wel voldoen aan een aantal eisen. Zo moeten ze de exacte oplossing fatsoenlijk benaderen en het aantal operaties mag niet te veel zijn.

Een soort methoden die de oplossing goed benaderen is bedacht door Carl Runge in 1895 en later uitgebreid en verfijnd door Wilhelm Kutta in 1905. Deze methoden heten heel toepasselijk de Runge Kutta of RK-methoden. Het idee was om een aantal benaderingen van de functie op de volgende tijdstap te bepalen met behulp van de functiewaarde op de huidige tijdstap. Vervolgens werd het gemiddelde van deze benaderingen genomen en hadden wij een betere benadering.

Door de vele berekeningen konden deze methoden aan het begin niet realistisch gebruikt worden. Het kostte te veel tijd. Echter rond 1950 met de uitvinding van de computer zijn deze methoden realiseerbaar geworden. Men kon nu veel meer operaties per minuut uitvoeren en omdat deze methoden de oplossing na weinig iteraties goed benaderen, werden ze vaak in gebruik genomen. In de huidige tijd zijn de Runge Kutta methoden nu één van de meest gebruikte numerieke methoden om differentiaalvergelijkingen te bepalen.

Inhoudsopgave

1	Introductie	3		
2	Basis van numerieke methoden voor differentiaalvergelijkingen2.1Basisbegrippen2.2Belangrijke stellingen	3 3 4		
3	Algemene Runge-Kutta Methoden 3.1 Algemene Runge Kutta methoden 3.2 Stabiliteit	5 5 7		
4	RK2 methoden 4.1Algemene RK2 methoden4.2Truncatiefout4.3Stabiliteit	9 9 13 16		
5	Specifieke RK-methoden5.1Forward Euler5.2Expliciete Midpuntregel5.3Heun-methode5.4Ralston-methode5.5Eerste orde methode	18 18 19 20 20 21		
6	RK toegepast op voorbeelden 6.1Chemische reactie6.2Slinger6.3Predator Prey Model6.4Lorenz stelsel	22 22 26 31 35		
7	Conclusie	39		
Referenties		39		
Ap	Appendix			

1 Introductie

Zoals eerder gezegd komen differentiaalvergelijkingen vaak voor. De meesten hiervan hebben geen exacte oplossing. Om deze reden worden numerieke methoden gebruikt om de exacte oplossing te benaderen. In deze scriptie gaan wij onderzoek doen naar RK2 methoden. Dit zijn numerieke methoden die zijn gevonden door Runge en Kutta.

Eerst zal de basis van numerieke methoden voor differentiaalvergelijkingen kort behandeld worden. Begrippen als de tijdstapgrootte en de truncatiefout zullen gedefinieerd worden. In deze paragraaf zullen wij ook de stellingen van Taylor behandelen. Deze zullen veel gebruikt worden door de scriptie heen.

Daarna zullen algemene Runge Kutta methoden besproken worden en zullen er bepaalde concepten geïntroduceerd worden. Concepten zoals stabiliteit en consistentie zullen gedefinieerd worden. Uit de definitie van consistentie zal gelijk volgen dat de methode consistent is als hij een orde groter of gelijk aan één heeft.

Daarop volgt de theorie van de RK2 methoden. De algemene theorie zal gespecificeerd worden naar deze deelklasse van Runge Kutta methoden. Wij zullen zien wanneer een methode orde één en twee heeft. Daarna leiden we een bovengrens af voor de fout. Hieruit volgt dat de RK2 methoden maximaal orde twee hebben.

Vervolgens zullen we specifieke RK2 methoden behandelen. Sommige bekende methoden als de methodes van Heun en Ralston worden behandeld en wij zullen een eigen RK2 methode maken. Wij zullen de vorm en de stabiliteit van deze methoden behandelen.

Deze methoden zullen toegepast worden op voorbeelden uit de praktijk zoals de slinger en een eenvoudig predator-prey model. De implementatie van deze methoden wordt gedaan in Matlab. De code is terug te lezen in de Appendix. Uit deze numerieke tests zullen bepaalde voordelen en nadelen van de verschillende RK2 methoden duidelijk naar voren komen. Hieruit kunnen wij dan een conclusie trekken over de RK2 methoden en hun nut.

2 Basis van numerieke methoden voor differentiaalvergelijkingen

In deze paragraaf zullen we de basis van de numerieke methoden bespreken. Eerst zullen wij de differentiaalvergelijkingen definiëren. Hierbij geven wij een algemene vorm. Daarna gaan wij het hebben over concepten als tijdstapgrootte en lokale truncatiefout. Verder zullen wij twee stellingen behandelen die later in de scriptie een aantal keer gebruikt worden.

2.1 Basisbegrippen

De algemene vorm van de differentiaalvergelijkingen die wij willen oplossen is:

$$y'(t) = f(t, y(t))$$

In het vervolg van de scriptie zullen wij aannemen dat *y* afhankelijk is van *t*, maar dit niet expliciet vermelden. Wij zullen dit schrijven als y' = f(t, y).

Nu kunnen we overgaan op numerieke methoden. Hiervoor zullen wij een tijdsinterval nodig hebben om de differentiaalvergelijkingen over te beschouwen. Laten we aannemen dat het tijdsinterval voor onze differentiaalvergelijkingen gelijk is aan [0, T]. Het idee van de numerieke methode is nu dat we het tijdsinterval verdelen in een aantal tijdstappen t_n . Wij definiëren nu een grootheid genaamd de tijdstapgrootte.

Definitie 2.1.1. Neem aan dat er N tijdstappen gekozen zijn voor het tijdsinterval [0,T]. De **tijdstapgrootte** h is nu gelijk aan $\frac{T}{N}$.

Wij zullen de tijdstappen altijd evenredig verdelen. Dit betekent dat de tijdstapgrootte h gelijk is aan $t_{n+1} - t_n$. Her verschil tussen twee opeenvolgende tijdstappen is gelijk aan h.

Het idee van numerieke methoden is nu om de functie te benaderen op iedere tijdstap t_n . Wij krijgen hierdoor een benadering van de functie y. Wij noemen deze benadering vanaf nu \tilde{y}_n . Verder definiëren wij de exacte oplossing op tijdstap als $y_n = y(t_n)$.

Nu dat wij een methode hebben willen wij een uitspraak kunnen doen over de fout tussen de benadering en de exacte oplossing. De meest voorkomende fout is de zogeheten lokale truncatiefout.

Definitie 2.1.2. Neem y_n de exacte oplossing van y' = f(t, y) en \tilde{y}_n een benadering van de exacte oplossing. De **lokale truncatiefout** E_n wordt gegeven door:

$$E_n = |y_n - \tilde{y}_n|$$

Dit is de fout die we krijgen door het verwaarlozen van termen in de benadering. Hiervoor geven wij een voorbeeld.

Voorbeeld 2.1.3. Stel de exacte oplossing y_n is gelijk aan $x + 2x^2 + \mathcal{O}(x^3)$ en de benaderde oplossing \tilde{y}_n is gelijk aan $x + 2x^2$. De lokale truncatiefout is nu gelijk aan $\mathcal{O}(x^3)$.

Verder hebben wij te maken met een computer en dus bepaalde precisie van de computer. Hieruit kunnen afrondfouten en rekenfouten tevoren komen. Hiermee moeten wij ook rekening houden als wij een goede methode willen creëren.

2.2 Belangrijke stellingen

Ten slotte geven wij twee belangrijke stellingen zonder bewijs. Deze stellingen zullen in de loop van de scriptie regelmatig gebruikt worden voor bewijzen van onze stellingen.

Stelling 2.2.1 (Stelling van Taylor). *Laat* f(x) *een willekeurig vaak differentieerbare functie zijn. Er geldt nu dat:*

$$f(x) = \sum_{k=0}^{\infty} f^{(k)}(x^*) \frac{(x-x^*)^k}{k!}$$

Hierbij heet x^{*} *het steunpunt van de Taylorexpansie*

Als wij de som eerder afbreken en dan de overige termen vervangen voor het Landau symbool dan spreken we van een k-de orde Taylorexpansie. Een voorbeeld is hier op zijn plaats:

Voorbeeld 2.2.2. De tweede orde Taylorexpansie wordt gegeven door

$$f(x) = f(x^*) + f'(x^*)(x - x^*) + \frac{1}{2}f''(x^*)(x - x^*)^2 + \mathcal{O}((x - x^*)^3)$$

Hiermee zijn wij precies tot orde twee en daarna gebruiken wij het Landau symbool om de overige termen te onthouden. De tweede stelling die we nodig hebben is de tweedimensionale variant van de stelling van Taylor. Hiervoor zullen wij alleen de tweede orde tweedimensionale Taylor expansie geven. We hebben in onze scriptie geen hogere orde expansie nodig.

Stelling 2.2.3. *Laat* f(x,y) *tweemaal differentieerbaar. De tweede orde Taylorexpansie wordt nu gegeven door:*

$$f(x + \Delta x, y + \Delta y) = f(x, y) + \Delta x \frac{df}{dx}(x, y) + \Delta y \frac{df}{dy}(x, y) + \frac{1}{2} \left(\Delta x^2 \frac{d^2 f}{dx^2}(x, y) \right.$$
$$\left. + \Delta x \Delta y \frac{d^2 f}{dx dy}(x, y) + \Delta y^2 \frac{d^2 f}{dy^2}(x, y) \right) + \mathcal{O}(\Delta x^3 + \Delta y^3)$$

Hierbij noemen wij (x,y) *het steunpunt*

Deze tweediemnsionale Taylorexpansie zullen wij terugzien als wij de toestanden van de Runge Kutta methoden willen herschrijven. Een belangrijk gevolg van de tweede orde Taylorexpansie van f(x,y) is de eerste orde expansie. Deze wordt beschreven in het volgende gevolg

Gevolg 2.2.4. *De eerste orde Taylorexpansie van* f(x,y) *wordt gegeven door:*

$$f(x + \Delta x, y + \Delta y) = f(x, y) + \Delta x \frac{df}{dx}(x, y) + \Delta y \frac{df}{dy}(x, y) + \mathcal{O}(\Delta x^2 + \Delta y^2)$$

3 Algemene Runge-Kutta Methoden

3.1 Algemene Runge Kutta methoden

Wij zullen eerst de algemene Runge Kutta methode definieren voordat we ingaan op specifiekere situaties. Het doel van de RK methoden is het vinden van een oplossing voor de differentiaalvergelijking y' = f(t, y).

Definitie 3.1.1. De vorm van een algemene expliciete Runge Kutta methode is:

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i$$

waarbij

$$k_{1} = f(t_{n}, y_{n})$$

$$k_{2} = f(t_{n} + c_{2}h, y_{n} + ha_{21}k_{1})$$

$$k_{3} = f(t_{n} + c_{3}h, y_{n} + h(a_{31}k_{1} + a_{32}k_{2}))$$

$$\vdots$$

$$k_{s} = f(t_{n} + c_{s}h, y_{n} + h\sum_{j=1}^{p-1} a_{sj}k_{j})$$

`

Hierbij is s een geheel getal en zijn c_i, b_i en a_{ij} reele coefficienten. Nu zullen we een aantal van deze constanten benoemen zodat we hier later gemakkelijk naar kunnen refereren.

Definitie 3.1.2. Stel we hebben een algemene Runge Kutta methode zoals hierboven gedefinieerd.

- Het aantal verschillende k_i noemen we het het aantal **fases/toestanden**
- Een Runge Kutta methode met s toestanden heet een RKs methode
- De *b_i* heten de **gewichten** en de *c_i* heten de **knopen**
- De matrix $A = [a_{ij}]$ met $1 \le i, j \le s$ heet de **Runge-Kutta Matrix**

Wat wij gaan onderzoeken zijn expliciete RK2-methoden. Dit zijn expliciete Runge Kutta methoden met twee toestanden. Merk op dat de k_i uniek gedefinieerd zijn voor elke tijdstap n, en niet één vaste waarde hebben. Dit betekent dat voor iedere tijdstap de k_i een nieuwe waarde zal aannemen. Dit zorgt ervoor dat de RK methoden veel operaties nodig hebben om een goede benadering te geven. Nu is de algemene Runge Kutta methode dedefinieerd. Deze methoden worden vaak op de volgende manier genoteerd

Definitie 3.1.3. Stel we hebben een algemene Runge Kutta methode met s toestanden, zoals in Definitie 3.1.1. Geef de methode op de volgende manier weer

Deze notatie heet een Butcher Tableau voor de RKs-methode

Het is gebruikelijk om de 0 elementen niet weer te geven in het Butcher Tableau. We zien dat de Runge Kutta matrix A en de gewichtenvector $B = (b_1, b_2, \dots, b_s)$ gemakkelijk af te lezen zijn. Verder valt het op dat de Runge Kutta matrix van een expliciete methode benedendriehoeks is.

Nu we een idee hebben van wat een Runge Kutta methode inhoudt, willen wij een uitspraak doen over de fout tussen de exacte oplossing op tijdstip t_n en de benadering door middel van de Runge Kutta methode. Hiervoor is de volgende definitie noodzakelijk:

Definitie 3.1.4. Een Runge Kutta methode heeft **orde** n als de lokale truncatiefout E_n gelijk is aan $\mathcal{O}(h^{n+1})$

Nu dat wij de orde hebben gedefinieerd willen we uitspraken kunnen doen over de groei van de fout. Hiervoor voeren wij de begrippen consistentie en convergentie in.

Definitie 3.1.5. De Runge Kutta methode heet consistent, als

$$\lim_{h \to 0} \frac{E_n}{h} = 0$$

Het begrip consistentie houdt dus in dat de lokale truncatiefout naar 0 gebracht kan worden, door verkleiningen op de tijdstapgrootte h. We zien vanuit de definities dat een Runge Kutta methode met orde n > 0 altijd consistent is. Laten wij nu convergentie definiëren

Definitie 3.1.6. Laat y_n de exacte oplossing zijn van de differentiaalvergelijking y' = f(t,y) en \tilde{y}_n de benadering door middel van de numerieke methode. De methode heet **convergent** indien:

$$\lim_{n \to \infty} |y_n - \tilde{y}_n| = 0$$

Dit betekent dat de totale fout naar 0 groeit. Hieruit is het verschil tussen convergentie en consistentie duidelijk zichtbaar. Consistentie geeft alleen een uitspraak over de lokale truncatiefout terwijl convergentie gaat over de totale fout inclusief bijvoorbeeld afrondfouten. We concluderen dat convergentie impliceert consistentie maar niet andersom.

3.2 Stabiliteit

In deze paragraaf zullen we de stabiliteit behandelen. Dit concept houdt rekening met afrondfouten en beginwaardefouten. Bovendien zorgt een stabiele methode ervoor dat dit soort fouten verkleint per tijdstap. Het grote voordeel van Runge Kutta methoden is hun grote stabiliteit. Daarom zullen wij een functie voor het stabiliteitsgebied afleiden.

Om de stabiliteit te onderzoeken wordt de methode toegepast op de testvergelijking $f(t_n, y_n) = \lambda y_n$. Hierna zullen wij de daaruitvolgende uitdrukking herschrijven naar de vorm $y_{n+1} = R(h\lambda)y_n$. De methode is nu stabiel indien $|R(h\lambda)| < 1$. De functie $R(h\lambda)$ heet vanaf nu de **stabiliteitsfunctie**

Stelling 3.2.1. ¹ *Stel wij hebben een Runge Kutta methode. De stabiliteitsfunctie wordt als volgt gegeven.*

$$R(\mu) = 1 + \mu B (\mathbb{I} - \mu A)^{-1} \mathbb{1}$$
(1)

Hierbij is

¹Jason Frank. Stability of Runge Kutta Methods.

• $\mu = h\lambda$

• $B = (b_1, b_2, \dots, b_s)^T$ waarbij b_i de gewichten van de Runge Kutta methode

- A is de Runge Kutta matrix
- $1 = (1, 1, \dots, 1)^T$

Bewijs: Vanwege Definitie 3.1.1 heeft een algemene RKs methode de vorm:

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i$$
(2)

$$k_{i} = f(t_{n} + c_{i}h, y_{n} + h\sum_{j=1}^{s-1} a_{ij}k_{j})$$
(3)

Als nu de testvergelijking $f(t_n, y_n) = \lambda y_n$ toegepast wordt op (3), dan vinden wij

$$k_i = \lambda \left(y_n + h \sum_{j=1}^{s-1} a_{ij} k_j \right)$$

Als nu deze term tot een matrixvermenigvuldiging wordt herschreven en wij introduceren de volgende variabelen $K = (k_1, k_2, ..., k_s)^T$, $\mathbb{1} = (1, 1, ..., 1)^T$ en $A = [a_{ij}]$, dan vinden wij:

$$K = \lambda \mathbb{1} y_n + h \lambda A K$$
$$K - h \lambda A K = \lambda \mathbb{1} y_n$$
$$(\mathbb{I} - h \lambda A) K = \lambda \mathbb{1} y_n$$

Als beide kanten vanaf links worden vermenigvuldigd met de inverse van $(\mathbb{I} - h\lambda A)$, dan concluderen wij dat:

$$K = \left(\mathbb{I} - h\lambda A\right)^{-1}\lambda \mathbb{1}y_n \tag{4}$$

Vervolgens zullen we (2) uitwerken tot

$$y_{n+1} = y_n + hBK$$

waarbij $B = (b_1, b_2, \dots, b_s)$. Als wij nu (4) invullen, dan vinden wij

$$y_{n+1} = y_n + hB(\mathbb{I} - h\lambda A)^{-1}\lambda \mathbb{1}y_n$$
$$= (1 + hB(\mathbb{I} - h\lambda A)^{-1}\lambda \mathbb{1})y_n$$

Ten slotte keren we terug naar convergentie. We hebben nu de begrippen consistentie en stabiliteit geintroduceerd voor numerieke methoden. Wij zien dat als aan deze twee eisen voldaan zijn voor een numerieke methode, dan is de methode convergent.

4 RK2 methoden

Nu we het algemene geval hebben behandeld en een aantal handige definities hebben, kunnen we dieper ingaan op de Runge Kutta met twee toestanden. Eerst zullen we de algemene RK2 methode bekijken. Aan welke eisen moet een RK2 methode voldoen om consistent en stabiel te zijn? Welke orde heeft de methode en wat is de lokale truncatiefout?

4.1 Algemene RK2 methoden

Definitie 4.1.1. De algemene vorm van een RK2 methode is:

$$y_{n+1} = y_n + h(ak_1 + bk_2)$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f(t_n + \alpha h, y_n + \beta k_1 h)$$

Hierbij zijn a, b, α en β reële constanten.

Merk op dat het volgende Butcher Tableau de algemene RK2-methode representeert.

$$\begin{array}{c|c} 0 \\ \alpha & \beta \\ \hline a & b \end{array}$$

Nu dat de RK2-methode is gedefinieerd, willen we uitspraken doen over de orde, de fout en de stabiliteit. Wij zullen voorwaarden geven waarmee de methode orde een of twee krijgt. Verder zullen we een bovengrens voor de lokale truncatiefout afleiden. Hieruit zal volgen dat de RK2-methoden maximaal orde twee kunnen hebben. Eerst zullen we de voorwaarden bepalen waarvoor de RK2 methode orde 1 heeft. Wij geven de volgende stelling.

Stelling 4.1.2. *De RK2 methode gedefinieerd in 4.1.1 heeft orde één indien* a + b = 1

Voor het bewijs hiervoor hebben we de volgende opmerkingen nodig

Opmerking. De eerste orde Taylor expansie van de exacte oplossing $y_{n+1} = y(t_{n+1})$ in het steunpunt t_n wordt gegeven door

$$y(t_{n+1}) = y(t_n) + y'(t_n)(t_{n+1} - t_n) + \mathcal{O}((t_{n+1} - t_n)^2)$$

= $y(t_n) + f(t_n, y_n)h + \mathcal{O}(h^2)$

Nu dat wij de exacte oplossing hebben uitgedrukt in termen van y_n , kan hetzelfde gedaan worden met de benaderde oplossing \tilde{y}_{n+1} . Wij geven het volgende lemma:

Lemma 4.1.3. *De door middel van de RK2-methode bepaalde benadering kan op de volgende manier worden weergegeven.*

$$\tilde{y}_{n+1} = \tilde{y}_n + (a+b)hf(t_n, y_n) + \mathcal{O}(h^2)$$

Bewijs: Wij bepalen eerst de 'nulde' orde Taylorexpansie van $k_2 = f(t_n + \alpha h, y_n + \beta k_1 h)$. Wij zien dat deze gelijk is aan

$$k_2 = f(t_n, y_n) + \mathcal{O}(h)$$

Als wij deze nu invullen in de definitie van de RK2 methode dan wordt deze:

$$\tilde{y}_{n+1} = y_n + ahf(t_n, y_n) + bh(f(t_n, y_n) + \mathcal{O}(h))$$
$$= y_n + (a+b)hf(t_n, y_n) + \mathcal{O}(h^2)$$

Nu dat wij een uitdrukking hebben voor de exacte oplossing y_n en de benaderde oplossing \tilde{y}_n , volgt het bewijs van 4.1.2 gemakkelijk.

Bewijs van 4.1.2: Als wij de eerste orde Taylor expansie van de exacte oplossing y_n gelijkstellen aan de expansie van de benaderde oplossing \tilde{y}_n , dan volgt dat deze twee gelijk zijn tot orde twee, indien a+b=1.

Nu hebben wij gezien wat een eerste orde Rk2 methode inhoudt. Wij zullen nu hetzelfde doen voor tweede orde methoden. Opnieuw zullen er voorwaarden volgen waarmee de RK2 methode orde twee wordt. We formuleren de volgende stelling.

Stelling 4.1.4. De methode gedefinieerd in 4.1.1 is een tweede orde RK2 methode indien:

• a + b = 1• $\alpha b = \frac{1}{2}$ • $\beta b = \frac{1}{2}$

Voor het bewijs van deze stelling hebben wij een aantal lemma's nodig. Het eerste lemma geeft een andere representatie voor de tweede afgeleide $\frac{d^2y}{dt^2}(t)$.

Lemma 4.1.5. Stel y is een oplossing van de differentiaalvergelijking $\frac{dy}{dt} = f(t, y)$, dan geldt

$$\frac{d^2y}{dt^2}(t) = \frac{df}{dt}(t,y) + f(t,y)\frac{df(t,y)}{dy}$$
(5)

Bewijs:

$$\frac{d^2y}{dt^2}(t) = \frac{df(t,y)}{dt}$$

Vanwege de kettingregel voor differentiëren, kan dit herschreven worden naar

$$= \frac{df}{dt}(t,y) + \frac{df(t,y)}{dy}\frac{dy(t)}{dt}$$
$$= \frac{df}{dt}(t,y) + f(t,y)\frac{df(t,y)}{dy}$$

Vervolgens willen wij de exacte oplossing $y(t_{n+1})$ uitdrukken in termen geevalueerd op tijdstap t_n . Omdat wij een tweede orde methode willen aantonen, hoeven we de termen h^k met k > 2 niet precies mee te nemen in onze bepaling. We verkrijgen het volgende lemma

Lemma 4.1.6. Stel y is een oplossing van de differentiaalvergelijking $\frac{dy}{dt} = f(t, y)$

$$y(t_{n+1}) = y(t_n) + hf(t_n, y) + \frac{h^2}{2} \left(\frac{df}{dt}(t_n, y) + f(t_n, y) \frac{df(t_n, y)}{dy} \right) + \mathcal{O}(h^3)$$

Bewijs: Om dit lemma aan te tonen, gebruiken we de stelling van Taylor in het steunpunt t_n

$$y(t_{n+1}) = y(t_n) + h \left. \frac{dy}{dt} \right|_{t=t_n} + \frac{h^2}{2} \left. \frac{d^2 y}{dt^2} \right|_{t=t_n} + \mathcal{O}(h^3)$$
$$= y(t_n) + h f(t_n, y) + \left. \frac{h^2}{2} \left. \frac{d^2 y}{dt^2} \right|_{t=t_n} + \mathcal{O}(h^3)$$

Wij zullen nu Lemma 4.1.5 gebruiken

$$= y(t_n) + hf(t_n, y) + \frac{h^2}{2} \left(\frac{df}{dt}(t, y) + f(t, y) \frac{df(t, y)}{dy} \right) \Big|_{t=t_n} + \mathcal{O}(h^3)$$

= $y(t_n) + hf(t_n, y) + \frac{h^2}{2} \left(\frac{df}{dt}(t_n, y) + f(t_n, y) \frac{df(t_n, y)}{dy} \right) + \mathcal{O}(h^3)$

Nu hebben wij de exacte oplossing *y* op tijdstap t_n herschreven. Voor het laatste lemma zullen wij de term k_2 anders schrijven, zodat wij hem kunnen vergelijken met het bovenstaande lemma. Hieruit kunnen we dan conclusies trekken over de benodigde voorwaarden op de methode gedefinieerd in Definitie 4.1.1

Lemma 4.1.7. *Zij* $f : \mathbb{R}^2 \to \mathbb{R}$ *continu dan geldt*

$$f(t_n + \alpha h, y_n + \beta k_1 h) = f(t_n, y_n) + \alpha h \frac{df}{dy}(t_n, y_n) + h\beta f(t_n, y_n) \frac{df}{dt}(t_n, y_n) + \mathcal{O}(h^2)$$

Bewijs: Voor het bewijs zal een tweedimensionale Taylorexpansie in het steunpunt (t_n, y_n) gebruikt worden.

$$k_{2} = f(t_{n} + \alpha h, y_{n} + \beta k_{1}h)$$

= $f(t_{n}, y_{n}) + \alpha h \frac{df}{dt}(t_{n}, y_{n}) + h\beta k_{1} \frac{df}{dy}(t_{n}, y_{n}) + \mathcal{O}(h^{2})$
= $f(t_{n}, y_{n}) + \alpha h \frac{df}{dt}(t_{n}, y_{n}) + h\beta f(t_{n}, y_{n}) \frac{df}{dy}(t_{n}, y_{n}) + \mathcal{O}(h^{2})$

Nu hebben wij alle benodigde hulpstukken om onze stelling over tweede orde RK2 methoden te bewijzen.

Bewijs van Stelling 4.1.4: Wij herschrijven onze oorspronkelijke methode uit Definitie 4.1.1 met behulp van Lemma 4.1.7. Dit doen wij op de volgende manier.

$$y_{n+1} = y_n + h(ak_1 + bk_2)$$

= $y_n + ahf(t_n, y_n) + bhf(t_n + \alpha h, y_n + \beta k_1 h)$

Nu zullen wij gebruik maken van Lemma (4.1.7)

$$= y_{n} + ahf(t_{n}, y_{n}) + bh\left(f(t_{n}, y_{n}) + \alpha h\frac{df}{dt}(t_{n}, y_{n}) + h\beta f(t_{n}, y_{n})\frac{df}{dt}(t_{n}, y_{n}) + \mathcal{O}(h^{2})\right)$$

$$= y_{n} + (a+b)hf(t_{n}, y_{n}) + \alpha h^{2}b\frac{df}{dt}(t_{n}, y_{n}) + b\beta h^{2}f(t_{n}, y_{n})\frac{df}{dy}(t_{n}, y_{n}) + \mathcal{O}(h^{3})$$
(6)

Voor een tweede orde RK2 methode moeten Lemma 4.1.6 en (6) overeenkomen. Nu trekken wij de volgende conclusies over de constanten.

• a+b=1• $\alpha h^2 b = \frac{h^2}{2}$ oftewel $\alpha b = \frac{1}{2}$ • $\beta h^2 b = \frac{h^2}{2}$ oftewel $\beta b = \frac{1}{2}$

We merken op door de restricties dat er een vrije keuze is voor een van de constanten. Immers hebben wij een stelsel van drie vergelijkingen met vier onbekenden. Dit betekent ook dat er meerdere tweede orde RK2 methoden bestaan. Wij schrijven nu de constanten a, b, α en β om in de variabele γ . De restricties veranderen nu naar:.

• $\gamma = \alpha = \beta$ • $b = \frac{1}{2\gamma}$ • $a = 1 - \frac{1}{2\gamma}$

Als we dit nu invullen in onze oorspronkelijke RK2 methode, dan krijgen wij de volgende definitie:

Definitie 4.1.8. De algemene tweede orde Runge Kutta methode met twee toestanden kan worden weergegeven door:

$$y_{n+1} = y_n + h\left(\left(1 - \frac{1}{2\gamma}\right)k_1 + \frac{1}{2\gamma}k_2\right)$$
$$k_1 = f(t_n, y_n)$$
$$k_2 = f(t_n + \gamma h, y_n + \gamma k_1 h)$$

Het Butcher Tableau van de algemene tweede orde Runge Kutta methode wordt gegeven door



4.2 Truncatiefout

Wij hebben nu de algemene RK2 methode van orde een en twee behandeld. In deze paragraaf zullen we zien dat er een bovengrens bestaat op de lokale truncatiefout op tijdstap t_n . Hieruit zal volgen dat RK2 methoden geen orde hoger dan twee kunnen hebben.

Stelling 4.2.1. Voor een tweede orde Runge Kutta methode met twee toestanden geldt de volgende uitspraak over de lokale truncatiefout E_n :

$$E_n < ML^2 \left(4 \left| \frac{1}{6} - \frac{\gamma}{4} \right| + \frac{1}{3} \right) h^3$$

waarbij in een omgeving rondom (t_n, y_n) geldt $|f(t_n, y_n)| < M$ en $|\frac{d^{i+j}f}{dt^i dy^j}| < \frac{L^{i+j}}{M^{j-1}}$

Voor overzichtelijkheid introduceren wij de volgende twee functies.

$$F(t,y) = \frac{df}{dt}(t,y) + f(t,y)\frac{df(t,y)}{dy}.$$
(7)

$$\Phi(t,y) = \frac{d^2f}{dt^2}(t,y) + 2\frac{d^2f}{dydt}(t,y)f(t,y) + \frac{d^2f}{dy^2}(t,y)(f(t,y))^2$$
(8)

Vanwege Lemma 4.1.5 geldt dat

$$F(t,y) = \frac{d^2y}{dt^2}(t,y)$$

Voor het bewijs van Stelling 4.2.1 zijn een aantal lemma's nodig. Het eerste lemma geeft een andere weergave van de derde afgeleide $\frac{d^3y}{dt^3}$

Lemma 4.2.2. *Voor een functie* $f : \mathbb{R}^2 \to \mathbb{R}$ *geldt*

$$\frac{d^3y}{dt^3} = \Phi(t,y) + \frac{df}{dy}(t,y)F(t,y)$$

Bewijs Gebruik dat $\frac{d^3y}{dt^3} = \frac{d}{dt}\frac{d}{dt}\frac{dy}{dt}$. We weten vanwege het gegeven dat $\frac{dy}{dt} = f(t,y)$. Vanuit Lemma 4.1.5 kan men concluderen dat $\frac{d}{dt}\frac{dy}{dt} = \frac{d}{dt}f(t,y) = \frac{df}{dt}(t,y) + f(t,y)\frac{df(t,y)}{dy}$ Nu volgt

$$\begin{aligned} \frac{d^3y}{dt^3} &= \frac{d}{dt} \left(\frac{df}{dt}(t,y) + f(t,y) \frac{df(t,y)}{dy} \right) \\ &= \frac{d}{dt} \left(\frac{df}{dt}(t,y) \right) + \frac{d}{dt} \left(f(t,y) \frac{df(t,y)}{dy} \right) \\ &= \frac{d^2f}{dt^2}(t,y) + \frac{d^2f}{dydt}(t,y)f(t,y) + \frac{df}{dy}(t,y) \frac{df}{dt}(t,y) + \left(\frac{df}{dy}(t,y) \right)^2 f(t,y) + \frac{d^2f}{dydt}(t,y)f(t,y) + \frac{d^2f}{dy^2}(t,y)(f(t,y))^2 \\ &= \frac{d^2f}{dt^2}(t,y) + 2\frac{d^2f}{dydt}(t,y)f(t,y) + \frac{df}{dy}(t,y) \frac{df}{dt}(t,y) + \left(\frac{df}{dy}(t,y) \right)^2 f(t,y) + \frac{d^2f}{dy^2}(t,y)(f(t,y))^2 \end{aligned}$$

Merk nu op dat: $\frac{df}{dy}(t,y)F(t,y) = \frac{df}{dy}(t,y)\frac{df}{dt}(t,y) + \left(\frac{df}{dy^2}(t,y)\right)^2 f(t,y)$. We consider dat

$$:\frac{d^3y}{dt^3} = \Phi(t,y) + \frac{df}{dy}(t,y)F(t,y)$$

Vervolgens bepalen we de derde orde Taylor benadering van de exacte oplossing $y(t_{n+1})$ in het steunpunt t_n . Merk hierbij op dat $t_{n+1} - t_n = h$. De bepaling ziet er als volgt uit:

Lemma 4.2.3. Laat y een oplossing zijn van de differentiaalvergelijking $\frac{dy}{dt} = f(t,y)$. Nu geldt

$$y(t_{n+1}) = y(t_n) + (t_{n+1} - t_n)\frac{dy}{dt}(t_n) + (t_{n+1} - t_n)\frac{d^2y}{dt^2}(t_n) + \frac{(t_{n+1} - t_n)^3}{6}\frac{d^3y}{dt^3} + \mathcal{O}(h^4)$$

We passen nu het gegeven en Lemma's 4.1.5 en 4.2.2 toe. We vinden:

$$= y(t_n) + hf(t_n, y) + \frac{h^2}{2}F(t_n, y) + \frac{h^3}{6} \left(\Phi(t, y) + \frac{df}{dy}(t, y)F(t, y) \right) + \mathcal{O}(h^4)$$

Wij hebben nu een uitdrukking gevonden voor de exacte oplossing y. Vervolgens gaan wij hetzelfde doen voor de benaderde oplossing \tilde{y} . Het laatste lemma zal daarom een uitspraak doen over de twee dimensionale Taylorexpansie die we gebruiken om de tweede toestand k_2 te beschrijven.

Lemma 4.2.4. De twee dimensionale Taylorexpansie van $f(t_{n+1}, y_{n+1})$ in het steunpunt (t_n, y_n) is gelijk aan:

$$f(t_{n+1}, y_{n+1}) = f(t_n, y_n) + \gamma h F(t_n, y_n) + \frac{1}{2} \gamma^2 h^2 \Phi(t_n, y_n) + \mathcal{O}(h^3)$$

Bewijs We hebben vanwege de twee dimensionale Taylor expansie van $f(t_{n+1}, y_{n+1})$

dat:

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

= $f(t_n, y_n) + \gamma h\left(\frac{df}{dt}(t_n, y_n) + f(t_n, y_n)\frac{df}{dy}(t_n, y_n)\right)$
+ $\frac{1}{2}\left(\frac{d^2f}{dt^2}(t, y)\gamma^2h^2 + 2\frac{d^2}{dydt}(t, y)\gamma^2k_1h^2 + \frac{d^2f}{dy^2}(t, y)\gamma^2k_1^2h^2\right) + \mathcal{O}(h^3)$
= $f(t_n, y_n) + \gamma hF(t_n, y_n) + \frac{1}{2}\gamma^2h^2\Phi(t_n, y_n) + \mathcal{O}(h^3)$

Wij kunnen dit lemma nu gebruiken om de benaderde oplossing \tilde{y}_n van de RK2 methode te herschrijven tot:

$$\begin{split} \tilde{y}_{n+1} &= \tilde{y}_n + h(1 - \frac{1}{2\gamma})f(t_n, \tilde{y}_n) + \frac{1}{2\gamma}hf(t_n + \gamma h, \tilde{y}_n + \gamma f(t_n, \tilde{y}_n)h) \\ &= \tilde{y}_n + h(1 - \frac{1}{2\gamma})f(t_n, \tilde{y}_n) + \frac{1}{2\gamma}h\left(f(t_n, \tilde{y}_n)) + \gamma hF(t_n, \tilde{y}_n)\right) + \frac{1}{2}\gamma^2 h^2 \Phi(t_n, \tilde{y}_n)\right) + \mathcal{O}(h^3) \\ &= \tilde{y}_n + hf(t_n, \tilde{y}_n) + \frac{1}{2}h^2 F(t_n, \tilde{y}_n) + \frac{1}{4}\gamma h^3 \Phi(t_n, y_n) + \mathcal{O}(h^4) \end{split}$$

Nu dat wij een Taylor benadering van de benaderde oplossing \tilde{y}_n en van de exacte oplossing *y* hebben verkregen, kunnen wij de totale lokale truncatiefout van onze methode uitdrukken. Hiervoor zullen we eerst een bovengrens op Φ en $F * \frac{df}{dy}$ bepalen. Voor Φ vinden we het volgende.

$$\begin{aligned} |\Phi(t,y)| &= \left| \frac{d^2 f}{dt}(t,y) + \frac{d^2 f}{dy dt}(t,y) f(t,y) + \frac{d f}{dy}(t,y) \frac{d f}{dt}(t,y) \right| \\ &\leq \left| \frac{d^2 f}{dt}(t,y) \right| + 2 \left| \frac{d^2 f}{dy dt}(t,y) \right| * |f(t,y)| + \left| \frac{d f}{dy}(t,y) \right| * \left| \frac{d f}{dt}(t,y) \right| \end{aligned}$$

Wij eisen nu dat in een omgeving van (t, y) dat |f(t, y)| < M en $\left|\frac{d^{i+j}f}{dt^i dy^j}\right| < \frac{L^{i+j}}{M^{j-1}}$. Als we dit nu toepassen, dan vinden wij

$$|\Phi(t,y)| < ML^2 + 2L^2 * M + L * LM = 4ML^2$$
(9)

Wij gaan op een zelfde manier te werk voor $F \star \frac{df}{dy}$. We hebben eerder al gezien dat:

$$\left|F(t,y)\frac{df}{dy}(t,y)\right| = \left|\frac{df}{dy}(t,y)\frac{df}{dt}(t,y) + \left(\frac{df}{dy}(t,y)\right)^2 f(t,y)\right| \le \left|\frac{df}{dy}(t,y)\right| \left|\frac{df}{dt}(t,y)\right| + \left|\left(\frac{df}{dy}(t,y)\right)^2\right| |f(t,y)| \le \left|\frac{df}{dy}(t,y)\right| \le \left|\frac{df}{dy}(t,y)\right| + \left|\frac{df}{dy}(t,y)\right| \le \left|\frac{df}$$

Nu maken we gebruik van de eisen |f(t,y)| < M en $\left|\frac{d^{i+j}f}{dt^i dy^j}\right| < \frac{L^{i+j}}{M^{j-1}}$ die gelden in een omgeving van (t,y). We vinden nu:

$$|F * \frac{df}{dy}| < L * LM + (L)^2 * M = 2ML^2$$
(10)

Nu hebben wij alle ingredienten om 4.2.1 te bewijzen.

Bewijs van Stelling 4.2.1: Neem aan dat op tijdstap t_n de lokale truncatiefout gelijk is aan 0 Voor de lokale truncatiefout E_{n+1} hebben wij nu

$$E_{n+1} = |y(t_{n+1}) - \tilde{y}_{n+1}| \\ = \left| \left[y(t_n) + hf(t_n, y_n) + \frac{h^2}{2} F(t_n, y_n) + \frac{h^3}{6} \left(\Phi(t_n, y_n) + \frac{df}{dy}(t_n, y_n) F(t_n, y_n) \right) + \mathcal{O}(h^4) \right] \\ - \left[\tilde{y}_n + hf(t_n, \tilde{y}_n) + \frac{h^2}{2} F(t_n, \tilde{y}_n) + \frac{1}{4} \gamma h^3 \Phi(t_n, \tilde{y}_n) + \mathcal{O}(h^4) \right] \right|$$

Neem nu aan dat alle termen tot orde 2 elkaar opheffen. We hebben immers een orde twee methode. Verder beschouwen we de orde vier en hoger termen als verwaarloosbaar. Ten slotte is de lokale truncatiefout $E_n = |y_n - \tilde{y}_n| = 0$, oftewel $y_n = \tilde{y}_n$

$$= \left| \frac{h^{3}}{6} \left(\Phi(t_{n}, y_{n}) + \frac{df}{dy}(t_{n}, y_{n})F(t_{n}, y_{n}) \right) - \frac{1}{4}\gamma h^{3}\Phi(t_{n}, y_{n}) \right|$$

$$\leq h^{3} \left| \frac{1}{6} \Phi(t_{n}, y_{n}) - \frac{1}{4}\gamma \Phi(t_{n}, y_{n}) \right| + \frac{1}{6} \left| \frac{df}{dy}(t_{n}, y_{n})F(t_{n}, y_{n}) \right|$$

$$\leq h^{3} \left| \frac{1}{6} - \frac{1}{4}\gamma \right| |\Phi(t_{n}, y_{n})| + \frac{1}{6} \left| \frac{df}{dy}(t_{n}, y_{n})F(t_{n}, y_{n}) \right|$$

Wij kunnen nu onze bovengrenzen (9) en (10) gebruiken. Dit leidt tot:

$$E < h^{3} \left(\left| \frac{1}{6} - \frac{1}{4} \gamma \right| 4ML^{2} + \frac{1}{6} 2ML^{2} \right)$$
$$= h^{3} \left(4 \left| \frac{1}{6} - \frac{1}{4} \gamma \right| + \frac{1}{3} \right) ML^{2}$$

4.3 Stabiliteit

Wij hebben nu de fout afgeleid. Nu willen we een algemene formule afleiden voor het stabiliteitsgebied. Hierbij maken we gebruik van Definitie 4.1.1 en het bijbehorende Butcher Tableau. De Runge Kutta matrix A en gewichtenvector B worden dus weergegeven door:

$$A = \begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix} \text{ en } B = \begin{pmatrix} a & b \end{pmatrix}$$

Voor het stabiliteitsgebied gaan wij deze twee matrices toepassen in Stelling 3.2.1. Als dit gedaan wordt, dan is het resultaat gelijk aan de algemene stabiliteitsfunctie. De

stabiliteitsfunctie wordt weergegeven door:

$$R(\mu) = 1 + \mu * (a \quad b) * \left(\mathbb{I} - \mu * \begin{pmatrix} 0 & 0 \\ \beta & 0 \end{pmatrix} \right)^{-1} * \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$= 1 + (\mu a \quad \mu b) * \begin{pmatrix} 1 & 0 \\ -\beta \mu & 1 \end{pmatrix}^{-1} * \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$= 1 + (\mu a \quad \mu b) * \begin{pmatrix} 1 & 0 \\ \beta \mu & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$= 1 + \beta b \mu^{2} + (a + b) \mu$$

We zullen deze nog een keer in een gevolg zetten zodat we er later naar kunnen verwijzen:

Gevolg 4.3.1. De algemene stabiliteitsfunctie van de RK2 methode is gelijk aan:

$$R(\mu) = 1 + \beta b \mu^2 + (a+b)\mu$$

Nu wij een algemene stabiliteitsfunctie hebben, kunnen wij kijken naar de eerste en tweede orde RK2 methode Voor de tweede orde RK2 hadden we een aantal voorwaardes gesteld. Deze voorwaardes waren dat: a + b = 1 en $\beta b = \alpha b = \frac{1}{2}$. Als we deze voorwaarden invullen in onze algemene stabiliteitsfunctie, dan is deze gelijk aan:

$$R(\mu) = 1 + \mu + \frac{1}{2}\mu^2 \tag{11}$$

De stabiliteitsfunctie voor een tweede orde RK2 methode is dus altijd identiek. Als we $|R(\mu) < 1$ nu plotten, dan krijgen we Figuur 1



Figuur 1: Stabiliteitsgebied van de tweede orde RK2 Methode

Voor de eerste orde RK2 methoden moet gelden dat a + b = 1. Hiermee wordt de stabiliteitsfunctie dus gelijk aan:

 $1+\beta b\mu^2+\mu$

De stabiliteit van de eerste orde methode heeft dus een bepaalde invloed van de keuze van β . Wij zullen nu een paar stabiliteitsgebieden plotten met verschillende $b\beta$. Deze zijn te zien in 2

Wij zien dat wij het stabiliteitsgebied respectievelijk groter of kleiner kunnen krijgen door andere keuzes van $b\beta$. Merk op dat de eerste orde methode met $b\beta = \frac{1}{4}$ een groter stabiliteitsgebied heeft dan de tweede orde RK2 methode.



Figuur 2: Stabiliteitsgebieden met verschillende waarden voor $b\beta$

5 Specifieke RK-methoden

5.1 Forward Euler

De meest eenvoudige expliciete Runge Kutta methode is de **Forward Euler Methode**. Dit is een eerste orde Runge Kutta methode bedacht door Leonhard Euler en voor het eerst gepubliceerd in het boek *Institutionum calculi integralis*. Deze methode ziet er als volgt uit:

Definitie 5.1.1. Voor het beginwaardeprobleem $\begin{cases} y' = f(y,t) \\ y(t_0) = y_0 \end{cases}$ wordt de Forward Euler

methode gedefinieerd als

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Omdat de Forward Euler methode een eerste orde Runge Kutta methode is de lokale truncatiefout gelijk aan $\mathcal{O}(h^2)$. Dit betekent dat de lokale fout evenredig is aan de stapgrootte in het kwadraat. De Forward Euler is over het algemeen geen goede methode om een beginwaardeprobleem op te lossen. Het is daarentegen wel een goed begin om een idee te krijgen van de oplossing.

Het Butcher Tableau van de Forward Euler ziet er als volgt uit

De Runge Kutta matrix A is gelijk aan (0) en de vector B = (1). Wij bepalen nu het stabiliteitsgebied met behulp van Stelling 3.2.1. Wij vinden dat de stabiliteitsfunctie gelijk is aan:

$$R(\mu) = 1 + \mu (1) (\mathbb{I} - \mu (0))^{-1} \mathbb{1}$$

= 1 + (\mu) (\mu)^{-1} (1)
= 1 + \mu

Het stabiliteitsgebied wordt weergegeven door

$$|1 + \mu| < 1$$

Het stabiliteitsgebied is te zien in Figuur 3. Dit geeft een cirkel rondom (-1,0) met straal 1. Als de eigenwaarden van ons beginwaarde probleem binnen deze cirkel vallen, dan zal de Forward Euler methode convergeren naar de juiste oplossing.



Figuur 3: Stabiliteitsgebied van de Forward Euler Methode

5.2 Expliciete Midpuntregel

Een verbetering van de Forward Euler methode wordt gegeven door de **Expliciete Midpuntregel**. Dit is een tweede orde Runge Kutta methode met een toestand. Als wij kijken naar Definitie 4.1.1, dan is de Expliciete Midpuntregel gelijk aan de RK2 methode waarbij $\gamma = \frac{1}{2}$. Deze methode heeft daarom de volgende vorm

Definitie 5.2.1. Voor het beginwaardeprobleem $\begin{cases} y' = f(y,t) \\ y(t_0) = y_0 \end{cases}$ wordt de expliciete

midpuntregel als volgt gegeven

$$y_{n+1} = y_n + hf(t_n + \frac{h}{2}, y_n + \frac{h}{2}f(t_n, y_n))$$

Dit kunnen we op de volgende manier verwoorden. De benadering op tijd t_{n+1} is gelijk aan de benadering op tijdstip t_n plus de waarde van de functie f op het middelpunt $t_n + \frac{h}{2}$. Het Butcher Tableau van de Expliciete Midpuntregel wordt gegeven door:



De expliciete midpuntregel is een vreemde RK2 methode. De eerste toestand k_1 wordt alleen gebruikt om de tweede toestand te bepalen en heeft verder geen invloed op de benadering van de oplossing. Omdat de expliciete midpuntregel een tweede orde RK2 methode is, heeft hij het volgende stabiliteitsgebied

$$|1 + \mu + \frac{1}{2}\mu^2| < 1$$

5.3 Heun-methode

De volgende methode die we bespreken is de tweede orde RK2-methode met $\gamma = 1$. Deze methode heet de Methode van Heun of de expliciete trapeziumregel. De definitie is de volgende:

Definitie 5.3.1. Voor het beginwaardeprobleem $\begin{cases} y' = f(y,t) \\ y(t_0) = y_0 \end{cases}$ is de methode van Heun gedefinieerd als:

$$y_{n+1} = y_n + \frac{1}{2}h(f(t_n, y_n) + f(t_n + h, y_n + hf(t_n, y_n)))$$

Merk op dat $y_n + hf(t_n, y_n)$ is de Forward Euler methode. Deze methode maakt dus een schatting van y_{n+1} met de Forward Euler methode en gebruikt deze schatting vervolgens om een betere te maken.

Het Butcher Tableau van de methode van Heun is de volgende:

Omdat de methode van Heun een tweede orde RK2 methode is, hebben we dat zijn stabiliteitsfunctie gelijk is aan:

$$1+\mu+\frac{1}{2}\mu^2$$

5.4 **Ralston-methode**

Ten slotte bekijken we de tweede orde RK2-methode met $\gamma = \frac{2}{3}$. Deze methode heet de Ralston methode. Deze methode is als volgt gedefinieerd:

Definitie 5.4.1. Voor het beginwaardeprobleem $\begin{cases} y' = f(y,t) \\ y(t_0) = y_0 \end{cases}$ wordt de Ralston methode gedefinieerd door:

$$y_{n+1} = y_n + \frac{1}{4}hf(t_n, y_n) + \frac{3}{4}f(t_n + \frac{2}{3}h, y_n + \frac{2}{3}hf(t_n, y_n))$$

Het Butcher Tableau van de Ralston methode is als volgt:

Herinner dat de lokale truncatiefout E wordt afgeschat door:

$$|E| < h^3 \left(4 \left| \frac{1}{6} - \frac{1}{4}\gamma \right| + \frac{1}{3} \right) ML^2$$

Hierbij waren M en L bovengrenzen op de functie f in de omgeving van (t_n, y_n) . Een eenvoudige berekening laat zien dat deze bovengrens op E minimaal is, indien $\gamma = \frac{2}{3}$. Deze bovengrens wordt dan $|E| < \frac{1}{3}ML^2$ Om deze reden verwachten we van de Ralston methode een kleinere lokale truncatiefout dan bijvoorbeeld de Heun methode.

Daarentegen is de stabiliteit van de Ralston methode identiek aan de stabiliteit van de Heun methode. We hebben eerder laten zien dat tweede orde RK2 methode ieder dezelfde stabiliteitsfunctie hebben namelijk:

$$1+\mu+\frac{1}{2}\mu^2$$

5.5 **Eerste orde methode**

We hebben eerder gezien dat we met eerste orde RK2 methode nog een vrijheid hadden voor het stabiliteitsgebied. Daarom definieren we de volgende methode. Wij noemen deze methode de Orde-1 RK2 methode

Definitie 5.5.1. Voor het beginwaardeprobleem $\begin{cases} y' = f(y,t) \\ y(t_0) = y_0 \end{cases}$ wordt de **Orde-1 RK2**

methode gedefinieerd door:

$$y_{n+1} = y_n + \frac{1}{2}h(f(t_n, y_n) + f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(t_n, y_n)))$$

Het Butcher Tableau ziet er als volgt uit:

$$\begin{array}{c|cccc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ & \frac{1}{2} & \frac{1}{2} & \\ & & \frac{1}{2} & \frac{1}{2} \end{array}$$

~ I

We hebben deze methode zo geconstrueerd zodat het stabiliteitsgebied groter is dan de stabiliteit van de orde 2 RK2 methode. De stabiliteitsfunctie van deze kunnen we afleiden met behulp van 4.3.1. Hiervoor vullen we in a + b = 1 en $\beta b = \frac{1}{4}$. De stabiliteitsfunctie is nu de volgende:

$$R(\mu) = 1 + \mu + \frac{1}{4}\mu^2$$

Dit leidt tot het stabiliteitsgebied wat te zien is in Figuur 4. Dit stabiliteitsgebied is dus een cirkel rond (-2,0) met straal 2. We zullen bij het implementeren zien hoe deze methode verhoudt tot de methoden met hogere ordes.



Figuur 4: Stabiliteitsgebied van de Orde-1 RK2 Methode

6 RK toegepast op voorbeelden

Nu hebben wij de theorie van de RK2 methoden behandeld. In het volgende onderdeel zullen de RK2 methoden toegepast worden op voorbeelden uit de praktijk. Dit wordt gedaan in Matlab. Deze zullen wij vergelijken met de exacte oplossing van de praktijk gevallen. Hieruit zullen conclusies volgen over welke methode beter is in welke gevallen. De exacte oplossing bepalen we met de ODE45 functie van Matlab. Deze functie gebruikt een orde 4 Runge Kutta methode met 4 toestanden.

6.1 Chemische reactie

Het eerste praktijkgeval wat we gaan behandelen komt uit de scheikunde. Wij zullen kijken naar een differentiaalvergelijking die de temperatuur van een chemische reactie beschrijft. Dit is een eerste orde differentiaalvergelijking die er als volgt uit ziet.

$$T(t) = \alpha(2 - T(t))e^{\beta - \frac{\beta}{T(t)}}$$

met $T(0) = T_0$ de begintemperatuur. Hierbij is T de temperatuur als functie van de tijd t en zijn α en β reële constanten. In ons geval nemen we aan dat $\beta = \frac{1}{4}$ en $\alpha = 2$. Verder nemen we de begintemperatuur gelijk aan de kamertemperatuur, namelijk

20 graden Celsius. Verder hebben wij 500 stappen genomen en een tijdsinterval van [0,10] beschouwt. Dit betekent dat de tijdstapgrootte *h* gelijk is aan $10 \div 500 = 0.02$. De exacte oplossing van deze differentiaalvergelijking die we verkrijgen met behulp van de ODE45 methode is te zien in figuur 5



Figuur 5: Exacte Oplossing van de Chemische Reactie

Wij zien dat de oplossing exponentieel afneemt naar de waarde 2. Verder merken we op dat na t = 5 de functie ongeveer gelijk is aan 2. Wij zullen nu de differentiaalvergelijking proberen op te lossen met de numerieke die we hebben beschreven in Hoofdstuk 5. We beginnen met het toepassen van de Euler Forward methode. Dit leidt tot de volgende figuren.



Figuur 6: Resultaten van de Euler Forward methode

Het valt op dat de Forward Euler methode een redelijke benadering geeft voor de temperatuur. De fout is in het begin relatief groot maar groeit al snel naar 0 toe. Wij zullen nu kijken wat dit resultaat betekent voor hogere orde methode. Als wij de expliciete midpuntregel toepassen dan zien wij het volgende.



Figuur 7: Resultaten van de Expliciete Midpuntregel

De oplossing lijkt zich hetzelfde te gedragen als de oplossing van de Euler Forward methode. Deze methode doet het alleen nog beter. De fout is nog kleiner en groeit ook naar 0. Er verschijnt wel een vreemde oscillatie in de fout. Wij zullen nu kijken hoe de methode van Heun het doet voor onze temperatuurvergelijking.



Figuur 8: Resultaten van de methode van Heun

De methode van Heun voldoet aan hetzelfde patroon als de twee voorgaande methoden. Eerst hebben wij een kleine stijging in de fout, waarna de fout afneemt. Ook de oscillatie zien wij terugkomen en de snelheid waarmee de fout naar 0 groeit is nagenoeg hetzelfde als de snelheid van de fout van de expliciete midpuntregel. Wij zullen nu kijken hoe de laatste tweede orde RK2 methode het doet.



Figuur 9: Resultaten van de methode van Ralston

De methode van Ralston vertoont hetzelfde gedrag als de methode van Heun. De fout is over het algemeen klein en groeit al snel naar 0. Het verschil tussen de methode van Ralston en de methode van Heun kunnen wij hier nog niet duidelijk zien. Hierom zullen wij het verschil tussen deze twee plotten.



Figuur 10: Verschil tussen Heun en Ralston

Het verschil tussen de methode van Heun en de methode van Ralston is bij het toepassen op de temperatuurvergelijking verwaarloosbaar. Er zit een gering foutje aan het begin waarna het verschil snel naar nul groeit. Ten slotte gaan wij kijken of een kleinere orde hetzelfde resultaat geeft. Dit doen wij met de Orde-1 RK2 methode.



Figuur 11: Resultaten van de Orde-1 RK2 methode

Wij zien dat deze methode een grotere fout heeft dan de methode van Heun bijvoorbeeld. De fout heeft ongeveer dezelfde aantal tijdstappen nodig om bij nul te komen, maar de fout is over het algemeen wel groter dan de fout van de methode van Heun.

Wij hebben gezien dat we de temperatuur van de chemische reactie het beste kunnen benaderen met orde twee RK2 methoden. De methode van Ralston en de methode van Heun hebben over het algemeen de kleinste fout. Wel hebben deze fouten oscillaties waardoor er soms een stijging in fout plaatsvindt.

Ook de expliciete midpuntregel geeft een goede benadering. Dezelfde oscillatie trad wel op. Daarentegen geeft de Orde-1 RK2 methode een slechte benadering. De fout is op iedere tijdstap groter dan de fout van de orde 2 methoden.

6.2 Slinger

Ons volgende praktijkvoorbeeld komt uit de natuurkunde. We beschouwen een massa aan een uiteinde van een koord of staaf. Dit heet de slinger en deze slinger zal gaan bewegen als wij hem een beginpositie en snelheid geven. Wij willen deze beweging nu modelleren en bepalen. Het stelsel differentiaalvergelijkingen voor dit slingerexperiment wordt gegeven door:

$$\begin{pmatrix} v'(t) \\ s'(t) \end{pmatrix} = \begin{pmatrix} \alpha & 1 \\ -1 & -\alpha \end{pmatrix} \begin{pmatrix} v(t) \\ s(t) \end{pmatrix} = A * \begin{pmatrix} v(t) \\ s(t) \end{pmatrix}$$

Hierbij is v(t) de snelheid, s(t) de positie van de slinger en α een constante die de demping weergeeft. Als $\alpha > 0$ dan versnelt de slinger. Als $\alpha < 0$, dan zal de slinger uitdoven en stil komen te hangen in de ruststand (0,0). Verder geven we $(v(0), s(0)) = (v_0, s_0)$ de beginwaarden. Hierdoor hebben wij een beginwaardeprobleem gevonden voor de beweging van de slinger.

Voordat wij de resultaten van de numerieke methoden gaan bekijken, zullen eerst de eigenwaarden van de bovengenoemde matrix A bepaald worden. De eigenwaarden kunnen wij bepalen door middel van $det(A - \lambda I) = 0$ op te lossen voor λ . Dit leidt tot

de volgende vergelijking voor de eigenwaarden.

$$(\alpha - \lambda)^2 + 1 = 0 \Longrightarrow \lambda_{1,2} = \alpha \pm i$$

Nu de eigenwaarden van het slingerexperiment bepaald zijn, zullen wij de constanten α , v_0 en s_0 kiezen. Neem de beginsnelheid $v_0 = 0$ en de beginpositie $s_0 = 1$. Ten slotte nemen wij α gelijk aan -0.05. Wij kiezen dus voor een slinger die uitdooft naarmate de tijd vordert. De exacte oplossing van de slinger op tijdstappen tussen 0 en 50 is te zien in de Figuur 12. Wij plotten hierbij de functies v(t) en s(t) ten opzichte van de tijd.



Figuur 12: Exacte Oplossing van de slinger met dempingsfactor $\alpha = -0.05$

Wij zien een oscillatie ontstaan rond de ruststand. Naarmate de tijd vordert zien we dat beide functies v en s steeds kleinere amplitudes aannemen. Verder lijken beide functies periodiciteit te vertonen. Merk op dat de functies tussen -1 en 1 bestaan. Dit hebben wij verwacht aangezien de beginstand 1 is en de slinger demping heeft.

Voor stabiliteit van de numerieke methoden willen wij zorgen dat $h\lambda$ in het stabiliteitsgebied ligt. Om deze reden kiezen wij 1000 datapunten. De slinger beschouwen wij over het tijdsinterval [0,50]. Onze tijdstapgrootte *h* wordt daarom 50 ÷ 1000 = 0.05. Met deze tijdstapgrootte zorgen wij ervoor dat de slinger binnen ieder stabiliteitsgebied valt behalve die van de expliciete midpuntregel. Deze is onmogelijk omdat deze alleen stabiel is voor zuiver imaginaire $h\lambda$ tussen -i en *i*. Wij zullen bekijken nu de Forward Euler methode bekijken.



Figuur 13: Resultaten van de Forward Euler methode

Wij zien in uit de rechtergrafiek dat de Euler Forward te langzaam naar de ruststand groeit. Daarmee wordt de fout in het begin telkens groter. Als de exacte oplossing de ruststand bijna bereikt, dan zien we dat de fout ook kleiner wordt. Wij zien dat de fout maximaal 0.25 is. Voor ons slingerexperiment is dit dus geen goede benadering. De fout is relatief groot met de waarden waarmee wij werken. Wij kijken nu naar de eerste tweede orde methode: de expliciete midpuntregel.



Figuur 14: Resultaten van de expliciete midpuntregel

De benadering met behulp van de expliciete midpuntregel is een goede benadering voor de beweging van een slinger. In tegenstelling tot de Euler Forward is de fout van de expliciete midpuntregel klein en de methode lijkt redelijk snel te groeien naar de exacte oplossing. Na deze benadering gaan wij kijken naar de benadering met behulp van de methode van Heun



Figuur 15: Resultaten van de methode van Heun

Wij merken op dat de methode van Heun een goede benadering geeft voor de slingerbeweging. Als we de exacte oplossing vergelijken met de benadering met behulp van de methode van Heun, dan zien deze er identiek uit. Dit verschijnsel zien we ook terug in de fout. De fout is over het algemeen klein. Bovendien lijkt de fout naar nul te groeien. Na dit resultaat bekijken wij het gedrag van de andere tweede orde RK2 methode.



Figuur 16: Resultaten van de methode van Ralston

Dezelfde verschijnselen treden op bij de methode van Ralston. De fout is over het algemeen klein en ook de fout lijkt naar 0 te convergeren. Het verschil tussen de methode van Heun en de methode van Ralston is nog niet duidelijk zichtbaar uit dit resultaat. Om deze reden zullen wij het verschil tussen deze twee methoden in de volgende grafiek zetten.



Figuur 17: Verschil tussen Heun en Ralston

Het verschil tussen de methode van Heun en de methode van Ralston in nog kleiner dan bij de chemische reactie. In dit geval kunnen we de benaderingen als identiek beschouwen aangezien het verschil verwaarloosbaar is. Ten slotte bekijken we de Orde-1 RK2 methode.



Figuur 18: Resultaten van de Orde-1 RK2 methode

Wij zien uit de rechtergrafiek dat de orde één methode een slechter benadering geeft voor de beweging van de slinger. De benadering heeft een betere fout dan de Euler Forward, maar is slechter dan die van de orde twee methoden.

Conluderend hebben wij gezien dat de orde twee RK2 methoden een goede benadering geven voor de beweging van de slinger. De fout is over het algemeen klein en de fout convergeert naar nul. Daarentegen zagen we dat de orde één methode een veel grotere fout heeft dan de orde twee methoden. Om deze reden is de orde één methode een onpraktische methode voor de benadering van een slingerbeweging.

Verder zagen we dat de expliciete midpuntregel een goede benadering gaf. De benadering van de expliciete midpuntregel had een kleine fout en groeide relatief snel naar de juiste oplossing toe.

6.3 Predator Prey Model

Het volgende praktijkvoorbeeld komt uit de populatiedynamica. Wij hebben twee diersoorten. Eén soort jaagt op de anderen en heet daarom de predator-soort. Degene waarop gejaagd wordt heet de prey-soort. Wij willen nu de grootte van de populatie van deze twee soorten modelleren in de tijd. Dit soort model wordt een predator-prey model genoemd.

Wij zullen een eenvoudig model gebruiken om onze numerieke methoden te testen. Dit model is geformuleerd met behulp van de Lotka-Volterra vergelijkingen. Hierbij nemen wij aan dat de predator zich alleen kan voortplanten door interactie met de prey. De prey-soort heeft genoeg voeding om zichzelf voort te planten. Ten slotte nemen we aan dat de omgeving niet verandert en dus alle constanten die we introduceren onafhankelijk zijn van de tijd. Ons model ziet er nu als volgt uit:

$$\begin{cases} x_1'(t) = (\alpha - \beta x_2(t))x_1(t) \\ x_2'(t) = (\delta x_1(t) - \gamma)x_2(t) \end{cases}$$

Hierbij is x_1 de omvang van de prey-soort en x_2 de omvang van de predator soort. α beschrijft de groei van de prey-soort in de afwezigheid van de predator-soort. β representeert de afname van prey door toedoen van de predator-soort. γ is de afname van de predator-soort in de afwezigheid van de prey. Ten slotte is δ de groeifactor van de predator-soort.

Voor de numerieke tests die wij gaan uitvoeren, geven wij de constanten een waarde. Wij kiezen $\alpha = 0.8$, $\beta = 1.2$, $\gamma = 0.4$ en $\delta = 0.5$. Wij zullen het predator prey model beschouwen over het tijdsinterval [0,50].

Verder hebben wij de omvang van de populaties op het tijdstip t = 0 nodig. Wij kiezen deze als $x_1(0) = 0.9$ en $x_2(0) = 0.4$. Dit betekent dat de verhouding aan het begin 9:4 is. We nemen de beginwaarden klein zodat de functie een duidelijk patroon vertoont. Als wij met grote beginwaarden starten, dan zien we snel een daling in beide soorten. Daarna zullen beide soorten rond kleine waarden gedrag gaan vertonen. Dit gedrag is in een grafiek niet zichtbaar als we beginnen met grote waarden. Om deze reden kiezen wij de beginwaarden klein.

De exacte oplossing ziet er nu als volgt uit:



Figuur 19: Exacte oplossing van het Lotka Volterra model

Het valt op dat beide soorten periodiek gedrag vertonen. Dit valt als volgt te verklaren. Als de prey-soort maximaal is, dan zal een kleiner gedeelte van de predator-soort sterven. Er is immers genoeg te eten. Dit betekent een afname in de prey-soort. Nu naarmate de prey-soort kleiner wordt, zal ook een groter gedeelte van de predator-soort gaan sterven. Er is steeds minder prey om op te eten. Als de predator soort nu weer klein genoeg is, dan geeft dit meer ruimte voor de prey-soort om weer te gaan groeien. Als nu de prey-soort weer gaat toenemen, dan zal op een gegeven de predator-soort weer genoeg te eten hebben en ook weer toenemen. Dit leidt tot een eindeloze cykel en daarmee zal de oplossing periodiek gedrag vertonen.

Wij willen nu deze oplossing numeriek bepalen met de numerieke methoden. Hiervoor nemen wij 1000 stappen. Onze tijdstapgrootte h is nu $50 \div 1000 = 0.05$. Wij beginnen nu met het toepassen van de eerste numerieke methode: de Forward Euler methode.



Figuur 20: Resultaten van de Forward Euler methode

Wij merken op dat de Forward Euler aan het begin hetzelfde periodieke gedrag vertoont. Het verschil is dat naarmate de tijd vordert, de amplitude van de benaderingen groeit. Dit is in tegenstelling met de exacte oplossing waar de amplitudes constant blijven. Dit zien we ook terugkomen in de fout. De fout groeit naarmate de tijd groter





Figuur 21: Resultaten van de expliciete midpuntregel

In tegenstelling tot de Forward Euler methode houden de benaderingen van de expliciete midpuntregel wel een constante amplitude over het tijdsinterval. Dit is dus duidelijk een verbetering van de Forward Euler methode. Wij zien ook dat de fout klein is in vergelijking met de Forward Euler. Laten we nu kijken naar de andere orde twee RK2 methoden. We beginnen met de methode van Heun



Figuur 22: Resultaten van de methode van Heun

Wij zien dat deze benadering nagenoeg identiek is aan de benadering met behulp van de expliciete midpuntregel. De benadering heeft een constante amplitude en de fout is opnieuw klein. We zien daarentegen wel dat de fout van de prey soort pas na één periode maximaal is. Daarna lijkt de fout langzaam naar nul te groeien. Laten we nu kijken of de methode van Ralston hetzelfde gedrag vertoont als de methode van Heun



Figuur 23: Resultaten van de methode van Ralston

Inderdaad zien wij hetzelfde gedrag als de methode van Heun. Dit zijn dus goede benaderingen voor ons predator-prey model. Wij kunnen nog geen verschil opmerken tussen de methoden van Heun en Ralston. Wij maken een plot van het verschil tussen Heun en Ralston voor het predator-prey model.



Figuur 24: Verschil tussen Heun en Ralston

We merken op dat het verschil tussen de methodes van Heun en Ralston zeer klein is. Het maakt in het begin geen verschil welke we gebruiken aangezien het verschil verwaarloosbaar is. We zien echter wel dat het verschil tussen de twee methoden groeit. Als wij verder in de tijd kijken, dan zal er wel een duidelijk verschil ontstaan tussen de methoden van Heun en Ralston. Ten slotte zullen wij kijken of een lagere orde RK2 methode een ander resultaat oplevert bij toepassing op ons model. Hiervoor gebruiken we de Orde-1 RK2 methode.



Figuur 25: Resultaten van de Orde-1 RK2 methode

Wij zien dat de orde één methode slechte benaderingen geeft. De benaderingen hebben toenemende amplitudes en daardoor zal de fout blijven groeien. Daarentegen is de fout aan het begin wel klein en voor de eerste 10 tijdstappen is dit een redelijke benadering voor het predator-prey model

Concluderend zien wij dat de orde twee RK2 methoden goede benaderingen geven voor het predator prey model. De fout lijkt naar nul te groeien en is over het algemeen klein. De Orde-1 RK2 methode geeft in de eerste tien tijdstappen redelijke benaderingen, maar over een langer tijdsinterval zal de fout groter worden en dus de benaderingen onbruikbaar.

Ten slotte geeft de expliciete midpuntregel een goede benadering over het gegeven tijdsinterval. Voor een éénstapsmethode heeft deze een kleine fout in tegenstelling tot de Forward Euler. Om deze reden geeft de expliciete midpuntregel goede benaderingen voor het predator-prey model

6.4 Lorenz stelsel

Het laatste praktijvoorbeeld wat we bespreken is het Lorenz-stelsel. Dit stelsel staat er om bekend om chaotisch gedrag te vertonen bij bepaalde keuzes van parameters en beginwaarden. Het Lorenz-stelsel bestaat uit het volgende stelsel van differentiaalvergelijkingen.

$$\begin{cases} x' = \sigma(y-x) \\ y' = x(\rho-z) - y \\ z' = xy - \beta z \end{cases}$$

waarbij σ , ρ en β reële constanten zijn. Het Lorenz-stelsel is een stelsel van nietlineaire differentiaalvergelijkingen. Het vindt gebruik in de meteorologie en de atmosfeerdynamica. Voor onze numerieke experimenten kiezen wij de parameters zoals Lorenz ze zelf heeft gekozen namelijk $\sigma = 10$, $\rho = 28$ en $\beta = \frac{8}{3}$. Verder zullen wij dit stelsel bekijken over het tijdsinterval [0,50] en kiezen de beginwaarden (x_0, y_0, z_0) = (1,1,1). De exacte oplossing van dit stelsel ziet er nu als volgt uit.



Figuur 26: Exacte oplossing van het Lorenz Stelsel

Wij zien het chaotische gedrag duidelijk terug in Figuur 26. De oplossing oscilleert enorm. Wij zullen voor de numerieke tests een tijdstapgrootte $h = 25 \div 1000 = 0.025$.

Verder zal voor dit stelsel niet gekeken worden naar de Euler Forward methode. De Euler Forward zal met dit chaotische gedrag een gigantische fout hebben na een aantal tijdstappen. Om deze reden is de Euler Forward waardeloos om dit stelsel op te lossen. Wij zullen daarom beginnen met kijken naar de expliciete midpuntregel.



Figuur 27: Resultaten van de expliciete midpuntregel

Wij zien dat de benadering met behulp van de expliciete midpuntregel hetzelfde chaotische gedrag vertoont als de exacte oplossing. Echter blijft de fout groot naarmate de tijd vordert. Wij zien wel dat voor de eerste vijf tijdstappen de fout significant minder is. Wij zullen nu kijken naar een andere orde twee RK2 methode: de methode van Heun.



Figuur 28: Resultaten van de methode van Heun

Hetzelfde verschijnsel treedt op als bij de expliciete midpuntregel. De oplossing vertoont het chaotische gedrag. Daarnaast blijft de fout weer de eerste paar tijdseenheden relatief klein, waarna de fout groeit en de benadering dus slechter wordt. Wij zullen de numerieke resultaten nu vervolgen door het toepassen van de methode van Ralston op het Lorenz stelsel.



Figuur 29: Resultaten van de methode van Ralston

Opnieuw zien we duidelijk dat het Lorenz stelsel chaotisch gedrag vertoont. De oplossing oscilleert enorm. De omvang van de fout blijft hetzelfde als bij de methode van Heun, maar de fout heeft een groot verschil. Dit maken wij duidelijk door het verschil tussen de methoden van Heun en Ralston te plotten.



Figuur 30: Verschil tussen Heun en Ralston

In het begin zijn de benaderingen vrijwel identiek, maar na een paar tijdstappen verschijnt een groot verschil tussen de methoden. Dit is in tegenstelling tot de vorige praktijkvoorbeelden waar de methoden van Ralston en Heun vrijwel identieke benaderingen gaven. Wij kunnen uit de resultaten van het Lorenz stelsel niet concluderen of de verkleinde bovengrens van de methode van Ralston een betere methode geeft.

Ten slotte bekijken wij de Orde-1 RK2 methode. Misschien geeft een lagere orde methode ook een goede benadering voor het Lorenz stelsel.



Figuur 31: Resultaten van de Orde-1 RK2 methode

Hetzelfde probleem treedt op bij deze benadering. De chaos is duidelijk terug te zien in de benadering en ook de fout is groot. Wij zien dat vooral ver in de tijd, deze methode een even goede benadering geeft als de tweede orde RK2 methoden.

Concluderend zijn de RK2 methode niet sterk genoeg om het Lorenz stelsel goed te benaderen. Het chaotische gedrag van het stelsel geeft problemen voor de fout van de benadering. De tweede orde methode hebben in het begin nog een relatief kleine fout, maar deze groeit al snel.

7 Conclusie

Wij hebben gezien dat de RK2 methoden goede methoden zijn voor eenvoudige differentiaalvergelijkingen. Voorbeelden als de slinger en de temperatuurvergelijking worden goed benaderd door de tweede orde RK2 methoden. De fout was klein voor deze gevallen. De orde één methoden daarentegen gaf een slechte benadering. De stabiliteit van deze methode mag dan wel groter zijn, maar dit had uiteindelijk weinig invloed op de benadering.

Echter zagen wij wel dat een stelsel wat veel chaotisch gedrag vertoont niet goed benaderd kan worden door de RK2 methoden. Het Lorenz stelsel bijvoorbeeld had een gigantische fout.

Voor een vervolg op deze scriptie kan er onderzoek gedaan worden naar Runge Kutta methoden met meer toestanden. Dit zal betekenen dat de methoden hogere ordes aan kunnen nemen en potentieel betere benaderingen geven.

Verder hebben wij het verschil tussen de methode van Ralston en Heun niet duidelijk gezien. Men zou een experiment kunnen vinden waarmee dit verschil duidelijk wordt.

Referenties

- [1] Uri M. Ascher en Chen Grif. A First Course in Numerical Methods. SIAM, 2011.
- [2] Peter Knabner Christof Eck Harald Garcke. *Mathematical Modelling*. Springer, 2007.
- [3] Jason Frank. Stability of Runge Kutta Methods.
- [4] Tereza Koulikova. "*Runge Kutta Methods*". Masterscriptie. BRNO University of Technology, 2017.
- [5] Tristan van Leeuwen. Numerieke Wiskunde.
- [6] Anthony Ralston. *Runge-Kutta Methods with Minimum Error Bounds*. Hoboken, New Jersey.
- [7] L.F. Shampine. *Stability of the Leapfrog/Midpoint Method*. Southern Methodist University, Dallas, TX 75275.

Appendix

Listing 1: Phantom_kspace

```
close all
clear all
%N = aantal stappen
%T = einde van tijd interval
%B is the initial value
%f is the function
N=1000;
T=25;
```

```
%De Slinger
q = -.05;
fS1 = @(t, y) [q, 1; -1, q] * y;
BS1= [0,1];
%Chemische Reactie
a = 1/4;
b = 2;
fChe = @(t, y) a * (2-y) * exp((b-b/y));
BChe = 20;
%Lorenz-Stelsel
rho = 28;
sigma = 10;
iota = 8/3;
fLor = @(t,y) [sigma*(y(2)-y(1)); y(1)*(rho-y(3)); y(1)*y(2)-iota*y(3)];
BLor = [1;1;1];
%Predator-Prey Model
alpha = 0.8;
beta = 1.2;
gamma = 0.4;
delta = 0.5;
fPP=@(t, y) [alpha*y(1) - beta*y(1)*y(2); delta*y(1)*y(2) - gamma*y(2)];
BPP = [0.9, 0.4];
%Definieer variabelen
h=T/N;
t = [0:N] * h;
%Kies het probleem waarmee we willen werken(Che, Sl, Lor, PP)
f=fLor:
B=BLor;
%Exacte oplossing m.b.v ODE45
tspan = 0:h:T;
[tRS, yRS] = ode45(f, tspan, B);
figure ();
plot(t,yRS);
grid on
title('Echte_Oplossing')
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
%legend('x_1(t)', 'x_2(t)');
```

```
legend('x(t)','y(t)','z(t)')
yRS=yRS ';
%figure()
%plot3(yRS(:,1),yRS(:,2),yRS(:,3));
%Forward Euler
yFE=zeros(numel(B),N);
yFE(:,1) = B(:);
FoutFE=zeros(numel(B),N+1);
for i = 1:N
    yFE(:, i+1)=yFE(:, i)+h.*f(t(i), yFE(:, i));
    FoutFE(:, i)=abs(yRS(:, i)-yFE(:, i));
end
FoutFE(:,N+1)=abs(yRS(:,N+1)-yFE(:,N+1));
figure()
plot(t,yFE)
grid on
title('Euler_Forward');
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
figure()
plot(t,FoutFE)
grid on
title ('Fout_Euler_Forward');
xlabel('t');
%ylabel('T');
```

```
%legend('v(t)', 's(t)');
%legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
```

```
%Explicite Midpuntregel
yEM=zeros(numel(B),N);
yEM(:,1)=B(:);
FoutEM=zeros(numel(B),N+1);
```

```
for i = 1:N
    k1=f(t(i),yEM(:,i));
    yEM(:,i+1)=yEM(:,i)+h.*f(t(:,i)+h/2,yEM(:,i)+k1*h/2);
    FoutEM(:,i)=abs(yRS(:,i)-yEM(:,i));
```

```
end
FoutEM(:, N+1)=abs(yRS(:, N+1)-yEM(:, N+1));
figure ()
plot(t,yEM)
title ('Expliciete Midpuntregel')
grid on
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
%legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
%Plot de fout
figure()
plot(t,FoutEM)
grid on
title('Fout_Expliciete_Midpuntregel');
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
%Heun's Methode
yHM=zeros(numel(B),N);
yHM(:, 1) = B(:);
FoutHM=zeros (numel(B), N+1);
for i = 1:N
    k1 = f(t(i), yHM(:, i));
    yHM(:, i+1)=yHM(:, i)+h/2.*(k1+f(t(:, i)+h, yHM(:, i)+k1*h));
    FoutHM(:, i)=abs(yRS(:, i)-yHM(:, i));
end
FoutHM(:, N+1)=abs(yRS(:, N+1)-yHM(:, N+1));
figure ()
plot(t,yHM)
grid on
title ('Methode_van_Heun')
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
```

%Plot de fout

```
figure()
plot(t,FoutHM)
grid on
title('Fout_Methode_van_Heun');
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
\% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
%Ralston Methode
yRM=zeros(numel(B),N);
yRM(:,1) = B(:);
FoutRM=zeros (numel(B), N+1);
for i = 1:N
    k1 = f(t(i), yRM(:, i));
    yRM(:, i+1)=yRM(:, i)+1/4*h*k1+3*h/4.*f(t(:, i)+2*h/3, yRM(:, i)+k1*2*h/3);
    FoutRM(:, i) = abs(yRS(:, i)-yRM(:, i));
end
FoutRM(:, N+1)=abs(yRS(:, N+1)-yRM(:, N+1));
figure()
plot(t,yRM)
grid on
title ('Methode, van, Ralston')
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)','y(t)','z(t)')
%Plot de fout
figure ()
plot(t,FoutRM)
grid on
title('Fout_Methode_van_Ralston_');
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
%Verschil Heun en Ralston
DifHR=zeros(numel(B), N+1);
for i = 1:N+1
```

```
DifHR (:, i)= abs (yRM(:, i)-yHM(:, i));
end
figure()
plot(t,DifHR)
grid on
title ('Verschil_tussen_Heun_en_Ralston')
xlabel('t')
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
%Orde-1 RK2 methode
yO1=zeros(numel(B),N);
yO1(:,1)=B(:);
FoutO1=zeros (numel(B), N+1);
for i = 1:N
    k1=f(t(i), yO1(:, i));
    yO1(:, i+1)=yO1(:, i)+h/2.*(k1+f(t(:, i)+h/2, yO1(:, i)+k1*h/2));
    FoutO1(:, i)=abs(yRS(:, i)-yO1(:, i));
end
FoutO1(:,N+1)=abs(yRS(:,N+1)-yO1(:,N+1));
figure ()
plot(t, yO1)
grid on
title('Orde-1_RK2_methode')
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
%legend('x_1(t)', 'x_2(t)');
legend('x(t)','y(t)','z(t)')
%Plot de fout
figure()
plot(t, FoutO1)
grid on
title('Fout_Orde-1_RK2_methode');
xlabel('t');
%ylabel('T');
%legend('v(t)', 's(t)');
% legend('x_1(t)', 'x_2(t)');
legend('x(t)', 'y(t)', 'z(t)')
```