Einführung in Computational Engineering
Grundlagen der Modellierung und Simulation

Dr. Arne Nägel

TECHNISCHE UNIVERSITÄT DARMSTADT

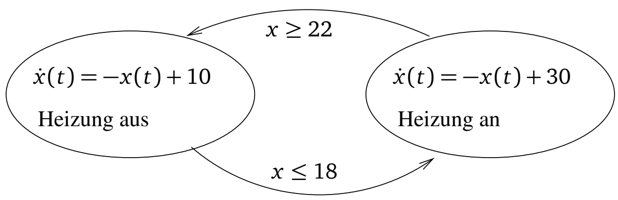
Wintersemester 2012/2013
Lösungsvorschlag der 9. Übung

Aufgabe 1 Dynamische Systeme mit Unstetigkeiten (10 Punkte)

Betrachtet werden soll ein Heizungssystem. Zur Regulierung der Raumtemperatur werde ein Thermostat so eingestellt, dass eine Heizung sich einschaltet, sobald die Temperatur unter 18°C fällt. Sobald die Raumtemperatur den Wert 22°C übersteigt, wird die Beheizung abgeschaltet.

Bei konstanter Außentemperatur lässt sich das dynamische Verhalten des Temperaturverlaufs in grober Näherung wie folgt beschreiben:

- Ist die Heizung aus, gilt für die Temperatur $x(t)$ die Beschreibung $\dot{x}(t) = -x(t) + 10$.
- Ist die Heizung in Betrieb, gilt für die Raumtemperatur $\dot{x}(t) = -x(t) + 30$.



Bearbeiten Sie nun folgende Aufgaben:

a) Zeigen Sie: Die Lösung des Anfangswertproblems $\dot{x}(t) = a(t)x(t) + b(t)$, $x(t_0) = x_0$ ist durch

$$x(t) = e^{f(t)} \left[\int_{t_0}^{t} b(s) e^{-f(s)}\, ds + x_0 \right], \quad f(t) := \int_{t_0}^{t} a(s) ds$$

gegeben. Wie lauten jeweils die Lösungen der Differentialgleichungen in den beiden diskreten Zuständen („Heizung an“ / „Heizung aus“) bei gegebenen Anfangswert $x(t_0) = x_0$?

b) Skizzieren Sie den zeitlichen Verlauf der Temperatur $x(t)$ bei gegebenem Anfangswert $x(t_0) =$ 15°C.

c) Geben Sie die Schaltfunktionen an, mit deren Hilfe Beginn und Ende der Beheizung in der Simulation der Temperatur berücksichtigt werden kann.

d) Betrachten Sie die Situation $x(10) = 18.2$°C bei abgeschalteter Heizung. Führen Sie einen Simulationsschritt der Länge $\Delta t = 0.2$ mit dem expliziten Euler-Verfahren durch.

e) Bestimmen Sie den aus (d) folgenden Schaltzeitpunkt und geben Sie Startwerte für den anschließenden Iterationsschritt der weiteren Simulation an.

Lösungsvorschlag

a) Die Behauptung ergibt sich über die Produktregel mittels Differenzieren (1 Punkt). Betrachte dann den Fall $a(t) = -1, b(s) = b$ (2 Punkte):

$$\begin{aligned}
f(t) &= \int_{t_0}^{t} -1\,ds = t_0 - t \\
x(t) &= e^{t_0-t}\left[\int_{t_0}^{t} be^{-(t_0-s)}\,ds + x_0\right] \\
&= e^{t_0-t}\left[b\int_{t_0}^{t} e^{s-t_0}\,ds + x_0\right] \\
&= e^{t_0-t}\left[b\int_{t_0}^{t} e^{s-t_0}\,ds + x_0\right] \\
&= e^{t_0-t}\left[b(e^{t-t_0}-1) + x_0\right] \\
&= b + (x_0 - b)e^{t_0-t}
\end{aligned}$$

Insbesondere folgt (1 Punkt): Heizung aus: $x(t) = 10 + (x_0 - 10)e^{t_0-t}$, Heizung an: $x(t) = 30 + (x_0 - 30)e^{t_0-t}$.

b) Starte bei $x = 15$. Darstellung in Abbildung 1A) (2 Punkte).

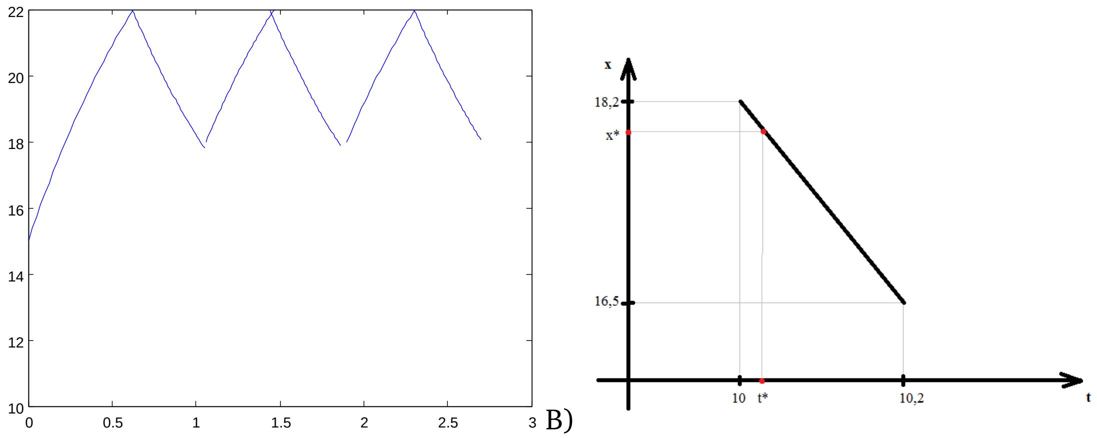
A) 

Abbildung 1: A) Skizze des Schaltens für Startwert für $x_0 = 15$, $t_0 = 0$. B) Bestimmung der Nullstelle der Schaltfunktion in Aufgabenteil e).

c) Mögliche Schaltfunktionen sind (1 Punkt):

$$q_1(x) := x - 22,\ q_2(x) := x - 18$$

d) Starte bei $x(10) = 18,2$, mit Zustand „Heizung aus“. Nächster Schritt mit explizitem Euler: $x(10,2) = 18,2 + 0,2(-18,2 + 10) = 16,56$ (1 Punkt).

e) Wegen $16,56 < 18$ erfolgt zwischen $t_k = 10$ $(x_k = 18.2)$ und $t_{k+1} = 10,2$ $(x_{k+1} = 16.56)$ ein Schaltvorgang. Gesucht ist der Zeitpunkt $t^*_{k+1} = t_k + h^*$ für den die Relation

$$18 = x^*_{k+1} = x_k + h^*(-x_k + 10) = 18,2 + h^*(-18,2 + 10)$$

gilt. Auflösen ergibt $h^* = (18 - 18,2)/(10 - 18,2) = 1/41 \approx 0,02439$. Für Zustand „Heizung an“ starte man also idealerweise mit $(t^*_{k+1}, x^*_{k+1})^T = (10,02439, 18)^T$ (2 Punkte).

Aufgabe 2 Schrittweitensteuerung (10 Punkte)

Es sei ein beliebiges Einschrittverfahren der Ordnung p gegeben. Der folgende Algorithmus realisiert dann ein (einfaches) Verfahren zur adaptive Schrittweitensteuerung. Wie üblich sei x_k die k. Iterierte und h_k die Schrittweite zum Zeitpunkt t_k.

- Berechne die nächste Iterierte $x_{k+1}^{h_k}$ durch einen Verfahrensschritt mit der Schrittweite h_k.
- Berechne $x_{k+1}^{h_k/2}$ durch zwei Schritte des gleichen Verfahrens mit der Schrittweite $h_k/2$.
- Schätze den lokalen Fehler mittels der Größe

$$\varepsilon := \frac{|x_{k+1}^{h_k/2} - x_{k+1}^{h_k}|}{1-2^{-p}}.$$

- Falls ε größer als eine vorgegebene Toleranz δ, wiederhole den Schritt mit einer neuen Schrittweite. Die neue Schrittweite ergebe sich dabei aus der alten gemäß der Vorschrift $h_{k,neu} = \left(\frac{\delta}{\varepsilon}\, h_{k,alt}^{p+1}\right)^{1/p}$.
- Ansonsten akzeptiere den Schritt und setze $x_{k+1} = x_{k+1}^{h_k/2}$, $h_{k+1} = h_k$ und $t_{k+1} = t_k + h_k$.

a) Betrachten Sie das Anfangswertproblem

$$\dot{x}(t) = -200 \cdot t \cdot x^2(t),\ x(-3) = 1/901$$

auf dem Intervall $[-3, 0]$. Zeigen Sie, dass dessen Lösung durch $x(t) = 1/(1+100t^2)$ gegeben ist. Skizzieren Sie diese Lösung. Warum ist die Verwendung eines adaptiven Verfahrens sinnvoll?

b) Führen Sie nun den o.g. Algorithmus für das Heun-Verfahren 2. Ordnung aus. Berechnen Sie dazu für das gegebene Anfangswertproblem die erste Iterierte x_1. Verwenden Sie $\delta = 10^{-9}$ und die Startschrittweite $h_0 = 1/10$. Rechnen Sie mit ausreichend vielen Nachkommastellen. Sie dürfen die Rechnung nach einer Wiederholung des Schrittes abbrechen. Welche Schrittweite $h_{0,\text{neu}}$ ergibt sich? Wie groß ist der Fehler zwischen numerischer und analytischer Lösung, $|x_1 - x(t_1)|$, in jedem Zwischenschritt?

Lösungsvorschlag

a) Für $x(t) = 1/(1+100t^2)$ gilt

$$\dot{x}(t) = -1/(1+100t^2)^2\, 200t = -200\, t\, x^2(t)$$

sowie

$$x(-3) = 1/(1+900) = 1/901.$$

In Abbildung 2 ist die Funktion skizziert. Man beachte, dass man zunächst große Schritte h machen kann, jedoch immer feiner werden sollte, je näher man sich an den Ursprung annähert. (2 Punkte)

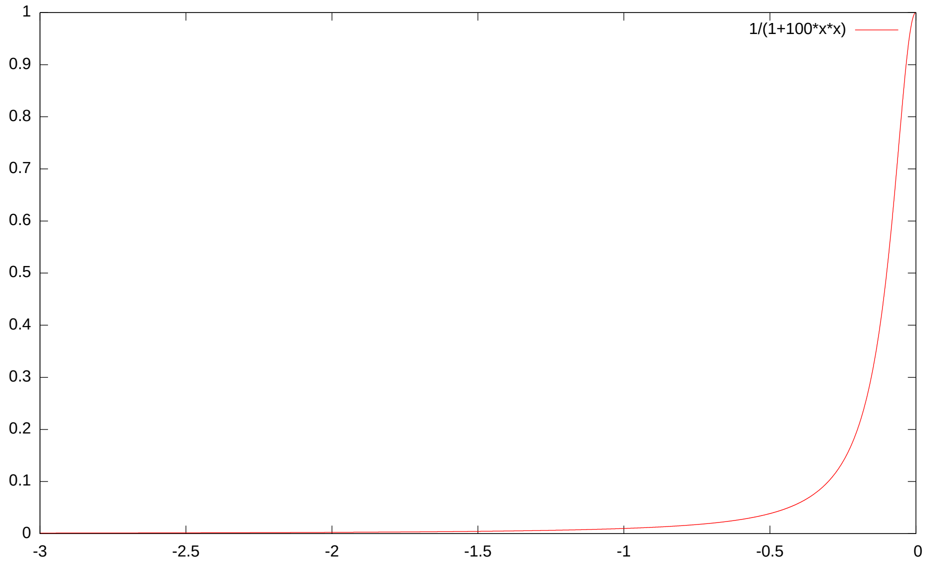


Abbildung 2: Darstellung der Lösung des AWP in Aufgabe 2.

b) Die Berechnungsvorschrift für das Heun-Verfahren 2. Ordnung lautet

$$
\begin{aligned}
s_1 &= f(t_k, x_k)\,, \\
s_2 &= f(t_k + h_k, x_k + h_k s_1)\,, \\
x_{k+1} &= x_k + \frac{h_k}{2}(s_1 + s_2)\,.
\end{aligned}
$$

Mit dem gegebenen Anfangswert $x(-3) = \frac{1}{901}$ und der Startschrittweite $h_0 = 0.1$ ergibt sich die Iterierte $x_1^{h_0}$ zu

$$
\begin{aligned}
s_1 &= f\left(-3, \frac{1}{901}\right) \\
&= -200 \cdot (-3) \cdot \left(\frac{1}{901}\right)^2 = 7.39097389631203 \cdot 10^{-4}\,, \\
s_2 &= f\left(-3 + 0.1, \frac{1}{901} + \frac{1}{10} s_1\right) \\
&= f(-2.9, 0.001171356116411) \\
&= -200 \cdot (-2.9) \cdot 0.001171356116411^2 = 8.12784859455225 \cdot 10^{-4}\,, \\
x_1^{h_0} &= \frac{1}{910} + \frac{1}{2 \cdot 10}(7.39097389631203 + 8.12784859455225) \cdot 10^{-4} \\
&= 0.00118747202588384\,. \qquad \textit{(1 Punkt)}
\end{aligned}
$$

Der erste Iterationsschritt mit halber Schrittweite $h_{0h} = \frac{h_0}{2} = 20^{-1}$ und dem Anfangswert $x(-3) = \frac{1}{910}$ resultiert in die Iterierte $x_{1,1}^{h_{0h}}$ mit

$$\begin{aligned}
s_1 &= f\left(-3, \frac{1}{901}\right) = 7.39097389631203 \cdot 10^{-4} , \\
s_2 &= f\left(-3 + \frac{1}{20}, \frac{1}{901} + \frac{1}{20} \cdot 7.39097389631203 \cdot 10^{-4}\right) \\
&= f(-2.95, 0.00114683278291108) \\
&= -200 \cdot (-2.95) \cdot 0.001135128607656^2 = 7.75983004856152 \cdot 10^{-4} , \\
x_{1,1}^{h_{0h}} &= \frac{1}{910} + \frac{1}{2 \cdot 20}(7.39097389631203 + 7.75983004856152) \cdot 10^{-4} \\
&= 0.00114775492329171 . \qquad \textit{(1 Punkt)}
\end{aligned}$$

Im zweiten Iterationsschritt mit halber Schrittweite $h_{0h} = \frac{h_0}{2} = 20^{-1}$ wird mit dem Anfangswert $x(-2.95) = x_{1,1}^{h_{0h}}$ die Iterierte $x_{1,2}^{h_{0h}}$ zu

$$\begin{aligned}
s_1 &= f(-2.95, 0.00114775492329171) \\
&= -200 \cdot (-2.95) \cdot 0.00114775492329171^2 = 7.77231404724807 \cdot 10^{-4} , \\
s_2 &= f\left(-2.95 + \frac{1}{20}, 0.00114775492329171 + \frac{1}{20} \cdot 7.77231404724807 \cdot 10^{-4}\right) \\
&= f(-2.9, 0.00118661649352795) \\
&= -200 \cdot (-2.9) \cdot 0.00118661649352795^2 = 8.16674047573285 \cdot 10^{-4} , \\
x_{1,2}^{h_{0h}} &= 0.001136020478378 + \frac{1}{2 \cdot 20}(7.77231404724807 + 8.16674047573285) \cdot 10^{-4} \\
&= 0.00118760255959916 \qquad \textit{(1 Punkt)}
\end{aligned}$$

bestimmt. Der lokale Fehler ε kann nun mit $p = 2$ zu

$$\varepsilon \approx \left| \frac{x_{1,2}^{h_{0h}} - x_1^{h_0}}{1 - 2^{-p}} \right| = 1.7404495375288 \cdot 10^{-7} > \delta = 10^{-9} \qquad \textit{(1 Punkt)}$$

berechnet werden. Der lokale Fehler ε ist damit größer als die vorgegebene Toleranz δ. Die neue Schrittweite h_1 ergibt sich zu

$$h_1 = \left(\frac{\delta}{|\varepsilon|} h_0^3\right)^{\frac{1}{p}} = 2.39700688818416 \cdot 10^{-3} . \qquad \textit{(1 Punkt)}$$

Die erneute Berechnung des Schrittes mit der neuen Schrittweite h_1 und $h_{1h} = h_1/2$ resultiert in

$$\begin{aligned}
x_1^{h_1} &= 0.00111165165511158 , \\
x_{1,1}^{h_{1h}} &= 0.00111076425423418 , \\
x_{1,2}^{h_{1h}} &= 0.00111165165680686 , \\
\varepsilon &\approx 2.26037851630556 \cdot 10^{-12} < 10^{-9} . \qquad \textit{(1 Punkt)}
\end{aligned}$$

Die Fehler lauten (2 Punkte):

$$
\begin{aligned}
|x_1^{h_0} - x(-3+h_0)| &= 1.7643 \cdot 10^{-7} > 10^{-9} \\
|x_{1,2}^{h_{0h}} - x(-3+h_0)| &= 4.58965 \cdot 10^{-8} > 10^{-9} \\
|x_1^{h_1} - x(-3+h_1)| &= 2.26106 \cdot 10^{-12} < 10^{-9} \\
|x_{1,2}^{h_{1h}} - x(-3+h_1)| &= 5.65773 \cdot 10^{-12} < 10^{-9}
\end{aligned}
$$

Programmieraufgabe P4 Numerische Differenzieren (20 Punkte)

Die Jacobimatrix beschreibt die erste Ableitung einer Funktion $\boldsymbol{f}$, mit $\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^m$. In der Vorlesung wurde der Vorwärtsdifferenzenquotient vorgestellt, um Ableitungen zu approximieren. Für die Funktion $\boldsymbol{f}$, einen Punkt $\boldsymbol{x} \in \mathbb{R}^n$ und eine Schrittweite $\boldsymbol{\delta} \in \mathbb{R}^n$, mit $\delta_j > 0$, für $j \in \{1, \ldots, n\}$, gilt demnach:

$$\frac{\partial f_i}{\partial x_j} \approx \frac{f_i(\boldsymbol{x} + \delta_j \boldsymbol{e}_j) - f_i(\boldsymbol{x})}{\delta_j}. \tag{1}$$

Der Vektor $\boldsymbol{e}_j \in \mathbb{R}^n$ bezeichne den j-ten Einheitsvektor.

a) Erstellen Sie eine Funktion `numdiff` für Matlab, die mit Hilfe der gegebenen Formel für eine beliebige Funktion $\boldsymbol{f}: \mathbb{R}^n \to \mathbb{R}^m$, ein $\boldsymbol{x} \in \mathbb{R}^n$ und eine Schrittweite $\boldsymbol{\delta} \in \mathbb{R}^n$ eine Approximation der Jacobimatrix liefert. Ist die Funktion $\boldsymbol{f}$ durch eine Matlab-Funktion mit dem Namen `fun.m` gegeben, soll der Aufruf der von Ihnen erstellten Funktion mit

```
[f, J] = numdiff(@fun,x, delta)
```

den Funktionswert `f` und eine Approximation der Jacobimatrix `J` als Ausgabe liefern.

b) Schreiben Sie eine Funktion `diffplot(@fun, @jac, x)`, welche für `f`$: \mathbb{R} \to \mathbb{R}$ (d.h. $n = m = 1$) und gegebenes $x \in \mathbb{R}$ die Abweichungen aus der Berechnung mit Ihren bisherigen Funktionen in Abhängigkeit von δ grafisch darstellt. Wie in Aufgabenteil a) gibt `@fun` die Funktionsvorschrift an. Die entsprechende Ableitung sei durch `@jac` vorgegeben. Beide sollen im Punkt `x` ausgewertet werden.

   Berechnen Sie dazu die Abweichungen $||J^{(num)} - J^{(ex)}||$ zwischen der numerischen Ableitung $J^{(num)}$ und der exakten Ableitung $J^{(num)}$ für verschiedene Werte $\delta \in \{10^{(-i)} | i = 0, 1, \ldots, 18\}$. Stellen Sie diese in einer aussagekräftigen Grafik dar. Achten Sie dabei insbesondere auf sinnvolle Achsen-Skalierung und Beschriftungen.

c) Testen Sie Ihr Programm anhand der Testvorlage in der Datei `P4FiniteDifferenz.zip` im Lernportal. In dieser sind auch die Funktionsrümpfe vorgegeben.

Hinweise zur Programmieraufgabe

- Für die Abgabe der Programmieraufgabe komprimieren Sie die bearbeiteten Dateien in einem unverschlüsselten `zip`-Archiv, das Sie mit dem ersten Buchstaben des Vornamens `V`, dem ersten Buchstaben des Nachnamens `N` und den letzten beiden Ziffern der Matrikelnummer `mm` jedes Gruppenmitglieds nach dem Muster `VNmm_VNmm_VNmm.zip` benennen. Eine korrekte Benennung könnte beispielsweise `AB12_CD34_EF56.zip` sein. Laden Sie das `zip`-Archiv im Lernportal Informatik über den zur Programmieraufgabe gehörenden Datei-Upload hoch.

Lösungsvorschlag