

3d-Funktionsplotter für xfig

Copyright 2006 Xenia Rendtel

4. Januar 2006 Version 1.0

Zusammenfassung

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 2 der Lizenz oder (nach Ihrer Option) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE IRGENDNEINE GARANTIE, sogar ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.

Sie sollten ein Exemplar der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110, USA.

Bei Fehlern schreiben Sie diese einfach an webmaster@rendtel.de

Inhaltsverzeichnis

1	3d-Funktionsplotter für xfig	2
1.1	Möglichen Befehle	2
1.1.1	Skalierung	2
1.1.2	Koordinatensystem	2
1.1.3	Gitter	3
1.1.4	Vektoren	4
1.1.5	Markierungen	4
1.1.6	XYZ-Markierung	5
1.1.7	Verschiedene Objekte zeichnen	6
1.2	Quellcode	7

1 3d-Funktionsplotter für xfig

Um mir das Leben als Mathematiklehrerin einfacher zu machen habe ich mir das Perl-Skript `3dplotter.pl` geschrieben, welches einige Objekte im dreidimensionalen in xfig automatisch zeichnet.

Dazu müssen in einer `txt`-Datei verschiedene Objekte angegeben, z.B. Vektoren, und diese `txt`-Datei wird dann durch das Perl-Skript auf der Kommandozeile mit dem Aufruf

```
./3dplotter.pl datei.txt
```

in eine `datei.fig` verwandelt.

Im folgenden erläutere ich zunächst die möglichen Befehle und darauf folgt dann der Quellcode.

1.1 Möglichen Befehle

Um eine xfig-Objekte zu erzeugen gibt es einige Befehle die in eine `txt`-Datei geschrieben werden muss. In der `3dbefehle.txt`-Datei stehen diese alle zusammengefasst. Hier werden einmal alle Befehle mit Beispielen angesprochen.

1.1.1 Skalierung

Als erstes sollte in der `txt`-Datei enthalten sein, wie groß man die Datei skaliert haben möchte. Dazu gibt es zwei Möglichkeiten

```
skalierung <faktor>  
skalierung x=<faktor> y=<faktor>
```

Wenn man nur `skalierung 3` setzt, so wird der Skalierungsfaktor für die x - und y -Achse auf den gleichen Wert gesetzt.

1.1.2 Koordinatensystem

Im dreidimensionalen können die drei Achsen über die folgenden Befehle aufgerufen werden:

```
xachse3d layer=<n> bereich=<xmin>,<xmax>,<schrittweite>  
skala=<keine | nurstriche | nureins | normal> text=<"achsenbeschriftung">
```

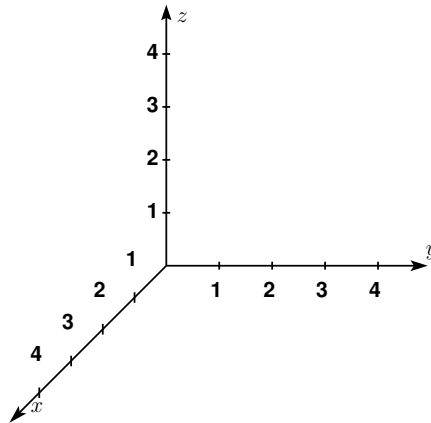
Analog für die anderen beiden Achsen. Dabei hat man die Möglichkeit den Layer einzustellen und die Bereiche der Achsen über `xmin` und `xmax` anzugeben. Die `schrittweite` gibt an, in welchem Abstand die Achsenbeschriftung und kleine Linien gezeichnet werden sollen. Diese kann man über `skala` einstellen und `text` gibt an, welche Beschriftung an der Achse stehen soll.

Analog gibt es die Befehle `yachse3d` und `zachse3d`.

Beispiel

skalierung 1

```
xachse3d layer=10 bereich=0,5,1 skala=normal text="$x$"  
yachse3d layer=10 bereich=0,5,1 skala=normal text="$y$"  
zachse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
```



1.1.3 Gitter

Mit dem Befehl

```
gitter [layer=<n>] bereich=<xmax>,<ymax>,<zmax>,<schrittweite>
```

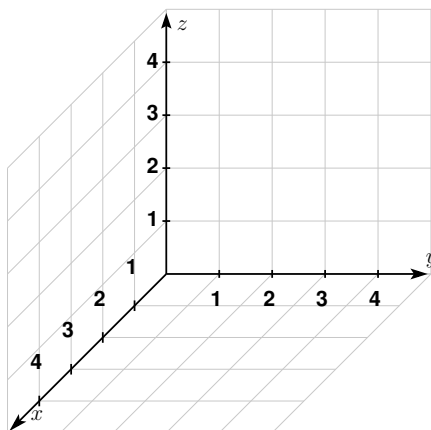
kann ein Gitter im 3d-Koordinatensystem gezeichnet werden.

Beispiel

skalierung 1

```
xachse3d layer=10 bereich=0,5,1 skala=normal text="$x$"  
yachse3d layer=10 bereich=0,5,1 skala=normal text="$y$"  
zachse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
```

```
gitter bereich=5,5,5,1
```



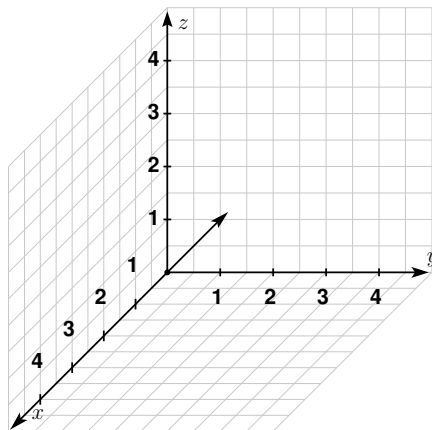
1.1.4 Vektoren

Es können ebenfalls Vektoren gezeichnet werden. Die Vektoren sind über ihre Anfangskoordinate (x_1, y_1, z_1) und Endkoordinate (x_2, y_2, z_2) definiert. Der Befehl lautet:

```
plotvektor3d [layer=<n>] [farbe=<n>]
             k1=<xstart>,<ystart>,<zstart>
             k2=<xende>,<yende>,<zende>
```

Beispiel

```
skalierung 1
gitter bereich=5,5,5,0.5
xachse3d layer=10 bereich=0,5,1 skala=normal text="$x$"
yachse3d layer=10 bereich=0,5,1 skala=normal text="$y$"
zachse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
plotvektor3d layer=80 farbe=0 k1=0,0,0 k2=3,3,3
```



1.1.5 Markierungen

Es gibt mehrere Möglichkeiten eine Koordinate zu markieren oder auch noch eine weitere Gerade einzuzeichnen. Mit dem Befehl

```
punktezeichnen3d [layer=<n>] [farbe=<n>]
                 k1=<xstart>,<ystart>,<zstart>
                 k2=<xende>,<yende>,<zende>
                 [verbunden=<ja | nein>]
```

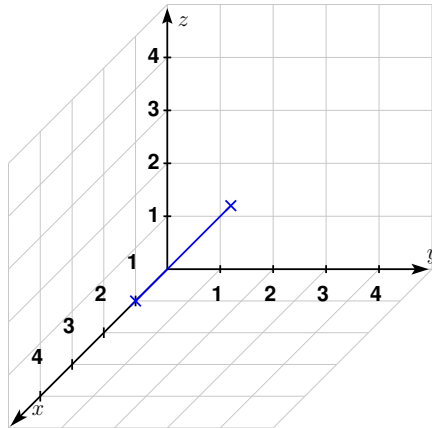
kann man zwei Punkte einzeichnen, die standardmäßig verbunden sind. Setzt man `verbunden=nein`, so werden die beiden Punkte einzeln gezeichnet.

Beispiel

```
skalierung 1
xachse3d layer=10 bereich=0,5,1 skala=normal text="$x$"
```

```
yachse3d layer=10 bereich=0,5,1 skala=normal text="$y$"
zachse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
```

```
gitter bereich=5,5,5,1
punktezeichnen3d layer=10 farbe=1 k1=1,0,0 k2=3,3,3
```



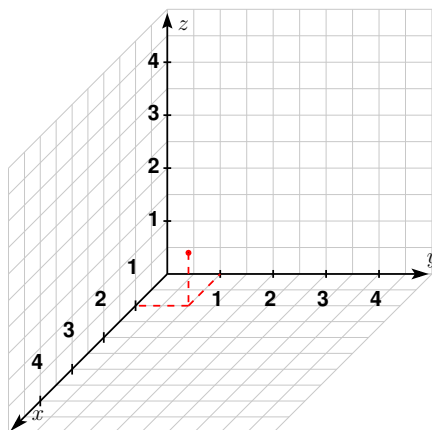
1.1.6 XYZ-Markierung

Mit dem Befehl `xyzmarkierung` kann man eine einzelne Koordinate einzeichnen oder auch diesen Punkt in die jeweiligen Ebenen projizieren.

```
xyzmarkierung [layer=<n>] [farbe=<n>]
               k=<x>,<y>,<z> [linie=<xy | xz | yz | xyz>]
```

Beispiel

```
skalierung 1
gitter bereich=5,5,5,0.5
xachse3d layer=10 bereich=0,5,1 skala=normal text="$x$"
yachse3d layer=10 bereich=0,5,1 skala=normal text="$y$"
zachse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
xyzmarkierung layer=8 farbe=4 k=1,1,1 linie=xyz
```



1.1.7 Verschiedene Objekte zeichnen

Es gibt ebenfalls die Möglichkeit Rechtecke und Dreiecke im Koordinatensystem zu zeichnen. Die beiden Objekte können verschieden farbig gezeichnet werden. Angegeben werden sie jeweils über ihre Eckpunkte.

Hier kommen die beiden Befehle:

1. Der Dreiecksbefehl:

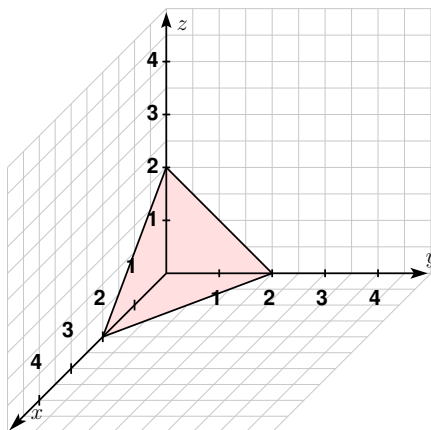
```
dreieck3d [layer=<n>] [farbe=<farbe>,<gefüllt>,<füllfarbe>]
          (<x1>,<y1>,<z1>) (<x2>,<y2>,<z2>) (<x3>,<y3>,<z3>)
```

2. Der Rechteckbefehl

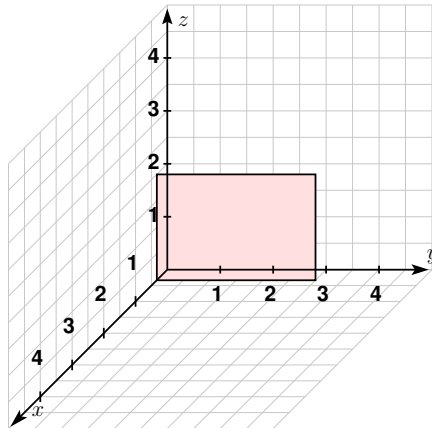
```
rechteck3d [layer=<n>] [farbe=<farbe>,<gefüllt>,<füllfarbe>]
           (<x1>,<y1>,<z1>) (<x2>,<y2>,<z2>) (<x3>,<y3>,<z3>)
           (<x4>,<y4>,<z4>)
```

Beispiele

```
skalierung 1
gitter bereich=5,5,5,0.5
achse3d layer=10 bereich=0,5,1 skala=normal text="$x$"
achse3d layer=10 bereich=0,5,1 skala=normal text="$y$"
achse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
dreieck3d (0,0,2) (0,2,0) (2,0,0)
```



```
skalierung 1
gitter bereich=5,5,5,0.5
achse3d layer=10 bereich=0,5,1 skala=normal text="$x$"
achse3d layer=10 bereich=0,5,1 skala=normal text="$y$"
achse3d layer=10 bereich=0,5,1 skala=normal text="$z$"
rechteck3d (2,4,1) (2,1,1) (2,1,3) (2,4,3)
```



1.2 Quellcode

Hier kommt nun der Quellcode:

```
#!/usr/bin/perl -w

#####
# 3d-Funktionsplotter Skript zum erstellen von 3d-Grafiken in Xfig
# Copyright (C) 2006 Xenia Rendtel
#
# Dieses Programm ist freie Software. Sie können es unter den Bedingungen der
# GNU General Public License, wie von der Free Software Foundation
# veröffentlicht, weitergeben und/oder modifizieren, entweder gemäß Version 2
# der Lizenz oder (nach Ihrer Option) jeder späteren Version.
#
# Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, daß es Ihnen
# von Nutzen sein wird, aber OHNE IRGENDNEINE GARANTIE, sogar ohne die implizite
# Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN
# ZWECK. Details finden Sie in der GNU General Public License.
#
# Sie sollten ein Exemplar der GNU General Public License zusammen mit diesem
# Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110, USA.
#
# Bei Fehlern schreiben Sie diese einfach an
# webmaster@rendtel.de
#####

#####
##### Letzte Änderung: 03.01.2006
#####

use POSIX qw /floor ceil /;
use Math::Trig;

#####
##### Konstanten und Variablen
#####

use constant {
    DASHED => 1,
    SOLID  => 0,
```

```

DOTTED           => 2,
DASHDOTTED       => 3,
DASHDOUBLEDOTTED => 4,
DASHTRIPPLEDOTTED => 5,
DFONT             => 0,
ROMAN            => 1,
BOLD             => 2,
ITALIC           => 3,
SANSSERIF        => 4,
NARROWHELVETICABOLD => 18,
ALIGNLEFT        => 0,
ALIGNCENTER      => 1,
ALIGNRIGHT       => 2,
PAPERX           => 29.7,
PAPERY           => 21,
ZEICHENDICKE     => 1,
LINIENDICKE      => 2,
XFIGSCALE        => 450,
EBENE            => +0.6
};

#####
##### Funktionen für das öffnen und schließen von
##### xfig-Dateien
#####

sub openfig {
  my $name=shift (@_);

  open FIGFILE, ">".$name;
  print FIGFILE "#FIG 3.2 Produced by script\n";
  print FIGFILE "Landscape\n";
  print FIGFILE "Center\n";
  print FIGFILE "Metric\n";
  print FIGFILE "A4\n";
  print FIGFILE "100.00\n";
  print FIGFILE "Single\n";
  print FIGFILE "-2\n";
  print FIGFILE "1200 2\n";
}

sub closefig {
  close FIGFILE;
}

#####
##### Positionen bestimmen in Xfig
#####

sub figx {
  my $x=shift (@_);
  return $x*XFIGSCALE;
}

sub figy {
  my $y=shift (@_);

```

```

    return (PAPERY-$y)*XFIGSCALE;
}

#####
##### Öffnen von verschiedenen Objektgruppen in xfig
#####

# eine Polyline wird geöffnet. Dabei muss angegeben werden, wie viele Punkte
# die Linie hat
sub openpline {
    my $layer=shift(@_);
    my $linestyle=shift(@_);
    my $thickness=shift(@_);
    my $farbe=shift(@_);
    my $points=shift(@_);

    printf(FIGFILE "2 1 %i %i %i 7 %i -1 -1 4.000 0 0 -1 0 0 %i\n",
           $linestyle, $thickness, $farbe, $layer, $points);
}

# eine Box wird geöffnet. Möchte man ein Rechteck zeichnen, muss man der Box 5
# Punkte übergeben und mit plinepoint fünf Punkte angeben. Die vier Eckpunkte
# eines Rechtecks und den Anfangspunkt als fünften Punkt.
sub openbox {
    my $layer=shift(@_);
    my $linestyle=shift(@_);
    my $thickness=shift(@_);
    my $farbe=shift(@_);
    my $gefüllt=shift(@_);
    my $füllfarbe=shift(@_);
    my $points=shift(@_);

    printf(FIGFILE "2 3 %i %i %i %i %i -1 %i 0.000 0 0 -1 0 0 %i\n",
           $linestyle, $thickness, $farbe, $füllfarbe, $points);
}

# Text wird geöffnet
sub opentext {
    my $layer=shift(@_);
    my $font=shift(@_);
    my $fontsize=shift(@_);
    my $align=shift(@_);
    printf(FIGFILE "4 %i 0 %i -1 %i %i 0.0000 2 225 525 ",
           $align, $layer, $font, $fontsize);
}

# Postscripttext wird geöffnet
sub openpstext {
    my $layer=shift(@_);
    my $font=shift(@_);
    my $fontsize=shift(@_);
    my $align=shift(@_);
    printf(FIGFILE "4 %i 0 %i -1 %i %i 0.0000 4 225 525 ",
           $align, $layer, $font, $fontsize);
}

```

```

# Ein Pfeil wird geöffnet
sub openarrow {
  my $layer=shift(@_);
  my $linestyle=shift(@_);
  my $thickness=shift(@_);
  my $arrowtype=shift(@_);
  my $arrowstyle=shift(@_);
  my $arrowthick=shift(@_);
  my $arrowwidth=shift(@_);
  my $arrowheight=shift(@_);
  my $farbe=shift(@_);
  my $points=shift(@_);

  printf(FIGFILE "2 1 %i %i %i 7 %i -1 -1 0.000 0 0 -1 1 0 %i\n      %i %i
%i %i %i\n",
        $linestyle, $thickness, $farbe, $layer, $points, $arrowtype,
        $arrowstyle, $arrowthick, $arrowwidth, $arrowheight);
}

# Compound öffnen und schließen
sub opencompound {
  my $xobenlinks=shift(@_);
  my $yobenlinks=shift(@_);
  my $xuntenrechts=shift(@_);
  my $yuntenrechts=shift(@_);
  printf(FIGFILE "6 %i %i %i %i\n ",
        figx($xobenlinks),
        figy($yobenlinks),
        figx($xuntenrechts),
        figy($yuntenrechts));
}

sub closecompound {
  printf(FIGFILE "-6\n");
}

#####
##### Schreiben von verschiedenen Objekten in xfig
#####

# Ein Kreis wird geschrieben
sub circle {
  my $layer=shift(@_);
  my $linestyle=shift(@_);
  my $thickness=shift(@_);
  my $x1=shift(@_);
  my $y1=shift(@_);
  my $radius=shift(@_);
  my $farbe=shift(@_);

  printf(FIGFILE "1 3 %i %i %i 20 %i -1 0 4.000 1 1 %i %i %i %i %i %i %i %i\n
",
        $linestyle, $thickness, $farbe, $layer, figx($x1), figy($y1),
        $radius,
        $radius, $x1, $y1, $x1, $y1);
}

```

```

# ein Rechteck wird gezeichnet. Angabe zwei gegenüberliegender Punkte genügt
sub rechteck3d {
  my $layer=shift(@_);
  my $x1=shift(@_);
  my $y1=shift(@_);
  my $z1=shift(@_);
  my $x2=shift(@_);
  my $y2=shift(@_);
  my $z2=shift(@_);
  my $x3=shift(@_);
  my $y3=shift(@_);
  my $z3=shift(@_);
  my $x4=shift(@_);
  my $y4=shift(@_);
  my $z4=shift(@_);
  my $farbe=shift(@_);
  my $gefüllt=shift(@_);
  my $fuellfarbe=shift(@_);

  openbox($layer,SOLID,LINIENDICKE,$farbe,$gefüllt,$fuellfarbe,5);
  plinepoint(x3d($x1,$y1,$z1)*$scalex,
             y3d($x1,$y1,$z1)*$scaley);
  plinepoint(x3d($x2,$y2,$z2)*$scalex,
             y3d($x2,$y2,$z2)*$scaley);
  plinepoint(x3d($x3,$y3,$z3)*$scalex,
             y3d($x3,$y3,$z3)*$scaley);
  plinepoint(x3d($x4,$y4,$z4)*$scalex,
             y3d($x4,$y4,$z4)*$scaley);
  plinepoint(x3d($x1,$y1,$z1)*$scalex,
             y3d($x1,$y1,$z1)*$scaley);
}

# Text wird geschrieben
sub ptext {
  my $x=shift(@_);
  my $y=shift(@_);
  my $text=shift(@_);

  printf(FIGFILE "%i %i %s \\001\n",
         figx($x), figy($y), $text);
}

# x und y-Koordinaten werden geschrieben
sub plinepoint {
  my $x=shift(@_);
  my $y=shift(@_);

  printf(FIGFILE "      %i %i\n",
         figx($x), figy($y));
}

#####
##### Farben werden geschrieben
#####

# neue RGB-Farbe definieren, mindestens Nr. 32
sub neuefarbe {

```

```

    my $nummer=shift (@_);
    my $r=shift (@_);
    my $g=shift (@_);
    my $b=shift (@_);

    printf(FIGFILE "0 %i %#02x%02x%02x\n",
           $nummer, $r, $g, $b);
}

# für eine garue Farbe sind die rgb-Werte alle gleich
sub neugrau {
    my $nummer=shift (@_);
    my $grau=shift (@_);
    neuefarbe ($nummer, $grau, $grau, $grau);
}

#####
##### Zeichenroutinen für zusammengesetzte Objekte
#####

# Ein Kreuzchen in (x,y) zeichnen
sub kreuzchen {
    my $layer=shift (@_);
    my $x=shift (@_);
    my $y=shift (@_);
    my $farbe=shift (@_);

    my $groesse=0.2;

    $groesse=$groesse/2;

    my $xlinks=$x-$groesse*$scalex;
    my $xrechts=$x+$groesse*$scalex;
    my $yunten=$y-$groesse*$scaley;
    my $yoben=$y+$groesse*$scaley;

    opencompound (abrunden ($xlinks, 0.1), aufrunden ($yoben, 0.1),
                  aufrunden ($xrechts, 0.1), abrunden ($yunten, 0.1));

    openline ($layer, SOLID, LINIENDICKE, $farbe, 2);
    plinepoint ($xlinks, $yunten);
    plinepoint ($xrechts, $yoben);
    openline ($layer, SOLID, LINIENDICKE, $farbe, 2);
    plinepoint ($xlinks, $yoben);
    plinepoint ($xrechts, $yunten);

    closecompound;
}

# Kreuzchen im 3-dimensionalen
sub kreuzchen3d {
    my $layer=shift (@_);
    my $x=shift (@_);
    my $y=shift (@_);
    my $z=shift (@_);
    my $farbe=shift (@_);

```

```

    kreuzchen($layer ,x3d($x,$y,$z),y3d($x,$y,$z),$farbe)
}

# Ein Pluszeichen in (x,y) zeichnen
sub pluszeichen {
    my $layer=shift(@_);
    my $x=shift(@_);
    my $y=shift(@_);
    my $farbe=shift(@_);

    my $groesse=0.2;

    $groesse=$groesse/2;

    my $xlinks=$x-$groesse*$scalex;
    my $xrechts=$x+$groesse*$scalex;
    my $yunten=$y-$groesse*$scaley;
    my $yoben=$y+$groesse*$scaley;

    opencompound( abrunden($xlinks,0.1), aufrunden($yoben,0.1),
                  aufrunden($xrechts,0.1), abrunden($yunten,0.1) );
    openline($layer,SOLID,LINIENDICKE,$farbe,2);
    plinepoint($xlinks,$y);
    plinepoint($xrechts,$y);
    openline($layer,SOLID,LINIENDICKE,$farbe,2);
    plinepoint($x,$yoben);
    plinepoint($x,$yunten);
    closecompound;
}

sub pluszeichen3d {
    my $layer=shift(@_);
    my $x=shift(@_);
    my $y=shift(@_);
    my $farbe=shift(@_);

    pluszeichen3d($layer ,x3d($x,$y,$z),y3d($x,$y,$z),$farbe)
}

#####
##### Funktionen zum auf- und abrunden auf das nächste
##### Vielfache einer gegebenen Zahl
#####

sub abrunden {
    my $wert=shift(@_);
    my $schritt=shift(@_);
    return $schritt*floor($wert/$schritt);
}

sub aufrunden {
    my $wert=shift(@_);
    my $schritt=shift(@_);
    return $schritt*ceil($wert/$schritt);
}

#####

```

```

##### Berechnungen für Vektoren
#####

# 3d- Koordinaten -> z-Achse liegt in den Raum hinein
sub xz3d {
  my $x=shift (@_);
  my $z=shift (@_);
  return $x-(EBENE*$z);
}

sub yz3d {
  my $y=shift (@_);
  my $z=shift (@_);
  return $y-(EBENE*$z);
}

# 3d- Koordinaten -> x-Achse liegt in den Raum hinein
sub x3d {
  my $x=shift (@_);
  my $y=shift (@_);
  my $z=shift (@_);

  return $y-(EBENE*$x);
}

sub y3d {
  my $x=shift (@_);
  my $y=shift (@_);
  my $z=shift (@_);
  return $z-(EBENE*$x);
}

#####
##### Erzeugen von einem räumlichen Gitter
#####

sub gitter {
  my $layer=shift (@_);
  my $xmax=shift (@_);
  my $ymax=shift (@_);
  my $zmax=shift (@_);
  my $abstand=shift (@_);

  my $xmin=0;
  my $ymin=0;
  my $zmin=0;
  my $farbe=32;
  my $xmingitter=abrunden($xmin,$abstand);
  my $xmaxgitter=aufrunden($xmax,$abstand);
  my $ymingitter=abrunden($ymin,$abstand);
  my $ymaxgitter=aufrunden($ymax,$abstand);
  my $zmingitter=abrunden($zmin,$abstand);
  my $zmaxgitter=aufrunden($zmax,$abstand);

  # Anfangs- und Endwerte für die Zeichenschleife
  my $anfangx=$xmingitter/$abstand;
  my $endex=$xmaxgitter/$abstand;
}

```

```

my $anfangy=$ymingitter/$abstand;
my $sendey=$ymaxgitter/$abstand;
my $anfangz=$zmingitter/$abstand;
my $sendez=$zmaxgitter/$abstand;
my $schritt;

# Gitter in den drei Ebenen
for ($schritt=$anfangy; $schritt<=$sendey; $schritt++) {
  openpline($layer,SOLID,ZEICHENDICKE,$farbe,2);
  plinepoint(x3d(0,$schritt*$abstand,$zmingitter)*$scalex,
             y3d(0,$schritt*$abstand,$zmingitter)*$scaley);
  plinepoint(x3d(0,$schritt*$abstand,$zmaxgitter)*$scalex,
             y3d(0,$schritt*$abstand,$zmaxgitter)*$scaley);
}
for ($schritt=$anfangz; $schritt<=$sendez; $schritt++) {
  openpline($layer,SOLID,ZEICHENDICKE,$farbe,2);
  plinepoint(x3d($xmingitter,0,$schritt*$abstand)*$scalex,
             y3d($xmingitter,0,$schritt*$abstand)*$scaley);
  plinepoint(x3d($xmaxgitter,0,$schritt*$abstand)*$scalex,
             y3d($xmaxgitter,0,$schritt*$abstand)*$scaley);
}
for ($schritt=$anfangx; $schritt<=$sendex; $schritt++) {
  openpline($layer,SOLID,ZEICHENDICKE,$farbe,2);
  plinepoint(x3d($schritt*$abstand,0,$zmingitter)*$scalex,
             y3d($schritt*$abstand,0,$zmingitter)*$scaley);
  plinepoint(x3d($schritt*$abstand,0,$zmaxgitter)*$scalex,
             y3d($schritt*$abstand,0,$zmaxgitter)*$scaley);
}
for ($schritt=$anfangy; $schritt<=$sendey; $schritt++) {
  openpline($layer,SOLID,ZEICHENDICKE,$farbe,2);
  plinepoint(x3d($xmingitter,$schritt*$abstand,0)*$scalex,
             y3d($xmingitter,$schritt*$abstand,0)*$scaley);
  plinepoint(x3d($xmaxgitter,$schritt*$abstand,0)*$scalex,
             y3d($xmaxgitter,$schritt*$abstand,0)*$scaley);
}
for ($schritt=$anfangx; $schritt<=$sendex; $schritt++) {
  openpline($layer,SOLID,ZEICHENDICKE,$farbe,2);
  plinepoint(x3d($schritt*$abstand,$ymingitter,0)*$scalex,
             y3d($schritt*$abstand,$ymingitter,0)*$scaley);
  plinepoint(x3d($schritt*$abstand,$ymaxgitter,0)*$scalex,
             y3d($schritt*$abstand,$ymaxgitter,0)*$scaley);
}

for ($schritt=$anfangz; $schritt<=$sendez; $schritt++) {
  openpline($layer,SOLID,ZEICHENDICKE,$farbe,2);
  plinepoint(x3d(0,$ymingitter,$schritt*$abstand)*$scalex,
             y3d(0,$ymingitter,$schritt*$abstand)*$scaley);
  plinepoint(x3d(0,$ymaxgitter,$schritt*$abstand)*$scalex,
             y3d(0,$ymaxgitter,$schritt*$abstand)*$scaley);
}
}

#####
##### Ein Koordinatensystem soll gezeichnet werden,
##### dazu werden die Achsen einzeln gezeichnet
#####

```

```

# die y-Achse wird gezeichnet
sub yachse3d {
  my $layer=shift(@_);
  my $ymin=shift(@_);
  my $ymax=shift(@_);
  my $yschritt=shift(@_);
  my $yskala=shift(@_);
  my $ytext=shift(@_);

  # Dicke und Länge des Pfeils wird definiert
  my $dicke=12;
  my $laenge=15;

  opencompound( abrunden($ymin,0.1) , aufrunden(0.075,0.1) ,
                aufrunden($ymax,0.1) , -0.6);
  # der Achsenpfeil wird gezeichnet
  openarrow($layer , SOLID, ZEICHENDICKE+1, 2, 1, 4, $dicke , $laenge , 0 , 2);
  plinepoint($ymin*$scalex , 0);
  plinepoint($ymax*$scalex , 0);

  # Striche müssen innerhalb des Pfeils liegen
  my $yminachse=aufrunden($ymin+0.25*$yschritt , $yschritt);
  my $ymaxachse=abrunden($ymax-0.25*$yschritt , $yschritt);

  # Anfangs und Endwerte für die Schleife werden gesetzt
  my $anfangy=$yminachse/$yschritt;
  my $sendey=$ymaxachse/$yschritt;

  my $schritt;

  if ($yskala eq "keine") { }
  if ($yskala eq "nurstriche") {
    for ($schritt=$anfangy; $schritt<$sendey; $schritt++) {
      ymarkierung3d($layer , $schritt , "" , "ps"); }
  }
  if ($yskala eq "nureins") {
    ymarkierung3d($layer , 1 , 1 , "ps"); }
  if ($yskala eq "normal") {
    for ($schritt=$anfangy; $schritt<=$sendey; $schritt++) {
      if ($schritt != 0) {
        ymarkierung3d($layer , $schritt , $schritt , "ps");
      }
      else {
        ymarkierung3d($layer , $schritt , "" , "ps"); }
    }
  }
}

# Achsenbeschriftung
if ($ytext eq "\$y\$") {
  opentext($layer , SANSSERIF, 12, ALIGNLEFT);
  ptext((($ymax-0.1)*$scalex , 0.2 , $ytext); }
else {
  if ($ytext ne "") {
    opentext($layer , SANSSERIF, 12, ALIGNLEFT);
    ptext($ymax*$scalex , (-0.4)*$scaley , $ytext); }
}
closecompound;

```

```

}

# z-Achse wird gezeichnet
sub zAchse3d {
  my $layer=shift(@_);
  my $zmin=shift(@_);
  my $zmax=shift(@_);
  my $zschritt=shift(@_);
  my $zskala=shift(@_);
  my $ztext=shift(@_);

  # Dicke und Länge des Pfeils wird definiert
  my $dicke=12;
  my $laenge=15;

  opencompound(-0.3, aufrunden($zmax,0.1),
               aufrunden(0.075,0.1), abrunden($zmin,0.1));
  # der Achsenpfeil wird gezeichnet
  openarrow($layer, SOLID, ZEICHENDICKE+1, 2, 1, 4,
            $dicke, $laenge, 0, 2);
  plinepoint(0, $zmin*$scaley);
  plinepoint(0, $zmax*$scaley);

  # Striche müssen innerhalb des Pfeils liegen
  my $zminachse=aufrunden($zmin+0.25*$zschritt,
                          $zschritt);
  my $zmaxachse=abrunden($zmax-0.25*$zschritt,
                          $zschritt);

  # Anfangs und Endwerte für die Schleife werden gesetzt
  my $anfangz=$zminachse/$zschritt;
  my $endez=$zmaxachse/$zschritt;

  my $schritt;

  if ($zskala eq "nurstriche") {
    for ($schritt=$anfangz; $schritt<=$endez; $schritt++) {
      zmarkierung3d($layer, $schritt, "", "ps");
    }
  }

  if ($zskala eq "nureins") {
    zmarkierung3d($layer, 1, 1, "ps");
  }

  if ($zskala eq "normal") {
    for ($schritt=$anfangz; $schritt<=$endez; $schritt++) {
      if ($schritt != 0) {
        zmarkierung3d($layer, $schritt, $schritt, "ps");
      } else {
        zmarkierung3d($layer, $schritt, "", "ps");
      }
    }
  }
}

# Achsenbeschriftung
if ($ztext eq "") { }
if ($ztext eq "\$z\$") {

```

```

        opentext($layer,SANSSERIF,12,ALIGNLEFT);
        ptext(0.2,($zmax-0.4)*$scaley,$ztext);
    } else {
        opentext($layer,SANSSERIF,12,ALIGNLEFT);
        ptext((-0.4)*$scalex,$zmax*$scaley,$ztext);
    }
    closecompound;
}

# y-Achse wird gezeichnet
sub xachse3d {
    my $layer=shift(@_);
    my $zmin=shift(@_);
    my $zmax=shift(@_);
    my $zschrift=shift(@_);
    my $zskala=shift(@_);
    my $ztext=shift(@_);

    # Dicke und Länge des Pfeils wird definiert
    my $dicke=12;
    my $laenge=15;

    opencompound(-0.3,aufunden($zmax,0.1),
                 aufunden(0.075,0.1),abrunden($zmin,0.1));
    # der Achsenpfeil wird gezeichnet
    openarrow($layer,SOLID,ZEICHENDICKE+1,2,1,4,$dicke,$laenge,0,2);
    plinepoint(xz3d(0,$zmin)*$scalex,yz3d(0,$zmin)*$scaley);
    plinepoint(xz3d(0,$zmax)*$scalex,yz3d(0,$zmax)*$scaley);

    # Striche müssen innerhalb des Pfeils liegen
    my $zminachse=aufunden($zmin+0.25*$zschrift,$zschrift);
    my $zmaxachse=abrunden($zmax-0.25*$zschrift,$zschrift);

    # Anfangs und Endwerte für die Schleife werden gesetzt
    my $anfangy=$zminachse/$zschrift;
    my $sendey=$zmaxachse/$zschrift;

    my $schritt;

    if ($zskala eq "nurstriche") {
        for ($schritt=$anfangy;$schritt<=$sendey;$schritt++) {
            xmarkierung3d($layer,$schritt,"","ps");
        }
    }

    if ($zskala eq "nureins") {
        zmarkierung3d($layer,1,1,"ps");
    }

    if ($zskala eq "normal") {
        for ($schritt=$anfangy;$schritt<=$sendey;$schritt++) {
            if ($schritt != 0) {
                xmarkierung3d($layer,$schritt,$schritt,"ps");
            } else {
                xmarkierung3d($layer,$schritt,"","ps");
            }
        }
    }
}

```

```

# Achsenbeschriftung
if ($ztext eq "\$z\$") {
  opentext($layer,SANSSERIF, 12, ALIGNLEFT);
  ptext(xz3d(0.2,$zmax-0.4),yz3d(0,$zmax-0.4),$ztext);
} else {
  if ($ztext ne "") {
    opentext($layer,SANSSERIF, 12, ALIGNLEFT);
    ptext(xz3d(0.2,$zmax-0.4),yz3d(0,$zmax-0.4),$ztext);
  }
}
closecompound;
}

#####
##### Markierungen an den Achsen werden gezeichnet
#####

sub ymarkierung3d {
  my $layer=shift(@_);
  my $xwert=shift(@_);
  my $text=shift(@_);
  my $font=shift(@_);

  openpline($layer,SOLID,ZEICHENDICKE+1,0,2);
  plinepoint($xwert*$scalex,(-0.075)*$scaley);
  plinepoint($xwert*$scalex,0.075*$scaley);
  if ($text ne "") {
    if ($font eq "ps") {
      openpstext($layer,NARROWHELVETICABOLD, 12, ALIGNCENTER);
    }
    else {
      opentext($layer,SANSSERIF, 10, ALIGNCENTER);
    }
    ptext($xwert*$scalex,-0.6,$text);
  }
}

sub zmarkierung3d {
  my $layer=shift(@_);
  my $ywert=shift(@_);
  my $text=shift(@_);
  my $font=shift(@_);

  openpline($layer,SOLID,LINIENDICKE,0,2);
  plinepoint((-0.075)*$scalex,$ywert*$scaley);
  plinepoint(0.075*$scalex,$ywert*$scaley);
  if ($text ne "") {
    if ($font eq "ps") {
      openpstext($layer,NARROWHELVETICABOLD, 12, ALIGNCENTER);
    }
    else {
      opentext($layer,SANSSERIF, 10, ALIGNCENTER);
    }
    ptext(-0.2*$scalex,($ywert-0.1)*$scaley,$text);
  }
}

```

```

sub xmarkierung3d {
  my $layer=shift (@_);
  my $zwert=shift (@_);
  my $text=shift (@_);
  my $font=shift (@_);

  openpline($layer ,SOLID,LINIENDICKE,0 ,2);
  plinepoint(xz3d(0,$zwert)*$scalex ,yz3d(-0.1,$zwert)*$scaley);
  plinepoint(xz3d(0,$zwert)*$scalex ,yz3d(0.1,$zwert)*$scaley);
  if ($text ne "") {
    if ($font eq "ps") {
      openpstext($layer ,NARROWHELVETICABOLD, 12, ALIGNCENTER);
    }
    else {
      opentext($layer ,SANSSERIF, 10, ALIGNCENTER);
    }
    ptext(xz3d(0,$zwert) ,yz3d(0.6,$zwert) ,$text);
  }
}

# Eine (x;y)-Markierung wird gezeichnet
sub xymarkierung {
  my $layer=shift (@_);
  my $func=shift (@_);
  my $xwert=shift (@_);
  my $ywert=shift (@_);
  my $xtext=shift (@_);
  my $ytext=shift (@_);
  my $farbe=shift (@_);
  my $linie=shift (@_);
  my $striche=shift (@_);
  xmarkierung($layer , $xwert , $xtext , "ps") if ($striche =~ /x/);
  ymarkierung($layer , $ywert , $ytext , "ps") if ($striche =~ /y/);

  kreuzchen($layer , $xwert*$scalex , $ywert*$scaley , $farbe);

  if ($linie =~ /x/) {
    openpline($layer ,SOLID,LINIENDICKE, $farbe ,2);
    plinepoint($xwert*$scalex ,0*$scaley);
    plinepoint($xwert*$scalex , $ywert*$scaley);
  }
  if ($linie =~ /y/) {
    openpline($layer ,SOLID,LINIENDICKE, $farbe ,2);
    plinepoint($xwert*$scalex , $ywert*$scaley);
    plinepoint(0*$scalex , $ywert*$scaley);
  }
}

sub xyzmarkierung {
  my $layer=shift (@_);
  my $xwert=shift (@_);
  my $ywert=shift (@_);
  my $zwert=shift (@_);
  my $xtext=shift (@_);
  my $ytext=shift (@_);
  my $ztext=shift (@_);
  my $farbe=shift (@_);

```

```

my $linie=shift(@_);

if ($linie eq "xy") {
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,$ywert,$zwert)*$scalex,
             y3d($xwert,$ywert,$zwert)*$scaley);
  plinepoint(x3d($xwert,$ywert,0)*$scalex,
             y3d($xwert,$ywert,0)*$scaley);
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,$ywert,0)*$scalex,
             y3d($xwert,$ywert,0)*$scaley);
  plinepoint(x3d($xwert,0,0)*$scalex,
             y3d($xwert,0,0)*$scaley);
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,$ywert,0)*$scalex,
             y3d($xwert,$ywert,0)*$scaley);
  plinepoint(x3d(0,$ywert,0)*$scalex,
             y3d(0,$ywert,0)*$scaley);
}

if ($linie eq "xz") {
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,$ywert,$zwert)*$scalex,
             y3d($xwert,$ywert,$zwert)*$scaley);
  plinepoint(x3d($xwert,0,$zwert)*$scalex,
             y3d($xwert,0,$zwert)*$scaley);
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,0,$zwert)*$scalex,
             y3d($xwert,0,$zwert)*$scaley);
  plinepoint(x3d($xwert,0,0)*$scalex,
             y3d($xwert,0,0)*$scaley);
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,0,$zwert)*$scalex,
             y3d($xwert,0,$zwert)*$scaley);
  plinepoint(x3d(0,0,$zwert)*$scalex,
             y3d(0,0,$zwert)*$scaley);
}

if ($linie eq "yz") {
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d($xwert,$ywert,$zwert)*$scalex,
             y3d($xwert,$ywert,$zwert)*$scaley);
  plinepoint(x3d(0,$ywert,$zwert)*$scalex,
             y3d(0,$ywert,$zwert)*$scaley);
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d(0,$ywert,$zwert)*$scalex,
             y3d(0,$ywert,$zwert)*$scaley);
  plinepoint(x3d(0,$ywert,0)*$scalex,
             y3d(0,$ywert,0)*$scaley);
  openpline($layer,DASHED,LINIENDICKE,$farbe,2);
  plinepoint(x3d(0,$ywert,$zwert)*$scalex,
             y3d(0,$ywert,$zwert)*$scaley);
  plinepoint(x3d(0,0,$zwert)*$scalex,
             y3d(0,0,$zwert)*$scaley);
}

if ($linie eq "xyz") {

```

```

    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,$ywert,$zwert)*$scalex,
              y3d($xwert,$ywert,$zwert)*$scaley);
    plinepoint(x3d($xwert,$ywert,0)*$scalex,
              y3d($xwert,$ywert,0)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,$ywert,0)*$scalex,
              y3d($xwert,$ywert,0)*$scaley);
    plinepoint(x3d($xwert,0,0)*$scalex,
              y3d($xwert,0,0)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,$ywert,0)*$scalex,
              y3d($xwert,$ywert,0)*$scaley);
    plinepoint(x3d(0,$ywert,0)*$scalex,
              y3d(0,$ywert,0)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,$ywert,$zwert)*$scalex,
              y3d($xwert,$ywert,$zwert)*$scaley);
    plinepoint(x3d($xwert,0,$zwert)*$scalex,
              y3d($xwert,0,$zwert)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,0,$zwert)*$scalex,
              y3d($xwert,0,$zwert)*$scaley);
    plinepoint(x3d($xwert,0,0)*$scalex,
              y3d($xwert,0,0)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,0,$zwert)*$scalex,
              y3d($xwert,0,$zwert)*$scaley);
    plinepoint(x3d(0,0,$zwert)*$scalex,
              y3d(0,0,$zwert)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d($xwert,$ywert,$zwert)*$scalex,
              y3d($xwert,$ywert,$zwert)*$scaley);
    plinepoint(x3d(0,$ywert,$zwert)*$scalex,
              y3d(0,$ywert,$zwert)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d(0,$ywert,$zwert)*$scalex,
              y3d(0,$ywert,$zwert)*$scaley);
    plinepoint(x3d(0,$ywert,0)*$scalex,
              y3d(0,$ywert,0)*$scaley);
    openpline($layer,DASHED,LINIENDICKE,$farbe,2);
    plinepoint(x3d(0,$ywert,$zwert)*$scalex,
              y3d(0,$ywert,$zwert)*$scaley);
    plinepoint(x3d(0,0,$zwert)*$scalex,
              y3d(0,0,$zwert)*$scaley);
}

circle(50,0,3,x3d($xwert,$ywert,$zwert)*$scalex,
      y3d($xwert,$ywert,$zwert),10,$farbe);
}

sub plotpunkte3d {
  my $layer=shift(@_);
  my $x1=shift(@_);
  my $y1=shift(@_);
  my $z1=shift(@_);

```

```

my $x2=shift (@_);
my $y2=shift (@_);
my $z2=shift (@_);
my $farbe=shift (@_);
my $verbunden=shift (@_);

my $xtext="";

xymarkierung($layer,"0",x3d($x1,$y1,$z1),y3d($x1,$y1,$z1),
             $xtext,$xtext,$farbe,"","");
xymarkierung($layer,"0",x3d($x2,$y2,$z2),y3d($x2,$y2,$z2),+
             $xtext,$xtext,$farbe,"","");
if ($verbunden eq "ja") {
    openpline($layer,SOLID,LINIENDICKE,$farbe,2);
    plinepoint(x3d($x1,$y1,$z1)*$scalex,y3d($x1,$y1,$z1)*$scaley);
    plinepoint(x3d($x2,$y2,$z2)*$scalex,y3d($x2,$y2,$z2)*$scaley);
}
}

#####
##### Vektoren werden gezeichnet
#####

# Vektor wird durch seinen Startpunkt und Endpunkt definiert

sub plotvektor3d {
my $layer=shift (@_);
my $xstart=shift (@_);
my $ystart=shift (@_);
my $zstart=shift (@_);
my $xende=shift (@_);
my $yende=shift (@_);
my $zende=shift (@_);
my $gestrichelt=shift (@_);

my $radius=10;
my $pdicke=10;
my $plaenge=15;

if (($xstart != $xende) || ($ystart != $yende) || ($zstart != $zende)) {
    if ($gestrichelt==0) {
        openarrow($layer,SOLID,LINIENDICKE,2,1,4,$pdicke,$plaenge,$farbe,2)
        ;
        plinepoint(x3d($xstart,$ystart,$zstart)*$scalex,
                  y3d($xstart,$ystart,$zstart)*$scaley);
        plinepoint(x3d($xende,$yende,$zende)*$scalex,
                  y3d($xende,$yende,$zende)*$scaley);
        circle(50,0,3,x3d($xstart,$ystart,$zstart)*$scalex,
              y3d($xstart,$ystart,$zstart),$radius,$farbe);
    } else {
        openpline($layer,$gestrichelt,LINIENDICKE,0,2);
        plinepoint(x3d($xstart,$ystart,$zstart)*$scalex,
                  y3d($xstart,$ystart,$zstart)*$scaley);
        plinepoint(x3d($xende,$yende,$zende)*$scalex,
                  y3d($xende,$yende,$zende)*$scaley);
    }
}
}

```

```

    }
}

sub dreieck3d {
    my $layer=shift(@_);
    my $x1=shift(@_);
    my $y1=shift(@_);
    my $z1=shift(@_);
    my $x2=shift(@_);
    my $y2=shift(@_);
    my $z2=shift(@_);
    my $x3=shift(@_);
    my $y3=shift(@_);
    my $z3=shift(@_);
    my $farbe=shift(@_);
    my $gefüllt=shift(@_);
    my $fuellfarbe=shift(@_);

    openbox($layer,SOLID,LINIENDICKE,$farbe,$gefüllt,$fuellfarbe,4);
    plinepoint(x3d($x1,$y1,$z1)*$scalex,
               y3d($x1,$y1,$z1)*$scaley);
    plinepoint(x3d($x2,$y2,$z2)*$scalex,
               y3d($x2,$y2,$z2)*$scaley);
    plinepoint(x3d($x3,$y3,$z3)*$scalex,
               y3d($x3,$y3,$z3)*$scaley);
    plinepoint(x3d($x1,$y1,$z1)*$scalex,
               y3d($x1,$y1,$z1)*$scaley);

}

#####
#####
##### Einlesen der Befehlsdatei #####
#####

$kommandodateinummer=0;
while (exists $ARGV[$kommandodateinummer]) {

    $kommandodatei=$ARGV[$kommandodateinummer];
    $xfigdatei=$ARGV[$kommandodateinummer];
    $xfigdatei=~s/\.[a-z][a-z][a-z]$//i;
    $xfigdatei=$xfigdatei.".fig";
    $kommandodateinummer++;

    open(KOMMANDO,$kommandodatei) || die $kommandodatei.": $!";

    printf("Erzeuge %s...\n",$xfigdatei);
    openfig($xfigdatei);
    neuefarbe(35,0xff,0xf3,0xb5);
    neugrau(32,200);

    $scalex=1;
    $scaley=1;
    while ($zeile = <KOMMANDO>) {
        # Zeilenenden beseitigen, Kommentare und Leerzeilen ignorieren

```

```

$zeile=~ s/[\r\n]//g;
$zeile=~ s/ +/ /g;
$zeile=~ s/ +$/g;
$zeile=~ s/^ +//g;
next if ($zeile =~ /\^#\//);
next if ($zeile =~ /\^$/);

# Befehl "skalierung"
if ($zeile =~ /\^skalierung( x=([0-9\.]+))?( y=([0-9\.]+))?$\//i) {
    $scalex=$2 if (defined $1);
    $scaley=$4 if (defined $3);
    next;
}
if ($zeile =~ /\^skalierung ([0-9\.]+)/i) {
    $scalex=$scaley=$1;
    next;
}

# Befehl für das Koordinatensystem
if ($zeile =~ /\^zachse3d( layer=([0-9]+))? bereich=([-0-9\.]+),([-0-9\.]+)
,([-0-9\.]+) skala=(keine|nurstriche|nureins|normal) text="\([^"]*)"\$/
i) {
    $layer=20;
    $layer=$2 if (defined $1);
    $zmin3d=$3;
    $zmax3d=$4;
    $zschritt3d=$5;
    $zskala3d=$6;
    $ztext3d=$7;
    zachse3d($layer,$zmin3d,$zmax3d,$zschritt3d,$zskala3d,$ztext3d);
    next;
}

if ($zeile =~ /\^xachse3d( layer=([0-9]+))? bereich=([-0-9\.]+),([-0-9\.]+)
,([-0-9\.]+) skala=(keine|nurstriche|nureins|normal) text="\([^"]*)"\$/
i) {
    $layer=20;
    $layer=$2 if (defined $1);
    $xmin3d=$3;
    $xmax3d=$4;
    $xschritt3d=$5;
    $xskala3d=$6;
    $xtext3d=$7;
    xachse3d($layer,$xmin3d,$xmax3d,$xschritt3d,$xskala3d,$xtext3d);
    next;
}

if ($zeile =~ /\^yachse3d( layer=([0-9]+))? bereich=([-0-9\.]+),([-0-9\.]+)
,([-0-9\.]+) skala=(keine|nurstriche|nureins|normal) text="\([^"]*)"\$/
i) {
    $layer=20;
    $layer=$2 if (defined $1);
    $ymin3d=$3;
    $ymax3d=$4;
    $yschritt3d=$5;
    $yskala3d=$6;
    $ytext3d=$7;
}

```

```

    yachse3d($layer, $ymin3d, $ymax3d,$yschritt3d, $yskala3d, $ytext3d);
    next;
}

# Befehl um 3d-Gitter zu zeichnen
if ($zeile =~/^gitter( layer=[0-9]+)? bereich=[(-0-9\.)+],([0-9\.)+],([0-9\.)+],([0-9\.)+)$/i) {
    $layer=90;
    $layer=$2 if (defined $1);
    $xmax=$3;
    $ymax=$4;
    $zmax=$5;
    $abstand=$6;
    gitter($layer,$xmax,$ymax,$zmax,$abstand);
    next;
}

# MARKIERUNG eines Punktes im Raum

if ($zeile =~/^xyzmarkierung( layer=[0-9]+)?( farbe=[0-9]+)? k
=([0-9\.)+],([0-9\.)+),([0-9\.)+)( xtext="\([^"]*\)"? ( ytext
="\([^"]*\)"? ( ztext="\([^"]*\)"? ( linie=(xy|xz|yz|xyz))?$)/i) {
    $layer=90;
    $layer=$2 if (defined $1);
    $farbe=0;
    $farbe=$4 if (defined $3);
    $xwert=$5;
    $ywert=$6;
    $zwert=$7;
    $xtext="";
    $xtext=$9 if (defined $8);
    $ytext="";
    $ytext=$11 if (defined $10);
    $ztext="";
    $ztext=$13 if (defined $12);
    $linie="";
    $linie=$15 if (defined $14);
    xyzmarkierung($layer,$xwert,$ywert,$zwert,
        $xtext,$ytext,$ztext,$farbe,$linie);
    next;
}

if ($zeile =~/^punktezeichnen3d( layer=[0-9]+)?( farbe=[0-9]+)? k1
=([0-9\.)+],([0-9\.)+),([0-9\.)+ k2=([0-9\.)+],([0-9\.)+],
([0-9\.)+)( verbunden=(ja|nein))?$)/i) {
    $layer=20;
    $layer=$2 if (defined $1);
    $farbe=0;
    $farbe=$4 if (defined $3);
    $x1=$5;
    $y1=$6;
    $z1=$7;
    $x2=$8;
    $y2=$9;
    $z2=$10;
    $verbunden="ja";
    $verbunden=$12 if (defined $11);
}

```

```

    plotpunkte3d($layer,$x1,$y1,$z1,$x2,$y2,$z2,$farbe,$verbunden);
    next;
}

if ($zeile =~/^plotvektor3d( layer=([0-9]+))?( farbe=([0-9]+))? k1
=([-0-9\.]+),([-0-9\.]+),([-0-9\.]+) k2=([-0-9\.]+),([-0-9\.]+)
,([-0-9\.]+)$/i) {
    $layer=80;
    $layer=$2 if (defined $1);
    $farbe=0;
    $farbe=$4 if (defined $3);
    $xstart=$5;
    $ystart=$6;
    $zstart=$7;
    $xende=$8,
    $yende=$9;
    $zende=$10;
    plotvektor3d($layer,$xstart,$ystart,$zstart,
                $xende,$yende,$zende,0);
    next;
}

if ($zeile =~/^dreieck3d( layer=([0-9]+))?( farbe=([0-9]+),([0-9]+)
,([0-9]+))? \((([-0-9\.]+),([-0-9\.]+),([-0-9\.]+)\) \((([-0-9\.]+)
,([-0-9\.]+),([-0-9\.]+)\) \((([-0-9\.]+),([-0-9\.]+),([-0-9\.]+)\)$/i) {
    $layer=80;
    $layer=$2 if (defined $1);
    $farbe=0;
    $farbe=0;
    $gefüllt=35;
    $füllfarbe=27;

    if (defined $3) {
        $farbe=$4;
        $gefüllt=$5;
        $füllfarbe=$6;
    }

    $x1=$7;
    $y1=$8;
    $z1=$9;
    $x2=$10;
    $y2=$11;
    $z2=$12;
    $x3=$13;
    $y3=$14;
    $z3=$15;
    dreieck3d($layer,$x1,$y1,$z1,$x2,$y2,$z2,
              $x3,$y3,$z3,$farbe,$gefüllt,$füllfarbe);
    next;
}

if ($zeile =~/^rechteck3d( layer=([0-9]+))?( farbe=([0-9]+),([0-9]+)
,([0-9]+))? \((([-0-9\.]+),([-0-9\.]+),([-0-9\.]+)\) \((([-0-9\.]+)
,([-0-9\.]+),([-0-9\.]+)\) \((([-0-9\.]+),([-0-9\.]+),([-0-9\.]+)\)
\((([-0-9\.]+),([-0-9\.]+),([-0-9\.]+)\)$/i) {
    $layer=80;

```

```

$layer=$2 if (defined $1);
$farbe=0;
$farbe=0;
$gefüllt=35;
$füllfarbe=27;

if (defined $3) {
    $farbe=$4;
    $gefüllt=$5;
    $füllfarbe=$6;
}

$x1=$7;
$y1=$8;
$z1=$9;
$x2=$10;
$y2=$11;
$z2=$12;
$x3=$13;
$y3=$14;
$z3=$15;
$x4=$16;
$y4=$17;
$z4=$18;

rechteck3d($layer,$x1,$y1,$z1,$x2,$y2,$z2,
           $x3,$y3,$z3,$x4,$y4,$z4,$farbe,$gefüllt,$füllfarbe);
next;
}

# Alle anderen Befehle sind Schrott
printf("Unbekannter Befehl '%s'\n", $zeile);
}

closefig;
close(KOMMANDO);
}

```