

Informatik mit Matlab – Labor 11

Thema des Labors: Wiederholung und Vorbereitung zur Klausur.

Hinweis: Sie sollten versuchen, diese Aufgaben alleine zu lösen, ohne zu Beginn auf die Musterlösungen zu schauen. als Hilfsmittel nur das Buch und evtl. eine zweiseitige Formelsammlung. Wenn Ihnen das bei der Mehrzahl der Aufgaben gelingt, dann sind Sie gut für die Klausur vorbereitet. Üben Sie die Aufgaben zuhause auch rein schriftlich, ohne Computer.

Sie werden aber aus Zeitgründen nicht alle 9 Aufgaben im Labor schaffen. In der Klausur gibt es voraussichtlich 5 Aufgaben.

Aufgabe 1: `[t,y] = fourierReihe(A, nf, f, SR, tm)`

Schreiben Sie die Funktion **fourierReihe**, die eine Fourier-Reihe aus sin-Tönen zu (evtl. auch ungeradzahligem) Vielfachen der Frequenz f erzeugt, mit einer Zeitdauer von tm Sekunden und der Sample-Rate SR . Die Werte der Vielfachen der Frequenzen stehen im Vektor nf . Die Amplituden $A(n)$ der sin-Beiträge zu den Frequenzen $nf(n) * f$ stehen im Vektor A . Die Funktion gibt die berechneten Zeit- und Signal-Werte zurück.

Möglicher Aufruf:

```
>> [t,y] = fourierReihe([1,1/3,1/5],[1,3,5], 440, 40000, 0.1 );
>> plot( t,y )
>> axis( [ 0 10/440 -1 1 ] )
```

Aufgabe 2: `num = bild100(fn_in, fn_out)`

Schreiben Sie die MATLAB-Funktion **bild100**. Die Funktion liest die Daten des Bildes zum Namen fn_in ein und berechnet die Anzahl num der Pixel des Bildes.

Dann bestimmt sie alle Pixel, für die sowohl der R-Wert < 100 als auch der B-Wert < 100 sind. Bei all diesen Pixeln wird die RGB-Farbe auf Schwarz gesetzt.

Das abgeänderte Bild wird unter dem Namen fn_out abgespeichert. Zum Schluss werden sowohl das Original-Bild als auch das geänderte Bild auf dem Bildschirm angezeigt.

Möglicher Aufruf:

```
>> num = bild100( 'Katze.bmp', 'KS.bmp' )
```

Aufgabe 3: `num = calcRoots(pol)`

Schreiben Sie die MATLAB-Funktion **calcRoots**, die zum übergebenen Koeffizienten-Vektor pol die Anzahl num aller reellen Nullstellen des zugehörigen Polynoms berechnet und zurückgibt.

Mögliche Aufrufe:

```
>> num = calcRoots( [1,0] )
>> num = calcRoots( [1,-1,-2,1] )
```

Aufgabe 4: [a,w0,phi0] = fit2(t, phi)

Schreiben Sie die MATLAB-Funktion **fit2**. Die übergebenen Vektoren t und phi gehören zu einer Messreihe einer beschleunigten Kreisbewegung, für die die Formel gilt:

$$phi(t) = a/2 t^2 + w_0 t + phi_0$$

Bestimmen Sie durch einen geeigneten Polynom-Fit der t - und phi -Werte die Parameter a , w_0 und phi_0 . Zeichnen Sie die Fit-Kurve und markieren Sie in der Zeichnung auf die übergebenen Messpunkte, z.B. durch ein Quadrat.

Möglicher Aufruf:

```
>> t = [0 2 4 6 8 10];
>> phi = [ 0.01 0.81 1.19 1.21 0.79 0.02];
>> [a,w0,phi0] = fit2( t, phi )
```

Aufgabe 5: x_max = dgl(k, tm, x0, v0)

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

$$d^2x / dt^2 = -k * dx/dt$$

Die Anfangswerte $x(0) = x_0$ und $dx/dt(0) = v_0$, der Parameter k und die Länge des Zeitintervalls tm werden beim Aufruf der Funktion als Parameter übergeben. Außerdem wird der Maximalwert von x berechnet und als x_max von der Funktion zurückgegeben.

Testen Sie Ihre Funktion für verschiedene Werte von x_0 , v_0 , k und tm , z.B.

```
>> x_max = dgl( 1, 3, 1, 1 )
```

Aufgabe 6: [mw,sf] = stdFehler(xFile)

Schreiben Sie die MATLAB-Funktion **stdFehler**, die den Mittelwert mw und den Standardfehler sf der Reihe x direkt durch Summationen berechnet, bzw. alternativ unter Verwendung der Funktionen *mean* bzw. *std*.

Die Daten der Reihe x stehen zeilenweise untereinander in der Textdatei, deren Namen in dem Parameter $xFile$ übergeben wird.

Mittelwert	Standardabweichung	Standardfehler
$\bar{g} = \frac{\sum_{i=1}^n g_i}{n}$	$\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^n (g_i - \bar{g})^2}{n-1}}$	$\Delta g = \frac{\sigma_{n-1}}{\sqrt{n}}$

Testen Sie Ihre Funktion für verschiedene x -Reihen z.B.

```
>> [mw,sf] = stdFehler( 'xData.txt' )
```

Aufgabe 7: WochenStunden

Schreiben Sie die MATLAB-Klasse **WochenStunden**, die die Arbeitsstunden eines Mitarbeiters für eine Kalenderwoche zählt,

mit den Eigenschaften

- *name* (Name des Mitarbeiters)
- *kw* (Kalenderwoche)
- *summe* (Summe der Arbeitsstunden, initialisiert mit 0)

und den Methoden

- *Konstruktor* mit den Eingangsparametern *name* und *kw*
- *arbeitszeit* mit Eingangsparameter *anz* (Anzahl der am Tag geleisteten Stunden)

Erzeugen Sie im Command-Window ein Mitarbeiter-Objekt *m1* und rufen Sie die Methode *arbeitszeit* zweimal auf, einmal mit 8.5 und einmal mit 8 Stunden.

Lassen Sie sich die Summe der Arbeitszeiten vom Objekt *m1* zurückgeben.

Aufgabe 8: Invertierung von Wörtern

Schreiben Sie eine Funktion: `w = invertiereWort(wort)`. Die Funktion bekommt ein Wort übergeben und gibt den Text in umgekehrter Reihenfolge zurück.

Beispiel: `>> w = invertiereWort('Baum')`
`w = 'muaB'`

Aufgabe 9: Taylor-Reihe

Schreiben Sie eine Funktion `[y,N] = taylor(x, schranke)` zur Berechnung der Funktion $y = f(x)$ mittels der Taylorreihe:

$$y = \sum_{n=0}^{nMax} (-1)^n (n+1) x^n$$

Übergeben wird das Argument *x* und der Maximalfehler *schranke*. Bestimmen Sie die minimale Anzahl der Reihenglieder *N*, bis der aktuelle Summen-Term $(-1)^n (n+1) x^n$ dem Betrag nach kleiner wird als der Wert von *schranke*. Die damit berechnete Näherung für *y* und die Anzahl *N* sind die Rückgabewerte der Funktion.