

Numerieke simulatie van differentiaalvergelijkingen

Giovanni Samaey

Les 4

Stijke vergelijkingen

Oplossen van niet-lineaire stelsels bij impliciete methodes

Recap: Lineaire multistapmethodes

- Beginwaardeprobleem:

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad t \geq t_0, \quad \mathbf{y}(t_0) = \mathbf{y}_0$$

- We zoeken een benaderende oplossing op tijdstip $t > t_0$

$$\mathbf{y}_n \approx \mathbf{y}(t_n), \quad t_n = t_0 + n h$$

- Lineaire multistapmethode berekent benadering als

$$\sum_{m=0}^s a_m \mathbf{y}_{n+m} = h \sum_{m=0}^s b_m \mathbf{f}(t_{n+m}, \mathbf{y}_{n+m})$$

- Bedoeling: hogere orde door hergebruik van punten
- Probleem: niet altijd convergent door multistapkarakter

Recap: Runge-Kutta methodes

$$\xi_j = \mathbf{y}_n + h \sum_{i=1}^{j-1} a_{j,i} \mathbf{f}(t_n + c_i h, \xi_i), \quad j = 1, \dots, \nu$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{\nu} b_j \mathbf{f}(t_n + c_j h, \xi_j)$$

Terminologie: RK matrix, RK knopen, RK gewichten, RK tableaux

Voorbeelden: (uitgewerkt op bord – met kleur)

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \\ \mathbf{b}^\top & \end{array}$$

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array},$$

$$\begin{array}{c|cc} 0 & & \\ \frac{2}{3} & \frac{2}{3} & \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

and

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

Het probleem van stijfheid

- Beschouw het stelsel differentiaalvergelijkingen

$$\mathbf{y}' = \Lambda \mathbf{y}, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad \Lambda = \begin{bmatrix} -100 & 1 \\ 0 & -1/10 \end{bmatrix}.$$

- Exacte oplossing $\mathbf{y}(t) = c_1 e^{-100t} \mathbf{v}_1 + c_2 e^{-t/10} \mathbf{v}_2$

$$\Lambda = V D V^{-1}, \quad V = \begin{bmatrix} 1 & 1 \\ 0 & 999/10 \end{bmatrix}, \quad D = \begin{bmatrix} -100 & 0 \\ 0 & -1/10 \end{bmatrix}$$

- Los op met de voorwaartse Eulermethode

$$\mathbf{y}_n = (I + h\Lambda)^n \mathbf{y}_0$$

- Numerieke oplossing:

$$\mathbf{y}_n = c_1 (1 - 100h)^n \mathbf{v}_1 + c_2 (1 - h/10)^n \mathbf{v}_2$$

Stijve vergelijkingen

- Illustratie net code op Toledo: niet op te lossen met gepaste beginvoorwaarde !
- Eigenschappen van de vergelijking zijn van belang:
 - Snelle componenten (kleine tijdstap nodig)
 - Trage componenten (veel tijdstappen nodig)
 - Simulatie is dus te duur voor een gevraagde nauwkeurigheid
- Eigenschappen van de methode zijn van belang:
 - Voorwaartse Eulermethode heeft problemen
 - Trapeziumregel heeft geen problemen

Het probleem van stijfheid

- Beschouw het stelsel differentiaalvergelijkingen

$$\mathbf{y}' = \Lambda \mathbf{y}, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad \Lambda = \begin{bmatrix} -100 & 1 \\ 0 & -1/10 \end{bmatrix}.$$

- Exacte oplossing $\mathbf{y}(t) = c_1 e^{-100t} \mathbf{v}_1 + c_2 e^{-t/10} \mathbf{v}_2$
- Los op met trapeziumregel $\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{1}{2} h \Lambda (\mathbf{y}_n + \mathbf{y}_{n+1})$

- Numerieke oplossing: $\mathbf{y}_n = \left(\frac{I + h\Lambda}{I - h\Lambda} \right)^n \mathbf{y}_0$

$$\mathbf{y}_n = c_1 \left(\frac{1 - 50h}{1 + 50h} \right)^n \mathbf{v}_1 + c_2 \left(\frac{1 - h/20}{1 + h/20} \right)^n \mathbf{v}_2$$

Een informele definitie van stijfheid

Een stelsel differentiaalvergelijkingen is stijf indien een numerieke simulatie met sommige methodes een excessief kleine staplengte vereist, enkel en alleen om instabiliteit te vermijden, zonder dat dit nodig is voor de nauwkeurigheid.

- Een mogelijke grootheid: stijfheidsratio (verhouding tussen grootste en kleinste eigenwaarde)
- In voorbeeld is dit 1000; in het echt vaak 10^{10} - 10^{15}

Lineaire scalaire *testvergelijking*

- Beschouw de scalaire lineaire vergelijking

$$y' = \lambda y, \quad t \geq 0, \quad y(0) = 1, \quad \boxed{\lambda \in \mathbb{C}}$$

- Exacte oplossing

$$y(t) = \exp(\lambda t), \quad \lim_{t \rightarrow \infty} y(t) = 0 \iff \operatorname{Re} \lambda < 0$$

- Belang van deze vergelijking

- Eenvoudig te analyseren
 - Veralgemeenbaar naar stelsels
 - Veralgemeenbaar naar niet-lineaire systemen

Lineair stabilitetsgebied \mathcal{D}

- In elke tijdsdiscretisatie komen h en λ samen voor, bvb. voorwaartse Euler

$$y_{n+1} = (1 + h\lambda) y_n, \quad y_n = (1 + h\lambda)^n$$

- Lineair stabilitetsgebied \mathcal{D} van een tijdsintegratiemethode is de verzameling van alle getallen $h\lambda$ zodanig dat

$$\lim_{n \rightarrow \infty} y_n = 0$$

- Voor voorwaartse Euler is het stabilitetsgebied een cirkel met straal 1 en middelpunt $(-1,0)$ (figuur zie bord)

$$\mathcal{D}_{\text{Euler}} = \{z \in \mathbb{C} : |1 + z| < 1\}$$

Veralgemening naar stelsels

- Beschouw vergelijking

$$\mathbf{y}' = \Lambda \mathbf{y}, \quad t \geq 0, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad \Lambda \in \mathbb{C}^{d \times d}$$

- Eigenwaardenontbinding geeft

$$\Lambda = V D V^{-1}, \quad D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$$

- Numerieke oplossing wordt dan

$$\mathbf{y}_n = \sum_{k=1}^d (1 + h\lambda_k)^n \mathbf{x}_k$$

- De maximale tijdstap wordt bepaald door de *stijfste component* (eigenwaarde met meest negatieve reëel deel)

Veralgemening naar niet-lineaire systemen

- Beschouw de niet-lineaire vergelijking

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad t \geq 0, \quad \mathbf{y}(0) = \mathbf{y}_0$$

- Lokaal goed benaderd door (let op: fout in boek).

$$\mathbf{y}' = \mathbf{f}(t_n, \mathbf{y}_n) + J_n(\mathbf{y} - \mathbf{y}_n), \quad J_n := \frac{\partial \mathbf{f} (t_n, \mathbf{y}_n)}{\partial \mathbf{y}}$$

- Eigenwaarden van Jacobiaan zijn

$$\lambda_{n,1}, \lambda_{n,2}, \dots, \lambda_{n,d}$$

- Stabiliteitsvoorwaarde verandert elke tijdstap

$$h\lambda_{n,1}, h\lambda_{n,2}, \dots, h\lambda_{n,d} \in \mathcal{D}$$

Stabiliteitsgebied van trapeziumregel

- Trapeziumregel voor de lineaire testvergelijking

$$y_{n+1} = y_n + \frac{h}{2}\lambda(y_n + y_{n+1}), \quad y_n = \left(\frac{1 + \frac{1}{2}h\lambda}{1 - \frac{1}{2}h\lambda}\right)^n$$

- Daaruit volgt stabiliteitsgebied

$$\mathcal{D}_{\text{TR}} = \left\{ z \in \mathbb{C} : \left| \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right| < 1 \right\}$$

- Makkelijk in te zien dat (zie kennisclip)

$$\mathbb{C}^- := \{z \in \mathbb{C} : \operatorname{Re} z < 0\} \equiv \mathcal{D}_{\text{TR}}$$

Definitie van A-stabiliteit van methode

- Een methode is A-stabiel als haar stabiliteitsgebied het linker halfvlak omvat:

$$\mathbb{C}^- \subseteq \mathcal{D}$$

- Dit betekent dat er geen enkele stabiliteitsrestrictie staat op de stapgrootte. Stapgrootte kan bepaald worden op basis van enkel nauwkeurigheidseisen.
- Welke methodes zijn A-stabiel ?
 - Trapeziumregel is A-stabiel.
 - Voorwaartse Euler is niet A-stabiel.
 - Algemene uitspraken mogelijk ?

Stabiliteitsfunctie van RK-methodes

- Runge-Kuttamethode toepassen op lineaire testvergelijking geeft (zie kennisclip)

$$y_{n+1} = \left[1 + h\lambda \mathbf{b}^T (I - h\lambda A)^{-1} \mathbf{1} \right] y_n$$

- Beschouw $\mathbb{P}_{\alpha/\beta}$ de verzameling van rationale functies (veelterm van graad α in teller en graad β in noemer).
- Lemma 1: Elke Runge-Kuttamethode voor de testvergelijking kan geschreven worden als

$$y_n = [r(h\lambda)]^n, \quad r \in \mathbb{P}_{\nu/\nu}$$

- Voor een *expliciete* methode is de stabiliteitsfunctie $r \in \mathbb{P}_\nu$ (een veelterm)

Lineair stabiliteitsgebied van RK-methodes

- Lemma 2: Wanneer een numerieke methode voor de testvergelijking de volgende oplossing geeft

$$y_n = [r(h\lambda)]^n, \quad r \text{ willekeurig}$$

dan is het stabiliteitsgebied van die methode gegeven als

$$\mathcal{D} = \{z \in \mathbb{C} : r(z) < 1\}$$

- Gevolg: Geen enkele *expliciete* RK methode kan A-stabiel zijn. (Reden: $r \in \mathbb{P}_\nu$.)
- De stabiliteitsfunctie uitgewerkt voor een specifiek schema (zie kennisclip).
- Schema's kunnen identieke stabiliteitsfuncties hebben !

A-stabiliteit van RK methodes nagaan

- Stabiliteitsvoorwaarde $|r(z)| < 1$ nagaan met brute kracht. Bvb. via transformatie naar poolcoördinaten. (Zie boek.)
- Gebruik een elegantere stelling om na te gaan of

$$\forall z \in \mathbb{C}^- : |r(z)| < 1$$

(de functie r is dan A-acceptabel.)

- Stelling: (voorbeeld, zie boek)

$$\forall z \in \mathbb{C}^- : |r(z)| < 1 \iff \begin{array}{l} \bullet \text{ Alle polen van } r \text{ liggen in } \mathbb{C}^+ \\ \bullet |r(it)| < 1, \quad \forall t \in \mathbb{R} \end{array}$$

Padé-benadering als stabiliteitsfunctie

- Voor een discretisatiemethode van orde p weten we dat

$$r(z) = \exp(z) + \mathcal{O}(z^{p+1})$$

- We definiëren de orde van de stabiliteitsfunctie aldus.

- Let op: niet noodzakelijk orde van methode !

- Stelling: Voor elke twee getallen α, β bestaat er een unieke stabiliteitsfunctie van orde $p = \alpha + \beta$

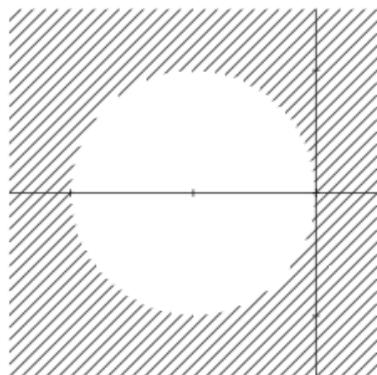
$$\hat{r}_{\alpha/\beta} = \frac{\hat{p}_{\alpha/\beta}}{\hat{q}_{\alpha/\beta}}, \quad \hat{q}_{\alpha/\beta}(0) = 1$$

- Stelling: De Padé-benadering $\hat{r}_{\alpha/\beta}$ is A-acceptabel iff

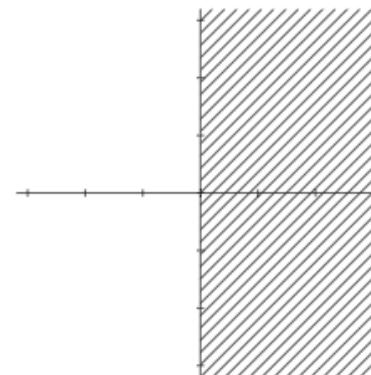
$$\alpha \leq \beta \leq \alpha + 2$$

Voorbeelden van stabilitiesgebieden

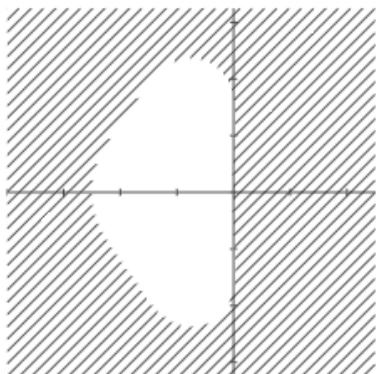
$$\hat{r}_{1/0}(z) = 1 + z$$



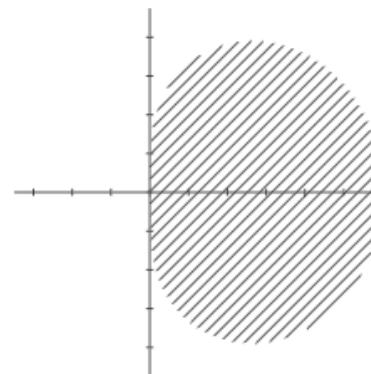
$$\hat{r}_{1/1}(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}$$



$$\hat{r}_{3/0}(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3$$



$$\hat{r}_{1/2}(z) = \frac{1 + \frac{1}{3}z}{1 - \frac{2}{3}z + \frac{1}{6}z^2}$$



LMS methodes: convergentie vs. stabiliteit

- Extra probleem: hangt af van oplossing op meerdere tijdstippen in het verleden.
- We krijgen dus parasitaire oplossingen.
- Voor convergentie moeten deze verdwijnen als $h \rightarrow 0$. Dit gebeurt wanneer nulpunten van de veelterm...

$$\rho(w) := \sum_{m=0}^s a_m w^m$$

- ... modulus kleiner dan 1 hebben.
- Stabiliteit gaat over $n \rightarrow \infty$, voor vaste h .
- Is er een vergelijkbare voorwaarde voor stabiliteit ?

Stabiliteit van lineaire multistapmethodes (1)

- Lineaire multistapmethode voor de testvergelijking

$$\sum_{m=0}^s a_m y_{n+m} = h\lambda \sum_{m=0}^s b_m y_{n+m}$$

- Herschrijf als lineaire differentievergelijking

$$\sum_{m=0}^s (a_m - h\lambda b_m) y_{n+m} = 0$$

- Definieer de karakteristieke veelterm

$$\eta(w; z) := \sum_{m=0}^s (a_m - b_m z) w^m$$

- De nulpunten w_1, w_2, \dots, w_q (meervoudigheid k_1, k_2, \dots, k_q) zullen belangrijk zijn voor de stabiliteit.

Intermezzo: lineaire differentievergelijkingen

- Oplossingen van de lineaire differentievergelijking ...

$$\sum_{m=0}^s (a_m - b_m z) y_{n+m} = 0$$

- ... hebben de vorm ...

$$x_n = \sum_{i=1}^q \left(\sum_{j=0}^{k_i-1} c_{i,j} n^j \right) w_i^n$$

- ... met w_1, w_2, \dots, w_q de nulpunten (meerv. k_1, k_2, \dots, k_q) ...
- ... van de karakteristieke veelterm

$$\eta(w; z) := \sum_{m=0}^s (a_m - b_m z) w^m$$

Lineair stabilitetsgebied van LMS methodes

- Beschouw de karakteristieke veelterm

$$\eta(w; z) := \sum_{m=0}^s (a_m - h\lambda b_m) w^m$$

- ... met nulpunten w_1, w_2, \dots, w_q .
- Het lineaire stabilitetsgebied van de LMS methode is

$$\mathcal{D} := \{z : |w_i(z)| < 1, i = 1, \dots, q(z)\}$$

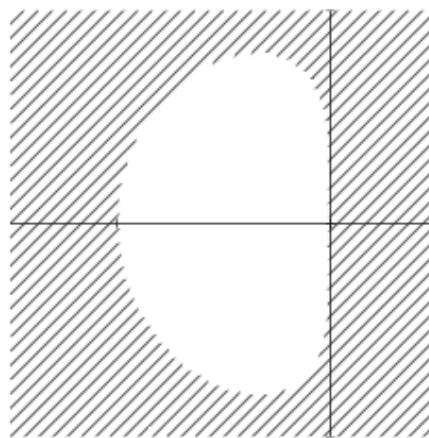
- Zoals voor RK methodes, is een LMS methode A-stabiel iff

$$\mathbb{C}^- \subseteq \mathcal{D}$$

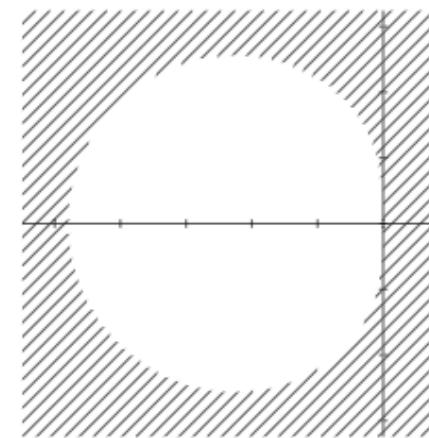
- Bekijk stabilitetsgebieden (figuur 4.3, p. 65)

Stabiliteitsgebieden van Adams methodes

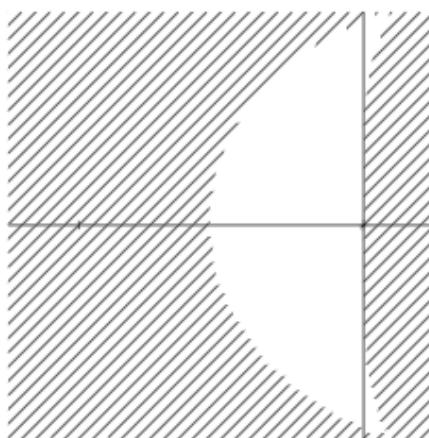
Adams–Bashforth, $s = 2$



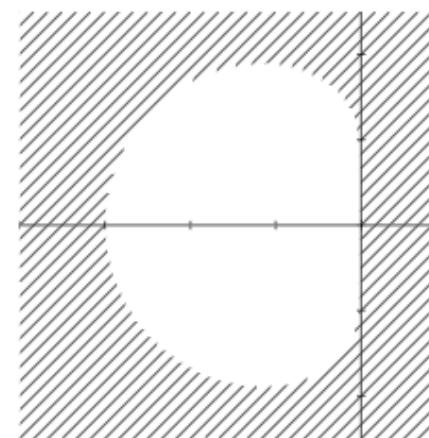
Adams–Moulton, $s = 2$



Adams–Bashforth, $s = 3$



Adams–Moulton, $s = 3$

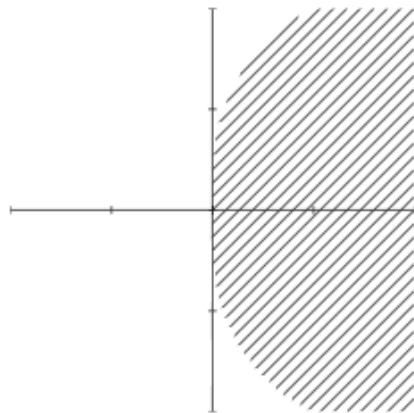


A-stabiliteit van LMS methodes

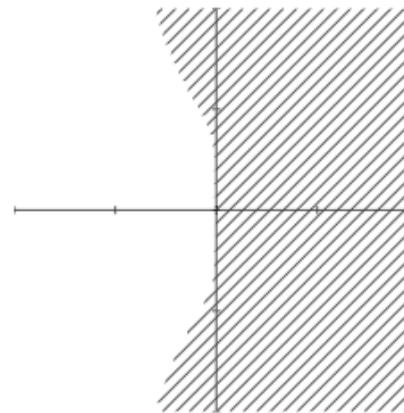
- Adams-Bashforth en Adams-Moulton methodes zijn verre van A-stabiel !
- Zijn er ook nooit voor ontworpen. BDF-formules wel...

Stabiliteitsgebieden van BDF methodes

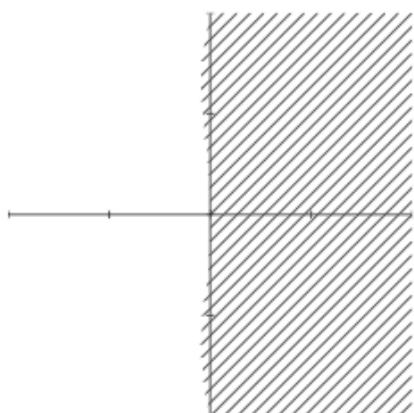
$s = 2$



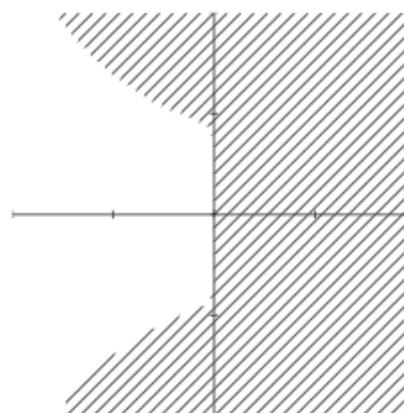
$s = 4$



$s = 3$



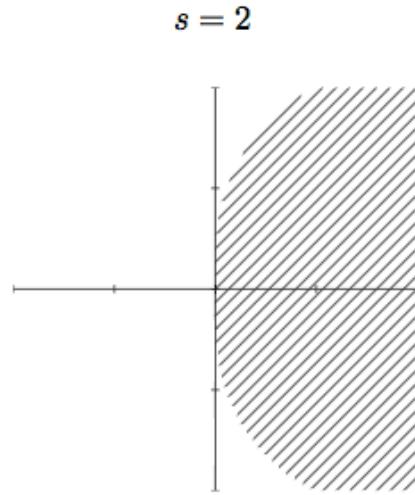
$s = 5$



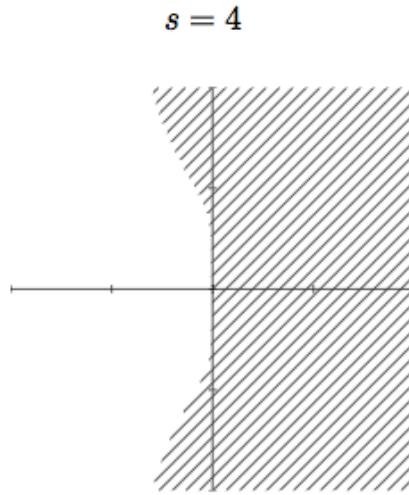
A-stabiliteit van LMS methodes

- Adams-Bashforth en Adams-Moulton methodes zijn verre van A-stabiel !
- Zijn er ook nooit voor ontworpen. BDF-formules wel...
- ... maar die lijken niet A-stabiel voor $s \geq 2$!!
- Stelling: De tweestaps-BDF methode is A-stabiel.
- Stelling (tweede barrière van Dahlquist): De hoogste orde voor een A-stabiele multistapmethode is 2.

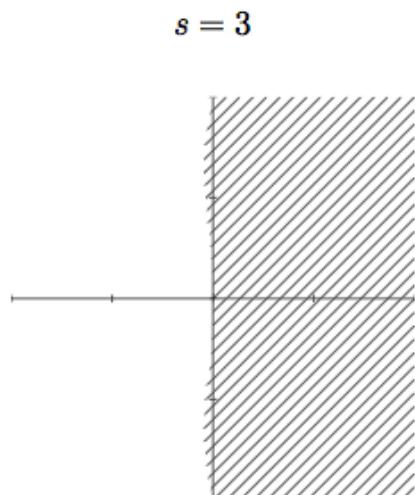
Zijn BDF-formules dan toch niet nuttig?



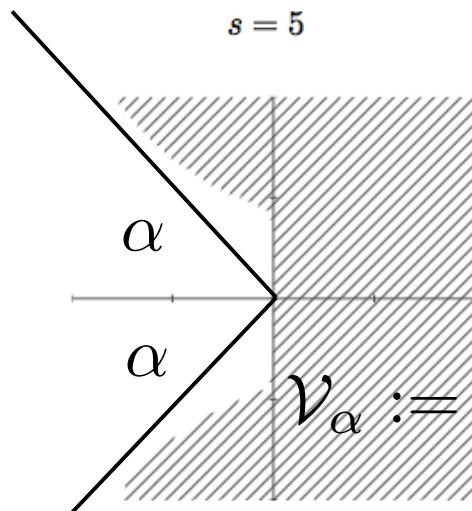
$s = 2$



$s = 4$



$s = 3$



$s = 5$

- Toch wel !
 - A-stabiliteit is vaak een overdreven eis.
 - $A(\alpha)$ -stabiliteit is vaak genoeg
- $$\mathcal{V}_\alpha \subseteq \mathcal{D} \subseteq \mathbb{C}^-$$
- Met
- $$\mathcal{V}_\alpha := \{\rho \exp(i\theta) : \rho > 0, |\theta + \pi| < \alpha\}$$

Enkele laatste bedenkingen

- Hoewel stijve vergelijkingen moeilijk handelbaar lijken, valt het nog mee...
- Stabiliteit impliceert dat fouten gedempt worden in volgende tijdstappen.
- Eens je de concepten kent, en de juiste methodes gebruikt (impliciete RK methodes, BDF), kan er weinig fout gaan.
- Dit zal niet langer waar zijn voor de problemen in Hfst. 5 !

Voor het examen

- Definitie van stabiliteitsgebied kennen, alsook karakterisatie ervan voor RK en LMS methodes.
- **Gedrag** van methodes herkennen en verklaren aan de hand van stabiliteitsgebieden en numerieke oplossingen.
- Een kwalitatief idee hebben van de grootte van stabiliteitsgebieden voor verschillende klassen van methodes.
- Een gepaste klasse van integratoren voorstellen voor een probleem (stijve problemen herkennen).

Impliciete lineaire multistapmethodes

- LMS methode $\sum_{m=0}^s a_m \mathbf{y}_{n+m} = h \sum_{m=0}^s b_m \mathbf{f}(t_{n+m}, \mathbf{y}_{n+m})$

- Splits bekenden van onbekenden γ

$$\mathbf{y}_{n+s} = h b_s \mathbf{f}(t_{n+s}, \mathbf{y}_{n+s}) + \boxed{\sum_{m=0}^{s-1} (b_m \mathbf{f}(t_{n+m}, \mathbf{y}_{n+m}) - a_m \mathbf{y}_{n+m})}$$

- Herschrijf als ...

$$\mathbf{w} = h \mathbf{g}(\mathbf{w}) + \beta$$

- Met hierin ...

$$\mathbf{w} = \mathbf{y}_{n+s}$$

$$\mathbf{g}(\mathbf{w}) = b_s \mathbf{f}(t_{n+s}, \mathbf{w})$$

$$\beta = \gamma$$

Impliciete Runge-Kuttamethodes

- RK methode

$$\xi_j = \mathbf{y}_n + h \sum_{i=1}^{\nu} a_{j,i} \mathbf{f}(t_n + c_i h, \xi_i), \quad j = 1, \dots, \nu$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{j=1}^{\nu} b_j \mathbf{f}(t_n + c_j h, \xi_j)$$

- Herschrijf als ...

$$\mathbf{w} = hg(\mathbf{w}) + \boldsymbol{\beta}$$

- Met hierin ...

$$\mathbf{w} = [\mathbf{w}_j]_{j=1}^{\nu}, \quad \mathbf{w}_j = \xi_j$$

$$g(\mathbf{w}) = [g_j(\mathbf{w})]_{j=1}^{\nu}, \quad g_j(\mathbf{w}) = \sum_{i=1}^{\nu} a_{j,i} f(t_n + c_i h, \mathbf{w}_i)$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_j]_{j=1}^{\nu}, \quad \boldsymbol{\beta}_j = \mathbf{y}_n$$

Impliciete methodes: iteratieve oplossing

- Algemene vorm van niet-lineair stelsel

$$\mathbf{w} = h\mathbf{g}(\mathbf{w}) + \boldsymbol{\beta}$$

- Maak een iteratie van de vorm

$$\mathbf{w}^{[i+1]} = \mathbf{s}(\mathbf{w}^{[i]})$$

- Eisen op de iteratie

- Convergent : $\mathbf{w}^{[i]} \rightarrow \hat{\mathbf{w}}$

- Lage kost per iteratie

- Convergentie is snel (weinig iteraties nodig)

Eerste idee: functionele iteratie

- Algemene vorm van niet-lineair stelsel

$$\mathbf{w} = h\mathbf{g}(\mathbf{w}) + \boldsymbol{\beta}$$

- Maak een functionele iteratie van de vorm

$$\mathbf{w}^{[i+1]} = h\mathbf{g}(\mathbf{w}^{[i]}) + \boldsymbol{\beta}$$

- Voorbeeld: Trapeziumregel

$$\mathbf{w}^{[i+1]} = \frac{1}{2}h\mathbf{f}(t_{n+1}, \mathbf{w}^{[i]}) + \left[\mathbf{y}_n + \frac{1}{2}h\mathbf{f}(t_n, \mathbf{y}_n) \right]$$

- Wanneer convergeert functionele iteratie ?

- Banach vaste-puntstelling (bewijs, zie kennisclip)

Convergentie van functionele iteratie

- Voorwaarde voor afleidbare $g(\cdot)$. We hebben dat ...

$$\|g(u) - g(v)\| \leq \left\| \frac{\partial g}{\partial w}(v^*) \right\| \|u - v\|$$

- En dus is nodig dat ...

$$h \left\| \frac{\partial g}{\partial w}(v^*) \right\| \leq 1$$

- Deze beperking op de stapgrootte is vergelijkbaar met die op de stap voor expliciete methodes voor stijve problemen
- Voorbeeld op bord: Trapeziumregel voor lineair probleem

$$\mathbf{w}^{[i+1]} = \frac{1}{2} h f(t_{n+1}, \mathbf{w}^{[i]}) + \left[\mathbf{y}_n + \frac{1}{2} h f(t_n, \mathbf{y}_n) \right]$$

(Gewijzigde) methode van Newton-Raphson

- Vertrek van een tweede-orde expansie

$$\begin{aligned}\mathbf{w} &= \boldsymbol{\beta} + hg(\mathbf{w}^{[i]} + (\mathbf{w} - \mathbf{w}^{[i]})) \\ &= \boldsymbol{\beta} + hg(\mathbf{w}^{[i]}) + h \frac{\partial g(\mathbf{w}^{[i]})}{\partial \mathbf{w}} (\mathbf{w} - \mathbf{w}^{[i]}) + \mathcal{O}\left(\|\mathbf{w} - \mathbf{w}^{[i]}\|^2\right)\end{aligned}$$

- Verwaarloos tweede-ordeterm en los op naar $\mathbf{w} - \mathbf{w}^{[i]}$

$$\mathbf{w}^{[i+1]} = \mathbf{w}^{[i]} - \left[I - h \frac{\partial g(\mathbf{w}^{[i]})}{\partial \mathbf{w}} \right]^{-1} \left[\mathbf{w}^{[i]} - \boldsymbol{\beta} - hg(\mathbf{w}^{[i]}) \right]$$

- We krijgen een specificatie van Newton-Raphson in onze context.

Eigenschappen van Newton-Raphson

- Voor voldoende kleine h : kwadratische convergentie
 - Volgt uit het laten vallen van de kwadratische term
 - Heeft niets te maken met stijfheid. (Voor lineair probleem convergeert de methode in 1 iteratie, hoe stijf ook...)
- Jacobiaanmatrix nodig in elke iteratie (=heel duur)
- In elke iteratie moet een lineair stelsel opgelost worden
 - Geen problemen met slechte conditie van die stelsels
 - Maar kan toch duur zijn in rekenkost
- We zullen dus een gewijzigde Newton-methode opstellen

Gewijzigde Newton-Raphson

- Newton-Raphson

$$\boldsymbol{w}^{[i+1]} = \boldsymbol{w}^{[i]} - \left[I - h \frac{\partial \mathbf{g}(\boldsymbol{w}^{[i]})}{\partial \boldsymbol{w}} \right]^{-1} \left[\boldsymbol{w}^{[i]} - \boldsymbol{\beta} - h \mathbf{g}(\boldsymbol{w}^{[i]}) \right]$$

- Gewijzigde Newton-Raphson

$$\boldsymbol{w}^{[i+1]} = \boldsymbol{w}^{[i]} - \left[I - h \frac{\partial \mathbf{g}(\boldsymbol{w}^{[0]})}{\partial \boldsymbol{w}} \right]^{-1} \left[\boldsymbol{w}^{[i]} - \boldsymbol{\beta} - h \mathbf{g}(\boldsymbol{w}^{[i]}) \right]$$

- Slechts 1 evaluatie van Jacobiaan per tijdstap
- Bij Gauss-eliminatie kunnen we LU-factorisatie opslaan
- Verlies van kwadratische convergentie

Convergentie van gewijzigde Newton-Raphson

- Gewijzigde Newton-Raphson

$$\boldsymbol{w}^{[i+1]} = \boldsymbol{w}^{[i]} - \left[I - h \underbrace{\frac{\partial \mathbf{g}(\boldsymbol{w}^{[0]})}{\partial \boldsymbol{w}}}_{\mathbf{J}} \right]^{-1} \beta \left[\boldsymbol{w}^{[i]} \mathbf{g}(\boldsymbol{w}^{[i]}) \right]$$

- Schrijf als functionele iteratie

$$\begin{aligned}\boldsymbol{w}^{[i+1]} &= h(I - h\mathbf{J})^{-1} \left[\mathbf{g}(\boldsymbol{w}^{[i]}) - \mathbf{J}\boldsymbol{w}^{[i]} \right] + (I - h\mathbf{J})^{-1} \beta \\ &= h\tilde{\mathbf{g}}(\boldsymbol{w}^{[i]}) + \tilde{\beta}\end{aligned}$$

- Convergentie bepaald door (zie dat stijfheid helpt !)

$$\frac{\|\tilde{\mathbf{g}}(\mathbf{u}) - \tilde{\mathbf{g}}(\mathbf{v})\|}{\|\mathbf{u} - \mathbf{v}\|} \leq \left\| [I - h\mathbf{J}]^{-1} \right\| \times \left\| \frac{\partial \mathbf{g}(\mathbf{z})}{\partial \boldsymbol{w}} - \mathbf{J} \right\|$$

Een geval waarin Newton-Raphson efficiënt is

- Newton-Raphson is snel als Jacobiaanmatrix ijl is
 - Snel opstellen van de Jacobiaan
 - Snel oplossen van de resulterende stelsels
- Bijvoorbeeld voor semidiscretisatie van partiële differentiaalvergelijkingen: reactie-diffusievergelijking

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2} + \varphi(u)$$

$$y_k \approx u(x_k, t), \quad y'_k = \frac{1}{\Delta x^2} (y_{k+1} - 2y_k + y_{k-1}) + \varphi(y_k)$$

- Uitwerking van ijle Jacobiaan aan bord

Het vinden van goede startwaarden

- Gewijzigde Newton-Raphson laat weliswaar slechtere startwaardes toe zonder convergentie te verliezen.
- Rekenkost is evenredig met het aantal iteraties
- Hoe beter de startwaarde, hoe minder iteraties nodig.
 - Buit de context van gewone differentiaalvergelijkingen uit.
 - Gebruik een expliciete methode van dezelfde orde als predictor
 - Impliciete methode heet dan de corrector

Het stoppen van de iteratie (1)

- Klinkt triviaal: stop wanneer de opeenvolgende updates niets meer bijbrengen

$$\left\| \mathbf{w}^{[i+1]} - \mathbf{w}^{[i]} \right\| \leq \epsilon$$

- Tolerantie ϵ zou dan kleiner moeten zijn dan δ , de tolerantie van het tijdstappen zelf
- Maar: we zijn een differentiaalvgl. aan het simuleren!
- Trage convergentie kan ook wijzen op een te grote tijdstap!
- Dan kan de iteratie gestopt worden, en hernomen met kleinere tijdstap.

Het stoppen van de iteratie (2)

- Iteratie tot convergentie
 - Maximaal aantal iteraties
 - Indien geen convergentie, hernemen met kleinere tijdstap
- PECE-methode
 - Gebruik voor predictie/correctie de methodes uit Milne device
 - Voer slechts 1 enkele iteratie uit
 - Indien tolerantie niet gehaald, hernemen met kleinere tijdstap
- Keuze hangt af van kost van veranderen stap vs. iteratie
- Stabiliteitsgebied wordt ook aangetast.

Stijve versus niet-stijve systemen

Niet-stijve systemen

- Functionele iteratie mogelijk (makkelijk te herstarten)
 - Stapgrootte veranderen is goedkoop (enkel remeshing)
 - Grootte stabiliteitsgebied is niet zo belangrijk
- PECE methode is superieur

Stijve systemen

- Gewijzigde Newton-Raphson nodig
 - Stapgrootte veranderen is duur (LU-factorisatie)
 - Grootte stabiliteitsgebied is heel belangrijk
- Itereer tot convergentie

Voor het examen

- Samenhang kunnen aangeven tussen
 - Stabiliteit en nauwkeurigheid van methodes
 - Praktische overwegingen, zoals selectie van stapgrootte, oplossen van niet-lineaire stelsels
 - Eigenschappen van probleem (stijf/niet stijf)
 - Keuze van gepaste set methodes
- Een verantwoorde keuze kunnen maken van een bepaalde (set van) methodes voor een gegeven probleem, op basis van bovengenoemde aspecten.