

Informatik mit Matlab – Labor 9

Thema des Labors: Mathematik, Polynome, Polynom-Fit, Statistik, Numerische Verfahren

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: `checkParabel(a, b, c, x1, x2)`

Schreiben Sie die Funktion `checkParabel` für eine Kurvendiskussion der Parabel

$$y = a x^2 + b x + c, \quad \text{im } x\text{-Intervall } [x1, x2], \text{ mit 100 Stützstellen}$$

Die Funktion berechnet die Nullstellen und Extrema der Funktion, mit Angabe des Typs (Minima / Maxima) und dem Check auf deren Existenz auf der reellen Achse (Funktion *isreal*). Zeichnen Sie anschließend die Funktion und markieren Sie die berechneten Punkte.

Aufgabe 2: `[mw,sf] = temperaturen()`

Legen Sie durch die folgenden Temperatur-Messwerte zu den Zeiten 1, 2, 3, ... eine Ausgleichsgerade und stellen Sie die Messwerte und die Gerade grafisch dar, analog *ausgleichsgerade.m*:

$$16.2, 15.8, 17.6, 19.1, 19.2, 21.0, 20.9, 22.5, 23.1, 23.7$$

Berechnen Sie außerdem den Mittelwert *mw* der Temperaturen und den Standardfehler *sf* = Standardabweichung / Wurzel(Anzahl der Messwerte).

Aufgabe 3: `n = strahlung(T, U)`

Für die Strahlungsleistung P eines heißen Körpers (Temperatur T), gemessen mit einem Thermoelement, gilt eine polynomiale Abhängigkeit: $P = k \cdot T^n$. Die abgelesene Thermospannung U ist proportional zu P . In einem Versuch werden T und U gemessen, um aus der Gleichung

$$\log(U) = n \cdot \log(T) + c$$

den Koeffizienten n zu erhalten. Schreiben Sie die Funktion `n = strahlung(T,U)`, die aus den übergebenen T - und U -Werten (Vektoren, erzeugt z.B. über „*strahlungVal.m*“) mittels einer Ausgleichsgeraden den Wert des Exponenten n bestimmt und diesen als Funktionswert zurückgibt (der Logarithmus wird in MATLAB mit der Funktion *log* berechnet).

Berechnen Sie dann für alle Temperaturen T die Werte

$$U1 = U0 \cdot \left(\frac{T}{T0} \right)^n, \quad \text{mit: } U0 = U(1), T0 = T(1)$$

Plotten Sie in einer Grafik die Werte (T,U) und die berechnete Kurve $(T,U1)$.

Aufgabe 4: $[v, x_0, diff] = \text{interpolExcel}(name)$

Erstellen Sie die Funktion **interpolExcel**. In der Excel-Datei mit dem Namen *name* sind Werte einer Ortsmessung $x(t)$ gespeichert, in der Spalte A die t-Werte und in der Spalte B die x-Werte. Für die x-Werte erwartet man den funktionalen Zusammenhang:

$$x(t) = v t + x_0$$

Lesen Sie die t- und x-Daten aus der Excel-Datei aus. Interpolieren Sie die x-Werte mit einem Polynom geeigneten Grades und bestimmen Sie dadurch die Werte für v und x_0 , die als Funktionswerte zurückgegeben werden. Berechnen Sie den Mittelwert *diff* des Betrags der Abweichung der Messwerte von der Funktion. Testen die Funktion mit der Datei *xData.xls*.

Aufgabe 5: statistik()

Erzeugen Sie mit Hilfe des MATLAB-Aufrufs `w=rand(100,1)` ein Datenfeld mit 100 gleichmäßig im Intervall $[0,1]$ verteilten Werten. Berechnen Sie dazu den Mittelwert und die Standardabweichung, analog *datenauswertung.m*.

Wiederholen Sie die Auswertung für den Aufruf `w=randn(100,1)`, durch den normal-verteilte Daten um den Ursprung erzeugt werden.

Plotten Sie die Datenpunkte über ihrer Nummer und zeichnen Sie den Mittelwert und die Standardabweichung in die Grafik ein.

Wiederholen die Tests mit 10000 Zufallswerten.

Zusatzaufgabe: $f = \text{trapezRegel}(fun, a, b)$

Erstellen Sie die Funktion **trapezRegel**, die ähnlich arbeitet wie die in der Vorlesung besprochene Funktion *rechteckRegel*, nur dass beim Integral über die Funktion *fun* zur Flächenberechnung nicht die Rechteckregel, sondern die Trapezregel verwendet wird. Das Trapez über einem Intervall ist definiert durch die Werte der Funktion *fun* an den jeweils beiden Endpunkten der Teil-Intervalle, also $Trapezfläche = (fun(x_n) + fun(x_{n+1})) * d / 2$. Zur Zerlegung des Intervalls $[a,b]$ können Sie beispielsweise 1000 Teil-Intervalle verwenden.

Testen Sie die Integration mittels *trapezRegel* an mehreren, unterschiedlichen Funktionen.

Zusatzaufgabe: RandomWalk

Erzeugen Sie die Klasse *RandomWalk* mit den Eigenschaften: *nZ*, *nS*, *mult* = 10 und *start*.

Der Konstruktor übergibt die Werte für *nZ* und *nS* (Zahl der Zeilen bzw. Spalten) und berechnet den Startpunkt *start* als Mittelpunkt, also $nZ/2$ und $nS/2$.

Zusätzlich gibt es die Methode *walk(obj, steps)*, die einen Zufallsweg für die Anzahl *steps* von Schritten berechnet und zeichnet. Mit Hilfe des MATLAB-Aufrufs `d=rand(1,2)` erzeugt man pro Schritt zwei Zufallszahlen im Intervall $[0,1]$. Um gleich oft Wege in positiver und negativer Richtung zu bekommen, zieht man 0.5 von den Zufallszahlen ab und erhält dadurch Werte im Intervall $[-0.5, +0.5]$.

Der nächste Schritt wird berechnet aus $Vorgänger + mult * (d - 0.5)$. Der Parameter *mult* (z.B. mit dem Wert 5) definiert dabei eine Streckung des Wegs.

Zeichnen Sie die einzelnen Wege, jeweils als Linie vom Vorgänger zum Nachfolger und wählen Sie den Zeichnungs-Ausschnitt (axis) von 0 bis nS bzw. 0 bis nZ .

Zusatzaufgabe: polynom()

Erstellen Sie die Funktion *polynom*, die eine Kurvendiskussion für Polynome vom beliebigen Grad durchführt, in Anlehnung an die Funktion in *kurvendisk.m*.

Als erstes fragt die Funktion den Anwender nach dem Grad des Polynoms, dann nach den Werten der einzelnen Koeffizienten [Anzahl = grad + 1: $a(n), \dots, a(1), a(0)$].

Mit diesen Daten werden die Nullstellen und die Extrema des Polynoms berechnet und ausgegeben. Zum Schluss wird das Polynom für ein gewünschtes Intervall gezeichnet. In der Grafik werden auch die Nullstellen und Extrema markiert.

Achten Sie darauf, dass nur reelle Nullstellen wirkliche Nullstellen der Funktion sind. Sie müssen deshalb mit der Funktion *isreal* die zurückgegebenen Nullstellen testen.