Numerieke simulatie van differentiaalvergelijkingen

Giovanni Samaey

Les 3

Runge-Kutta methodes (H3)
Foutenschatting – ingebedde RK methodes (H6.3)

Recap: De voorwaartse Eulermethode

- Beginwaardeprobleem:

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad t \geq t_0, \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0$$

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

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

- Voorwaartse Euler berekent die benadering als

$$\boxed{\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h \boldsymbol{f}(t_n, \boldsymbol{y}_n),} \qquad n = 0, 1, \ldots$$

- Vaststelling:
  - $\boldsymbol{y}_{n-1}$ wordt weggegooid !
  - Methode is van orde 1: $\boldsymbol{y}_n = \boldsymbol{y}(t_n) + \mathcal{O}(h)$

Recap: Lineaire multistapmethodes

- Beginwaardeprobleem:

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad t \geq t_0, \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0$$

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

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

- Lineaire multistapmethode berekent benadering als

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

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

Alternatief idee: Runge-Kutta methodes

- Beginwaardeprobleem:

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad t \geq t_0, \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0$$

- Exacte oplossing:

$$\begin{aligned} \boldsymbol{y}(t_{n+s}) &= \boldsymbol{y}(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{y}'(\tau) d\tau \\ &= \boldsymbol{y}(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau)) d\tau \end{aligned}$$

- Bedoeling: hogere orde door goede kwadratuurregels

$$\int_a^b f(\tau)\omega(\tau) d\tau = \sum_{j=1}^{\nu} b_j f(c_j)$$

c_j knopen

b_j gewichten

Intermezzo: kwadratuurregels

- Kwadratuurregel is van orde p als ze exact is voor veeltermen van graad $p-1$

- Benaderingsfout

$$\left| \int_a^b f(\tau)\omega(\tau)d\tau - \sum_{j=1}^{\nu} b_j f(c_j) \right| \leq c \max_{a \leq t \leq b} \left| f^{(p)}(t) \right|$$

- Stelling 1: Voor elke set niet-samenvallende knopen $\{c_j\}_{j=1}^{\nu}$ zijn er gewichten $\{b_j\}_{j=1}^{\nu}$ te vinden zodat $p = \nu$.

- Stelling 2: Met een optimale keuze van de knopen $\{c_j\}_{j=1}^{\nu}$ en gewichten $\{b_j\}_{j=1}^{\nu}$ kan een orde $p = 2\nu$ bekomen worden.

Keuzes van kwadratuurregels

- Newton-Cotesregels: Equidistante knopen
- Orthogonale veeltermen (zie 'Numerieke modellering en benadering')
  - Definitie van scalair product $\langle f, g\rangle_\omega := \int_a^b f(\tau)g(\tau)\omega(\tau)d\tau$
  - Orthogonale veeltermen

$$\langle p_m, \hat{p}\rangle = 0, \qquad \text{for every } \hat{p} \in \mathbb{P}_{m-1}$$

  - Als $\omega(\tau) = 1$, dan heten die veeltermen Legendreveeltermen
- Gauss-kwadratuur (heeft op "magische" wijze orde $p = 2\nu$)
  - Kies als knopen $\{c_j\}_{j=1}^{\nu}$ de nulpunten van p_ν (dit kan altijd !)
  - Kies gewichten $\{b_j\}_{j=1}^{\nu}$ om orde $p = \nu$ te bekomen

Een veralgemeend resultaat

- Beschouw een veelterm r van graad ν zodat

$$\langle r, \hat{p} \rangle = 0, \qquad \text{for every } \hat{p} \in \mathbb{P}_{m-1}$$

- Beschouw volgende kwadratuurregel
  - Kies als knopen $\{c_j\}_{j=1}^{\nu}$ de nulpunten van r (dit kan altijd !)
  - Kies gewichten $\{b_j\}_{j=1}^{\nu}$ om orde $p = \nu$ te bekomen

- De resulterende kwadratuurregel zal orde $p = \nu + m$ hebben.

Alternatief idee: Runge-Kutta methodes

- Beginwaardeprobleem:

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad t \geq t_0, \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0$$

- Exacte oplossing:

$$\begin{aligned} \boldsymbol{y}(t_{n+s}) &= \boldsymbol{y}(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{y}'(\tau) d\tau \\ &= \boldsymbol{y}(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau)) d\tau \end{aligned}$$

- Bedoeling: hogere orde door goede kwadratuurregels

$$\int_a^b f(\tau)\omega(\tau) d\tau = \sum_{j=1}^{\nu} b_j f(c_j)$$

c_j knopen

b_j gewichten

Expliciete Runge-Kutta methodes (1)

- Benader exacte oplossing via kwadratuurregel

$$\begin{aligned}
\boldsymbol{y}(t_{n+1}) &= \boldsymbol{y}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))d\tau \\
&= \boldsymbol{y}(t_n) + h\int_0^1 \boldsymbol{f}(t_n + h\tau, \boldsymbol{y}(t_n + \tau))d\tau \\
&\approx \boldsymbol{y}(t_n) + h\sum_{j=1}^{\nu} b_j \boldsymbol{f}(t_n + c_j h, \boldsymbol{y}(t_n + c_j h))
\end{aligned}$$

- Probleem: we kennen de oplossing niet in de knopen ! We hebben een benadering nodig voor

$$\boldsymbol{y}(t_n + c_j h)$$

Expliciete Runge-Kutta methodes (2)

- <u>Vertrekpunt</u>:

$$\boldsymbol{y}(t_{n+1}) \approx \boldsymbol{y}(t_n) + h \sum_{j=1}^{\nu} b_j \boldsymbol{f}(t_n + c_j h, \boldsymbol{y}(t_n + c_j h))$$

- Benader $\boldsymbol{\xi}_j \approx \boldsymbol{y}(t_n + c_j h)$ op basis van vorige knopen

$$\begin{aligned}
\boldsymbol{\xi}_1 &= \boldsymbol{y}_n, \\
\boldsymbol{\xi}_2 &= \boldsymbol{y}_n + h a_{2,1} \boldsymbol{f}(t_n, \boldsymbol{\xi}_1), \\
\boldsymbol{\xi}_3 &= \boldsymbol{y}_n + h a_{3,1} \boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + h a_{3,2} \boldsymbol{f}(t_n + c_2 h, \boldsymbol{\xi}_2) \\
&\vdots \\
\boldsymbol{\xi}_\nu &= \boldsymbol{y}_n + h \sum_{i=1}^{\nu-1} a_{\nu,i} \boldsymbol{f}(t_n + c_i h, \boldsymbol{\xi}_i)
\end{aligned}$$

Definitie van de Runge-Kutta methode

$$\boldsymbol{\xi}_j = \boldsymbol{y}_n + h \sum_{i=1}^{j-1} a_{j,i} \boldsymbol{f}(t_n + c_i h, \boldsymbol{\xi}_i), \qquad j = 1, \ldots, \nu$$

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

<u>Terminologie</u>: RK matrix, RK knopen, RK gewichten, RK tableaux

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

<u>Voorbeelden</u>: (uitgewerkt op bord – met kleur)

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

Ordevoorwaarden voor Runge-Kutta

- Plaats op bord de condities die je direct kunt inzien
- Directe Taylorexpansie (kennisclip voor geval $\nu = 2$)
  - Intuïtieve condities komen effectief terug
  - Ontzettend veel rekenwerk voor hogere orde
- Taylorexpansie voor autonome vgl. (mag voor $\nu \leq 3$)
  - Iets minder rekenwerk, maar toch nog vervelend om te doen
- <u>Belangrijk</u>: maximaal te bereiken orde met expliciete RK
  - Voor $\nu \leq 4$ is de maximale orde $p = \nu$
  - Voor $p = 5$ moet $\nu \geq 6$; wordt erger voor hogere ordes

Impliciete Runge-Kuttamethodes

$$\boldsymbol{\xi}_j = \boldsymbol{y}_n + h \sum_{i=1}^{\overset{\nu}{\cancel{j-1}}} a_{j,i} \boldsymbol{f}(t_n + c_i h, \boldsymbol{\xi}_i), \qquad j = 1, \ldots, \nu$$

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

- We blijven nodig hebben voor consistentie dat

$$\sum_{i=1}^{\nu} a_{j,i} = c_j, \qquad j = 1, \ldots, \nu$$

- Relatief grote stelsels / Maximale orde wordt $p = 2\nu$
- <u>Vergelijk rekenwerk per stap!</u> Expliciet/impliciet – RK/LMS

Impliciete RK methodes: voorbeeld

$$\boldsymbol{\xi}_1 = \boldsymbol{y}_n + \frac{1}{4}h\left[\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) - \boldsymbol{f}(t_n + \frac{2}{3}h, \boldsymbol{\xi}_2)\right]$$

$$\boldsymbol{\xi}_2 = \boldsymbol{y}_n + \frac{1}{12}h\left[3\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + 5\boldsymbol{f}(t_n + \frac{2}{3}h, \boldsymbol{\xi}_2)\right]$$

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \frac{1}{4}h\left[\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + 3\boldsymbol{f}(t_n + \frac{2}{3}h, \boldsymbol{\xi}_2)\right]$$

Tableau

$$\begin{array}{c|cc} 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}$$

Deze methode heeft orde 3
(zie afleiding in kennisclip)

Omweg: het principe van collocatie

- Zoek een benaderende veelterm $\boldsymbol{u}$ van graad ν
- Eis dat deze veelterm voldoet aan de beginvoorwaarde…

$$\boldsymbol{u}(t_n) = \boldsymbol{y}_n$$

- … en exact aan de differentiaalvergelijking in ν collocatiepunten

$$\boldsymbol{u}'(t_n + c_j h) = \boldsymbol{f}(t_n + c_j h, \boldsymbol{u}(t_n + c_j h)), \qquad j = 1, \ldots, \nu$$

- Zet vervolgens

$$\boldsymbol{y}_{n+1} = \boldsymbol{u}(t_{n+1})$$

Collocatie als impliciete RK methode

Lemma 3.5 *Set*

$$q(t) := \prod_{j=1}^{\nu}(t - c_j), \qquad q_\ell(t) := \frac{q(t)}{t - c_\ell}, \quad \ell = 1, 2, \ldots, \nu,$$

and let

$$a_{j,i} := \int_0^{c_j} \frac{q_i(\tau)}{q_i(c_i)}\,\mathrm{d}\tau, \qquad j, i = 1, 2, \ldots, \nu, \tag{3.13}$$

$$b_j := \int_0^{1} \frac{q_j(\tau)}{q_j(c_j)}\,\mathrm{d}\tau, \qquad j = 1, 2, \ldots, \nu. \tag{3.14}$$

The collocation method (3.12) *is identical to the IRK method*

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

Collocatie versus impliciete RK formulering

- Elke collocatiemethode is een impliciete RK methode; niet elke impliciete RK methode is een collocatiemethode
- De collocatieformulering heeft geen voordelen voor implementatie; enkel voor analyse

Theorem 3.7 *Suppose that*

$$\int_0^1 q(\tau)\tau^j \,\mathrm{d}\tau = 0, \qquad j = 0, 1, \ldots, m-1, \tag{3.18}$$

for some $m \in \{0, 1, \ldots, \nu\}$. *(The polynomial* $q(t) = \prod_{\ell=1}^{\nu}(t - c_\ell)$ *has been defined already in the proof of Lemma 3.5.) Then the collocation method* (3.12) *is of order* $\nu + m$.[1]

- Gebruik van nulpunten van orthogonale veeltermen geeft hogere orde!

Oorzaak: resultaat over kwadratuur

- Beschouw een veelterm r van graad ν zodat

$$\langle r, \hat{p} \rangle = 0, \qquad \text{for every } \hat{p} \in \mathbb{P}_{m-1}$$

- Beschouw volgende kwadratuurregel
  - Kies als knopen $\{c_j\}_{j=1}^{\nu}$ de nulpunten van r (dit kan altijd !)
  - Kies gewichten $\{b_j\}_{j=1}^{\nu}$ om orde $p = \nu$ te bekomen

- De resulterende kwadratuurregel zal orde $p = \nu + m$ hebben.

Foutenschatting bij RK-methodes
Ingebedde Runge-Kuttamethodes

- Een eenvoudige foutenconstante bestaat niet voor Runge-Kuttamethodes. (Waarom niet?)
- Opnieuw twee methodes in tandem, maar nu met verschillende orde.
- Beschouw

$$\boldsymbol{y}_{n+1} = \tilde{\boldsymbol{y}}(t_{n+1}) + \boldsymbol{l}h^{p+1} + \mathcal{O}(h^{p+2})$$

$$\boldsymbol{x}_{n+1} = \tilde{\boldsymbol{y}}(t_{n+1}) + \mathcal{O}(h^{p+2})$$

- Dan schatten we $\boldsymbol{l}h^{p+1} \approx \boldsymbol{y}_{n+1} - \boldsymbol{x}_{n+1}$
- … en we krijgen $\kappa := \left\| \boldsymbol{y}_{n+1} - \boldsymbol{x}_{n+1} \right\|$

Implementatie van ingebedde RK-methodes

- Remeshing is niet nodig bij een 1-stap methode

- Rekenkost van foutenschatting beperkt houden in verhouding met de kost van een tijdstap ?
  - Naïef: (minstens) verdubbeling van rekenwerk (omwille van de stap met twee methodes)
  - Slimmer: hergebruik knopen in de tweede methode (vandaar de naam ingebed !)

(Merk op dat de rekenkost voor foutenschatting bij LMS-methodes beperkt werd door de fout voor een impliciete methode te schatten via een expliciete methode.)

Constructie van ingebedde RK-methodes

- Beschouw twee RK-methodes

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^\top \end{array} \qquad \text{and} \qquad \begin{array}{c|c} \tilde{\boldsymbol{c}} & \tilde{A} \\ \hline & \tilde{\boldsymbol{b}}^\top \end{array}$$

- Wanneer de RK-methodes expliciet zijn, kiezen we

$$\tilde{\boldsymbol{c}} = \begin{bmatrix} \boldsymbol{c} \\ \hat{\boldsymbol{c}} \end{bmatrix}, \qquad \tilde{A} = \begin{bmatrix} A & O \\ & \hat{A} \end{bmatrix}$$

- Compacte notatie

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

- Voorbeeld op bord.

Voorbeeld: RK23

- Beschouw het volgende ingebedde paar

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{2}{3} & \frac{2}{3} & & \\
\frac{2}{3} & 0 & \frac{2}{3} & \\
\hline
 & \frac{1}{4} & \frac{3}{4} & \\
\hline
 & \frac{1}{4} & \frac{3}{8} & \frac{3}{8}
\end{array}
$$

- Voor dit voorbeeld hebben we $p = 2 \qquad \tilde{p} = 3$

$$\kappa = \frac{3}{8} \left\| \boldsymbol{f}(t_n + \frac{2}{3}h, \xi_3) - \boldsymbol{f}(t_n + \frac{2}{3}h, \xi_2) \right\|$$

Illustratie: van der Pol vergelijking

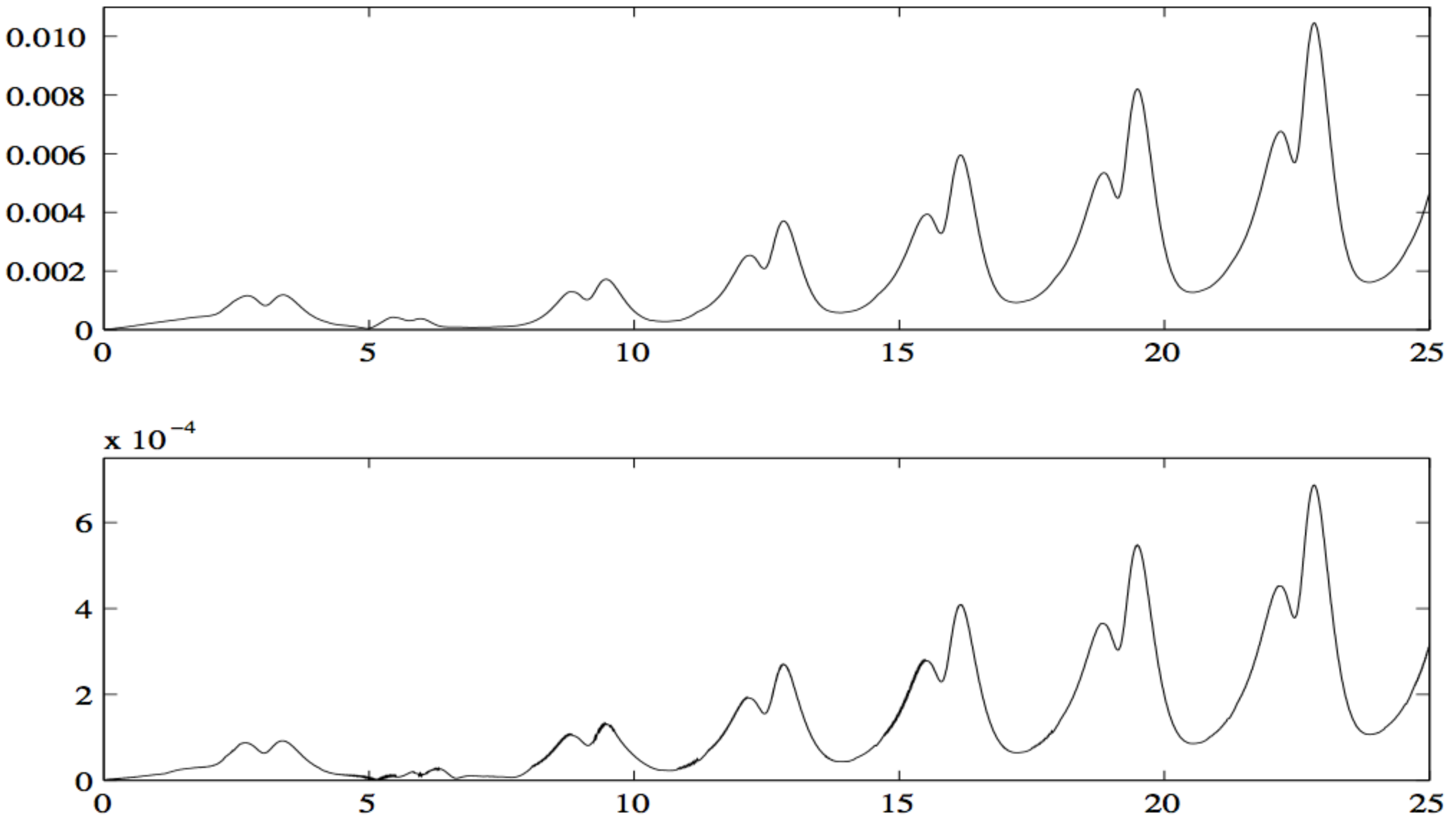
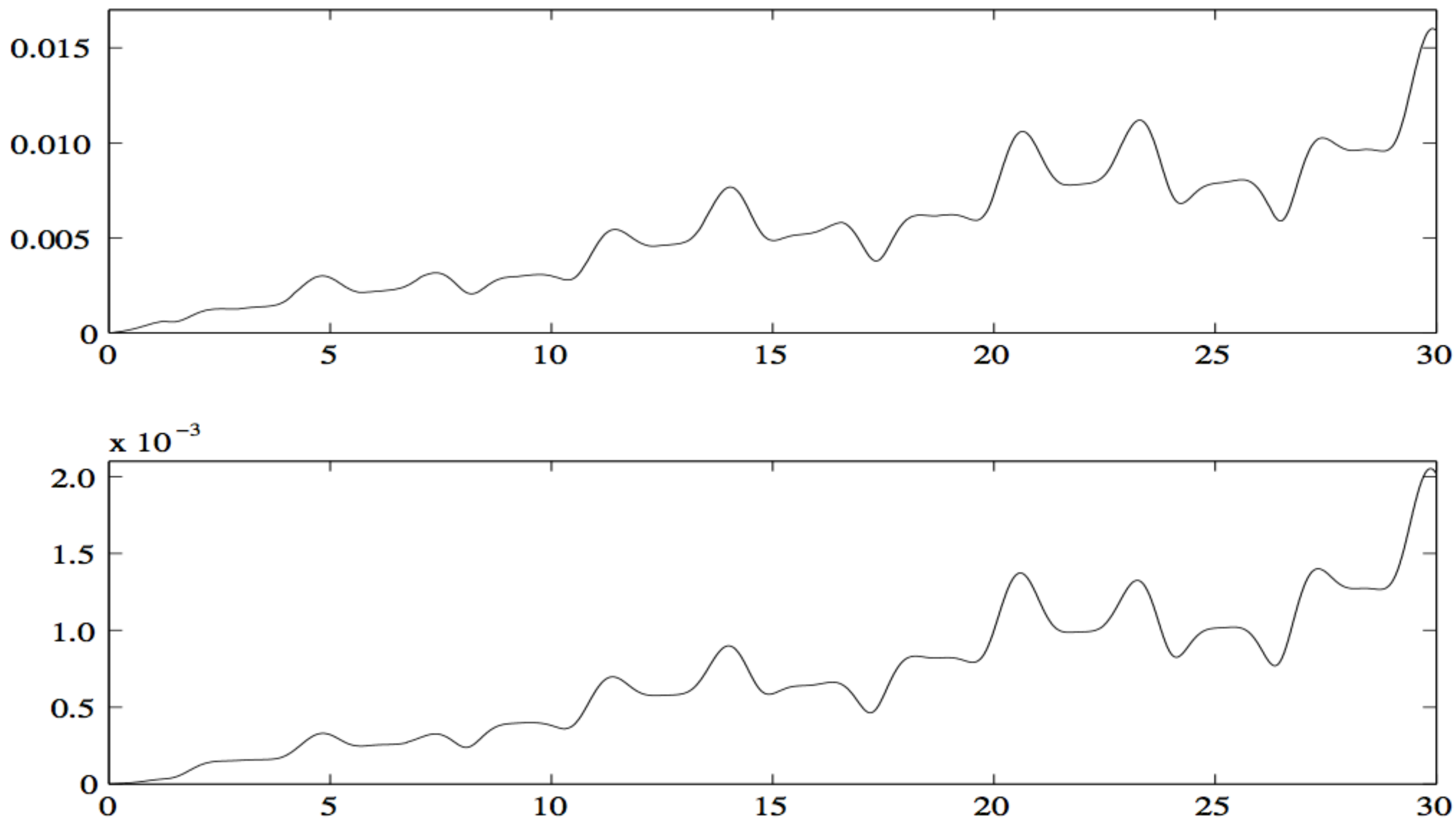
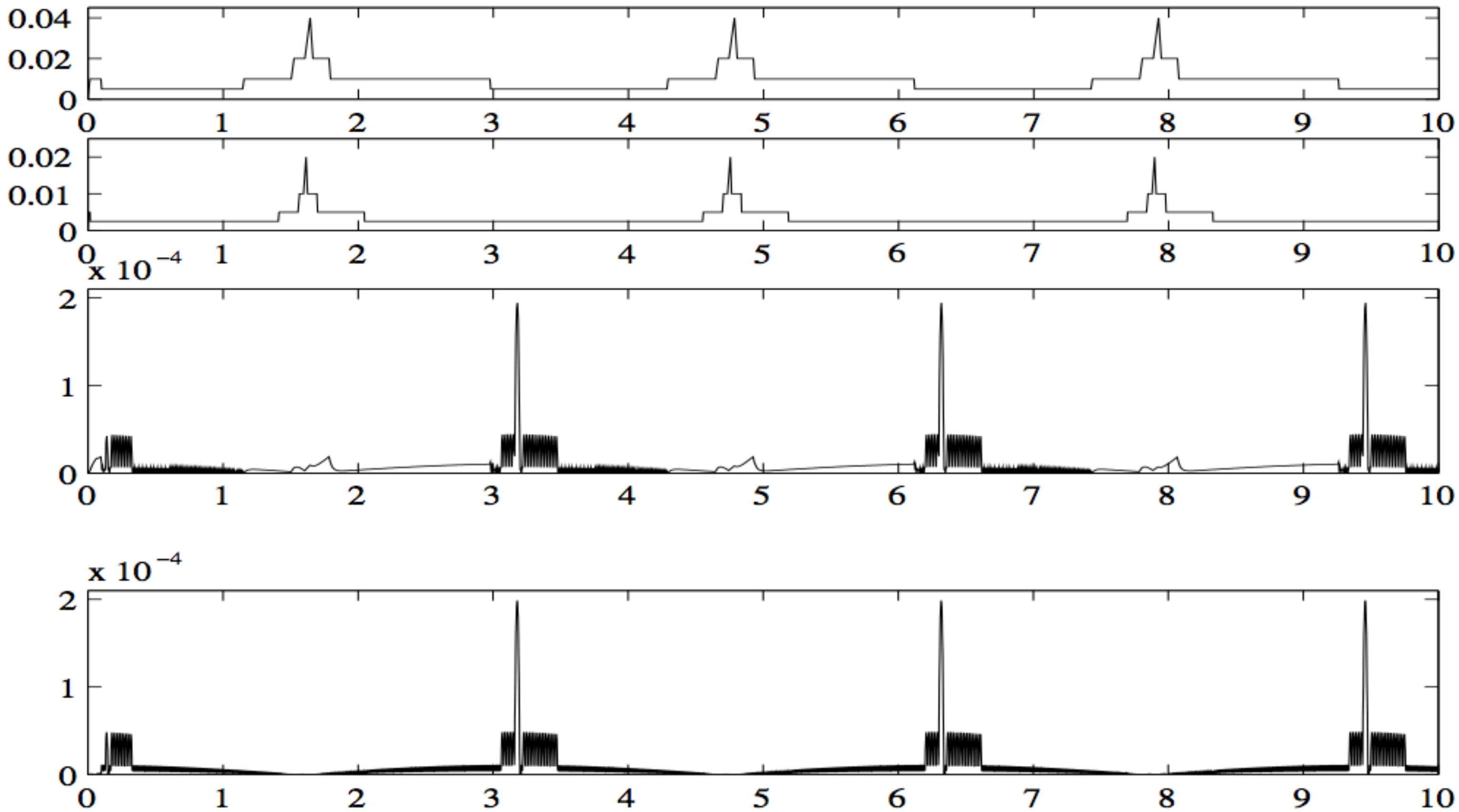


Figure 6.5 Global errors for the numerical solution of the van der Pol equation (6.2) by the embedded RK pair (6.15) with $\delta = 10^{-3}, 10^{-4}$.

Illustratie: Mathieuvergelijking



Illustratie: Curtiss-Hirschfelder vergelijking



Enkele bijkomende commentaren

- Voor LMS-methodes: orde en beginvoorwaarden ? Gear integration method
  - Variabele stap, variabele orde: start met een 1-stapmethode
  - Remeshing kan veel handiger door eindige differenties van oplossing bij te houden ipv oplossing (Nordsieck representatie)
- Zelf implementaties maken is vooral nuttig voor inzicht.
  - Er is zoveel werk gedaan aan het optimaliseren van software, dat je de beste implementaties nauwelijks kan overtreffen.
  - Matlab / Python-SciPy / SUNDIALS / …

Voor het examen

- Methodes kunnen **<u>herkennen</u>**.
  - Als er een formule staat, kunnen aangeven of het een lineaire multistap of Runge-Kutta methode is.
  - Impliciet, expliciet, uitspraken over orde kunnen doen.
- Gedrag van methodes kunnen **<u>herkennen en verklaren</u>**.
  - Als er een figuur of tabel gegeven wordt, kunnen afleiden welke methode gebruikt werd.
  - Kunnen aangeven op basis waarvan je tot die conclusie komt.
- Werking foutenschatters kunnen uitleggen.
- Lineaire multistapmethodes en Runge-Kuttamethodes kunnen vergelijken.