

“Erste Schritte mit DSTOOL”

Jörg Härterich
Freie Universität Berlin

22. Mai 2006

Diese Kurzanleitung soll dazu dienen, sich in ca. 20 Minuten mit den wichtigsten Möglichkeiten von DSTOOL bekannt zu machen. DSTOOL lässt sich auf den Linux-Poolrechnern und auf den wenigen Solaris-Rechnern durch den Befehl `dstool` starten. Es erscheint oben links ein Fenster, das im Folgenden “Hauptfenster” heißt.

1 Voreingestellte Dynamische Systeme

DSTOOL kennt schon etwa zehn Dynamische Systeme, die man im Hauptfenster mit dem Menü **Models** auswählen kann.

Wähle unter “Standard Vector Fields” das Modell “Planar Cubic Vector Field” aus. Dieses Modell simuliert die Differentialgleichung

$$\begin{aligned}\dot{x} &= y \\ \dot{y} &= -(x^3 + rx^2 + nx + m) + y(b - x^2)\end{aligned}$$

mit voreingestellten Parametern $r = n = m = b = \frac{1}{10}$. Wir wollen als erstes einzelne Trajektorien bzw. ein Phasenportrait zeichnen. Wähle im Hauptfenster aus dem Menü **Panels** den Punkt **Two-D** aus. Es erscheint ein neues Fenster, in dem die Ergebnisse der Simulation dargestellt werden.

Nun können wir die ersten Trajektorien berechnen. Fahre mit der Maus in das Fenster und drücke nacheinander die linke und die mittlere Maustaste. Durch die linke Taste wird der Anfangspunkt festgelegt, beim Drücken der mittleren Taste startet die Simulation. Experimentiere mit anderen Anfangsbedingungen, bis Du eine Vorstellung davon hast, wie das Phasenportrait aussieht.

Wähle mit der linken Maustaste im Hauptfenster unter **Panels** zunächst **Orbits** und dann **Selected**. Es erscheinen zwei weitere Fenster. Im Fenster “Selected Point” kann man von Hand die Anfangsbedingung für die Simulation und den Wert der Parameter vorgeben. Ändere den Parameterwert von $b = 0.1$ auf $b = 0.5$. Mit den Knöpfen **Forward** und **Backward** im “Orbit”-Fenster startet man dann die Simulation in positiver, bzw. negativer Zeitrichtung. Mit der Maus zeichnet man Rückwärtsorbits, indem man zuerst die rechte und danach die mittlere Maustaste drückt.

Im “Orbit”-Fenster kann man noch weitere Einstellungen vornehmen, beispielsweise mit “Step Size” die Größe der Zeitschritte vorgeben oder unter “Stop” die Anzahl der Schritte. Will man das numerische Verfahren ändern, bietet sich unter **Propagation** die Möglichkeit dazu.

Lösche mit **Clear All** die bisher gezeichneten Trajektorien. Setze dann zunächst “Stop=2500”, “Stepsize=0.05” und starte die Simulation mit Anfangsbedingungen $x = y = 3$. Nun wollen wir das Bild dieser Trajektorie ausdrucken: Wähle dazu im Menü “Options” den Punkt “Print...” aus. Es empfiehlt sich gelegentlich, Achsenbezeichnungen zu ändern oder Parameterwerte etc. im Titel des Fensters anzugeben, damit man später noch weiß, was man eigentlich ausgedruckt hat.

Um die Lösung $x(t)$ zu zeichnen, wählt man mittels der Menüs im linken, oberen Eck des **2D-View**-Fensters **time** und **x** als Achsen aus.

2 Kontinuierliche Systeme: Volterra-Lotka-Modelle

Meist wollen wir nicht die in DSTOOL vorhandenen Gleichungen benutzen, sondern selbst definierte Dynamische Systeme simulieren.

Starte dazu DSTOOL und wähle im Menü **Model** zunächst “Parser”, dann “Parsed Dynamical System...”.

```
# The Lorenz system
x' = sigma ( y - x )
y' = rho x - y - x z
z' = -beta z + x y
```

```
INITIAL sigma 10.0 rho 28.0 beta 2.66666666
RANGE x -30 30 y -30 30 z -20 50
```

Dieser Text enthält Informationen über die Differentialgleichung, Anfangsbedingungen (“INITIAL”), sowie den darzustellenden Bereich (“RANGE”). Hier ist das die Lorenz-Gleichung

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

mit den Parameterwerten $\sigma = 10$, $\rho = 28$ und $\beta = \frac{8}{3}$.
Ersetze den Text durch:

```
# A Lotka-Volterra Model
x' = x - 2*x*y
y' = -3*y + 0.5*x*y
INITIAL x 1.2 y 0.4
RANGE x 0 3 y 0 3
```

Drücke dann den Knopf **Build**, um das neue Modell einzulesen.

Öffne jetzt das “2D-View”-Fenster und berechne per Mausklick einige Trajektorien. Vergrößere dann den Bereich auf $0 \leq x \leq 10$, $0 \leq y \leq 15$. Anschließend **Refresh** drücken, erst dann ist die neue Größe eingestellt! Ausschnittsvergrößerungen sind auch möglich, indem man mit gedrückter Shift-Taste mit der Maus einen neuen Ausschnitt markiert. Berechne eine weitere Trajektorie. Sind die Trajektorien wirklich alle periodisch?

Wie groß kann man den Zeitschritt “Step Size” wählen, bevor man mit dem bloßen Auge eine Abweichung von der periodischen Bahn schon bei einem Umlauf sehen kann?

Trage dann im “View”-Fenster horizontal die Zeit t (“time”) und vertikal $x(t)$ auf.

Hinweise zum DSTOOL-Parser:

1. Mit den Knöpfen **Vector Field** bzw. **Mapping** gibt man an, ob es sich um ein kontinuierliches oder ein diskretes dynamisches System handelt.
Achtung: Auch bei diskreten Systemen steht im Parser $x' = \dots$ etc.
2. Der Parser versteht nur Gleichungen 1. Ordnung, Gleichungen höherer Ordnung müssen als System 1. Ordnung umgeschrieben werden,
3. ebenso müssen nicht-autonome Gleichungen in ein autonomes System umgeschrieben werden.
4. Der Parser versteht **ausschließlich** die folgenden Befehle:

```
Konstanten: PI, pi, E
Operatoren: +, -, *, /, % (mod), ^
Mathematische Funktionen: sin, cos, tan, asin, acos, atan, sinh, cosh,
                        tanh, log, ln, exp, abs, sqrt
```

5. Mit **INITIAL** gibt man die voreingestellten Anfangsbedingungen, mit **RANGE** den voreingestellten Darstellungsbereich an. Beides kann man natürlich später ändern. Mit **PERIODIC** kann man einzelne Variablen periodisch machen, um Systeme auf der Kreislinie oder auf dem Torus zu simulieren.
PERIODIC x 0 1 y 0 1 macht die Variablen x und y beide 1-periodisch.
Hilfsfunktionen kann man *nach* der Definition des dynamischen Systems eingeben, wie die Gleichungen selbst, jedoch ohne Strich, also z.B. $\text{Ham} = 0.5*(x*x + y*y)$.
6. Das neue dynamische System wird durch den Button **Build** geladen.
7. Speichern lassen sich die eigenen dynamischen Systeme mittels **Save...** als Textdatei.