Lehrsätze aus der ebenen Geometrie

mit DERIVE und CABRI

entstanden anlässlich eines Vortrags beim
Bundesseminar „Elektronische Lernmedien im Mathematikunterricht“
im März 2003 in Amstetten

Walter Wegscheider

KLASSISCHE SÄTZE DER GEOMETRIE

Betrachtung mit dynamischer Geometrie (**Cabri**) und analytisch/algebraisch (**Derive**)

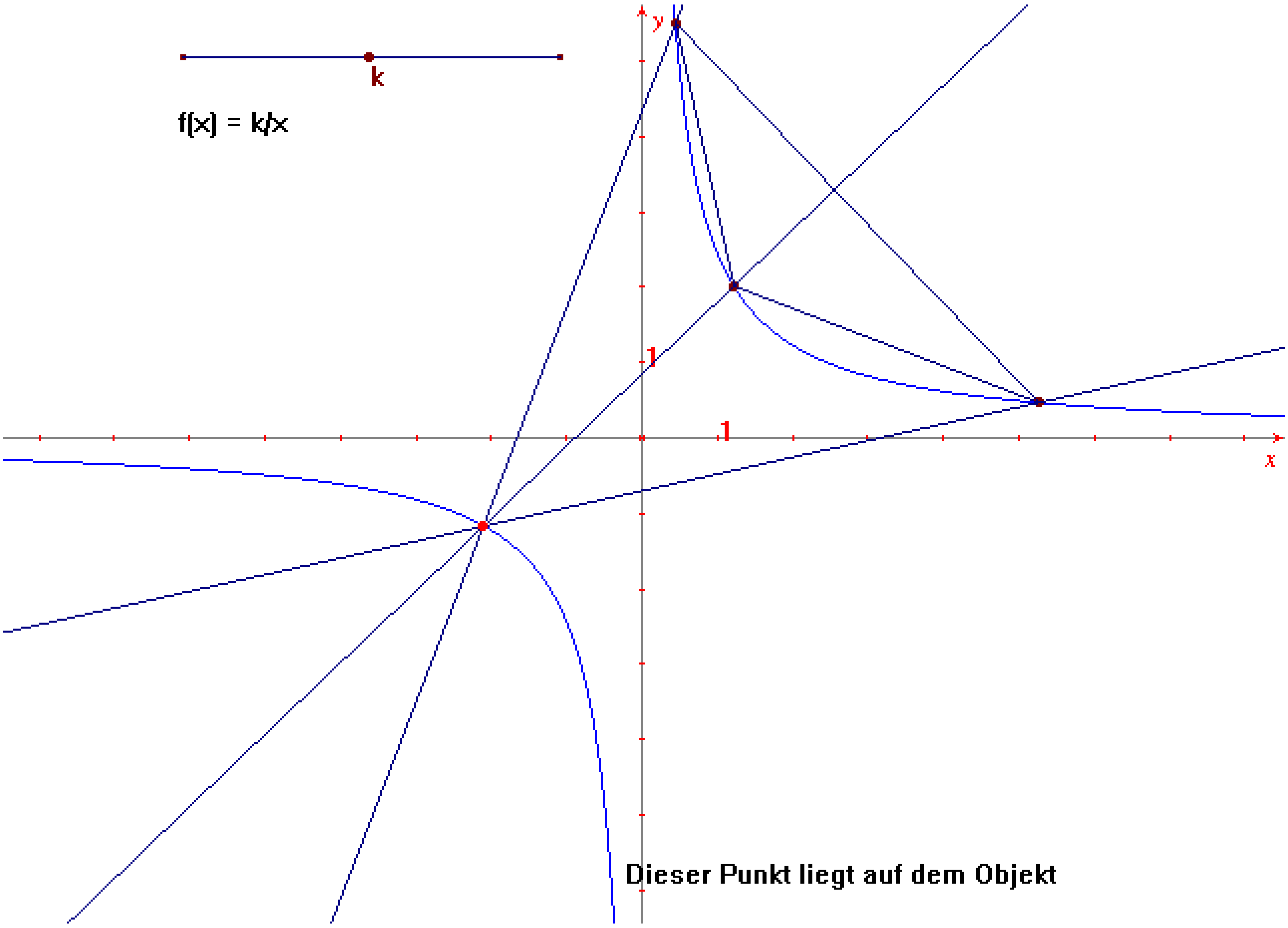
1. DREIECK – HYPERBEL:

Annahme A:

Der Höhenschnittpunkt eines Dreiecks, dessen Eckpunkte auf einer gleichseitigen (rechtwinkeligen) Hyperbel liegen, ist wieder ein Punkt auf der Hyperbel.

A.1. Geometrische Realisierung mit Cabri Geometré:

Wir betrachten den Höhenschnittpunkt eines Dreiecks, das auf dem Ast einer gleichseitigen Hyperbel liegt (Konstruktion über die Funktion: y = k/x). Anschließend können über den Zugmodus der Parameter k (in einem bestimmten – für die Sichtbarkeit der Hyperbel im Bildschirmfenster sinnvollen Intervall) und die Dreieckseckpunkte verändert werden. Cabri überprüft mit der Funktion <ELEMENT>, ob der konstruierte Höhenschnittpunkt tatsächlich auf der Hyperbel liegt.



A.2. Algebraische Überprüfung mit Derive:

Wir erzeugen eine Hyperbel der Form: y = k/x

Vorgang für spezielle Werte

- Zuerst erzeugen wir über den Zufallsgenerator den Parameter k im Intervall [-5; 5] und die drei x-Werte (der Eckpunkte des Dreiecks) im Intervall [-10; 10].

```
RANDOM(0)
k:=10*RANDOM(1)-5
xval:=VECTOR(10*RANDOM(1)-10,i,3)
```

- Dann definieren wir die Funktion f(x): y = k/x und berechnen die y-Werte der Eckpunkte des Dreiecks.

```
fhyp(x):=k/x
aeck:=[xval SUB 1,fhyp(xval SUB 1)]
```

```
beck:=[xval SUB 2,fhyp(xval SUB 2)]
ceck:=[xval SUB 3,fhyp(xval SUB 3)]
```

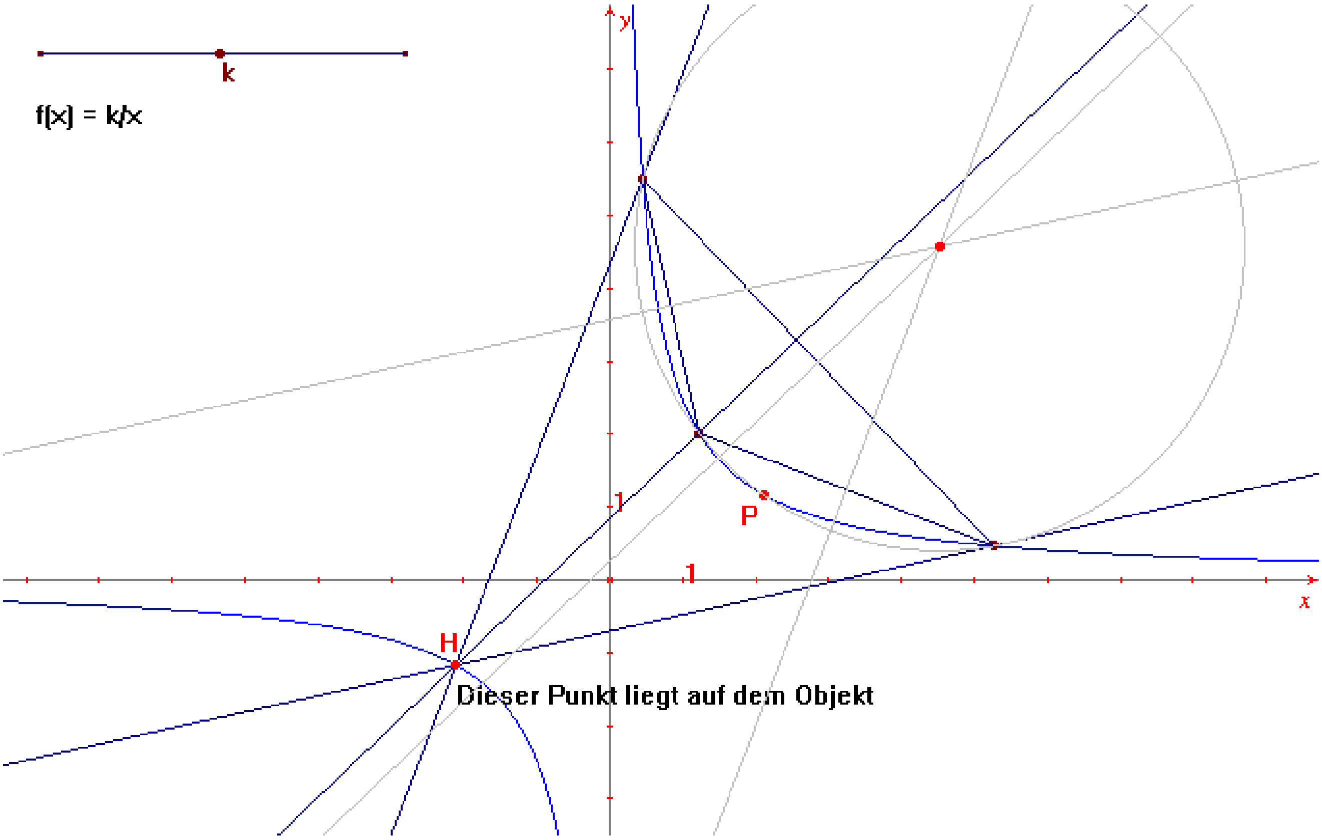
- Wir berechnen weiter über das Skalarprodukt (Vektoren aufeinander normal ... Skalarprodukt gleich 0!) den Höhenschnittpunkt des Dreiecks und überprüfen, ob dieser tatsächlich die Funktion erfüllt (durch Approximation minimale Ungenauigkeiten möglich).

```
hoehe:=SOLUTIONS(([x,y]-aeck).(beck-ceck)=0 AND ([x,y]-beck).(aeck-ceck)=0,[x,y])
fhyp(hoehe SUB 1 SUB 1)=hoehe SUB 1 SUB 2
```

Annahme B:

Der am Ursprung gespiegelte Höhenschnittpunkt liegt am Umkreis des Dreiecks.

B.1. Geometrische Realisierung mit Cabri Geometré:



B.2. Algebraische Überprüfung mit Derive:

Vorgang für spezielle Werte

- Wir berechnen die Koordinaten des Umkreismittelpunktes und überprüfen wieder, ob er die Funktionsgleichung erfüllt.

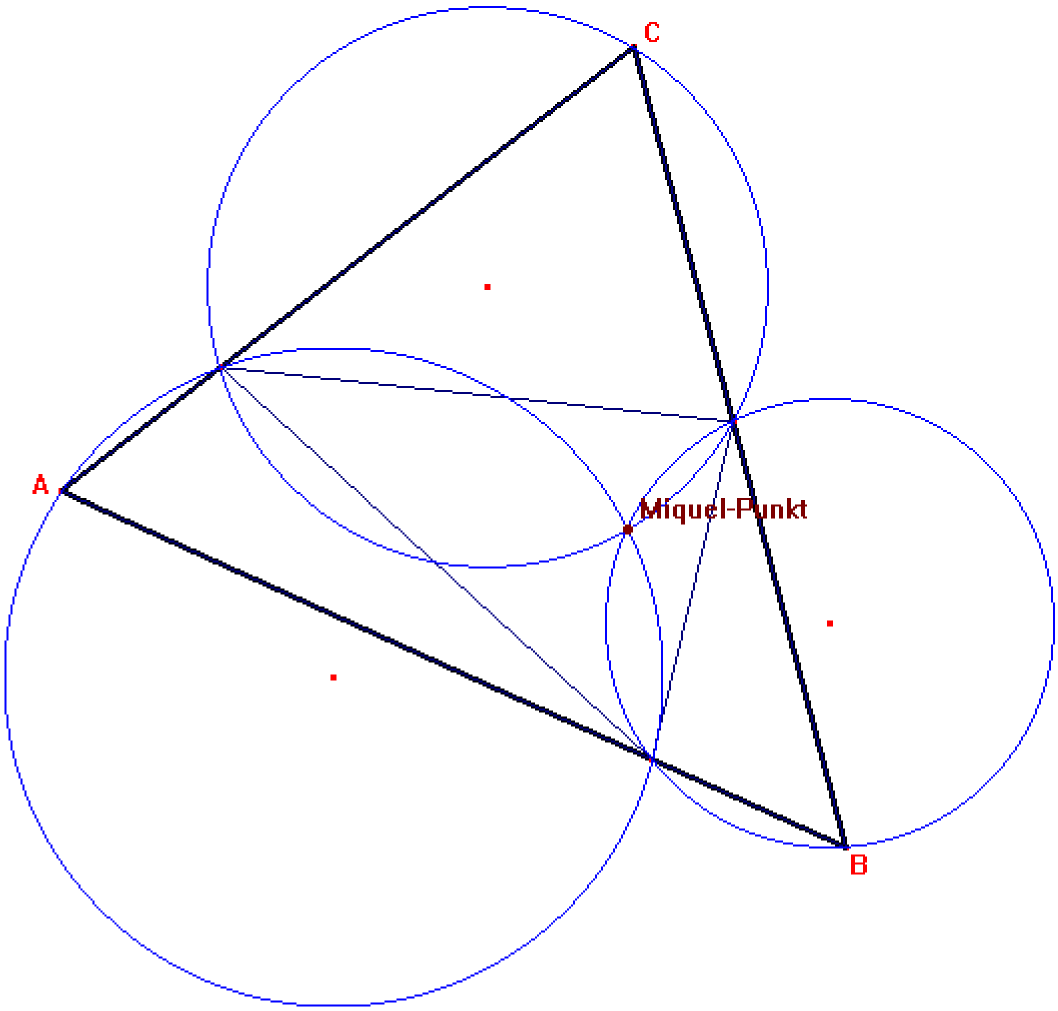
```
umkreis:=(SOLUTIONS(([x,y]-(aeck+beck)/2) . (aeck-beck)=0
      AND ([x,y]-(aeck+ceck)/2) . (aeck-ceck)=0,[x,y])) SUB 1
umk_rad:=ABS(aeck-umkreis)

(-hoehe SUB 1 SUB 1-umkreis SUB 1)^2+(-hoehe SUB 1 SUB 2-umkreis SUB 2)^2
      =umk_rad^2
```

2. Dreieck – Miquel's Theorem

Nimmt man auf jeder Seite eines gegebenen Dreiecks einen Punkt beliebig an und zeichnet durch jede Ecke und die beiden Punkte, die auf den der Ecke benachbarten Seiten liegen, Kreise, so gehen die drei Kreise durch einen Punkt, der Miquelscher Punkt genannt wird.

1. Geometrische Realisierung mit Cabri Geometré:



2. Algebraische Überprüfung mit Derive:

Vorgang für spezielle Werte

- Wir legen über den Zufallsgenerator die Koordinaten der Eckpunkte des Dreiecks (ae, be, ce) und die Punkte auf den Seiten (p1, p2, p3) fest.

```
RANDOM(0)
kvals:=VECTOR(RANDOM(1),i,3)
xvals:=VECTOR(20*RANDOM(1)-10,i,3)
yvals:=VECTOR(20*RANDOM(1)-10,i,3)

ae:=[xvals SUB 1,yvals SUB 1]
be:=[xvals SUB 2,yvals SUB 2]
ce:=[xvals SUB 3,yvals SUB 3]

p1:=ae+kvals SUB 1*(be-ae)
p2:=be+kvals SUB 2*(ce-be)
p3:=ce+kvals SUB 3*(ae-ce)
```

- Anschließend werden die Kreise aufgestellt und geschnitten. Wir überprüfen, ob beim Schnitt der drei Kreise (paarweise Schnittmenge bestimmen) tatsächlich ein eindeutiger Punkt – der Miquel-Punkt – herauskommt.

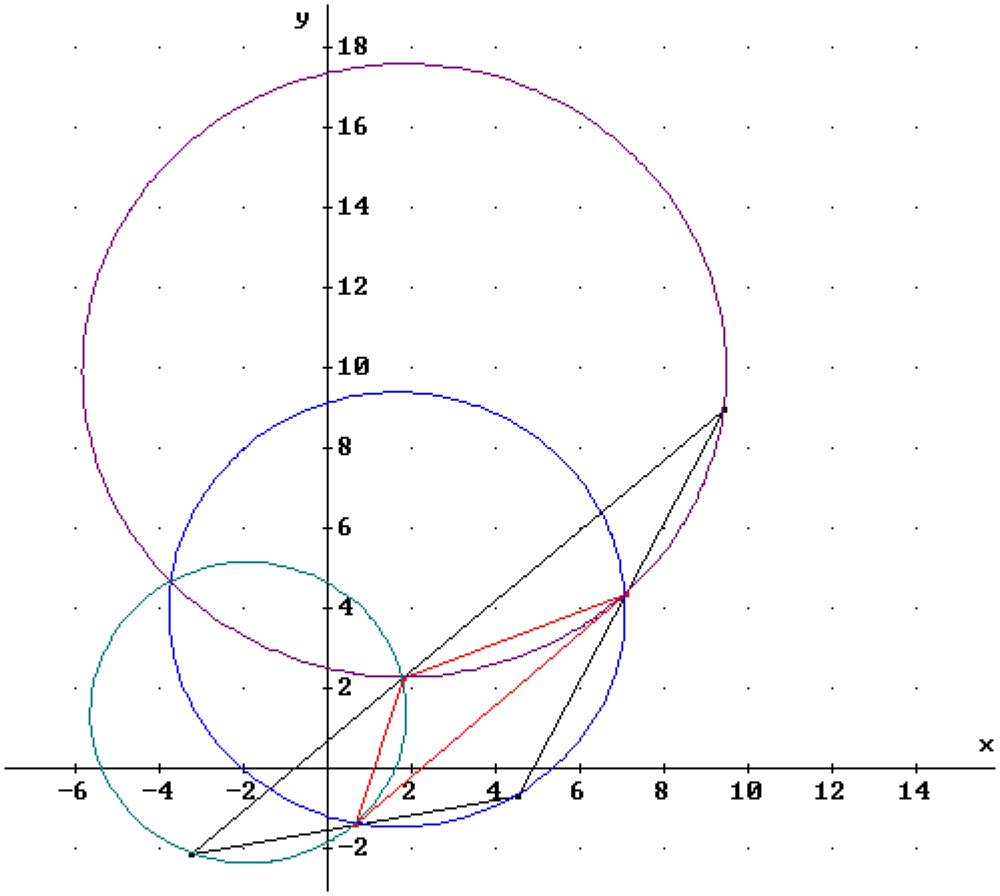
```
u1m:=(SOLUTIONS(([x,y]-(ae+p1)/2) . (ae-p1)=0
      AND ([x,y]-(ae+p3)/2) . (ae-p3)=0,[x,y])) SUB 1
u2m:=(SOLUTIONS(([x,y]-(be+p1)/2) . (be-p1)=0
      AND ([x,y]-(be+p2)/2) . (be-p2)=0,[x,y])) SUB 1
u3m:=(SOLUTIONS(([x,y]-(ce+p2)/2) . (ce-p2)=0
      AND ([x,y]-(ce+p3)/2) . (ce-p3)=0,[x,y])) SUB 1

u1r:=ABS(u1m-ae)
u2r:=ABS(u2m-be)
u3r:=ABS(u3m-ce)

SOLUTIONS(([x,y]-u1m)^2=u1r^2 AND ([x,y]-u2m)^2=u2r^2,[x,y])
SOLUTIONS(([x,y]-u2m)^2=u2r^2 AND ([x,y]-u3m)^2=u3r^2,[x,y])
SOLUTIONS(([x,y]-u1m)^2=u1r^2 AND ([x,y]-u3m)^2=u3r^2,[x,y])
```

- Eine geometrische Überprüfung ist auch in Derive möglich – über den 2D-Plot können die Sachverhalte auch graphisch ausgegeben werden (Achtung: <Display-Options> <Points> sollte auf „Connect" gestellt sein).

Beispiel für Miquel-Punkt:



Die rechnerische Überprüfung ergibt als Schnittpunkt der drei Kreise den Punkt mit den gerundeten Koordinaten. In unserem Fall für die in der Graphik verwendeten Punkte ergibt die Rechnung: Miquel = (- 3,71 / 4,64).

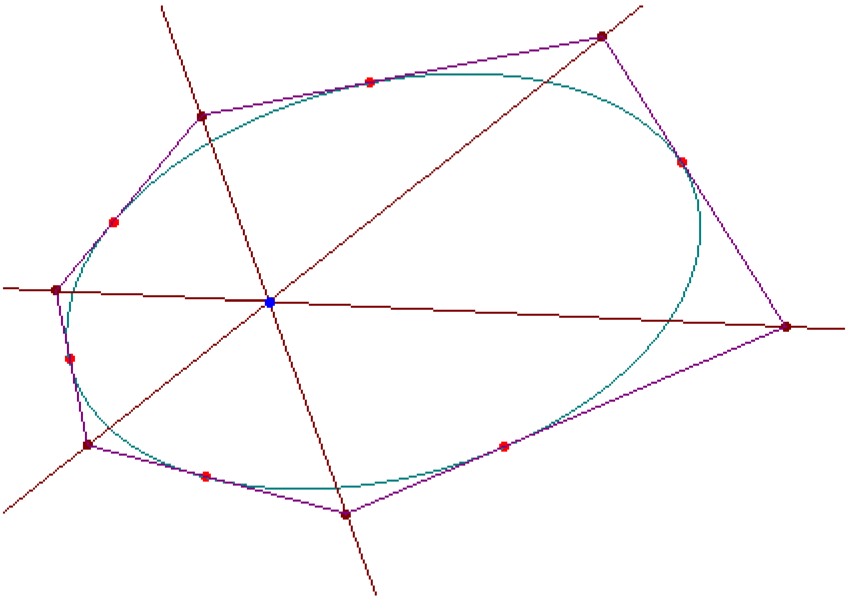
3. SATZ VON BRIANCHON

Wenn man einem beliebigen Kegelschnitt ein geschlossenes Sechsseit umschreibt und die entstehenden Ecken von 1 bis 6 durchnumeriert, dann gehen die Verbindungsgeraden von gegenüberliegenden Ecken ([1,4], [2,5] und [3,6]) immer durch einen gemeinsamen Punkt, den *BRIANCHONschen Punkt*.

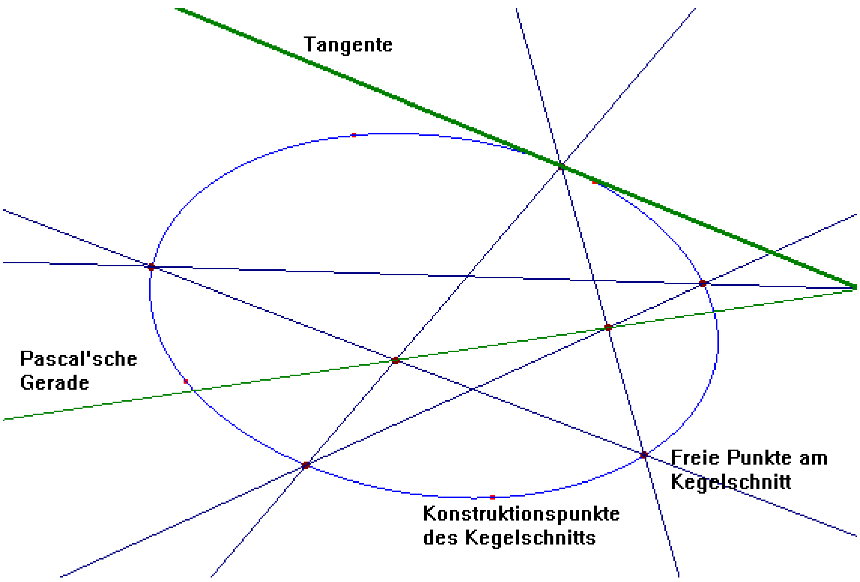
1. Geometrische Realisierung mit Cabri Geometré:

Die Realisierung mit Cabri kann auf verschiedene Arten gelöst werden. Wenn man die eingebaute Funktion zur Erzeugung eines Kegelschnitts benützt, hat man das Problem, dass gerade der Satz von Pascal und sein duales Gegenstück, der Satz von Brianchon intern benützt werden, um den Kegelschnitt zu erzeugen!

Ein weiteres Problem ergibt sich bei der Erzeugung der Tangenten. Hier stellt Cabri keine direkte Funktion zur Verfügung. (ein gutes Anwendungsbeispiel für ein Makro!). Der Nachteil ist wiederum, dass die Erzeugung der Tangenten mit Hilfe des Satzes von Pascal erfolgt.



Tangentenmakro:



2. Algebraische Überprüfung mit Derive:
(nach Josef Böhm)

- Löschen der Variablen

```
[ellipse:=,pts:=,diags:=,brpt:=,ips:=,tgs:=]
```

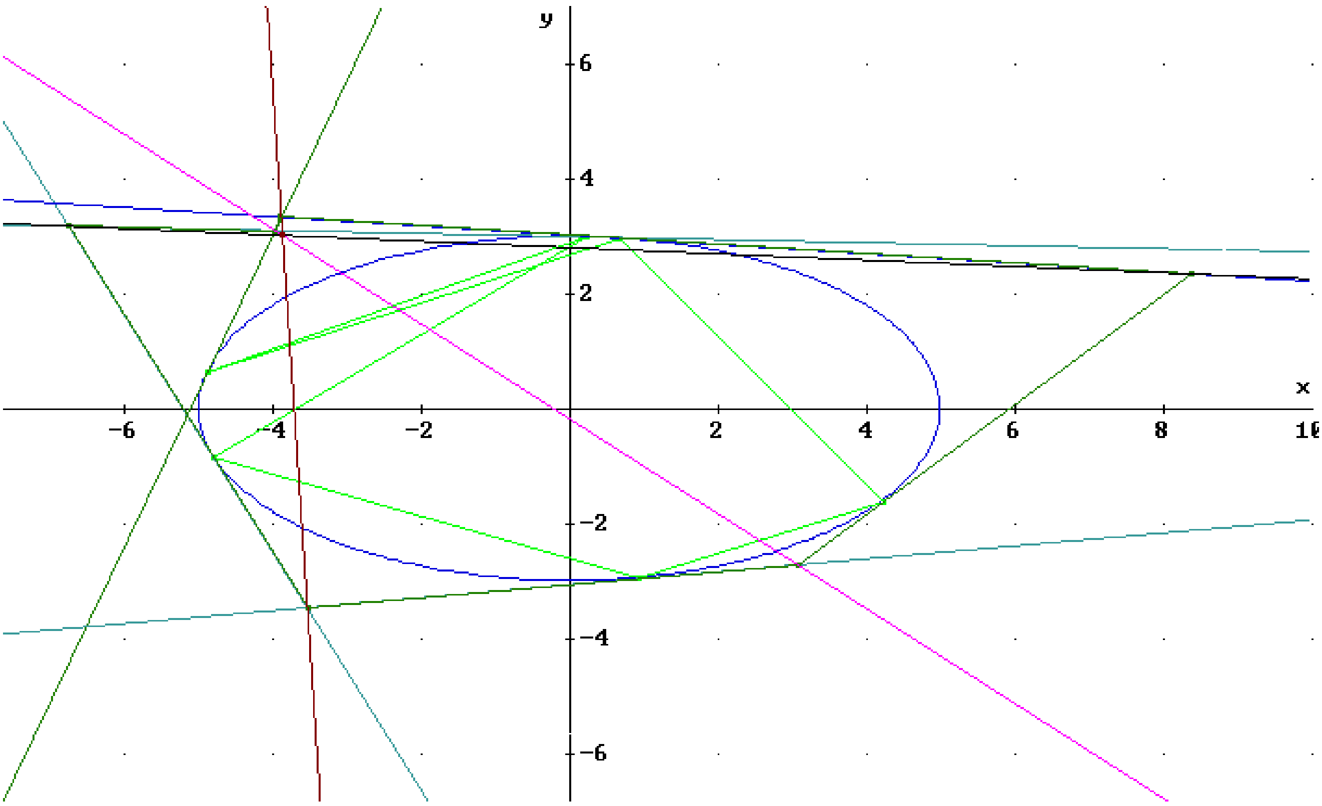
- Aufstellen eines Programms, das den Brianchon'schen Punkt für eine Ellipse mit 6 zufällig gewählten Punkten berechnet.
- a ... Hauptachsenparameter, b ... Nebenachsenparameter

```
brianchon2(a,b,dummy,vals):=
PROG
   dummy:=RANDOM(0),
   vals:=VECTOR(2*pi*RANDOM(1),k,6)
   vals:=APPEND(vals,[vals SUB 1])
   pts:=VECTOR([a*COS(t),b*SIN(t)],t,vals)
   tgs:=VECTOR([SUBST(y=-b*COS(t)/(a*SIN(t))*(x-a*COS(t))+b*SIN(t),t,k)],k,vals)
   ips:=VECTOR((SOLUTIONS(tgs SUB i SUB 1
       AND tgs SUB (i+1) SUB 1,[x,y])) SUB 1,i,6)
   ips:=APPEND(ips,[ips SUB 1])
   diags:=VECTOR(EXPAND([y=(ips SUB k SUB 2-ips SUB (k+3) SUB 2)/
       (ips SUB k SUB 1-ips SUB (k+3) SUB 1)*
       (x-ips SUB k SUB 1)+ips SUB k SUB 2]),k,1,3)
   brpt:=(SOLUTIONS(diags SUB 1 SUB 1 AND
       diags SUB 2 SUB 1,[x,y])) SUB 1
   ellipse:=b^2*x^2+a^2*y^2=a^2*b^2
   RETURN("Zeichne ellipse, pts, tgs,ips,diags,brpt,marks")
```

- Aufruf des Programms mit den gewünschten Achsenparametern

```
brianchon2(4,2)
```

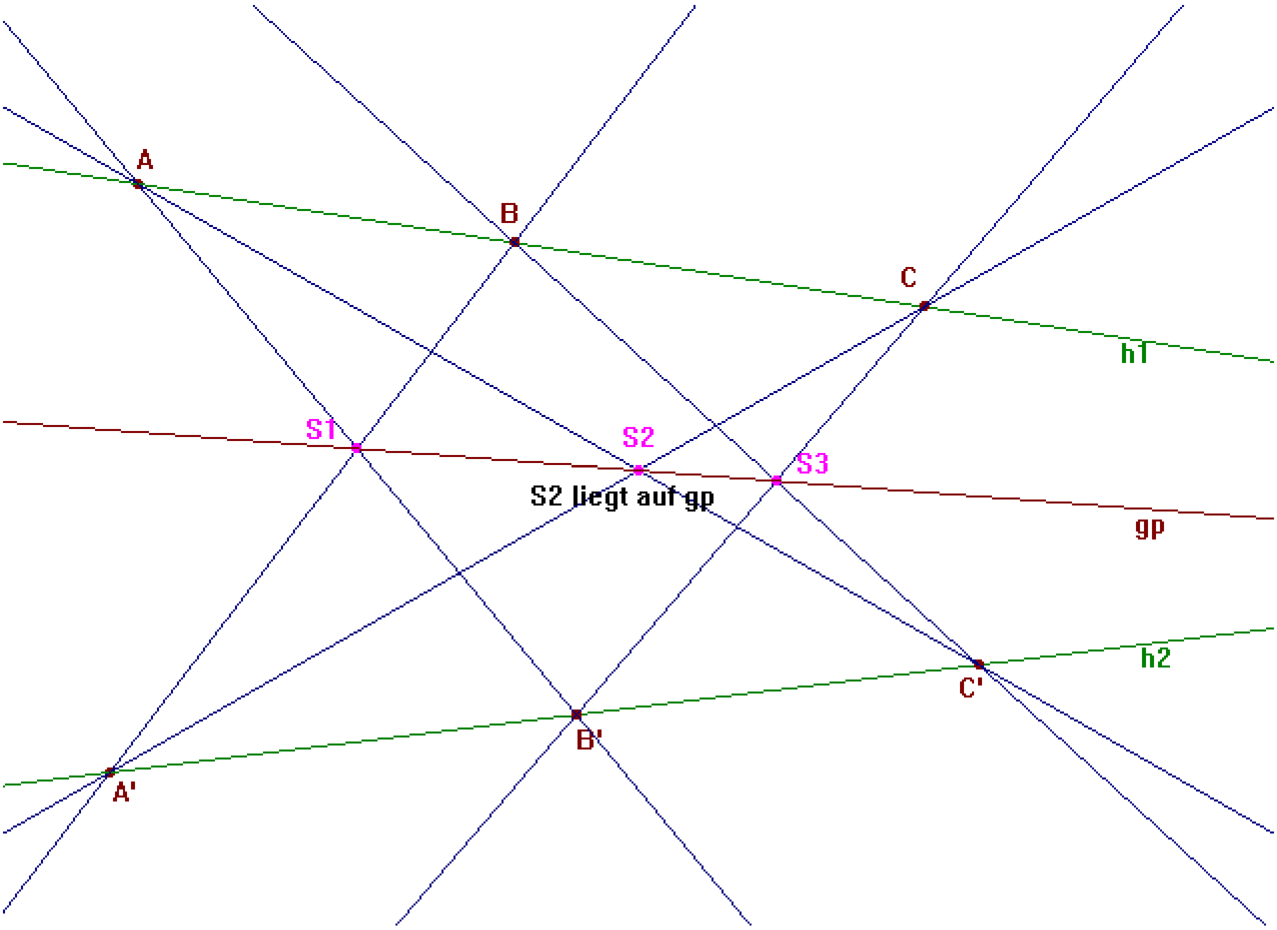
- Über die Variablen **ellipse**, **pts**, **tgs**, **ips**, **diags** und schließlich **brpt** (Brianchon-Punkt) kann mittels graphischer oder numerischer Ausgabe die Richtigkeit des Satzes überprüft werden.



4. SATZ VON PAPPUS

Seien A, B und C drei Punkte auf einer Geraden h_1 und A', B' und C' drei Punkte auf einer Geraden h_2, ferner S_1 der Schnittpunkt der Geraden g_{AB} durch A und B' mit der Geraden g_{BA} durch B und A', S_2 der Schnittpunkt der Geraden g_{AC} durch A und C' mit g_{CA} durch C und A' und S_3 der Schnittpunkt von g_{BC} durch B und C' mit g_{CB} durch C und B', dann liegen die Punkte S_1, S_2 und S_3 auf einer Geraden g_P.

1. Geometrische Realisierung mit Cabri Geometré:



2. Algebraische Überprüfung mit Derive:

a) Überprüfung mit Hilfe von zufällig gewählten Punkten:

- Starten des Zufallsgenerators und Erzeugung der freien Punkte P1, P2, Q1 und Q2

```
dummy:=RANDOM(0)
vals := VECTOR(10*RANDOM(1)-5,t,10)
p1 := [vals SUB 1,vals SUB 2]
p2 := [vals SUB 3,vals SUB 4]
q1 := [vals SUB 5,vals SUB 6]
q2 := [vals SUB 7,vals SUB 8]
```

- Erzeugen einer Funktion zur Erzeugung einer Geraden aus zwei Punkten

```
gerade(a,b):=(b SUB 2-a SUB 2)/(b SUB 1-a SUB 1)*x+
      a SUB 2-a SUB 1*(b SUB 2-a SUB 2)/(b SUB 1-a SUB 1)
```

- Aufstellen der beiden Geraden g und h aus den erzeugten Punkten

```
g(x):=gerade(p1,p2)
h(x):=gerade(q1,q2)
```

- Wir erstellen die restlichen beiden Punkte P3 und Q3

```
p3:=[vals SUB 9,g(vals SUB 9)]
q3:=[vals SUB 10,h(vals SUB 10)]
```

- Aufstellen einer Funktion zum Schnitt von zwei Geraden. Berechnen der x-Koordinaten der drei Schnittpunkte.

```
schnitt(g1,h1):=(SOLUTIONS(g1=h1,x)) SUB 1
rx1:=schnitt(gerade(p1,q2),gerade(p2,q1))
rx2:=schnitt(gerade(p1,q3),gerade(p3,q1))
rx3:=schnitt(gerade(p2,q3),gerade(p3,q2))
```

- Für die Berechnung der y-Koordinaten werden die Verbindungsgeraden aufgestellt und anschließend die drei Schnittpunkte fertig ausgerechnet.

```
v1(x):=gerade(p1,q2)
v2(x):=gerade(p2,q1)
v3(x):=gerade(p1,q3)
v4(x):=gerade(p3,q1)
v5(x):=gerade(p2,q3)
v6(x):=gerade(p3,q2)

r1:=[rx1,v1(rx1)]
r2:=[rx2,v3(rx2)]
r3:=[rx3,v5(rx3)]
```

- Erstellen der Geraden durch zwei der Schnittpunkte und Überprüfung, ob der dritte Schnittpunkt ebenfalls auf dieser Geraden liegt (Ergebnis 0!).

```
i(x):=gerade(r1,r2)
i(r3 SUB 1)-r3 SUB 2
```

b) Überprüfung mit Hilfe von beliebigen Punkten (allgemeiner Nachweis):

```
p1:=[x1,y1]
p2:=[x2,y2]
q1:=[x4,y4]
q2:=[x5,y5]

gerade(a,b):=(b SUB 2-a SUB 2)/(b SUB 1-a SUB 1)*x+
      a SUB 2-a SUB 1*(b SUB 2-a SUB 2)/(b SUB 1-a SUB 1)

g(x):=gerade(p1,p2)
h(x):=gerade(q1,q2)

p3:=[x3,g(x3)]
q3:=[x6,h(x6)]

schnitt(g1,h1):=(SOLUTIONS(g1=h1,x)) SUB 1

rx1:=schnitt(gerade(p1,q2),gerade(p2,q1))
rx2:=schnitt(gerade(p1,q3),gerade(p3,q1))
rx3:=schnitt(gerade(p2,q3),gerade(p3,q2))

v1(x):=gerade(p1,q2)
v3(x):=gerade(p1,q3)
v5(x):=gerade(p2,q3)

r1:=[rx1,v1(rx1)]
r2:=[rx2,v3(rx2)]
r3:=[rx3,v5(rx3)]

i(x):=gerade(r1,r2)
i(r3 SUB 1)-r3 SUB 2
```

- Auch hier erhalten wir als Ergebnis 0, der dritte Schnittpunkt liegt auf der Geraden, die mit Hilfe der beiden anderen berechneten Schnittpunkte erzeugt wurde.