

Statik: Anwendungenn mit dem Modul "initStatik"

Teile dieses Files (Liste):

1. Modul "initStatik" initiieren aus dem File StatikModulinit.nb, Erklärungen zu den einzelnen Funktionen abrufen.
Alternative: Modul "initStatikOhneE" initiieren aus dem File StatikModulinitOhneErkl.nb, ohne die umfangreichen Erklärungen. Die Funktionen sind dieselben wie in "initStatik".
2. Einzelne Funktionen testen
3. Ein Statik-Beispiel
4. Putzmaschine ==> Alles löschen ==> Modul muss nochmals initiiert werden
(alles gelöscht)

Teile dieses Files (Work):

1. Modul "initStatik" initiieren, Erklärungen zu den einzelnen Funktionen abrufen

Initierter Modul aufrufen, damit der Erklärungstext kommt: Cursor in das Feld initStatik[1] platzieren oder rechts auf den zugehörigen Balken klicken, anschliessend Enter-Taste drücken (Enter, nicht Wagenrücklauf!)

```
initstatik[1]
```

Kurze Erklärungen:

1. Berchnung der resultierenden Kraft zu zwei oder mehreren Kräften:

```
kResultOut[6,7,3,4,5] gibt die Resultierende der aufgezählten skalaren Kräfte 6,7,3,4,5

kResultOut[{{1,2},{1,3}}] gibt die
Resultierende der aufgezählten vektoriellen Kräfte {1,2},{1,3}

kResultOut[{{1,2},{1,3},{1,4},{1,5}}] gibt die
Resultierende der aufgezählten vektoriellen Kräfte {1,2},{1,3},{1,4},{1,5}

Apply[kResult,{1,2},{1,3},{1,4},{1,5}]] gibt die
Resultierende der aufgezählten vektoriellen Kräfte {1,2},{1,3},{1,4},{1,5}

{{1,2},{1,3},{1,4},{1,5}}//kResultOut gibt die
Resultierende der aufgezählten vektoriellen Kräfte {1,2},{1,3},{1,4},{1,5}
```

2a. Schnittpunktsberechnung zu zwei Wirkungslinien

```
p1={1,2};p2={1,3};q1={4,1};q2={4,4}; geradeSchnittOut[p1,p2,
q1,q2] gibt den Schnittpunkt zu den Geraden p1,p2 und q1,q2 (hier leer)

p1={1,2};p2={2,3};q1={4,1};q2={4,4}; geradeSchnittOut[
p1,p2,q1,q2] gibt den Schnittpunkt zu den Geraden p1,p2 und q1,q2
```

2b. Komponentenzerlegung von Kräften und Plots

```
p1={1,2};p2={-1,3};q1={3,1};q2={6,4};kraft={4,-3}; kraftPunktZerleg[
p1,p2,q1,q2,kraft]; s1=graphZerleg[p1,p2,q1,q2,kraft]; gibt die
Komponentenzerlegung von kraft in Richtung p1,p2 und q1,q2 sowie einen Plot dazu

p1={5,2};p2={2,3};q1={4,1};q2={4,4};kraft={3,5}; kraftPunktZerleg[
p1,p2,q1,q2,kraft]; s2=graphZerleg[p1,p2,q1,q2,kraft]; gibt die
Komponentenzerlegung von kraft in Richtung p1,p2 und q1,q2 sowie einen Plot dazu

Show[s2,s1]; legt die obigen beiden Plots zusammen
```

3. Komponentenzerlegung einer Kraft in eine Richtungskraft und die Normalkraft mit Plot

```
p1={1,2};q1={3,1};kraft={4,-3}; kraftPunktZerlegSenkr[p1,q2,kraft];
graphZerlegSenkr[p1,q2,kraft]; gibt die Komponentenzerlegung von kraft
in Richtung p1,p2 und in der dazu senkrechten Richtung sowie einen Plot dazu
```

4. Plots von Punkten, Linienzügen, Kräften

```
p1={1,2};p2={7,2};q1={3,1}; s3=showPointsLines[
{p1,p2,q1,{5,6}}, {p1,p2,q1,{5,6},p1,q1}]; Show[s1,s2,s3];

p1={1,2};p2={7,2};q1={3,1}; s3=showPointsLines[{p1,p2,q1,{5,6}}, {p1,p2,q1,{5,6},p1,q1}];
gibt einen Plot der Punkte p1,p2,q1,{5,6} und des Linienzuges p1,p2,q1,{5,6},p1,q1

Show[s1,s2,s3]; legt die Plots s1,s2 und s3 übereinander
```

```

p0={0,0}; p1={1,2}; p2={-1,3}; p3={4,1}; p4=
{2,3}; showArrow[p1,p2]; zeigt den Plot mit dem Pfeil (p0,p1)

p0={0,0}; p1={1,2}; p2={-1,3}; p3={4,1}; p4={2,3};
showArrow[{p1,p2}]; zeigt ebenfalls den Plot mit dem Pfeil (p0,p1)

p0={0,0}; p1={1,2}; p2={-1,3}; p3={4,1}; p4={2,3}; showTabArrow[p1,p2,p3,p4]; zeigt
den Plot mit den Pfeilen (p0,p1), (p0,p2), ..., (p0,p4) (Anzahl Pfeile beliebig)

showTabArrow[{p1,p2,p3,p4}]; macht dasselbe wie showTabArrow[p1,p2,p3,p4]

p0={0,0}; p1={1,2}; p2={-1,3}; p3={4,1}; p4={2,3}; showTabP0Arrow[p0,p1,
p2,p3,p4]; zeigt den Punkt p0 sowie die Pfeile (p0,p1), (p0,p2), ..., (p0,p4)

showTabP0Arrow[{p0,p1,p2,p3,p4}]; macht dasselbe wie showTabP0Arrow[p0,p1,p2,p3,p4];

showBigPoints[2,{p1,p2,p3,p4}]; zeigt die um den Faktor 2 vergrößerten Punkte p1, p2, ..., p4

showGroundLine[0.4,p0]; zeigt p0 mit einer horizontalen Linie
der Länge 2 mal 0.4, beidseitig von p0. Die Länge hängt vom Massstab ab.

Show[showGroundLine[0.4,p0],showGroundLine[0.4,
p1]]; zeigt zwei entsprechende Linien wie oben um p0 und um p1

Show[showGroundLine[p0],showGroundLine[0.6,p1]]; zeigt eine Linie um
p0 mit der Default-Länge 2 mal 0.2 und eine um p1 mit der Länge 2 mal 0.6

```

5. Momente und Normalkraft aus den Momenten

```

drehpunkt={-5,4}; p1={6,7}; p2={9,8}; p3={1,2}; q={4,6}; pkraft1={2,2}; pkraft2=
{4,2}; pkraft3={2,2}; mengePunkte={p1,p2,p3}; mengeKraft={pkraft1,pkraft2,
pkraft3}; moment[drehpunkt,mengePunkte,mengeKraft] ergibt das totale Moment

drehpunkt={-5,4}; p1={6,7}; p2={9,8}; p3={1,2}; q={4,6}; pkraft1={2,2}; pkraft2={4,2};
pkraft3={2,2}; mengePunkte={p1,p2,p3}; mengeKraft={pkraft1,pkraft2,pkraft3};
resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,q] ergibt die in
q wirkende Normalkraft für das Momentengleichgewicht bezüglich dem Drehpunkt

drehpunkt={-5,4}; p1={6,7}; p2={9,8}; p3={1,2}; q={4,6}; pkraft1={2,2};
pkraft2={4,2}; pkraft3={2,2}; mengePunkte={p1,p2,p3}; mengeKraft={pkraft1,
pkraft2,pkraft3}; resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,
q]/N ergibt die in q wirkende Normalkraft für das Momentengleichgewicht
bezüglich dem Drehpunkt numerisch, falls die Eingaben exakt sind

drehpunkt={-5,4}; p1={6,7}; p2={9,8}; p3={1,2}; q={4,6}; pkraft1={2,2};
pkraft2={4,2}; pkraft3={2,2}; mengePunkte={p1,p2,p3}; mengeKraft={pkraft1,
pkraft2,pkraft3}; N[resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,
q]] ergibt die in q wirkende Normalkraft für das Momentengleichgewicht
bezüglich dem Drehpunkt numerisch, falls die Eingaben exakt sind

''Moment = '', moment[drehpunkt,mengePunkte,mengeKraft], '', Kraft = '',
resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,q] // N] mit richtig
gesetzten Anführungsstrichen (siehe Anwendung) ergibt eine Ausgabe mit Moment und Kraft

```

6. Erklärungen: Als Output auf dem Schirm lesbar, wenn der Modul initiiert worden ist

2. Einzelne Funktionen testen

**Anwendungen zur der Funktion "kResultOut" (In) für die resultierenden:
Ebenfalls auf den jeweiligen Balken klicken und Enter drücken, Achtung:
Berechnungen greifen teilweise auf vorher berechnete Resultate ==> Reihenfolge
beachten!**

```
kResultOut[6,7,3,4,5]
25

kResultOut[{{1,2},{1,3}}]
{2, 5}

kResultOut[{{1,2},{1,3},{1,4},{1,5}}]
{4, 14}

Apply[kResult,{{1,2},{1,3},{1,4},{1,5}}]
kResult[{1, 2}, {1, 3}, {1, 4}, {1, 5}]
{{1,2},{1,3},{1,4},{1,5}}//kResultOut
{4, 14}

kResultPrint[{{1,2},{1,3},{1,4},{1,5}}]
Resultierende Kraft kResultOut = {4, 14}
```

**Anwendungen zur der Funktion "kraftPunktZerleg" (In) für die Komponenten
"kKomp1Out, kKomp2Out" und "geradeSchnittOut" sowie eine Skizze (Out)
"graphZerleg" (Reihenfolge muss übernommen werden!)**

```
(* Keine Angst vor den Fehlermeldungen! Hier gibt es keinen Schnittpunkt, da die
Gerade parallel laufen!!! *)

p1={1,2};p2={1,3};q1={4,1};q2={4,4};
geradeSchnittOut[p1,p2,q1,q2]

Kein Schnitt

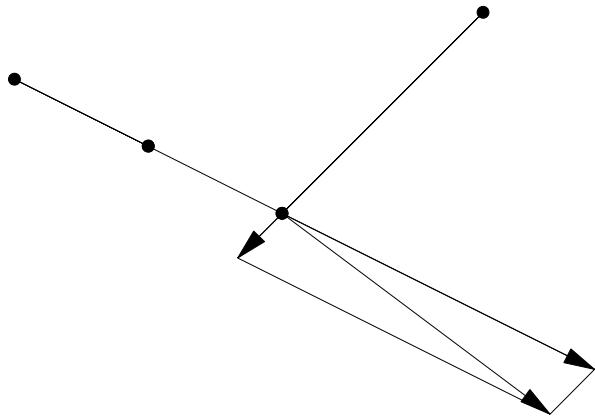
p1={1,2};p2={2,3};q1={4,1};q2={4,4};
geradeSchnittOut[p1,p2,q1,q2]

{4, 5}

p1={1,2};p2={-1,3};q1={3,1};q2={6,4};kraft={4,-3};
kraftPunktZerleg[p1,p2,q1,q2,kraft];
graphZerleg[p1,p2,q1,q2,kraft];

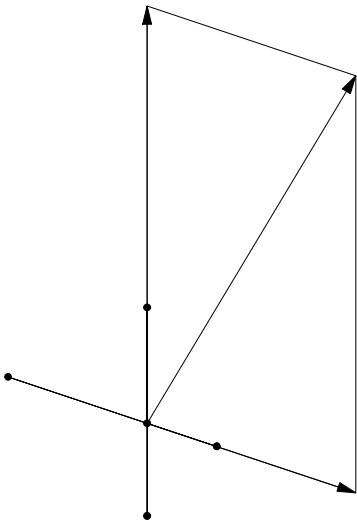
kKomplout = { $\frac{14}{3}$ , - $\frac{7}{3}$ } kKomp2out = {- $\frac{2}{3}$ , - $\frac{2}{3}$ }
```

```
s1=Show[graphZerleg[p1,p2,q1,q2,kraft], display];
```

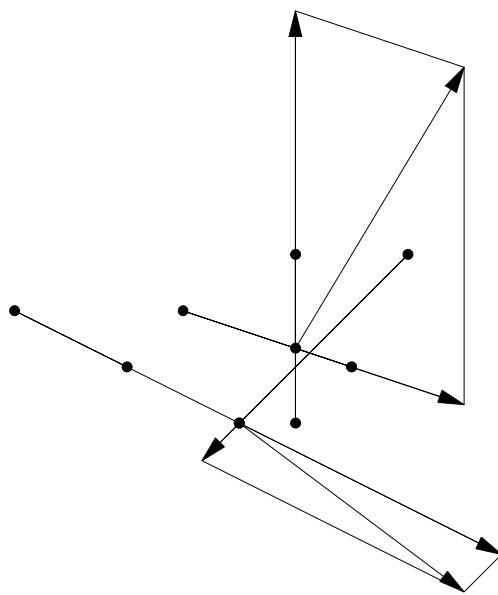


```
p1={5,2};p2={2,3};q1={4,1};q2={4,4};kraft={3,5};  
kraftPunktZerleg[p1,p2,q1,q2,kraft];  
s2=Show[graphZerleg[p1,p2,q1,q2,kraft], display];
```

```
kKomplout = {3, -1} kKomp2out = {0, 6}
```



```
Show[s2,s1];
```

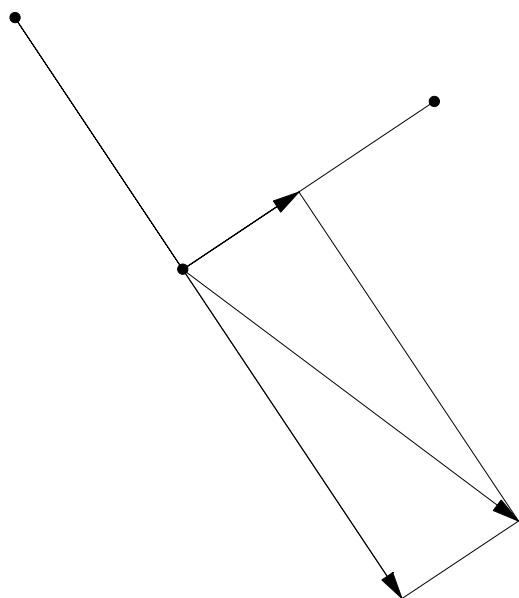


Anwendungen zur der Funktion "kraftPunktZerlegSenkr" (In) für die Komponenten "kKomp1Out, kKomp2Out" im Punkt qG1 sowie eine Skizze (Out) "graphZerleg" (Reihenfolge muss übernommen werden!)

```
p1={1,2};q1={3,1};kraft={4,-3};
kraftPunktZerlegSenkr[p1,q2,kraft];
graphZerlegSenkr[p1,q2,kraft];

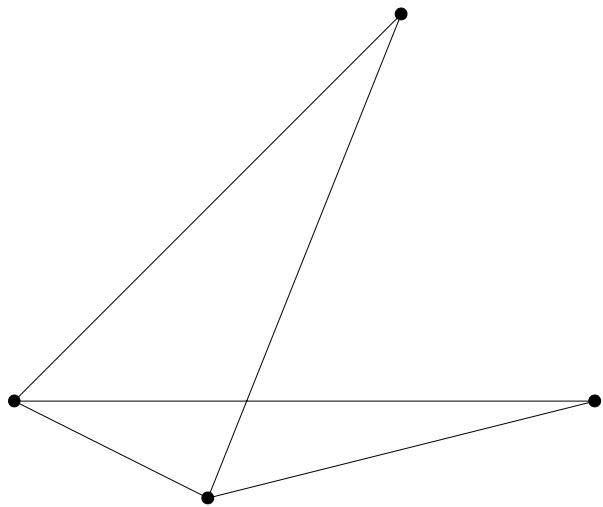
kKomplout = { 18/13 , 12/13 }   kKomp2out = { 34/13 , -51/13 }

Show[graphZerlegSenkr[p1,q2,kraft],display];
```

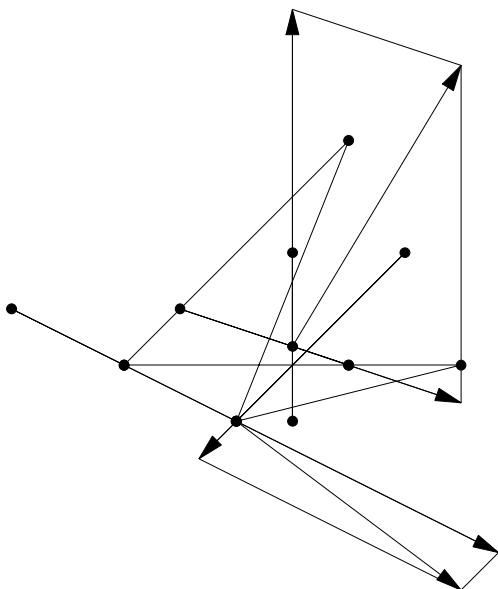


Anwendungen zur der Funktion "showPointsLines" und weiterer Diagrammfunktionen

```
p1={1,2};p2={7,2};q1={3,1};  
s3=Show[showPointsLines[{p1,p2,q1,{5,6}}, {p1,p2,q1,{5,6},p1,q1}],display];
```

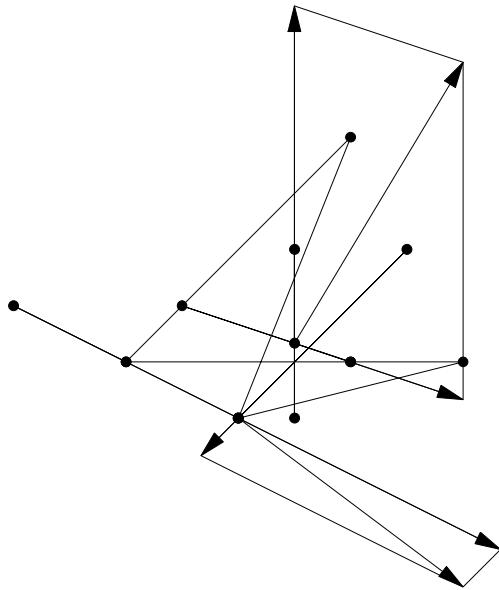


```
p1={1,2};p2={7,2};q1={3,1};  
s3=showPointsLines[{p1,p2,q1,{5,6}}, {p1,p2,q1,{5,6},p1,q1}];  
Show[s1,s2,s3];
```

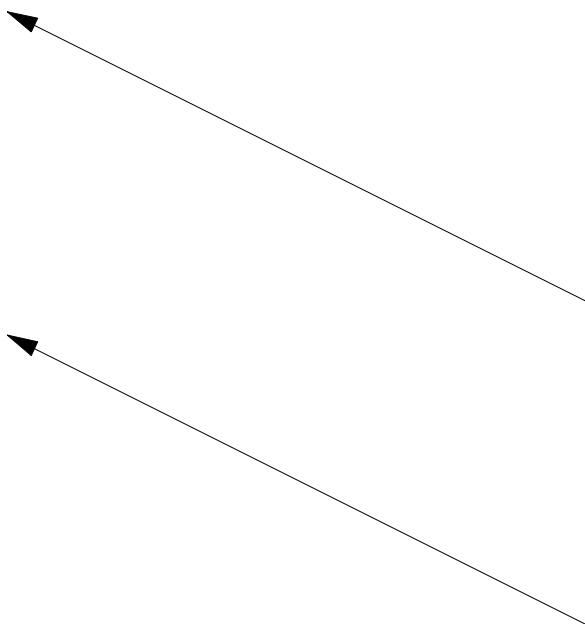


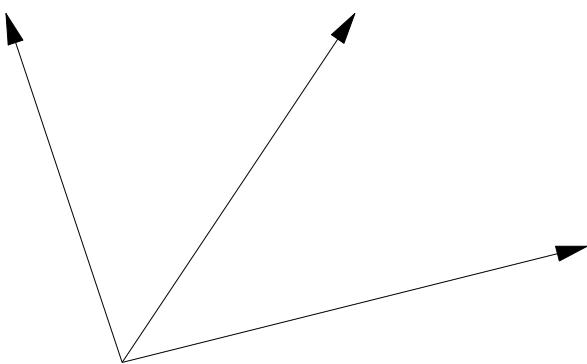
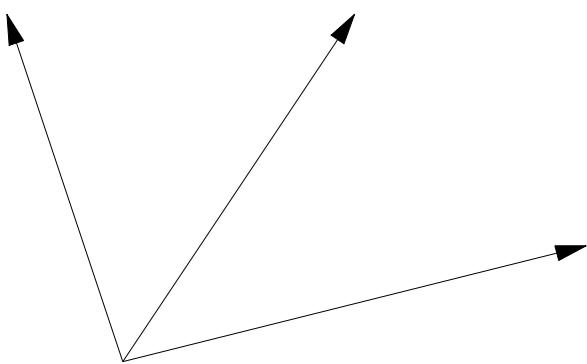
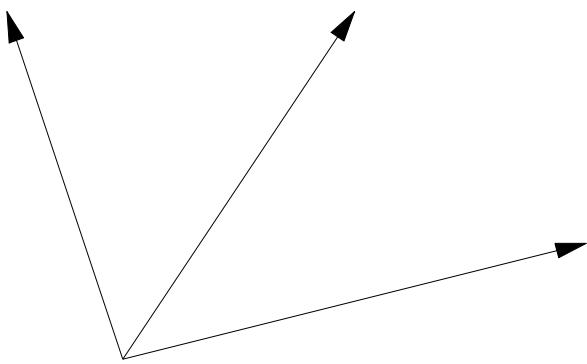
```
Show[s3,s2,s1];
```

```
Show[Show[s3,s2,s1],display];
```

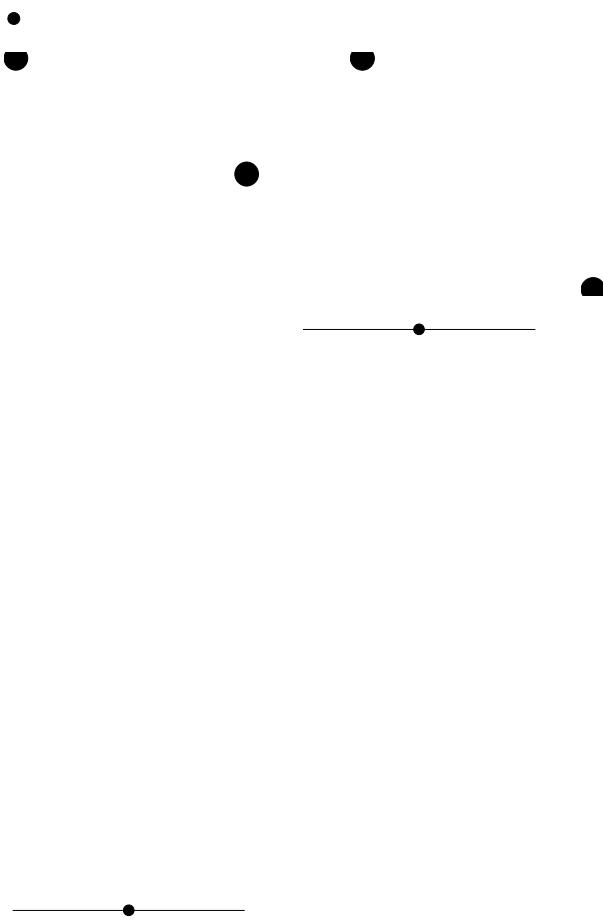
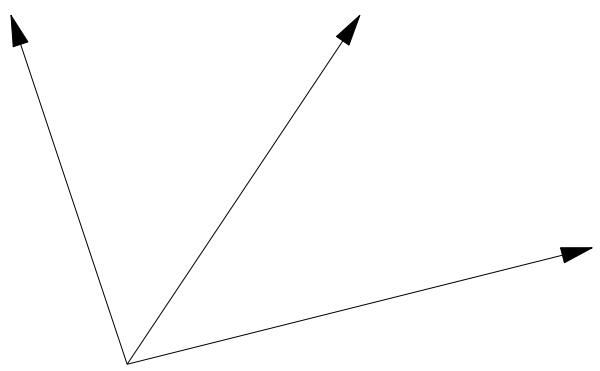


```
p0={0,0}; p1={1,2}; p2={-1,3}; p3={4,1}; p4={2,3};
Show[showArrow[p1,p2],display];
Show[showArrow[{p1,p2}],display];
Show[showTabArrow[p1,p2,p3,p4],display];
Show[showTabArrow[{p1,p2,p3,p4}],display];
Show[showTabP0Arrow[p0,p1,p2,p3,p4],display];
Show[showTabP0Arrow[{p0,p1,p2,p3,p4}],display];
Show[showBigPoints[2,{p1,p2,p3,p4}],display];
Show[Show[showGroundLine[0.4,p0],showGroundLine[0.4,p1]],display];
Show[Show[showGroundLine[p0],showGroundLine[0.6,p1]],display];
```





•





Anwendungen zur der Funktionen "moment" und "resultMomentKraftAbsolut" und weiterer Diagrammfunktionen

```

drehpunkt={-5,4}; p1={6,7}; p2={9,8}; p3={1,2}; q={4,6};
pkraft1={2,2}; pkraft2={4,2}; pkraft3={2,2};
mengePunkte={p1,p2,p3}; mengeKraft={pkraft1,pkraft2,pkraft3};
moment[drehpunkt,mengePunkte,mengeKraft]

44

resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,q]

44
────────
√85

resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,q] // N

4.77247

drehpunkt={0,0}; p1={1,0}; p2={2,0}; q={0,4};
pkraft1={0,2}; pkraft2={0,4};
mengePunkte={p1,p2}; mengeKraft={pkraft1,pkraft2};
Print["Moment = ", moment[drehpunkt,mengePunkte,mengeKraft], ", Kraft =
",resultMomentKraftAbsolut[drehpunkt,mengePunkte,mengeKraft,q] // N]

Moment = 10, Kraft = 2.5

```

3. Ein Statik-Beispiel:

"initStatikOhneE" initialisieren in StatikModullInitOhneErkl.nb

```

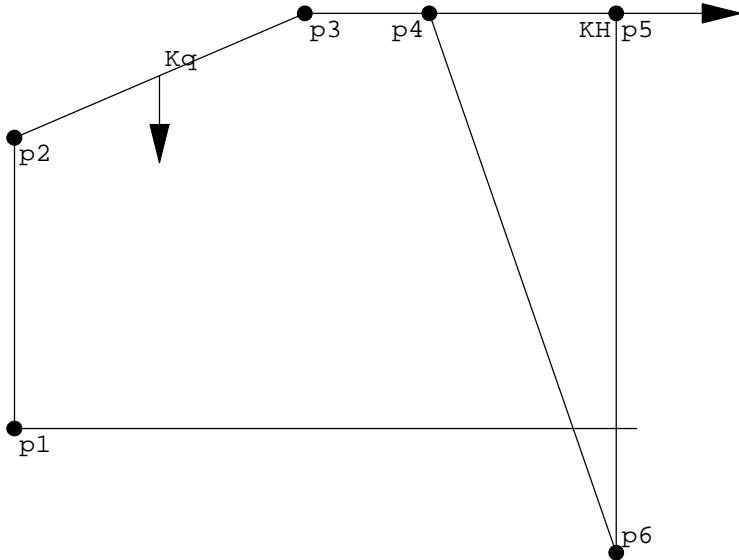
initstatikOhneE[1]

initStatikOhneE[1]

```

Skizze:

p2 und der Stab (p3,p5) sind steif, die anderen Punkte gelenkig. p1 und p6 sind am Boden. Bei p3 kann man daher das Fachwerk in zwei steife Teile trennen. p2 ist steif und rechts hat man ebenfalls ein steifes gebilde, da die Dreiecksform versteift.



Methode: Going down. Erst wird die gesamte äussere Kraft K_{res} ermittelt (eine Kraft ist ein Vektor, hier mit 2 Komponenten). Das ganze Stabgebilde ist steif wegen der Figur (p_4, p_5, p_6) und den festen Lagern am Boden. Daher kann das ganze Gebilde als ein starrer Körper betrachtet werden und die Kräfte auf die Fusspunkte ermittelt werden.

Kraft- und Graphikelemente eingeben

```

p1={0,0};
p2={0,7};
p3={7,10};
p23=(p2+p3)/2;
p4={10,10};
p5={14.5,10};
p6={14.5,-3};
Kq=3*7*{0,-1};
p24=p23+Kq;
KH=30*{1,0};
strecken=1/10 (* Massstabfaktor *);

```

Resultierende Kraft von aussen und Schnittpunkt der Wirkungslinien

(Obwohl die Rechnungen hier trivial sind, benutzen wir zur Demonstration die Module)

```

Kres = kResultOut[KH,Kq] (* resultierende Kraft von aussen *)
{30, -21}

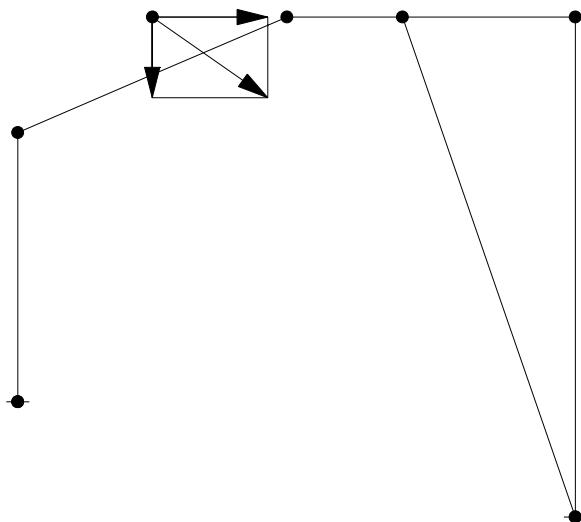
```

```
{p23,p24,p3,p5}
{{7/2, 17/2}, {7/2, -25/2}, {7, 10}, {14.5, 10}}
ps1=geradeschnittOut[p23,p24,p3,p5]
{7/2, 10.}
```

Skizze der momentanen Situation

("display" ist ein im Programm geschriebener Befehl zur Erzeugung der Graphik auf dem Schirm. Dieser Befehl ist notwendig, damit nicht riesige Mengen von zwischendurch erzeugten Graphiken erscheinen.)

```
listP1={p1,p2,ps1,p3,p4,p5,p6};
listL1={p1,p2,p23,ps1,ps1+ Kq strecken, ps1+ Kres strecken, ps1+ KH strecken, ps1, p23, p3, p4, p5, p6, p4};
sk1 = showPointsLines[listP1, listL1];
sk2 = Show[showGroundLine[0.3,p1],showGroundLine[0.3,p6]];
sk3 = showTabArrow[ps1,Kq strecken, KH strecken, Kres strecken];
sk4 = Show[sk1,sk2,sk3,display];
```



Zerlegung der resultierenden Kraft in Komponenten in Richtung der Fusspunkte

```
Kres
{30, -21}

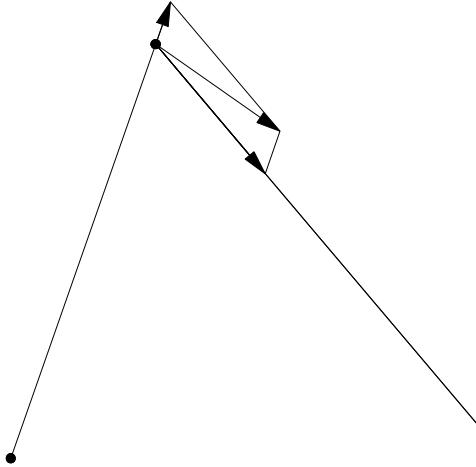
kraftPunktZerleg[p1,ps1,p6,ps1,Kres];
kKomplout = {3.57878, 10.2251} kKomp2out = {26.4212, -31.2251}
```

```
kResP1 = kKomplout;
kResP6 = kKomp2out;
{kResP1,kResP6}

{{3.57878, 10.2251}, {26.4212, -31.2251}}

Kress=Kres strecken;
kraftPunktZerleg[p1,pS1,p6,pS1,Kress];
graphZerleg[p1,pS1,p6,pS1,Kress];
sk5 = Show[graphZerleg[p1,pS1,p6,pS1,Kress], display];

kKomplout = {0.357878, 1.02251} kKomp2out = {2.64212, -3.12251}
```

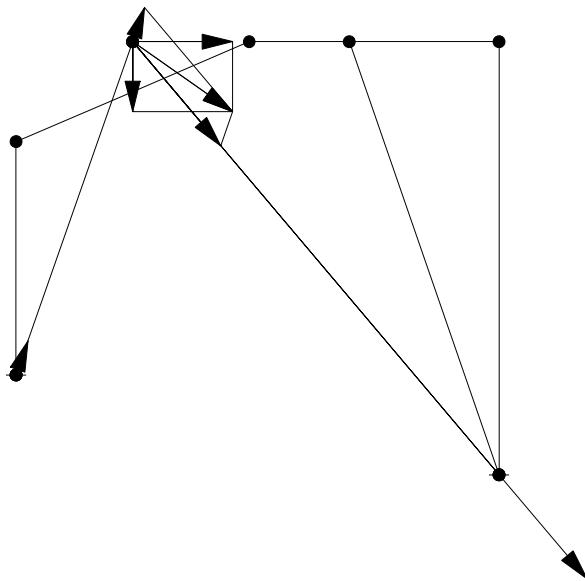


```
kResP1 strecken

{0.357878, 1.02251}

sk6=showArrow[p1, p1 + kResP1 strecken];
sk7=showArrow[p6, p6 + kResP6 strecken];
```

```
sk8 = Show[sk4, sk5, sk6, sk7];
```



```
kraftPunktZerlegSenkr[p1, p2, kResP1];
```

```
kKomplout = {0, 10.2251}  kKomp2out = {3.57878, 0}
```

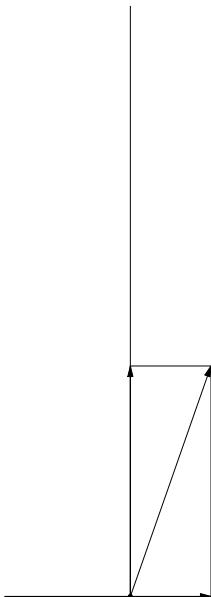
```
kResP12 = kKomplout;
kResP1n = kKomp2out;
{kResP12, kResP1n}
```

```
{ {0, 10.2251}, {3.57878, 0} }
```

```
kraftPunktZerlegSenkr[p1, p2, kResP1 strecken];
```

```
sk9 = Show[graphZerlegSenkr[p1, p2, kResP1 strecken], display];
```

```
kKomplout = {0, 1.02251}  kKomp2out = {0.357878, 0}
```



```
kraftPunktZerlegSenkr[p6, p5, kResP6];
```

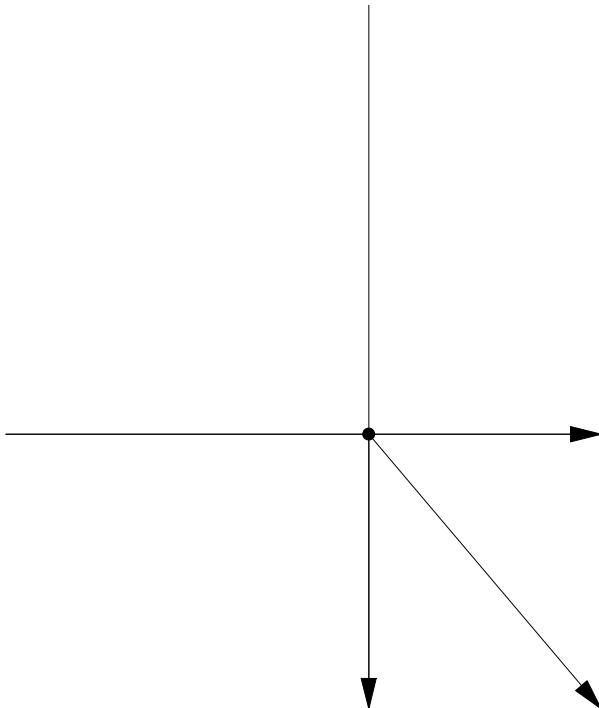
```
kKomplout = {0., -31.2251}  kKomp2out = {26.4212, 0.}
```

```
kResP65n = kKomplout;
kResP6n = kKomp2out;
{kResP65n,kResP6n}

{{0., -31.2251}, {26.4212, 0.}}

kraftPunktZerlegSenkr[p6,p5,kResP6 strecken];
sk10 = Show[graphZerlegSenkr[p6,p5,kResP6 strecken],display];

kKomplout = {0., -3.12251}  kKomp2out = {2.64212, 0.}
```



```
kraftPunktZerleg[p6,p4,p6,p5,kResP6];

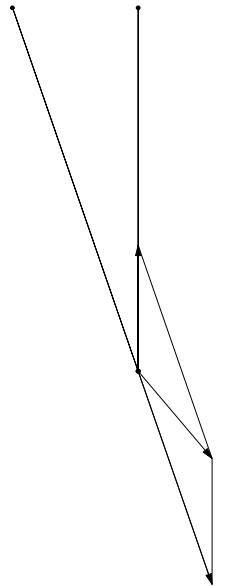
kKomplout = {26.4212, -76.328}  kKomp2out = {0., 45.1029}

kResP64 = kKomplout;
kResP65 = kKomp2out;
{kResP64,kResP65}

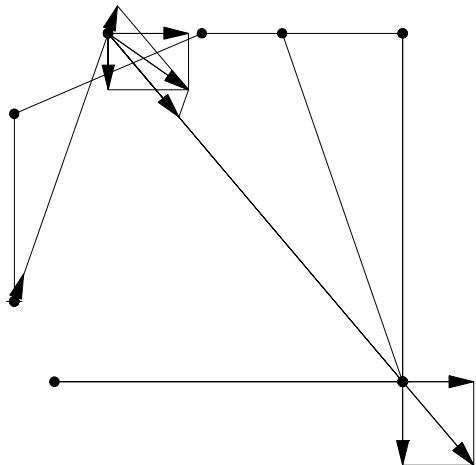
{{26.4212, -76.328}, {0., 45.1029}}
```

```
kraftPunktZerleg[p6,p4,p6,p5,kResP6 strecken];
sk11 = Show[graphZerleg[p6,p4,p6,p5,kResP6 strecken],display];

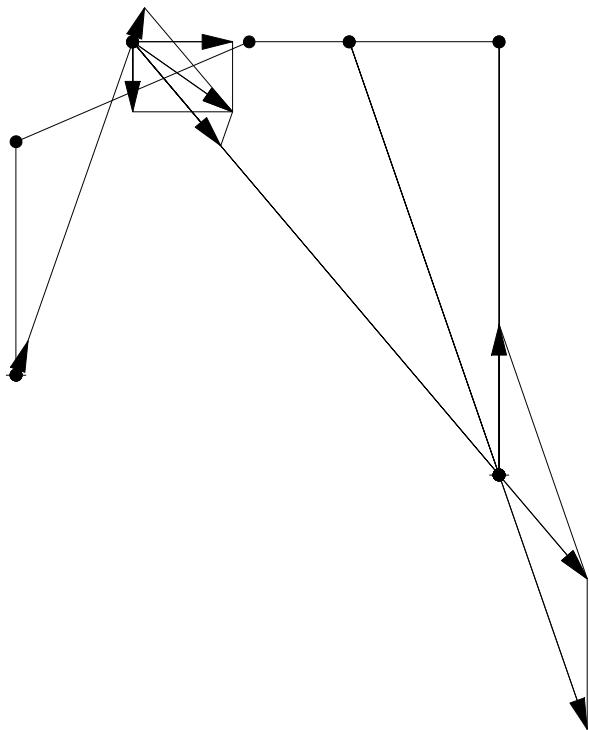
kKomplout = {2.64212, -7.6328} kKomp2out = {0., 4.51029}
```



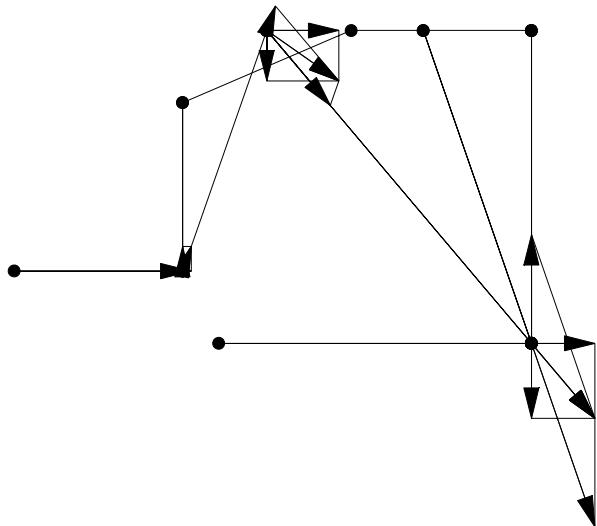
```
sk12 = Show[sk8,sk10];
```



```
sk12 = Show[sk8,sk11];
```



```
sk13 = Show[sk8,sk9,sk10,sk11];
```



Achtung: Im letzten Diagramm ist der Normalvektor zu (p1, p2) eingezeichnet. Der wagrechte Pfleil unten links erscheint daher überzeichnet. Das ändert aber nichts an den numerischen Werten.

(kKomp1out = {0,1.02251} kKomp2out = {0.357878,0}, 1.02251 > 0.357878)

```
kraftPunktZerlegSenkr[p3,p1,kResP1];
```

```
kKomplout = {5.98064, 8.54378} kKomp2out = {-2.40186, 1.68131}
```

```

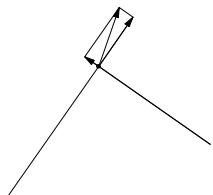
kResP13 = kKomplout;
kResP13n = kKomp2out;
{kResP13,kResP13n}

{{5.98064, 8.54378}, {-2.40186, 1.68131}};

kraftPunktZerlegSenkr[p3,p1,kResP1 strecken];
sk14 = Show[graphZerlegSenkr[p3,p1,kResP1 strecken],display];

kKomplout = {0.598064, 0.854378}  kKomp2out = {-0.240186, 0.168131}

```

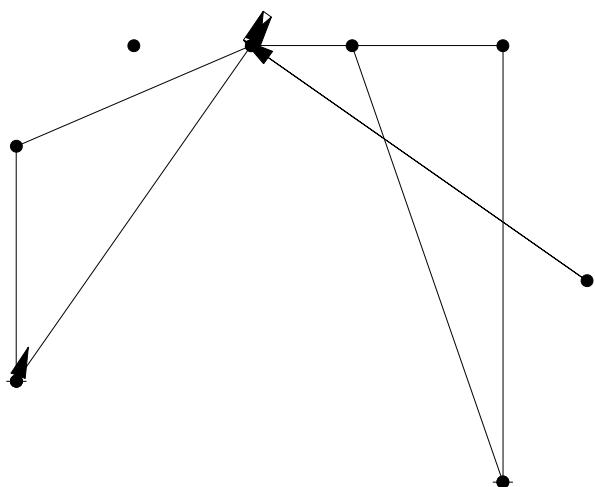


```

listP1={p1,p2,p51,p3,p4,p5,p6};
listL1={p1,p2, p23, p3,p4,p5,p6,p4};
sk1a = showPointsLines[listP1, listL1];

sk13 = Show[sk1a,sk2,sk6,sk14,display];

```



```

kraftPunktZerlegSenkr[p3,p6,kResP6];

kKomplout = {20.1138, -34.864}  kKomp2out = {6.3074, 3.63888}

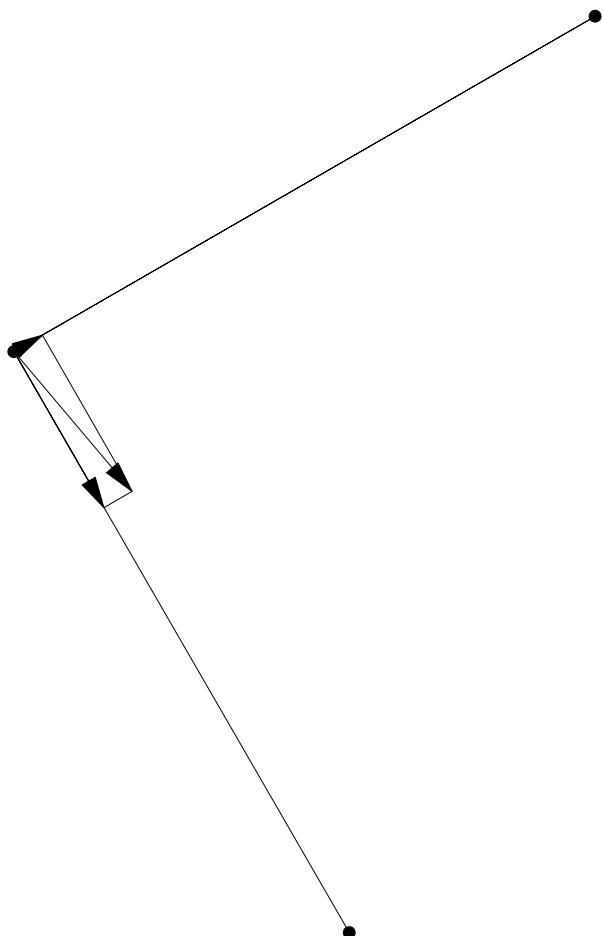
```

```
kResP63 = kKomplout;
kResP63n = kKomp2out;
{kResP63,kResP63n}

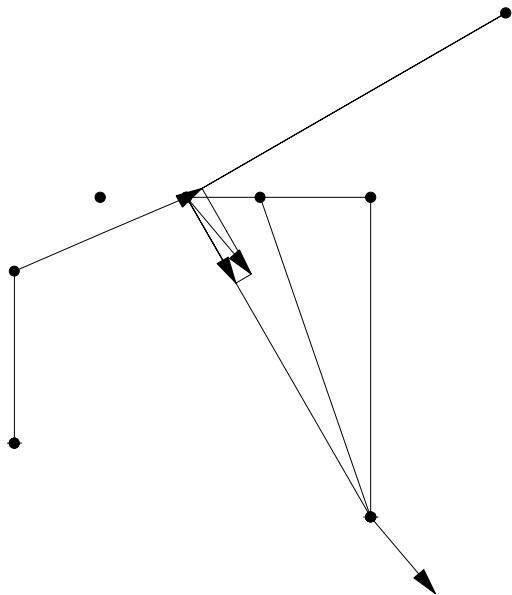
{{20.1138, -34.864}, {6.3074, 3.63888}};

kraftPunktZerlegSenkr[p3,p6,kResP6 strecken];
sk15 = Show[graphZerlegSenkr[p3,p6,kResP6 strecken],display];

kKomplout = {2.01138, -3.4864}  kKomp2out = {0.63074, 0.363888}
```



```
sk16 = Show[sk1a,sk2,sk7,sk15,display];
```

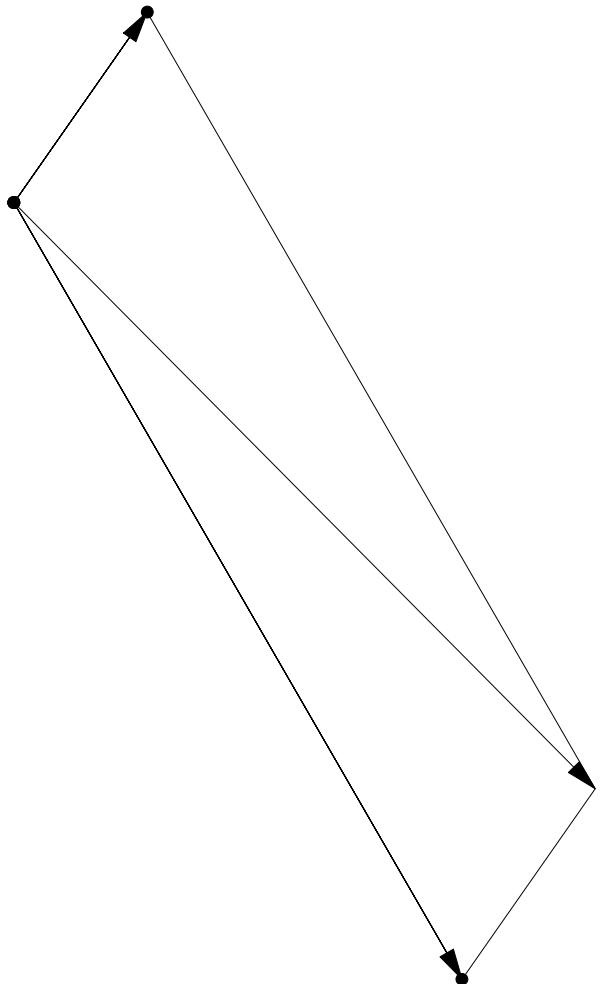


```
KRes3 = kResP13+kResP63
```

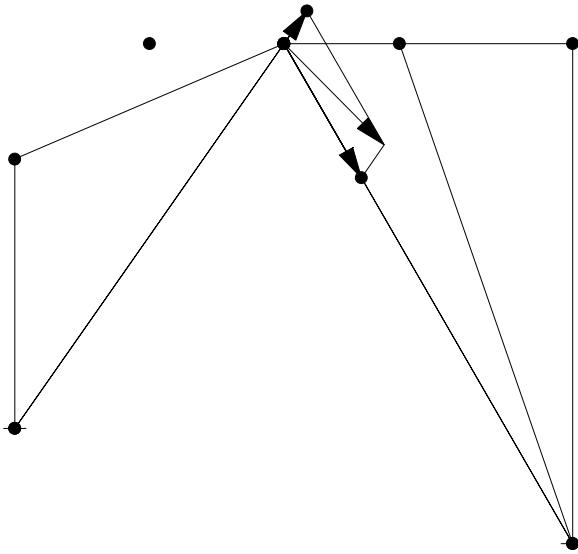
```
{26.0945, -26.3202}
```

```
Kres3S=KRes3 strecken;
kResP13S=kResP13 strecken;
kResP63S=kResP63 strecken;
kraftPunktZerleg[p3,p3+kResP13S,p3,p3+kResP63S,Kres3S];
graphZerleg[p3,p3+kResP13S,p3,p3+kResP63S,Kres3S];
sk17 = Show[graphZerleg[p3,p3+kResP13S,p3,p3+kResP63S,Kres3S], display];

kKomplout = {0.598064, 0.854378} kKomp2out = {2.01138, -3.4864}
```



```
listP1={p1,p2,ps1,p3,p4,p5,p6};  
listL1={p1,p2,p3,p1,p3,p4,p5,p6,p3,p6,p4};  
sk1b = showPointsLines[listP1, listL1];  
sk18 = Show[sk1b,sk2,sk17,display];
```



4. Putzmaschine ==> Alles löschen ==> Modul muss nochmals initiiert werden (alles gelöscht)

```
(* 1 *)  
Remove["Global`*"];  
(* 2 *)  
Off[];  
Off[General::spell];  
Off[General::spell1];
```