FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Fachbereich Elektrotechnik
und Informationstechnik

Prof. Georg Hoever

SS 2025

3. Praktikum zur Höhere Mathematik 2 für (Wirtschafts-)Informatik

Ziel dieses Praktikums ist eine Implementierung des Euler- und des Heun-Verfahrens zur numerischen Lösungen von Differenzialgleichungen.

Dabei wird die Klasse `CMyVektor` des ersten Praktikums weiter verwendet. Includieren Sie dazu Ihre entsprechende `h`- und `cpp`-Datei.

Hinweis: Die Implementierung der Methoden zu Aufgabe 1 sowie die Implementierung der Funktion zu Aufgabe 2 und das Euler- und Heunverfahren zu Aufgabe 3 sollen alle zusammen in **einer** `cpp`-Datei enthalten sein.

1. Aufgabe

Deklarieren Sie eine Klasse `C_DGLSolver`, die sowohl mit einem Differenzialgleichungssystem erster Ordnung

$$\begin{aligned} y_1' &= f_1(y_1, y_2, \ldots, y_n, x) \\ y_2' &= f_2(y_1, y_2, \ldots, y_n, x) \\ &\vdots \\ y_n' &= f_n(y_1, y_2, \ldots, y_n, x) \end{aligned}$$

als auch mit einer Differenzialgleichung höherer Ordnung

$$y^{(n)} = f(y, y', \ldots, y^{(n-1)}, x)$$

umgehen kann.

Dazu soll es jeweils einen Konstruktor geben, der aufgerufen werden kann

- mit einem Funktionspointer

  `CMyVektor (*f_DGL_System)(CMyVektor y, double x)`,

  zu einer Funktion, die die rechte Seite eines DGL-Systems repräsentiert,
- mit einem Funktionspointer

  `double (*f_DGL_nterOrdnung)(CMyVektor y, double x)`,

  zu einer Funktion, die eine DGL höherer Ordnung repräsentiert.

Der entsprechende Funktionspointer soll als privates Attribut abgespeichert werden. Ferner soll ein privates Attribut speichern, ob ein DGL-System oder eine DGL höherer Ordnung behandelt werden soll; dieses soll bei Aufruf des entsprechenden Konstruktors gesetzt werden.

2. Aufgabe

Für den Fall einer DGL höherer Ordnung muss diese in ein DGL-System erster Ordnung transformiert werden (vgl. Skript, S. 19).

Implementieren Sie dazu in `C_DGLSolver` eine private Methode

`CMyVektor ableitungen(CMyVektor y, double x)`,

die

- für den Fall, dass eine Funktion für ein DGL-System übergeben wurde, diese Funktion an der Stelle $(y; x)$ auswertet,
- für den Fall, dass eine Funktion für eine DGL höherer Ordnung übergeben wurde, eine Transformation in ein DGL-System durchführt und dazu die rechte Seite eines entsprechenden Systems ausgibt (vgl. Skript, S. 19).

3. Aufgabe

Implementieren Sie in `C_DGLSolver` jeweils eine öffentliche Methode, die das Euler- und die das Heun-Verfahren realisiert unter Benutzung der `ableitungen`-Methode, so dass je nach Initialisierung ein DGL-System erster Ordnung oder eine DGL höherer Ordnung gelöst wird.

Den Methoden sollen als Parameter übergeben werden:

- der Startwert x_{Start} und der Endwert x_{End},
- die Anzahl der Schritte, um von x_{Start} zu x_{End} zu kommen,
- die Startwerte `CMyVektor y_Start`, die je nach Initialisierung die Startwerte

  $$y_1(x_{\text{Start}}), \quad y_2(x_{\text{Start}}), \quad \ldots, \quad y_n(x_{\text{Start}})$$

  (bei einem DGL-System) bzw.

  $$y(x_{\text{Start}}), \quad y'(x_{\text{Start}}), \quad \ldots, \quad y^{(n-1)}(x_{\text{Start}})$$

  (bei einer DGL n-ter Ordnung) angeben.

4. Aufgabe

Testen Sie Ihren Solver

- an dem DGL-System

$$\begin{aligned} y_1' &= 2y_2 - xy_1, \\ y_2' &= y_1 y_2 - 2x^3 \end{aligned}$$

mit den Startwerten

$$y_1(0) = 0, \quad y_2(0) = 1.$$

Berechnen Sie zu y_1 und y_2 Näherungen für die Stelle $x_{\text{End}} = 2$ mit 100 Schritten mit dem Euler- und dem Heun-Verfahren.

(Für Interessierte: Die exakte Lösung ist $y_1(x) = 2x$, $y_2(x) = x^2 + 1$. Sie können untersuchen, wie weit die berechneten Lösungen beim Euler- bzw. Heun-Verfahren in Abhängigkeit von der Schrittzahl von der exakten Lösung abweicht.)

- an der DGL dritter Ordnung

$$y''' = 2xy'y'' + 2y^2y'$$

mit den Anfangswerten

$$y(1) = 1, \quad y'(1) = -1, \quad y''(1) = 2$$

und berechnen Sie einen Näherungswert für $y(2)$ mit 10, 100, 1000 und 10000 Schritten mit dem Euler- und dem Heun-Verfahren.

Die exakte Lösung ist $y(x) = \frac{1}{x}$. Berechnen Sie die Abweichungen der Euler- bzw. Heun-Näherungen von der exakten Lösung $y(2) = 0.5$ für die verschiedenen Schrittweiten. Erkennen Sie eine Systematik?

5. Aufgabe (optional, passend zu Bonus-e-Test 6b)

Häufig gibt es in einer Differenzialgleichung Parameter, deren Werte unbekannt sind.

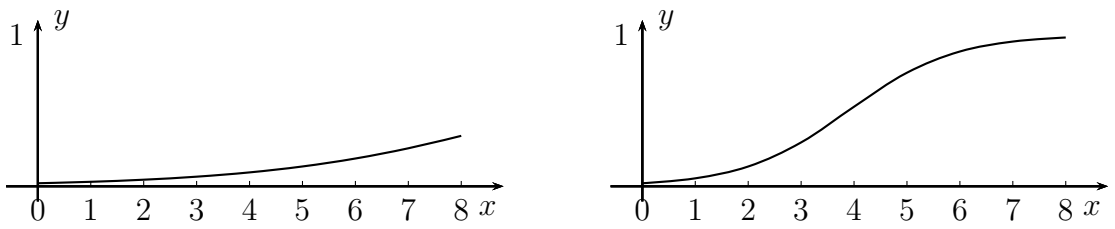
Beispiel:

Um die Ausbreitung einer Epidemie zu modellieren, kann man die Differenzialgleichung

$$y' = \lambda \cdot y \cdot (1 - y)$$

betrachten. Der Wert von y entspricht dabei dem Anteil der Bevölkerung, der mit der Krankheit infiziert ist. Die Anzahl der Neuansteckungen (also die Änderung y') ist dann proportional zum einen zur Anzahl der Infizierten (das entspricht dem Faktor y auf der rechten Seite) und zum anderen zum Anteil derer, die bisher noch nicht infiziert sind (das entspricht dem Faktor $(1 - y)$ auf der rechten Seite). Der Proportionalitätsfaktor λ wird davon beeinflusst, wie ansteckend die Krankheit ist, und wieviele Kontakte es in der Bevölkerung gibt.

(Man kann λ auffassen als Produkt eines Ansteckungsfaktors und eines Kontaktfaktors; ist beispielsweise der Ansteckungsfaktor durch medizinische Untersuchungen bekannt, bleibt aber weiter der Kontaktfaktor unbekannt; durch Kontaktbeschränkungen kann er aber beeinflusst werden. Im Folgenden soll nur λ als Ganzes weiter betrachtet werden.)



Im Bild links ist der Lösungsverlauf zu $\lambda = 0.4$, im Bild rechts zu $\lambda = 1.0$, jeweils mit Anfangsbedingung $y(0) = 0.02$ dargestellt.

Man kann nun das Verhalten des Systems, das durch die Differenzialgleichung beschrieben wird, beobachten und daraus Rückschlüsse auf den oder die Parameter ziehen.

Beispiel (Fortsetzung):

Mit den gemeldeten Krankheitsfällen gibt es eine Möglichkeit den Parameter λ entsprechend der gemeldeten Fälle anzupassen.

Man kann die entsprechende Differenzialgleichung für bestimmte Parameterwerte lösen und dann die Parameterwerte bestimmen, die eine Bestapproximation der berechneten Werte mit den gemessenen Werten liefert. Als Bestapproximation kann man die *least-squares*-Anpassung (vgl. Aufgabe 4 von Praktikum 1) betrachten. Man erhält so eine Funktion $d(\vec{p})$ in Abhängigkeit des Parametervektors $\vec{p}$.

Die Optimierung der Parameter kann dann beispielsweise durch das Gradientenverfahren (Praktikum 1) oder das Newton-Verfahren angewendet auf den Gradienten (vgl. Praktikum 2) durchgeführt werden. Dabei ist es essenziell, dass die dabei benötigten Ableitungen numerisch berechnet werden, denn zu gegebenen Parameterwerten $\vec{p}$ kann man

$d(\vec{p})$ berechnen (nämlich durch numerische Lösung der Differenzialgleichung und dann Berechnung der *least-squares*-Summe), aber eine analytische Berechnung der Ableitung nach einem der Parameterwerte ist quasi unmöglich.

Beispiel (Fortsetzung):

Beim Bonus-e-Test 6b, Aufgabe 3, werden ihnen sieben Meldedaten vorgegeben.

Gesucht ist nun die DGL-Lösung, die möglichst gut mit den gemeldeten Werten übereinstimmt.

Neben dem DGL-Parameter λ ist auch die genaue Anfangsbedingung y_0 der DGL unbekannt. Wären die Meldedaten exakt, so wäre $y_0 = m_0$. Man kann nun einen Parameter u einführen, der die (relative) Ungenauigkeit der Meldung m_0 beschreibt, d.h., man setzt $y_0 = u \cdot m_0$.

Gesucht sind nun die Parameter λ und u, so dass für die Lösung $y_{\lambda,u}$ des Anfangswertproblems

$$y' = \lambda \cdot y \cdot (1 - y) \qquad \text{mit} \quad y(0) = u \cdot m_0$$

der Wert von

$$d(\lambda, u) = \sum_{k=1}^{N} (y_{\lambda,u}(k) - m(k))^2$$

minimal wird.

- Auf meiner Internetseite finden Sie eine Excel-Datei „DGL_Anpassung.xls“.

  Laden Sie sich die Datei herunter und tragen Sie die Melde-Daten aus Ihrem Bonus-e-Test 6b, Aufgabe 3, in die gelb hinterlegten Zellen ein.

  In der Datei können Sie mit Schiebereglern die Parameter λ und u einstellen.

  Angezeigt werden die Melde-Daten und die Lösung der DGL; berechnet wird die Summe der quadratischen Abweichungen $d(\lambda, u)$.

- Berechnen Sie mit Ihren Methoden aus Praktikum 1 und 2 optimale Parameterwerte, z.B. ausgehend vom Startpunkt $(1, 1)$. Eine Möglichkeit ist, direkt das Newton-Verfahren zur Nullstellenbestimmung des Gradienten zu probieren. Alternativ kann man zunächst ein paar Schritte mit dem Gradientenverfahren machen und dann auf das Newton-Verfahren wechseln. Das ist stabiler aber langsamer. Das Gradientenverfahren alleine ist sehr langsam.

  Sie können die zu optimierende Funktion selbst programmieren oder die c++-Funktionen aus „DGL_Anpassung.cpp“ auf meiner Internetseite nutzen. Ähnlich wie bei Aufgabe 4 von Praktikum 1 erwähnt, müssen meine Funktionen ggf. noch auf Ihr Projekt angepasst werden.

- Stellen Sie die Schieberegler der Excel-Datei auf die berechneten Werte ein, und überzeugen Sie sich, dass eine gute Approximation erreicht wird.

Ausgehend von den berechneten Parametern könnten Sie nun berechnen, in welcher Woche mit der größten Zunahme der Infizierten zu rechnen ist. Kennt man den Anteil der Neuinfizierten, die intensivmedizinisch betreut werden müssen, und die Dauer der intensivmedizinischen Betreuung, kann man ausrechnen, ob unser Gesundheitssystem überlastet werden wird.

Weitere Hinweise:

- Die angegebene Differenzialgleichung ist sogar analytisch lösbar, Stichwort „logistische Differenzialgleichung“. Die Bestimmung der optimalen Parameter ist aber weiterhin nur numerisch möglich.
- Eine etwas detailliertere Modellierung einer Epidemie bietet das sogenannte SIR-Modell. Dies ist ein Differenzialgleichungssystem, dessen Lösung nicht analytisch angebbar ist.

Wenn Sie weiter interessiert sind: Diese Dinge erläutere ich ausführlich in meinen Videos „Mathematik im Umfeld von Corona“ (s. Wochenplan).