

## 3D MetaPost Ergänzungen – tf3d.mp

<code>ddd</code> <code>x_part,y_part,z_part</code> <code>length P, abs P</code> <code>A dotprod B, A crossprod B</code> <code>unitvector P</code> <code>rotatedaround(R,angle)</code>	3D-Punkt erzeugen, Typstest (alles analog zu <code>pair</code> ) Komponenten von 3D-Punkten Länge von P (=Euklidische Norm) Skalarprodukt, Kreuzprodukt Einheitsvektor berechnen Drehung um die Achse R mit Winkel <code>angle</code>
<code>dddpoint(expr x,y,z)</code> <code>parallelperspective(xangle,xscale,yangle,yscale,   zangle,zscale)</code> <code>kavaliersperspective(scale)</code> <code>centralperspective(xscale,yscale,zscale,P)</code>	Projiziert einen 3D-Punkt in die Ebene Parallelprojektion einstellen (Achsenwinkel und Achsen- skalierung) – wirkt auf nachfolgende <code>dddpoint</code> -Befehle Kavaliersperspektive (eine spezielle Parallelprojektion) Zentralprojektion – Projektion auf die <i>y-z</i> -Ebene (Ach- senskalierung und Beobachtungspkt. P werden festgelegt)
<code>dddfuncpic(function)(xvar,yvar)(xmin,xmax,xgrid,   ymin,ymax,ygrid)</code> <code>threecolorlighting(topcolor,xcolor,ycolor)</code> <code>graylighting(g)</code> <code>undersidecolor:=&lt;color&gt;</code> <code>boundary:=&lt;boolean&gt;</code>	3-dimensionaler Funktionsgraph  Färbung von 3D-Graphen einstellen Variable für die Farbe der Unterseite von 3D-Graphen Zellen bei 3D-Graph umranden

```

input tftools;
input tf3d;

vardef achsen(expr xl,yl,zl)=
  ahlength:=8pt; ahangle:=30pt; fullarrows:=false;
  pickup pencircle scaled 0.5pt;
  drawarrow (0,0)--dddpoint(xl,0,0);
  label.lft(btex  $x$  etex,dddpoint(xl,0,0));
  drawarrow (0,0)--dddpoint(0,yl,0);
  label.lrt(btex  $y$  etex,dddpoint(0,yl,0));
  drawarrow (0,0)--dddpoint(0,0,zl);
  label.urrt(btex  $z$  etex,dddpoint(0,0,zl));
enddef;

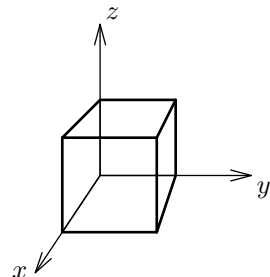
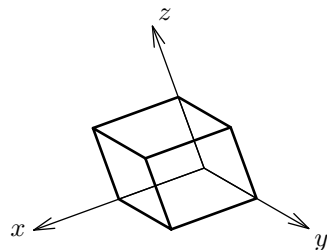
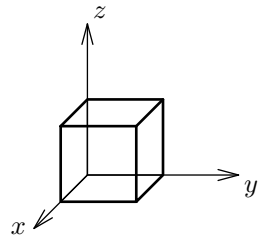
vardef wuerfel=
  pair A,B,C,D,E,F,G,H;
  A:=dddpoint(1,0,0); E:=dddpoint(1,0,1);
  B:=dddpoint(1,1,0); F:=dddpoint(1,1,1);
  C:=dddpoint(0,1,0); G:=dddpoint(0,1,1);
  D:=dddpoint(0,0,0); H:=dddpoint(0,0,1);
  pickup pencircle scaled 1pt;
  draw B--C--G--H--E--A--B--F--E; draw F--G;
  pickup pencircle scaled 0.5pt;
  draw A--D--C; draw D--H;
enddef;

beginfig(1);
  kavaliersperspective(1cm);
  achsen(2,2,2);
  wuerfel;
endfig;

beginfig(2);
  parallelperspective(-160,1.2cm,-30,0.8cm,110,1cm);
  achsen(2,2,2);
  wuerfel;
endfig;

beginfig(3);
  centralperspective(0.8cm,1cm,1cm,(4cm,2cm,3cm));
  achsen(1.5,2,2);
  wuerfel;
endfig;

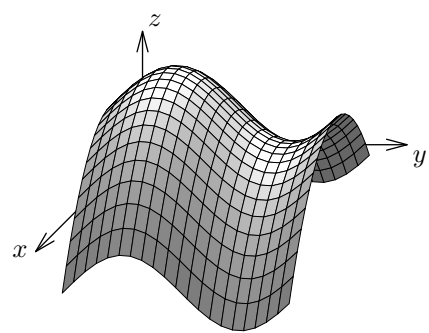
```



```

beginfig(4);
  kavaliersperspective(0.5cm);
  achsen(8,7,3);
  graylighting(1.2);
  pickup pencircle scaled 0.3pt;
  draw dddfuncpic(sin(y)+1.5x-0.3x*x)(x,y)(0,6,20,0,6,20);
endfig;

```



```

beginfig(5);
  centralperspective(0.3cm,0.5cm,0.5cm,(4cm,5cm,2cm));
  achsen(6,7,4);
  threecolorlighting(red+blue,0.6*blue,0.8*red);
  undersidecolor:=0.6green;
  boundary:=false;
  pickup pencircle scaled 0.3pt;
  draw dddfuncpic(sin(y)+1.5x-0.3x*x)(x,y)
    (0,6,20,0,6,20);
endfig;

```

