Numerieke methoden voor stelsels gewone differentiaalvergelijkingen

Prof. Dr. Marnix Van Daele

Deel II

Lineaire Meerstapsmethoden

Hoofdstuk 6

Predictor–Corrector-paren

6.1 Predictor-corrector-modes

Veronderstel dat we het standaard IVP oplossen m.b.v. een impliciete LMM. In dat geval moet in elke stap y_{n+k} bepaald worden uit het impliciete stelsel

$$y_{n+k} + \sum_{j=0}^{k-1} \alpha_j \, y_{n+j} = h \, \beta_k \, f(x_{n+k},\, y_{n+k}) + h \, \sum_{j=0}^{k-1} \beta_j \, f_{n+j} \, . \tag{6.1}$$

Daartoe maken we normaal gebruik van een iteratieve procedure

$$y_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} \alpha_j \, y_{n+j} = h \, \beta_k \, f(x_{n+k},\, y_{n+k}^{[\nu]}) + h \, \sum_{j=0}^{k-1} \beta_j \, f_{n+j} \, , \tag{6.2}$$

waarbij $y_{n+k}^{[0]}$ voor $n = 0,\, 1,\, \ldots$ willekeurig is. Dit schema convergeert naar de unieke oplossing zodra

$$h < \frac{1}{|\beta_k| \, L} \, ,$$

waarbij L de Lipschitz-constante is voor f t.o.v. y. Alhoewel (6.2) zal convergeren voor willekeurige $y_{n+k}^{[0]}$, kan het aantal iteraties sterk gereduceerd worden door een goeie initiële schatting $y_{n+k}^{[0]}$ te nemen. Deze schatting bekomt men in de praktijk door een afzonderlijke expliciete LMM te gebruiken. We geven die expliciete methode de naam *predictor* mee en de impliciete methode (6.2) noemen we de *corrector*. Samen vormen ze een *predictor–corrector paar*.

Het zal blijken voordelig te zijn dat de predictor en de corrector van dezelfde orde zijn, wat meestal betekent dat het stapgetal van de predictor groter zal zijn dan dit van de corrector.

Om niet steeds te moeten spreken van twee verschillende stapgetallen, wordt het stapgetal van de predictor, dat we k zullen noemen, het stapgetal van het paar genoemd en er wordt niet meer van de corrector geëist dat $|\alpha_0| + |\beta_0| \neq 0$ zou gelden. Zo bijvoorbeeld is

$$y_{n+2} - y_{n+1} = \frac{h}{2}\,(3\,f_{n+1} - f_n)\,, \qquad y_{n+2} - y_{n+1} = \frac{h}{2}\,(f_{n+2} + f_{n+1})$$

een predictor-corrector paar met stapgetal 2, alhoewel de corrector een één-stapsmethode is.

In de algemene notatie zullen we steeds een onderscheid maken tussen de predictor en corrector door *-en te hechten aan de coëfficiënten (en alle andere parameters, zoals orde en foutconstante) van de predictor.

Definitie 6.1.1 *Een k-staps PC-paar is van de vorm*

$$\begin{aligned} \mathrm{P}: &\quad \sum_{j=0}^{k} \alpha_j^*\, y_{n+j} = h \sum_{j=0}^{k-1} \beta_j^*\, f_{n+j} \\ \mathrm{C}: &\quad \sum_{j=0}^{k} \alpha_j\, y_{n+j} = h \sum_{j=0}^{k} \beta_j\, f_{n+j} \end{aligned} \tag{6.3}$$

waarbij $\alpha_k^* = \alpha_k = 1$ en $|\alpha_0^*| + |\beta_0^*| \neq 0$. □

Er zijn verschillende manieren of *modes* waarop het paar (6.3) kan geïmplementeerd worden.

- Vooreerst zouden we de predictor kunnen gebruiken voor de eerste schatting $y_{n+k}^{[0]}$ en nadien kunnen we de iteratie (6.2) uitvoeren tot we convergentie bereikt hebben. In de praktijk betekent dat een criterium zoals

  $$\|y_{n+k}^{[\nu+1]} - y_{n+k}^{[\nu]}\| < \epsilon$$

  voldaan is, waarbij ϵ de orde van de afrondingsfout is. Deze mode heet *correctie tot convergentie*. Hierin speelt de predictor een ondergeschikte rol en de karakteristieken van het PC-paar zijn deze van de corrector alleen. Het feit dat we niet op voorhand kunnen voorspellen hoeveel iteraties – en daardoor hoeveel functie-evaluaties – van de corrector noodzakelijk zullen zijn bij elke stap, maakt die mode onaantrekkelijk. Wanneer we een code ontwerpen gebaseerd op deze mode, leveren we iets af dat korte of lange tijd kan lopen. We hebben het niet in de hand. Dit is uiteraard te vermijden, zeker in het geval van *real-time* problemen.

- Een meer aanvaardbare procedure bestaat erin vooraf te bepalen hoeveel iteraties bij de corrector per stap toegelaten worden. Normaal houden we dit aantal klein, nl. 1 of 2. De karakteristieken van de PC-methode in deze mode hangen duidelijk af van zowel de predictor als de corrector; deze afhankelijkheid zullen we in de volgende paragrafen onderzoeken. Om deze modes te beschrijven zullen we gebruik maken van een kortschrifttaal, waarbij P en C respectievelijk de predictor en corrector voorstellen en E de evaluatie van de functie f aanduidt wanneer x en y gekend zijn. Omdat bij ODEs van grote dimensie de functie-evaluaties een groot deel van het rekenwerk uitmaken, is het zinvol het aantal functie-evaluaties per stap te beschouwen als een ruwe indicator van de rekeninspanning vereist door een methode. Veronderstel dat

we een predictor toepassen om $y_{n+k}^{[0]}$ te evalueren, vervolgens $f_{n+k}^{[0]} = f(x_{n+k},\, y_{n+k}^{[0]})$ berekenen en (6.2) eenmaal toepassen om $y_{n+k}^{[1]}$ te bekomen, dan kunnen we die mode noteren als PEC. Als we de iteratie een tweede maal oproepen om $y_{n+k}^{[2]}$ te bekomen, wat eerst de bepaling van $f_{n+k}^{[1]} = f(x_{n+k},\, y_{n+k}^{[1]})$ vereist, wordt de mode beschreven als PECEC of $\mathrm{P(EC)}^2$. Er is dan nog een verdere beslissing die moet genomen worden : bij het einde van de $\mathrm{P(EC)}^2$ stap beschikken we over de waarde $y_{n+k}^{[2]}$ voor y_{n+k} en de waarde $f_{n+k}^{[1]}$ voor $f(x_{n+k},\, y_{n+k})$. We kunnen verkiezen nog een betere waarde voor f te bepalen door de bijkomende evaluatie $f_{n+k}^{[2]} = f(x_{n+k},\, y_{n+k}^{[2]})$; deze mode zou dan beschreven worden door $\mathrm{P(EC)}^2\mathrm{E}$. Die twee klassen van modes $\mathrm{P(EC)}^\mu\mathrm{E}$ en $\mathrm{P(EC)}^\mu$ kunnen genoteerd worden als één enkele mode $\mathrm{P(EC)}^\mu\mathrm{E}^{1-t}$, waarbij μ een positief geheel getal is en $t = 0$ of 1.

Definitie 6.1.2 *De PC-mode* $\mathrm{P(EC)}^\mu\mathrm{E}^{1-t}$ *is het algoritme*

$$
\begin{aligned}
\mathrm{P}: \quad & y_{n+k}^{[0]} + \sum_{j=0}^{k-1} \alpha_j^*\, y_{n+j}^{[\mu]} = h \sum_{j=0}^{k-1} \beta_j^*\, f_{n+j}^{[\mu-t]} \\
(\mathrm{EC})^\mu: \quad & f_{n+k}^{[\nu]} = f(x_{n+k},\, y_{n+k}^{[\nu]})\,, \quad \nu = 0,\, 1,\, \ldots,\, \mu-1 \\
& y_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} \alpha_j\, y_{n+j}^{[\mu]} = h\,\beta_k\, f_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j\, f_{n+j}^{[\mu-t]} \\
\mathrm{E}^{1-t}: \quad & f_{n+k}^{[\mu]} = f(x_{n+k},\, y_{n+k}^{[\mu]}) \quad \textit{als } t = 0
\end{aligned}
$$

□

6.2 De LAF van PC-methoden

Als het PC-paar toegepast wordt in de "correctie tot convergentie"-mode is de LAF van het PC-paar deze van de corrector alleen. Als het paar echter toegepast wordt in de $\mathrm{P(EC)}^\mu\mathrm{E}^{1-t}$-mode met $t = 0$ of 1 wordt de LAF van de corrector beïnvloed door deze van de predictor. Dit wordt duidelijk in de volgende uiteenzetting.

We beschouwen een predictor en corrector gedefinieerd door (6.3) met resp. geassocieerde lineaire differentieoperatoren $\mathcal{L}^*$ en $\mathcal{L}$ van orde p^* en p en foutconstanten $C_{p^*+1}^*$ en C_{p+1}. Voor de berekening van de LAF maken we zoals steeds de veronderstelling dat de startwaarden exact zijn, m.a.w. $y_{n+j}^{[\nu]} = y(x_{n+j})$, $j = 0,\, 1,\, \ldots,\, k-1$ en we duiden de benaderingen van y in x_{n+k} aan als $\tilde{y}_{n+k}^{[\nu]}$. We veronderstellen tevens dat $y(x) \in C^{\hat{p}+1}$ met $\hat{p} = \max(p^*,\, p)$. Uit de bovenstaande veronderstellingen volgt

$$
\begin{aligned}
\mathcal{L}^*[y(x);\, h] &= C_{p^*+1}^*\, h^{p^*+1}\, y^{(p^*+1)}(x) + \mathcal{O}(h^{p^*+2}) \\
\mathcal{L}[y(x);\, h] &= C_{p+1}\, h^{p+1}\, y^{(p+1)}(x) + \mathcal{O}(h^{p+2})\,. \qquad (6.4)
\end{aligned}
$$

Daardoor geldt voor de predictor

$$
\sum_{j=0}^{k} \alpha_j^*\, y(x_{n+j}) = h \sum_{j=0}^{k-1} \beta_j^*\, f(x_{n+j},\, y(x_{n+j})) + \mathcal{L}^*[y(x_n);\, h]\,. \qquad (6.5)
$$

Uit Definitie 6.1.2 volgt ook

$$\tilde{y}_{n+k}^{[0]} + \sum_{j=0}^{k-1} \alpha_j^* \, y_{n+j}^{[\mu]} = h \sum_{j=0}^{k-1} \beta_j^* \, f(x_{n+j},\, y_{n+j}^{[\mu-t]}) \tag{6.6}$$

Door (6.6) lid aan lid af te trekken van (6.5) vinden we door de veronderstelling $y_{n+j} = y(x_{n+j})$, $j = 0,\, 1,\, \ldots,\, k-1$:

$$y(x_{n+k}) - \tilde{y}_{n+k}^{[0]} = C_{p^*+1}^* \, h^{p^*+1} \, y^{(p^*+1)}(x_n) + \mathcal{O}(h^{p^*+2}) \; . \tag{6.7}$$

Vermits het PC-paar toegepast wordt in de P(EC)$^\mu$E^{1-t}-mode zijn de corresponderende vergelijkingen voor de corrector :

$$\sum_{j=0}^{k} \alpha_j \, y(x_{n+j}) = h \sum_{j=0}^{k} \beta_j \, f(x_{n+j},\, y(x_{n+j})) + \mathcal{L}[y(x_n); h] \tag{6.8}$$

en voor $\nu = 0,\, 1,\, \ldots,\, \mu - 1$,

$$\tilde{y}_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} \alpha_j \, y_{n+j}^{[\mu]} = h \, \beta_k f(x_{n+k},\, \tilde{y}_{n+k}^{[\nu]}) + h \sum_{j=0}^{k-1} \beta_j \, f(x_{n+j},\, y_{n+j}^{[\mu-t]}) \, . \tag{6.9}$$

Door deze vergelijkingen lid aan lid van elkaar af te trekken, bekomen we :

$$\begin{aligned} y(x_{n+k}) - \tilde{y}_{n+k}^{[\nu+1]} &= h \, \beta_k \, [f(x_{n+k}, y(x_{n+k})) - f(x_{n+k},\, \tilde{y}_{n+k}^{[\nu]})] + \mathcal{L}[y(x_n); h] \\ &= h \, \beta_k \, \bar{J}(x_{n+k},\, \eta_\nu) \, [y(x_{n+k}) - \tilde{y}_{n+k}^{[\nu]}] + C_{p+1} \, h^{p+1} \, y^{(p+1)}(x_n) \\ &\quad + \mathcal{O}(h^{p+2}), \quad \nu = 0,\, 1,\, \ldots,\, \mu - 1 \; , \end{aligned} \tag{6.10}$$

waarbij $J = \frac{\partial f}{\partial y}$. Wat nu volgt hangt af van de relatieve groottes van p^* en p.

- We beschouwen eerst het geval $p^* \geq p$. Bij substitutie van (6.7) in (6.10) met $\nu = 0$ ontstaat

  $$y(x_{n+k}) - \tilde{y}_{n+k}^{[1]} = C_{p+1} \, h^{p+1} \, y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}) \; .$$

  Deze uitdrukking voor $y(x_{n+k}) - \tilde{y}_{n+k}^{[1]}$ kan nu gesubstitueerd worden in (6.10) met $\nu = 1$, wat resulteert in

  $$y(x_{n+k}) - \tilde{y}_{n+k}^{[2]} = C_{p+1} \, h^{p+1} \, y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}) \; .$$

  Zo verdergaand bekomen we finaal

  $$y(x_{n+k}) - \tilde{y}_{n+k}^{[\mu]} = C_{p+1} \, h^{p+1} \, y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}) \; .$$

  We vinden dat voor $p^* \geq p$ de PLAF van de P(EC)$^\mu$E^{1-t}-mode voor alle $\mu \geq 1$ precies degene is van de corrector alleen.

- Veronderstel dat $p^* = p - 1$. Bij substitutie van (6.7) in (6.10) met $\nu = 0$ verkrijgen we

$$y(x_{n+k}) - \tilde{y}^{[1]}_{n+k} = \left[\beta_k \, \bar{J} \, C^*_p \, y^{(p)}(x_n) + C_{p+1} \, y^{(p+1)}(x_n)\right] \, h^{p+1} + \mathcal{O}(h^{p+2}) \, .$$

  Als $\mu = 1$, d.w.z. als de mode PECE^{1-t} is, is de PLAF van het PC-paar niet identiek aan deze van de corrector, maar de orde van het PC-paar valt wel samen met die van de corrector. Voor $\mu \geq 2$ vinden we wel opnieuw door substitutie in (6.10)

$$y(x_{n+k}) - \tilde{y}^{[\mu]}_{n+k} = C_{p+1} \, h^{p+1} \, y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}) \, ,$$

  en de PLAF van de PC-mode wordt deze van de corrector alleen.

- Beschouwen we nu het geval $p^* = p - 2$. Bij substitutie van (6.7) in (6.10) met $\nu = 0$ bekomen we

$$y(x_{n+k}) - \tilde{y}^{[1]}_{n+k} = \beta_k \, \bar{J} \, C^*_{p-1} \, h^p \, y^{(p-1)}(x_n) + \mathcal{O}(h^{p+1}) \, , \tag{6.11}$$

  De orde van zo'n PC-paar voor $\mu = 1$ is dus slechts $p - 1$. Na substitutie van (6.11) in (6.10) met $\nu = 1$ verkrijgen we

$$y(x_{n+k}) - \tilde{y}^{[2]}_{n+k} = \left[\left(\beta_k \, \bar{J}\right)^2 C^*_{p-1} \, y^{(p-1)}(x_n) + C_{p+1} \, y^{(p+1)}(x_n)\right] \, h^{p+1} + \mathcal{O}(h^{p+2}) \, ,$$

  waaruit volgt dat voor $\mu = 2$ de orde van het PC-paar deze van de corrector is, maar de twee PLAF-waarden zijn niet dezelfde. Voor $\mu \geq 3$ vinden we dan weer

$$y(x_{n+k}) - \tilde{y}^{[\mu]}_{n+k} = C_{p+1} \, h^{p+1} \, y^{(p+1)}(x_n) + \mathcal{O}(h^{p+2}) \, ,$$

  en de PLAF is opnieuw deze van de corrector alleen.

Het moge duidelijk zijn dat de orde en de PLAF van een PC-paar afhangt (i) van het verschil tussen p^* en p en (ii) van μ, het aantal keer dat de corrector wordt opgeroepen. In het bijzonder:

(i) als $p^* \geq p$ (of als $p^* < p$ en $\mu > p - p^*$) hebben het PC-paar en de corrector dezelfde orde en dezelfde PLAF,

(ii) als $p^* < p$ en $\mu = p - p^*$ hebben het PC-paar en de corrector dezelfde orde maar verschillende PLAF-waarden,

(iii) als $p^* < p$ en $\mu \leq p - p^* - 1$ is de orde van de het PC-paar $p^* + \mu (< p)$.

Merk wel op dat de modes $\text{P(EC)}^\mu\text{E}$ en P(EC)^μ altijd dezelfde orde en PLAF hebben.

6.3 De Milne-schatting en lokale extrapolatie

PC-methoden bieden het voordeel t.o.v. LMMn dat het mogelijk is een schatting van de PLAF van de PC-methode te maken zonder een rechtstreekse schatting van $y^{(p+1)}(x_n)$ in te voeren en dit dank zij een origineel idee afkomstig van Milne. Dit idee werkt enkel als $p^* = p$. In dat geval is

$$C^*_{p+1}\, h^{p+1}\, y^{(p+1)}(x_n) = y(x_{n+k}) - \tilde{y}^{[0]}_{n+k} + \mathcal{O}(h^{p+2})$$

en

$$C_{p+1}\, h^{p+1}\, y^{(p+1)}(x_n) = y(x_{n+k}) - \tilde{y}^{[\mu]}_{n+k} + \mathcal{O}(h^{p+2})\,.$$

Bij aftrekking bekomen we

$$(C^*_{p+1} - C_{p+1})\, h^{p+1}\, y^{(p+1)}(x_n) = \tilde{y}^{[\mu]}_{n+k} - \tilde{y}^{[0]}_{n+k} + \mathcal{O}(h^{p+2})\,,$$

waaruit

$$C_{p+1}\, h^{p+1}\, y^{(p+1)}(x_n) = \frac{C_{p+1}}{C^*_{p+1} - C_{p+1}} \left(\tilde{y}^{[\mu]}_{n+k} - \tilde{y}^{[0]}_{n+k}\right) + \mathcal{O}(h^{p+2})\,.$$

Zo bekomen we de *Milne-schatting voor de PLAF*

$$PLAF = C_{p+1}\, h^{p+1}\, y^{(p+1)}(x_n) \approx \frac{C_{p+1}}{C^*_{p+1} - C_{p+1}} \left(y^{[\mu]}_{n+k} - y^{[0]}_{n+k}\right) \tag{6.12}$$

Merk op dat in de rechterzijde van (6.12) $\tilde{y}^{[\mu]}_{n+k}$ vervangen is door $y^{[\mu]}_{n+k}$. Men kan er immers van uitgaan dat de PLAF een aanvaardbare maat is voor de lokale nauwkeurigheid, ook als de startwaarden niet exact zijn.

De schatting (6.12) wordt hoofdzakelijk gebruikt om de staplengte h te controleren. Deze zou kunnen verkleind worden als de norm van de foutschatting een gegeven tolerantie zou overschrijden en vergroot worden als die norm kleiner is dan die tolerantie. Men is echter geneigd, zoals het geval is met alle foutschattingen, om die foutschatting toe te voegen aan de numerieke oplossing, waardoor de nauwkeurigheid opgedreven kan worden. Die toevoeging wordt *lokale extrapolatie* genoemd. Het uitvoeren van lokale extrapolatie komt overeen met het verhogen van de orde van de methode met een eenheid. Het is zeer gebruikelijk geworden in moderne codes om lokale extrapolatie uit te voeren bij elke stap, maar tevens gebruik te maken van (6.12) om te staplengte aan te passen. Men kiest de stap zo dat de foutschatting (6.12) aanvaardbaar is voor $y^{[\mu]}_{n+k}$, waardoor men aanneemt dat die staplengte voor de nauwkeuriger lokaal geëxtrapoleerde waarde ook aanvaardbaar blijft. Er zijn hier wel enkele gevaren aan verbonden, maar de techniek is in de moderne codes aanvaard.

Lokale extrapolatie kan op verschillende wijzen toegepast worden. Enerzijds kunnen we de lokale extrapolatie uitvoeren na elke oproep van de corrector. Als we dan de letter L gebruiken om lokale extrapolatie aan te duiden, kunnen de resulterende modes genoteerd worden $\mathrm{P(ECL)}^{\mu}\mathrm{E}^{1-t}$, $t = 0$ of 1. Anderzijds kunnen we verkiezen lokale extrapolatie slechts toe te passen na de finale inwerking van de corrector, wat resulteert in de modes $\mathrm{P(EC)}^{\mu}\mathrm{LE}^{1-t}$,

$t = 0$ of 1. De twee families van modes vallen samen als $\mu = 1$. Merk tevens op dat uit (6.12) volgt dat lokale extrapolatie equivalent is met het vervangen van $y_n^{[\nu]}$ door

$$y_n^{[\nu]} + \frac{C_{p+1}}{C_{p+1}^* - C_{p+1}} (y_n^{[\nu]} - y_n^{[0]}) = \frac{C_{p+1}^*}{C_{p+1}^* - C_{p+1}} y_n^{[\nu]} - \frac{C_{p+1}}{C_{p+1}^* - C_{p+1}} y_n^{[0]} .$$

Zo komen we tot de volgende formele definities :

Definitie 6.3.1 *De PC-mode* $\mathrm{P(ECL)^{\mu}E^{1-t}}$ *is het algoritme*

$$\begin{aligned}
\mathrm{P}: \quad & y_{n+k}^{[0]} + \sum_{j=0}^{k-1} \alpha_j^* \, y_{n+j}^{[\mu]} = h \sum_{j=0}^{k-1} \beta_j^* \, f_{n+j}^{[\mu-t]} \\
(\mathrm{ECL})^{\mu}: \quad & f_{n+k}^{[\nu]} = f(x_{n+k}, \, y_{n+k}^{[\nu]}) \quad \textit{voor} \quad \nu = 0, \, 1, \, \ldots, \, \mu - 1 \\
& \hat{y}_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} \alpha_j \, y_{n+j}^{[\mu]} = h \, \beta_k \, f_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j \, f_{n+j}^{[\mu-t]} \\
& y_{n+k}^{[\nu+1]} = \frac{C_{p+1}^*}{C_{p+1}^* - C_{p+1}} \hat{y}_{n+k}^{[\nu+1]} - \frac{C_{p+1}}{C_{p+1}^* - C_{p+1}} y_{n+k}^{[0]} \\
\mathrm{E}^{1-t}: \quad & f_{n+k}^{[\mu]} = f(x_{n+k}, \, y_{n+k}^{[\mu]}) \; \textit{als} \; t = 0 \, .
\end{aligned}$$

□

Definitie 6.3.2 *De PC-mode* $\mathrm{P(EC)^{\mu}LE}$ *is het algoritme*

$$\begin{aligned}
\mathrm{P}: \quad & \hat{y}_{n+k}^{[0]} + \sum_{j=0}^{k-1} \alpha_j^* \, y_{n+j}^{[\mu]} = h \sum_{j=0}^{k-1} \beta_j^* \, f_{n+j}^{[\mu]} \\
(\mathrm{EC})^{\mu}: \quad & \hat{f}_{n+k}^{[\nu]} = f(x_{n+k}, \hat{y}_{n+k}^{[\nu]}) \quad \textit{voor} \quad \nu = 0, \, 1, \, \ldots, \, \mu - 1 \\
& \hat{y}_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} \alpha_j \, y_{n+j}^{[\mu]} = h \, \beta_k \, \hat{f}_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j \, f_{n+j}^{[\mu]} \\
\mathrm{L}: \quad & y_{n+k}^{[\mu]} = \frac{C_{p+1}^*}{C_{p+1}^* - C_{p+1}} \hat{y}_{n+k}^{[\mu]} - \frac{C_{p+1}}{C_{p+1}^* - C_{p+1}} \hat{y}_{n+k}^{[0]} \\
\mathrm{E}: \quad & f_{n+k}^{[\mu]} = f(x_{n+k}, \, y_{n+k}^{[\mu]})
\end{aligned}$$

□

Definitie 6.3.3 *De PC-mode* $\mathrm{P(EC)^{\mu}L}$ *is het algoritme*

$$\begin{aligned}
\mathrm{P}: \quad & \hat{y}_{n+k}^{[0]} + \sum_{j=0}^{k-1} \alpha_j^* \, y_{n+j}^{[\mu]} = h \sum_{j=0}^{k-1} \beta_j^* \, \hat{f}_{n+j}^{[\mu-1]} \\
(\mathrm{EC})^{\mu}: \quad & \hat{f}_{n+k}^{[\nu]} = f(x_{n+k}, \hat{y}_{n+k}^{[\nu]}) \; \textit{voor} \, \nu = 0, \, 1, \, \ldots, \, \mu - 1 \\
& \hat{y}_{n+k}^{[\nu+1]} + \sum_{j=0}^{k-1} \alpha_j \, y_{n+j}^{[\mu]} = h \, \beta_k \, \hat{f}_{n+k}^{[\nu]} + h \sum_{j=0}^{k-1} \beta_j \, \hat{f}_{n+j}^{[\mu-1]} \\
\mathrm{L}: \quad & y_{n+k}^{[\mu]} = \frac{C_{p+1}^*}{C_{p+1}^* - C_{p+1}} \hat{y}_{n+k}^{[\mu]} - \frac{C_{p+1}}{C_{p+1}^* - C_{p+1}} \hat{y}_{n+k}^{[0]}
\end{aligned}$$

□

Opmerking 6.3.1

Vele PC-methoden die opgenomen zijn in programmabibliotheken of die op het Internet terug te vinden zijn, zijn opgebouwd m.b.v. Adams–Bashforth-methoden als predictors en Adams–Moulton-methoden als correctors. Zulke PC-paren (waarbij P en C dezelfde orde hebben) worden aangeduid als *ABM*-methoden (ABMMn). □

6.4 De lineaire stabiliteit van PC-paren

Het is evident dat de stabiliteit van een PC-paar afhangt van de mode waarin het paar wordt toegepast. Zonder veel dieper in te gaan op de theorie, vermelden we de volgende resultaten voor PC-paren waarbij zowel predictor als corrector de orde k hebben :

$$\pi_{\mathrm{P(EC)}^{\mu}}(r,\,\hat{h}) = \beta_k\, r^k \left(\rho(r) - \hat{h}\,\sigma(r)\right) + \frac{(\beta_k\,\hat{h})^{\mu}\,(1-\beta_k\,\hat{h})}{1-(\beta_k\,\hat{h})^{\mu}}\,(\rho^*(r)\,\sigma(r) - \sigma^*(r)\,\rho(r)) \tag{6.13}$$

$$\pi_{\mathrm{P(EC)}^{\mu}\mathrm{E}}(r,\,\hat{h}) = \rho(r) - \hat{h}\,\sigma(r) + \frac{(\beta_k\,\hat{h})^{\mu}\,(1-\beta_k\,\hat{h})}{1-(\beta_k\,\hat{h})^{\mu}}\,(\rho^*(r) - \hat{h}\,\sigma^*(r)) \tag{6.14}$$

Over het algemeen is $\pi_{\mathrm{P(EC)}^{\mu}\mathrm{E}}(r,\,\hat{h})$ van graad k, terwijl $\pi_{\mathrm{P(EC)}^{\mu}}(r,\,\hat{h})$ van graad $2\,k$ is. Om stabiliteit te bereiken moeten in $\mathrm{P(EC)}^{\mu}$-mode dubbel zoveel wortels onder controle gehouden worden als in $\mathrm{P(EC)}^{\mu}$E-mode, hetgeen suggereert dat $\mathrm{P(EC)}^{\mu}$E-mode gepaard gaat met grotere gebieden van absolute stabiliteit.

Voorbeeld 6.4.1

Beschouw de ABMM waarbij predictor en corrector van tweede orde zijn. Passen we deze methode toe op $y' = \lambda\, y$ in PECE-mode, dan geldt

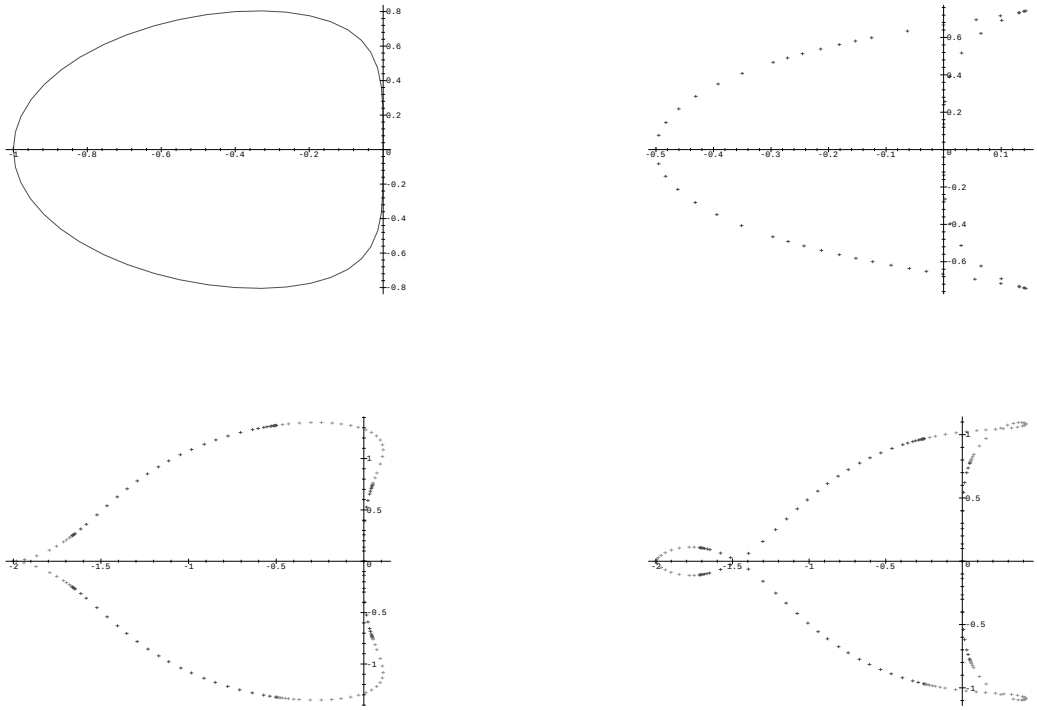
$$\begin{aligned}
\mathrm{P} \;:\;& y_{n+1}^{[0]} - y_n^{[1]} = \frac{\hat{h}}{2}\left(3\,y_n^{[1]} - y_{n-1}^{[1]}\right), \\
\mathrm{C} \;:\;& y_{n+1}^{[1]} - y_n^{[1]} = \frac{\hat{h}}{2}\left(y_{n+1}^{[0]} + y_n^{[1]}\right),
\end{aligned}$$

zodat

$$\begin{aligned}
y_{n+1}^{[1]} &= y_n^{[1]} + \frac{\hat{h}}{2}\left(y_n^{[1]} + \frac{\hat{h}}{2}\left(3\,y_n^{[1]} - y_{n-1}^{[1]}\right) + y_n^{[1]}\right) \\
&= \left(1 + \hat{h} + \frac{3}{4}\,\hat{h}^2\right) y_n^{[1]} - \frac{1}{4}\,\hat{h}^2\, y_{n-1}^{[1]}
\end{aligned}$$

waaruit volgt

$$\pi_{\mathrm{PECE}}(r,\,\hat{h}) = r^2 - \left(1 + \hat{h} + \frac{3}{4}\,\hat{h}^2\right) r + \frac{1}{4}\,\hat{h}^2\,.$$



Figuur 6.1: $\partial\mathcal{R}_A$ voor de ABMM van tweede orde in PE, PEC, PECE en PECEC mode.

Beschouw nu daarentegen hetzelfde paar in PEC-mode. Dan is

$$\begin{aligned} \mathrm{P} \;:\; & y_{n+1}^{[0]} - y_n^{[1]} = \tfrac{\hat{h}}{2}\left(3\,y_n^{[0]} - y_{n-1}^{[0]}\right), \\ \mathrm{C} \;:\; & y_{n+1}^{[1]} - y_n^{[1]} = \tfrac{\hat{h}}{2}\left(y_{n+1}^{[0]} + y_n^{[0]}\right), \end{aligned}$$

Om de $y_n^{[1]}$ te bepalen moeten de $y_n^{[0]}$ hieruit geëlimineerd worden. Dit kan door de predictor te herschrijven met $n \to n+1$ en de corrector met $n \to n+1$ en met $n \to n-1$. Op deze manier beschikken we over 5 lineaire vergelijkingen in de vier onbekenden $y_{n+p}^{[0]}$, $p = -1,\ 0,\ 1,\ 2$. Dit stelsel,

$$\begin{bmatrix} \frac{1}{2}\hat{h} & -\frac{3}{2}\hat{h} & 1 & 0 \\ 0 & \frac{1}{2}\hat{h} & -\frac{3}{2}\hat{h} & 1 \\ -\frac{1}{2}\hat{h} & -\frac{1}{2}\hat{h} & 0 & 0 \\ 0 & -\frac{1}{2}\hat{h} & -\frac{1}{2}\hat{h} & 0 \\ 0 & 0 & -\frac{1}{2}\hat{h} & -\frac{1}{2}\hat{h} \end{bmatrix} \begin{bmatrix} y_{n-1}^{[0]} \\ y_n^{[0]} \\ y_{n+1}^{[0]} \\ y_{n+2}^{[0]} \end{bmatrix} = \begin{bmatrix} y_n^{[1]} \\ y_{n+1}^{[1]} \\ y_{n-1}^{[1]} - y_n^{[1]} \\ y_n^{[1]} - y_{n+1}^{[1]} \\ y_{n+1}^{[1]} - y_{n+2}^{[1]} \end{bmatrix},$$

heeft een oplossing a.s.a. de determinant van de verhoogde matrix 0 is. Dit leidt, als $y_{n+i}^{[1]} \to r^i$, tot

$$\pi_{\mathrm{PEC}}(r,\,\hat{h}) = r\,\left((-4\,r^2 + 3\,r - 1)\,\hat{h} - 2\,r^2 + 2\,r^3\right).$$

De gevonden veeltermen kunnen ook afgeleid worden uit (6.13) en (6.14) door voor de predictor $\rho^*(r) = r^2 - r$, $\sigma^*(r) = \frac{1}{2}(3\,r - 1)$ te stellen en voor de corrector $\rho(r) = r^2 - r$ en $\sigma(r) = \frac{1}{2}(r + r^2)$.
Tenslotte tonen we in Figuur 6.1 de randen van de stabiliteitsgebieden die corresponderen met de verschillende modes. Voor de P(EC)^2-mode moeten de kleine begrensde gebieden (bij $x \approx -1.5$ en bij $x \approx 0.2$) uitgesloten worden. Men stelt vast dat de stabiliteitsgebieden (i) groeien naarmate de μ toeneemt en (ii) in P(EC)$^\mu$E-mode groter zijn dan in P(EC)$^\mu$-mode. □

6.5 VSVO-algoritmen

Een code gebaseerd op PC-paren is vaak van het VSVO-type. In Hoofdstuk 3 hebben we de verschillende elementen opgesomd die aan bod komen in zo'n VSVO-algoritme:

(i) een familie van methoden

(ii) een startprocedure

(iii) een lokalefout-schatter

(iv) een strategie om te beslissen wanneer staplengte en/of orde moeten veranderd worden

(v) een techniek voor de verandering van staplengte en/of orde.

Merk op dat we in deze lijst niet verwezen hebben naar lineaire stabiliteit. Algoritmen testen normaal niet of de voorwaarden van absolute stabiliteit voldaan zijn voor een gegeven staplengte, maar steunen op het feit dat de foutschatting sterk zal toenemen als aan die voorwaarde niet zou voldaan zijn en het algoritme zal dan automatisch een aangepaste actie ondernemen. Een andere optie die niet werd vermeld is het testen op *stiffness*. Tal van codes voorzien thans in een dergelijke test. Gaat men van een stijf regime over naar een niet-stijf regime of omgekeerd, dan wordt voor een andere familie basismethoden gekozen.

Familie van methoden: Voor niet-stijve problemen opteert bijna altijd voor de familie van Adams–Bashforth–Moulton-methoden (ABMMn) met ordes variërend van 1 tot rond de 13. In sommige algoritmen worden de ABMMn van laagste orde vervangen door PCMn met betere gebieden van absolute stabiliteit, omdat de begrenzing van de staplengte bij lagere orde meestal het gebrek aan stabiliteit is, eerder dan de nauwkeurigheid. Een variëteit aan modes kan gebruikt worden, maar PECE met lokale extrapolatie is de populairste. Voor stijve problemen worden veelal BDF-methoden ingeschakeld.

Startprocedures: Meestal is dit de eenvoud zelf. Het algoritme start altijd met het één-staps ABM-paar (of zijn alternatief) waarvoor geen bijkomende startwaarden vereist zijn en laat nadien toe dat de staplengte/ordeverandering strategie dan overneemt. Dit geeft meestal aanleiding tot snelle ordeverhoging in de eerste stappen.

Lokalefout-schatter: Dit wordt doorgaans geleverd door een vorm van de Milne-schatting die een eenvoudige en efficiënte vorm aanneemt voor ABMMn.

Strategie: Een mogelijke strategie is de volgende. Veronderstel dat het algoritme op een bepaald ogenblik werkt met een k-de orde methode; als E_k de norm is van de lokalefout-schatting in x_{n+1} en als τ de nauwkeurigheidstolerentie is geëist door de gebruiker, dan is een voor de hand liggend criterium om de stap van x_n naar x_{n+1} te aanvaarden dat

$$E_k \leq \tau \;, \tag{6.15}$$

d.i. het zgn. *fout per stap* criterium. Voor een k-de orde ABM-paar bijvoorbeeld is de norm van de foutschatting gemakkelijk berekenbaar wanneer de methode geïmplementeerd is via eindige differenties.

Op het eerste zicht staan we voor een onmogelijke taak; we hebben slechts één criterium om twee stukken informatie af te leiden, nl. wat de staplengte *en* de orde moeten zijn bij de volgende stap. Nochtans zijn er vergelijkbare situaties in de elementaire calculus waar we vragen voor één stuk informatie en zonder meer er twee krijgen! Wanneer we bijvoorbeeld vragen wat het maximum van een functie is, krijgen we samen met het antwoord ook het punt waar dit maximum bereikt wordt. Hetzelfde mechanisme komt ons hier ook redding brengen.

Veronderstel dat bij het beëindigen van de stap van x_n naar x_{n+1} het criterium (6.15) voldaan is en de berekende waarde voor $y_{n+1}^{[\mu]}$ wordt aanvaard, zodat we klaar zijn voor een volgende stap. Vooraleer hiermee te beginnen, vragen we ons af welke maximale staplengte we zouden hebben kunnen gebruiken bij de juist uitgevoerde stap van x_n naar x_{n+1} door ABMMn te gebruiken van orden $k-1$, k en $k+1$. De grootste onder die drie staplengten wordt de staplengte die we zullen gebruiken voor de nieuwe stap en de waarde voor k waarmee die maximum staplengte wordt gebruikt zal de orde zijn die we zullen gebruiken voor de volgende stap.

Als de staplengte gebruikt met de k-de orde methode voor de stap van x_n naar x_{n+1} h_k is, en vermits die stap succesvol was, moet het een foutschatting produceren die voldoet aan (6.15) :

$$\begin{aligned} E_k &= \beta\,\tau, \quad 0 \le \beta \le 1 \\ &\approx K\,h_k^{k+1}\,, \end{aligned} \tag{6.16}$$

waarbij K een constante is. De maximum staplengte $\bar{h}_k$ die we hadden kunnen nemen met deze k-de orde methode zou een foutschatting $\bar{E}_k$ hebben geproduceerd die voldoet aan:

$$\bar{E}_k = \tau \approx K\,\bar{h}_k^{k+1}\,. \tag{6.17}$$

Uit (6.16) en (6.17) volgt dat

$$\beta \approx \left(\frac{h_k}{\bar{h}_k}\right)^{k+1}\,,$$

waaruit

$$\bar{h}_k \approx h_k \left(\frac{\tau}{E_k}\right)^{1/(k+1)}\,, \tag{6.18}$$

wat te berekenen valt. Veronderstel dat we de stap van x_n naar x_{n+1} hadden gemaakt met ABMMn van ordes $k-1$ en $k+1$ en respectieve staplengten h_{k-1} en h_{k+1}. Dan kan in een ABM-formalisme E_{k+i}, $i = -1,\, 0,\, 1$ vrij eenvoudig berekend worden. De argumentatie voor de berekening van (6.18) kan hier herhaald worden. Zo vinden we dat

$$\bar{h}_{k+i} \approx h_{k+i} \left(\frac{\tau}{E_{k+i}}\right)^{1/(k+1+i)}\,, \quad i = -1,\, 0,\, 1\,.$$

De orde die we dan gebruiken voor de volgende stap is dan $k + \bar{i}$, waarbij

$$\bar{h}_{k+\bar{i}} = \max_{i \in \{-1,0,1\}} \bar{h}_{k+i} \ ,$$

en de maximum stap die we kunnen gebruiken in de volgende stap is $\bar{h}_{k+\bar{i}}$.

In dit domein van implementaties is er groot stuk heuristiek. De meeste algoritmen zullen de maximum staplengte die volgt uit de bovenstaande redenering, met een heuristisch gekozen factor, kleiner dan maar dicht bij één, vermenigvuldigen. Soms plaatst men in algoritmen ook instructies die vermijden dat de staplengte te dikwijls wordt veranderd.

De eerste twee VSVO-implementaties werden praktisch gelijktijdig op dezelfde conferentie gepresenteerd in 1968 door C.W. Gear (DIFSUB) en F.T. Krogh (DVDQ). DVDQ liet enkel stapverdubbeling en staphalvering toe. De DIFSUB-code liet willekeurige h-veranderingen toe door gebruik te maken van de Nordsieck-vector. GEAR is een aangepaste en uitgebreide versie van DIFSUB. Een veel gebruikte code, EPISODE en de bijgewerkte code VODE (van G.D. Byrne en A.C. Hindmarsh) maken gebruik van de veranderlijkecoëfficiënt-techniek, zoals ook de code DE/STEP (van L.F. Shampine en M.K. Gordon)