

RA
DUPLICAAT

stichting
mathematisch
centrum



REKENAFDELING

NR 15/71

APRIL

RA

P.W. HEMKER
LINEAIRE MEERSTAPSMETHODEN MET VARIABELE STAPLENGTE

2e boerhaavestraat 49 amsterdam

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Voorwoord

Dit rapport beschrijft enig onderzoek dat gedaan werd in het kader van de werkgroep stijve differentiaalvergelijkingen van het Mathematisch Centrum.

De schrijver wil zijn dank betuigen aan de deelnemers van deze werkgroep voor vele inspirerende opmerkingen en in het bijzonder aan Dr. T.J. Dekker en Dr. P.J. v.d. Houwen voor hun waardevolle suggesties.

INHOUD

	pag.
<u>Hoofdstuk I. De methoden</u>	
§1. Inleiding	1
§2. Polynoombenaderingen van een functie	3
§3. De methode gebaseerd op integratie	7
§4. Het oplossen zonder gebruikmaking van de Jacobiaan	9
§5. Het oplossen door gebruik te maken van de Jacobiaan	11
§6. De methode gebaseerd op differentiatie	14
§7. Het oplossen zonder gebruikmaking van de Jacobiaan	15
§8. Het oplossen door gebruik te maken van de Jacobiaan	16
§9. Stelsels vergelijkingen	17
<u>Hoofdstuk II. De stabiliteit</u>	
§10. Inleiding	23
§11. Stabiliteit voor variabele staplengte methoden	24
§12. De stabiliteit voor $n = 0$ en $n = 1$	28
§13. De relatie tot de lineaire meerstapsmethoden met vaste staplengte	34
§14. Stabiliteitsgebieden	35
<u>Hoofdstuk III. ALGOL 60-procedures en testresultaten</u>	
§15. De procedure multistep	48
§16. De procedures multistep 1 en multistep n	54
§17. Twee lineaire testvoorbeelden	60
§18. Een stijve differentiaalvergelijking uit de biochemie	64
<u>Appendix. Notatie en voorbeelden</u>	68
<u>Literatuur</u>	73

Hoofdstuk I. De Methoden

§1. Inleiding

De methoden die ten grondslag liggen aan de hier aangeboden algoritmen voor het integreren van gewone differentiaalvergelijkingen, zijn een uitbreiding van de klassieke lineaire meerstapsmethoden. Deze meerstapsmethoden voor het oplossen van de differentiaalvergelijking

$$y^{(d)} = f(x, y, y', y'', \dots, y^{(d-1)}); y(0) = y_0$$

waarvan de oplossing wordt voorgesteld door $y = y(x)$ zijn gebaseerd op benaderingen d.m.v. polynomen.

We kunnen bij deze methoden al direkt twee soorten onderscheiden:

- 1° de methoden die de functie f op een bepaald gebied door een polynoom benaderen en door integreren van de benadering voor een bepaald punt een waarde voor y verkrijgen;
- 2° de methoden die de functie y op een bepaald gebied door een polynoom benaderen, zodanig dat door differentieren in een bepaald punt de daarbijbehorende waarde van f verkregen wordt.

De eerste soort waartoe o.a. de Adams-Bashforth en Adams-Moulton methoden behoren werd veelvuldig beschreven o.a. door Hamming [1962] en Henrici [1962].

De tweede soort werd het eerst beschreven door Curtiss en Hirschfelder [1952]. Deze methoden worden toegepast door Gear [1967, 1968a, 1968b]. Een onderzoek naar de stabiliteit werd al in 1954 verricht door Mitchell en Craggs.

Beide soorten methoden kunnen op verschillende manieren toegepast worden.

1. Men kan eerst d.m.v. een predictor-formule de nieuwe waarde voor y benaderen en daarna een vast aantal malen (m) hierop

een corrector-formule toepassen. Iedere correctie zal gepaard moeten gaan met een functie-evaluatie en bij de integrerende methode zal ook de predictie voorafgegaan moeten worden door een functie-evaluatie. Symbolisch geven we deze handelwijzen weer door (vgl. Krogh [1966])

$EP(EC)^m$ voor de Adams-methoden

en

$P(EC)^m$ voor de Curtiss-methoden.

2. Men kan een predictie doen voor y en daarna iteratief een corrector-formule toepassen totdat een limietwaarde verkregen is. We kunnen dit aangeven met $EP(EC)^\infty$ c.q. $P(EC)^\infty$. Convergentie van de corrector is evenwel een zware eis, in het bijzonder voor stijve vergelijkingen (zie §4, §7).
3. We kunnen ook eerst een predictie doen voor y en daarna y berekenen door een stelsel vergelijkingen op te lossen met de methode van Newton (vgl. Gear [1968a, 1968b], Liniger [1970]). Hiervoor moeten we de Jacobiaan evalueren. Wanneer de differentiaalvergelijking niet-lineair is, is ook het oplossen van dit stelsel een iteratief proces. De convergentie hiervan is echter onafhankelijk van de stijfheid van de differentiaalvergelijking. Deze wijze van oplossen geven we aan met IMPL.

Opvallend is het feit dat in de literatuur de lineaire meerstaps-methoden bijna uitsluitend toegepast worden met een constante staplengte en/of constante orde. Uitzonderingen hierop zijn Gear [1968b] die de orde aanpast (hij kan niet dynamisch de staplengte variëren), Krogh [1968] en Meershoek [1969] die voor de Adams-methode zowel orde als staplengte variëren. De dynamische staplengte-besturing wordt bij Krogh mogelijk gemaakt door gewijzigde differentiequotienten, zie Blanch [1953]. Het gebruik hiervan wordt door ons niet nagevolgd omdat het geen voordeel bleek te bieden boven het gebruik van de gewone differentiequotienten.

We zullen algorithmen beschrijven die de differentiaalvergelijking (of een stelsel differentiaalvergelijkingen) volgens bovenstaande methoden integreren en bovendien dynamisch staplengte en orde kunnen variëren.

Er moet nog worden onderzocht volgens welke strategieën staplengte, orde en het aantal iteraties per stap vastgesteld moeten worden om de oplossing aan nader te specificeren eisen te laten voldoen.

§2. Benadering van een functie door middel van een eenvoudig te differentieren of te integreren polynoom

Zij f een functie $E \subset \mathbb{R} \rightarrow \mathbb{R} \subset \mathbb{R}$ waarvan de waarde in $n+1$ steunpunten op een bepaald gebied bekend is. Wanneer we deze functie op dit gebied willen benaderen door een n -de graads polynoom, kunnen wij dit op eenvoudige wijze doen d.m.v. Newton's formule met differentiequotienten. Om deze formule voor ons doel geschikt te maken - we willen het polynoom namelijk op eenvoudige wijze kunnen differentieren of integreren - introduceren we enkele definities, vgl. De Vogelaere [1970]:

$$N = M = K = \{0, 1, 2, \dots, n\}$$

$$N_+ = M_+ = K_+ = \{0, 1, 2, \dots, n, n+1\}$$

$$S_N = \{x_0, x_1, \dots, x_n\}$$

$$S_N[j] = x_j; \quad j \in N, \quad x_j \in S_N \subset E$$

$$f_N = \{f(x_0), f(x_1), \dots, f(x_n)\}$$

$$f_N[j] = f(x_j); \quad j \in N, \quad f(x_j) \in f_N \subset \mathbb{R}$$

$$G_{E \times K} = \{1, (x-x_0), (x-x_0)(x-x_1), \dots, (x-x_0) \dots (x-x_{n-1})\}$$

$$G_{E \times K}(x)[k] = \prod_{i=0}^{k-1} (x-x_i); \quad k \in K, \quad x \in E, \quad x_j \in S_N$$

$$H_{E \times M} = \{1, (x-x_0), (x-x_0)^2, \dots, (x-x_0)^n\}$$

$$H_{E \times M}(x)[m] = (x-x_0)^m; \quad m \in M, \quad x \in E, \quad x_0 \in S_N.$$

Analoog definiëren we S_{N+} , $G_{E \times K+}$, $H_{E \times M+}$.

We definiëren bovendien een vermenigvuldiging $*$ als volgt

$$P_{L \times M} * Q_{M \times N} = R_{L \times N} \iff \forall_{l \in L, n \in N} R[l, n] = \sum_{m \in M} P[l, m].Q[m, n].$$

We geven de n-de graads polynoombenadering van f op de steunpunten S_N aan met f_E .

Newtons formule laat zich nu schrijven:

$$f_E = G_{E \times K} * a_K$$

waarin a_K de vector van de differentiequotienten voorstelt:

$$a = a_K = \{f[x_0], f[x_0, x_1], f[x_0, x_1, x_2], \dots, f[x_0, x_1, \dots, x_n]\}$$

hierin is gedefinieerd voor $j \geq i$

$$f[x_i, x_{i+1}, \dots, x_j] := \underline{\text{if}} \ i = j \ \underline{\text{then}} \ f(x_i) \\ \underline{\text{else}} \ (f[x_i, \dots, x_{j-1}] - f[x_{i+1}, \dots, x_j]) / (x_i - x_j).$$

We kunnen nu invoeren de matrices

G gedefinieerd door

$$G[j, k] = G_{N \times K}[j, k] = G_{E \times K}(S_N[j])[k]$$

$$\forall_{j \in N, k \in K}$$

H gedefinieerd door

$$H[j, m] = H_{N \times M}[j, m] = H_{E \times M}(S_N[j])[m]$$

$$\forall_{j \in N, m \in M}$$

Aangezien $f_E(x_j) = f(x_j)$ voor $x_j \in S_N$, krijgen we

$$f_N[j] = f_E(S_N[j]) = G_{E \times K}(S_N[j]) * a_K = G_{N \times K}[j,] * a_K$$

zodat

$$f_N = G_{N \times K} * a_K.$$

Omdat een n-de graads polynoom door n+1 punten volledig bepaald is, zal ook voor een zekre vector p_M gelden:

$$f_E = H_{E \times M} * p_M$$

$$f_N = H_{N \times M} * p_M.$$

Combinatie van beide uitdrukkingen voor f_N geeft

$$G_{N \times K} * a_K = H_{N \times M} * p_M.$$

Met de definitie

$$A = A_{M \times K} = (H_{N \times M})^{-1} * G_{N \times K}$$

leidt dit tot

$$p_M = A_{M \times K} * a_K.$$

Stelling

De matrix $A_{M \times K}$ is de bovendriehoeksmatrix waarvan de elementen worden bepaald door:

$$A[0,0] = 1; j \neq 0 \rightarrow A[0,j] = 0$$

en de recursierelatie

$$A[i,j] = (x_0 - x_{j-1}) * A[i,j-1] + A[i-1,j-1].$$

Bewijs

Zij $P_{M \times K}$ een bovendriehoeksmatrix met de in de stelling genoemde eigenschappen. Dan geldt voor $k = 0, 1, 2, \dots, n$.

$$1^\circ \quad \sum_{i=0}^n H[k,i]P[i,0] = H[k,0].P[0,0] = H[k,0] = 1 = G[k,0].$$

$$2^\circ \quad \forall j = 1, 2, 3, \dots, n$$

$$\begin{aligned} \sum_{i=0}^n H[k,i]P[i,j] &= \sum_{i=1}^n H[k,i]P[i,j] \\ &= \sum_{i=1}^n H[k,i] \{ (x_0 - x_{j-1}) P[i,j-1] + P[i-1,j-1] \} \\ &= (x_0 - x_{j-1}) \sum_{i=1}^n H[k,i]P[i,j-1] + \sum_{i=0}^{n-1} H[k,i+1]P[i,j-1] \\ &= (x_0 - x_{j-1}) \sum_{i=0}^n H[k,i]P[i,j-1] + (x_k - x_0) \sum_{i=0}^{n-1} H[k,i]P[i,j-1] \\ &= (x_k - x_{j-1}) \sum_{i=0}^n H[k,i]P[i,j-1] \\ &= \left(\sum_{i=0}^n H[k,i]P[i,j-1] \right) \cdot G[k,j] / G[k,j-1]. \end{aligned}$$

D.m.v. volledige inductie volgt uit 1° en 2°

$$\forall j, k = 0, 1, 2, \dots, n \quad \sum_{i=0}^n H[k,i]P[i,j] = G[k,j]$$

ofwel

$$H_{N \times M} * P_{M \times K} = G_{N \times K} = H_{N \times M} * A_{M \times K}.$$

$$i \neq j \rightarrow x_i \neq x_j \Rightarrow \det G_{N \times K} \neq 0 \Rightarrow \det H_{N \times M} \neq 0 \Rightarrow$$

$$H_{N \times M} \text{ is regulier} \Rightarrow P_{M \times K} = A_{M \times K}.$$

$A_{M \times K}$ heeft de in de stelling genoemde eigenschappen.

Q.E.D.

We kunnen nu het benaderende polynoom f_E schrijven in een vorm die zeer geschikt is om te differentieren of te integreren:

$$f_E = H_{E \times M} * A_{M \times K} * a_K.$$

Deze vorm laat tevens nog een voordeel zien van de toegepaste notatie:

$H_{E \times M}$ bevat uitsluitend informatie over het type functies waarmee f benaderd wordt.

$A_{M \times K}$ bevat uitsluitend de benodigde administratieve gegevens over de steunpunten.

a_K bevat uitsluitend de differentiequotienten.

§3. Beschrijving van de methode gebaseerd op integratie

Zij $\eta(x)$ oplossing van de differentiaalvergelijking

$$\eta^{(d)}(x) = f(x, \eta, \eta', \dots, \eta^{(d-1)})$$

met beginvoorwaarden $\eta(a), \eta'(a), \dots, \eta^{(d-1)}(a)$ en zij f_E een polynoombenadering van $f(x, \eta, \eta', \dots, \eta^{(d-1)})$ gebaseerd op de steunpunten S_N (of S_{N+}) dan definiëren we:

$$\Gamma = \{0, 1, 2, \dots, d\}$$

$$y_{0\Gamma} = \{y(x_0), y'(x_0), \dots, y^{(d)}(x_0)\}$$

de benaderde oplossing van de differentiaalvergelijking voor het punt $x = x_0$ (bijv. $x_0 = a$)

$$y_{\Gamma \times E} = \{y(x), y'(x), y''(x), \dots, y^{(d)}(x)\}$$

de benaderde oplossing gedefinieerd door:

$$y_{\Gamma \times E}[d](x) = y^{(d)}(x) = f_E(x)$$

en

$$y_{\Gamma \times E}[\gamma](x_0) = y^{(\gamma)}(x_0) = y_{0\Gamma}[\gamma] \quad \gamma = 0, 1, \dots, d-1.$$

Alle elementen van $y_{\Gamma \times E}$ zijn nu bepaald n.l.

$$y^{(d)}(x) = f(x)$$

$$y^{(d-1)}(x) = y^{(d-1)}(x_0) + \int_{x_0}^x f_E(t_1) dt_1$$

$$y^{(d-2)}(x) = y^{(d-2)}(x_0) + (x-x_0) y^{(d-1)}(x_0) + \int_{x_0}^x \int_{x_0}^{t_1} f_E(t_2) dt_2 dt_1$$

⋮

$$y^{(d-\gamma)}(x) = \sum_{j=0}^{\gamma-1} y^{(d-\gamma+j)}(x_0) \cdot (x-x_0)^j / j! + \int_{x_0}^x \int_{x_0}^{t_1} \dots \int_{x_0}^{t_{\gamma-1}} f_E(t_\gamma) dt_\gamma dt_{\gamma-1} \dots dt_1.$$

Aangezien $f_E = H_{E \times M} * A_{M \times K} * a_K$ kunnen we voor de γ -voudige integraal ook schrijven:

$$\int_{x_0}^x \int_{x_0}^{t_1} \dots \int_{x_0}^{t_{\gamma-1}} H_{E \times M}(t_\gamma) dt_\gamma \dots dt_1 * A_{M \times K} * a_K.$$

Als we nog B definiëren door

$$\begin{aligned} B[\gamma, m] &= B_{\Gamma \times M}[\gamma, m] = \int_{x_0}^{x_{n+1}} \dots \int_{x_0}^{t_{\gamma-1}} H_{E \times M}(t_\gamma) [m] dt_\gamma \dots dt_1 = \\ &= (x_{n+1} - x_0)^{m+\gamma} m! / (\gamma+m)! \end{aligned}$$

dan volgt:

$$\begin{aligned} (3.1) \quad y_{\Gamma} [d-\gamma] &= y_{\Gamma \times E} [d-\gamma](x_{n+1}) = \\ &= B_{\Gamma \times M} [\gamma,] * A_{M \times K} * a_K + \sum_{j=0}^{\gamma-1} (x_{n+1} - x_0)^j y_0 [d-\gamma+j] / j! \end{aligned}$$

Merken we nu op dat de matrices $A_{M \times K}$ en $B_{\Gamma \times M}$ uitsluitend afhankelijk zijn van de waarden van $(x_i - x_0)$ $i = 1, 2, \dots, n+1$ dan ligt het voor de hand te definiëren

$$U = U_{\Gamma \times K} = B_{\Gamma \times M} * A_{M \times K}$$

A bevat de informatie over de oude steunpunten.

B bevat de informatie over het nieuwe steunpunt.

§4. Het oplossen zonder gebruikmaking van de Jacobiaan

Is de benadering van f gebaseerd op S_N dan geldt

$$(4.1) \quad f_E := H_{E \times M} * A_{M \times K} * a_K$$

en (3.1) levert een voorspellende waarde voor

$$y_{h\Gamma} = \{y(x_{n+1}), y'(x_{n+1}), \dots, y^{(d)}(x_{n+1})\}.$$

Na een functievevaluatie hebben we nu een schatting voor $f(x_{n+1}, y(x_{n+1}), \dots, y^{(d-1)}(x_{n+1}))$. Hiermee kunnen we de benadering van f baseren op S_{N+} en

$$(4.2) \quad f_E := H_{E \times M+} * A_{M+ \times K+} * a_{K+}.$$

Uit het feit dat $A_{M+ \times K+}$ een bovendriehoeksmatrix is, die uit $A_{M \times K}$ gevormd wordt door toevoegen van de $n+1$ ste rij en $n+1$ ste kolom, volgt dat (4.1) een speciaal geval is van (4.2): beide vormen zijn n.l. gelijk voor $a_{K+}[n+1] = 0$. Dit houdt in dat de predictor op dezelfde wijze berekend kan worden als de corrector.

We kunnen nu de convergentie nagaan voor het geval dat we de corrector herhaald willen toepassen. Voor alle $\gamma \in \Gamma$ geldt: ($m=0, 1, 2, \dots$)

$$y_{h'}^{m+1}[d-\gamma] = U_{\Gamma \times K+}[\gamma,] * a_{K+}'^{m+1} + \sum_{j=0}^{\gamma-1} (x_{n+1} - x_0)^j y_0[d-\gamma+j]/j!$$

$$y_{h'}^m[d-\gamma] = U_{\Gamma \times K+}[\gamma,] * a_{K+}'^m + \sum_{j=0}^{\gamma-1} (x_{n+1} - x_0)^j y_0[d-\gamma+j]/j!$$

$$(4.3) \quad (y^h{}^{m+1} - y^h{}^m)[d-\gamma] = (a^{m+1} - a^m)_{K+} * U_{\Gamma \times K+}[\gamma,] = \\ = (a^{m+1} - a^m)[n+1]. U[\gamma, n+1].$$

Voor $\gamma = 0$ is dit:

$$(y^h{}^{m+1} - y^h{}^m)[d] = (f^{m+1} - f^m)(x_{n+1}) \\ = (a^{m+1} - a^m)[n+1]. U[0, n+1]$$

zodat voor alle $\gamma \in \Gamma$

$$(4.4) \quad (y^h{}^{m+1} - y^h{}^m)[d-\gamma] = (f^{m+1} - f^m)(x_{n+1}). U[\gamma, n+1] / U[0, n+1].$$

De eis voor convergentie luidt:

$$(4.5) \quad \left| \frac{(f^{m+1} - f^m)(x_{n+1})}{(y^h{}^m - y^h{}^{m-1})[d-\gamma]} \cdot \frac{U[\gamma, n+1]}{U[0, n+1]} \right| < 1.$$

Uit (4.4) volgt dat

$$(4.6) \quad \frac{(y^h{}^m - y^h{}^{m-1})[d-\gamma]}{U[\gamma, n+1]} = \frac{(f^m - f^{m-1})(x_{n+1})}{U[0, n+1]}$$

onafhankelijk is van γ zodat

$$(4.7) \quad (4.5) \Rightarrow \left| \sum_{j=0}^{d-1} \frac{\partial f}{\partial y^{(j)}} \cdot (y^h{}^m - y^h{}^{m-1})[j] \cdot \frac{U[d-j, n+1]}{(y^h{}^m - y^h{}^{m-1})[j]} \right| = \\ = \left| \sum_{j=0}^{d-1} \frac{\partial f}{\partial y^{(j)}} U[d-j, n+1] \right| < U[0, n+1].$$

Als we schrijven $x_0 - x_j = h_j \quad j = 1(1)n$

$$x_{n+1} - x_0 = h$$

en bovendien aannemen dat $h_j \sim jh$ (d.w.z. alle stappen van dezelfde orde van grootte) dan volgt

$$U[\gamma, n+1] = O(h^{\gamma+n+1})$$

$$(4.7) \Rightarrow \left| \sum_{j=0}^{d-1} \frac{\partial f}{\partial y^{(j)}} \cdot O(h^{d-j+n+1}) \right| < O(h^{n+1})$$

$$\Rightarrow \forall_{j < d} \left| \frac{\partial f}{\partial y^{(j)}} \right| < O(h^{j-d}).$$

Opmerking 1

Hieruit volgt, dat het voor de convergentie bijzonder gunstig is een stelsel differentiaalvergelijkingen als hogere orde differentiaalvergelijking te schrijven, als daardoor in de uitdrukking voor f de hoogste afgeleiden van y verdwijnen.

Opmerking 2

Uit de vorige opmerking volgt dat geen mogelijkheid bestaat stijve differentiaalvergelijkingen op deze wijze efficiënt op te lossen. Als n.l. zo'n hogere orde differentiaalvergelijkingen omgezet wordt in een stelsel eerste orde vergelijkingen dan komt de coëfficiënt van de hoogste afgeleide van de vergelijking overeen met het spoor van de Jacobiaan en dus met de som van de eigenwaarden.

§5. Het oplossen door gebruik te maken van de Jacobiaan

Gebruikten we in de vorige paragraaf een expliciete methode om y_h te berekenen, in deze paragraaf zullen we met de methode van Newton de vector y_h impliciet berekenen. Hiervoor zullen we echter niet alleen telkens de functiewaarden, maar ook de Jacobiaan moeten evalueren (vgl. Liniger en Willoughby [1970] en Gear[1968b]).

Zij $x_h = x_{n+1}$ een vast gekozen waarde en zij $y_h = y_{h_T}$ de te berekenen vector:

$$\{y(xh), y'(xh), y''(xh), \dots, y^{(d)}(xh)\}$$

waarvan een schatting yh^p bekend is.

Definieren we:

$$\Delta y_{\Gamma} = y_{\Gamma} - y_{\Gamma}^p = \Delta y$$

$$\Delta f = f(xh, y) - y_{\Gamma}^p[d]$$

$$\begin{cases} Fy[j](x) = \frac{\partial f}{\partial y^{(j)}}(x) & j = 0(1) d-1 \\ Fy[d] = 0 \end{cases}$$

dan kunnen we $f(xh, y)$ in een Taylor ontwikkeling schrijven

$$f(xh, y) = f(xh, y^p) + \sum_{j=0}^{d-1} \frac{\partial f}{\partial y^{(j)}}(xh) \Delta y[j] + \dots$$

Nemen we aan dat f een lineaire functie is in y , dan kunnen we dit schrijven als

$$(5.1) \quad f(xh, y) = f(xh, y^p) + \sum_{j=0}^{d-1} Fy[j](xh) \cdot \Delta y[j].$$

We vonden in §3 de relaties:

$$y_{\Gamma}^p[d-\gamma] = \sum_{j=0}^{d-1} y_0[d-\gamma+j] \cdot h^j/j! + U_{\Gamma \times K}[\gamma,] * a_K$$

- f gebaseerd op S_N (de predictor) -

$$y_{\Gamma}^p[d-\gamma] := \sum_{j=0}^{d-1} y_0[d-\gamma+j] \cdot h^j/j! + U_{\Gamma \times K+}[\gamma,] * a_{K+}$$

- f gebaseerd op S_{N+} -.

In feite moet $a_{K+}[n+1]$, de enige vrije parameter, zodanig bepaald worden dat

$$\begin{aligned} y_h[d] &= f(x_h, y_h) \\ &= U_{\Gamma \times K}[0,] * a_K + U[0, n+1] \cdot a[n+1] \end{aligned}$$

$$\begin{aligned} y_h^p[d] &= U_{\Gamma \times K}[0,] * a_K \Rightarrow \\ a[n+1] &= \frac{f(x_h, y_h) - y_h^p[d]}{U[0, n+1]} \end{aligned}$$

zodat de relatie bestaat:

$$(5.2) \quad (y_h - y_h^p)[d-\gamma] = (f(x_h, y_h) - y_h^p[d]) \cdot \frac{U[\gamma, n+1]}{U[0, n+1]}$$

Combineren we de definitie van Δf en (5.2) met (5.1) dan krijgen we

$$\Delta y_h[d-\gamma] = (\Delta f + \sum_{j=0}^{d-1} F_y[j](x_h) \cdot \Delta y_h[j]) \cdot \frac{U[\gamma, n+1]}{U[0, n+1]}$$

Uit deze relatie moeten we Δy_h oplossen:

$$(5.3) \quad \sum_{j=0}^d (U[0, n+1] \delta_{j, d-\gamma} - U[\gamma, n+1] F_y[j](x_h)) \Delta y_h[j] = U[\gamma, n+1] \cdot \Delta f$$

$$\gamma = 0, 1, \dots, d.$$

($\delta_{i,j}$ is de Kronecker-delta.)

Uit dit lineaire stelsel kan Δy_h berekend worden, waarna dan $y_h = y_h^p + \Delta y_h$ de gecorrigeerde waarde van y_h levert. Voor niet-lineaire differentiaalvergelijkingen zal bovenstaande procedure iteratief toegepast moeten worden.

§6. Beschrijving van de methode gebaseerd op differentiatie

Benaderen we de oplossing van

$$\eta^{(d)}(x) = f(x, \eta, \eta', \dots, \eta^{(d-1)})$$

met beginvoorwaarden $\eta(a)$, $\eta'(a)$, ..., $\eta^{(d-1)}(a)$ door de functie $y(x)$ en zijn reeds bekend de waarden van y op de steunpunten S_N , dan kunnen we y benaderen met een polynoom y_E , zoals beschreven is in §2.

Analoog aan de benadering van f in §3 krijgen we nu

$$y_E = H_{E \times M} * A_{M \times K} * c_K,$$

waarin c_K de vector van de differentiequotienten voorstelt

$$c = c_K = \{y[x_0], y[x_0, x_1], \dots, y[x_0, x_1, \dots, x_n]\}$$

hierin is gedefinieerd voor $j \geq i$

$$y[x_i, x_{i+1}, \dots, x_j] = \underline{\text{if}} \ i = j \ \underline{\text{then}} \ y(x_i) \\ \underline{\text{else}} \ (y[x_i, \dots, x_{j-1}] - y[x_{i+1}, \dots, x_j]) / (x_i - x_j).$$

Evenzo geldt nu:

$$(6.1) \quad y_{\Gamma}[\gamma] = y_E^{(\gamma)}(x_{n+1}) = H_{E \times M}^{(\gamma)}(x_{n+1}) * A_{M \times K} * c_K.$$

Definieren we b en u als volgt:

$$b[\gamma, m] = b_{\Gamma \times M}[\gamma, m] = H_{E \times M}^{(\gamma)}(x_{n+1})[m] = \left. \left(\frac{d}{dt} \right)^{\gamma} (x - x_0)^m \right|_{x=x_{n+1}} = \\ \underline{\text{if}} \ m < \gamma \ \underline{\text{then}} \ 0 \ \underline{\text{else}} \ (x_{n+1} - x_0)^{m-\gamma} m! / (m-\gamma)!,$$

$$u = u_{\Gamma \times K} = b_{\Gamma \times M} * A_{M \times K},$$

dan volgt uit (6.1)

$$(6.2) \quad y_{\Gamma} = b_{\Gamma \times M} * A_{M \times K} * c_K = u * c.$$

Ook hier blijkt duidelijk het voordeel van de notatie:

c bevat de differentiequotienten die de functie karakteriseren;

A bevat de administratie betreffende de reeds behandelde stappen;

b bevat de benodigde gegevens voor de nieuwe stap.

§7. Het oplossen zonder gebruikmaking van de Jacobiaan

De berekening loopt hier geheel analoog aan de berekening in §4.

Gebaseerd op de steunpunten S_N krijgen we

$$y_{\Gamma}^p = y_{\Gamma}^{0'} = u_{\Gamma \times K} * c_K.$$

Gebaseerd op de steunpunten S_{N+} , na evaluatie van $f(x_{n+1}, y_h)$, krijgen we

$$y_h^{m'} = u_{\Gamma \times K+} * c_{K+}^{m'}, \quad m = 1, 2, 3, \dots$$

$$(y_h^{m+1'} - y_h^{m'})_{\Gamma} = u_{\Gamma \times K+} * (c_{K+}^{m+1'} - c_{K+}^{m'})$$

$$(y_h^{m+1'} - y_h^{m'})[d] = (f^{m+1'} - f^{m'})(x_{n+1})$$

$$= u[d, n+1] \cdot (c^{m+1'} - c^{m'})[n+1]$$

zodat

$$(7.1) \quad (y_h^{m+1'} - y_h^{m'})[\gamma] = (f^{m+1'} - f^{m'})(x_{n+1}) \cdot \frac{u[\gamma, n+1]}{u[d, n+1]}$$

De eis voor convergentie luidt:

$$\left| \frac{(f^{m+1'} - f^{m'})(x_{n+1})}{(y_h^{m'} - y_h^{m+1'})[\gamma]} \cdot \frac{u[\gamma, n+1]}{u[d, n+1]} \right| < 1$$

ofwel

$$\left| \sum_{j=0}^{d-1} \frac{\partial f(x_{n+1})}{\partial y^{(j)}} \cdot u[j, n+1] \right| < u[d, n+1]$$

omdat uit (7.1) volgt dat

$$u[\gamma, n+1] / (y h^{m+1} - y h^m) [\gamma]$$

onafhankelijk is van γ .

Evenals in §4 kunnen we ook de orde van grootte aangeven voor $u[j, n+1]$ als alle staplengten weer van dezelfde grootte-orde zijn. Hier geldt $u[\gamma, n+1] = O(h^{n+1-\gamma})$. Derhalve wordt de staplengte-beperkende voorwaarde

$$\left| \sum_{j=0}^{d-1} \frac{\partial f(x_{n+1})}{\partial y^{(j)}} O(h^{n+1-j}) \right| < O(h^{n+1-d}).$$

De eis voor convergentie is dus dezelfde als in §4:

$$(7.2) \quad \forall j < d \quad \left| \frac{\partial f}{\partial y^{(j)}} \right| < O(h^{j-d}).$$

§8. Het oplossen door gebruik te maken van de Jacobiaan

Ook met betrekking tot de berekening van y_h met de methode van Newton lopen de integrerende en differentierende methoden parallel (vgl. §5).

Baseren we y_E op S_N dan $y_h^D = u_{\Gamma \times K} * c_K$.

Willen we y_E baseren op S_{N+} dan $y_h^D = u_{\Gamma \times K+} * c_{K+}$.

In het rechterlid is uitsluitend $c[n+1]$ onbekend, terwijl y_h^D zo bepaald moet worden dat

$$f(x_h, y_h^D) = y_h^D[d].$$

Het gevolg is dat

$$\begin{aligned} (y_h - y_h^p) &= c[n+1] * u[,n+1] \\ &= (f(x_h, y_h) - y_h^p[d]) \cdot \frac{u[,n+1]}{u[d,n+1]} \end{aligned}$$

Het lineaire stelsel dat opgelost moet worden luidt dus

$$(8.1) \quad \Delta y_{h\Gamma} = \left(\Delta f + \sum_{j=0}^{d-1} \frac{\partial f(x_h)}{\partial y^{(j)}} \cdot \Delta y_h[j] \right) \cdot \frac{u[,n+1]}{u[d,n+1]} \rightarrow$$

$$\sum_{j=0}^d (u[d,n+1] \cdot \delta_{j,\gamma} - F_{y[j]}(x_h) \cdot u[\gamma,n+1]) \Delta y_h[j] = u[\gamma,n+1] \cdot \Delta f$$

$$\gamma = 0, 1, \dots, d.$$

Wanneer we de waarden $F_{y[j]}(x_h)$ ter beschikking hebben kan het lineaire stelsel opgelost worden, waarna $y_{h\Gamma} = y_h^p + \Delta y_{h\Gamma}$ de gecorrigeerde waarde van $y_{h\Gamma}$ levert.

§9. Stelsels vergelijkingen

Tot slot van dit hoofdstuk geven we het formularium voor het geval van een stelsel differentiaalvergelijkingen van dezelfde orde.

Zij $Z = \{1, 2, \dots, N\}$.

N het aantal vergelijkingen waaruit het stelsel bestaat.

$\Gamma = \{0, 1, \dots, d\}$.

d de orde van de differentiaalvergelijkingen.

$f_{Z \times E}$ de rij functies die het stelsel differentiaalvergelijkingen definieert.

$$f_{Z \times E}[z] \in E \rightarrow R$$

$y_{E \times Z}$ een benadering van de oplossing van het gegeven stelsel.

$$y_{E \times Z}[z] \in E \rightarrow R$$

$$y_{\Gamma \times Z}^0[\gamma, z] = \left(\frac{d}{dt}\right)^\gamma y_{E \times Z}(x_0)[z] = y_0[\gamma, z]$$

$$y_{\Gamma \times Z}^h[\gamma, z] = \left(\frac{d}{dt}\right)^\gamma y_{E \times Z}(x_{n+1})[z] = y_h[\gamma, z]$$

$$\Delta y_{\Gamma \times Z} = y_{\Gamma \times Z}^h - y_{\Gamma \times Z}^p = \Delta y_h$$

$$\Delta f_Z[z] = f_Z[z](x_h, y_{\Gamma \times Z}^p) - y_{\Gamma \times Z}^p[d, z]$$

$$\left. \begin{aligned} F_{Y_{Z \times \Gamma \times Z}}[z, j, i] &= \frac{\partial f[z]}{\partial y[j, i]} & j = 0, 1, 2, \dots, d-1 \\ F_{Y_{Z \times \Gamma \times Z}}[z, d, i] &= 0. \end{aligned} \right\}$$

Voor de Adams-methode krijgen we hier parallel aan de berekening in §3:

$$(9.1) \quad y_{\Gamma \times Z}^h[d-\gamma,] = U_{\Gamma \times K}[\gamma,] * a_{K \times Z} + \sum_{j=0}^{\gamma-1} (x_{n+1} - x_0)^j y_{\Gamma \times Z}^0[d-\gamma+j,]/j!$$

$$\gamma = 0, 1, \dots, d$$

$$(9.2) \quad (y_h^{m+1} - y_h^m)[d-\gamma, z] = (a^{m+1} - a^m)[n+1, z].U[\gamma, n+1]$$

$$\begin{aligned} (y_h^{m+1} - y_h^m)[d, z] &= (f^{m+1} - f^m)[z](x_{n+1}) \\ &= (a^{m+1} - a^m)[n+1, z].U[0, n+1] \end{aligned}$$

$$(9.3) \quad (y_h^{m+1} - y_h^m)[d-\gamma,] = (f^{m+1} - f^m)(x_{n+1}).U[\gamma, n+1]/U[0, n+1]$$

$$(f^{m+1} - f^m)[z](x_{n+1}) = \sum_{i \in Z} \sum_{j=0}^{d-1} \frac{\partial f[z]}{\partial y[j, i]} \cdot (y_h^m - y_h^{m-1})[j, i].$$

De convergentiefactor is

$$\frac{(y_h^{m+1} - y_h^m)[d-\gamma, z]}{(y_h^m - y_h^{m-1})[d-\gamma, z]} =$$

$$= \sum_{i \in Z} \sum_{j \in \Gamma} F_y[z, j, i] \frac{(y_h^m - y_h^{m-1})[j, i]}{(y_h^m - y_h^{m-1})[j, z]} \cdot \frac{U[d-j, n+1]}{U[0, n+1]}$$

omdat, vanwege (9.2), $(y_h^{m+1} - y_h^m)[d-\gamma, z]/U[\gamma, n+1]$ onafhankelijk is van γ .

We kunnen op eenvoudige wijze y_h impliciet berekenen als er sprake is van een ontkoppeld stelsel d.w.z.

$$\forall z, i \in Z \quad i \neq z \rightarrow \frac{\partial f[z]}{\partial y[j, i]} = 0.$$

Dan geldt namelijk

$$\frac{(y_h^{m+1} - y_h^m)[d-\gamma, z]}{(y_h^m - y_h^{m-1})[d-\gamma, z]} = \sum_{j \in \Gamma} F_y[z, j, z] \cdot U[d-j, n+1]/U[0, n+1]$$

en dan (9.3) \Rightarrow

$$(9.4) \quad \Delta y_{\Gamma \times Z}[d-\gamma,] = \frac{\Delta f_Z \cdot U_{\Gamma \times K^+}[\gamma, n+1]}{U[0, n+1] - \sum_{j \in \Gamma} F_y[z, j, z] \cdot U[d-j, n+1]}$$

Wanneer het stelsel differentiaalvergelijkingen niet ontkoppeld is loopt de impliciete berekening iets ingewikkelder. De berekening gaat dan analoog aan die in §5.

We moeten $y_{\Gamma \times Z}$ zodanig zien te vinden dat

$$y_h[d, z] = f(z, x_h, y_h) \quad \forall z \in Z$$

we weten

$$yh[d,z] = U_{\Gamma \times K}[0,] * a_{K \times Z} + U[0,n+1].a[n+1,z]$$

$$yh^p[d,z] = U_{\Gamma \times K}[0,] * a_{K \times Z}$$

$$\Rightarrow a[n+1,z] = \frac{f(z,xh,yh) - yh^p[d,z]}{U[0,n+1]}$$

$$\begin{aligned} (yh - yh^p)[d-\gamma, z] &= U[\gamma, n+1].a[n+1, z] = \\ &= \{f(z,xh,yh) - yh^p[d,z]\} \cdot U[\gamma, n+1] / U[0, n+1]. \end{aligned}$$

Met de notatie van het begin van de paragraaf wordt dit, mede vanwege

$$f(z,xh,yh) = f(z,xh,yh^p) + \sum_{z \in Z} \sum_{j=0}^{d-1} \frac{\partial f[z]}{\partial y[j,i]} \cdot \Delta yh[j,i],$$

tot

$$\Delta yh[d-\gamma, z] = \{\Delta f[z] + \sum_{z \in Z} \sum_{j \in \Gamma} \{Fy[z, j, i] \cdot \Delta yh[j, i]\} \frac{U[\gamma, n+1]}{U[0, n+1]}}$$

hetgeen identiek is met het lineaire stelsel:

$$\begin{aligned} (9.5) \quad \sum_{\substack{i \in Z \\ j \in \Gamma}} \{\delta_{j, \gamma} \delta_{i, z} \cdot U[0, n+1] - Fy[z, j, i] \cdot U[d-\gamma, n+1]\} \Delta yh[j, i] \\ = \Delta f[z] \cdot U[d-\gamma, n+1] \end{aligned}$$

$$\forall \gamma \in \Gamma, z \in Z.$$

Dezelfde berekeningen kunnen we uitvoeren voor de Curtiss-methode (vgl. §7 en §8).

$$(9.6) \quad yh_{\Gamma \times Z}^p = u_{\Gamma \times K} * c_{K \times Z}$$

(7.1) en (9.3) \rightarrow

$$(9.7) \quad (y^{h'm+1'} - y^{h'm'})[\gamma, z] = (f^{m+1'} - f^{m'})[z](x_{n+1}) \cdot \frac{u[\gamma, n+1]}{u[d, n+1]}$$

$$(f^{m+1'} - f^{m'})[z](x_{n+1}) = \sum_{i \in Z} \sum_{j=0}^{d-1} \frac{\partial f[z]}{\partial y[j, i]} \cdot (y^{h'm'} - y^{h'm-1'})[j, i].$$

De convergentiefactor is dus

$$\frac{(y^{h'm+1'} - y^{h'm'})[\gamma, z]}{(y^{h'm'} - y^{h'm-1'})[\gamma, z]} =$$

$$= \sum_{i \in Z} \sum_{j \in \Gamma} Fy[z, j, i] \cdot \frac{(y^{h'm'} - y^{h'm-1'})[j, i]}{(y^{h'm'} - y^{h'm-1'})[\gamma, z]} \cdot \frac{u[\gamma, n+1]}{u[d, n+1]}.$$

Als we te maken hebben met een ontkoppeld stelsel kunnen we weer schrijven (omdat $(y^{h'm'} - y^{h'm-1'})[\gamma, z]/u[\gamma, n+1]$ onafhankelijk is van γ . vgl. (9.7)):

$$\frac{(y^{h'm+1'} - y^{h'm'})[\gamma, z]}{(y^{h'm'} - y^{h'm-1'})[\gamma, z]} = \sum_{j \in \Gamma} Fy[z, j, z] u[j, n+1]/u[d, n+1]$$

zodat

$$(9.8) \quad \Delta y_{\Gamma \times Z} = \frac{\Delta f_Z \cdot u_{\Gamma \times K^+}[\cdot, n+1]}{u[d, n+1] - \sum_{j \in \Gamma} Fy[z, j, z] \cdot u[j, n+1]}.$$

Als het stelsel niet ontkoppeld is moeten we weer op andere wijze te werk gaan.

Uit (9.7) volgt, dat we $y_{\Gamma \times Z}$ zodanig moeten zien te vinden dat

$$(y - y^P)[\gamma, z] = \{f(z, x_h, y_h) - y^P[d, z]\} \cdot \frac{u[\gamma, n+1]}{u[d, n+1]}$$

ofwel

$$\forall \gamma \in \Gamma, z \in Z.$$

$$\Delta y_{\Gamma \times Z}[\gamma, z] = \{\Delta f[z] + \sum_{i \in Z} \sum_{j \in \Gamma} F_{\gamma}[z, j, i] \Delta y_{\Gamma \times Z}[j, i]\} \frac{u[\gamma, n+1]}{u[d, n+1]}.$$

D.i.

$$(9.9) \quad \sum_{i \in Z} \sum_{j \in \Gamma} \{\delta_{\gamma, j} \delta_{i, z} \cdot u[d, n+1] - F_{\gamma}[z, j, i] \cdot u[\gamma, n+1]\} \Delta y_{\Gamma \times Z}[j, i] \\ = \Delta f[z] \cdot u[\gamma, n+1].$$

Uit dit lineaire stelsel is $\Delta y_{\Gamma \times Z}$ eenvoudig op te lossen.

Hoofdstuk II. De Stabiliteit

§10. Inleiding

Voor zover het lineaire meerstapsmethoden betreft met constante staplengte, bestaat er over stabiliteit en consistentie een uitgebreide literatuur, o.a. Dahlquist [1956, 1963] en Henrici [1962]. Voor lineaire meerstapsmethoden met variabele staplengte is over stabiliteit weinig of geen onderzoek gedaan.

Aan de eis voor consistentie [Henrici 1962 p. 224] is door de in hoofdstuk I vermelde methoden voldaan daar ze immers van orde 1 of hoger zijn.

De kwestie van stabiliteit ligt aanzienlijk ingewikkelder. Daarom willen we hier aandacht schenken aan de voorwaarden voor stabiliteit voor de methoden uit hoofdstuk I en deze vergelijken met enkele in de literatuur genoemde.

We onderzoeken niet de stabiliteit in de zin van Dahlquist [Dahlquist 1951, 1956] [Henrici 1962 p. 217], d.i. stabiliteit voor $h \rightarrow 0$, maar stabiliteit voor eindige staplengte [Hamming 1962], d.i. Henrici's "strong stability" [Henrici 1962 p. 235], een eigenschap waarvan verschillende varianten bekend zijn onder diverse namen (A-stability [Dahlquist 1963, Widlund 1967], $A(\alpha)$ -stability [Nørsett 1969a, 1969b], stiff-stability [Gear 1968b]).

Zoals bekend is het voor stabiliteitsonderzoek voldoende zich te beperken tot een enkele differentiaalvergelijking $y' = \lambda y$, wanneer we ook complexe waarden van λ toelaten. Er geldt dan voor stelsels dat alle eigenwaarden van de Jacobiaan binnen het complexe stabiliteitsgebied moeten liggen. Om de stabiliteit van de genoemde methoden te onderzoeken, gaan we na wat het gevolg is van een fout in de vectorgrootte y_N (dat is de discretisering van y_E op de steunpunten S_N) bij de berekening van de grootte y_h .

We zullen eerst (in §11) voor de $(n+1)$ -staps-methoden met variabele staplengte de belangrijkste gevallen behandelen.

We zullen telkens een polynoom in ϕ vinden, waarvan de coëfficiënten afhankelijk zijn van λ en van de verdeling der steunpunten. Er is sprake van stabiliteit wanneer alle nulpunten van dit karakteriserend polynoom in absolute waarde kleiner dan één zijn. Hierdoor kan men hetzij bij een gegeven steunpuntenverdeling nagaan welke waarden van λ toelaatbaar zijn, hetzij bij een bepaalde λ nagaan welke verdeling van de steunpunten stabiliteit garandeert.

In de volgende paragraaf (§12) zullen we in het bijzonder het gedrag van de 1- en 2-stapsmethoden nagaan. Vervolgens laten we in §13 zien hoe onze methoden in de gebruikelijke Lagrange-vorm geschreven kunnen worden. In §14 tekenen we voor enkele gevallen de stabiliteitsgebieden.

§11. Stabiliteit voor de variabele staplengte methoden

In deze paragraaf onderzoeken we de stabiliteit voor drie gevallen van de Adams-formules:

- 1° uitsluitend de predictor (Adams-EP)
- 2° de predictie gecombineerd met één correctie (Adams-EPEC)
- 3° de geïtereerde corrector (Adams-IMPL)

en voor twee gevallen van de Curtiss-formules:

- 1° de predictie gecombineerd met één correctie (Curtiss-PEC)
- 2° de geïtereerde corrector (Curtiss-IMPL).

i) de expliciete Adams-formule EP.

We beschouwen de differentiaalvergelijking $y' = f = \lambda y$.

Zijn gedefinieerd

$$y_N = \{y(x_0), y(x_1), \dots, y(x_n)\}$$

$$f_N = \{f(x_0, y(x_0)), \dots, f(x_n, y(x_n))\}$$

dan

$$f_N = \{\lambda y(x_0), \lambda y(x_1), \dots, \lambda y(x_n)\} = \lambda y_N.$$

Ook geldt (§2):

$$f_N = G_{N \times K} * a_K$$

zodat

$$a_K = (G_{N \times K})^{-1} * f_N = \lambda G_{K \times N}^{-1} * y_N.$$

(3.1) \rightarrow

$$yh_{\Gamma}^p[0] = B_{\Gamma \times M}[1,] * A_{M \times K} * a_K + y_{0\Gamma}[0]$$

$$= y_{0\Gamma}[0] + \lambda \cdot B_{\Gamma \times M}[1,] * H_{M \times N}^{-1} * y_N$$

ofwel

$$y_{n+1} - y_0 = \lambda \cdot B_{\Gamma \times M}[1,] * H_{M \times N}^{-1} * y_N.$$

Het karakteriserend polynoom is derhalve (N.B. de telling van de steunpunten is: $x_n, x_{n-1}, \dots, x_1, x_0, x_{n+1} \rightarrow$)

$$(11.1) \quad \phi^{n+1} - \phi^n = \lambda \sum_{j=0}^n \{B_{\Gamma \times M}[1,] * H_{M \times N}^{-1}[, n-j]\} \phi^j.$$

ii) de expliciete Adams-formule EPEC.

(3.1) \rightarrow

$$yh_{\Gamma}^p[0] = B[1,] * A * a + y_0[0]$$

$$yh_{\Gamma}^p[1] = B[0,] * A * a$$

$$\Delta f = f(yh^p[0]) - f(yh^p[1])$$

$$= \lambda^2 B[1,] * H^{-1} * y_N + \lambda y_{0\Gamma}[0]$$

$$- \lambda B[0,] * H^{-1} * y_N$$

$$= \lambda \{y_0[0] + (\lambda B[1,] - B[0,]) * H^{-1} * y_N\}$$

(4.4) \Rightarrow

$$\Delta y_{\Gamma}[0] = \Delta f \cdot U_{\Gamma \times K}[1, n+1] / U_{\Gamma \times K}[0, n+1]$$

$$y_{\Gamma}[0] = y_0[0] + B[1,] * H^{-1} * y_N \cdot \lambda +$$

$$\{\lambda \cdot y_0[0] + \{\lambda \cdot B[1,] - B[0,]\} * H^{-1} * y_N \cdot \lambda\} U[1, n+1] / U[0, n+1]$$

ofwel

$$y_{n+1} \cdot U[0, n+1] = y_0 \{U[0, n+1] + \lambda U[1, n+1]\} +$$

$$+ \{\lambda^2 U[1, n+1] B[1,] + \lambda \{B[1,] U[0, n+1] - B[0,] U[1, n+1]\} * H^{-1} * y_N\}.$$

Het karakteriserend polynoom is dan

$$(11.2) \quad \phi^{n+1} U[0, n+1] = \phi^n \{U[0, n+1] + \lambda U[1, n+1]\} +$$

$$+ \sum_{j=0}^n \lambda \{U[0, n+1] B[1,] - U[1, n+1] B[0,]\} H^{-1} [, n-j] \phi^j$$

$$+ \sum_{j=0}^n \lambda^2 U[1, n+1] B[1,] H^{-1} [, n-j] \phi^j.$$

iii) de impliciete Adams-formule.

 y_{Γ}^P en Δf worden weer berekend als bij de expliciete formule(5.3) \Rightarrow

$$\Delta y_{\Gamma}[0] = \Delta f \cdot U[1, n+1] / (U[0, n+1] - \lambda U[1, n+1])$$

zodat

$$y_{\Gamma}[0] = y_0[0] + B[1,] * H^{-1} * y_N \cdot \lambda +$$

$$+ \frac{\lambda \cdot y_0[0] + \{\lambda \cdot B[1,] - B[0,]\} * H^{-1} * y_N \cdot \lambda}{U[0, n+1] - \lambda U[1, n+1]} \cdot U[1, n+1]$$

ofwel

$$y_{n+1} U[0, n+1] - \lambda y_{n+1} U[1, n+1] = y_0 U[0, n+1] +$$

$$+ \{U[0, n+1] B[1,] - U[1, n+1] B[0,]\} * H^{-1} * y_N \cdot \lambda.$$

Het karakteriserend polynoom is

$$(11.3) \quad \phi^{n+1}(U[0,n+1]-\lambda U[1,n+1]) = \phi^n U[0,n+1] + \\ \lambda \sum_{j=0}^n \{U[0,n+1]B[1,] - U[1,n+1]B[0,]\} * H^{-1}[,n-j]\phi^j$$

iv) de expliciete Curtiss-formule PEC.

Zijn y_N en f_N gedefinieerd als in het begin van deze paragraaf.

Zij c gedefinieerd als in §6

dan geldt:

$$y_N = G_{N \times K} * c_K$$

$$c_K = (G_{N \times K})^{-1} * y_N$$

$$A_{M \times K} * c_K = A_{M \times K} * G_{K \times N}^{-1} * y_N = H_{M \times N}^{-1} * y_N .$$

Uit (6.1) volgt

$$yh_{\Gamma}^p = b_{\Gamma \times M} * A_{M \times K} * c_K = b_{\Gamma \times M} * H_{M \times N}^{-1} * y_N$$

$$\Delta f = f(yh^p) - yh_{\Gamma}^p[1] = \lambda \cdot yh[0] - yh[1]$$

$$= \{\lambda \cdot b[0,] - b[1,]\} * H^{-1} * y_N .$$

Uit (7.1) volgt

$$\Delta y_{h_{\Gamma}}[0] = \Delta f \cdot u[0,n+1]/u[1,n+1]$$

$$y_{h_{\Gamma}}[0] = b[0,] * H^{-1} * y_N +$$

$$+ \{\lambda b[0,] - b[1,]\} * H^{-1} * y_N \cdot u[0,n+1]/u[1,n+1]$$

ofwel

$$y_{n+1} u[1,n+1] = \lambda \cdot u[0,n+1] b[0,] * H^{-1} * y_N +$$

$$+ \{u[1,n+1]b[0,] - u[0,n+1]b[1,]\} * H^{-1} * y_N .$$

Het karakteriserende polynoom is

$$(11.4) \quad \phi^{n+1} u[1, n+1] = \lambda \sum_{j=0}^n u[0, n+1] b[0, j] * H^{-1}[, n-j] \cdot \phi^j + \\ + \sum_{j=0}^n \{u[1, n+1] b[0, j] - u[0, n+1] b[1, j]\} * H^{-1}[, n-j] \phi^j.$$

v) de impliciete Curtiss-formule

y_{Γ}^p en Δf worden berekend als bij de expliciete formule

(8.1) \Rightarrow

$$\Delta y_{\Gamma}[0] = \Delta f \cdot u[0, n+1] / (u[1, n+1] - u[0, n+1] \cdot \lambda)$$

zodat

$$y_{\Gamma}[0] = b[0, j] * H^{-1} * y_N + \frac{\{\lambda \cdot b[0, j] - b[1, j]\} * H^{-1} * y_N \cdot u[0, n+1]}{u[1, n+1] - \lambda u[0, n+1]}$$

ofwel

$$y_{n+1} u[1, n+1] - \lambda y_{n+1} u[0, n+1] =$$

$$\{u[1, n+1] b[0, j] - u[0, n+1] b[1, j]\} * H^{-1} * y_N$$

waaruit direkt het karakteriserend polynoom volgt:

$$(11.5) \quad \phi^{n+1} (u[1, n+1] - \lambda u[0, n+1]) = \\ \sum_{j=0}^n \{u[1, n+1] b[0, j] - u[0, n+1] b[1, j]\} * H^{-1}[, n-j] \phi^j.$$

§12. De stabiliteit voor $n = 0$ en $n = 1$

Om bovenstaande formules in concreto toe te passen laten we zien hoe

- I. voor $n = 0$ de methoden ontaarden in bekende één-staps methoden;
- II. voor $n = 1$ de stabiliteit afhankelijk is van de verdeling van de steunpunten.

We gebruiken de notatie $x_{n+1} - x_0 = h$ en $x_0 - x_1 = K$.
Hierdoor geldt voor $n = 1$:

$$H_{M \times N}^{-1} = \frac{1}{K} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix}.$$

i) voor de expliciete Adams-formule EP.

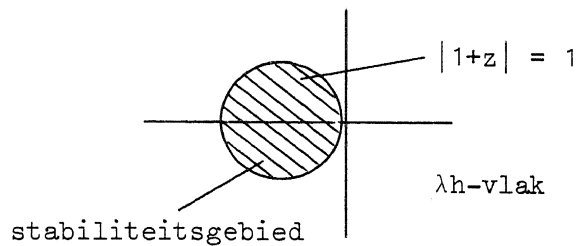
(11.1) \Rightarrow

$$\phi - 1 = \lambda B[1,0] H^{-1}[0,0] = \lambda.h.1.$$

$$\phi - (1+\lambda h) = 0.$$

Beschouwen we $z = \lambda h$ als een complexe grootheid dan zal voor bepaalde waarden van z de wortel van het polynoom in ϕ absoluut kleiner zijn dan 1. De verzameling van deze waarden van z in het complexe vlak noemen we het stabiliteitsgebied voor een bepaalde methode.

Voor het onderhavige geval is dit $1 = |\phi| = |1+z|$



De formule is de forward-Euler-formule.

ii) $n = 1$ voor de expliciete Adams-formule EP.

(11.1) \Rightarrow

$$\begin{aligned} \phi^2 - \phi &= \lambda \begin{pmatrix} h \\ \frac{1}{2}K^2 \end{pmatrix} \frac{1}{K} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix} \\ &= \frac{\lambda h}{2K} \{(2K+h)\phi - h\} \end{aligned}$$

$$2K\phi^2 - 2K\phi = \lambda h \{(2K+h)\phi - h\}.$$

Dit houdt o.a. in dat de stabiliteitsgrens voor reële λ bepaald wordt door (stel $\phi=1$)

$$-\lambda h = \frac{2K}{K+h}$$

zodat

$$h \rightarrow 0 \Rightarrow -\lambda h \rightarrow 2$$

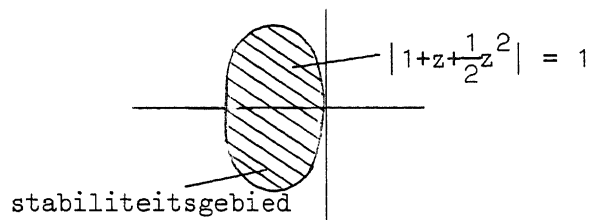
$$h \rightarrow \infty \Rightarrow -\lambda h \rightarrow 0$$

iii) $n = 0$ voor de expliciete Adams-formule EPEC.

(11.2) \Rightarrow

$$\phi \cdot h = \left\{ h + \lambda \cdot \frac{1}{2} h^2 \right\} + \lambda \left\{ h \cdot h - \frac{1}{2} h^2 \right\} + \lambda^2 \frac{1}{2} h^2 h$$

$$\phi = 1 + \lambda h + \frac{1}{2} \lambda^2 h^2$$



De formule is de formule van Heun.

iv) $n = 1$ voor de expliciete Adams-formule EPEC.

(11.2) \Rightarrow

$$\begin{aligned} \phi^2 h(h+K) &= \phi \left\{ h(h+K) + \lambda h^2 \left(\frac{1}{2} K + \frac{1}{3} h \right) \right\} + \\ &+ \frac{\lambda}{K} \begin{pmatrix} h(h+K)h & -h^2 \left(\frac{1}{2} K + \frac{1}{3} h \right) \cdot 1 \\ h(h+K) \frac{1}{2} h^2 & -h^2 \left(\frac{1}{2} K + \frac{1}{3} h \right) h \end{pmatrix} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix} + \\ &+ \frac{\lambda^2}{K} \cdot h^2 \left(\frac{1}{2} K + \frac{1}{3} h \right) \begin{pmatrix} h \\ \frac{1}{2} h^2 \end{pmatrix} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix} \end{aligned}$$

$$(12.1) \quad (\phi^2 - \phi) h(h+K) = -\frac{\lambda h^2}{6K} \{\phi(h^2 + 6hK + 6K^2) - h^2\} + \\ + \frac{\lambda^2 h^3}{12K} (3K+2h) \{(2K+h)\phi - h\}.$$

Om de grens van het stabiliteitsgebied voor reële λ te berekenen nemen we $\phi = 1$, zodat (12.1) $\Rightarrow \lambda h = -\frac{6(h+K)}{2h+3K}$.

We kunnen de maximaal toegestane staplengte h berekenen uit $2\lambda h^2 + (3K\lambda+6)h + 6K = 0$. In de praktijk zal dit echter niet aantrekkelijk zijn omdat λ in het algemeen onbekend is.

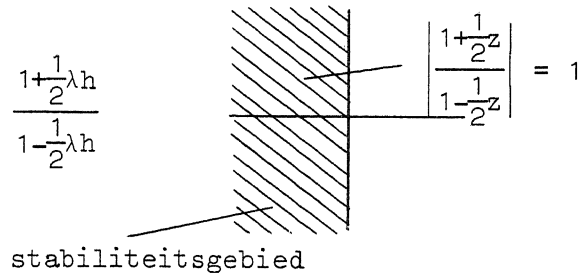
De relatie tussen steunpunten-verdeling en stabiliteitsgebied wordt dus betrekkelijk ingewikkeld. Het stabiliteitsgebied van deze methode kan men vinden in §14 voor het geval $k = 1$; $h = 0.8$, $h = 1.0$ en $h = 1.2$.

v) $n = 0$ voor de impliciete Adams-formule.

$$(11.3) \Rightarrow$$

$$\phi(h - \lambda \cdot \frac{1}{2}h^2) = h + \lambda \{h \cdot h - \frac{1}{2}h^2\}$$

$$\phi = \frac{1 + \frac{1}{2}\lambda h}{1 - \frac{1}{2}\lambda h}$$



De formule is de trapezium-formule.

vi) $n = 1$ voor de impliciete Adams-formule.

$$(11.3) \Rightarrow$$

$$\phi^2(h(h+K) - \lambda h^2(\frac{1}{2}K + \frac{1}{3}h)) = \phi h(h+K) + \\ + \frac{\lambda}{K} \begin{pmatrix} h(h+K)h & -h^2(\frac{1}{2}K + \frac{1}{3}h) \cdot 1 \\ h(h+K)\frac{1}{2}h^2 & -h^2(\frac{1}{2}K + \frac{1}{3}h) \cdot h \end{pmatrix} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix}$$

$$(\phi^2 - \phi) 6(h+K) = \frac{\lambda h}{K} \{ \phi^2(3K+2h)K + \phi(3K+h)(K+h) - h^2 \}.$$

Dit levert als stabiliteitsgrens voor reële λ ($\phi = -1$):

$$12(h+K) = \frac{\lambda h}{K} \{-2h(K+h)\}$$

ofwel

$$-\lambda h = 6 \frac{K}{h}; \quad h_{\text{grens}} = \sqrt{-6K/\lambda}.$$

vii) $n = 0$ voor de expliciete Curtiss-formule PEC.

$$(11.4) \Rightarrow$$

$$\phi = \lambda \cdot h \cdot 1 \cdot 1 + (1 \cdot 1 - h \cdot 0)$$

$$\phi = 1 + \lambda h.$$

Evenals de expliciete Adams-formule EP onttaardt deze formule tot de forward-Euler-formule.

viii) $n = 1$ voor de expliciete Curtiss-formule PEC.

$$(11.4) \Rightarrow$$

$$\phi^2(K+2h) = \lambda h(h+K) \begin{pmatrix} 1 \\ h \end{pmatrix} \frac{1}{K} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix}$$

$$+ \begin{pmatrix} (K+2h) \cdot 1 - h(h+K) \cdot 0 \\ (K+2h) \cdot h - h(h+K) \cdot 1 \end{pmatrix} \frac{1}{K} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix}$$

d.i.

$$\phi^2 K(K+2h) - \phi(K+h)^2 + h^2 = \lambda h(h+K)(\phi(h+K) - h).$$

Dit houdt o.a. in dat de stabiliteitsgrens voor reële λ bepaald wordt door:

$$-\lambda h = \frac{2(h+K)}{2h+K}$$

$$(-\lambda h)_{\text{grens}} \xrightarrow{h \rightarrow 0} 2$$

$$(-\lambda h)_{\text{grens}} \xrightarrow{h=K} 4/3$$

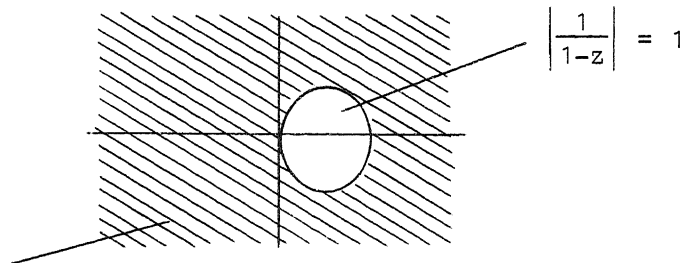
$$(-\lambda h)_{\text{grens}} \xrightarrow{h=\infty} 1$$

ix) $n = 0$ voor de impliciete Curtiss-formule.

(11.5) \implies

$$\phi(1-\lambda h) = 1$$

$$\phi = \frac{1}{1-\lambda h}$$



stabiliteitsgebied

De formule is de backward-Euler-formule.

x) $n = 1$ voor de impliciete Curtiss-formule.

(11.5) \implies

$$\phi^2((K+2h)-\lambda h(h+K)) = \frac{1}{K} \begin{pmatrix} (K+2h) \cdot 1 \\ (K+2h) \cdot h - h(h+K) \end{pmatrix} \begin{pmatrix} K & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \phi \\ 1 \end{pmatrix}$$

d.i.

$$\phi^2(K+2h)K - \phi^2 \lambda h (h+K)K = (K+h)^2 \phi - h^2.$$

Dit houdt in dat de stabiliteitsgrens voor reële λ bepaald wordt door

$$\lambda h = \frac{2(h+K)}{K}.$$

§13. De relatie tot de lineaire meerstapmethoden met vaste staplengte.De orde van de methoden

Om het verband te zien tussen onze formules en de methoden met vaste staplengte, zoals behandeld door Henrici, kunnen we de variabele stapmethoden natuurlijk ook weergeven in lineaire combinaties van y_i en f_i ($i = n+1, 0, 1, 2, \dots, n$).

(1) de open Adams-formule

$$y_{n+1} - y_0 = B_{\Gamma \times M} [1,] * H_{M \times N}^{-1} * f_N.$$

(2) de gesloten Adams-formule

$$(y_{n+1} - y_0) U[0, n+1] = f_{n+1} U[1, n+1] + \\ + \{B[1,]U[0, n+1] - B[0,]U[1, n+1]\} * H^{-1} * f_N.$$

(3) de open Curtiss-formule

$$y_{n+1} = b_{\Gamma \times M} [0,] * H_{M \times N}^{-1} * y_N.$$

(4) de gesloten Curtiss-formule

$$y_{n+1} u[1, n+1] + \{b[1,]u[0, n+1] - b[0,]u[1, n+1]\} * H^{-1} * y_N = \\ = u[0, n+1] . f_{n+1}.$$

Wanneer we deze methoden gebruiken met vaste staplengte gaan ze over in een deelverzameling van de methoden met vaste staplengte, welke in het algemeen geschreven worden in de volgende Lagrange-vorm:

$$\alpha_k y_{n+k} + \alpha_{k-1} y_{n+k-1} + \dots + \alpha_0 y_n + \\ h\{\beta_k f_{n+k} + \beta_{k-1} f_{n+k-1} + \dots + \beta_0 f_n\} = 0.$$

We kunnen de relatie als volgt schematisch weergeven:

	α_k	α_{k-1}	$(\alpha_{k-2}, \dots, \alpha_0)$	β_k	$(\beta_{k-1}, \dots, \beta_0)$
Adams open	-1	+1	0	0	$B[1,] * H^{-1}$
Adams gesl.	$-U[0, n+1]$	$U[0, n+1]$	0	$U[1, n+1]$	$\{B[1,]U[0, n+1] - B[0,]U[1, n+1]\} * H^{-1}$
Curtiss open	-1	$b[0,] * H^{-1}$		0	0
Curtiss gesl.	$-u[1, n+1]$	$\{b[0,]u[1, n+1] - b[1,]u[0, n+1]\} H^{-1}$		$u[0, n+1]$	0

De orde van de gebruikte methoden kan men op eenvoudige wijze nagaan.

Gebruiken we een Curtiss-formule gebaseerd op $n + 1$ steunpunten, dan wordt een n -de graads polynoom nog exact gerepresenteerd.

Voor de open Curtiss-formule is de orde derhalve n en voor de gesloten Curtiss-formule is de orde $n + 1$.

Gebruiken we een Adams-methode voor een D -de orde differentiaalvergelijking, dan wordt de D -de afgeleide benaderd door een n -de graads polynoom: voor de open Adams-formule is de orde $D + n$, voor de gesloten Adams-formule is de orde $D + n + 1$.

§14. Stabiliteitsgebieden

In §12 hebben we voor $n = 1$ expliciet uitgeschreven aan welke voorwaarden de eigenwaarden van de Jacobiaan en de steunpuntenverdeling moeten voldoen wanneer we stabiliteit eisen. Voor $n > 1$ worden deze relaties zo onoverzichtelijk, dat we het best in een tekening kunnen aangeven welke eis we moeten stellen aan de eigenwaarden bij een bepaalde verdeling van de steunpunten. Van de stabiliteitsgebieden hebben we tekeningen gemaakt voor $n = 0(1)7$, waarbij we als keuze voor de steunpunten hebben genomen

$$x_{j-1} - x_j = x_0 - x_1 \quad \text{voor } j = 1(1)n$$

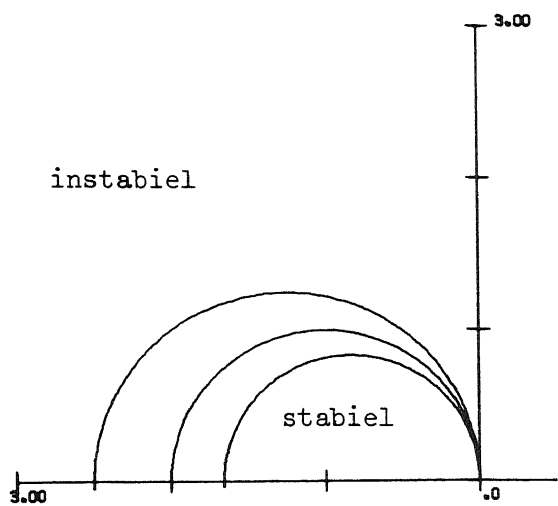
(equidistantie voor de n voorlaatste stappen)

en

$$x_{n+1} - x_0 = p(x_0 - x_1) \quad p = 0.8, 1, 1.2.$$

(de laatste stap kleiner, gelijk of groter dan de voorafgaande).

Uiteraard geldt telkens dat voor kleinere stappen grotere waarden van $-\text{Re}(\lambda)$ geoorloofd zijn.

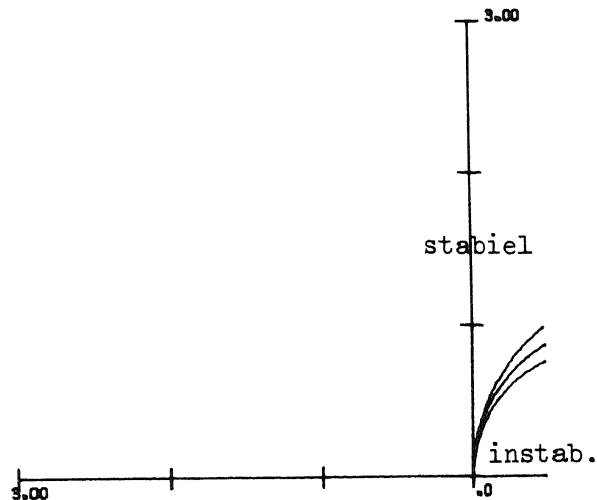
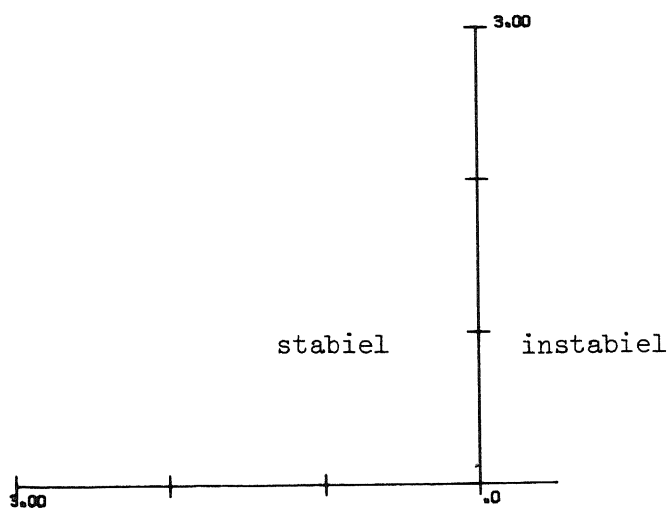
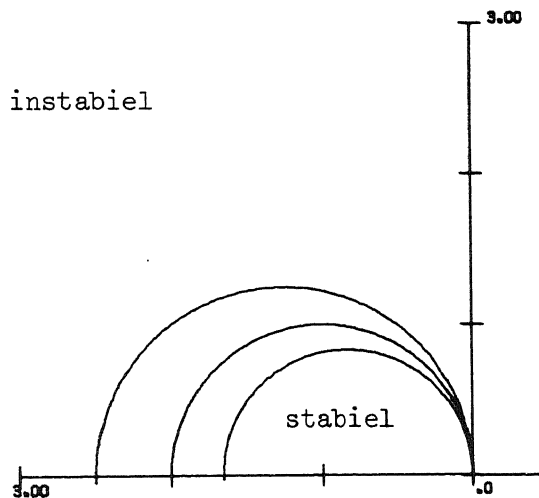
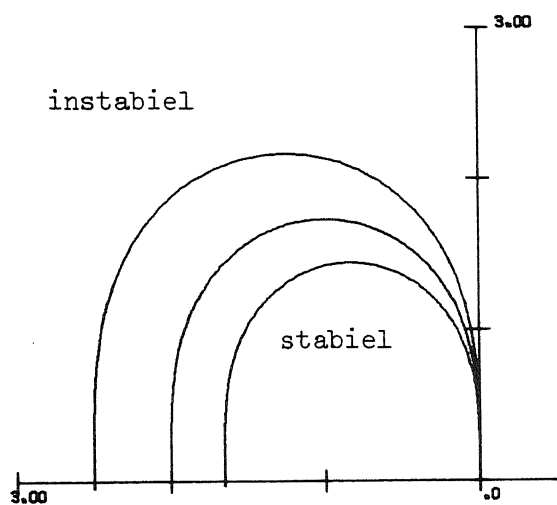


DE STABILITEITSGEBIEDEN
VOOR N = 0

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL

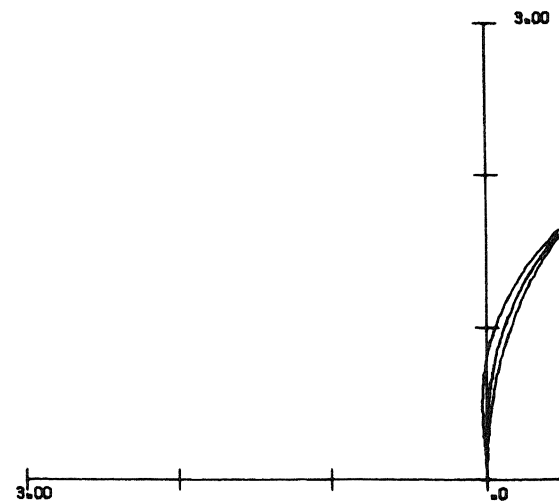
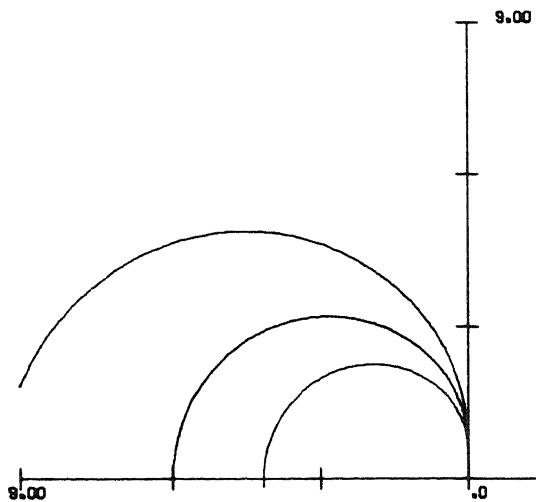
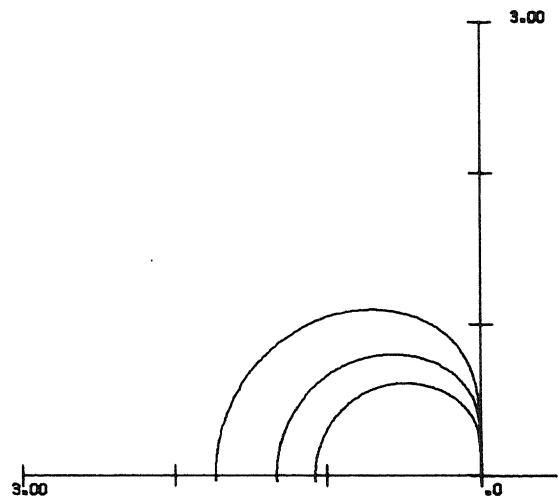
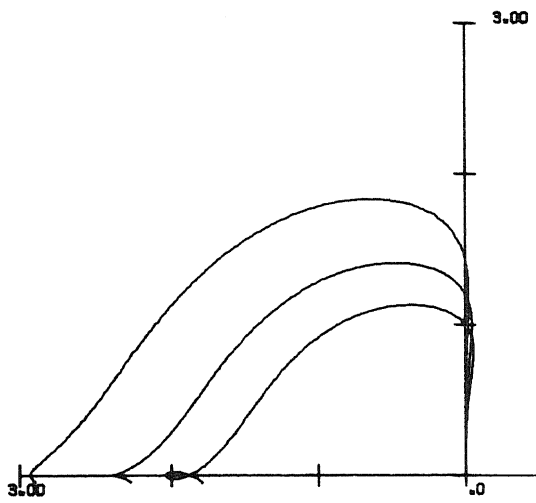
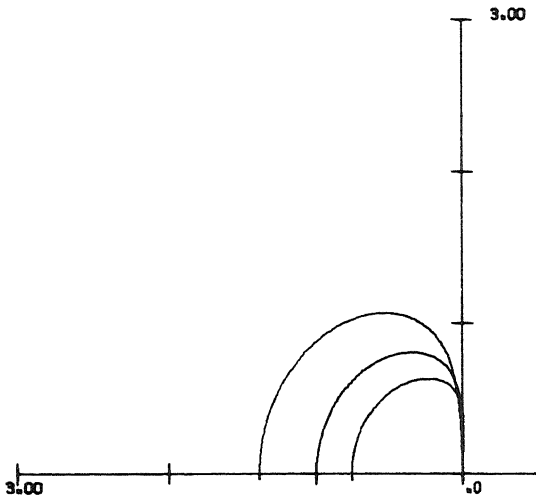


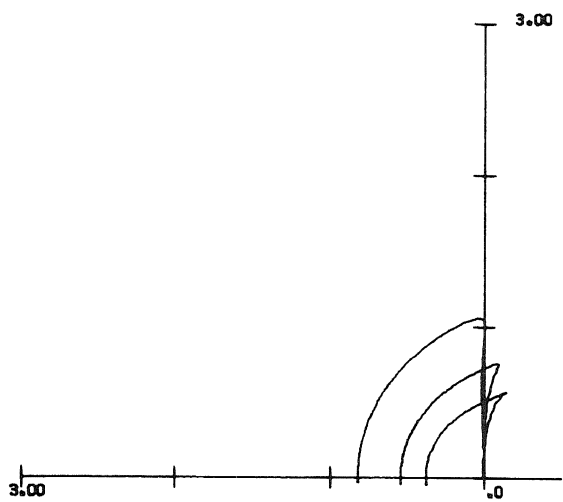
DE STABILITEITSGEBIEDEN
VOOR N = 1

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL



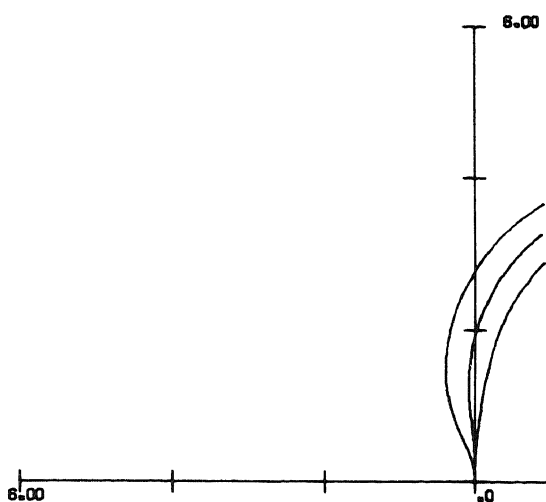
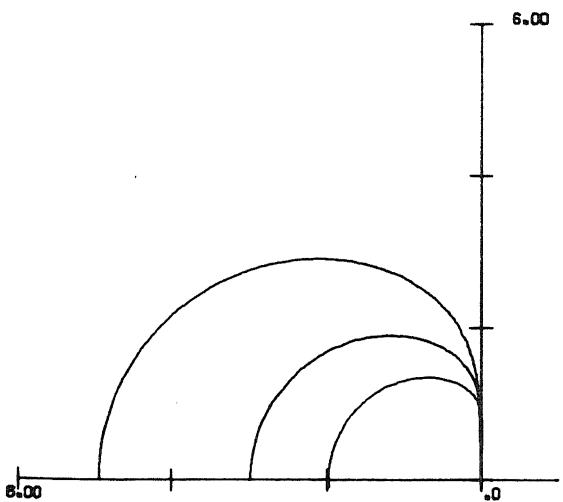
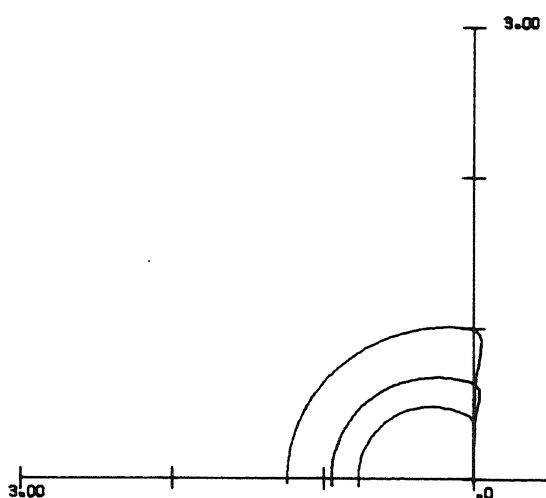
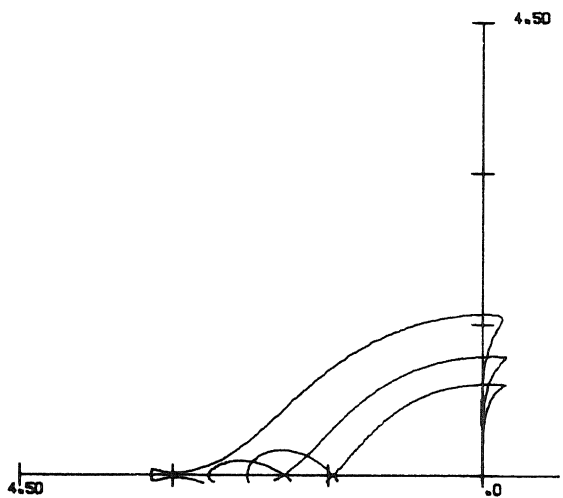


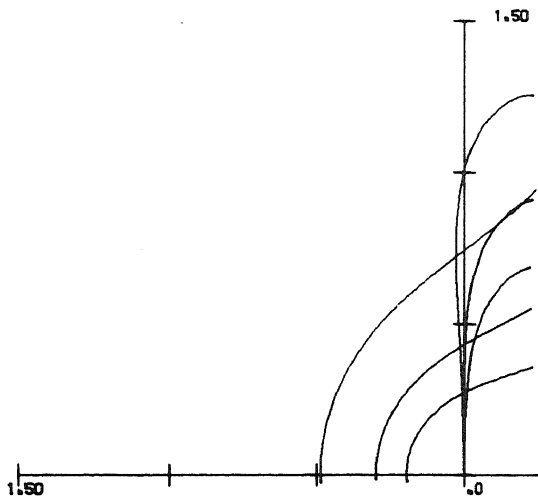
DE STABILITEITSGBIEDEN
VOOR N = 2

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL



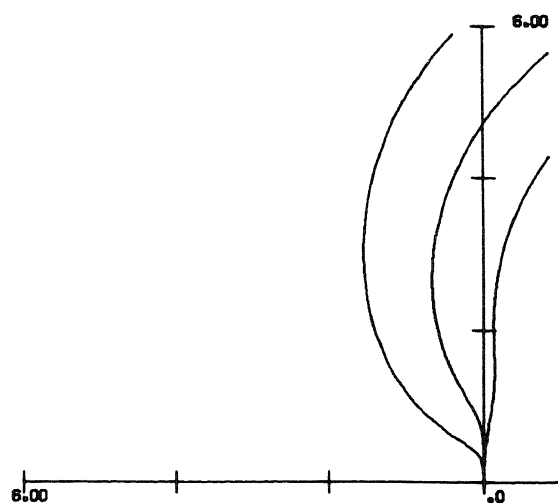
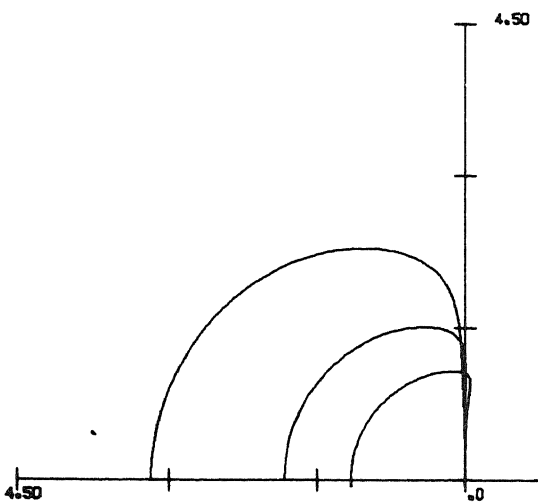
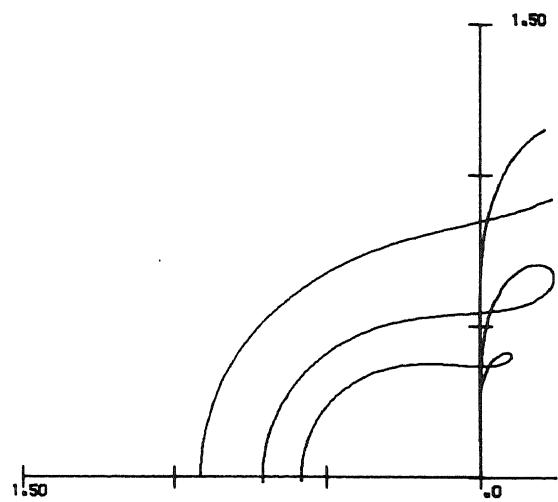
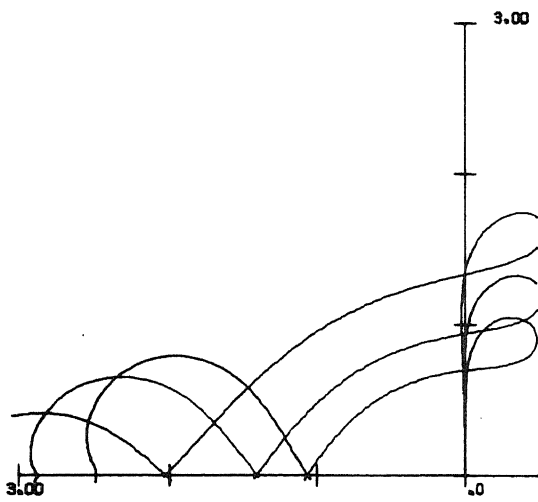


DE STABILITEITSGBIEDEN
VOOR N = 3

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL

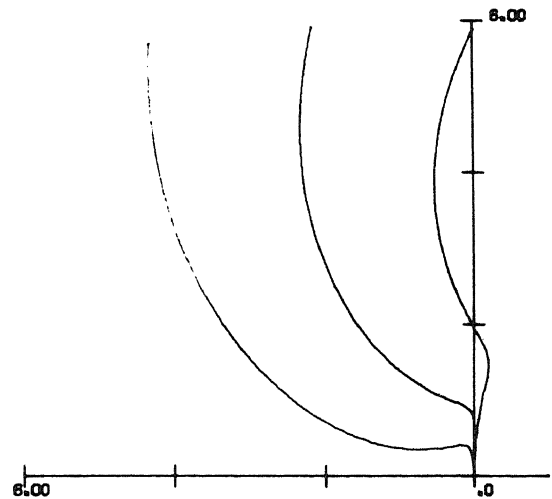
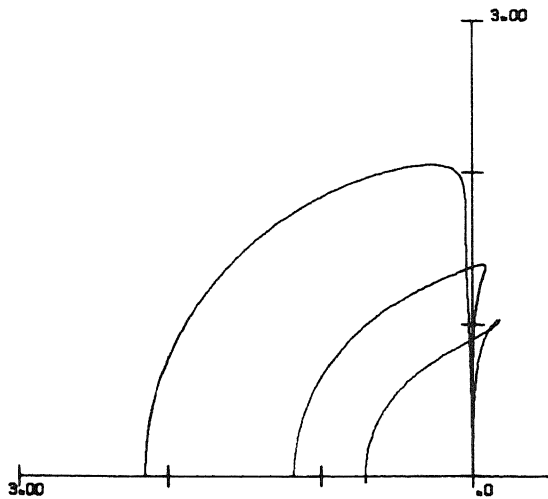
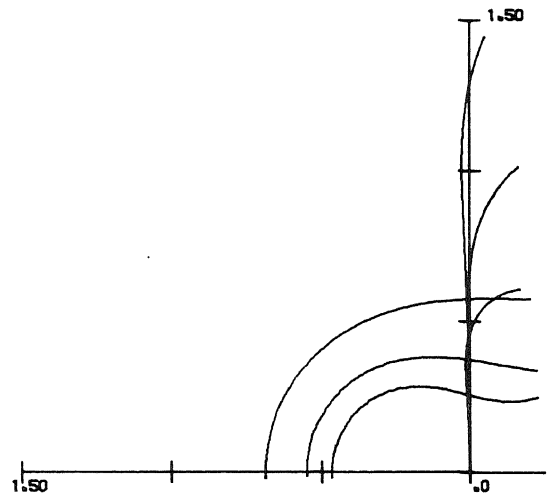
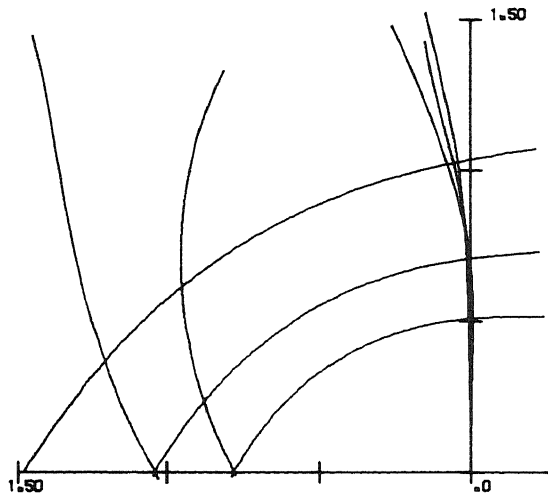
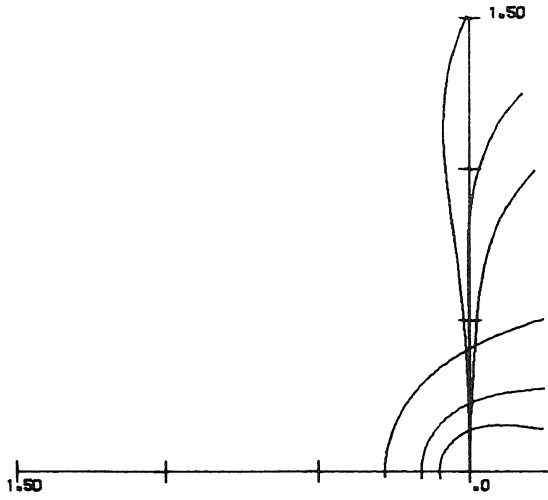


DE STABILITEITSGEBIEDEN
VOOR N = 4

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL

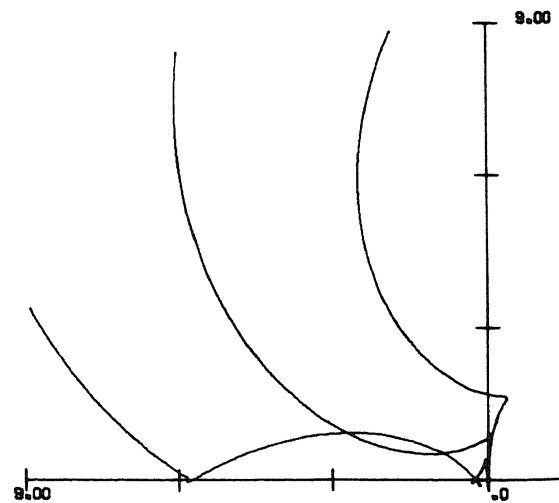
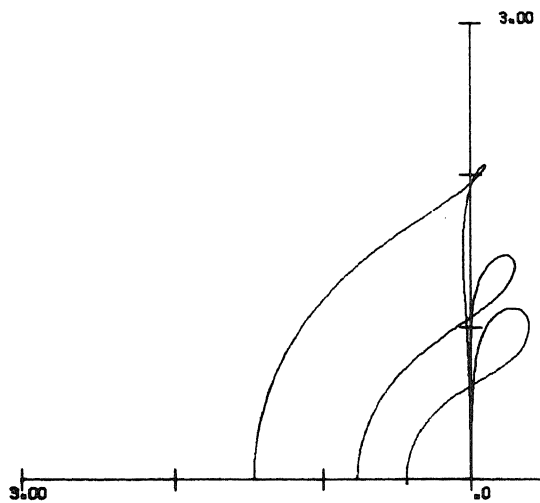
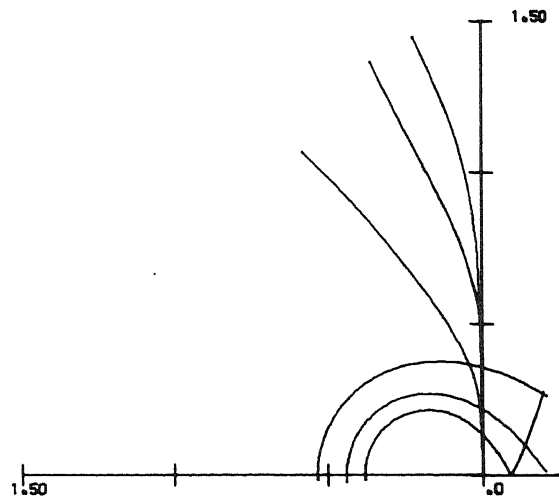
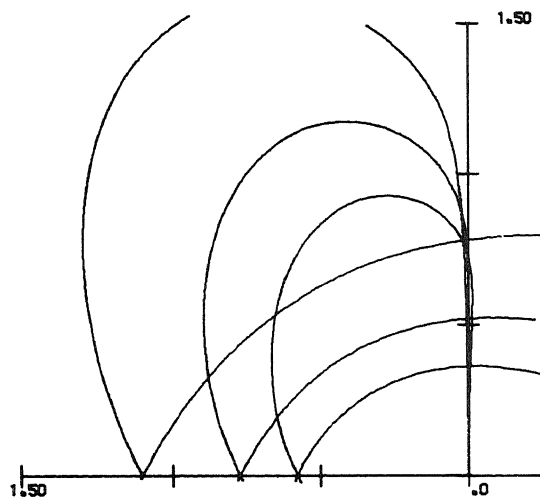
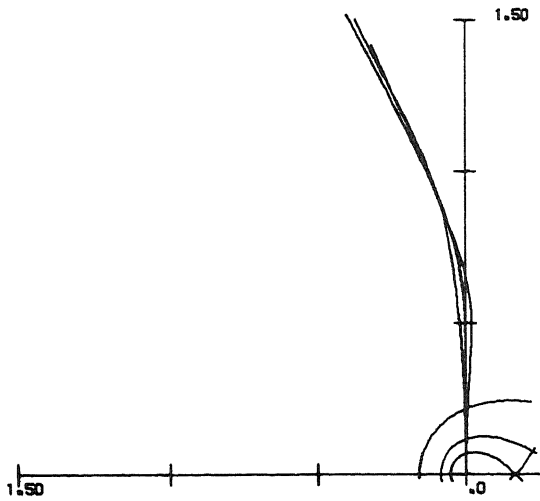


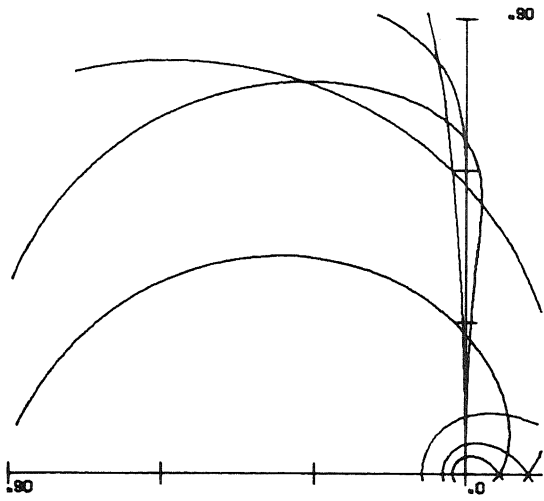
DE STABILITEITSGEBIEDEN
VOOR N = 5

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL



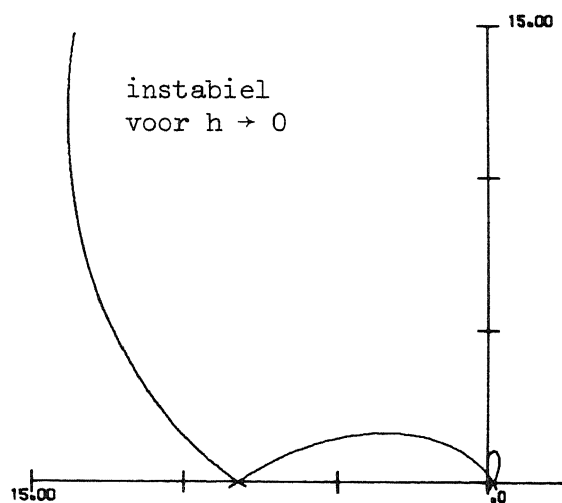
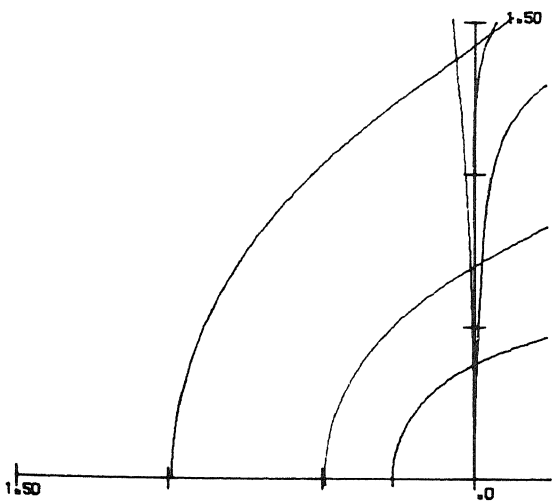
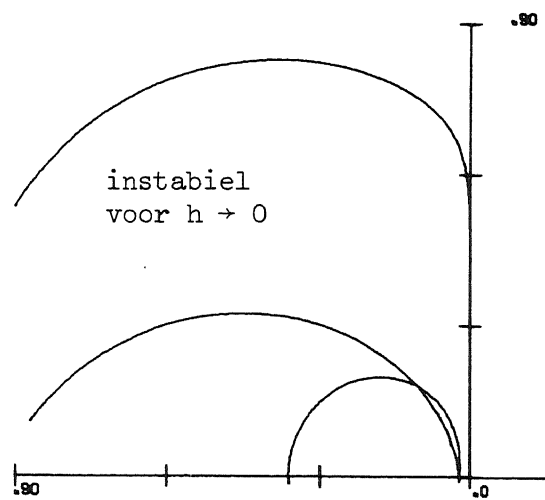
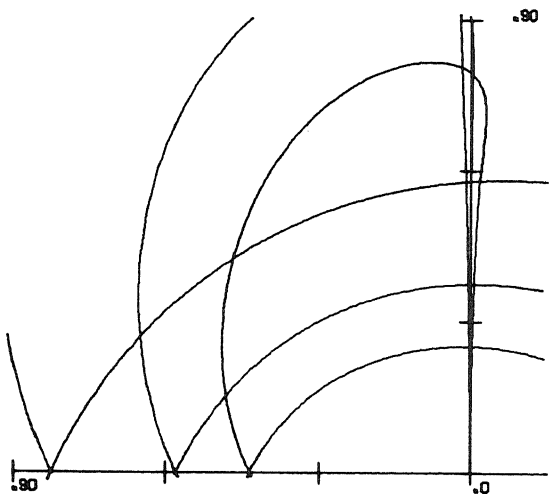


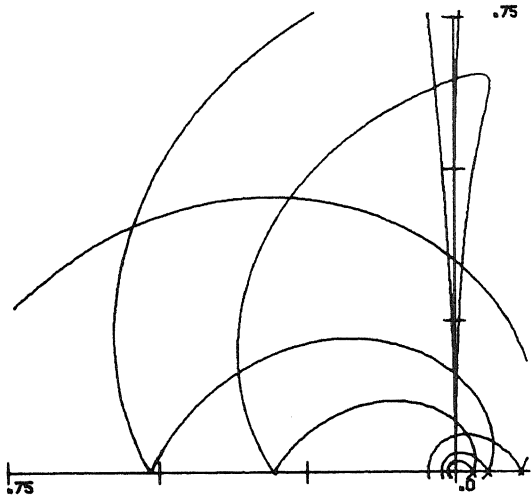
DE STABILITEITSGEBIEDEN
VOOR $N = 6$

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL



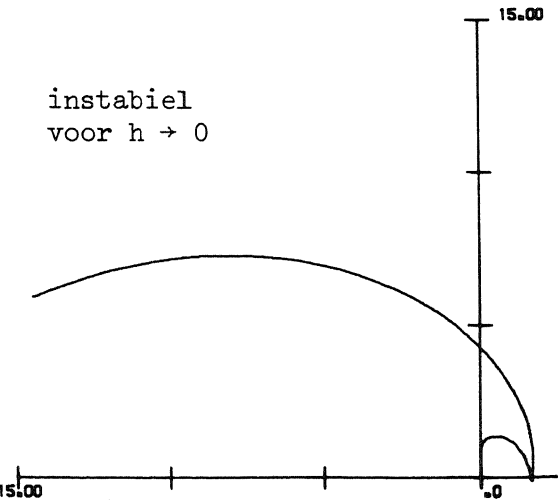
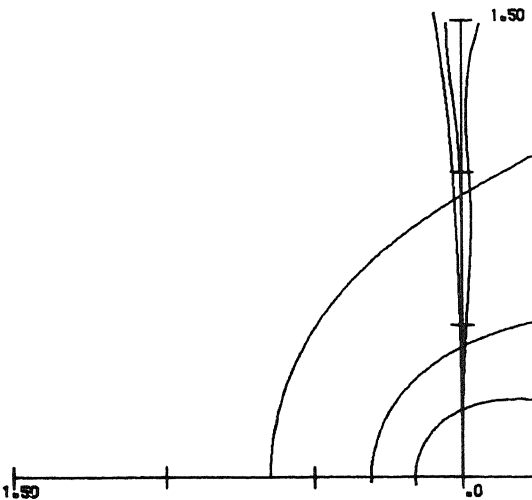
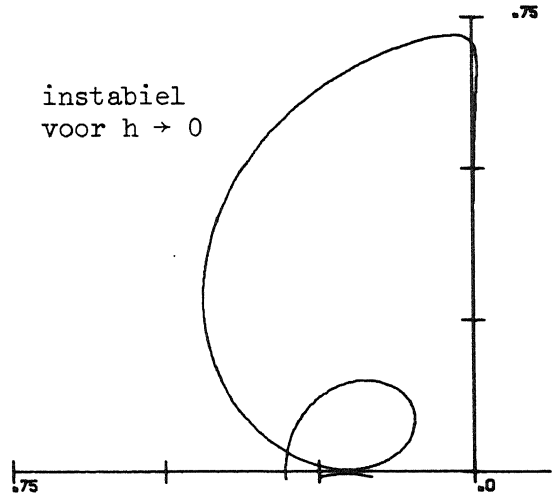
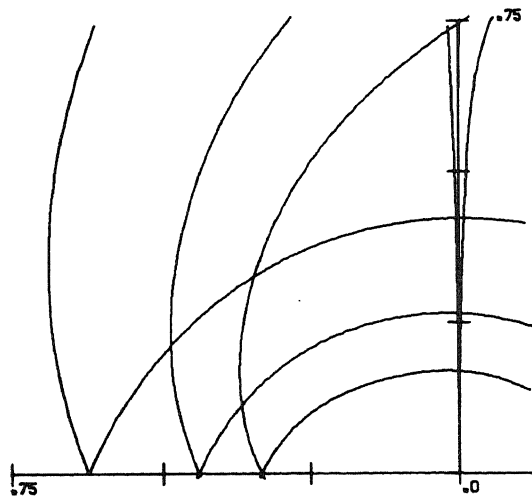


DE STABILITEITSGEBIEDEN
VOOR N = 7

Adams
EP

Adams Curtiss
EPEC PEC

Adams Curtiss
IMPL IMPL



Tot slot laten we aan de hand van stabiliteitsgebieden zien, dat het op grond van stabiliteit beter is bij een corrector van een bepaalde orde een predictor te gebruiken van lagere orde. Deze keuze hebben we dan ook gemaakt, in tegenstelling tot hetgeen volgens de literatuur gebruikelijk is [Crane and Klopfenstein 1965, Meershoek 1969].

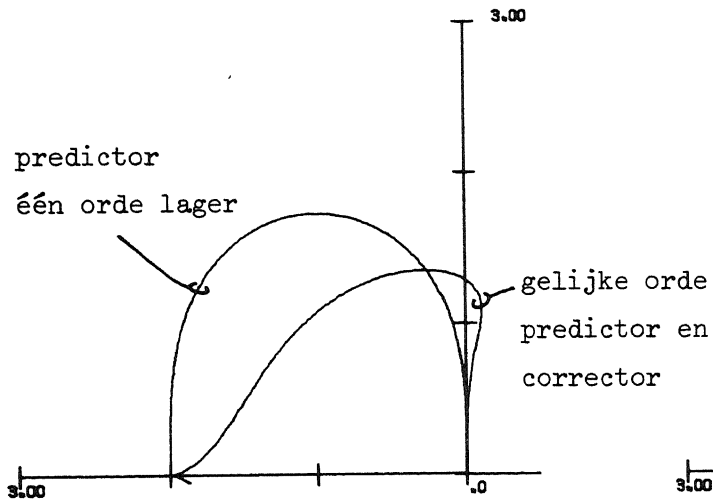
We hebben de stabiliteitsgebieden getekend voor het equidistante geval van de Adams-EPEC en Curtiss-PEC methoden, waarbij predictor- en corrector-formule van dezelfde orde zijn (orde = 2(1)6). Ter vergelijking is in iedere figuur ook het stabiliteitsgebied getekend van de overeenkomstige methode, waarbij de predictor èèn orde lager is. Dit gebied, dat ook op de vorige pagina's te vinden is, is telkens het grootste van de twee.

De stabiliteitsgebieden

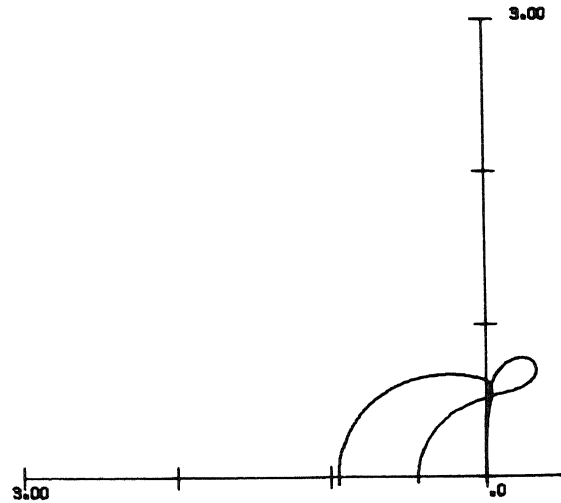
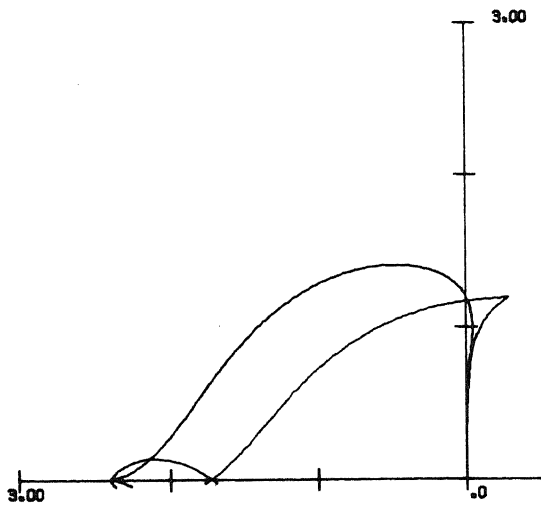
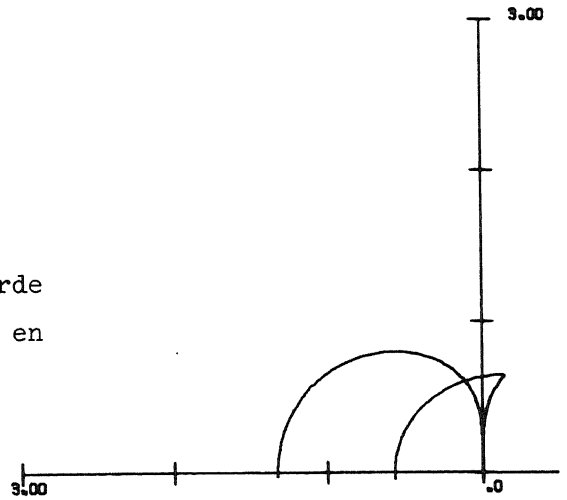
voor methoden met predictor en corrector van dezelfde orde.

Van boven naar beneden zijn de gebieden getekend voor orde = 2, 3, 4, 5, 6.

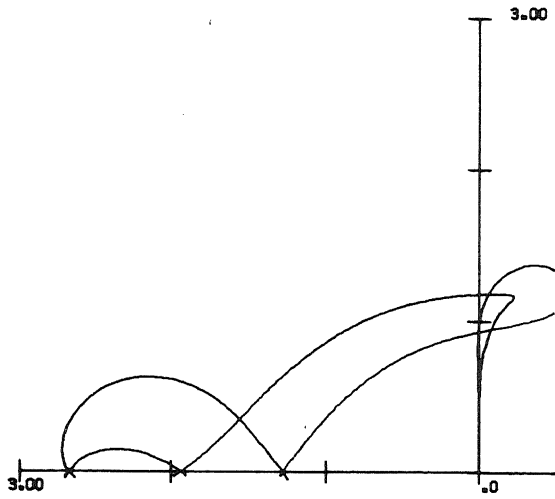
Adams EPEC



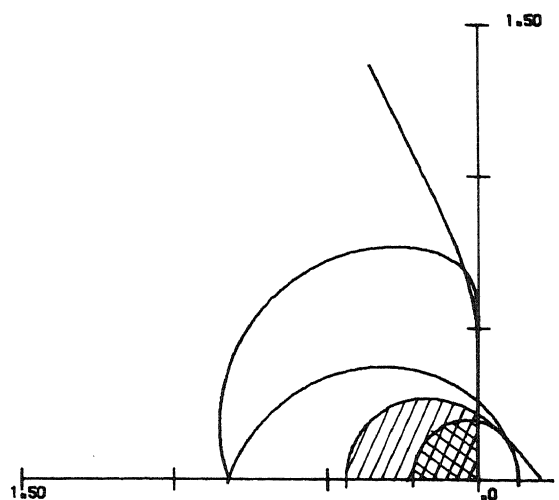
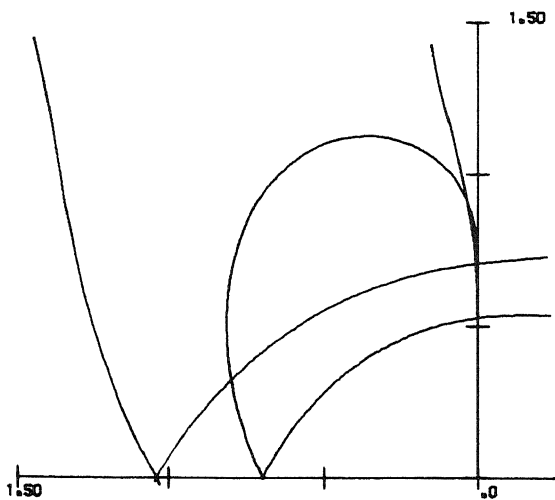
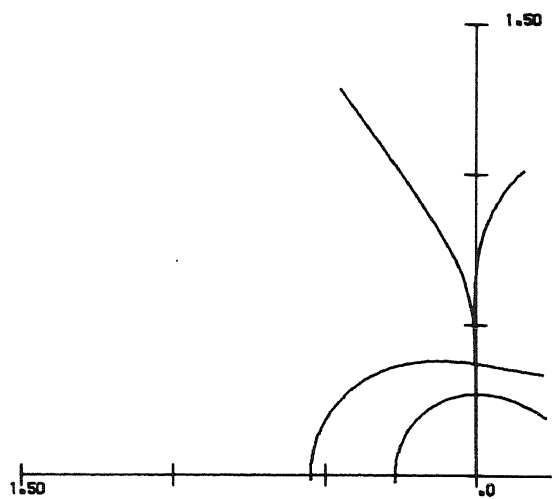
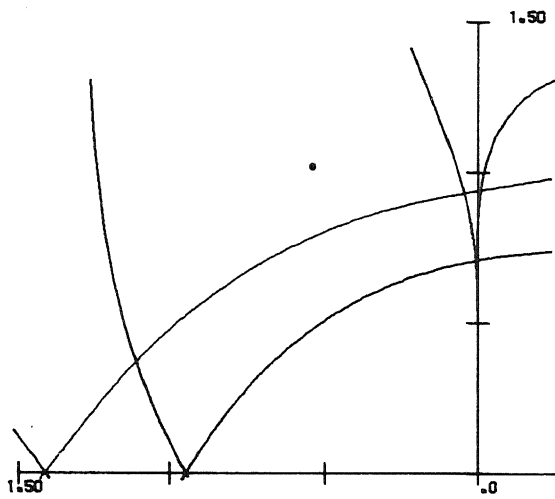
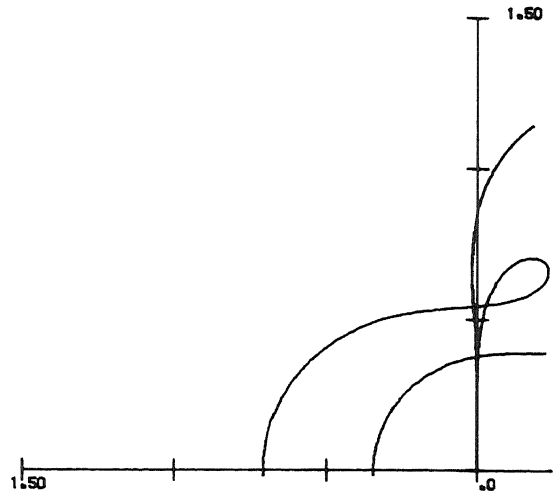
Curtiss PEC



Adams EPEC



Curtiss PEC



Hoofdstuk III. ALGOL 60-procedures en testresultaten

§15. De procedure multistep

In dit hoofdstuk beschrijven we een drietal ALGOL 60-procedures. De procedure multistep bevat de gehele algoritme zoals hij beschreven staat in hoofdstuk I. Hij kan gebruikt worden voor het oplossen van een stelsel differentiaalvergelijkingen van onderling gelijke orde.

De procedures multistep 1 en multistep n bevatten slechts gedeelten van de genoemde algoritme; multistep 1 is bedoeld voor het oplossen van een enkele hogere orde differentiaalvergelijking, multistep n is bedoeld voor het oplossen van een stelsel eerste orde vergelijkingen. In deze twee procedures wordt uitsluitend gebruik gemaakt van de impliciete methoden met gebruikmaking van de Jacobiaan. Met deze eenvoudiger versies is het, door de opgelegde beperkingen, mogelijk de berekening efficiënter uit te voeren.

In alle drie procedures is de keuze tussen Adams- of Curtiss-methoden aan de gebruiker overgelaten, evenals de keuze van de strategieën betreffende het aantal iteratieslagen, de orde en de staplengte. De gebruiker moet zelf deze strategieën aangeven door enige procedures te declareren die als actuele parameters aan multistep (c.q. multistep 1 of multistep n) worden meegegeven.

Gemeenschappelijke parameters.

De volgende parameters zijn voor de drie integratie-procedures gemeenschappelijk:

curtiss: Een Boolean parameter die aangeeft of het stelsel volgens de methode van Curtiss of volgens de methode van Adams opgelost moet worden.

iterproc: Een Boolean procedure met drie parameters, achtereenvolgens itnum, delta y, y.

De waarde van deze procedure geeft aan of de (itnum)-de iteratieslag (dat is berekening van de corrector c.q. Newton-stap) wel of niet uitgevoerd moet worden. Om een beslissing te kunnen nemen

worden in de formele parameters de laatst berekende correctie (delta y) en de daaruit verkregen laatste waarde van y ter beschikking gesteld.

nmax: Een integer parameter die een bovengrens geeft voor de mogelijke waarden van nproc.

nproc: Een integer procedure met drie parameters, achtereenvolgens n, a, u.

De waarde van deze procedure geeft aan dat de volgende stap met een (nproc + 1)-stapsmethode berekend zal worden. (Voor de relatie tussen nproc en orde zie §13). De formele parameter n geeft een bovengrens voor nproc. Om een beslissing te kunnen nemen staan bovendien ter beschikking de array's a en u. Als gerekend wordt met een Adams-methode komt a overeen met a_{K+} en komt u overeen met $U_{\Gamma \times K+}$; als gerekend wordt met een Curtiss-methode komt a overeen met c_{K+} en komt u overeen met $u_{\Gamma \times K+}$ (zie hoofdstuk I en appendix).

hproc: Een real procedure met drie parameters, achtereenvolgens n, a, u.

De waarde van de procedure geeft de staplengte voor de volgende stap; n, a en u hebben dezelfde betekenis als bij nproc met dien verstande dat u overeenkomt met $U_{\Gamma \times K+}$ (c.q. $u_{\Gamma \times K+}$) van de vorige stap.

go on: Een Boolean procedure die aangeeft of de integratie voortgezet moet worden. In het algemeen zal deze procedure als neveneffect het afleveren van resultaten hebben.

De procedure multistep wordt gebruik om stelsels hogere-orde differentiaalvergelijkingen

$$(15.1) \quad y_j^{(D)} = f_j(x, y_1, y_1', \dots, y_1^{(D-1)}, y_2, \dots, y_N, y_N', \dots, y_N^{(D-1)}),$$

$$y_j^{(i)}(x_0) = Y_j^{(i)}, \quad j = 1(1)N, \quad i = 0(1)D-1$$

op te lossen.

Eerst zullen we de formele parameters, die in de procedure voorkomen, bespreken.

N: Het aantal vergelijkingen waaruit het stelsel bestaat.

D: De gemeenschappelijke orde van de vergelijkingen uit het stelsel.

x: Dit is de onafhankelijke variabele. Vóór de aanroep van multistep moet x de startwaarde krijgen; tijdens het rekenproces wordt x telkens verhoogd.

y: De afhankelijke variabele, een array waarvan $y[i,j]$ de waarde van $y_j^{(i)}$ $j = 1(1)N$, $i = 0(1)D$ aangeeft. Vóór de aanroep van multistep moet y de beginvoorwaarden bevatten:

$$y[i,j] := Y_j^{(i)} \quad j = 1(1)N, i = 0(1)D-1.$$

fproc: Een real procedure waarvan de heading luidt:

real procedure fproc (j,x,y);

integer j; real x; array y;

De waarde van de procedure komt overeen met de waarde van f_j in (15.1).

uncoupled: Een Boolean parameter die true gemaakt kan worden als het stelsel volledig ontkoppeld is. De berekening wordt dan efficiënter uitgevoerd.

Fy proc: Een real procedure waarvan de heading luidt:

real procedure Fy proc (x,y,z,i,j);

integer z, i, j; real x; array y;

De waarde van de procedure komt overeen met de waarde

$$\text{van } \frac{\partial f_z}{\partial y_j^{(i)}} \quad 1 \leq j, z \leq N, 0 \leq i < D.$$

with jacobian: Een Boolean parameter die aangeeft of van de

Jacobiaan gebruik gemaakt moet worden. Als with

jacobian = false dan is de waarde van Fy proc irrelevant.

```

procedure multistep (x,y)
    method:(with jacobian) according to:(curtiss)
    problem:(fproc,Fy proc,go on,D,N,uncoupled)
    strategy:(iterproc,nmax,nproc,hproc);
value with jacobian,curtiss,uncoupled,D,N,nmax;
Boolean with jacobian,curtiss,uncoupled; Boolean procedure iterproc,go on;
integer nmax,D,N; integer procedure nproc;
real x; real procedure fproc,Fy proc,hproc;
array y;
begin integer i,j,k,m,n,z,gam,itnum;
    real xh,hh;
    array h[0:nmax], A[0:nmax,0:nmax], b,u[0:D,0:nmax],
        a[0:nmax,1:N], yh,delta yh[0:D,1:N], delta f[1:N];
    real procedure div dif(i,base); value i; integer i; real base;
    begin div dif:= a[i,z];
        a[i,z]:= if i=0 then base
                else (-div dif(i-1,base)+a[i-1,z])/h[i]
    end div dif;

iniz: for m:=0 step 1 until nmax do
    for k:=0 step 1 until nmax do
    A[m,k]:= if m=k then 1 else 0;

    n:= if curtiss then D-1 else 0;
    for i:=0 step 1 until n do n[i]:=0;

    for z:=1 step 1 until N do
    for k:=0 step 1 until n do
    a[k,z]:= if curtiss
        then y[k,z]/facpol(1,k)
        else fproc(z,x,y) ;

step: for k:= 2 step 1 until n+1 do
    for m:= 1 step 1 until k-1 do
    A[m,k]:= h[k-1] x A[m,k-1] + A[m-1,k-1];

width: hh:=hproc(n,a,u); xh:=x + hh;
    for gam:= 0 step 1 until D do
    for m:= 0 step 1 until n+1 do
    b[gam,m]:= if curtiss
        then (if m<gam then 0 else hh^(m-gam)xfacpol(m-gam,m))
        else hh^(m+gam)/facpol(m,m+gam);

    for gam:=0 step 1 until D do
    for k:=0 step 1 until n+1 do
    u[gam,k]:= matmat(0,n+1,gam,k,b,A);

order: n:=nproc(n,a,u);
    for gam:=0 step 1 until D do
    for z:= 1 step 1 until N do
    yh[gam,z]:= if curtiss
        then matmat(0,n, gam,z,u,a)
        else matmat(0,n,D-gam,z,u,a) +
            sum(i,0,D-gam-1,hh^i x y[gam+i,z]/facpol(1,i));

```

```

iter:  for itnum:= 0,itnum+1 while iterproc(itnum,delta yh,yh) do
      begin for z:=1 step 1 until N do
          delta f[z]:= fproc(z,xh,yh) - yh[D,z];

          if 7 with jacobian V uncoupled then
              begin array inprod[1:N];

                  for z:=1 step 1 until N do
                      inprod[z]:= if with jacobian then      (if curtiss
                          then sum(j,0,D-1,u[ j,n+1]xFy proc(xh,yh,z,j,z))
                          else sum(j,0,D-1,u[D-j,n+1]xFy proc(xh,yh,z,j,z)))
                                  else 0 ;

                  for gam:=0 step 1 until D do
                      for z:= 1 step 1 until N do
                          delta yh[gam,z]:= delta f[z]x(if curtiss
                              then u[ gam,n+1]/(u[D,n+1]-inprod[z])
                              else u[D-gam,n+1]/(u[0,n+1]-inprod[z]));
                      end else
              begin array delta yh vector[1:(D+1)XN], matrix[1:(D+1)XN,1:(D+1)XN];

                  for gam:=0 step 1 until D do
                      for z:= 1 step 1 until N do
                          delta yh vector[gam XN+ z]:= delta f[z] x
                              (if curtiss then u[gam,n+1] else u[D-gam,n+1]);

                  for gam:=0 step 1 until D do
                      for z := 1 step 1 until N do
                      for j := 0 step 1 until D do
                      for i := 1 step 1 until N do
                          matrix[ gam XN+ z, j XN+ i]:= if curtiss
                              then (if j=gam ^ i=z then u[D,n+1] else 0)-
                                  (if j=D then 0 else Fy proc(xh,yh,z,j,i)xu[ gam,n+1])
                              else (if j=gam ^ i=z then u[0,n+1] else 0)-
                                  (if j=D then 0 else Fy proc(xh,yh,z,j,i)xu[D-gam,n+1]);

                  detsol(matrix,(D+1)XN,delta yh vector);

                  for j:=0 step 1 until D do
                      for z:=1 step 1 until N do
                          delta yh[j,z]:=delta yh vector[ j XN+ z]
                      end;

                  for gam:=0 step 1 until D do
                      for z:= 1 step 1 until N do
                          yh[gam,z]:=yh[gam,z] + delta yh[gam,z]
                      end;
              end;

```



```

accep: if n+1<nmax then n:=n+1;
      x:=xh;
      for i:=n step -1 until 1 do
        h[i]:=h[i-1]+hh;

      for z:=1 step 1 until N do begin
        div dif (n, if curtiss then yh[0,z] else fproc(z,xh,yh));

      for i:=0 step 1 until D do
        y[i,z]:=yh[i,z] end;

      if go on then goto step

end multistep;

```

comment De procedures sum en facpol luiden a.v.;

```

real procedure sum(i,a,b,x); value b;
integer i,a,b; real x;
begin real s; s:=0;
      for i:=a step 1 until b do s:=s+x;
      sum:=s
end sum;

```

```

real procedure facpol(a,b); value a,b;
integer a,b;
begin real c; integer i; c:=1;
      if a<b then for i:=a+1 step 1 until b do c:=c*i;
      facpol:=c
end facpol;

```

comment Er wordt bovendien gebruik gemaakt van de
procedures matvec,matmat en detsol uit:

T.J.Dekker
ALGOL 60 procedures in numerical algebra,part 1.
Mathematical Centre Tracts 22 (1968);

§16. De procedures multistep 1 en multistep n

De procedure multistep 1 wordt gebruikt om een enkele hogere orde differentiaalvergelijking

$$(16.1) \quad \begin{aligned} y^{(D)} &= f(x, y, y', \dots, y^{(D-1)}), \\ y^{(i)}(x_0) &= Y^{(i)}, \quad i = 0(1)D-1 \end{aligned}$$

op te lossen.

Behalve curtiss, iterproc, nmax, nproc en hproc (zie §15) komen hierin de volgende parameters voor

D: De orde van de differentiaalvergelijking.
 x: De onafhankelijke variabele. Vóór de aanroep van multistep 1 moet x de startwaarde krijgen; tijdens het rekenproces wordt x telkens verhoogd.
 y: De afhankelijke variabele, een array met elementen $y[0], y[1], \dots, y[D]$. $y[i]$ komt overeen met $y^{(i)}$ in (16.1). Vóór de aanroep van multistep 1 moet y de beginvoorwaarden bevatten:
 $y[i] := Y^{(i)}, \quad i = 0(1)D-1.$

fproc: Een real procedure waarvan de heading luidt:
real procedure fproc (x,y);
real x; array y;
 De waarde van de procedure komt overeen met de waarde van f in (16.1).

Fy proc: Een real procedure waarvan de heading luidt:
real procedure Fy proc (x,y,i);
integer i; real x; array y;
 De waarde van de procedure komt overeen met de waarde

$$\frac{\partial f}{\partial y^{(i)}}, \quad 0 \leq i < D.$$

```

procedure multistep 1 (x,y)
    method according to:(curtiss)
    problem:(fproc,Fy proc,go on,D)
    strategy:(iterproc,nmax,nproc,hproc);
value    curtiss,nmax,D;
Boolean curtiss; Boolean procedure iterproc,go on;
integer nmax,D; integer procedure nproc;
real x; real procedure fproc,Fy proc,hproc;
array y;
begin integer i,j,k,m,n,z,gam,itnum;
    real xh,hh,base,real,factor;
    array A[0:nmax,0:nmax], b,u[0:D,0:nmax],
        a,h[0:nmax], yh,delta yh[0:D];

    procedure administration;
    begin for k:= 2 step 1 until n+1 do
        for m:= 1 step 1 until k-1 do
             $A[m,k] := h[k-1] \times A[m,k-1] + A[m-1,k-1];$ 
        for gam:=0 step 1 until D do
            for k:=0 step 1 until n+1 do
                 $u[gam,k] := \text{matmat}(0,k,gam,k,b,A);$ 
            n:=nproc(n,a,u)
        end administration;

    procedure preparation next step (base 0);
    real base 0;
    begin x:=xh; if n+1<nmax then n:=n+1;
        for i:=n step -1 until 1 do h[i]:=h[i-1]+hh;
        base:= base 0;
        for i:=0 step 1 until n-1 do
            begin real:= (base-a[i])/h[i+1];
                a[i]:=base; base:=real
            end; a[n]:=base;
        for i:=0 step 1 until D do y[i]:=yh[i]
    end preparation;

    for m:=0 step 1 until nmax do
        begin A[0,m]:=0; A[m,m]:=1 end;

```

```

if curtiss then
begin n:= D-1;
  For k:=0 step 1 until n do
  begin h[k]:=0; a[k]:= y[k]/facpol(1,k) end;
  For hh:=hproc(n,a,u) while go on do
  begin xh:=x + hh;
    For gam:= 0 step 1 until D do
    For m:= 0 step 1 until n+1 do
    b[gam,m]:= if m<gam then 0 else hh^(m-gam)×facpol(m-gam,m);
    administration;
    for gam:=0 step 1 until D do
    yh[gam]:= matvec(0,n,gam,u,a);

    for itnum:= 0,itnum+1 while iterproc(itnum,delta yh,yh) do
    begin factor:=( fproc(xh,yh) - yh[D])/
      (u[D,n+1]-sum(j,0,D-1,u[j,n+1]×Fy proc(xh,yh,j)));
      for gam:=0 step 1 until D do
      begin delta yh[gam]:= u[gam,n+1] × factor;
      yh[gam]:=yh[gam] + delta yh[gam]
      end
    end iteration;

    preparation next step (yh[0])
  end end else

begin n:=0; h[0]:=0; a[0]:=fproc(x,y);
  For hh:=hproc(n,a,u) while go on do
  begin xh:=x + hh;
    For gam:= 0 step 1 until D do
    For m:= 0 step 1 until n+1 do
    b[gam,m]:= hh^(m+gam)/facpol(m,m+gam);
    administration;
    for gam:=0 step 1 until D do
    yh[gam]:= matvec(0,n,D-gam,u,a) +
      sum(i,0,D-gam-1,hh^i×y[gam+i]/facpol(1,i));

    for itnum:= 0,itnum+1 while iterproc(itnum,delta yh,yh) do
    begin factor:= (fproc(xh,yh) - yh[D])/
      (u[0,n+1]-sum(j,0,D-1,u[D-j,n+1]×Fy proc(xh,yh,j)));
      for gam:=0 step 1 until D do
      begin delta yh[gam]:= u[D-gam,n+1] × factor;
      yh[gam]:=yh[gam] + delta yh[gam]
      end
    end iteration;

    preparation next step (fproc(xh,yh));
  end end
end multistep 1;

comment zie pag.53 ;

```

De procedure multistep n wordt gebruikt om een stelsel eerste orde differentiaalvergelijkingen

$$(16.2) \quad \begin{aligned} y'_j &= f_j(x, y_1, y_2, \dots, y_N), \\ y_j(x_0) &= Y_j, \quad j = 1(1)N \end{aligned}$$

op te lossen.

Behalve curtiss, iterproc, nmax, nproc en hproc (zie §15) komen hierin de volgende parameters voor

- N: Het aantal vergelijkingen waaruit het stelsel bestaat.
- x: De onafhankelijke variabele. Vóór de aanroep van multistep n moet x de startwaarde krijgen; tijdens het rekenproces wordt x telkens verhoogd.
- y: De afhankelijke variabelen, een array met elementen $y[0,1], y[0,2], \dots, y[0,N], y[1,1], \dots, y[1,N]$.
 $y[0,j]$ komt overeen met y_j in (16.2)
 en
 $y[1,j]$ komt overeen met y'_j in (16.2).
 Vóór de aanroep van multistep n moet y de beginvoorwaarden bevatten:
 $y[0,j] := Y_j, \quad j = 1(1)N$
- fproc: Een real procedure waarvan de heading luidt:
real procedure fproc (j,x,y);
integer j; real x; array y;
 De waarde van de procedure komt overeen met de waarde van f_j in (16.2).
- Fy proc: Een real procedure waarvan de heading luidt:
real procedure Fy proc (x,y,z,i);
integer z, i; real x; array y;
 De waarde van de procedure komt overeen met
 $\frac{\partial f}{\partial y_i}, \quad 1 \leq z, i \leq N.$

```

procedure multistep n (x,y)
    method according to:(curtiss)
    problem:(fproc,Fy proc,go on,N)
    strategy:(iterproc,nmax,nproc,hproc);
value curtiss,nmax,N;
Boolean curtiss; Boolean procedure iterproc,go on;
integer nmax,N; integer procedure nproc;
real x; real procedure fproc,Fy proc,hproc;
array y;
begin integer i,j,k,m,n,z,gam,gNz,itnum;
    real hh,delta f,real,base;
    array h[0:nmax], A[0:nmax,0:nmax], b,u[0:1,0:nmax],
        a[0:nmax,1:N], delta y[1:2×N], matrix[1:2×N,1:2×N];

    procedure administration;
    begin for k:=2 step 1 until n+1 do
        for m:=1 step 1 until k-1 do
            A[m,k]:= h[k-1] × A[m,k-1] + A[m-1,k-1];
        for k:=0 step 1 until n+1 do for gam:=0,1 do
            u[gam,k]:=matmat(0,k,gam,k,b,A);
            n:=nproc(n,a,u)
        end administration;
    end administration;

    procedure preparation next step (base 0);
    begin if n+1<nmax then n:=n+1;
        for i:=n step -1 until 1 do h[i]:=h[i-1]+hh;
        for z:=1 step 1 until N do
            begin base:= base 0;
                for k:=0 step 1 until n-1 do
                    begin real:=(base-a[k,z])/h[k+1];
                        a[k,z]:=base; base:=real
                    end;a[n,z]:=base
                end
            end preparation;
    end preparation;

    procedure solve y;
    begin detsol(matrix,2×N,delta y);
        for z:=1 step 1 until N do for gam:=0,1 do
            y[gam,z]:=y[gam,z] + delta y[gam ×N+ z]
        end solve y;

    n:=0; h[0]:=0;
    for m:=0 step 1 until nmax do
        begin A[0,m]:=0; A[m,m]:=1 end;

```

```

if curtiss then
begin for z:=1 step 1 until N do a[0,z]:= y[0,z];
  for hh:=hproc(n,a,u) while go on do
  begin x:= x + hh;
    for m:= 0 step 1 until n+1 do for gam:= 0,1 do
    b[gam,m]:= if m<gam then 0 else hh^(m-gam)*facpol(m-gam,m);
    administration;
    for z:= 1 step 1 until N do for gam:=0,1 do
    y[gam,z]:= matmat(0,n,gam,z,u,a);
    for itnum:= 0,itnum+1 while iterproc(itnum,delta y,y) do
    begin for z:=1 step 1 until N do
      begin delta f:= fproc(z,x,y) - y[1,z];
        for gam:=0,1 do
        begin gNz:= gam *N+ z;
          delta y[gNz]:= delta f * u[gam,n+1];
          for i:= 1 step 1 until N do
          begin matrix[gNz,i+N]:= 0;
            matrix[gNz,i]:= -Fy proc(x,y,z,i)*u[gam,n+1]
          end;matrix[gNz,gNz]:=matrix[gNz,gNz] + u[1,n+1]
        end end;
        solve y;
      end iteration;
    preparation next step(y[0,z]);
  end end else
  begin for z:=1 step 1 until N do a[0,z]:= fproc(z,x,y);
    for hh:=hproc(n,a,u) while go on do
    begin x:= x + hh;
      for m:= 0 step 1 until n+1 do for gam:= 0,1 do
      b[gam,m]:= hh^(m+gam)/facpol(m,m+gam);
      administration;
      for z:= 1 step 1 until N do
      begin y[0,z]:= matmat(0,n,1,z,u,a) + y[0,z];
        y[1,z]:= matmat(0,n,0,z,u,a)
      end;
      for itnum:= 0,itnum+1 while iterproc(itnum,delta y,y) do
      begin for z:=1 step 1 until N do
        begin delta f:= fproc(z,x,y) - y[1,z];
          for gam:=0,1 do
          begin gNz:= gam *N+ z;
            delta y[gNz]:= delta f * u[1-gam,n+1];
            for i:= 1 step 1 until N do
            begin matrix[gNz,i+N]:= 0;
              matrix[gNz,i]:= -Fy proc(x,y,z,i)*u[1-gam,n+1]
            end;matrix[gNz,gNz]:= matrix[gNz,gNz] + u[0,n+1]
          end end;
          solve y;
        end iteration;
      preparation next step(fproc(z,x,y));
    end end
  end multistep n;

comment zie pag.53 ;

```

§17. Twee lineaire testvoorbeelden

In deze en de volgende paragraaf zullen we enkele resultaten laten zien van berekeningen, uitgevoerd met de beschreven procedures. In deze paragraaf behandelen we twee voorbeelden waarvan de analytische oplossing bekend is. In §18 behandelen we een stelsel niet-lineaire differentiaalvergelijkingen dat zijn oorsprong vond bij het simuleren van een biochemisch proces.

De doeltreffendheid van de beschreven integratie-procedures, voor een bepaald probleem, is geheel afhankelijk van de wijze waarop ze gestuurd worden door de strategie-bepalende procedures hproc, nproc en iterproc. Om het gedrag van de integratie-procedure niet achter het effect van een gekozen strategie te laten schuilgaan, willen we in de testproblemen van deze en de volgende paragraaf de strategie-bepalende procedures bijzonder eenvoudig houden.

De procedures die we gebruikten voor de lineaire problemen luiden als volgt:

```

Boolean procedure iterproc (itnum, delta y, y);
    integer itnum; array delta y, y;
    iterproc:= false;
integer procedure nproc (n,a,u);
    integer n; array a, u;
    nproc:= if n > nmax then nmax else n;
real procedure hproc (n,a,u);
    integer n; array a, u;
    hproc:= if x > hmax then hmax else
           if x = 0 then hmax * 2 ↑ (-1-nmax) else x;

```

N.B. Voor lineaire differentiaalvergelijkingen is deze procedure iterproc, behalve de eenvoudigst denkbare, ook de beste.

We zullen laten zien wat het effect is voor verschillende waarden van hmax (d.i. de staplengte nà de start) en nmax (nà de start is sprake van een (nmax + 1)-staps methode).

Voorbeeld 1

Eerst zullen we een differentiaalvergelijking behandelen die vrijwel gelijk is aan een voorbeeld dat gegeven wordt door Curtiss en Hirschfelder [1952]. Deze differentiaalvergelijking luidt

$$y' = -1000y + x^2,$$

$$y(0) = 1.$$

De analytische oplossing luidt

$$\tilde{y}(x) = (10^{-3}x^2 - 2 \cdot 10^{-6}x + 2 \cdot 10^{-9}) + (1 - 2 \cdot 10^{-9})e^{-1000x}.$$

We berekenen met de procedure multistep 1 de waarden van $y(x)$ voor $x = 0.1(0.1)1$.

We geven in het volgende overzicht de globale fout voor verschillende waarden van h_{\max} en n_{\max} .

De globale fout wordt hier gedefinieerd als

$$\max_{x=0.1(0.1)1} |y(x) - \tilde{y}(x)|.$$

In het overzicht staat de letter e voor 10 . Wanneer een bepaalde keuze van h_{\max} en n_{\max} instabiliteit veroorzaakt (zich manifesterend als exponentiele toename van de globale fout), staat dit aangegeven met I.

		Adams-methode				
nmax \ hmax	0	1	2	3	4	
0.1	.85	I	I	I	I	
0.05	.67	I	I	I	I	
0.02	.20	I	I	I	I	
0.01	.48 e- 2	I	I	I	I	
0.005	.13 e- 8	.27 e- 2	I	I	I	
0.002	.18 e-14	.18 e-14	.29 e- 8	I	I	
0.001	.89 3-15	.89 e-15	.89 e-15	.89 e-15	.46 e- 8	
0.0005	.18 e-14	.36 e-14	.18 e-14	.36 e-14	.18 e-14	

		Curtiss-methode				
nmax \ hmax	0	1	2	3	4	
0.1	.38 e- 3	.19 e- 2	.16 e- 1	.20	.34 e+ 1	
0.05	.29 e- 4	.27 e- 3	.39 e- 2	.12 e- 1	.42 e+ 1	
0.02	.62 e- 7	.18 e- 5	.70 e- 3	.25 e- 1	.59	
0.01	.10 e- 7	.55 e- 7	.17 e- 4	.19 e- 2	.15	
0.005	.50 e- 8	.24 e-12	.58 e- 8	.26 e- 5	.97 e- 2	
0.002	.20 e- 8	.18 e-14	.18 e-14	.32 e-10	.61 e- 5	
0.001	.10 e- 8	.18 e-14	.89 e-15	.89 e-15	.25 e-11	
0.0005	.50 e- 9	.18 e-14	.36 e-14	.18 e-14	.18 e-14	

Voorbeeld 2

Als tweede voorbeeld nemen we een lineair stelsel met ver uiteenliggende eigenwaarden, ontleend aan Fowler en Warten [1969]. We kiezen het stelsel

$$\vec{y}' = \begin{pmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{pmatrix} \vec{y} + \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

$$\vec{y}(0) = \begin{pmatrix} -0.1 \\ +0.1 \end{pmatrix}.$$

De eigenwaarden van de Jacobiaan zijn -1 en -1000 .

De analytische oplossing luidt

$$\vec{y}(x) = -2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{-x} - 0.1 \begin{pmatrix} 1 \\ -1 \end{pmatrix} e^{-1000x} + \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

We berekenen $\vec{y}(x)$ met de procedure multistep n voor $x = 0.1(0.1)1$.

De strategie-bepalende procedures waren dezelfde als die in het vorige voorbeeld. Hier definiëren we de globale fout als

$$\max_{x=0.1(0.1)1} \|\vec{y}(x) - \vec{y}(x)\|_{\infty}.$$

We geven weer een overzicht van de globale fout voor verschillende waarden van h_{\max} en n_{\max} .

Als we de resultaten vergelijken met die van het vorige voorbeeld moeten we opmerken

- 1° dat de kleinste eigenwaarde van de Jacobiaan in beide gevallen -1000 is;
- 2° dat in het eerste voorbeeld de asymptotische oplossing van de tweede graad is.

Adams-methode

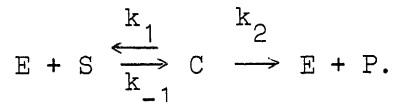
nmax \ hmax	0	1	2	3	4
0.1	.85 e- 1	I	I	I	I
0.05	.67 e- 1	I	I	I	I
0.02	.20 e- 1	I	I	I	I
0.01	.48 e- 3	I	I	I	I
0.005	.15 e- 5	.27 e- 3	I	I	I
0.002	.24 e- 6	.30 e- 9	.29 e- 9	I	I
0.001	.61 e- 7	.13 e- 9	.89 e-10	.12 e- 9	.46 e- 9

Curtiss-methode

nmax \ hmax	0	1	2	3	4
0.1	.34 e- 1	.17 e- 2	.16 e- 2	.20 e- 1	.34
0.05	.18 e- 1	.47 e- 3	.45 e- 3	.16 e- 2	.42
0.02	.72 e- 2	.78 e- 4	.83 e- 4	.26 e- 2	.60 e- 1
0.01	.36 e- 2	.19 e- 4	.49 e- 5	.21 e- 3	.15 e- 1
0.005	.18 e- 2	.50 e- 5	.79 e- 6	.41 e- 5	.98 e- 3
0.002	.73 e- 3	.80 e- 6	.13 e- 6	.61 e- 6	.36 e- 5
0.001	.37 e- 3	.20 e- 6	.32 e- 7	.15 e- 6	.74 e- 6

§18. Een stijve differentiaalvergelijking uit de biochemie

In de enzymkinetica (de leer van de reactiesnelheden van chemische reacties waarbij enzymen betrokken zijn) speelt de volgende chemische reactie een belangrijke rol (zie Dixon en Webb [1965] en Ottaway ed. [1969])



Het gedrag van de concentraties S en C (afhankelijk van de tijd) kan men beschrijven met een stelsel differentiaalvergelijkingen:

$$\frac{d}{dt} S = -k_1(E_0 - C)S + k_{-1}C$$

$$\frac{d}{dt} C = k_1(E_0 - C)S - (k_2 + k_{-1})C$$

met de beginvoorwaarden $C(0) = 0$, $S(0) = S_0$ en $E(0) = E_0$.

Met de volgende substituties kunnen we het probleem in een dimensieloze vorm schrijven.

$$s(t) = S/S_0$$

$$c(t) = C/E_0$$

$$p = (k_2 + k_{-1}) / (k_1 S_0)$$

$$q = k_{-1} / (k_1 S_0)$$

$$\varepsilon = E_0 / S_0$$

$$\tau = t k_1 E_0.$$

Als resultaat krijgen we

$$(18.1) \quad \left\{ \begin{array}{l} \frac{d}{d\tau} s = -(1-c)s + qc \\ \varepsilon \frac{d}{d\tau} c = (1-c)s - pc \end{array} \right.$$

$$s(0) = 1, c(0) = 0.$$

Aangezien in de praktijk de concentratie van S in het algemeen veel groter is dan de concentratie van E en C samen, is ε een kleine parameter (vgl. Hemker en Hemker [1969]). Na enige berekening kan men ook aantonen dat het stelsel, waarvan geen analytische oplossing bekend is, een typisch stijf karakter heeft.

De Jacobiaan van het stelsel luidt

$$\begin{pmatrix} -(1-c) & q+s \\ (1-c)/\varepsilon & -(p+s)/\varepsilon \end{pmatrix}$$

voor de eigenwaarden geldt

$$\lambda_{\max} < \lambda_{\min} < 0$$

en

$$2\left(1 + \frac{\lambda_{\max}}{\lambda_{\min}}\right) > \frac{(\lambda_{\min} + \lambda_{\max})^2}{\lambda_{\min} \cdot \lambda_{\max}} = \frac{\{(1-c) + (p+s)/\varepsilon\}^2}{(1-c)(p-q)/\varepsilon} > \frac{(p+s)^2}{\varepsilon(p-q)}.$$

Hieruit volgt dat de eigenwaarden van de Jacobiaan sterk uiteenliggen als ε en $(p-q)$ kleine waarden hebben. (In de praktijk is $\varepsilon \sim 10^{-4}$ en $(p-q) \sim 10^{-4}$ niet ongebruikelijk.)

We hebben het stelsel, met de procedure multistep n, numeriek opgelost voor de parameters

$$\varepsilon = 0.001, p = 1, q = 0.99.$$

We berekenden de waarde van $s(\tau)$ en $c(\tau)$ voor $\tau = 1(1)50$.

Vanwege de zeer uiteenliggende eigenwaarden gebruikten we uitsluitend de Curtiss-methode. De strategie waarmee multistep n bestuurd werd was de volgende:

```

Boolean procedure iterproc (itnum, delta y, y);
    integer itnum; array delta y, y;
    iterproc:= itnum < 6 ^
        (delta y[1]>y[0,1]*0.110-5 v delta y[2]>y[0,2]*0.110-5);
integer procedure nproc (n,a,u);
    integer n; array a, u;
    nproc:= if n > nmax then nmax else
        if a[1,2] > 1 then 0 else n;
real procedure hproc (n,a,u);
    integer n; array a, u;
    begin real h; if x = 0 then hproc:= eps/(2* (p+1)) else
        begin h:= 0.05/abs(a[1,2]);
            if x + h > eindpunt then h:= eindpunt - x;
            hproc:= h
        end
    end    end;

```

Aangezien $a[1,2]$ gelijk is aan het eerste differentiequotient van de tweede onbekende (c) houdt dit in dat we een $(n_{\max}+1)$ -staps methode gebruiken, nadat we het begin snel hebben doorlopen m.b.v. de 1-staps methode.

Voor verschillende waarden van n_{\max} vergeleken we de resultaten met die welke, met voldoende precisie, verkregen waren met de procedure RK1n, zie Zonneveld [1964].

nmax	$\max_{x=1(1)50} s(x) - \tilde{s}(x) $	$\max_{x=1(1)50} c(x) - \tilde{c}(x) $
0	.30 e-3	.10 e-3
1	.43 e-6	.23 e-5
2	.19 e-5	.19 e-2
3	.42 e-5	.42 e-2
4	.64 e-5	.64 e-2
5	.86 e-5	.86 e-2
6	.11 e-4	.11 e-1

Uit de resultaten bleek bovendien dat voor grotere nmax de onnauwkeurigheid die door het begineffect veroorzaakt wordt, langer doorwerkte.

We geven, in het volgende overzicht, telkens het eind van het begineffect (b) en de resterende globale fout.

nmax	$\max_{x=b(1)50} s(x) - \tilde{s}(x) $		$\max_{x=b(1)50} c(x) - \tilde{c}(x) $	
0	1	.30 e-3	1	.10 e-3
1	1	.43 e-6	2	.14 e-6
2	2	.99 e-7	5	.28 e-7
3	4	.25 e-7	9	.68 e-8
4	5	.11 e-6	9	.48 e-7
5	5	.39 e-6	12	.10 e-6
6	6	.85 e-6	12	.88 e-6
	b		b	

Opmerkelijk is het verschil in aantal benodigde functie-evaluaties bij gebruik van een Runge-Kutta-methode en de Curtiss-methoden. Voor RK1n moest ±160000 maal een rechterlid van de differentiaalvergelijking geevalueerd worden (ongeacht de gevraagde nauwkeurigheid), terwijl voor multistep n dit aantal ±100 bedroeg. Hierbij moet worden opgemerkt dat voor multistep n evenzovele malen de Jacobiaan moest worden geevalueerd en een lineair stelsel moest worden opgelost.

Appendix. Notatie en voorbeelden

Aangezien onze notatie niet algemeen gebruikelijk is zullen we hier, in een kort overzicht, door middel van een voorbeeld laten zien volgens welke regels de notatie is samengesteld.

f_N : een vector waarbij N de indexverzameling aangeeft.

$f_N[i]$: i-de element van f_N .

$G_{N \times K}$: een matrix waarvan N en K de indexverzamelingen zijn van resp. de eerste en de tweede index.

$G_{N \times K}[i,j]$: element uit de matrix $G_{N \times K}$.

$G_{N \times K}[i,]$: i-de rij-vector uit de matrix $G_{N \times K}$.

$G_{N \times K}[, j]$: j-de kolom-vector uit de matrix $G_{N \times K}$.

Wanneer een indexverzameling het continuüm $E \subset \mathbb{R}$ is, is in plaats van een vector of matrix sprake van een functie resp. een rij functies.

$y_{\Gamma \times E}[d]$: d-de element van de rij functies $y_{\Gamma \times E}$:

$$y_{\Gamma \times E}[d] \in E \rightarrow \mathbb{R}.$$

$y_{\Gamma \times E}[d](x_0)$: functiewaarde van $y_{\Gamma \times E}[d]$

$$y_{\Gamma \times E}[d](x_0) \in \mathbb{R}.$$

In sommige gevallen hebben we, om de formules overzichtelijk te houden, matrices en vectoren gedefinieerd waarbij de indexverzamelingen niet expliciet vermeld zijn.

Zo geldt

$$\begin{array}{lll} G \equiv G_{N \times K} & a \equiv a_K & B \equiv B_{\Gamma \times M} \\ H \equiv H_{N \times M} & c \equiv c_K & U \equiv U_{\Gamma \times K} \\ A \equiv A_{M \times K} & y_0 \equiv y_{0\Gamma} & b \equiv b_{\Gamma \times M} \\ y_h \equiv y_{h\Gamma} & \Delta y_h \equiv \Delta y_{h\Gamma} & u \equiv u_{\Gamma \times K} \end{array}$$

N.B. $u[,n+1] \equiv u_{\Gamma \times K+}[,n+1].$

$$y_{\Gamma \times E}[\gamma](x) = y^{(\gamma)}(x)$$

y_E, y_N, y^0 en y_h zijn korte notaties van speciale gevallen

$$\begin{aligned} y_E(x) &= y(x) &= y_{\Gamma \times E}[0](x) \\ y_N[i] &= y(x_i) &= y_{\Gamma \times E}[0](x_i) &= y_i \\ y^0_{\Gamma}[\gamma] &= y^{(\gamma)}(x_0) &= y_{\Gamma \times E}[\gamma](x_0) &= y^0[\gamma] \\ y_h_{\Gamma}[\gamma] &= y^{(\gamma)}(x_{n+1}) &= y_{\Gamma \times E}[\gamma](x_{n+1}) &= y_h[\gamma]. \end{aligned}$$

We laten nu, in het kort, nogmaals het verband zien tussen de in §2 - §8 gedefinieerde matrices en vectoren. We illustreren dit door enkele matrices volledig uit te schrijven.

De Newtonformule m.b.v. differentiequotienten waarvan we in §2 gebruik maken, is gebaseerd op de volgende differentietabel.

$\frac{S_N}{x_n}$	$\frac{f_N}{f_n}$					
		x				
x_{n-1}	f_{n-1}		x			
		x	x			
		x	x	x		$\frac{a_K}{a_n}$
x_2	f_2		x	a_{n-1}	a_{n+1}	
		x	x	x		
x_1	f_1		a_2	x		
		a_1	x			
x_0	$f_0 = a_0$		x			
		x				
x_{n+1}	f_{n+1}					

$$G_{E \times K}(x)[k] = \prod_{i=0}^{k-1} (x-x_i) \quad (\text{def. } G_{E \times K})$$

$$H_{E \times M}(x)[m] = (x-x_0)^m \quad (\text{def. } H_{E \times M})$$

$$G_{N \times K}[n,k] = \prod_{i=0}^{k-1} (x_n-x_i) = G[n,k] \quad (\text{def. } G)$$

$$H_{N \times M}[n,m] = (x_n-x_0)^m = H[n,m] \quad (\text{def. } H)$$

$$f_N = G * a = H * p$$

$$f_E = G_{E \times K} * a_K = H_{E \times M} * p_M = \quad (\text{def. } a)$$

$$= H_{E \times M} * H^{-1} * G * a = \quad (\text{def. } A)$$

$$= H_{E \times M} * A * a$$

$$\int \dots \int_{\gamma\text{-voudig}} f_E = \int \dots \int \{H_{E \times M} * A * a\} =$$

$$= \left\{ \int \dots \int H_{E \times M} \right\} * A * a \quad = (\text{def. } B)$$

$$= B * A * a + \text{const.} = (\text{def. } U)$$

$$= U * a + \text{const.}$$

$$y_N = G * c = H * q$$

$$y_E = G_{E \times K} * c_K = H_{E \times M} * q_M = \quad (\text{def. } c)$$

$$= H_{E \times M} * H^{-1} * G * c = \quad (\text{def. } A)$$

$$= H_{E \times M} * A * c$$

$$\begin{aligned}
\left(\frac{d}{dt}\right)^{\gamma} y_E &= \left(\frac{d}{dt}\right)^{\gamma} \{H_{E \times M} * A * c\} = \\
&= \left\{ \left(\frac{d}{dt}\right)^{\gamma} H_{E \times M} \right\} * A * c = && \text{(def. b)} \\
&= b * A * c = && \text{(def. u)} \\
&= u * c
\end{aligned}$$

Ter illustratie laten we zien hoe de matrices er uit zien voor $n = 3$.

$$G_{N \times K} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & (x_1 - x_0) & 0 & 0 \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & 0 \\ 1 & (x_3 - x_0) & (x_3 - x_0)(x_3 - x_1) & (x_3 - x_0)(x_3 - x_1)(x_3 - x_2) \end{pmatrix}$$

$$H_{N \times M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & (x_1 - x_0) & (x_1 - x_0)^2 & (x_1 - x_0)^3 \\ 1 & (x_2 - x_0) & (x_2 - x_0)^2 & (x_2 - x_0)^3 \\ 1 & (x_3 - x_0) & (x_3 - x_0)^2 & (x_3 - x_0)^3 \end{pmatrix}$$

Schrijven we $h_i = x_0 - x_i$, $i = 1(1)n$ en $h = x_{n+1} - x_0$.

$$A_{M \times K} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & h_1 & h_1 h_2 \\ 0 & 0 & 1 & h_1 + h_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B_{\Gamma \times M} = \begin{pmatrix} 1 & h & h^2 & h^3 \\ h & \frac{1}{2}h^2 & \frac{1}{3}h^3 & \frac{1}{4}h^4 \\ \frac{1}{2}h^2 & \frac{1}{6}h^3 & \frac{1}{12}h^4 & \frac{1}{20}h^5 \\ \frac{1}{6}h^3 & \frac{1}{24}h^4 & \frac{1}{60}h^5 & \frac{1}{120}h^6 \end{pmatrix}$$

$$U_{\Gamma \times K} = \begin{pmatrix} 1 & h & h(h+h_1) & h(h+h_1)(h+h_2) \\ h & \frac{1}{2}h^2 & h^2\left(\frac{1}{2}h_1 + \frac{1}{3}h\right) & h^2\left(\frac{1}{2}h_1h_2 + \frac{1}{3}hh_1 + \frac{1}{3}hh_2 + \frac{1}{4}h^2\right) \\ \frac{1}{2}h^2 & \frac{1}{6}h^3 & h^3\left(\frac{1}{6}h_1 + \frac{1}{12}h\right) & \dots \\ \frac{1}{6}h^3 & \frac{1}{24}h^4 & h^4\left(\frac{1}{24}h_1 + \frac{1}{60}h\right) & \dots \end{pmatrix}$$

$$b_{\Gamma \times M} = \begin{pmatrix} 1 & h & h^2 & h^3 \\ 0 & 1 & 2h & 3h^2 \\ 0 & 0 & 2 & 6h \\ 0 & 0 & 0 & 6 \end{pmatrix}$$

$$u_{\Gamma \times K} = \begin{pmatrix} 1 & h & h(h+h_1) & h(h+h_1)(h+h_2) \\ 0 & 1 & h_1+2h & h_1h_2+2hh_1+2hh_2+3h^2 \\ 0 & 0 & 2 & 2h_1+2h_2+6h \\ 0 & 0 & 0 & 6 \end{pmatrix}$$

Literatuur

Blanch G.

On modified divided differences.

Math. Comp. 8(1954) p. 1-11, p. 67-75.

Crane R.L. and Klopfenstein R.W.

A Predictor-Corrector Algorithm with an Increased Range of Absolute Stability.

J. ACM 12(1965) 227.

Curtiss C.F. and Hirschfelder J.O.

Integration of stiff equations.

Proc. Nat. Acad. Sci. U.S. 38(1952) 235.

Dahlquist G.

Fehlerabschätzungen bei Differenzenmethoden zur numerischen Integration gewöhnlicher Differentialgleichungen.

Z. Angew. Math. Mech. 31(1951) 239.

Convergence and Stability in numerical integration of ordinary differential equations.

Math. Scand. 4(1956) 33.

A special stability problem for linear multistep methods.

BIT 3(1963) 27.

A numerical method for some ordinary differential equations with large Lipschitz constants.

Proc. IFIP Congr. '68 p. 183.

Dekker T.J.

ALGOL 60 procedures in numerical algebra. part 1.

Mathematical Centre Tracts 22(1968).

Dill C. and Gear C.W.

A graphical search for stiffly stable methods for ordinary differential equations.

J. ACM 18(1971) 75.

Dixon M. and Webb E.

Enzymes.

Longmans Green and Co. (1965).

Fowler M.E. and Warten R.M.

A numerical integration technique for ordinary differential equations with widely separated eigenvalues.

IBM J. Res. Devel. (Sept. 1967).

Gear C.W.

The numerical integration of ordinary differential equations.
Math. Comp. 21(1967) 146.

The control of parameters in the automatic integration of ordinary differential equations.

Report c00-1469-007 Univ. Illinois (1968).

The automatic integration of stiff ordinary differential equations.

Proc. IFIP Congr. '68 p. 187.

Hamming R.W.

Numerical methods for scientists and engineers.

McGraw-Hill (1962).

Hemker H.C. and Hemker P.W.

General kinetics of enzyme cascades.

Proc. Roy. Soc. Ser. B 173(1969) 411.

Henrici P.

Discrete variable methods in ordinary differential equations.

John Wiley (1962).

Krogh F.T.

Predictor-Corrector methods of high order with improved stability characteristics.

J. ACM 13(1966) 374.

A test for instability in the numerical solutions of ordinary differential equations.

J. ACM 14(1967) 351.

A variable step variable order multistep method for the numerical solution of ordinary differential equations.
Proc. IFIP Congr. '68 p. 194.

Liniger W.

A criterion for A-stability of linear multistep integration formulae.

Computing 3(1968) 280.

Global accuracy and A-stability of one- and two-step integration formulae for stiff ordinary differential equations.
Conf. on the Num. Sol. of Diff. Eq. Dundee 1969 p. 188.

Liniger W. and Willoughby R.A.

Efficient integration methods for stiff systems of ordinary differential equations.

SIAM J. Numer. Anal. 7(1970) 47.

Meershoek H.Q.J.

PRECOR. Project 9000117.

T.H. Twente (1969).

Mitchell A.R. and Craggs J.W.

Stability of difference relations in the solution of ordinary differential equations.

Math. Comp. 7(1953) 127.

Nørsett S.P.

A criterion for $A(\alpha)$ -stability of linear multistep methods.
BIT 9(1969) 259.

An A-stable modification of the Adama-Bashforth methods.
Conf. on the Num. Sol. of Diff. Eq. Dundee 1969 p. 214.

Osborne M.R.

A method for finite difference approximations to ordinary differential equations.

Comp. J. 7(1964) 58.

Ottaway J.H. ed.

Computing techniques in biochemistry.

F.E.B.S.-Letters Vol. 2 Supplement (1969).

Vogelaere R. de

College over discretiseringsproblemen.

Univ. v. Amsterdam febr.-april 1970.

Widlund O.B.

A note on unconditionally stable linear multistep methods.

BIT 7(1967) 65.

Zonneveld J.A.

Automatic numerical integration.

Mathematical Centre Tracts 8(1964).