



Prof. Dr. Ludwig Paditz Page 1 13.01.2017



Nichtlineare, speziell arctan-Regression mit dem ClassPad II

(quasilineare Regression, Levenberg-Marquardt-Algorithmus)

Nichtlineare Regressionsmodelle können oftmals nur näherungsweise (iterativ) an vorliegende Datenerhebungen angepasst werden und die Parameterschätzungen sind mehr oder weniger ungenau, wenn das Iterationsverfahren zu zeitig abgebrochen wird oder nicht konvergiert. Zur Vereinfachung der Rechnung werden dann, sofern möglich, ersatzweise quasilineare Modelle genutzt.

Prof. Dr. Ludwig Paditz Page 2 13.01.2017



Nichtlineare, speziell arctan-Regression mit dem ClassPad II

(quasilineare Regression, Levenberg-Marquardt-Algorithmus)

Im Vortrag wird speziell auf die **arctan-Regression** mit dem ClassPad II eingegangen, die jedoch im ClassPad nicht bereitgestellt wird und daher neu programmiert werden musste.

Für das betrachtete Beispiel (Daten aus dem Chemieunterricht am Gymnasium Coswig) werden brauchbare Parameterschätzungen für die arctan-Regression erhalten. Mithilfe eines Iterationsverfahrens (Levenberg-Marquardt-Algorithmus) konnte die Parameterschätzung optimiert werden.

Prof. Dr. Ludwig Paditz Page 3 13.01.2017



Nichtlineare, speziell arctan-Regression mit dem ClassPad II

(quasilineare Regression, Levenberg-Marquardt-Algorithmus)

- Der Levenberg-Marquardt-Algorithmus, benannt nach <u>Kenneth Levenberg</u> und <u>Donald Marquardt</u>, ist ein <u>numerischer Optimierungsalgorithmus</u> zur Lösung nichtlinearer <u>Ausgleichs-Probleme</u> mit Hilfe der <u>Methode der kleinsten Quadrate</u>. Das Verfahren kombiniert das <u>Gauß-Newton-Verfahren</u> mit einer Regularisierungstechnik, die absteigende Funktionswerte erzwingt.
- Der Levenberg-Marquardt-Algorithmus ist deutlich <u>robuster</u> als das Gauß-Newton-Verfahren, das heißt, er konvergiert mit einer hohen Wahrscheinlichkeit auch bei schlechten Startbedingungen, allerdings ist auch hier Konvergenz nicht garantiert. Ferner ist er bei Anfangswerten, die nahe dem Minimum liegen, oft etwas langsamer.

Prof. Dr. Ludwig Paditz Page 4 13.01.2017



Das Beispiel:

Daten aus dem Chemieunterricht am Gymnasium Coswig: 50 Messpunkte (listx angepasst)

$$seq(x,x,1,50,1) \Rightarrow listx$$

{1, 2, 3, ..., 48, 49, 50}

{3.25, 3.35, 3.54, 3.65, 3.74, 3.82, 3.87, 3.94, 4.00, 4.06,

4.11, 4.22, 4.22, 4.27, 4.32, 4.34, 4.39, 4.44, 4.46, 4.52,

4.56, 4.62, 4.70, 4.73, 4.77, 4.82, 4.89, 4.93, 5.00, 5.09,

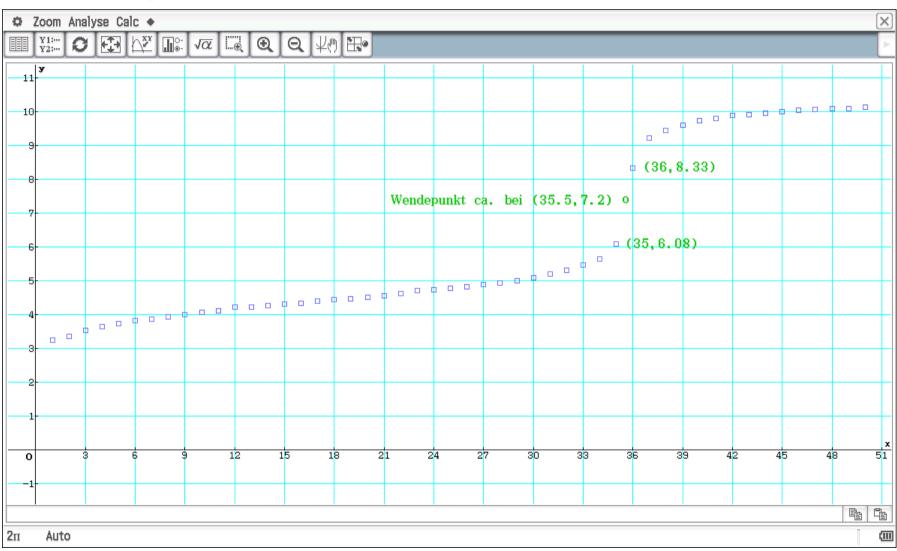
5.19, 5.31, 5.47, 5.65, 6.08, 8.33, 9.22, 9.44, 9.61, 9.74,

9.80, 9.88, 9.92, 9.96, 10.01, 10.05, 10.06, 10.08, 10.10,

10.13} ⇒listy



Das Beispiel: Wendepunkt ca. bei (b,d)=(35.5, 7.2)





Der Modellansatz: anfangs wurde die logistische Regression versucht

Logistische Regression: $y = c / (1 + a \cdot e^{(-b \cdot x)}) + d$

Problem: im CP400 ist nur $y = c / (1 + a \cdot e^{(-b \cdot x)})$ vorhanden.

In einem anderen Beispiel zeigte sich, dass diese Modellanpassung im CP400 sehr ungenau programmiert ist (zu schneller Abbruch des Iterationsverfahrens zur Bestimmung der Modellparameter a, b, c)

Vorschlag von Herrn Stauch (05/2016):

arctan-Regression: y = a*(arctan(c*(x - b)) + d Wendepunkt (b,d)

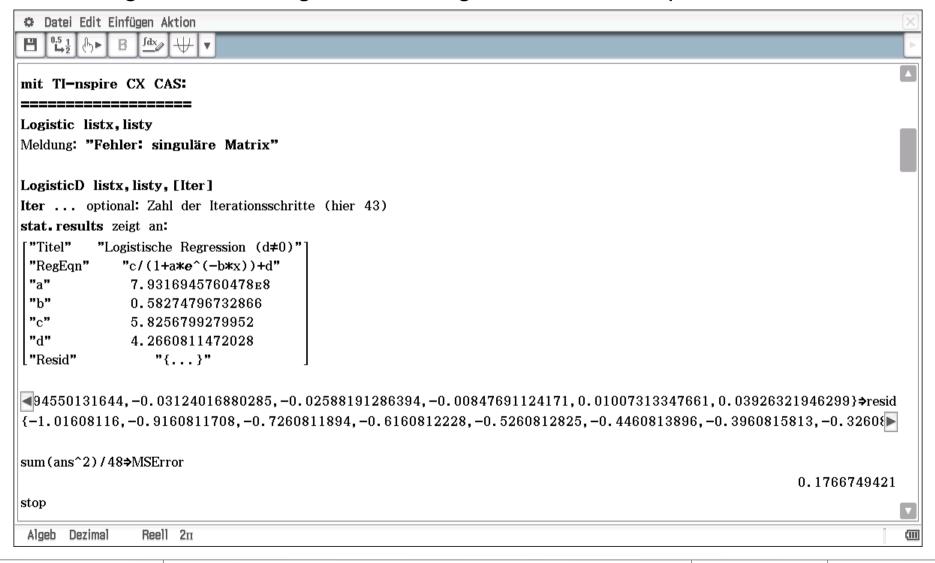
Problem: Gauß-Newton-Verfahren konvergiert schlecht.

Lösung: Levenberg-Marquardt-Verfahren auf dem ClassPad II

Inzwischen stellte sich heraus, dass die arctan-Regression sogar eine bessere Modellanpassung liefert als die logistische Regression mit dem Ansatz $y = c / (1 + a \cdot e^{-(-b \cdot x)}) + d$.

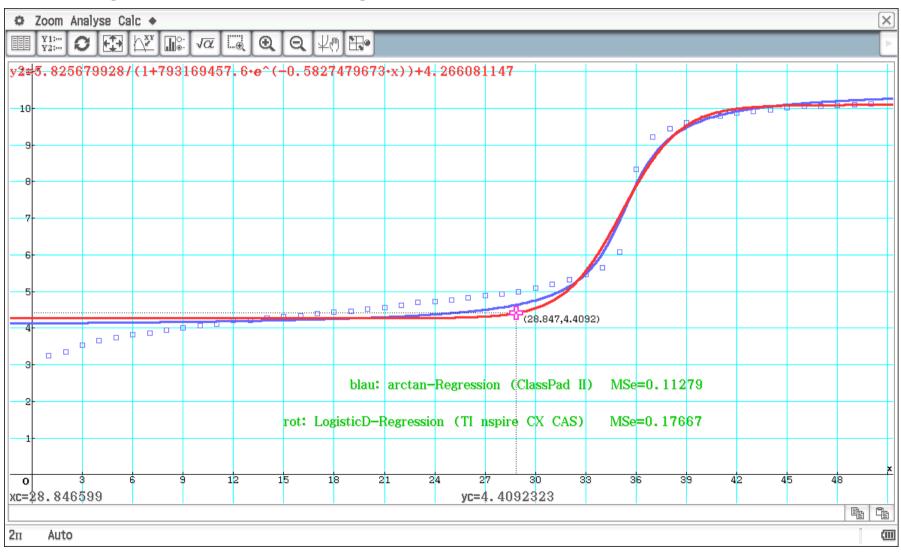


Das Ergebnis einer logistischen Regression mit TI nspire CX CAS:





Das Ergebnis der arctan-Regression mit ClassPad II:





Das Levenberg-Marquardt-Verfahren:

Quellen:

https://www.igpm.rwth-aachen.de/Numa/NumaMB/SS10/LevMarq.pdf

http://www.mathematik.uni-kl.de/fileadmin/AGs/techno/

lecture_files/SS14/HM-Numerik/BeispielLevenbergMarquardt.pdf

https://de.wikipedia.org/wiki/Levenberg-Marquardt-Algorithmus

Das L-M-Verfahren arbeitet stabiler als Gauss-Newton-Verfahren:

Gegeben ist die Funktion $y=f(x_1,...,x_n)$ mit n Variablen $x_1,...,x_n$ und p gesuchten Parametern $a_1,...,a_p$. Hier n=1 und p=4.

ClassPad400:

Define F(x,a,b,c,d) = a*(arctan(c*(x-b))+d x... xList

(die Modellfunktion)

Define r(x,y,a,b,c,d) = y-F(x,a,b,c,d) y...yList (die Residuen)



Programm arctDReg(xList,yList,a0,b0,c0,d0,µ)

Aufbau der symmetrischen Matrix SP := $trn(F')*F' + \mu^2*I$ (I ... Einheitsmatrix) vom Typ (4,4):

Berechnung der partiellen Ableitungen der Residuumsfunktion **r** nach jedem Parameter a₁=a, a₂=b, a₃=c, a₄=d :

Define $r_k(x,a,b,c,d)=diff(r(x,y,a,b,c,d),a_k), k=1,2,3$ Define $r_4(x,a,b,c,d)=diff(r(x,y,a,b,c,d),d)=-1$



Aufbau der Elemente SPjk (j,k=1,2,3,4 mit j≤k) in Matrix SP:

```
approx(sum((approx(r_j(listx,a,b,c,d)))^2)+\mu^2)\RightarrowSPjj , j=1,2,3,
approx(dim(listx)+\mu^2)\RightarrowSP44
approx(sum(approx(r_i(listx,a,b,c,d))*approx(r_k(listx,a,b,c,d))))\RightarrowSPjk
```

Berechnung der Korrekturen (vecs) für die Parameter a,b,c,d:

approx([[a],[b],[c],[d]]+vecs)⇒vecab

Prof. Dr. Ludwig Paditz Page 12 13.01.2017



Steuerung des Verfahrens mit Steuerparameter µ:

Parametersteuerung $\rho\mu_0$:= $((\|F(a,b,c,d)\|^2 - \|F(a+s1,b+s2,c+s3,d+s4)\|^2) / \\ (\|F(a,b,c,d)\|^2 - \|F(a,b,c,d) + F'_*[[s1],[s2],[s3],[s4]]\|^2))$

Steuerungsregel:

Wenn $0.2 < \rho \mu_0 < 0.8$, dann μ beibehalten

Wenn $\rho\mu_0 > 0.8$, dann μ halbieren

Wenn $\rho\mu_0$ < 0.2, dann μ verwerfen und mit **verdoppeltem** μ starten.

Anmerkung: die Schranken 0.2 und 0.8 können variiert werden, ebenso das Verkleineren (halbieren) und das Vergrößern (verdoppeln).

Prof. Dr. Ludwig Paditz Page 13 13.01.2017



Iteration nach Levenberg-Marquardt-Verfahren in Einzelschritten und Beobachtung der Iteration:

vecab[1,1] \Rightarrow a, vecab[2,1] \Rightarrow b, vecab[3,1] \Rightarrow c, vecab[4,1] \Rightarrow d approx(sum(FA^2)/48) \Rightarrow MSerr

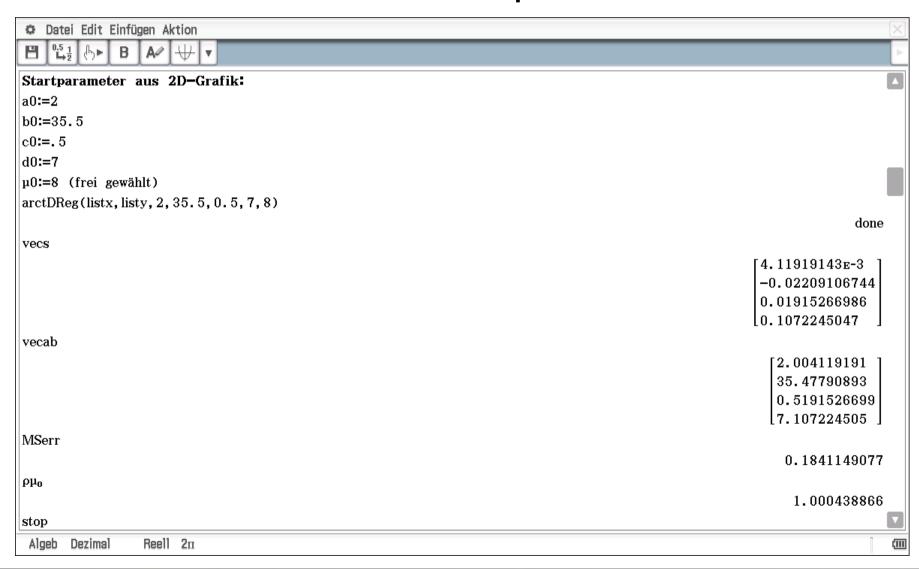
Aufruf des Korrekturvektors vecs, der korrigierten Parametervektors vecab, des MSerr und des Steuerindikators ρμο

Programmaufruf (Start): arctDReg(listx,listy,a0,b0,c0,d0,μ0)
Programmaufruf (z.B. mit μ=4):
arctDReg(listx,listy,vecab[1,1],vecab[2,1],vecab[3,1],vecab[4,1],4)

Prof. Dr. Ludwig Paditz Page 14 13.01.2017



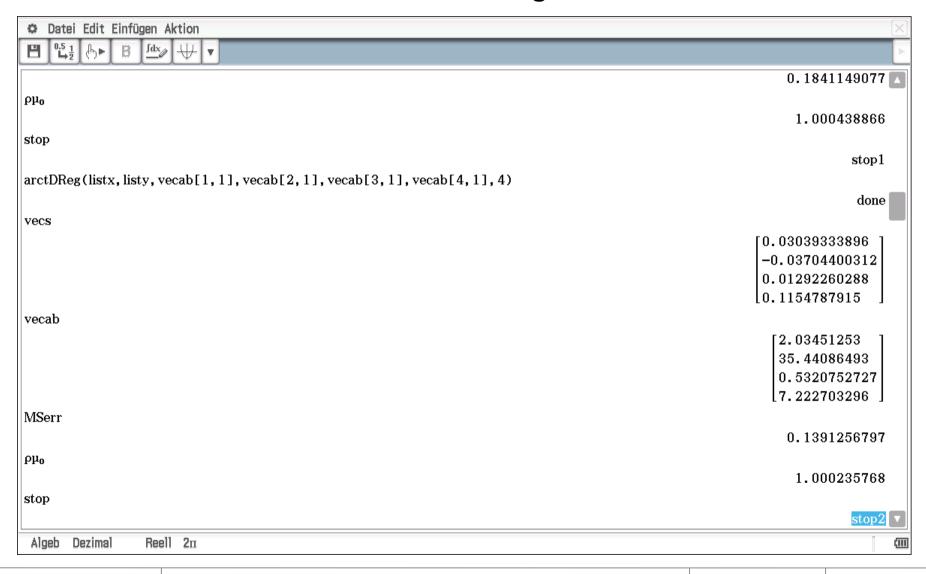
Der erste Iterationsschritt mit den Startparametern:



Prof. Dr. Ludwig Paditz

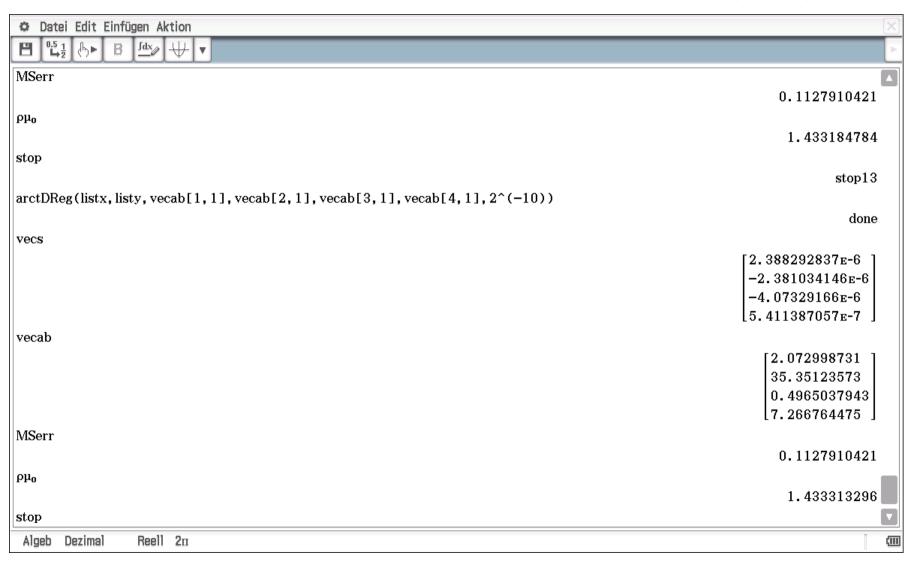


Der zweite Iterationsschritt mit den korrigierten Parametern:

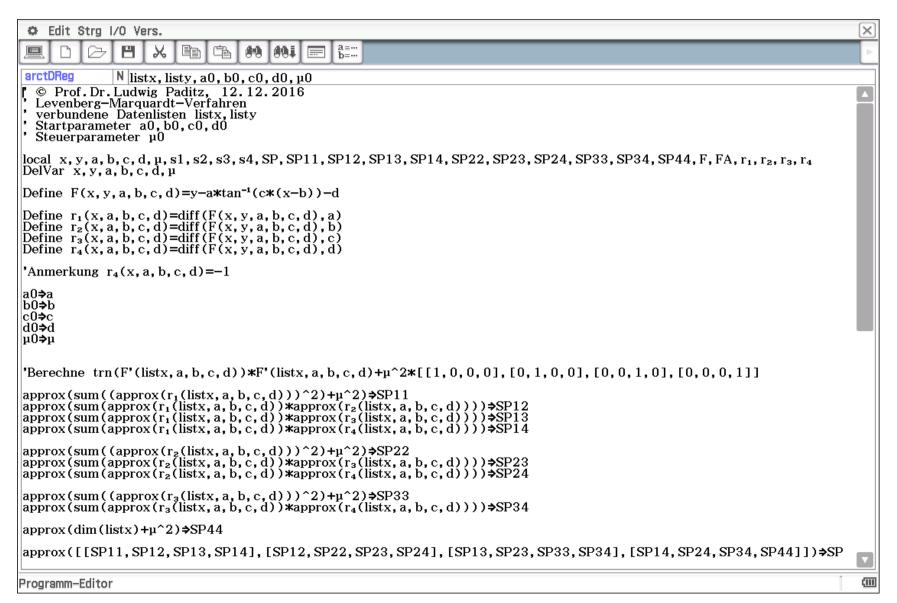




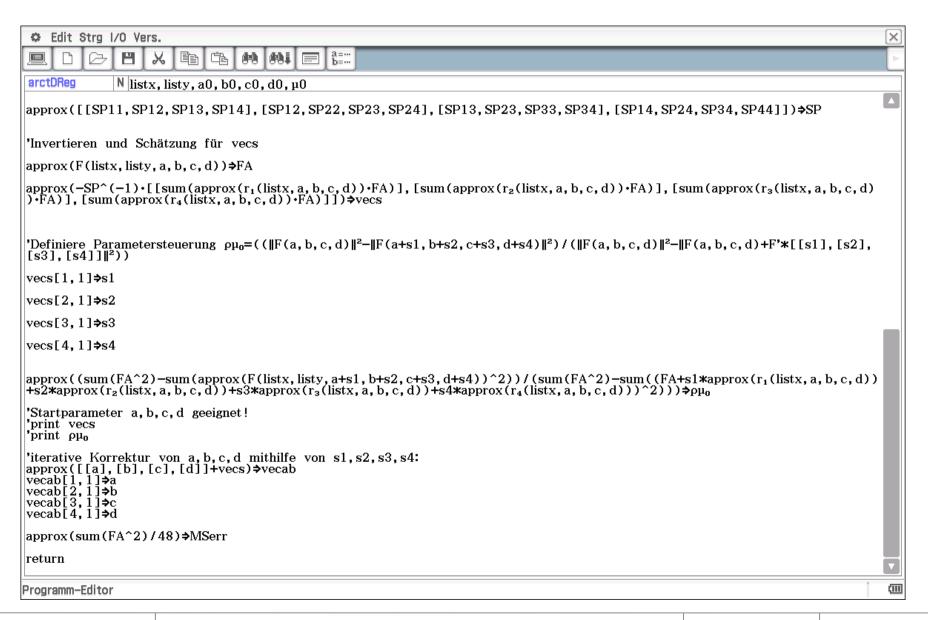
Der 14. Iterationsschritt mit den korrigierten Parametern (Ende):



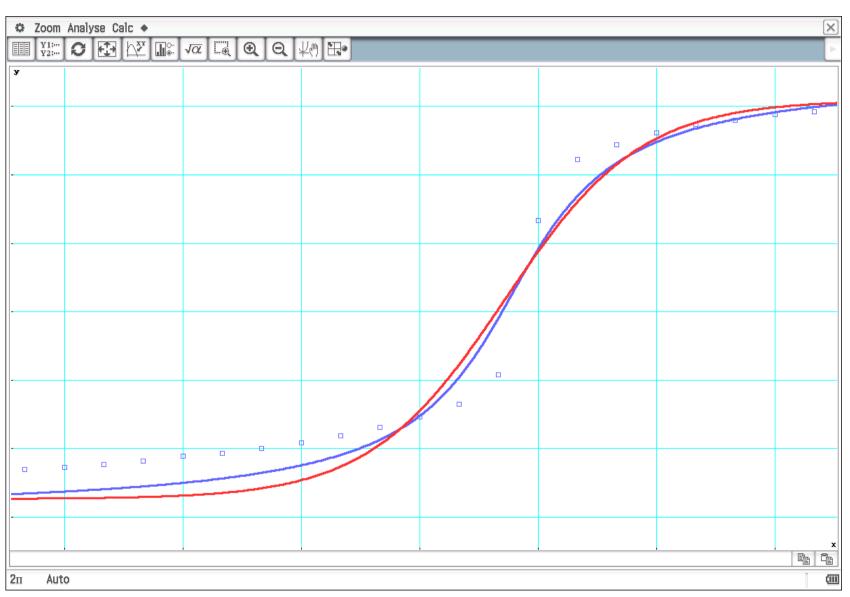






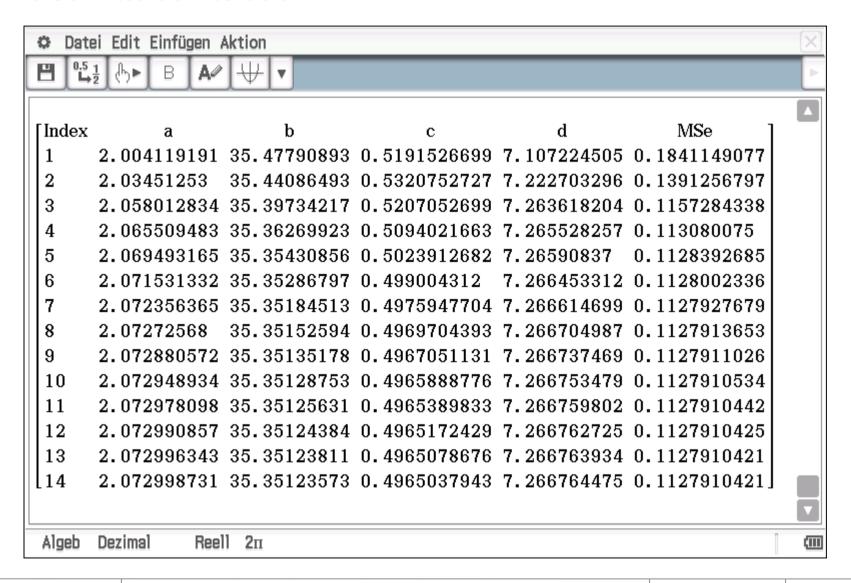








Einzelschritte der Iteration:





Nichtlineare Regression – quasilineare Regression

Beispiel Potenzregression: $y(x) = m^*x^k$

Datensatz: $seq(x,x,1,3,1) \Rightarrow listx$ {1, 2, 3}

 $\{1,4.1,8.8\} \Rightarrow \text{listy}$ $\{1,4.1,8.8\}$

optimale Parameter: k=1.918615954, m=1.071020122

MSe = 0.007852203424

Potenzregression (als quasilineare Regression) im ClassPad:

k=1.985583089, m=1.009394317

MSe = 0.03071481318

Fazit: die quasilin. Reg. ergibt nur eine Näherungslösung!

Prof. Dr. Ludwig Paditz Page 22 13.01.2017



Nichtlineare Regression – quasilineare Regression

Beispiel Potenzregression: $y(x) = m^*x^k$

Datensatz: $seq(x,x,1,3,1) \Rightarrow listx$ {1, 2, 3}

 $\{1,4.1,8.8\} \Rightarrow \text{listy}$ $\{1,4.1,8.8\}$

Quelle:

L.Paditz (Mitautor - Teil Stochastik)

Lehr- und Übungsbuch Mathematik Teil: Bd. 3.,

Lineare Algebra - Stochastik: mit 133 Beispielen und

249 Aufgaben mit Lösungen, 2. Aufl. 2001 (Hanser-Verl.)

ISBN: 978-3-446-21682-2

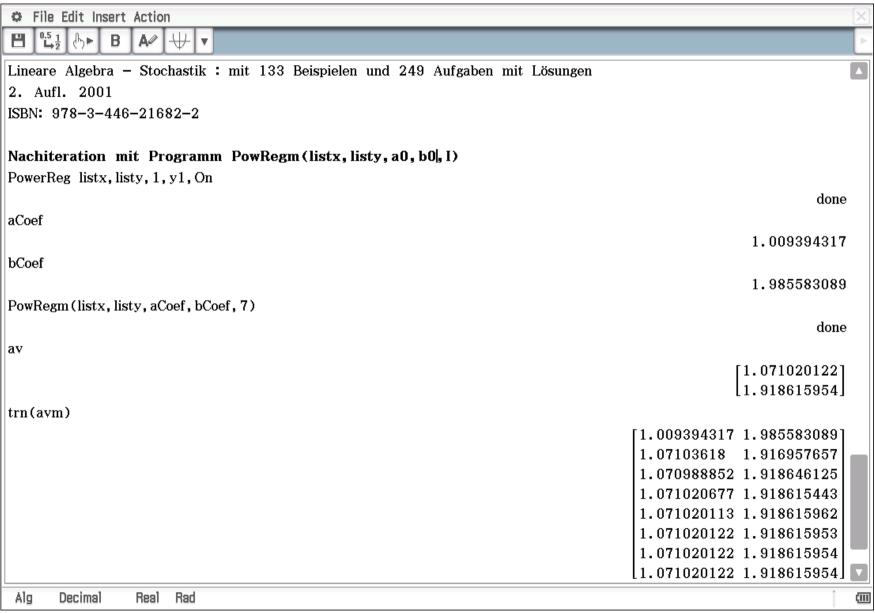
Fazit: die quasilin. Reg. ergibt nur eine Näherungslösung!

Prof. Dr. Ludwig Paditz Page 23 13.01.2017



❖ File Edit Insert Action ×
B A B A B A
$ \sum_{i=1}^{3} \left((\text{listy[i]-m*listx[i]^k})^2 \right) \{k=1.985583089, m=1.009394317\} $
0.03071481318
Offenbar ergeben das Ausgangsmodell (Power Reg) und das quasilineare Modell (Linear Reg mit transformierten Daten) das gleiche Ergebnis. Power Reg rechnet intern quasilinear!
Berechnung im Ausgangsmodell nach MKQ:
Define $F(m,k) = \sum_{i=1}^{3} ((listy[i]-m*listx[i]^k)^2)$
done
$\frac{\mathrm{d}}{\mathrm{dm}}(\mathrm{F}(\mathrm{m},\mathrm{k})) = 0 \Rightarrow \mathrm{Gl}1$
$0.1 \cdot (10 \cdot 2^{2 \cdot k+1} \cdot m + 20 \cdot 3^{2 \cdot k} \cdot m - 41 \cdot 2^{k+1} - 176 \cdot 3^{k} + 20 \cdot m - 20) = 0$
$\frac{\mathrm{d}}{\mathrm{dk}}(\mathrm{F}(\mathrm{m},\mathrm{k})) = 0 \Rightarrow \mathrm{Gl}2$
$0.1 \cdot (6.931471806 \cdot 2^{2 \cdot k+1} \cdot m^2 + 21.97224577 \cdot 3^{2 \cdot k} \cdot m^2 - 28.4190344 \cdot 2^{k+1} \cdot m - 193.3557628 \cdot 3^k \cdot m) = 0$
$\begin{bmatrix} Gl1 \\ Gl2 \\ m, k \end{bmatrix}$
{k=1.918615954, m=1.071020122}
F(m,k) {k=1.918615954, m=1.071020122} 7.852203424E-3
Alg Decimal Real Rad









Prof. Dr. Ludwig Paditz Page 26 13.01.2017