Einführung in Computational Engineering
Grundlagen der Modellierung und Simulation
Dr. Arne Nägel

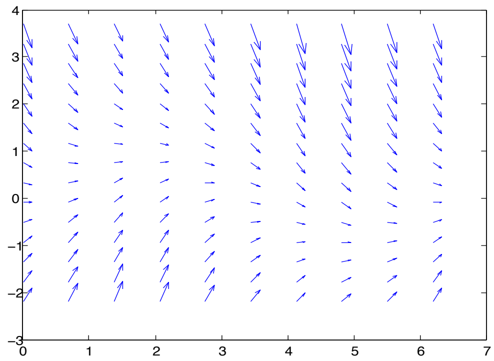
TECHNISCHE UNIVERSITÄT DARMSTADT

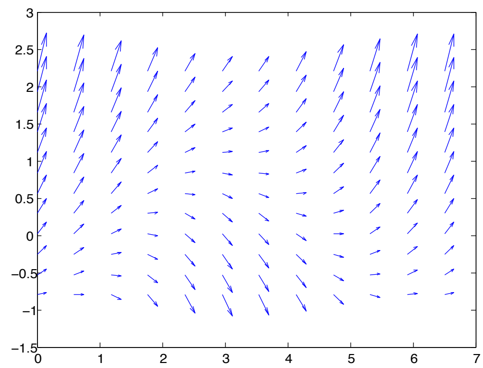
Wintersemester 2012/2013
Lösungsvorschlag der 3. Übung

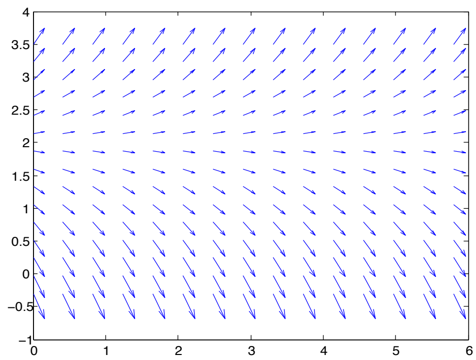
Aufgabe 1 Differentialgleichungssysteme (10 Punkte)

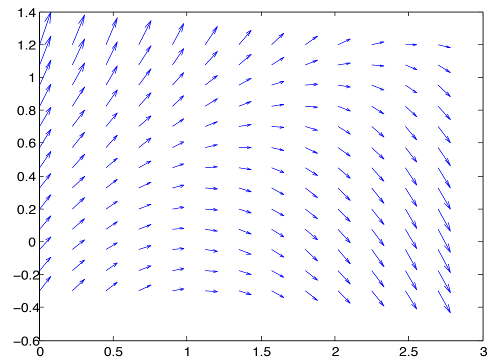
a) Ordnen Sie folgende Richtungsfelder und Differentialgleichungen einander zu:

$$
\begin{array}{llll}
(1) & \dot{x}(t) = x(t) - 2 & (2) & \dot{x}(t) = x(t)^2 - t + 1 \\
(3) & \dot{x}(t) = x(t) + \cos(t) & (4) & \dot{x}(t) + x(t) = \sin(t)
\end{array}
$$

(A)


(B)


(C)


(D)


b) Transformieren Sie die Differentialgleichung mit Anfangswert

$$\dot{x}(t) = 2x^2(t) + \cos(t), \; x(0) = 3\,.$$

in eine autonomes Differentialgleichungssystem mit Anfangswerten.

c) Zeigen Sie, dass die Gleichungen

$$a(t) \; = \; 2\cos(t) + 6\sin(t) - 2 - 6\,t, \qquad b(t) \; = \; -2\cos(t) + 2\sin(t) + 3$$

(mit den Zustandsvariablen: a und b) die exakte Lösung des folgenden Anfangswertproblems ist:

$$\ddot{a} + 2\dot{a} + 5b = 3, \quad \dot{a} + 2b = \dot{b},$$
$$a(0) = 0, \quad \dot{a}(0) = 0, \quad b(0) = 1\ .$$

d) Transformieren Sie die Differentialgleichungen und Anfangswerte aus Aufgabenteil c) in ein Differentialgleichungssystem der Form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}$, $\mathbf{x}(0) = \mathbf{x}_0$. Wählen Sie dazu einen passenden Zustandsvektor $\mathbf{x}(t)$.

Lösungsvorschlag

a) $(A) \longleftrightarrow (4)$, $(B) \longleftrightarrow (3)$, $(C) \longleftrightarrow (1)$, $(D) \longleftrightarrow (2)$ (4 Punkte).

b) Hinzunahme einer weiteren Gleichung liefert System von 2 DGLn: Definiere $x_1(t) := x(t), x_2(t) := t$ und erhalte damit

$$\begin{aligned} \dot{x}_1 &= 2x_1^2 + \cos x_2 \\ \dot{x}_2 &= 1 \end{aligned}$$

mit den Anfangswerten $x_1(0) = 3, x_2(0) = 0$ (2 Punkte).

c) Berechnen der Ableitungen:

$$\begin{aligned} \dot{a} &= -2\sin(t) + 6\cos(t) - 6 \\ \ddot{a} &= -2\cos(t) - 6\sin(t) \\ \dot{b} &= 2\sin(t) + 2cos(t) \end{aligned}$$

Einsetzen und Ausrechnen liefert Behauptung (2 Punkte).

d) Das System lautet wie folgt (2 Punkte):

$$\begin{pmatrix} \dot{a} \\ \ddot{a} \\ \dot{b} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & -5 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} a \\ \dot{a} \\ b \end{pmatrix} + \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}, \qquad \begin{pmatrix} a(0) \\ \dot{a}(0) \\ b(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Aufgabe 2 Stabilität (10 Punkte)

Wir betrachten die folgenden Gleichungen als leicht gestörte Variante des in der Vorlesung betrachteten *Linearen Schwingers*:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= x - x^3 - ry \end{aligned}$$

Dabei ist r eine reelle Zahl mit $0 < r < \sqrt{8}$.

a) Bestimmen Sie die möglichen Ruhelagen dieses Systems.

b) Führen Sie für diese drei Punkte nun eine Linearisierung um die Ruhelage durch. Bestimmen Sie dazu die Jacobi-Matrix und Ihre Eigenwerte. Sind die Ruhelagen stabil oder instabil?

c) Der Stabilitätsbegriff soll nun noch etwas theoretischer betrachtet werden. Es sei $x = (a + bi)$, $y = (c + di) \in \mathbb{C}$ mit $a, b, c, d \in \mathbb{R}$. Zeigen Sie:

$$xe^y = e^c \left\{(a\cos d - b\sin d) + i(a\sin d + b\cos d)\right\}$$

(*Bemerkung: Verwenden Sie hierzu die Definition der komplexen Multiplikation sowie die Eulersche Formel* $e^{i\omega} = \cos\omega + i\sin\omega$.)

d) Bewerten Sie unter Verwendung dieses Zusammenhangs das Verhalten der Funktion

$$z : \mathbb{R} \to \mathbb{C}, t \mapsto z(t) = xe^{yt}$$

für $t \to \infty$. Wann bleibt diese beschränkt? Wann geht diese gegen 0?

Lösungsvorschlag

a) Wir setzen die rechte Seite gleich 0 und erhalten: $y = 0$, $x \in 0, -1, 1$, Daher sind $\mathbf{p}_0 = (0,0)^T, \mathbf{p}_1 = (0,-1)^T, \mathbf{p}_2 = (0,1)^T$ die Ruhelagen.

b) Durch elementweises differenzieren ergibt sich die Jacobi-Matrix:

$$\mathbf{J} := \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & 1 \\ 1-3x^2 & -r \end{pmatrix}.$$

Aus dem Ansatz

$$0 = |\lambda Id - \mathbf{J}| = \det \begin{pmatrix} \lambda & -1 \\ 3x^2-1 & \lambda+r \end{pmatrix} = \lambda(\lambda+r) + (3x^2-1)$$

errechnet man die Eigenwerte. Für $\mathbf{p}_0$ gilt:

$$\lambda(\lambda+r) - 1 = 0 \Leftrightarrow \lambda = \frac{-r}{2} \pm \sqrt{\left(\frac{r}{2}\right)^2 + 1}$$

Somit ist einer der Eigenwerte positiv; die Ruhelage im Punkt $\mathbf{p}_0$ ist folglich instabil.

Für $\mathbf{p}_1, \mathbf{p}_2$:

$$\lambda(\lambda+r) + 2 = 0 \Leftrightarrow \lambda = \frac{-r}{2} \pm i\sqrt{2 - \left(\frac{r}{2}\right)^2}$$

Somit hat haben beide Eigenwerte negativen Realteil; die Ruhelage in den Punkten $\mathbf{p}_1, \mathbf{p}_2$ sind stabil (4 Punkte).

c) Ausrechnen (2 Punkte):

$$e^y = e^{c+id} = e^c e^{id} = e^c(\cos d + i \sin d) = e^c \cos d + i e^c \sin d$$

und dann

$$x\, e^y = (a + ib)e^y = e^c\left((a\cos d - b \sin d) + i(a \sin d + b \cos d)\right)$$

d) Mit c) erhält man

$$x e^{yt} = e^{ct}\left\{(a\cos d - b\sin d) + i(a\sin d + b\cos d)\right\}$$

und daher unter Verwendung der Dreiecksungleichung:

$$\begin{aligned}
|xe^{yt}| &= |e^{ct}|\,|(a\cos d - b\sin d) + i(a\sin d + b\cos d)| \\
&\leq |e^{ct}|\,(|a\cos d - b\sin d| + |a\sin d + b\cos d|) \\
&\leq |e^{ct}|\,(|a\cos d| + |b\sin d| + |a\sin d| + |b\cos d|) \\
&\leq |e^{ct}|\,(|a| + |b| + |a| + |b|) \\
&\leq |e^{ct}| 2(|a| + |b|)
\end{aligned}$$

Damit sieht man, dass die Funktion gegen 0 konvergiert, falls $c < 0$. Umgekehrt gilt

$$|xe^{yt}| \geq |e^{ct}|\,||a\cos d - b\sin d| - |a\sin d + b\cos d||$$

Das asymptotische Verhalten für $t \to \infty$ hängt nur von c ab. Falls $c < 0$, so strebt $xe^{yt} \to 0$ (2 Punkte).

Programmieraufgabe P1 Ereignissimulation mit Java (20 Punkte)

Auf der Kursseite im Lernportal Informatik können Sie das Paket *P1DEVS.zip* herunterladen. Es enthält ein Gerüst von Beispielklassen zur Durchführung einer diskreten Ereignissimulation (*discrete event simulation, DEVS*). Um das Gerüst nutzen zu können, entpacken Sie die Datei in einen Ordner Ihrer Wahl. Die Quelltexte finden sich im Ordner zum Paket `edu.eice2012`. Die Dateien darin können Sie mit einem Java-Editor Ihrer Wahl (z.B. Eclipse, NetBeans , ...) importieren.

a) Implementieren Sie ein Verfahren um das in der Vorlesung vorgestellte und auf den ersten Übungszetteln behandelte Modell einer Warteschlange (mit einem Server) zu implementieren. Erweitern Sie dazu die von `DEVS` abgeleitete Klasse `QueueDEVS` um Methoden zur Ereignisbehandlung. Darin sind für die Ankunfts- und Verlassensroutinen lediglich Funktionsrümpfe vorhanden. Implementieren Sie diese!

b) Ein ausführbares Hauptprogramm findet sich in der Datei `DEVSMain.java`. Darin können Sie die Zufallszahlengeneratoren für *Arrival* (`randA`) und *Departure* `randD` auswählen sowie die Länge der Simulation einstellen (`maxtime`). Die Erzeugung von Zufallszahlen wird im Interface `RandomNumberGenerator` festgelegt. Dieses enthält nur eine Funktion `nextDoubleValue()`, welche bei jedem Aufruf neue Zufallszahlen erzeugt.

   Konkrete Implementierungen sind die Klassen `MyExponential`, `MyNormalDist`. Diese enthalten die Zufallszahlen vom ersten Übungsblatt. Testen Sie im Rahmen einer ersten Validierung, ob Ihr Programm die richtigen Ergebnisse erzielt.

c) Außerdem soll das Programm wichtige statistische Informationen liefern, die während der Simulation gesammelt werden. Vervollständigen Sie dazu die Methode `statistics`. Diese wird am Ende der Simulation aufgerufen und soll dann Schatzer für die folgenden Informationen auf der Konsole ausgeben:

   - die mittlere Serverauslastung ϱ sowie
   - die mittlere Anzahl der Aufträge (Kunden) im System L.

d) In der Veranstaltung wurde ferner hergeleitet, dass für die Wahrscheinlichkeit, dass sich k Aufträge (Kunden) im System aufhalten gilt:

$$Pr[L(t) = k] = (1 - \varrho)\rho^k$$

   Diese Verteilung sollen Sie in einer Simulation schätzen. Messen Sie dazu die Zeitdauern T_k, in denen sich k Aufträge (Kunden) im System aufhalten. Erzeugen Sie dann aus diesen Werten und den Quotienten $p_k = T_k/T$ in eine solche Verteilung (*Bemerkung: Für die Simulationsdauer T gilt $T = \sum_{k=0}^{\infty} T_k$. Für die unendliche Summe beachten Sie: Zu vorgegebenem T gibt es in der Praxis ein $K \in \mathbb{N}$, so dass $T_k = 0$ für alle $k > K$.*).

   Testen Sie Ihr Programm für die Werte $\lambda = 5, \mu = 6$ also $\varrho = 5/6 < 1$ für 20000 Simulationszeiteinheiten. Wie schon bei Aufgabenteil b) ist auch hierzu bereits ein entsprechender Abschnitt in `DEVSMain.java` vorbereitet.

Hinweise:

- Ihr Code sollte gut kommentiert und lesbar sein. Laden Sie den kompletten Ordner `DEVS/` als komprimierte *.zip*-Datei mit Nennung von Namen und Matrikelnummer der maximal drei Gruppenmitglieder im Lernportal Informatik hoch. Aus technischen Gründen muss *jedes* Gruppenmitglied eine Abgabe einreichen.

- Fügen Sie Ihrer Abgabe zudem zwei Dateien `validierungB.txt` `validierungD.txt` bei. Darin sollen die Programmausgaben zu den in den Teilaufgaben b) und d) beschriebenen Tests gespeichert werden.

- Bitte benennen Sie das *.zip*-Archiv aller Gruppenmitglieder identisch und eindeutig von anderen Gruppen unterscheidbar nach folgender Konvention: `VNmm_VNmm_VNmm.zip`. Dabei steht `V` jeweils für den ersten Buchstaben des Vornamens, `N` für den ersten Buchstaben des Nachnamens und `mm` für die letzten beiden Ziffern der Matrikelnummer eines Gruppenmitglieds.

Lösungsvorschlag