

Josef Böhm, Würmla, Österreich

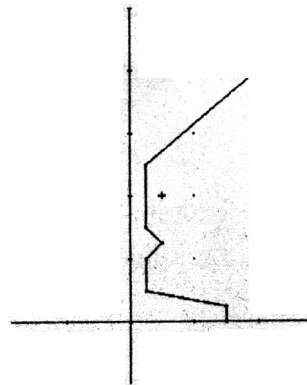
## Workshop: Grafik mit *DERIVE*

Der wesentliche Teil dieses Workshops behandelte die Darstellung von dreidimensionalen Objekten in Parallelprojektion. Neben der Abbildung eines einfachen Körpers (Doppelpyramide) war es das Ziel, das Gitternetz einer Drehfläche mit einem frei gewählten Meridian in Parallelprojektion zu in das *DERIVE* - 2D-Fenster zu bringen.

Das *DERIVE* - Listing wurde gemeinsam erarbeitet. Alle Teilnehmer konnten schließlich ihre Wunschfiguren darstellen. Im Anhang werden zwei Bilder einer möglichen Fortsetzung dieser Workshops gezeigt: die Darstellung der Objekte in Zentralprojektion.

Notation:=Decimal

NotationDigits:=3



Wir erzeugen zuerst punktweise den Meridian einer Drehfläche. Diesen Polygonzug sehen Sie in der ersten Abbildung.

```
m:=[[0,0],[3,0],[3,0.5],[0.5,1],[0.5,2],[1,2.5],[0.5,3],[0.5,5],[4,8]]
```

Die Funktion ROTFL(m,n) erzeugt alle Punkte, die bei Drehungen des Meridians m um einen Winkel von jeweils  $2\pi/n$  entstehen.

```
ROTFL(m,n):=VECTOR(VECTOR([m SUB [i,1]*COS(phi),m SUB [i,1]*SIN(phi)
                        m SUB [i,2]],phi,0,2*pi,2*pi/n),i,1,DIMENSION(m))
```

ROTFL(m,8) (erzeugt erst die Raumpunkte)

Diese Punkte müssen nun auf die Bildebene unter einem bestimmten Winkel und mit einem Verkürzungsfaktor projiziert werden, um einen Schrägriß zu erhalten.

PROJ(k,θ) ist die Projektionsmatrix, die auf das Objekt obj angewendet wird.

```
PROJ(k,theta):=[[1,0],[k*COS(theta*deg),k*SIN(theta*deg)],[0,1]]
```

```
BILD(obj,k,theta):=VECTOR(obj SUB 1 . PROJ(k,theta),i,1,DIMENSION(obj))
```

Die Funktion Bild liefert ein Abbild der waagrechten Parallelkreise.

COPROJECTION(v) ist eine Funktion aus der Utilitydatei GRAPHICS.MTH und hilft, die Punkte des Objektes in einer anderen Reihenfolge zu verbinden, sodaß die Meridiane dargestellt werden können.

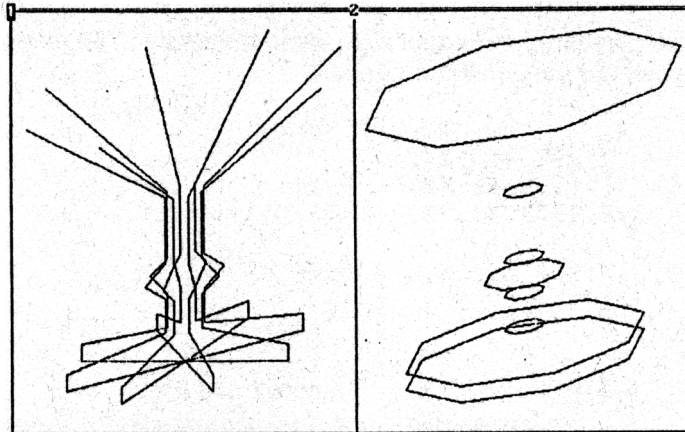
```
COPROJECTION(v):=VECTOR(VECTOR(u SUB n_,u_,v),n_,DIMENSION(v SUB 1))
```

```
MER(obj,k,theta):=COPROJECTION(BILD(obj,k,theta))
```

BILD(ROTFL(m,20),0.75,35) (approX und Plot → rechtes Fenster)

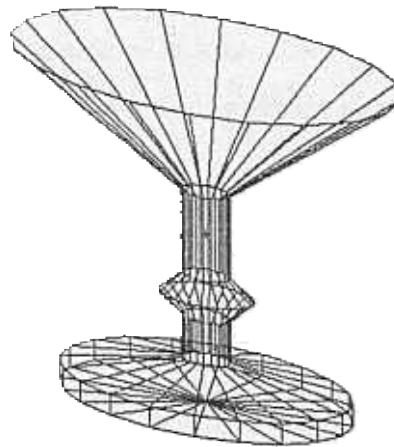
MER(ROTFL(m,20),0.75,35) (approX und Plot → linkes Fenster)

```
KOMPLETT(obj,k,theta):=APPEND(BILD(obj,k,theta),MER(obj,k,theta))
```



KOMPLETT(objekt,faktor,winkel) fügt beide Bilder zusammen und erzeugt auf Anhieb das komplette Gittermodell:

KOMPLETT (ROTFL (m, 20) , 0.7, 150)



#### Parallelprojektion einer Doppelpyramide:

Die Figur ist gegeben durch ihre Ecken (Matrix dp) und einer Liste, in der die Punkte der Reihe nach angeführt sind, die verbunden den Körper ergeben. (Vektor zug). Die Funktion FIGUR(p,z) verbindet die in p vorhandenen Punkte in der durch z vorgegebenen Reihenfolge.

```
dp:=[ [2,2,-2], [-2,2,-2], [-2,-2,-2], [2,-2,-2], [2,2,2], [-2,2,2], [-2,-2,2],
      [2,-2,2], [0,0,6], [0,0,-6]]
```

```
zug:=[1,2,10,1,5,9,8,5,6,2,3,10,4,8,7,6,9,7,3,4,1]
```

```
FIGUR(p,z):=APPEND(VECTOR({ELEMENT(p,ELEMENT(z,k))},k,DIMENSION(z))
```

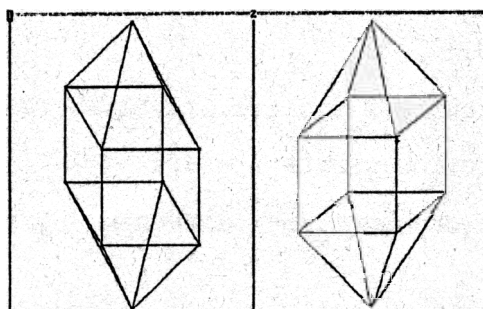
```
FIGUR(dp,zug)
```

```
BILD(FIGUR(dp,zug),2/3,40)
```

(Schrägriß rechts)

```
BILD(FIGUR(dp,zug),3/4,110)
```

(Schrägriß links)



### Ein Diamant entsteht

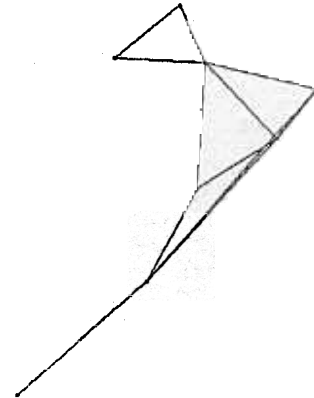
Eine andere Form eines „Drehkörpers“ ergibt sich, wenn man einen Teil eines „Diamanten“ `d_teil` konstruiert, der das Achtel der ganzen Figur darstellt.

```
d_teil:=[[2,-0.83,1.7],[2.9,0,1.2],
        [3.7,-1.53,0],[4,0,0],[2,0,-2.2],
        [0,0,-4],[2,0.83,1.7],[3.7,1.53,0]]
```

```
d_zug:=[7,1,2,7,2,3,4,8,2,4,3,5,4,8,5,6]
```

```
FIGUR(d_teil,d_zug)
```

```
BILD(FIGUR(d_teil,d_zug),0.8,40)      (approX und Plot
gibt das Bild des Achteldiamanten)
```



`ROT_Z(θ)` ist die Rotationsmatrix für eine Drehung um den Winkel  $\theta$ , mit der z-Achse als Drehachse.

```
ROT_Z(theta):=[[COS(theta),SIN(theta),0], [-SIN(theta),COS(theta),0],
               [0,0,1]]
```

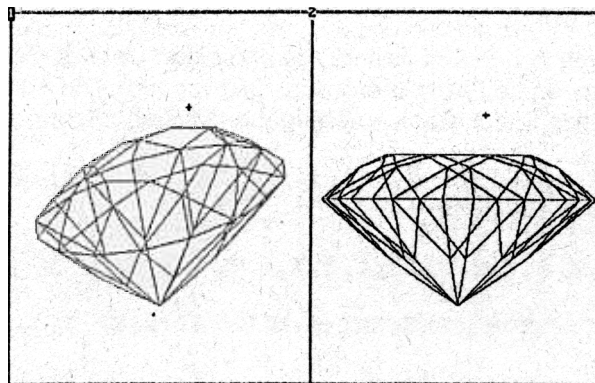
`DREH(obj,n)` dreht das Objekt `obj` `n` mal um jeweils den Winkel  $2\pi/n$  und bildet so die rotationssymmetrische Figur.

```
DREH(obj,n):=VECTOR(obj . ROT_Z(2*pi*k/n),k,1,n)
```

```
DREH(FIGUR(d_teil,d_zug),8)
```

```
BILD(DREH(FIGUR(d_teil,d_zug),8),0.8,0)      (ergibt eine besondere Ansicht)
```

```
BILD(DREH(FIGUR(d_teil,d_zug),8),0.8,40)      (ergibt wieder den Schrägriß)
```



Besondere Ansichten sind Grund- und Aufriß. Die Vorgangsweise ist aus dem Listing deutlich ersichtlich und bedarf keines besonderen Kommentars:

Zuerst der Grundriß für `BILD` und `MER`:

```
grr:=[[1,0],[0,1],[0,0]]
```

```
GRUNDR_B(obj):=VECTOR(obj SUB i . grr,i,1,DIMENSION(obj))
```

```
GRUNDR_M(obj):=COPROJECTION(GRUNDR_L(obj))
```

```
GRUNDRISS(obj):=APPEND(GRUNDR_B(obj),GRUNDR_M(obj))
```

Dann der Aufriß:

```
aufr:=[[1,0],[0,0],[0,1]]
```

```
AUFR_L(obj):=VECTOR(obj SUB i . aufr,i,1,DIMENSION(obj))
```

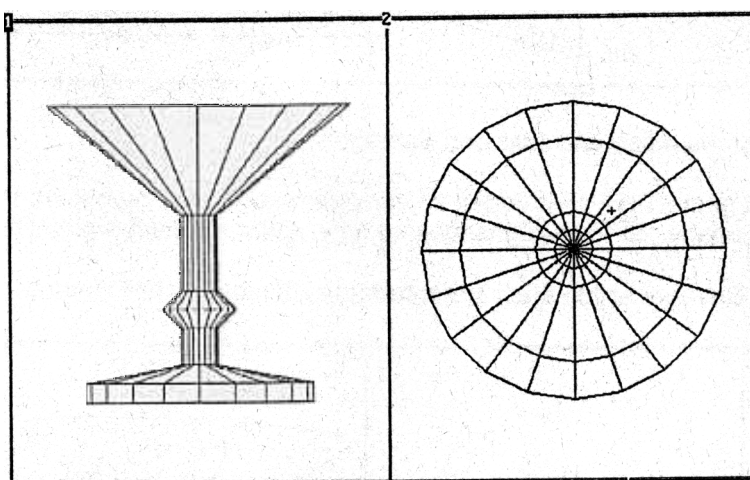
```
AUFR_M(obj):=COPROJECTION(AUFR_L(obj))
```

```
AUFRISS(obj):=APPEND(AUFR_L(obj),AUFR_M(obj))
```

Jetzt zeichnen wir Grund- und Aufriß unseres Glases:

```
GRUNDRISS(ROTFL(m,20))
```

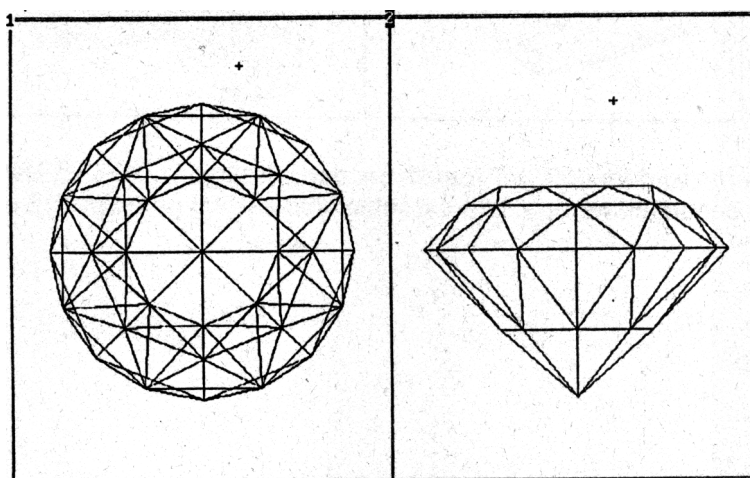
```
AUFRISS(ROTFL(m,20))
```



Auch für den Diamanten lassen sich die beiden Risse einfach ermitteln:

```
AUFRISS(DREH(FIGUR(d_teil,d_zug),8))
```

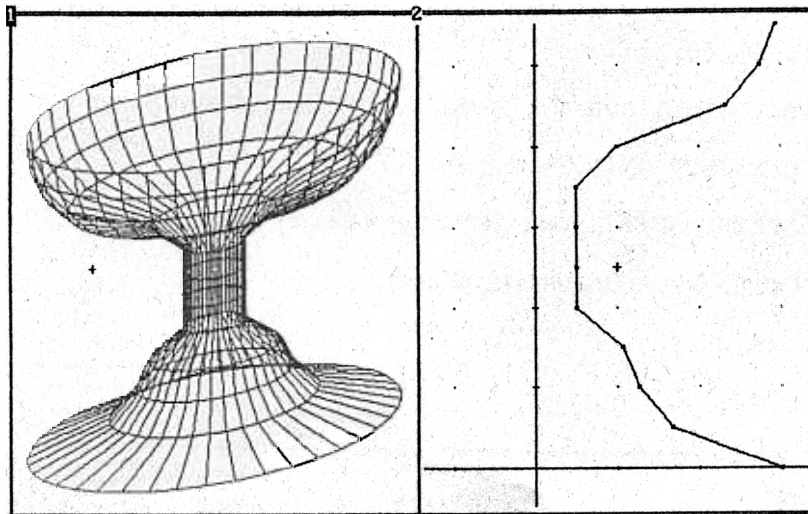
```
GRUNDRISS(DREH(FIGUR(d_teil,d_zug),8))
```



Der Meridian kann aber auch durch eine Funktion definiert werden, wie das folgende Beispiel deutlich macht. KOMPLETT liefert sofort ein ansprechendes Bild.

```
wein:=VECTOR([IF(t<2,3-SQRT(4-(t-2)^2),IF(t<4,0.5,1+SQRT(4-(t-6)^2))),t],  
t,0,6,0.5)
```

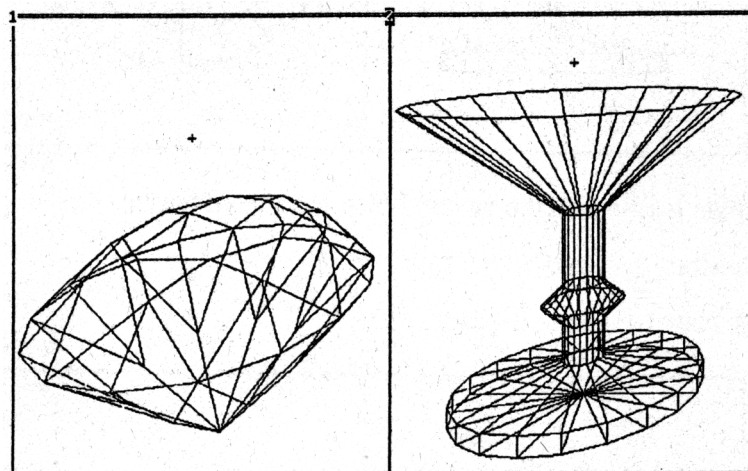
```
KOMPLETT(ROTFL(wein,40),0.7,35)
```



Rechts sehen Sie den Meridian und links den Schrägriß.

Leider war nicht Zeit genug die Projektionen zu verallgemeinern. In Frage kommt natürlich die Parallelprojektion durch Angabe der Projektionsrichtung oder weiter die Zentralprojektion.

Unser Glas und der Diamant sollen auch in Zentralprojektion dargestellt werden.



Das vorgestellte Thema wird im *DERIVE* Newsletter dem Organ der *DERIVE* User Group in einer der nächsten Ausgaben behandelt werden. Interessenten können sich gerne beim Leiter des Workshops melden.

Josef Böhm  
*DERIVE* User Group  
 D'Lust 1  
 A-3042 Würmla  
 Österreich

**Hinweis:** Falls Sie noch nicht über *DERIVE* 3.x verfügen, werden Sie mit `m SUB[i, j]` oder mit `obj SUB i` nicht viel anfangen können. Das ist aber nur eine elegantere Schreibweise für die *ELEMENT*-Funktion: `ELEMENT(m,i,j)`, bzw. `ELEMENT(obj,i)` sind die entsprechenden Schreibweisen in Prä-*DERIVE* 3.0 Versionen.