Die folgenden Aufgaben sind bis zur Übungsstunde vorzubereiten. In der Übung deklariert jede Gruppe, für welche Aufgabe sie imstande ist, eine Lösung zu präsentieren ('Kreuzerlliste' auf Papier). Für die Präsentation in der Übung wird pro Aufgabe eine UE-Gruppe aufgerufen.

- *steht für: Dies ist konkret zu tun.*

1.) *Ein einfaches homogenes System von linearen Differentialgleichungen. Untersuchung des Lösungsverhaltens.*

a) • *Bestimmen Sie die exakte Lösung* $\boldsymbol{y}(t) = (y_1(t), y_2(t))$ *des DGL-Systems* $\boldsymbol{y}'(t) = A \cdot \boldsymbol{y}(t)$, *mit*

$$A = \begin{pmatrix} -1 & 0 \\ \delta & -1 \end{pmatrix}$$

zu dem Anfangswert $\boldsymbol{y}(0) = (1, 1)$. (Dabei ist $\delta > 0$ ein reeller Parameter.)

b) • *Geht* $y_2(t)$ *gegen* ∞ *für* $t \to \infty$, *oder bleibt die Lösung beschränkt?*

c) • *An welcher Stelle* t *nimmt* $y_2(t)$ *seinen maximalen Wert an? Bleibt dieser maximale Wert endlich, wenn man* $\delta \to \infty$ *gehen lässt?*

2.) *Numerische Lösung des Systems aus Aufgabe* **1.)**

a) • *Lösen Sie das System aus Aufgabe* **1.)** *mit dem expliziten Euler-Verfahren für den Fall* $\delta = 1$. *Integrieren Sie von* $t = 0$ *bis* $t = 1$ *mit vorgegebener Schrittweite* h.

Anmerkung: Für Ihren Euler-Code ist h ein vorzugebender Parameter, $h = 1/n$, wenn n Schritte von $t = 0$ bis $t = 1$ durchgeführt werden.

b) • *Experimentieren Sie mit* $h = 1/2$ *(2 Schritte),* $h = 1/4$ *(4 Schritte),* $h = 1/8$ *(8 Schritte), usw. Aus Aufgabe* **1.)** *kennen Sie die exakte Lösung an der Stelle* $t = 1$ *und können daher vergleichen. Wie verhält sich der Approximationsfehler in Abhängigkeit von* h*?*

c) • *Analog zu* **b)***; verwenden Sie jedoch das klassische Runge-Kutta Verfahren. Vergleichen Sie die Approximationsqualität mit der des Euler-Verfahrens.*

3.) *Numerische Lösung von Aufgabe* **5b)** *aus Übung 1.*

a) • *Wählen Sie konkrete numerische Werte für die Halbachsen* a, b *der Ellipse und die erwünschte Geschwindigkeit* c *und lösen Sie die Differentialgleichung mit dem klassischen Runge-Kutta-Verfahren, und zwar*

- *mit Ihrem eigenen Code und konstanter Schrittweite,*
- *mit* MATLAB /`ode45`.

Hinweis: Transformieren Sie auf ein System 1. Ordnung. Integrieren Sie über einen etwas längeren Zeitraum, so dass mehrere Umläufe um die Ellipse erkennbar sind. Die Lösung $\varphi(t)$ muss t-periodisch sein! Wie lange dauert etwa ein Umlauf?

b) • *Werten Sie entlang der so enthaltenen Lösung die Geschwindigkeit* $v(t)$ *aus (siehe Angabe aus Übung 1) – diese muss (fast) konstant sein.*

4.) *Integration mit adaptiv angepasster Schrittweite.*

Wir betrachten eine skalare DGL $y'(t) = f(t, y(t))$ mit Anfangsbedingung $y(0) = y_0$ und denken uns darauf das explizite Euler-Verfahren angewendet. Wir verwenden jetzt aber eine *variable Schrittweite* $h_i = t_i - t_{i-1}$ über einem nichtäquidistanten Gitter $\{t_0, t_1, t_2, \ldots\}$ und versuchen die Schrittweiten h_i optimal anzupassen. (D.h., die t_i werden erst nach und nach gewählt.)

Als Vorüberlegung betrachten wir einen Euler-Schritt ausgehend von y_i an der Stelle t_i,

$$y_{i+1} = y_i + h_i\, f(t_i, y_i) \approx y(t_i + h_i),$$

wobei hier $y(t)$ die 'lokale' Lösung der gegebenen DGL zum Startwert $y(t_i) = y_i$ bezeichnet. Um für den *lokalen Fehler* $y_{i+1} - y(t_i + h_i)$ eine berechenbare Schätzung zu gewinnen, verwenden wir eine genauere Approximation, nämlich ein Runge-Kutta Verfahren der Ordnung 2, d.h. wir berechnen

$$\tilde{y}_{i+1} := y_i + h_i\, f\big(t_i + \tfrac{h_i}{2}, y_i + \tfrac{h_i}{2}\, f(t_i, y_i)\big),$$

und verwenden $y_{i+1} - \tilde{y}_{i+1}$ als Approximation für den lokalen Fehler. Eine theoretische Überlegung zeigt, dass gilt

$$e_i := y_{i+1} - \tilde{y}_{i+1} \approx C\, h_i^2,$$

mit einer Konstante C, die man zwar nicht kennt (sie hängt vom lokalen Lösungsverhalten ab), die aber nicht von der Schrittweite h_i abhängt.

a) Sei `tol` (z.B. `tol` = `1E-3`) ein Genauigkeitsniveau, das man anstrebt, d.h. der Betrag des lokalen Fehlers soll in jedem Schritt etwa gleich `tol` sein.

- *Überlegen Sie, wie man die Schrittweite h_i ändern muss, so dass zu erwarten ist, dass der 'neue' lokale Fehler an der Stelle $t_i + h_i^{(neu)}$ (mit dem 'neuen' $h_i^{(neu)}$) etwa gleich* `tol` *sein wird. Leiten Sie eine entsprechende Formel her, die $h_i^{(neu)}$ in Abhängigkeit von h_i, e_i und* `tol` *darstellt.*

b) • *Implementieren Sie einen adaptiven DGL-Solver,*[1] *der diese Strategie umsetzt und für eine gegebene DGL mit Anfangswert y_0 und gegebenes* `tol` *die numerische Lösung startend bei (t_0, y_0) bis zu einer Endstelle $t = t_{end}$ über einem adaptiv generierten Gitter berechnet.*

Anmerkung: Man startet (z.B.) mit $h_1 =$ `tol`, rechnet einen Schritt, berechnet $h_1^{(neu)}$ und rechnet den Schritt nochmals. Dann beginnt man den zweiten Schritt mit $h_2 = h_1^{(neu)}$, usw.

c) • *Wählen Sie als Beispiel (z.B.) die DGL $y' = y^2$ mit Anfangswert $y(0) = 1$ und integrieren Sie von $t_0 = 0$ bis $t_{end} = 1/2$.*

Integrieren Sie auch bis $t_{end} = 1$. (Achtung: blowup - wie geht man mit so etwas um?)

Vergleichen Sie Ihre Ergebnisse mit MATLAB / `ode45`.

5.) *Animation eines weit ausschwingenden Pendels.*

Die Bewegung eines Pendels in Abhängigkeit von der Zeit t wird durch den Verlauf der Funktion $\varphi = \varphi(t)$ beschrieben, wobei $\varphi(t)$ den Winkel weg von der Vertikallage bedeutet. $\varphi = 0$ entspricht der Vertikallage (Ruhelage). (Machen Sie eine Skizze.)

Wenn nur kleine Auslenkungen φ auftreten, dann entspricht die Bewegung des Pendels der Lösung einer linearen (genauer: linearisierten)[2] DGL der Gestalt

$$\ddot{\varphi}(t) = \varphi(t)$$

(oder ähnlich), die man exakt lösen kann; die Lösung $\varphi(t)$ ist durch sin- oder cos-Funktionen darstellbar. Bei größeren Auslenkungen muss man jedoch mit der nichtlinearen DGL

$$\ddot{\varphi}(t) = \sin(\varphi(t))$$

arbeiten, die man nicht exakt lösen kann.

[1] MATLAB / `ode45` beruht auf einer ähnliche Strategie, allerdings basierend auf zwei Runge-Kutta Verfahren höherer Ordnung.

[2] Anmerkung: Es gilt $\sin\varphi \approx \varphi$ für kleine φ.

a) • *Schreiben Sie das Problem als System 1. Ordnung an.*

b) • *Wählen Sie als Anfangswerte $\varphi(0) = \pi/2$ (große Auslenkung am Anfang!) und $\dot{\varphi}(0) = 0$ (Pendel in Ruhe, bevor es losgelassen wird), und lösen Sie sowohl das linearisierte Problem $\ddot{\varphi} = \varphi$ als auch das nichtlineare Problem $\ddot{\varphi} = \sin(\varphi)$ mit* MATLAB /`ode45`. *Integrieren Sie so weit, dass mehrere Schwingungen auftreten (dazu beobachten Sie einfach die Lösung) und machen Sie eine Animation, die die Schwingung des Pendels gemäß der 'falschen' (linearisierten) und der richtigen Lösung grafisch visualisiert.*

c) • *Zusatzfrage: Für die Anfangsbedingung $\varphi(0) = \pi$ und $\dot{\varphi}(0) = 0$ ist die exakte Lösung (in beiden Fällen) durch $\varphi(t) \equiv \pi$ für alle t gegeben. Verifizieren Sie das. Ist diese Lösung 'physikalisch gesehen' als realistisch anzusehen?*

6.) *Kepler-Problem (i).*

Die Bahnkurve $(x = x(t), y = y(t))$ einer Masse (z.B. Mond, Satellit, ...) um eine im Ursprung $(0, 0)$ konzentriert gedachte dominante Masse (z.B. Erde) wird in 2D durch ein System von Differentialgleichungen folgenden Typs beschrieben (leicht vereinfacht):

$$\ddot{x} = f(x, y) = -\frac{x}{(x^2 + y^2)^{3/2}}, \qquad \ddot{y} = g(x, y) = -\frac{y}{(x^2 + y^2)^{3/2}}$$

Durch Lösung dieses Systems kann man – je nach Anfangsbedingung – alle möglichen Bahnkurven erzeugen. Diese verlaufen entlang von Kegelschnitten in der xy-Ebene. Z.B. erhält man durch die Vorgabe

$$x(0) = 1,\ y(0) = 1,\ \dot{x}(0) = -1,\ \dot{y}(0) = 1/3$$

eine Lösungskurve $(x(t), y(t))$, die entlang einer Ellipse in der xy-Ebene verläuft.

a) • *Schreiben Sie das äquivalente 4-dimensionale System 1. Ordnung an,*

$$\dot{W} \;=\; F(W), \quad W(0) = W_0,$$

mit $W = W(t) = (x(t), \dot{x}(t), y(t), \dot{y}(t))$).
Hinweis: Wie in der Vorlesung erläutert, werden hier $\dot{x}$ und $\dot{y}$ als separate Unbekannte betrachtet.

b) • *Lösen Sie dieses System mit Hilfe des expliziten Euler-Verfahrens mit einer vernünftigen Schrittweite $h = 0.1$ (notfalls kleiner). (u_n ist dann eine Approximation für $u(t_n) = u(n\,h)$.*

Ein Umlauf ist zum Zeitpunkt $t \approx 38$ vollendet. Integrieren Sie jedoch weiter, etwa bis $t = 380$ (10 Umläufe), und plotten Sie die Bahnkurve $(x(t), y(t))$. Wird der Verlauf richtig wiedergegeben?

7.) *Kepler-Problem (ii).*

Das System aus **5a)** lässt sich in folgender Form anschreiben: Setze

$$X = \begin{pmatrix} x \\ y \end{pmatrix}, \quad U = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix},$$

Dann gilt

$$\dot{X} = F(U), \quad \dot{U} = G(X) \qquad (*)$$

a) Das System (*) hat eine spezielle, partitionierte Struktur. Beide Anteile sind zweidimensional, und die Gleichung für X hängt nur von U ab und umgekehrt.

• *Wie lauten die Vektorfelder $F(U)$ und $G(X)$?*

b) Für ein partitioniertes System der Gestalt (*) betrachten wir das sogenannte *symplektische Euler-Verfahren* (mit Schrittweite h): Ein Schritt $(X_{i-1}, U_{i-1}) \mapsto (X_i, U_i)$ ist gegeben durch

$$\frac{X_i - X_{i-1}}{h} = F(U_i),$$

gefolgt von

$$\frac{U_i - U_{i-1}}{h} = G(X_i)\,.$$

- *Worin besteht der Unterschied zum expliziten Euler-Verfahren?*
- *Implementieren Sie das symplektische Euler-Verfahren und vergleichen Sie mit* **5.)**.

Anmerkung: Der Unterschied ist signifikant bei Langzeitintegration (Verfolgung der Bahnkurven über viele Umläufe hinweg).

8.) *Eine Hundebahnkurve.*

Herr X. steht an der Stelle $(0,0)$ in der xy-Ebene, sein Hund Bello an der Stelle $(-c,0)$ c Meter links neben ihm. X. geht mit konstanter Geschwindigkeit v_1 in y-Richtung los. Bello läuft mit konstanter Geschwindigkeit v_2 hinterher, und zwar so, dass der Hundegeschwindigkeitsvektor in jedem Moment der Bewegung genau dorthin zeigt, wo sich Herr X. gerade befindet. Wir nehmen an, dass Bello langsamer läuft als Herr X. (d.h. $v_2 < v_1$). Gesucht ist die Hundebahnkurve

$$B(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

als Funktion der Zeit t.

Im Folgenden ist **a), b)** eine freiwillige Übung. Teil **c)** ist unabhängig davon realisierbar.

a) • *Stellen Sie ein System von zwei Differentialgleichungen auf, die die Hundebahnkurve beschreiben.*

b) • *Lösen Sie das System, z.B. für $c = 1$, und $v_1 = 2$, $v_2 = 1$, mit einer numerischen Methode Ihrer Wahl und stellen Sie den Verlauf der Bewegung von Herrn X. und von Bello grafisch dar. (Eine Animation wäre auch nett.)*

c) • *Wie* **b)***; realisieren und testen Sie jedoch Ihre eigene, sehr einfache aber plausible numerische Methode:*

Man wählt ein kleines Zeitintervall h und simuliert Bellos Bewegung ausgehend von der gegebenen Aufgabenstellung in folgender Weise:

- $t = 0 \ldots h$:
  - Herr X. bleibt noch bei $(0,0)$ stehen, und Bello läuft ein Stück auf ihn zu.
  - Bello bleibt kurz stehen, und X. geht ein Stück weiter.
- $t = h \ldots 2h$:
  - X. bleibt kurz stehen, und Bello läuft ein Stück auf ihn zu.
  - Bello bleibt kurz stehen, X. geht ein Stück weiter.
- usw.

Beachten Sie, dass alle Teilschritte in sehr einfacher Weise exakt ausführbar sind. Natürlich begeht man hier auch einen Approximationsfehler, aber das Verfahren ist sehr einfach realisierbar.

Anmerkung: Numerische Verfahren, die auf derartigen Ansätzen beruhen, werden als *Splitting-Verfahren* bezeichnet.

(Das symplektische Euler-Verfahren aus Aufgabe **7.)** beruht auf einer ähnlichen Idee. Es gibt auch genauere Verfahren dieses Typs.)