Institut für Numerische und Angewandte Mathematik
FB Mathematik und Informatik der Universität Münster
Prof. Dr. Christian Engwer, MSc Marcel Koch 12.10.2017

Übung zur Vorlesung

Wissenschaftliches Rechnen

WS 2017/18 — Blatt 1

Abgabe: 19.10.2017, 10:00 Uhr, Briefkasten 111
Code zusätzlich per e-mail an `marcel.koch@uni-muenster.de`

Achtung: Achten Sie darauf, Ihre Programme ordentlich zu formatieren und gut zu kommentieren. Die Form wird mit in die Bewertung eingehen.

Aufgabe 1 (Mathematisches Pendel) (4 Punkte)

In der Vorlesung haben wir ein mathematisches Modell für ein Pendel hergeleitet, das sogenannte mathematische Pendel (Kapitel 1.1.3). Dieses wird durch die gewöhnliche, nichtlineare Differentialgleichung $\varphi''(t) + \frac{g}{l}\sin(\varphi(t)) = 0$ beschrieben, die sich für kleine Auslenkungswinkel $\varphi \ll 1$ durch die Näherung $\sin\varphi \approx \varphi$ weiter vereinfachen lässt. Dadurch ergibt sich folgendes lineare Anfangswertproblem:

$$\varphi''(t) + \frac{g}{l}\varphi(t) = 0 \qquad \text{auf} \qquad I := [0,\infty), \tag{1a}$$
$$\varphi(0) = \varphi_0, \qquad \varphi_0 \in \mathbb{R}, \tag{1b}$$
$$\varphi'(0) = v_0, \qquad v_0 \in \mathbb{R} \tag{1c}$$

(a) Bestimmen Sie die analytische Lösung des Anfangswertproblems (1).

(b) Bestimmen Sie für Anfangswerte $\varphi_0 \neq 0$, $v_0 \neq 0$ mit Hilfe der analytischen Lösung die Winkelgeschwindigkeit $\varphi'(t)$ beim Durchgang der Ruheposition und den maximalen Auslenkungswinkel $\hat{\varphi}$ (die Winkelamplitude).

(c) Bestimmen Sie die Periodendauer T, d.h. die Dauer einer kompletten Schwingung (hin und zurück).

Tipp zu (a): Lösungen von (1a) *können mit Hilfe eines Fundamentalsystems dargestellt werden. Über die Anfangsbedingungen* (1b), (1c) *ergibt sich speziell die Lösung von* (1).

Tipp zu (b): Sie werden folgende Formeln gebrauchen können:

$$\sin(\arctan(x)) = \frac{x}{\sqrt{1+x^2}}, \quad \cos(\arctan(x)) = \frac{1}{\sqrt{1+x^2}}.$$

Tipp zu (c): Betrachten Sie die Nullstellen von φ.

Definition 1 (Explizites Euler-Verfahren)

Gegeben sei ein Anfangswertproblem (AWP) 1. Ordnung

$$\begin{aligned} y' &= f(t,y), \qquad \text{auf} \quad I := [t_0, T], \qquad && t_0, T \in \mathbb{R}_0^+, \\ y(t_0) &= y_0, && y_0 \in \mathbb{R}^n, \end{aligned}$$

mit $f : I \times \mathbb{R}^n \to \mathbb{R}^n$ stetig, $n \in \mathbb{N}$. Auf I sei $I_{\Delta t} := \{t_k = t_0 + k\Delta t \mid k = 0, 1, 2, \ldots \ \wedge \ t_k \leq T\}$ ein gewähltes Gitter mit zugehöriger Schrittweite $\Delta t \in \mathbb{R}^+$. Dann liefert die Iterationsvorschrift

$$y_{k+1} = y_k + \Delta t f(t_k, y_k), \quad k = 0, 1, 2, \ldots$$

Approximationen $y_k \approx y(t_k)$ der Lösung des AWPs. Hängt f nicht von der Zeit ab, d.h. für *autonome gewöhnliche Differentialgleichungen* $y' = f(y)$ mit $f : \mathbb{R}^n \to \mathbb{R}^n$, reduziert sich die Iterationsvorschrift zu $y_{k+1} = y_k + \Delta t f(y_k)$.

Aufgabe 2 (Mathematisches Pendel, Computermodell) (4 Punkte)

$$\varphi''(t) + \frac{g}{l} \sin(\varphi(t)) = 0 \qquad \text{auf} \quad I := [0, T], \qquad T \in \mathbb{R}^+, \tag{2a}$$

$$\varphi(0) = \varphi_0, \qquad \varphi_0 \in \mathbb{R}, \tag{2b}$$

$$\varphi'(0) = v_0, \qquad v_0 \in \mathbb{R}. \tag{2c}$$

Um das mathematische Pendel (2) zu simulieren, wollen wir für dieses ein Computermodell herleiten. Mit dem Ansatz $\varphi'(t) = v(t)$ lässt sich (2) in ein System von zwei gewöhnlichen Differentialgleichungen 1. Ordnung mit zugehörigen Anfangswerten transformieren:

$$\varphi'(t) = v(t), \qquad v'(t) = a \qquad \text{auf} \quad I, \tag{3a}$$

$$\varphi(0) = \varphi_0, \qquad v(0) = b. \tag{3b}$$

(a) Bestimmen Sie die fehlenden Terme a und b. Diskretisieren Sie das resultierende AWP (3) mit dem expliziten Euler-Verfahren indem Sie es wie in Definition 1 aufschreiben.

(b) Implementieren Sie das resultierende Computermodell in C++.

- Realisieren Sie die Konstanten $g, l, T, \Delta t$ und φ_0, v_0 im Hauptprogramm durch

```
const double g = 9.80665;          const double l = 1.0;
const double T = ...;              const double dt = ...;
const double phi_0 = ...;          const double v_0 = ...;
```

- Verwenden Sie Ihre Implementierung des expliziten Euler-Verfahrens aus Aufgabe 3. Wählen Sie dabei als Template-Parameter `VectorType` den Datentyp `std::array` und für die Komponenten des Vektors sowie als `TimeType` den Datentyp `double`.
- Lassen Sie sich t_k und den Auslenkungswinkel $\varphi_k \approx \varphi(t_k)$ in jedem Zeitschritt auf der Konsole ausgeben. Die Ausgabe Ihres Programms kann mit

  ```
  ./myprogramname > datafilename.txt
  ```

  in eine Textdatei `datafilename.txt` umgeleitet werden. Bei zeilenweiser Ausgabe im Format $t_k \sqcup \varphi_k$, wobei $\sqcup$ ein Leerzeichen darstellt, kann φ_k in `gnuplot` mit dem Befehl `plot "./datafilename.txt" smooth csplines` geplottet werden. Sie können ihr Programm auch gerne direkt in eine Textdatei schreiben lassen.

(c) Führen Sie mit Ihrem Programm verschiedene Simulationen durch. Benutzen Sie dabei den Anfangswert $v_0 = 0$ und die maximale Zeit $T = 12.0$.

(i) Wählen Sie $\varphi_0 = 0.5$ als festen Anfangswert.
Wählen Sie verschiedene Schrittweiten $\Delta t = 2^{-i}, \;\; i = 4, 5, 6, \ldots$.
Plotten Sie die Simulationsergebnisse in eine gemeinsame Grafik.

(ii) Wählen Sie die beiden verschiedenen Anfangswerte $\varphi_0 = 0.5$ und $\varphi_0 = 1.0$.
Wählen Sie $\Delta t = 2^{-15}$ als feste Schrittweite.
Plotten Sie die Simulationsergebnisse in eine gemeinsame Grafik.

(d) Beschreiben Sie Ihre Beobachtungen in (i). Beschreiben Sie Ihre Beobachtungen in (ii); betrachten Sie hier insbesondere die Periodendauer. Steht das Ergebnis in Konflikt mit dem von (c) aus Aufgabe 1? Begründen Sie Ihre Antwort.

Tipp zu (c): Auf der Vorlesungshomepage finden Sie Shell-Skripte und `gnuplot`*-Dateien, die Ihnen dabei helfen könnten die Simulationen automatisiert ablaufen zu lassen.*

Aufgabe 3 (Explizites Euler-Verfahren) (4 Punkte)

Implementieren Sie ein Klassentemplate `ExplicitEulerScheme` zur Durchführung des expliziten Euler-Verfahrens für den allgemeinen Fall eines nicht-autonomen AWPs 1. Ordnung. Das Verfahren soll für verschiedene Datentypen anwendbar sein, mit denen sich Vektoren in $\mathbb{R}^n$ und Zeiten in $\mathbb{R}_0^+$ repräsentieren lassen. Zu diesem Zweck soll `ExplicitEulerScheme` zwei Template-Parameter `VectorType` und `TimeType` besitzen. Setzen Sie voraus, dass `VectorType` eine Methode `operator[]` für den indexbasierten Zugriff auf die Elemente des Vektors und eine Methode `size` zu Verfügung stellt, wie es viele STL-Containerklassen machen (siehe z.B. `std::array`).

- Der Konstruktor des Klassentemplates soll die Funktion $f : I \times \mathbb{R}^n \to \mathbb{R}^n$ als Objekt der `std::function` Klasse mit Template-Parameter `Range(const TimeType&, const Domain&)` erhalten. Um die Verwendung dieses Typs etwas leichter zu gestalten, können Sie folgenden Typ-Alias verwenden:

```
template<typename TimeType, typename Range, typename Domain>
using TimeFunction = std::function<Range(const TimeType&, const Domain&)>;
```

- Mit der Methode

```
void apply (const TimeType& t, const TimeType& dt,
            const VectorType& y_old, VectorType& y_new) const
```

  sollen die durch die Iterationsvorschrift festgelegten Schritte des Verfahrens ausgeführt werden können. Dabei bezeichnen `t` sowie `dt` den Zeitpunkt t_k sowie die Schrittweite Δt des aktuellen Zeitschritts und `y_old` sowie `y_new` entsprechen den Approximationen y_k und y_{k+1} zum alten und neuen Zeitpunkt.