Programmierblatt 2.

Besprechungswoche: **16–20.11.2020**

Auf diesem Programmierblatt betrachten wir die Lösung gewöhnlicher Differentialgleichungen mit Hilfe von Mehrschrittverfahren. Vorgelegt sei dazu jeweils die gewöhnliche Differentialgleichung

$$\mathbf{y}'(x) = \mathbf{f}\big(x, \mathbf{y}(x)\big), \quad x \in [a, b] \tag{1}$$

mit $\mathbf{f}\colon [a, b] \times \mathbb{R}^d \to \mathbb{R}^d$ und dem Anfangswert $\mathbf{y}(a) = \mathbf{y}_a \in \mathbb{R}^d$.

Lineare Mehrschrittverfahren

Bekanntlich lauten lineare k-Schrittverfahren

$$\boldsymbol{\eta}_{i+k} := \sum_{\ell=0}^{k-1} \alpha_\ell \boldsymbol{\eta}_{i+\ell} + h \sum_{\ell=0}^{k} \beta_\ell \mathbf{f}(x_{i+\ell}, \boldsymbol{\eta}_{i+\ell}), \tag{2}$$

wobei $x_i := a + hi$ für $i \in \mathbb{N}$ ist. Sind die k Anlaufwerte

$$\boldsymbol{\eta}_0 = \mathbf{y}_a,\ \boldsymbol{\eta}_1 \approx \mathbf{y}(x_1),\ \ldots,\ \boldsymbol{\eta}_{k-1} \approx \mathbf{y}(x_{k-1})$$

vorgegeben, so berechnet man die weiteren Approximationen mit (2). Wenn $\beta_k = 0$ gilt, ist das Verfahren explizit und man benutzt direkt (2). Für $\beta_k \neq 0$ ist hingegen das Verfahren implizit und (2) im Allgemeinen eine nichtlineare Gleichung, welche in jedem Schritt mit einer numerischen Methode gelöst werden muss. Für dieses Blatt wollen wir dazu das Newton-Verfahren mit der Schrittweitensteuerung von Programmierblatt 1 anwenden. Dieses verwenden wir zur Lösung der Gleichung

$$\mathbf{g}_i(\boldsymbol{\eta}_{i+k}) \stackrel{!}{=} \mathbf{0} \quad \text{mit} \quad \mathbf{g}_i(\mathbf{z}) := \mathbf{z} - \sum_{\ell=0}^{k-1} \alpha_\ell \boldsymbol{\eta}_{i+\ell} - h \sum_{\ell=0}^{k-1} \beta_\ell \mathbf{f}(x_{i+\ell}, \boldsymbol{\eta}_{i+\ell}) - h\beta_k \mathbf{f}(x_{i+k}, \mathbf{z}).$$

Als Startnäherung $\mathbf{z}_0$ können wir dabei jeweils den Wert des vorhergehenden Zeitschritts, $\boldsymbol{\eta}_{i+k-1}$, benutzen.

Aufgabe 1 (explizite, lineare Mehrschrittverfahren)**.**

Schreiben Sie die Funktionen

```
function [x, y] = odeAB2(f, a, b, ya, n),
function [x, y] = odeAB3(f, a, b, ya, n),
```

welche das Anfangswertproblem (1) mit Hilfe des 2- respektive 3-Schritt-Adams-Bashforth-Verfahrens zur Schrittweite $h = (b-a)/n$ lösen. Die Funktion $\mathbf{f}$ soll dabei als *Function Handle* `f` übergeben werden. Die neben dem Anfangswert benötigten Anlaufwerte $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{k-1}$ berechnen Sie mit dem Runge-Kutta-Verfahren der Ordnung 4. Die Funktions-Outputs sollen von der Form `x` $= \begin{bmatrix} x_0 & x_1 & \cdots & x_n \end{bmatrix}$ und `y` $= \begin{bmatrix} \boldsymbol{\eta}_0 & \boldsymbol{\eta}_1 & \cdots & \boldsymbol{\eta}_n \end{bmatrix}$ sein.
Achten Sie bei der Implementierung darauf, dass Sie bereits berechnete Auswertungen der rechten Seite $\mathbf{f}$ wiederverwenden. Das bedeutet, dass pro Iterationsschritt nur eine Auswertung von $\mathbf{f}$ berechnet werden darf!

Aufgabe 2 (implizite, lineare Mehrschrittverfahren).

Schreiben Sie die Funktionen

```
function [x, y] = odeAM2(f, fy, a, b, ya, n, sigma, tol, maxiter),
function [x, y] = odeAM3(f, fy, a, b, ya, n, sigma, tol, maxiter),
function [x, y] = odeBDF2(f, fy, a, b, ya, n, sigma, tol, maxiter),
function [x, y] = odeBDF3(f, fy, a, b, ya, n, sigma, tol, maxiter),
```

welche das Anfangswertproblem (1) mit Hilfe des 2- respektive 3-Schritt-Adams-Moulton- respektive BDF-Verfahrens zur Schrittweite $h = (b-a)/n$ lösen. Die Funktionen $\mathbf{f}$ und $\mathbf{f_y}$ sollen dabei als *Function Handles* `f` und `fy` übergeben werden. Die neben dem Anfangswert benötigten Anlaufwerte $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{k-1}$ berechnen Sie mit dem θ-Schema für $\theta = 1/2$. Die anderen Inputs sind analog zu jenen der Funktionen `newtonIterationSWS`.

Prädiktor-Korrektor-Verfahren

Um eine Klasse von nichtlinearen Mehrschrittverfahren herzuleiten, betrachten wir ein implizites, lineares k-Schrittverfahren

$$\boldsymbol{\eta}_{i+k} := \sum_{\ell=0}^{k-1} \alpha_\ell \boldsymbol{\eta}_{i+\ell} + h \sum_{\ell=0}^{k-1} \beta_\ell \mathbf{f}(x_{i+\ell}, \boldsymbol{\eta}_{i+\ell}) + h\beta_k \mathbf{f}(x_{i+k}, \boldsymbol{\eta}_{i+k}).$$

Um diese Gleichung approximativ zu lösen, gehen wir wie folgt vor:

1. *Prädiktor-Schritt:* Wir berechnen eine Startnäherung $\boldsymbol{\eta}_{i+k}^{(0)}$ durch ein explizites, lineares k-Schrittverfahren
$$\boldsymbol{\eta}_{i+k}^{(0)} := \sum_{\ell=0}^{k-1} \widetilde{\alpha}_\ell \boldsymbol{\eta}_{i+\ell} + h \sum_{\ell=0}^{k-1} \widetilde{\beta}_\ell \mathbf{f}(x_{i+\ell}, \boldsymbol{\eta}_{i+\ell}).$$
2. *Korrektor-Schritte:* Wir verbessern die Startnäherung durch m Fixpunktiterationen,
$$\boldsymbol{\eta}_{i+k}^{(j)} := \sum_{\ell=0}^{k-1} \alpha_\ell \boldsymbol{\eta}_{i+\ell} + h \sum_{\ell=0}^{k-1} \beta_\ell \mathbf{f}(x_{i+\ell}, \boldsymbol{\eta}_{i+\ell}) + h\beta_k \mathbf{f}\Big(x_{i+k}, \boldsymbol{\eta}_{i+k}^{(j-1)}\Big),$$
mit $j = 1, \ldots, m$.
3. Wir setzen schliesslich $\boldsymbol{\eta}_{i+k} := \boldsymbol{\eta}_{i+k}^{(m)}$.

Aufgabe 3 (Prädiktor-Korrektor-Verfahren).

Schreiben Sie die Funktionen

```
function [x, y] = odeABM2(f, a, b, ya, n, m),
function [x, y] = odeABM3(f, a, b, ya, n, m),
```

welche das Anfangswertproblem (1) mit Hilfe des Prädiktor-Korrektor-Verfahrens zur Schrittweite $h = (b-a)/n$ lösen, welches als Prädiktor das Adams-Bashforth- und als Korrektor das Adams-Moulton-Verfahren je mit $k = 2$ respektive $k = 3$ verwenden. `m` bezeichnet die Anzahl Korrektor-Schritte m. Die neben dem Anfangswert benötigten Anlaufwerte $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{k-1}$ berechnen Sie mit dem Runge-Kutta-Verfahren der Ordnung 4.

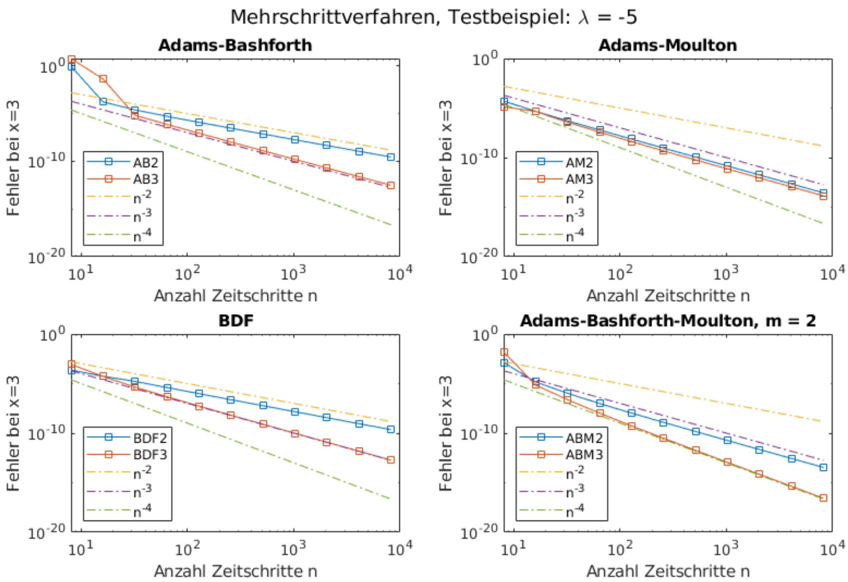
Achten Sie bei der Implementierung wieder darauf, dass Sie bereits berechnete Auswertungen der rechten Seite $\mathbf{f}$ wiederverwenden. Das bedeutet, dass pro Korrektor-Schritt nur eine Auswertung von $\mathbf{f}$ berechnet werden soll!

Numerische Beispiele

Mit den implementierten Verfahren zuhanden wollen wir diese nun an den Beispielen von Programmierblatt 1 testen.

Aufgabe 4 (Beispiel I: Testproblem)**.**

Vorgelegt sei das Anfangswertproblem $y' = \lambda y$ für $x \in [a, b]$ mit $a = 1$, $b = 3$ und $y(a) = 1$. Wählen Sie $\lambda = -5$ und benutzen Sie alle auf diesem Blatt implementierten Verfahren, um das Problem numerisch mit $n = 2^3, 2^4, \dots, 2^{13}$ Zeitschritten zu lösen und die Norm des Fehlers zwischen der analytischen Lösung und der jeweiligen Approximation an der Stelle $x = b$ zu berechnen und plotten. Zum Vergleich plotten Sie ebenfalls die Raten cn^{-p} für $p \in \{2, 3, 4\}$. Mit einem gut gewähltem $c > 0$ können Sie die Raten an eine gute Position setzen.
Für die impliziten Verfahren wählen Sie `sigma = 0.5`, `tol = 1e-12` und `maxiter = 42`; für die Prädiktor-Korrektor-Verfahren $m = 2$. Erstellen Sie damit einen Konvergenzplot in der Form:



Aufgabe 5 (Beispiel II: Lotka-Volterra-Modell)**.**

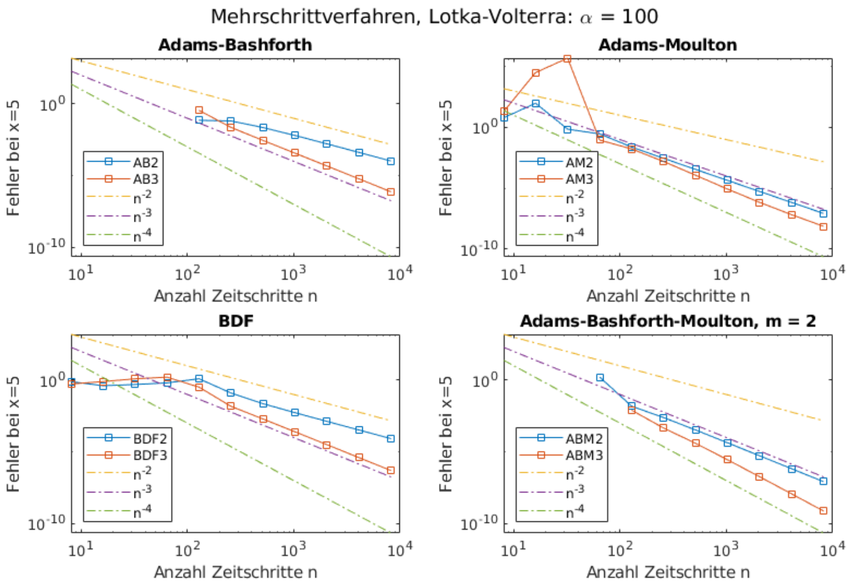
Wiederholen Sie Aufgabe 4, wobei Sie anstatt des Testproblems das Räuber-Beute-Modell aus Beispiel 1.2 aus der Vorlesung betrachten. Dieses ist durch

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) = \begin{bmatrix} \alpha y_1(1 - y_2) \\ y_2(y_1 - 1) \end{bmatrix}, \quad \alpha > 0,$$

gegeben. Wählen Sie $\alpha = 100$ und $a = 0$, $b = 5$ und $\mathbf{y}(a) = [3, 1]^\mathsf{T}$.
Die analytische Lösung im Punkt $b = 5$ lautet

$$\mathbf{y}(5) = \begin{bmatrix} 0.213646531148916 \\ 1.05468948232552 \end{bmatrix}.$$

Sie sollten hiermit einen Konvergenzplot der folgenden Form erhalten:



Aufgabe 6 (Beispiel III: Van-der-Pol-Oszillator)**.**

Wiederholen Sie Aufgabe 4, wobei Sie anstatt des Testproblems den Van-der-Pol-Oszillator, welcher durch

$$\begin{bmatrix} v' \\ u' \end{bmatrix} = \begin{bmatrix} \epsilon^{-1}(u - v^3/3 + v) \\ -v \end{bmatrix}$$

beschrieben wird, betrachten. Wählen Sie $\epsilon = 0.03$ und $a = 0$, $b = 10$ sowie $[v(0), u(0)]^\mathsf{T} = [0, 2]^\mathsf{T}$.

Die analytische Lösung bei $b = 10$ lautet

$$\begin{bmatrix} v(10) \\ u(10) \end{bmatrix} = \begin{bmatrix} 1.556042079300293 \\ -0.332027354872852 \end{bmatrix}.$$

Sie sollten hierfür einen Konvergenzplot der folgenden Form erhalten: