# KLEINSTE-QUADRATE-SCHÄTZUNG EINER DOPPELPARABEL - MODELL MIT FEHLERN IN DEN VARIABLEN.

VOLKER DRYGALLA und SIEGFRIED SCHMERLING

*Inst. für Numerische Mathematik, Fachbereich Mathematik und Informatik, Martin-Luther-Universität Halle-Wittenberg, Theodor-Lieser-Str. 5, D-06120 Halle(Saale)*
*email: drygalla@mail.mathematik.uni-halle.de, schmerling@mathematik.uni-halle.de*

ZUSAMMENFASSUNG. Vorliegende Meßpunkte $(x_i, y_i)$, $i = 1, ..., n$, deren graphische Darstellung näherungsweise eine Doppel-Parabel ergibt, werden durch eine Doppel-Parabel approximiert. Dabei wird berücksichtigt, daß sowohl die $x_i$ als auch die $y_i$ mit Meßfehlern behaftet sind (Schätzung eines Modells mit Fehlern in den Variablen). Die numerische Lösung des Kleinsten-Quadrate-Problems erfolgt nach dem Gauß-Newton-Verfahren, wobei die spezielle Struktur der auftretenden Jacobi-Matrix auf Vereinfachungen führt.

ABSTRACT. Let $(x_i, y_i)$, $i = 1, ..., n$ be measured points which demonstrate an approximate double parabola . These points are to be approximated by a double parabola. Because measuring errors are in both $x_i$ and $y_i$ there is here a model with errors in the variables. The LSE problem will be numerical solved by a Gauss-Newton algorithm. A special structure of the Jacobi matrix results in numerical simplifications.

*MSC2000 subject classification: 62J02, 62F10, 65D10*

**Key words: functional relationship, fitting of a double parabola, least squares, Gauss-Newton-Iteration**

## 1. Einleitung

Die Schätzung funktionaler Beziehungen im Sinne von M. Kendall (1951, 1952) mit Hilfe der Methode der kleinsten Quadrate führt im Falle einer Doppel-Parabel auf die Lösung nichtlinearer Gleichungssysteme. Beide Parabeläste gleichzeitig zu approximieren ist nach unserem Wissen bisher nicht versucht worden.

Die numerische Lösung des Kleinsten-Quadrate-Problems erfolgt mit dem Gauß-Newton-Verfahren, wobei unter Berücksichtigung der speziellen Struktur der hierbei auftretenden Jacobi-Matrix wesentliche Vereinfachungen erzielt werden.

## 2. Problemstellung

Eine Doppel-Parabel kann in Parameterdarstellung durch

$$\begin{aligned} y &= p \\ x &= x_0 + a + b(p - y_0)^2 \quad \text{rechter Zweig} \\ x &= x_0 - a - b(p - y_0)^2 \quad \text{linker Zweig} \end{aligned}$$

mit dem Parameter $p \in R^1$ dargestellt werden. Die nachfolgende Abbildung gibt ein Beispiel dafür an:

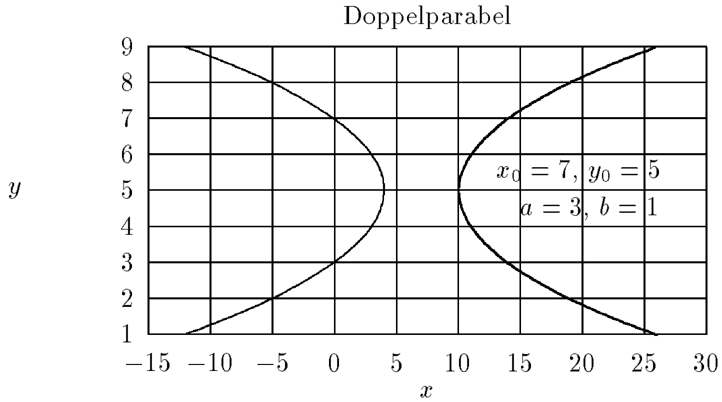


Abb. 1: Darstellung einer Doppelparabel gemäß obiger Parameterdarstellung

Die vorliegenden Meßwertepaare $(x_i, y_i)$ sollen die Darstellung

$$x_i = \xi_i + u_i \quad \text{und} \quad y_i = \eta_i + v_i \tag{2.1}$$

mit den „wahren“ Werten $\xi_i, \eta_i$ und den Meßfehlern $u_i, v_i$ für alle $i = 1, ..., n$ besitzen. Zwischen den fehlerfreien Größen $\xi_i$ und $\eta_i$ besteht die funktionale Beziehung

$$\eta_i = p_i \tag{2.2}$$

$$\xi_i = x_0 + sign(\xi_i - x_0) \cdot (a + b(p_i - y_0)^2) \tag{2.3}$$

für alle $i$ (Doppelparabel).

Die Approximationsaufgabe unter der Annahme gleicher Meßgenauigkeit von $x_i$ und $y_i$, und unter der Annahme, daß $sign(\xi_i - x_0) = sign(x_i - x_0)$ für alle $i = 1, ..., n$ gilt, kann durch

$$Q := \sum_{i=1}^{n} \left(x_i - x_0 - sign(x_i - x_0)\left(a + b(p_i - y_0)^2\right)\right)^2 + \tag{2.4}$$
$$+ \sum_{i=1}^{n} (y_i - p_i)^2 \Longrightarrow Minimum$$

formuliert werden.
In der Minimierungsaufgabe (2.4) ist $Q = Q(x_0, y_0, a, b, p_1, ..., p_n)$ eine Funktion von $n + 4$ unbekannten Parametern.
Gesucht sind Schätzwerte $\hat{x_0}$, $\hat{y_0}$, $\hat{a}$, $\hat{b}$, $\hat{p_1}$, ...,$\hat{p_n}$.

Weil für viele praktische Beispiele aus den Streubildern der Meßpunkte ersichtlich wurde, daß in einigen Fällen die Hauptachse der Doppelparabel leicht gedreht war (Wussling et al., 1997), soll noch eine Drehung um den Drehpunkt $(x_0, y_0)$ mit dem Drehwinkel $\phi$ in die Aufgabenstellung einbezogen werden.

Mit

$$\xi' = x_0 + (x_i - x_0) \cdot \cos\phi + (y_i - y_0) \cdot \sin\phi \tag{2.5}$$

$$\eta' = y_0 - (x_i - x_0) \cdot \sin\phi + (y_i - y_0) \cdot \cos\phi \tag{2.6}$$

erhält man

$$Q' := \sum_{i=1}^{n} \left((x_i - x_0)\cos\phi + (y_i - y_0)\sin\phi - sign(x_i - x_0) \cdot \left(a + b(p_i - y_0)^2\right)\right)^2 + \tag{2.7}$$
$$+ \sum_{i=1}^{n} \left(-(x_i - x_0)\sin\phi + (y_i - y_0)\cos\phi + y_0 - p_i\right)^2 \Longrightarrow Min.$$

Damit ist die Anzahl zu schätzender Parameter um den Parameter $\phi$ erhöht worden.
Die Optimierungsaufgabe (2.7) soll gelöst werden.

## 3. Numerische Lösung

Die zu lösende Optimierungsaufgabe führt auf ein nichtlineares Gleichungssystem für die zu schätzenden Parameter $p_1, ..., p_n$, $x_0$, $y_0$, $a$, $b$, und $\phi$.
Bezeichne $\underline{x} = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix}$ mit $\underline{x}_1 := (p_1, ..., p_n)^\top$ und $\underline{x}_2 := (x_0, y_0, a, b, \phi)^\top$ den $(n+5)-$Parametervektor und sei durch $F(\underline{x}) = \begin{pmatrix} F_1(\underline{x}) \\ F_2(\underline{x}) \end{pmatrix}$ mit

$$F_1(\underline{x}) := \tag{3.1}$$

$$= \begin{pmatrix} (x_1 - x_0)\cos\phi + (y_1 - y_0)\sin\phi - sign(x_1 - x_0)\cdot(a + b(p_1 - y_0)^2) \\ \vdots \\ (x_n - x_0)\cos\phi + (y_n - y_0)\sin\phi - sign(x_n - x_0)\cdot(a + b(p_n - y_0)^2) \end{pmatrix}$$

und

$$F_2(\underline{x}) := \begin{pmatrix} -(x_1 - x_0)\sin\phi + (y_1 - y_0)\cos\phi + y_0 - p_1 \\ \vdots \\ -(x_n - x_0)\sin\phi + (y_n - y_0)\cos\phi + y_0 - p_n \end{pmatrix} \tag{3.2}$$

ein Funktionsvektor definiert.
Dann kann die Optimierungsaufgabe (2.7) geschrieben werden als

$$Q' = \|F(\underline{x})\|_2^2 \Longrightarrow Minimum \tag{3.3}$$

Zur numerischen Lösung von (3.3) soll das Gauß-Newton-Verfahren herangezogen werden (vgl. z.B. Schwetlick, 1979).
Dabei hat man die Iterationsgleichung

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \left[\left\{\frac{\partial F(\underline{x}^{(k)})}{\partial \underline{x}}\right\}^\top \cdot \frac{\partial F(\underline{x}^{(k)})}{\partial \underline{x}}\right]^{-1} \cdot \left\{\frac{\partial F(\underline{x}^{(k)})}{\partial \underline{x}}\right\}^\top \cdot F(\underline{x}^{(k)}) \tag{3.4}$$

für $k = 0, 1, 2, ...$ zu lösen.
Die hierbei auftretende $(2n \times (n+5))$ Jacobi-Matrix $J \equiv \frac{\partial F(\underline{x}^{(k)})}{\partial \underline{x}}$ hat die Struktur

$$J = \begin{pmatrix} D_1 & A \\ D_2 & B \end{pmatrix}, \tag{3.5}$$

wobei $D_1$ und $D_2$ nichtsinguläre $(n \times n)$ Diagonalmatrizen sind und $A$ und $B$ $(n \times 5)$ Blöcke sind.
Um die Inverse von $J^\top J$ zu bestimmen, wird zunächst die orthogonale Matrix

$$O := \begin{pmatrix} (D_1^2 + D_2^2)^{-\frac{1}{2}} D_1 & (D_1^2 + D_2^2)^{-\frac{1}{2}} D_2 \\ -(D_1^2 + D_2^2)^{-\frac{1}{2}} D_2 & (D_1^2 + D_2^2)^{-\frac{1}{2}} D_1 \end{pmatrix} \tag{3.6}$$

gebildet.
Die Berechnung der Produkte $O \cdot J$ und $O \cdot F$ ergibt dann

$$O \cdot J = \begin{pmatrix} (D_1^2 + D_2^2)^{\frac{1}{2}} & (D_1^2 + D_2^2)^{-\frac{1}{2}} (D_1 A + D_2 B) \\ 0 & (D_1^2 + D_2^2)^{-\frac{1}{2}} (-D_2 A + D_1 B) \end{pmatrix} \tag{3.7}$$

und

$$O \cdot F = \begin{pmatrix} (D_1^2 + D_2^2)^{-\frac{1}{2}} (D_1 F_1 + D_2 F_2) \\ (D_1^2 + D_2^2)^{-\frac{1}{2}} (-D_2 F_1 + D_1 F_2) \end{pmatrix}, \tag{3.8}$$

welche mit den Abkürzungen

$$\begin{aligned} D &:= D_1^2 + D_2^2 \\ \overline{A} &:= D_1 A + D_2 B \\ \overline{B} &:= -D_2 A + D_1 B \\ \overline{F_1} &:= D_1 F_1 + D_2 F_2 \\ \overline{F_2} &:= -D_2 F_1 + D_1 F_2 \end{aligned}$$

auch als

$$\overline{J} := O \cdot J = \begin{pmatrix} D^{\frac{1}{2}} & D^{-\frac{1}{2}} \overline{A} \\ 0 & D^{-\frac{1}{2}} \overline{B} \end{pmatrix} \tag{3.9}$$

und als

$$O \cdot F = \begin{pmatrix} D^{-\frac{1}{2}}\overline{F_1} \\ D^{-\frac{1}{2}}\overline{F_2} \end{pmatrix} \tag{3.10}$$

geschrieben werden können.
Dann ist

$$J^\top J = J^\top O^\top O J = \overline{J}^\top \overline{J} = \begin{pmatrix} D & \overline{A} \\ \overline{A}^\top & \overline{A}^\top D^{-1}\overline{A} + \overline{B}^\top D^{-1}\overline{B} \end{pmatrix}. \tag{3.11}$$

Nach der Frobenius-Formel (vgl. F.R. Gantmacher, 1988) ist dann

$$(J^\top J)^{-1} = (\overline{J}^\top \overline{J})^{-1} =$$
$$= \begin{pmatrix} D^{-1} + D^{-1}\overline{A}(\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{A}^\top D^{-1} & -D^{-1}\overline{A}(\overline{B}^\top D^{-1}\overline{B})^{-1} \\ -(\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{A}^\top D^{-1} & (\overline{B}^\top D^{-1}\overline{B})^{-1} \end{pmatrix}.$$

Wegen

$$J^\top F = J^\top O^\top O F = \overline{J}^\top O F = \begin{pmatrix} \overline{F}_1^\top \\ \overline{A}^\top D^{-1}\overline{F}_1 + \overline{B}^\top D^{-1}\overline{F}_2 \end{pmatrix}$$

folgt

$$(J^\top J)^{-1} J^\top F = \begin{pmatrix} D^{-1}\overline{F}_1 - D^{-1}\overline{A}(\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{B}^\top D^{-1}\overline{F}_2 \\ (\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{B}^\top D^{-1}\overline{F}_2 \end{pmatrix}. \tag{3.12}$$

Somit bekommt die Iterationsgleichung (3.4) die Form

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \begin{pmatrix} D^{-1}\overline{F}_1 - D^{-1}\overline{A}(\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{B}^\top D^{-1}\overline{F}_2 \\ (\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{B}^\top D^{-1}\overline{F}_2 \end{pmatrix}, \tag{3.13}$$

welche wegen der möglichen Aufspaltung von $\underline{x}$ in zwei simultane Iterationsgleichungen zerfällt:

$$\underline{x}_1^{(k+1)} = \underline{x}_1^{(k)} - \left(D^{-1}\overline{F}_1 - D^{-1}\overline{A}(\overline{B}^\top D^{-1}\overline{B})^{-1}\ \overline{B}^\top D^{-1}\overline{F}_2\right) \tag{3.14}$$

und

$$\underline{x}_2^{(k+1)} = \underline{x}_2^{(k)} - (\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{B}^\top D^{-1}\overline{F}_2 \tag{3.15}$$

Die Iterationsgleichungen (3.14) und (3.15) erleichtern die numerische Behandlung.
Nach Wahl der Startwerte $\underline{x}_1^{(0)}$ und $\underline{x}_2^{(0)}$, die auch mit einem Hilfsprogramm automatisch gefunden werden können, berechnet man zuerst den fünfdimensionalen Vektor $V := (\overline{B}^\top D^{-1}\overline{B})^{-1}\overline{B}^\top D^{-1}\overline{F}_2$, der danach in die Iterationsgleichungen (3.14) und(3.15) eingesetzt wird:

$$\underline{x}_1^{(k+1)} = \underline{x}_1^{(k)} - \left(D^{-1}\overline{F}_1 - D^{-1}\overline{A}\cdot V\right) \tag{3.16}$$

$$\underline{x}_2^{(k+1)} = \underline{x}_2^{(k)} - V \tag{3.17}$$

Auf diese Weise werden nur $\mathcal{O}(n)$ Rechenoperationen pro Iterationsschritt benötigt.

## 4. Beispiel

Die Beobachtung von Calcium-evozierten Fluoreszensen in lebenden Herzzellen ergab eine Kontur, die näherungsweise aus einer Doppel-Parabel besteht (vgl. Wussling and Salz, 1996, und die dort angebene Literatur). Die Meßdaten[1] sind:

| x | y | x | y | x | y | x | y | x | y | x | y | x | y | x | y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 520 | 607 | 535 | 615 | 543 | 629 | 537 | 643 | 621 | 646 | 606 | 639 | 600 | 625 | 613 | 615 |
| 521 | 607 | 536 | 616 | 544 | 630 | 536 | 644 | 620 | 646 | 605 | 638 | 600 | 624 | 614 | 615 |
| 522 | 608 | 537 | 617 | 544 | 631 | 535 | 644 | 619 | 645 | 604 | 638 | 600 | 623 | 615 | 614 |
| 523 | 608 | 538 | 617 | 544 | 632 | 534 | 645 | 618 | 645 | 603 | 637 | 601 | 622 | 616 | 614 |
| 524 | 608 | 539 | 618 | 544 | 633 | 533 | 646 | 617 | 644 | 602 | 636 | 602 | 621 | 617 | 614 |
| 525 | 608 | 539 | 619 | 544 | 634 | 532 | 647 | 616 | 644 | 601 | 635 | 603 | 621 | 618 | 613 |
| 526 | 609 | 540 | 620 | 543 | 635 | 531 | 648 | 615 | 644 | 601 | 634 | 604 | 620 | 619 | 613 |
| 527 | 609 | 541 | 621 | 543 | 636 | 530 | 648 | 614 | 643 | 600 | 633 | 605 | 620 | 620 | 613 |
| 528 | 610 | 541 | 622 | 543 | 637 | 529 | 649 | 613 | 643 | 600 | 632 | 606 | 619 | 621 | 612 |
| 529 | 611 | 541 | 623 | 543 | 638 | 528 | 649 | 612 | 643 | 599 | 631 | 607 | 618 | 622 | 611 |
| 530 | 611 | 542 | 624 | 542 | 639 | 527 | 650 | 611 | 642 | 599 | 630 | 608 | 618 | 623 | 611 |
| 531 | 612 | 542 | 625 | 541 | 640 | 526 | 650 | 610 | 642 | 598 | 629 | 609 | 617 | | |
| 532 | 613 | 543 | 626 | 540 | 640 | 525 | 651 | 609 | 641 | 598 | 628 | 610 | 617 | | |
| 533 | 614 | 543 | 627 | 539 | 641 | 524 | 651 | 608 | 641 | 598 | 627 | 611 | 616 | | |
| 534 | 614 | 543 | 628 | 538 | 642 | 523 | 652 | 607 | 640 | 599 | 626 | 612 | 616 | | |

Die folgende Abbildung zeigt die graphische Darstellung der Meßpunkte zusammen mit der angepassten Doppel-Parabel.

[1] Wir danken Herrn Prof. Dr. M. Wussling vom Julius-Bernstein-Institut für Physiologie für die Bereitstellung der Meßwerte.

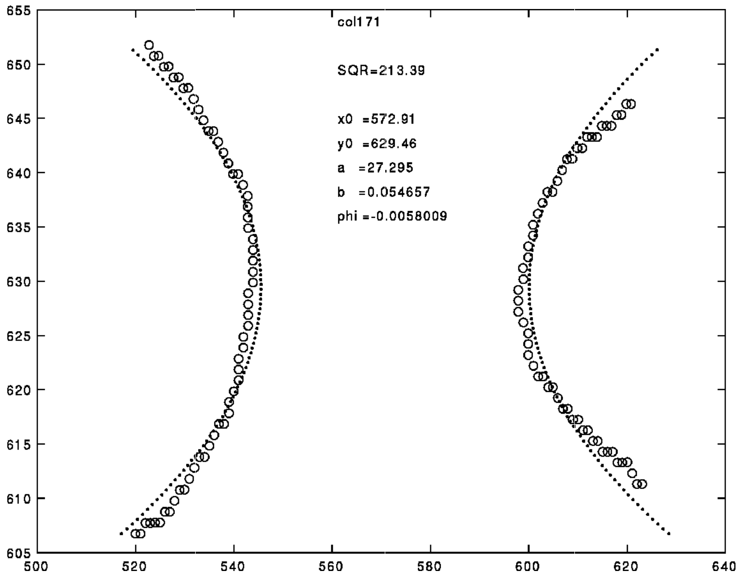


Abb. 2: Graphische Darstellung der Meßpunkte (∘) und der approximierenden Doppel-Parabel. Der Name des verwendeten Datenfiles (col171), der Betrag der erzielten quadratischen Restsumme SQR und die geschätzten Parameterwerte sind in der Abbildung aufgeschrieben.

Die Ausführung der Berechnungen gemäß den Iterationsgleichungen (3.16) und (3.17) erfolgten mit einem selbst erstellten m-File im Programm-Paket „MATLAB" . Im obigen Beispiel waren nur fünf Iterationsschritte erforderlich.

## Literatur

1. F.R. Gantmacher: *Theorie der Matrizen.* Verlag Nauka Moskau 1988 (russisch).
2. M.G. Kendall: *Regression, structure and functional relationship. Part I.* Biometrika 38 (1951) 11-25.
3. M.G. Kendall: *Regression, structure and functional relationship. Part II.* Biometrika 39 (1952) 96-108.
4. *MATLAB - Reference Manual.* The MathWorks, Inc., Natick, Mass. USA 1994.
5. H. Schwetlick : *Numerische Lösung nichtlinearer Gleichungen.* VEB Deutscher Verlag der Wissenschaften Berlin 1979.
6. M.H.P. Wussling and H. Salz: *Nonlinear propagation of spherical calcium waves in rat cardiac myocytes.* Biophysical Journal **70**(1996)1144 -1153.

7. M. H. P. Wussling, K. Scheufler, S. Schmerling and V. Drygalla: *Velocity-Curvature Relationship of colliding Spherical Calcium waves in rat cardiac myocytes.* Biophysical Journal **73**(1997)1232 -1242.

## 5. Anhang: MATLAB m-File

Im nachfolgenden m-File werden:

1. eine Ausgangsnäherung für die Parameter der gesuchten Doppelparabel bestimmt.
(Dabei wird vorausgesetzt, daß zwei Parabeläste vorhanden sind, die etwa gleich viele Meßpunkte ober- und unterhalb der Hauptachse sowie links und rechts der zur Hauptachse senkrechten Achse tragen.
Unter diesen Voraussetzungen sind die Anfangswerte für $x_0$ das arithmetische Mittel der $x_i$, für $y_0$ das arithmetische Mittel der $y_i$, für $a$ das Minimum der Beträge der Abweichungen von $x_i - x_0$ und für $b$ die Summe der Beträge der Abweichungen $|x_i - x_0| - a$ dividiert durch die Summe von $(y_i - y_0)^2$.
Bei anderen Meßpunktanordnungen wird die programmierte Startwertschätzung nicht erfolgreich sein.)

2. eine vorgebbare obere Schranke (`tol`) für die relative Änderung der quadratischen Restsummen (SQR) zweier aufeinanderfolgender Iterationsschritte berücksichtigt.

3. eine vorgebbare Anzahl (`maxit`) von Gauß-Newton-Schritten gemäß (3.16) und (3.17) ausgeführt.

Der Aufruf erfolgt mit `gnpar('datname',tol ,maxit)`, wobei die Meßdaten in einem Datenfile `datname.txt` enthalten sind (zwei Spalten, erste Spalte $x-$Werte, zweite Spalte $y-$Werte, getrennt durch eine Spalte Leerzeichen).

Nach Ablauf der vorgesehenen Anzahl von Iterationsschritten erscheint eine Abbildung der Meßpunkte zusammen mit der angepaßten Doppel-Parabel und den geschätzten Parameterwerten auf dem Bildschirm (vgl. obige Abbildung).

**gnpar.m:**

```
function x=gnpar(matrn,tol,maxit);
 T=['load ' matrn '.txt'];
 eval(T);
 matr=eval(matrn);

 n=max(size(matr));
 fx=matr(:,1);
```

```
fy=matr(:,2);
x0=sum(fx)/n;
y0=sum(fy)/n;
a=min(abs(fx-x0));
b=sum(abs(fx-x0)-a)/sum((fy-y0).^2);
p=fy;
phi=0;
x=[p;x0;y0;a;b;phi];
e=ones(n,1);z=zeros(n,1);

iter=1;
s=sin(phi);c=cos(phi);
fxr= c*(fx-x0)+s*(fy-y0)+x0;
fyr=-s*(fx-x0)+c*(fy-y0)+y0;
sig=sign(fxr-x0);
F1=fxr-x0-a*sig-b*sig.*(p-y0).^2;
F2=fyr-p;
SQR_old=F1'*F1+F2'*F2;
SQR=0;

while  abs(SQR-SQR_old)/SQR_old > tol & iter < maxit

 if iter>2,SQR_old=SQR;end
 D1=-2*b*sig.*(p-y0);
 D2=-e;
 D=(D1.*D1+D2.*D2).^(1/2);
 D1M=D1./D;
 D2M=D2./D;

 A=[-c*e -s*e+2*b*sig.*(p-y0) -sig -sig.*(p-y0).^2 fyr-y0];
 B=[ s*e -(c-1)*e              z    z              -fxr+x0];

 for j=1:5;
  AN(:,j)= D1M.*A(:,j)+D2M.*B(:,j);
  BN(:,j)=-D2M.*A(:,j)+D1M.*B(:,j);
 end

 F1N= D1M.*F1+D2M.*F2;
 F2N=-D2M.*F1+D1M.*F2;
 V=BN\F2N;
```

```
 searchd=[(F1N-AN*V)./D;V];

 lambda=2.0;

 iin=0; iter=iter+1;

 SQR=SQR_old+1;

 while SQR > SQR_old
   iin=iin+1;
   lambda=lambda/2;
   xt=x-lambda*searchd;

   p=xt(1:n);x0=xt(n+1);y0=xt(n+2);a=xt(n+3);
   b=xt(n+4);phi=xt(n+5);
   s=sin(phi);c=cos(phi);
   fxr= c*(fx-x0)+s*(fy-y0)+x0;
   fyr=-s*(fx-x0)+c*(fy-y0)+y0;
   sig=sign(fxr-x0);
   F1=fxr-x0-a*sig-b*sig.*(p-y0).^2;
   F2=fyr-p;
   SQR=F1'*F1+F2'*F2;

   if (iin > 10)
   disp('vorgebene Toleranz nicht erreichbar, erhoehe tol!')
   return; end
 end

 x=xt;
 if iter==1,SQR_old=SQR;SQR=2*SQR;end
 [iter,SQR]
end

if iter==maxit &  abs(SQR-SQR_old)/SQR_old > tol
disp('vorgebene Toleranz nicht erreicht, erhoehe maxit!')
return; end
```

```
p=x(1:n);x0=x(n+1);y0=x(n+2);a=x(n+3);b=x(n+4);phi=x(n+5);
ymi=min(fyr); xmi=min(fxr);
yma=max(fyr); xma=max(fxr);
for kk=1:100,py(kk)=ymi+(kk-1)*(yma-ymi)/100;end;
pxr=x0+a+b*(py-y0).^2;
pxl=x0-a-b*(py-y0).^2;
plot(fxr,fyr,'ro',pxl,py,'b.',pxr,py,'b.')
kx=xmi+0.8*(xma-xmi)/2;ky=yma;
kk=(ky-ymi)/20;
text(kx,ky,matrn);
text(kx,ky-kk*2,  ['SQR ='  num2str(SQR,5)]);
text(kx,ky-kk*4,['x0  ='  num2str(x0,5)]);
text(kx,ky-kk*5,['y0  ='  num2str(y0,5)]);
text(kx,ky-kk*6,['a   ='  num2str(a,5)]);
text(kx,ky-kk*7,['b   ='  num2str(b,5)]);
text(kx,ky-kk*8,['phi ='  num2str(phi,5)]);
```