Informatik mit Matlab – Labor 10

Thema des Labors: Physik, Differentialgleichungen (DGL)

Vorübung: Versuchen Sie die in der Vorlesung besprochenen Beispiele zum Laufen zu bringen. Die zu gehörigen M-Files finden Sie hier in diesem Verzeichnis.

Aufgabe 1: LoeseDgl1(x0, tm, c)

Schreiben Sie die MATLAB-Funktion **LoeseDgl1** und eine zugehörige DGL-Funktion **DglFnc1**, die folgende Differentialgleichung 1. Ordnung im Intervall [0,*tm*] mit Hilfe des Solvers ode45 löst und die Kurve $x(t)$ grafisch anzeigt:

$$dx/dt + 2\,c\,x^3 - \sin(\pi\,t) = 0$$

Der Anfangswert $x(0) = x0$, die Länge des Zeitintervalls *tm* un der Parameter *c* werden beim Aufruf der Funktion als Parameter übergeben.
Testen Sie Ihre Funktion für verschiedene Werte von *x0*, *tm* und c, z.B.

```
>> LoeseDgl1( 1, 5, 2 )
```

Aufgabe 2: LoeseDgl2(x0, v0, tm)

Schreiben Sie die MATLAB-Funktion **LoeseDgl2** und eine zugehörige DGL-Funktion **DglFnc2**, die folgende Differentialgleichung 2. Ordnung im Intervall [0,*tm*] mit Hilfe des Solvers ode45 löst und die Kurve $x(t)$ grafisch anzeigt:

$$d^2x \,/\, d\,t^2 = 3 \,/\, x^3 - 5 * dx/dt$$

Die Anfangswerte $x(0) = x0$ und $dx/dt\ (0) = v0$ und die Länge des Zeitintervalls *tm* werden beim Aufruf der Funktion als Parameter übergeben.
Testen Sie Ihre Funktion für verschiedene Werte von *x0*, *v0* und *tm*, z.B.

```
>> LoeseDgl2( 1, 3, 2 )
```

Aufgabe 3: laguerre(n, tm, x0, v0)

Schreiben Sie die Funktion **laguerre**, die folgende DGL zur Berechnung der Laguerre-Polynome vom Grad *n* löst:

$$\frac{d^2x}{dt^2} + \frac{(1-t)}{t}\frac{dx}{dt} + \frac{n}{t}\cdot x = 0$$

für den übergebenen *n*-Wert, für das Zeitintervall von *eps* bis *tm* (startet erst ab Matlab-Genauigkeit, wegen der Sigularität der DGL bei $t = 0$) und für die übergebenen Anfangswerte *x0* und *v0*.

Mit den berechneten Werten plottet die Funktion das *t*-*x*-Diagramm.

Beispielaufruf: `>> laguerre( 2, 5, 1, -2 )`

Zusatzaufgabe: [t,x] = EulerVerfahren2(DglFun, tspan, x0, h, par)

Schreiben Sie die MATLAB-Funktion **EulerVerfahren2**, die analog der in der Vorlesung besprochenen Funktion EulerVerfahren als DGL-Solver arbeitet, aber diesmal mit dem verbesserten Euler-Verfahren, also der Näherung

$$z(t_n) = z_{n-1} + h/2 * D(t_{n-1}, z_{n-1}) + h/2 * D(t_n, z_{n-1}+h*D(t_{n-1},z_{n-1})), \quad \text{mit } t_n = t_0 + n*h$$

Testen Sie die Funktion zum Beispiel mit den Funktionen *D1* und *D2* aus der Vorlesung oder der Funktion *DglFnc1* aus Aufgabe 1, also z.B.

```
>> [t,x] = EulerVerfahren2( @D1,[0,5],0,0.1,[] ); )
>> te = 0:0.1:5;
>> ze = te.^2;
>> plot( te,ze, '-b', t, x, ':r' )
```

Zusatzaufgabe: spirale(freq)

Die Dynamik eines 3-dimensionalen Systems sei durch folgendes DGL-System beschrieben:

$$\frac{d^2x}{dt^2} = -\omega^2 \cdot x$$

$$\frac{d^2y}{dt^2} = -\omega^2 \cdot y$$

$$\frac{d^2z}{dt^2} = 0$$

Schreiben Sie die zugehörige DGL-Funktion **Fun_Spirale**(*t*, *x*) und erstellen Sie eine Funktion **spirale**(f), die den Wert der Umlauffrequenz *f* als Übergabeparameter hat. Die Winkelgeschwindigkeit *w* = 2*pi**f* .

Die Funktion spirale löst dieses DGL-System mit dem Solver ode45 für das Zeitintervall [0,4] und verwendet hierfür die Anfangsbedingungen:

$x(0) = 2$, $dx/dt\,(0) = 0$, $y(0) = 0$, $dy/dt\,(0) = 10$, $z(0) = 0$, $dz/dt\,(0) = 1$

Die Lösung wird am Ende als 3D-Kurve grafisch dargestellt.
Testen Sie Ihre Funktion für verschiedene Werte von *f*, z.B. *f* = 1, *f* = 1.5, ...

```
>> spirale( 1 )
```

Erweiterung: Erstellen Sie ein Movie der Bewegung, vgl. Dokument *Inf2_MovieBall.pdf*.

Zusatzaufgabe: Randwertproblem

Erzeugen und testen Sie die in der Vorlesung besprochene Randwertaufgabe, zu der die folgenden Funktionen im Public-Bereich gehören:

- solve_BVP   Aufruf des Solvers bvp4c
- Fun_BVP   Definition der DGL
- BC_BVP   Definition der Randbedingungen

Aufruf: `>> solve_BVP`