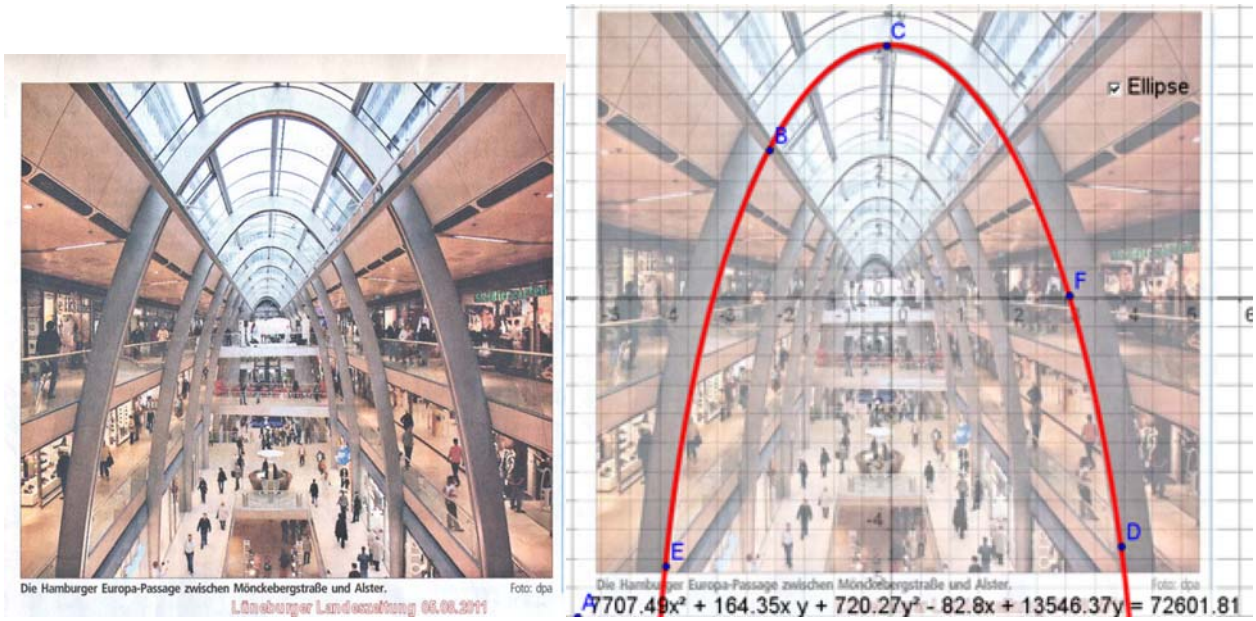
Ellipse Moenkebergpassage

Prof. Dr. Dörte Haftendorn: Mathematik mit MuPAD 4 Juli 07 Update 5.8.2011
Web: http://haftendorn.uni-lueneburg.de www.mathematik-verstehen.de

##



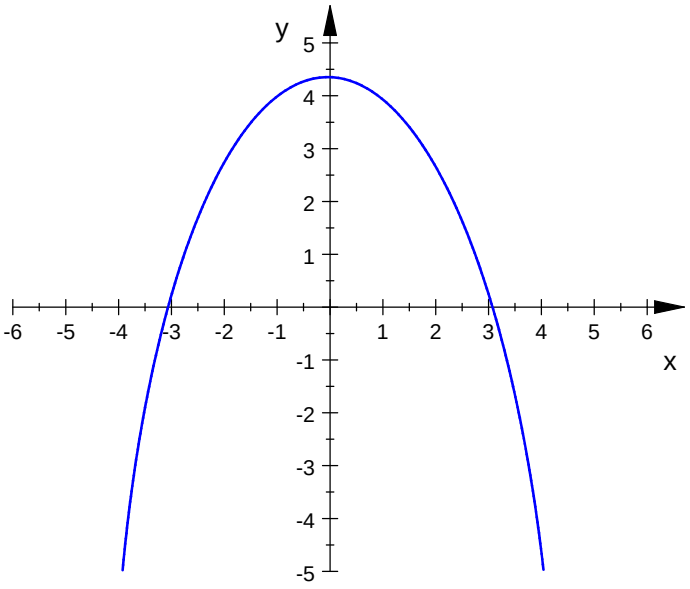
---

c: 7707.49x² + 164.35x y + 720.27y² - 82.8x + 13546.37y = 72601.81

```
keg:=matrix([[7707.49*x^2 + 164.35*x *y + 720.27*y^2 -
82.8*x + 13546.37*y - 72601.81]])
```

$$\left(7707.49 \cdot x^2 + 164.35 \cdot x \cdot y - 82.8 \cdot x + 720.27 \cdot y^2 + 13546.37 \cdot y - 72601.81\right)$$

```
kegp:=plot::Implicit2d(keg[1]=0,x=-6..6,y=-5..5):
plot(kegp,Scaling=Constrained):
```



1

Dieser Kegelschnitt ist offenbar eine Ellipse.

Durchführung einer Hauptachsentransformation

Durchführung einer Hauptachsentransformation

c: 7707.49x² + 164.35x y + 720.27y² - 82.8x + 13546.37y = 72601.81

```
A:= matrix([[7707.49,164.35/2],[164.35/2,720.27]]);
a:=matrix([-82.8,13546.37]): at:=linalg::transpose(a);
d:=-72601.81;
```

$$\begin{pmatrix} 7707.49 & 82.175 \\ 82.175 & 720.27 \end{pmatrix}$$

$$\begin{pmatrix} -82.8 & 13546.37 \end{pmatrix}$$

$$-72601.81$$

Bestimmung der Achsenrichtungen:

```
evli:=linalg::eigenvectors(A)
```

$$\left[\left[7708.456307,\ 1,\ \left[\begin{pmatrix} 0.9999308686 \\ 0.01175831833 \end{pmatrix}\right]\right],\ \left[719.3036934,\ 1,\ \left[\begin{pmatrix} 0.01175831833 \\ -0.9999308686 \end{pmatrix}\right]\right]\right]$$

Eigenwerte mit ihrer Vielfachheit und ihrem Eigenvektor.

```
ev1:=evli[1][3][1];
ev2:=-evli[2][3][1];
```

$$\begin{pmatrix} 0.9999308686 \\ 0.01175831833 \end{pmatrix}$$

$$\begin{pmatrix} -0.01175831833 \\ 0.9999308686 \end{pmatrix}$$

```
ew1:=evli[1][1];
ew2:=evli[2][1];
```

$$7708.456307$$

$$719.3036934$$

```
linalg::det(ev1.ev2)
```

$$1.0$$

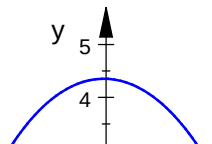
Wenn diese Determinate neagtiv ist, tauscht man die EV besser um.
Oder man nimmt einen der EV andersherum, Rechtsystem ist günstig.
Anderenfalls ist noch eineSpiegelung im Spiel.

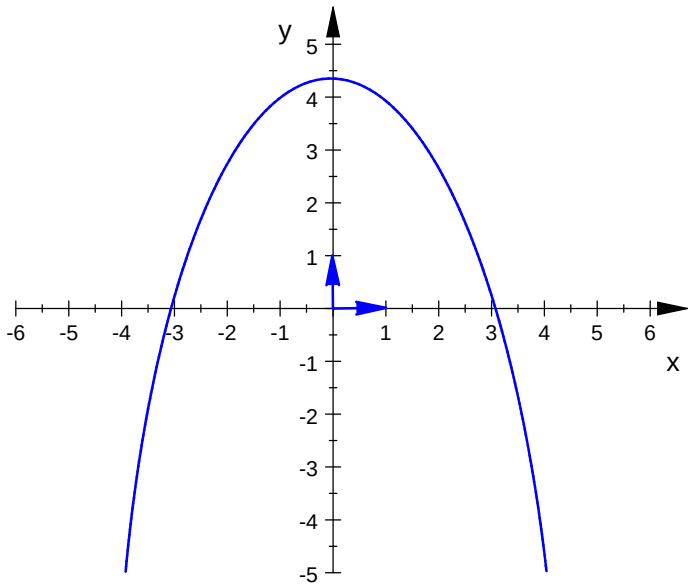
```
float(ev1),float(ev2)
```

$$\begin{pmatrix} 0.9999308686 \\ 0.01175831833 \end{pmatrix},\ \begin{pmatrix} -0.01175831833 \\ 0.9999308686 \end{pmatrix}$$

Einzeichnen der Eigenvektoren:

```
ev1p:=plot::Arrow2d(ev1):
ev2p:=plot::Arrow2d(ev2):
plot(kegp,ev1p,ev2p, Scaling=Constrained)
```



Wie erwartet sind die Eigenvektoren die Richtungen der Hauptachsen des Kegelschnittes.

Bestimmung der Kegelschnittgleichung

Die Theorie sagt, dass die Eigenwerte nun die Faktoren der quadratischen Glieder werden.
Das heißt hier, dass es zwei quadratisches Glieder mit positivem Vorzeichen gibt.
Damit muss es sich um eine Ellipse handeln, wenn es nicht ein entarteter Sonderfall wird.
Da wir oben schon gezeichnet haben, war ja schon klar, dass es eine Ellipse ist.
Soll die gegebene blaue Ellipse in Hauptachsenlage abgebildet werden,
so kann das geschehen durch eine Drehung um den Ursprung, evt. verknüpft mit einer Achsenspiegelung an einer der Koordinatenachsen, gefolgt von einer Verschiebung.

$$\vec{p}' = P^T \vec{p} \quad \text{und} \quad \vec{p}'' = \vec{p}' + \vec{t}$$

Aus den normierten Eigenvektoren erstellt man eine Matrix P.
Die Eigenvektoren stehen gleich von alleine senkrecht aufeinander, wenn die Eigenwerte verschieden sind.
P ist dann eine Orthonormal-Matrix. Für solche ist die inverse Matrix
gleich der transponierten Matrix, bezeichnet mit Pt (oder mit hochstelltem T).
Die Transformationsmatrix, die den Kegelschnitt in eine Lage bewegt, bei der die Hauptachsen parallel
zu den Koordinatenachsen sind, ist Pt.
Die Theorie sagt nun, dass Pt*A*P die Diagonalmatrix aus den Eigenwerten ist.
Davon kann man sich hier überzeugen:

```
ev1n:=linalg::normalize(ev1):
ev2n:=linalg::normalize(ev2):
P:=ev1n.ev2n
```

$$\begin{pmatrix} 0.9999308686 & -0.01175831833 \\ 0.01175831833 & 0.9999308686 \end{pmatrix}$$

```
float(P)
```

$$\begin{pmatrix} 0.9999308686 & -0.01175831833 \\ 0.01175831833 & 0.9999308686 \end{pmatrix}$$

```
Pt:=linalg::transpose(P)
```

$$\begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}$$

$$\begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}$$

```
float(Pt)
```

$$\begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}$$

Diagonalisierung, die klappt immer, wenn die EV eine ONB bilden, wenn man also so ein P aufstellen kann.

```
Simplify(Pt*A*P);
float(Pt*A*P)
```

$$\begin{pmatrix} 7708.456307 & 2.602085214 \cdot 10^{-18} \\ 0 & 719.3036934 \end{pmatrix}$$

$$\begin{pmatrix} 7708.456307 & 2.602085214 \cdot 10^{-18} \\ 0 & 719.3036934 \end{pmatrix}$$

Evt. winzige 10^-19 Werte sind numerisch 0.

Vektorschreibweise für die Abbildung $\vec{p} = P\,\vec{p}'$ und die Quadrikgleichungen, die sich durch Einsetzen ergeben:

$$Q: \quad \vec{p}^{\,T} A\,\vec{p} + \vec{a}^{\,T}\vec{p} + d = 0$$

$$Q': \quad \vec{p}'^{\,T} P^T A\,P\,\vec{p}' + \vec{a}^{\,T} P\vec{p}' + d = 0$$

$$Q': \quad \vec{p}'^{\,T} D_{EW}\,\vec{p}' + \vec{a}^{\,T} P\vec{p}' + d = 0$$

mit $D_{EW} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$

Abbildung der linearen Terme erfordert also

```
at
```

$$\begin{pmatrix} -82.8 & 13546.37 \end{pmatrix}$$

```
atb:=at*P
```

$$\begin{pmatrix} 76.48825473 & 13546.40711 \end{pmatrix}$$

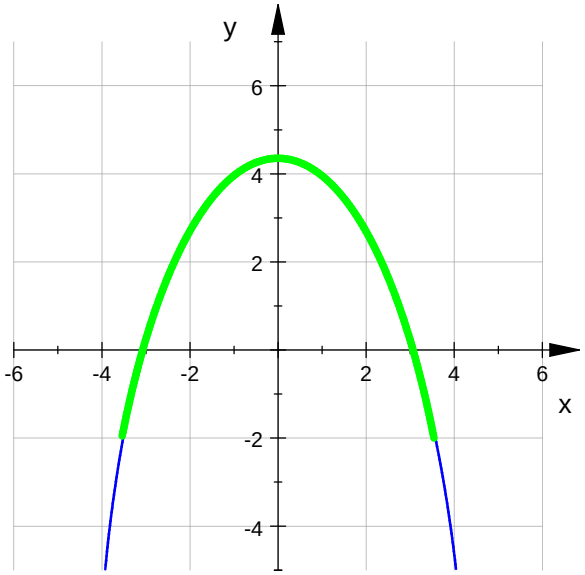
```
kegSD:=ew1*x^2+ew2*y^2+atb[1]*x+atb[2]*y+d
```

$$7708.456307 \cdot x^2 + 76.48825473 \cdot x + 719.3036934 \cdot y^2 + 13546.40711 \cdot y - 72601.81$$


```
kegSDp:=plot::Implicit2d(kegSD=0,x=-4..4,y=-2..7,

LineWidth=1,LineColor=[0,1,0],GridVisible=TRUE):
```

```
LineWidth=1,LineColor=[0,1,0],GridVisible=TRUE):
plot(kegp,kegSDp, Scaling=Constrained)
```



(Bei der Betrachtung dieses Bildes fällt auf, dass es sich um eine negative Drehung um den Ursprung handelt.)

```
Pt, float(Pt);
Dr:=phi->matrix([[cos(phi),-sin(phi)],
[sin(phi),cos(phi)]]): Dr(`&phiv;`);
```

$$\begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}, \begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}$$

$$\begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

Der Vergleich mit der allgemeinen Drehmatrix ergibt, dass es mit dem Anpassen von phi klappen kann:

```
cos(phi)=float(Pt[1,1]);sin(phi)=float(Pt[2,1]);
```

$$\cos(\mathrm{phi}) = 0.9999308686$$

$$\sin(\mathrm{phi}) = -0.01175831833$$

```
arccos((Pt[1,1]));
wi:=arcsin((Pt[2,1]));
```

$$0.01175858929$$

$$-0.01175858929$$

Daraus folgt umgerechnet ins Winkelmaß:

```
float(arccos((Pt[1,1]))/PI*180);
wig:=float(arcsin((Pt[2,1]))/PI*180);
```

$$0.6737175394$$

$$-0.6737175394$$

$$-0.6737175394$$

Damit kann die gemeinsame Lösung nur der negative Winkel wi sein.
So passt es auch zur Zeichnung.
Einzeichnen des Zwischenbildes:

```
p:=matrix([x,y]);
pt:=linalg::transpose(p);
```

$$\begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x & y \end{pmatrix}$$

```
float(Dr(wi)),float(Pt)
```

$$\begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}, \begin{pmatrix} 0.9999308686 & 0.01175831833 \\ -0.01175831833 & 0.9999308686 \end{pmatrix}$$

Beachte, dass die inverse Drehung die ist, die um den negativen Winkel dreht.

```
kegD:=simplify(pt*Dr(wi)*A*Dr(-wi)*p+at*Dr(-wi)*p+d);
```

$$\left(7708.456307 \cdot x^2 + 4.423544864 \cdot 10^{-17} \cdot x \cdot y + 76.48825472 \cdot x + 719.3036934 \cdot y^2 + 135 \right.$$

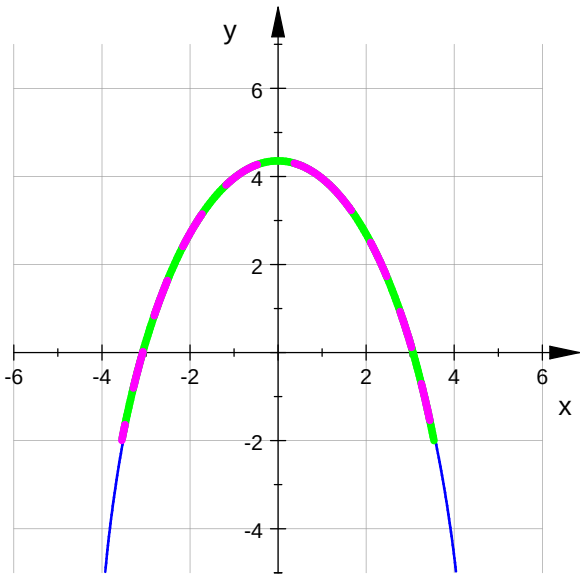
Dieses hat man ja auch als transformierte Gleichung erhalten.

```
kegSD
```

$$7708.456307 \cdot x^2 + 76.48825473 \cdot x + 719.3036934 \cdot y^2 + 13546.40711 \cdot y - 72601.81$$

```
kegDp:=plot::Implicit2d(kegD[1],x=-4..5,y=-2..5,

LineWidth=1,LineColor=[1,0,1],LineStyle=Dashed,GridVisible=TRUE):
plot(kegp,kegSDp,kegDp,Scaling=Constrained)
```



##################################

Bestimmung der Translation:


```
kegSD
```

$$7708.456307 \cdot x^2 + 76.48825473 \cdot x + 719.3036934 \cdot y^2 + 13546.40711 \cdot y - 72601.81$$

$$7708.456307 \cdot x^2 + 76.48825473 \cdot x + 719.3036934 \cdot y^2 + 13546.40711 \cdot y - 72601.81$$

Die Arbeitsweise ist dieselbe wie bei der Herstellung der Scheitelform einer Parabel.
Hier durch Hinsehen:

```
xterm:=hold(7708.456307*(x+1/2*76.48825473/7708.456307)^
2);expand(xterm);
yterm:=hold(719.3036930*(y+1/2*13546.40711/719.3036930)^
2);expand(yterm);
```

$$7708.456307 \cdot \left(x + \frac{\frac{76.48825473}{2}}{7708.456307}\right)^2$$

$$7708.456307 \cdot x^2 + 76.48825473 \cdot x + 0.1897413982$$

$$719.303693 \cdot \left(y + \frac{\frac{13546.40711}{2}}{719.303693}\right)^2$$

$$719.303693 \cdot y^2 + 13546.40711 \cdot y + 63778.74442$$

```
dd:=-0.1897413982-63779.74442+d
```

$$-136381.7442$$

Also

```
kegSDK:=xterm+yterm+dd
```

$$719.303693 \cdot (y + 9.416333631)^2 + 7708.456307 \cdot (x + 0.004961321157)^2 - 136381.7442$$

```
expand(kegSDK)    //Probe
```

$$7708.456307 \cdot x^2 + 76.48825473 \cdot x + 719.303693 \cdot y^2 + 13546.40711 \cdot y - 72602.81$$

c: 7707.49x² + 164.35x y + 720.27y² - 82.8x + 13546.37y = 72601.81

passt- bis auf Rundungsfehler.

Letzter Teil der Hauptachsentransformation ist die Translation t

```
t:=matrix([
1/2*76.48825473/7708.456307,1/2*13546.40711/719.3036930]
)
```

$$\begin{pmatrix} 0.004961321157 \\ 9.416333631 \end{pmatrix}$$

$\vec{p}'' = \vec{p}' + \vec{t}$ also $\vec{p}' = \vec{p}'' - \vec{t}$ Das ergibt:

```
kegH:=ew1*x^2+ew2*y^2+dd
```

$$7708.456307 \cdot x^2 + 719.3036934 \cdot y^2 - 136381.7442$$

Angabe der Gleichung in der üblichen Form:

```
ew1,ew2
```

7708.456307, 719.3036934

```
aq:=-dd/ew1; bq:=-dd/ew2;
sqrt(aq);sqrt(bq);
```

17.69248456

189.6024522

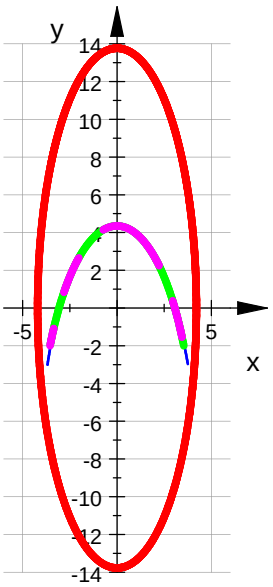
4.206243521

13.76962063

```
hold(x^2/4.2^2+y^2/13.8^2=1)
```

$$\frac{x^2}{4.2^2}+\frac{y^2}{13.8^2}=1$$

```
kegHp:=plot::Implicit2d(kegH,x=-5..5,y=-14..14,
LineWidth=1,LineColor=[1,0,0],GridVisible=TRUE):
plot(kegHp,kegp,kegSDp,kegDp,Scaling=Constrained)
```



Bestimmung des ursprünglichen Mittelpunktes:

$$\vec{m}''=\begin{pmatrix}0\\0\end{pmatrix},\quad \vec{m}'=-\vec{t},\quad \vec{m}=P(-\vec{t})$$

```
m:=P*(-t)
```

$$\begin{pmatrix}0.1057592701\\-9.415741003\end{pmatrix}$$

Bestimmung des Urbildes des eines Punktes der Parabelachse:

$$\vec{r}'' = \begin{pmatrix} r \\ 0 \end{pmatrix}, \vec{r}' = \begin{pmatrix} r \\ 0 \end{pmatrix} - \vec{t}, \vec{r} = P\begin{pmatrix} r \\ 0 \end{pmatrix} + \vec{m}, \; r = r \cdot \overrightarrow{ev_1} + \vec{m}$$

Mit ev1 rechts ist hier der normierte 1. Eigenvektor gemeint.P

```
//r:=sqrt(3): //auf der x-Achse
//rur:=r*ev1n+m;
```

$$\begin{pmatrix} 0.9999308686 \cdot \sqrt{3} + 0.1057592701 \\ 0.01175831833 \cdot \sqrt{3} - 9.415741003 \end{pmatrix}$$

```
mp:=plot::Point2d(m, PointColor=[0,1,1]):
msp:=plot::Point2d(-t, PointColor=[0,1,1]):
rurp:=plot::Point2d(rur):
rp:=plot::Point2d([r,0]):
ev1urp:=plot::Arrow2d(m,m+ev1):
ev2urp:=plot::Arrow2d(m,m+ev2):
tp:=plot::Arrow2d(-t,[0,0]):
plot(kegp,kegSDp,kegDp,kegHp, mp,msp,tp,
     ev1urp,ev2urp,PointSize=2,Scaling=Constrained)
```