ÜBUNG: Welche Funktionsparameter lassen sich linear bestimmen ?

Die Parameter sind dann mit einem lin. Gleichungs-system lösbar, wenn die Funktion als Skalarprodukt beschrieben werden kann, mit den zu bestimmenden Größen im 2. Vektor.

a) $y = a \cdot x^4 + b \cdot \sin(\frac{\pi}{2} \cdot x^3) + c$

$$y = \underbrace{\begin{pmatrix} x^4 & \sin(\frac{\pi}{2} \cdot x^3) & 1 \end{pmatrix}}_{\text{mit den Meßwerten, werden hieraus Zahlen}} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Wenn mind. 3 Messungen $(x_1, y_1), \ldots, (x_3, y_3)$ gegeben sind, dann können die Parameter mit einem <u>lin. Gleichungssystem</u> bestimmt werden!

b) $y = a \cdot x^b + 2 \cdot \sin(cx) + 1$

→ nicht als Vektorprodukt mit Parametervektor beschreibbar

→ Mit mind. 3 Messungen erhält man ein <u>nichtlineares</u> <u>Gleichungssystem</u>

c) $$y = \frac{ax^2 + bx + c}{dx^2 + ex + 1}$$

$$y\left(dx^2 + ex + 1\right) = ax^2 + bx + c$$

$$yx^2 \cdot d + yx \cdot e + y = ax^2 + bx + c$$

$$y = ax^2 + bx + c - yx^2 \cdot d - yx \cdot e$$

$$y = \left(x^2 \quad x \quad 1 \quad -yx^2 \quad -yx\right) \cdot \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix}$$

$\Rightarrow$ mit mind. 5 Meßwerten erhält man ein lin. Gleichungssystem

d) $$y = \frac{ax + b}{cx + d} + f$$

$$y = \frac{ax + b}{cx + d} + \frac{f(cx + d)}{cx + d} = \frac{ax + b + fcx + fd}{cx + d}$$

$$yx \cdot \underline{\underline{c}} + y \cdot \underline{\underline{d}} = ax + b + \underline{\underline{fc}} \cdot x + \underline{\underline{fd}}$$

$\Rightarrow$ mit mind. 5 Meßwerten erhält man ein

nichtlineares Gleichungssystem

ÜBUNG: Iterative Nullstellensuche 1

$$x^2 + 2xy^2 - e^{\frac{y}{x}} - 7 = 0$$

Welchen Wert hat y für x=1 ?
Startwert sei y_0=3.

Welchen Wert hat y für $x = 1$? (x: Eingangswert)

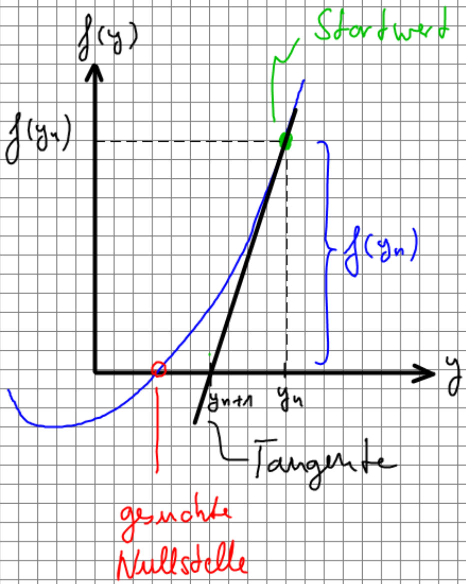
$x=1$ in Gleichung einsetzen:

$$1 + 2y^2 - e^y - 7 = 0$$

$$2y^2 - e^y - 6 = 0$$ ⇒ Umformung nach y nicht möglich!

⇒ Es wird die Nullstelle der Funktion

$$f(y) = 2y^2 - e^y - 6$$ gesucht!



$$f'(y_n) = \frac{f(y_n)}{y_n - y_{n+1}}$$ (Tangentensteigg.)

$$\Rightarrow \boxed{y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}}$$

→ Newton'sches Iterationsverfahren

$$y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}$$

Für $x=1$ gilt: $f(y) = 2y^2 - e^y - 6$

Beispiel: Der Startwert sei $y_0 = -3$

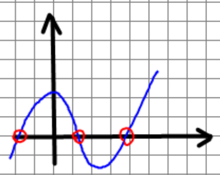
$$y_1 = y_0 - \frac{2y_0^2 - e^{y_0} - 6}{4y_0 - e^{y_0}} = -3 - \frac{18 - 0.05 - 6}{-12 - 0.05} \approx -2$$

$$y_2 = y_1 - \frac{2y_1^2 - e^{y_1} - 6}{4y_1 - e^{y_1}} = -2 - \ldots\ldots \approx -1.76$$

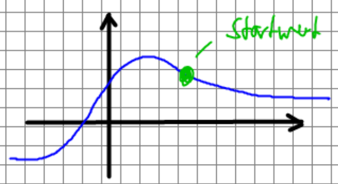
$$y_3 = y_2 - \ldots\ldots = \ldots\ldots \approx -1.795$$

$$x=1 \longrightarrow \boxed{} \longrightarrow y \approx -1.795$$

- es kann mehrere Lösungen geben (Startwertabhängig)



- Iteration kann divergieren



Was ist, wenn mehrere nichtlineare, gekoppelte Gleichungen mit mehreren Variablen vorliegen?

Die einfache Newton-Iteration ist:

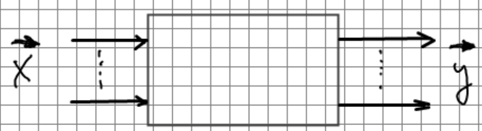
$$y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)}$$

oder etwas umgeformt

$$\boxed{f'(y_n) \cdot \Delta y = - f(y_n)}$$

mit $\Delta y = y_{n+1} - y_n$

daraus wird dann



$$\boxed{\underbrace{\underline{J}(\vec{y}_n)}_{\text{Jacobimatrix}} \cdot \Delta\vec{y} = \underbrace{- \vec{f}(\vec{y}_n)}_{\text{Lösungsvektor}}}$$

mit $\Delta\vec{y} = \vec{y}_{n+1} - \vec{y}_n$

hier steckt die verbesserte Lösung $\vec{y}_{n+1}$ der Ausgangsgröße drin

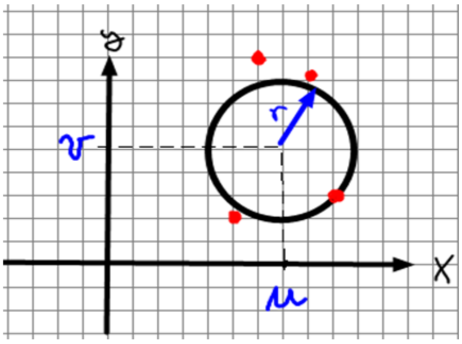
$$\underline{J} = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial y_1} & \frac{\partial f_n}{\partial y_2} & \cdots & \frac{\partial f_n}{\partial y_n} \end{pmatrix}$$

ÜBUNG: Ausgleichskreis bestimmen

Gesucht sind die Parameter u, v und r des Ausgleichskreises:

$$r^2 = (x-u)^2 + (y-v)^2$$

Geben Sie ein MATLAB-Programm zur Bestimmung der Kreisparameter an.



Als Kantenpunkte sind gegeben:
$(x_a, y_a) = (5, 2)$, $(x_b, y_b) = (9, 3)$,
$(x_c, y_c) = (6, 9)$, $(x_d, y_d) = (8, 8)$.

Verwenden Sie als Startwert u = v = r = 1.

allg. Lösungsansatz:

$$\boxed{\underline{J}(\vec{y}_n) \cdot \Delta\vec{y} = -\vec{f}(\vec{y}_n)} \quad \text{mit} \quad \Delta\vec{y} = \vec{y}_{n+1} - \vec{y}_n$$

Konkret gilt hier: $\underline{J}(u_n, v_n, r) \cdot \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta r \end{pmatrix} = -f(u_n, v_n, r_n)$

4 Punkte $\Rightarrow$ 4 Funktionen aufstellen: (deren Nullstelle gesucht wird ...)

$$\underbrace{\begin{bmatrix} (x_a-u)^2 + (y_a-v)^2 - r^2 \\ \vdots \\ (x_d-u)^2 + (y_d-v)^2 - r^2 \end{bmatrix}}_{\text{oder kurz } \vec{f}(u,v,r)} = \begin{bmatrix} f_a(u,v,r) \\ \vdots \\ f_d(u,v,r) \end{bmatrix} \overset{!}{=} \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Jacobimatrix aufstellen:

$$\underline{J} = \begin{pmatrix} \frac{\partial f_a}{\partial u} & \frac{\partial f_a}{\partial v} & \frac{\partial f_a}{\partial r} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_d}{\partial u} & \frac{\partial f_d}{\partial v} & \frac{\partial f_d}{\partial r} \end{pmatrix}$$

mit den Ableitungen $\frac{\partial f_k}{\partial u} = -2(x_k - u)$ $\quad k = a, \dots, d$

$$\frac{\partial f_k}{\partial v} = -2(y_k - v)$$

$$\frac{\partial f_k}{\partial r} = -2r$$

Jacobimatrix für den 1. Iterationsschritt aufstellen

$$\underline{J}_0 = \begin{pmatrix} -2(x_a - u_0) & -2(y_a - v_0) & -2r_0 \\ \vdots & & \\ -2(x_d - u_0) & -2(y_d - v_0) & -2r_0 \end{pmatrix}$$

mit
$u_0 = 1$
$v_0 = 1$
$r_0 = 1$

$$= \begin{pmatrix} -2(5-1) & -2(2-1) & -2 \\ \vdots & \vdots & \vdots \\ -2(8-1) & -2(8-1) & -2 \end{pmatrix} = \begin{pmatrix} -8 & -2 & -2 \\ \vdots & \vdots & \vdots \\ -14 & -14 & -2 \end{pmatrix}$$

$$\underline{J}_0 = \begin{pmatrix} -8 & -2 & -2 \\ -16 & -4 & -2 \\ -10 & -16 & -2 \\ -14 & -14 & -2 \end{pmatrix}$$

Zur Erinnerung:

$$\boxed{\underline{J}(\vec{y}_n) \cdot \Delta\vec{y} = -\vec{f}(\vec{y}_n)} \quad \text{mit} \quad \Delta\vec{y} = \vec{y}_{n+1} - \vec{y}_n$$

Lösungsvektor (rechte Seite) berechnen:

$$\vec{f}_0(u,v,r) = \begin{pmatrix} (x_a - u_0)^2 + (y_a - v_0)^2 - r_0^2 \\ \vdots \\ (x_d - u_0)^2 - (y_d - v_0)^2 - r_0^2 \end{pmatrix}$$

$$= \begin{pmatrix} (5-1)^2 + (2-1)^2 - 1^2 \\ (9-1)^2 + (3-1)^2 - 1^2 \\ (6-1)^2 + (9-1)^2 - 1^2 \\ (8-1)^2 + (8-1)^2 - 1^2 \end{pmatrix} = \begin{pmatrix} 16 \\ 67 \\ 88 \\ 97 \end{pmatrix}$$

Iterationsgleichung für den 1. Schritt

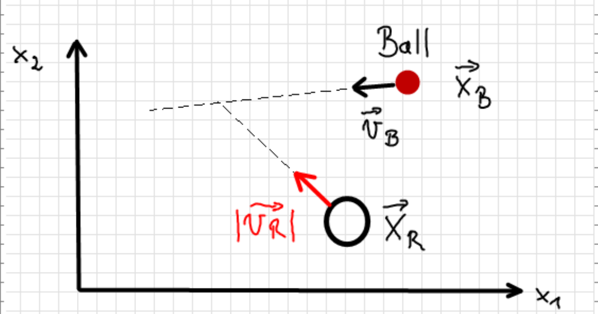
$$\underline{J}_0 \cdot \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta r \end{pmatrix} = -\begin{pmatrix} 16 \\ 67 \\ 88 \\ 97 \end{pmatrix} \quad \text{mit} \quad \begin{aligned} \Delta u &= u_1 - u_0 = u_1 - 1 \\ \Delta v &= v_1 - v_0 = v_1 - 1 \\ \Delta r &= r_1 - r_0 = r_1 - 1 \end{aligned}$$

Matlab - Snippet

```
% Eingangsgrößen
xa = 5;    ya = 2;
xb = 9;    yb = 3;
xc = 6;    yc = 9;
xd = 8;    yd = 8;

% initialer Schätzwert der Parameter
u = 5; v = 5; r=2;

% Definitionen und Initialisierungen
DeltaA     = ones(3,1);
J          = ones(4,3);
```

```
while norm(DeltaA) > 1e-4     % Fehlergrenze

  % aktuelle Jacobimatrix und Funktion berechnen
  J = [ -2*(xa-u),  -2*(ya-v), -2*r ; ...
        -2*(xb-u),  -2*(yb-v), -2*r ; ...
        -2*(xc-u),  -2*(yc-v), -2*r ; ...
        -2*(xd-u),  -2*(yd-v), -2*r ; ];

  % Lösungsvektor
  MinusF = -[ (xa-u)^2 + (ya-v)^2 - r^2 ; ...
              (xb-u)^2 + (yb-v)^2 - r^2 ; ...
              (xc-u)^2 + (yc-v)^2 - r^2 ; ...
              (xd-u)^2 + (yd-v)^2 - r^2 ;];

  % Lösen des Gleichungssystems
  DeltaA = linsolve(J, MinusF);

  % Die verbesserten Unbekannten bestimmen
  u = u + DeltaA(1); v = v + DeltaA(2); r = r + DeltaA(3);

end
```

ÜBUNG: Abfangen eines Balles



$$\vec{x}_B = (x_{B_1},\ x_{B_2})^T$$
$$\vec{v}_B = (v_{B_1},\ v_{B_2})^T$$
$$\vec{x}_R = (x_{R_1},\ x_{R_2})^T$$

Ort des Balls zum Zeitpunkt t:

$$\vec{x}_B(t) = \vec{x}_{B_0} + \vec{v}_B \cdot t$$

$\vec{v}_B$: Geschwindigkeitsvektor des Balls

$\vec{x}_{B_0}$: Ort des Balls zum Zeitpunkt $t = 0$

Für den Roboter gilt:

$$\vec{v}_R = (v_{R_1},\ v_{R_2})^T$$

Geschwindigkeitskomponenten in x- und y-Richtung (unbekannt)

$$|v_R| = v_R = \sqrt{v_{R_1}^2 + v_{R_2}^2} \qquad (1)$$

(bekannt)

Roboter und Ball treffen zusammen, wenn

$$\vec{x}_B(t) = \vec{x}_R(t)$$

hier leider nur der Betrag bekannt

$$\Leftrightarrow \quad \vec{x}_{B0} + \vec{v}_B \cdot t = \vec{x}_{R0} + \vec{v}_R \cdot t$$

oder aufgesplittet in die beiden Komponentenrichtgn.

$$x_{B1} + v_{B1} \cdot \underline{t} = x_{R1} + \underline{v_{R1}} \cdot \underline{t} \qquad (2)$$

$$x_{B2} + v_{B2} \cdot \underline{t} = x_{R2} + \underline{v_{R2}} \cdot \underline{t} \qquad (3)$$

und $$v_R^2 = \underline{v_{R1}}^2 + \underline{v_{R2}}^2 \qquad (1) \qquad \text{(s.o.)}$$

$\Rightarrow$ 3 Gleichungen mit 3 Unbekannten

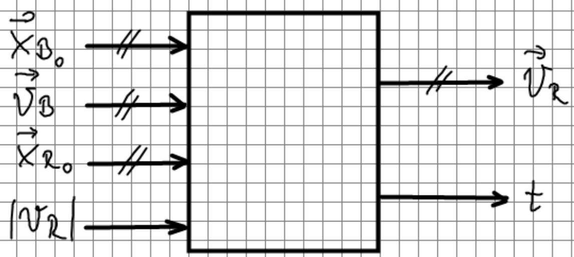
v_{R1}, v_{R2}, t $\Rightarrow$ kein lin. Gleichungssyst.

Als Nullstellensuche formuliert:

$$(4) \quad x_{B1} + v_{B1} \cdot t - x_{R1} - v_{R1} \cdot t = f_1(v_{R1}, t) \overset{!}{=} 0$$

$$(5) \quad x_{B2} + v_{B2} \cdot t - x_{R2} - v_{R2} \cdot t = f_2(v_{R2}, t) \overset{!}{=} 0$$

$$(6) \quad v_{R1}^2 + v_{R2}^2 - v_R^2 = f_3(v_{R1}, v_{R2}) \overset{!}{=} 0$$



Newton-Verfahren für mehrere Unbekannte

$$(4)\quad x_{B_1} + v_{B_1}\cdot t - x_{R_1} - v_{R_1}\cdot t = f_1(v_{R_1}, t)$$

$$(5)\quad x_{B_2} + v_{B_2}\cdot t - x_{R_2} - v_{R_2}\cdot t = f_2(v_{R_2}, t)$$

$$(6)\quad v_{R_1}^2 + v_{R_2}^2 - v_R^2 = f_3(v_{R_1}, v_{R_2})$$

$$\Downarrow$$

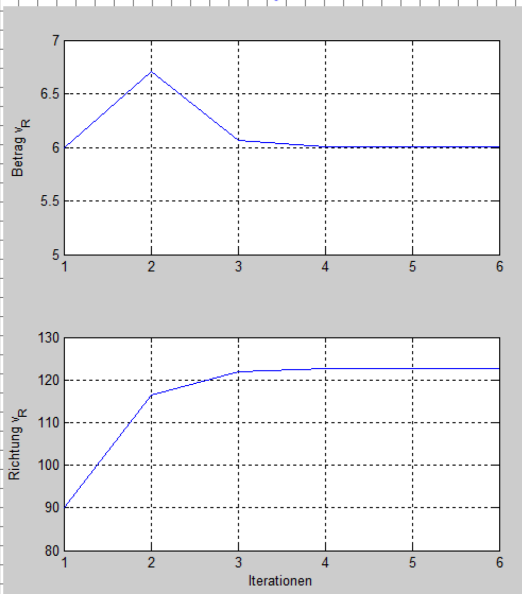
$$\begin{pmatrix} \frac{\partial f_1}{\partial v_{R_1}} & \frac{\partial f_1}{\partial v_{R_2}} & \frac{\partial f_1}{\partial t} \\ \frac{\partial f_2}{\partial v_{R_1}} & \frac{\partial f_2}{\partial v_{R_2}} & \frac{\partial f_2}{\partial t} \\ \frac{\partial f_3}{\partial v_{R_1}} & \frac{\partial f_3}{\partial v_{R_2}} & \frac{\partial f_3}{\partial t} \end{pmatrix} \cdot \begin{pmatrix} \Delta v_{R_1} \\ \Delta v_{R_2} \\ \Delta t \end{pmatrix} = - \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

$$\begin{pmatrix} -t & 0 & v_{B_1} - v_{R_1} \\ 0 & -t & v_{B_2} - v_{R_2} \\ 2v_{R_1} & 2v_{R_2} & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta v_{R_1} \\ \Delta v_{R_2} \\ \Delta t \end{pmatrix} = - \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

$\Rightarrow$ damit erhält man einen verbesserten Nährungswert

```
% Eingangsgrößen
xB = [12, 12];
xR = [10 ,4];
vB = [-4, 2];
vR_Betrag = 6;

% initialer Schätzwert
vR =[-0, 6];
t  = 2;

% Iterieren bis Fehlergrenze unterschritten
while norm(Delta) > 1e-8      % Fehlergrenze

    % aktuelle Jacobimatrix und Funktion berechnen
    J = [     -t,          0,      vB(1)-vR(1)  ;  ...
               0,         -t,      vB(2)-vR(2)  ; ...
        2*vR(1),   2*vR(2),      0];

    f = [ xB(1)    + vB(1)*t - xR(1) - vR(1)*t , ...
          xB(2)    + vB(2)*t - xR(2) - vR(2)*t , ...
          vR(1)^2 + vR(2)^2 - vR_Betrag^2 ];

    % Lösen des Gleichungssystems
    Delta = linsolve(J, -f');

    vR(1)  = vR(1)  + Delta(1);
    vR(2)  = vR(2)  + Delta(2);
    t      = t      + Delta(3);
end
```

Iterationsverlauf



Lösung

```
vR =

   -3.2370    5.0519

t =

    2.6213
```

ÜBUNG: Iterative Nullstellensuche 2

in diese Form bringen

Gegeben ist die Funktion $y = a \cdot e^{-bx}$ $\Rightarrow$ $\boxed{f = a \cdot e^{-bx} - y \overset{!}{=} 0}$

Weiter sind folgende Messwerte gegeben: (0.5, 1.1), (1.0, 0.4), (2.0, 0.055)
(x_1, y_1) (x_2, y_2) (x_3, y_3)

Die initialen Schätzwerte der Parameter sind: $a_0 = 4$, $b_0=3$
Die Parameter a und b sind zu bestimmen.

Unbekannt und gesucht: a, b

3 Gleichungen für 2 Unbekannte (→ Ausgleichen)

Das nichtlin. Gleichungssystem lautet:

$$f_1 = a \cdot e^{-b \cdot 0{,}5} - 1.1 \overset{!}{=} 0$$
$$f_2 = a \cdot e^{-b \cdot 1{,}0} - 0.4 \overset{!}{=} 0$$
$$f_3 = a \cdot e^{-b \cdot 2.0} - 0.055 \overset{!}{=} 0$$

$$\frac{\partial f}{\partial a} = e^{-bx} \qquad \frac{\partial f}{\partial b} = -x \cdot a \cdot e^{-bx}$$

Jacobi-Matrix aufstellen (d.h. jede Funktion nach jeder Ausgangsgröße ableiten)

$$\left.\begin{pmatrix} \frac{\partial f_1}{\partial a} & \frac{\partial f_1}{\partial b} \\ \frac{\partial f_2}{\partial a} & \frac{\partial f_2}{\partial b} \\ \frac{\partial f_3}{\partial a} & \frac{\partial f_3}{\partial b} \end{pmatrix}\right|_{\hat{a}, \hat{b}} = \begin{pmatrix} e^{-3 \cdot 0.5} & -0{,}5 \cdot 4 \cdot e^{-3 \cdot 0.5} \\ e^{-3 \cdot 1} & -1 \cdot 4 \cdot e^{-3 \cdot 1} \\ e^{-3 \cdot 2} & -2 \cdot 4 \cdot e^{-3 \cdot 2} \end{pmatrix} =$$

$\hat{a}, \hat{b}$ — mit den aktuellen Schätzwerten

$$\left.\begin{pmatrix} \frac{\partial f_1}{\partial a} & \frac{\partial f_1}{\partial b} \\ \frac{\partial f_2}{\partial a} & \frac{\partial f_2}{\partial b} \\ \frac{\partial f_3}{\partial a} & \frac{\partial f_3}{\partial b} \end{pmatrix}\right|_{\hat{a},\hat{b}} = \begin{pmatrix} 0.2231 & -0.4463 \\ 0.0498 & -0.1991 \\ 0.0025 & -0.0198 \end{pmatrix} = \underline{J} \quad \text{Jacobi-matrix}$$

Lösungsvektor $-\vec{f}$ aufstellen (mit den aktuellen Schätzwerten)

$$\underbrace{\begin{pmatrix} -f_1 \\ -f_2 \\ -f_3 \end{pmatrix}}_{-\vec{f}} = \begin{pmatrix} 1.1 - 4\cdot e^{-3\cdot 0.5} \\ 0.4 - 4\cdot e^{-3\cdot 1} \\ 0.055 - 4\cdot e^{-3\cdot 2} \end{pmatrix} = \begin{pmatrix} 0.2075 \\ 0.2009 \\ 0.0451 \end{pmatrix}$$

aktuelles a   aktuelles b

Gleichungssystem zur Verbesserung der Parameter

$$\underbrace{\begin{pmatrix} 0.2231 & -0.4463 \\ 0.0498 & -0.1991 \\ 0.0025 & -0.0198 \end{pmatrix}}_{\underline{J}} \cdot \underbrace{\begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix}}_{\Delta\vec{a}} \underset{=}{=} \underbrace{\begin{pmatrix} 0.2075 \\ 0.2009 \\ 0.0451 \end{pmatrix}}_{-\vec{f}}$$

$$\Delta\vec{a} = \begin{pmatrix} a_{n+1} - a_n \\ b_{n+1} - b_n \end{pmatrix}$$

2.3. Lösen mit $\underline{J}^T \underline{J} \cdot \Delta\vec{a} = \underline{J}^T \cdot (-\vec{f})$ Ausgleichslösung da überbestimmt

$$\Longrightarrow \quad \Delta\vec{a} = \begin{pmatrix} -2.2361 \\ -1.5820 \end{pmatrix}$$

Bestimmung der verbesserten Parameter

$$\Delta\vec{a} = \vec{a}_{n+1} - \vec{a}_n \quad \Longleftrightarrow$$

$$a_{n+1} = \Delta\vec{a} + \vec{a}_n = \begin{pmatrix} -2.2361 \\ -1.5820 \end{pmatrix} + \begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 1.76 \\ 1.42 \end{pmatrix}$$

nächste Iteration

verbesserte Parameter als neue Schätzwerte nehmen

$\underline{J}$ aktualisieren, $-\vec{f}$ aktualisieren

$\Delta\vec{a}$ per Ausgleichg. bestimmen ..... u.s.w.

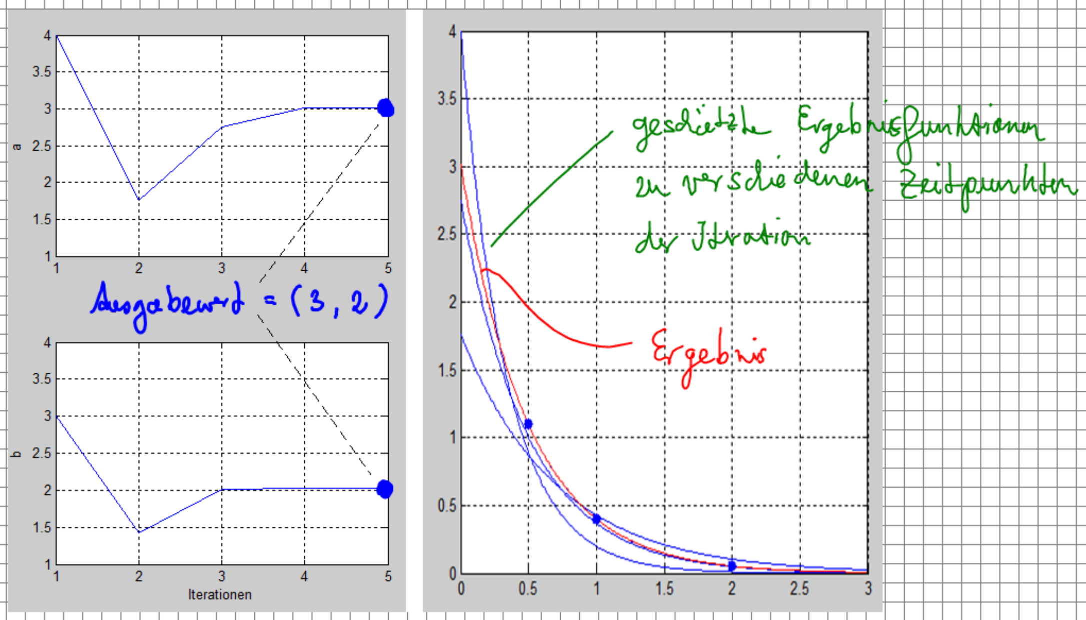
```
% Iterieren bis Fehlergrenze unterschritten
while norm(DeltaA) > 1e-4     % Fehlergrenze

    % aktuelle Jacobimatrix und Funktion berechnen
    J = [ exp(-b*x1),  -a*x1*exp(-b*x1); ...
          exp(-b*x2),  -a*x2*exp(-b*x2); ...
          exp(-b*x3),  -a*x3*exp(-b*x3); ];

    % Lösungsvektor
    MinusF = [ y1 - a*exp(-b*x1) ; ...
               y2 - a*exp(-b*x2) ; ...
               y3 - a*exp(-b*x3) ];

    % Lösen des Gleichungssystems
    DeltaA = linsolve(J, MinusF);

    a = a + DeltaA(1);
    b = b + DeltaA(2);
end
```

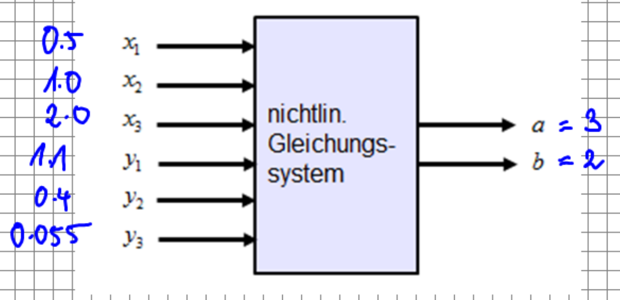
```
J =                        MinusF =        DeltaA =
   0.2231   -0.4463          0.2075           -2.2361
   0.0498   -0.1991          0.2009           -1.5820
   0.0025   -0.0198          0.0451
J =                        MinusF =        DeltaA =
   0.4921   -0.4340          0.2319            0.9883
   0.2422   -0.4272         -0.0272            0.5955
   0.0587   -0.2069         -0.0485
J =                        MinusF =        DeltaA =
   0.3654   -0.5028          0.0943            0.2664
   0.1335   -0.3675          0.0325            0.0064
   0.0178   -0.0981          0.0059
J =                        MinusF =        DeltaA =
   0.3643   -0.5498          0.0005         1.0e-003 *
   0.1327   -0.4005         -0.0005            0.0674
   0.0176   -0.1063          0.0019           -0.5129
J =                        MinusF =        DeltaA =
   0.3643   -0.5499          0.0002         1.0e-005 *
   0.1327   -0.4007         -0.0007           -0.2801
   0.0176   -0.1064          0.0018           -0.1560
```

soll gegen 0 gehen

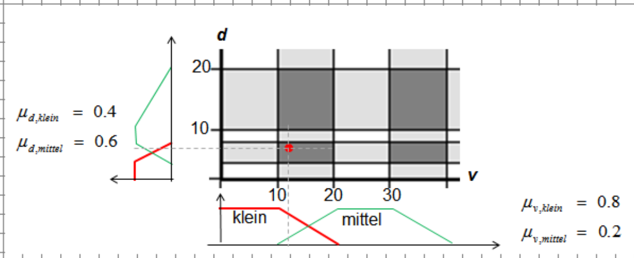
die Schritte werden immer kleiner
$\vec{a}_{n+1} - \vec{a}_n$

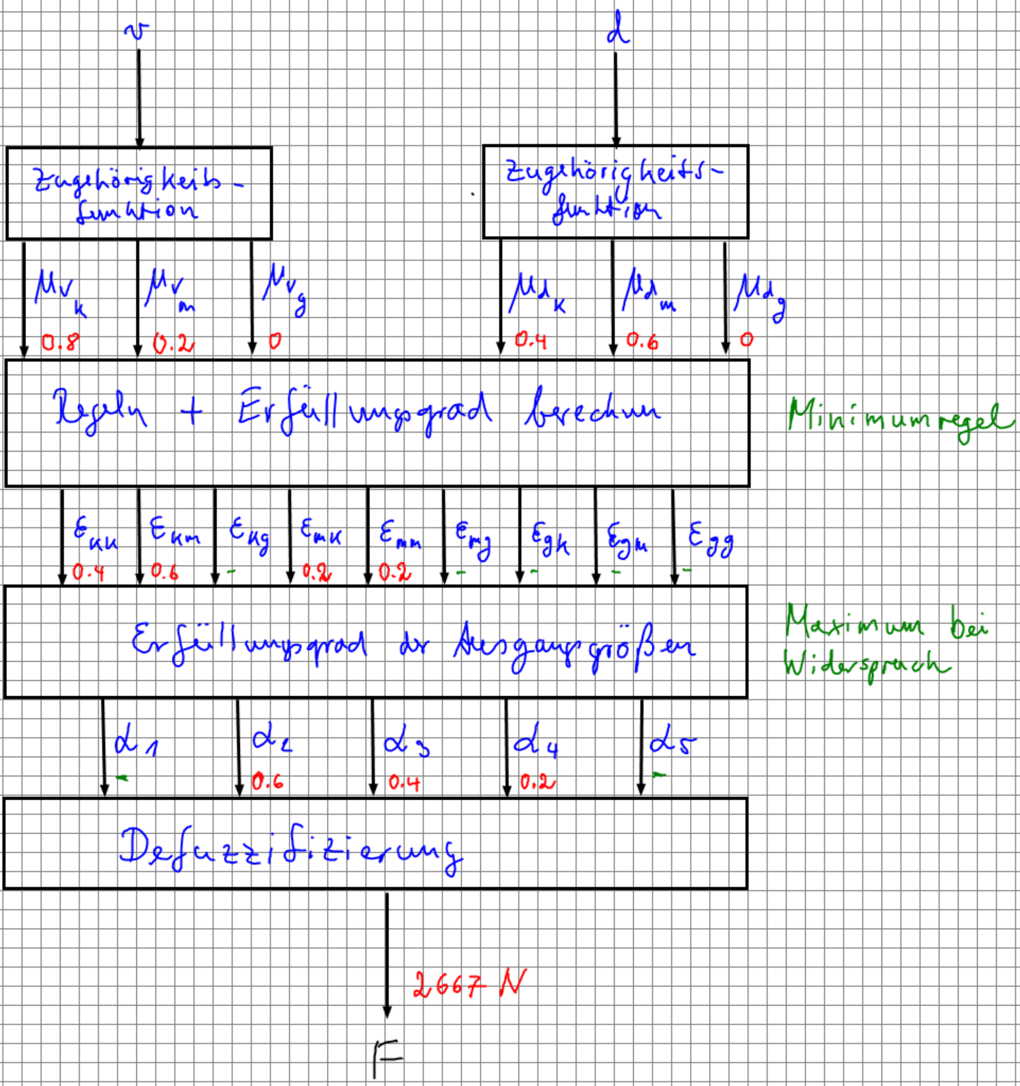
Entwicklung der Funktionsparameter a und b (Ausgangsgrößen)





Fuzzy – Logic


d
20
10
v
10
20
30
klein
mittel
$\mu_{d,klein}$ = 0.4
$\mu_{d,mittel}$ = 0.6
$\mu_{v,klein}$ = 0.8
$\mu_{v,mittel}$ = 0.2


v
d
Zugehörigkeits-funktion
Zugehörigkeits-funktion
μ_{v_k} 0.8
μ_{v_m} 0.2
μ_{v_g} 0
μ_{d_k} 0.4
μ_{d_m} 0.6
μ_{d_g} 0
Regeln + Erfüllungsgrad berechnen
Minimumregel
ε_{kk} 0.4
ε_{km} 0.6
ε_{kg} –
ε_{mk} 0.2
ε_{mm} 0.2
ε_{mg} –
ε_{gk} –
ε_{gm} –
ε_{gg} –
Erfüllungsgrad der Ausgangsgrößen
Maximum bei Widerspruch
α_1 –
α_2 0.6
α_3 0.4
α_4 0.2
α_5 –
Defuzzifizierung
2667 N
F