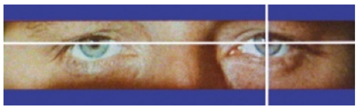
Labor 3 & 4: Optimierung und symbolisches Rechnen

Matlab-Inhalte:

- syms: deklaration von symbolischen Variablen
- diff: Differentiation einer Funktion
- solve: eine Gleichung nach einer Variablen auflösen
- subs: Ersetzen von symbolischen Variablen durch Zahlen
- find
- fprintf: formatierte Ausgabe an stdout

Heute fangen wir mit der Aufgabe an. Alle beschriebenen Schritte im Anschluss an die Aufgabe dienen der Lösung dieser.

Quizzes-Aufgabe Tatort



Eine Pathologin möchte den Todeszeitpunkt eines Mordopfers feststellen. Um 12:36 Uhr misst sie die Temperatur des Opfers, welche 26.8°C beträgt. Nach dem Newtonschen Abkühlungsgesetz ist die Abkühlung $T'(t)$ eines Körpers proportional zur Differenz von Körpertemperatur $T(t)$ und Außentemperatur a. Es gilt demnach

$$T'(t) = \alpha\,(T(t) - a)\,. \tag{1}$$

Aus Erfahrung weiß die Pathologin, dass die Proportionalitätskonstante α den Wert -10^{-2} hat. Die Außentemperatur a beträgt 21°C. Wir wollen herausfinden wann der Mord stattgefunden hat.

(a) Überzeugen Sie sich davon, dass die Funktion

$$T(t) = (T_0 - a)\,\mathrm{e}^{\alpha\,(t-t_0)} + a$$

in die Gleichung (1) "passt". Dabei ist $T_0 = T(t_0) = 36$ die Temperatur zum Todeszeitpunkt t_0. t ist die Zeit in Minuten, wobei der Tag bei $t = 0$ startet.

(b) Berechnen Sie t_0, so dass die Funktion zur Temperaturmessung passt. Verwenden Sie dabei gerne das symbolische Rechnen von Matlab. Müssen Sie aber nicht.

(c) Die Pathologin will auf Nummer Sicher gehen und misst um 14:08 Uhr abermals die Temperatur. Es sind dann 22.4°C. Passt das noch zum Todeszeitpunkt?

(d) Die Antwort sollte "Nein" lauten, denn Messungen beinhalten meist ein Rauschen, bzw. Messungenauigkeiten. Berechnen Sie den Todeszeitpunkt t_0, der optimal ist für die beiden Messungen durch Optimierung des Quadratfehlers

$$E(t_0) = (\text{Messung}_1 - T(\text{Zeitpunkt}_1))^2 + (\text{Messung}_2 - T(\text{Zeitpunkt}_2))^2 \,.$$

Wir gehen Schritt für Schritt voran:

Ableiten und Optimieren mit Matlab

(a) Wir überprüfen zunächst ob die angegebene Funktion T in die angegebene Gleichung (1) passt:

Für die Werte $\alpha = 10^{-2}$, $T_0 = 36$ und $a = 21$ berechnen wir die Ableitung der Funktion T, wobei wir die Variablen t und t_0 symbolisch behandeln:

```
syms t t0

T(t) = (T0-a)*exp(-alpha*(t-t0))+a;
dT(t) = diff(T(t),t);

% Ueberpruefung der Gleichung T'(t) = alpha*(T(t)-a)

dT(t)-alpha*(T(t)-a);

% Das Ergebnis sollte - na klar - 0 sein.
% Manchmal ist es gut, den Ausdruck zu vereinfachen:

simplify(dT(t)-alpha*(T(t)-a););
```

src/Blatt_C_syms.m

(b) Gut. Die Funktion passt also. Ploten wir sie mal über einen ganzen Tag für - sagen wir - den Todeszeitpunkt 12:00 Uhr. Nur als Beispiel. Eine symbolische Variabel ersetzen wir durch Substitution mit einer Zahl. So:

```
>Tb(t) = subs(T(t),t0,12*60);
```

Dann haben wir eine neue Funktion Tb (Wir könnten auch T überschreiben, brauchen die aber noch.), die nur noch von der Zeit t abhängt. Sie gibt uns die Temperaturkurve in jeder Minute, wobei der Mord um genau 12 uhr stattgefunden hat. Der Temperaturabfall beginnt ja genau dann. Wir brauchen also nur den Plot ab $T(t) = 36$ und wir tun einfach so als wüssten wir nicht, um welche Uhrzeit das ist.

```
Tb(t) = subs(T(t),t0,12*60);
tt = linspace(I(1),I(2),DIM); % I=[0,24*60];
TT = Tb(tt);
IndT = find(TT<=36); % Indexliste f\"ur die TT<=36 Grad gilt
plot(tt(IndT),TT(IndT));
```

src/Blatt_C_plotT.m

(c) Sieht ja soweit vernünftig aus. Jetzt wollen wir aber den Todeszeitpunkt nicht vorgeben sondern berechnen. Wir suchen also t_0 so dass die erste Messung von 26.8°C um 12:36 Uhr (12*60+36 = 756 Minuten) passt. Wir müssen

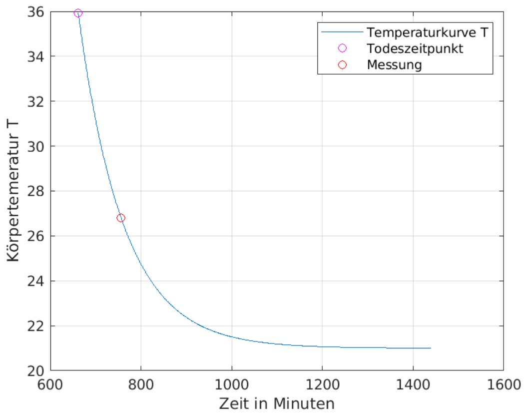
$$26.8 = T(756) = (36-21)\,\mathrm{e}^{-10^{-2}\,(756-t_0)} + 21$$

nach t_0 auflösen. Das kann man schick auf dem Papier machen oder - auch schick - mit Matlab:

```
t sol1=solve(T(756)==26.8,t0);

% Das setzen wir nun statt der 12 Uhr in die Funktion ein:

Tb1(t) = subs(T(t),t0,t_sol1);

% Speichern fuer den Plot die Funktionswerte,
% aber nur die unter 36 Grad:

TT1 = Tb1(tt);
IndT1 = find(TT1<=36);
tzp1 = tt(IndT1(1));                % Zeit als Uhrzeit:
hour = floor(tzp1/60);
minute = floor((tzp1/60-hour)*60);  % formatierte Ausgabe:
fprintf("Der Todeszeitpunkt laut erster Messung ist um
\%02d:\%02d Uhr\textbackslash n",hour,minute);
```

src/Blatt_C_Sol1.m

Dann ploten wir das den Graphen samt Messung und Todeszeitpunkt. Sollte ungefähr so aussehen:



(d) Um wieviel Uhr fand der Mord statt wenn Sie von der Messung 22.4°C um 14:08 Uhr ausgehen? Sie müssen nur die Schritte mit den anderen Messwerten wiederholen. Am besten Sie speichern die Messdaten und -zeiten in ein entsprechendes Feld ab. Etwa

`>Mt = [12*60+36,14*60+8];` % beide Zeiten
`>MC = [26.8,22.4];` % beide Messungen

Das macht das ganze modularer und übersichtlicher. Der Todeszeitpunkt ist nun ein anderer. Das ist nicht verwirrend. So eine Messung ist ja meist nicht superexakt mit Rauschen drauf und so. Viele Messungen sind immer gut. Die Frage ist nun welche Kurve bezüglich dieser beiden Messungen die optimale ist.

(e) Der Fehler den wir betrachten wollen ist der Abstand zwischen tatsächlichen Daten und dem Graphen, quadriert und aufsummiert

$$E(t_0) = (MC(1) - T(Mt(1)))^2 + (MC(2) - T(Mt(2)))^2$$

ergibt eine Quadrat-Fehler-Funktion, die nur noch von t_0 abhängt.

`>E(t0) = (MC(1) - T(Mt(1)))`^{2}`+(MC(2) - T(Mt(2)))`^{2}`;`

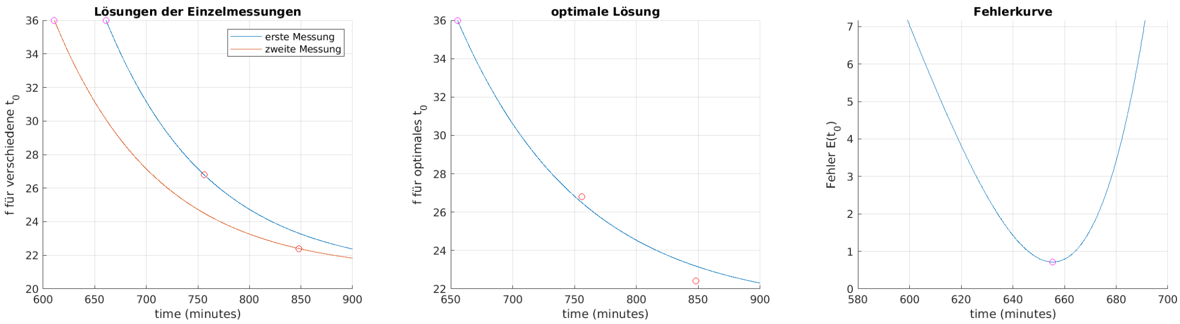
Wir wollen ermitteln, für welchen Wert von t_0 der Fehler E minimal ist. Dazu suchen wir die Nullstelle der ersten Ableitung:

```
% Fehlerfunktion
E(t0) = (MC(1) - T(Mt(1)))^2 +(MC(2) - T(Mt(2)))^2;

% erste Ableitung
dE(t0) = diff(E(t0),t0);

% Nullstelle der ersten Ableitung
t0_sol = solve(dE(t0) ==0,t0);
```

src/Blatt_C_E_Opt.m

Tataa. Jetzt machen wir schöne plots und schauen uns das ganze mal an.

So könnte/sollte Ihr Ergebnis aussehen:



Alle drei plots in ein Fenster positionieren Sie mit `subplot`:

```
% Fenster fuer drei plots in einer Zeile
% Position 1 (links)
subplot(1,3,1)

  plot(.....);

% Position 2 (in der Mitte)
subplot(1,3,2)

  plot(.....);

% Position 3 (rechts)
subplot(1,3,3)
```

src/Blatt_C_subplot.m