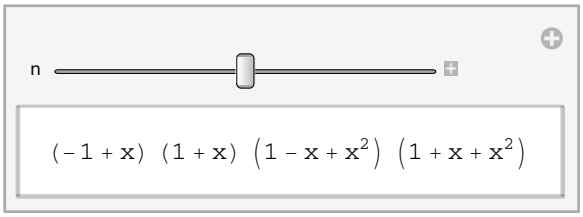
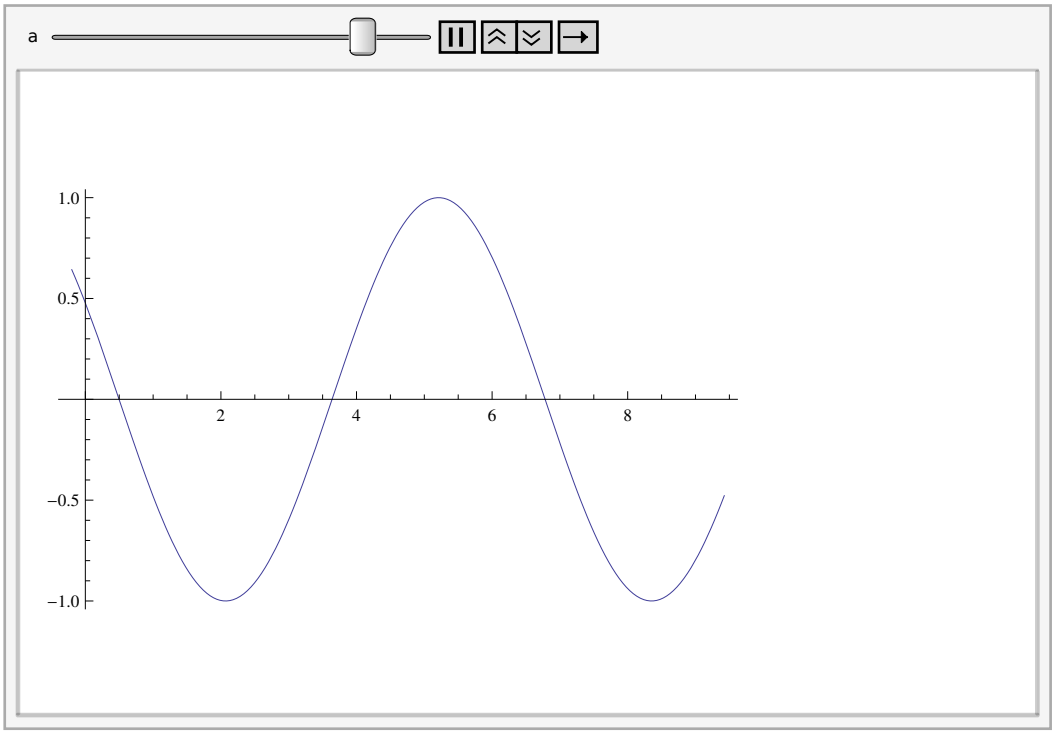
```
(*  Graphics Teil 3:  WQ,  FS 2013 *)
```

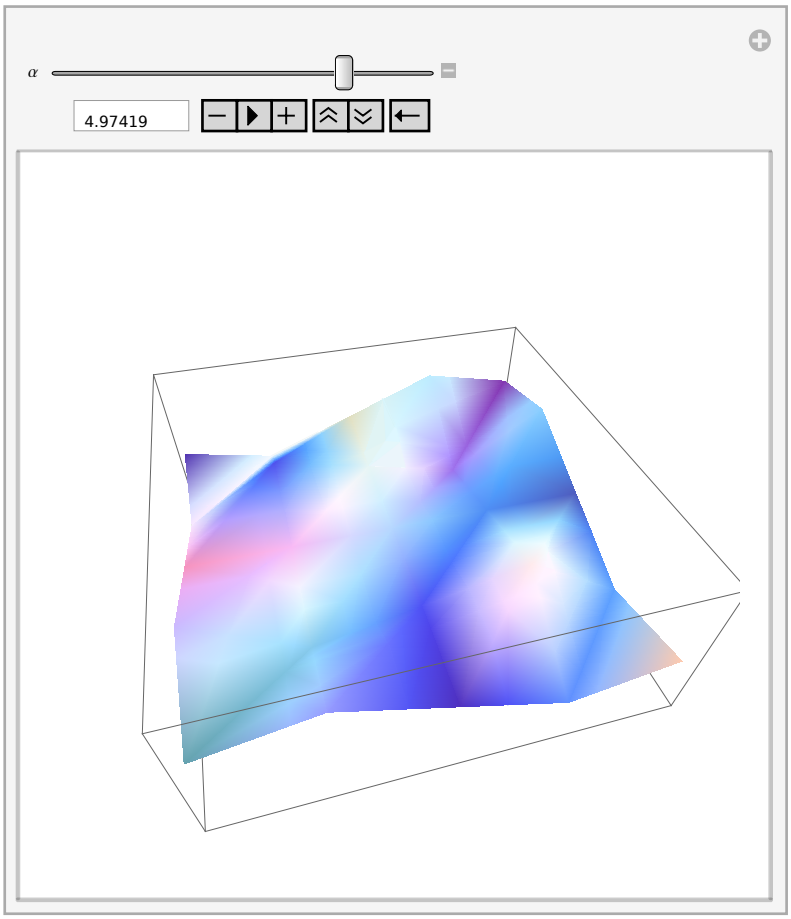
```
Manipulate[Factor[x^n - 1], {n, 2, 10, 1}]
```



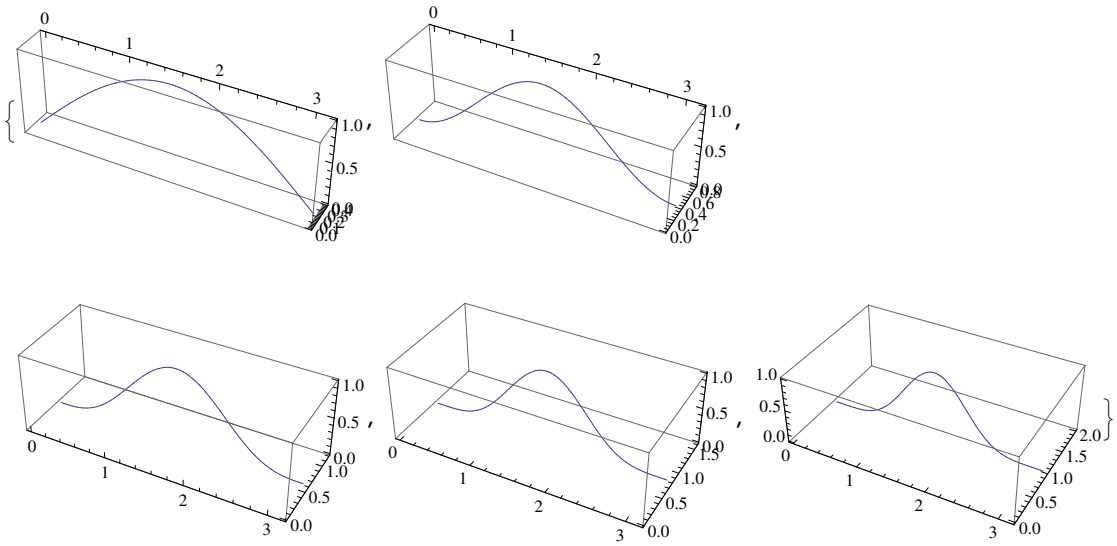

```
Animate[Plot[Sin[x + a], {x, -0.2, 3 Pi}], {a, 0, 2 Pi}]
```



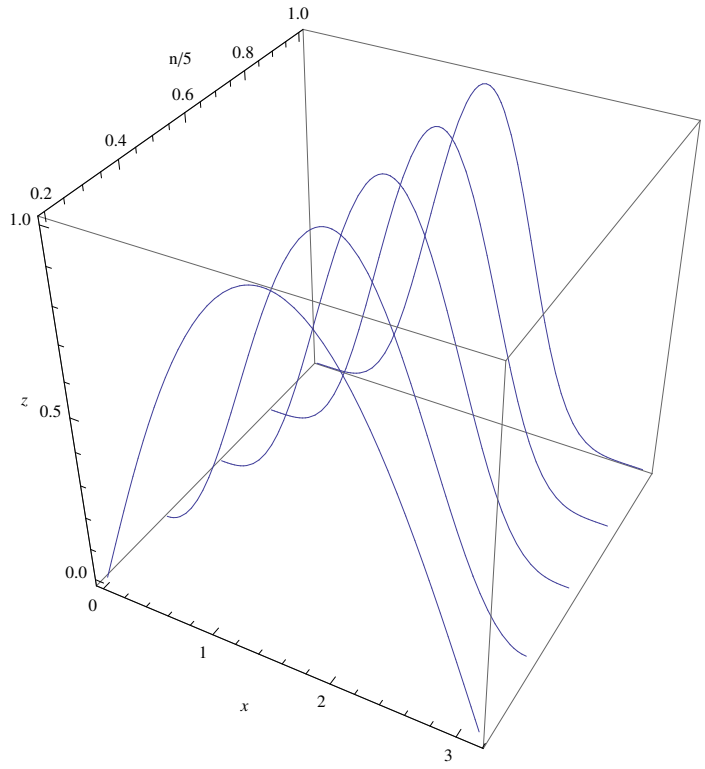
```
Manipulate[Plot3D[Sin[x y], {x, -Pi, Pi},
  {y, -Pi, Pi}, PlotPoints → 5, PlotRange → 2, Axes → None,
  ViewPoint → {Sin[α], Cos[α], 1}, ViewAngle → 20°], {α, 0., 2 Pi, Pi / 12}]
```




```
(* Zeige "gestapelte" 2D-Kurven als "Rad-Staender" *)
pp = Table[ParametricPlot3D[{x, n / 5, Sin[x]^n}, {x, 0, π}], {n, 5}]
```

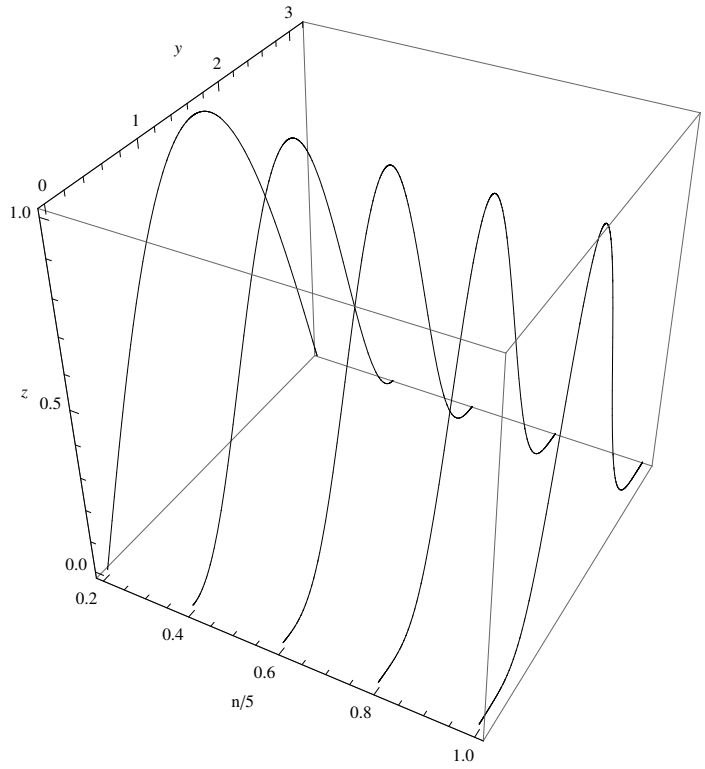


```
Show[pp[[5]], pp[[4]], pp[[3]], pp[[2]],
 pp[[1]] , Axes → True, AxesLabel → {x, "n/5", z},
 BoxRatios → {1, 1, 1}, PlotRange → {{0, Pi}, {0.2, 1}, {0, 1}}]
```




```
(* Hier die Mma - Version aus einer help - Seite *)
```

```
Graphics3D[MapIndexed[
  Cases[#1, Line[L_] :> Line[Thread[{#2[[1]] / 5, L[[All, 1]], L[[All, 2]]}]], -1] &,
  Table[Plot[Sin[x]^n, {x, 0, π}], {n, 5}]],
 Axes → True, AxesLabel → {"n/5", y, z}, BoxRatios → {1, 1, 1}]
```



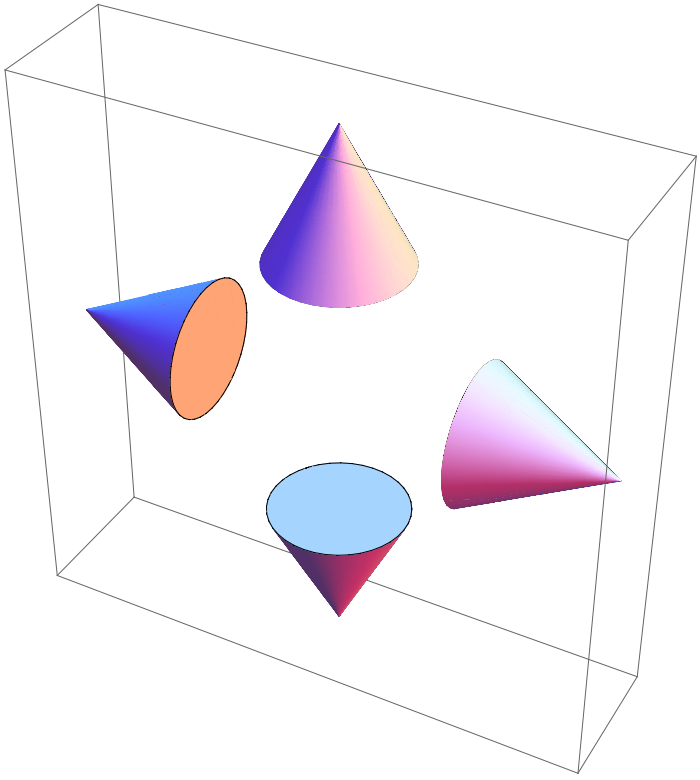

```
MapIndexed[f, {a}, -1]
```

```
{f[a, {1}]}
```

```
MapIndexed[f, {a, b, c}, -1]
```

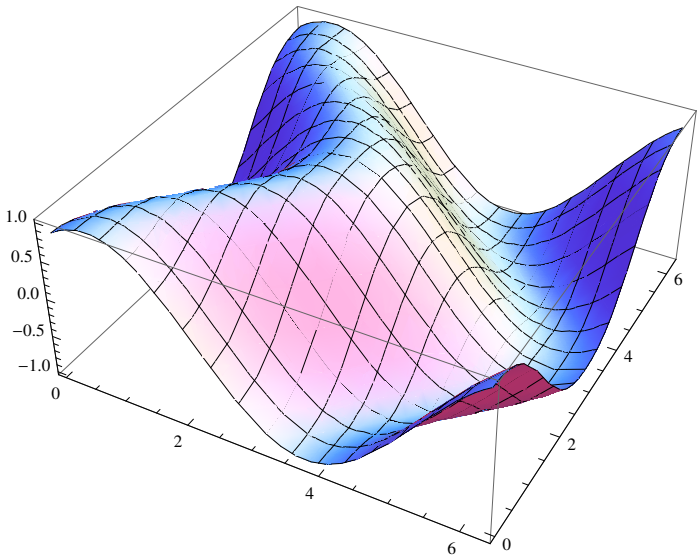
```
{f[a, {1}], f[b, {2}], f[c, {3}]}
```

```
Graphics3D[GeometricTransformation[Cone[{{0, 0, 0}, {0, 0, 1}}, .5],
  {{RotationMatrix[0 Degree, {0, 1, 0}], {0, 0, 1}},
   {RotationMatrix[90 Degree, {0, 1, 0}], {1, 0, 0}},
   {RotationMatrix[180 Degree, {0, 1, 0}], {0, 0, -1}},
   {RotationMatrix[-90 Degree, {0, 1, 0}], {-1, 0, 0}}}]]
```



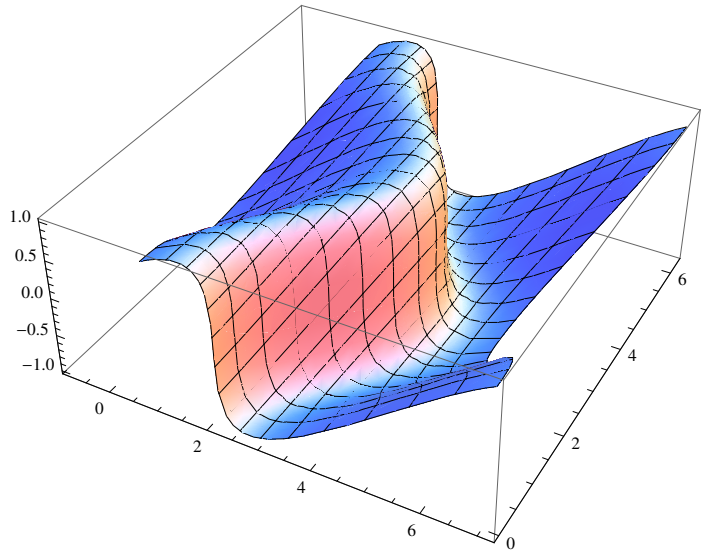
```
(* Dieses Bild soll transformiert werden *)
```

```
Plot3D[Sin[x + Cos[y]], {x, 0, 2 π}, {y, 0, 2 π}, PlotRange → All]
```



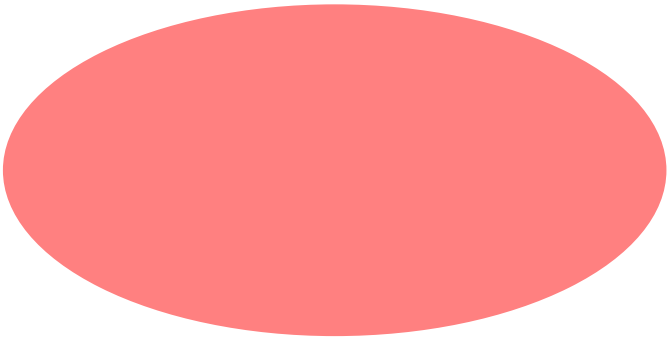
Mathematica kernel function GeometricTransformation

```
Plot3D[Sin[x + Cos[y]], {x, 0, 2 π}, {y, 0, 2 π}, PlotRange → All] /.
 Graphics3D[gr_, opts___] :→
  Graphics3D[GeometricTransformation[gr, {{1, 0, 1}, {0, 1, 0}, {0, 0, 1}}], opts]
```

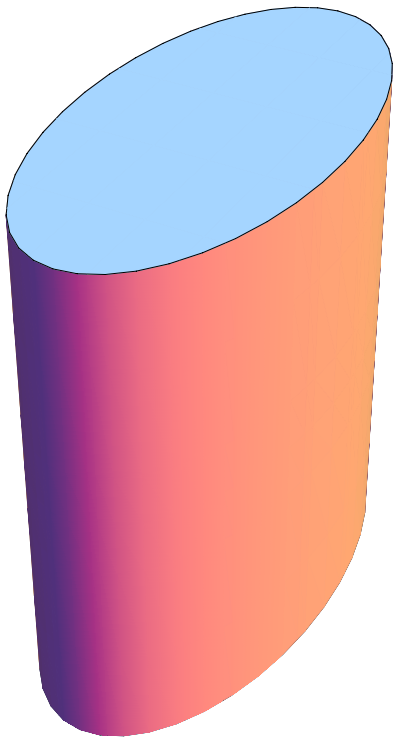



```
The functionality of former Shadow, ShadowPlot3D,
and ListShadowPlot3D is now function Scale
```

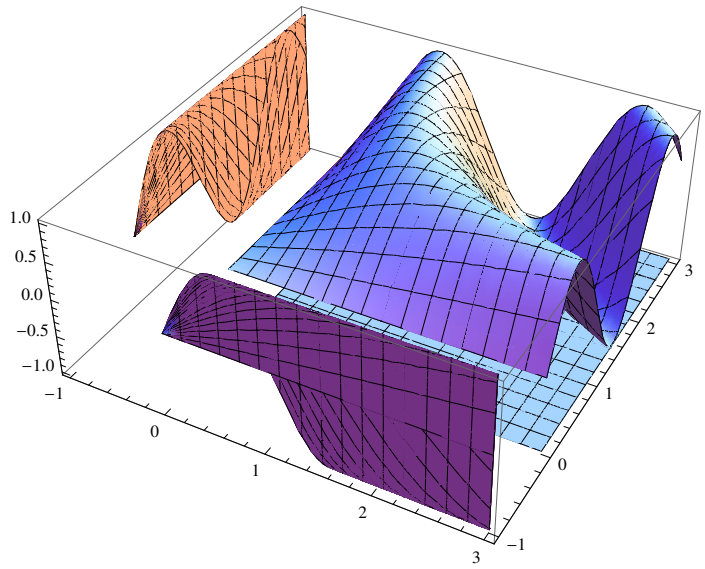
```
Graphics[{Pink, Scale[Disk[], {1, 1 / 2}]}]
```



```
Graphics3D[Scale[Cylinder[], {1 / 2, 1, 1}, {0, 0, 0}], Boxed → False]
```



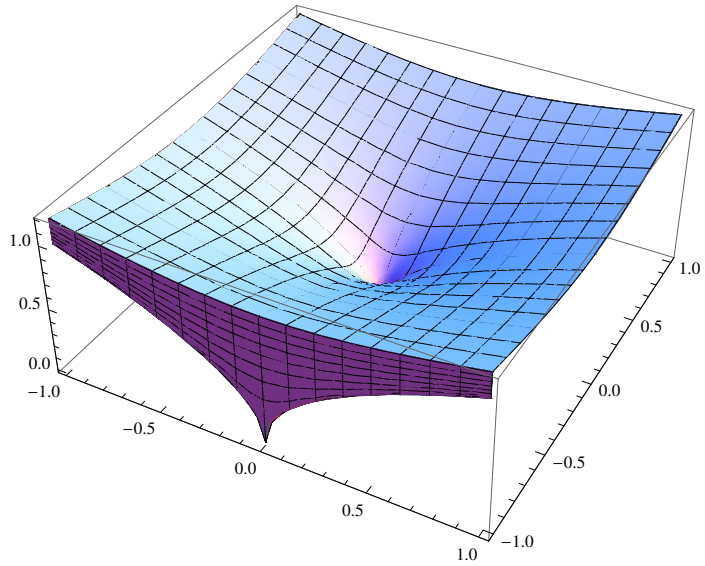
```
Plot3D[Sin[x y], {x, 0, 3}, {y, 0, 3}, PlotRange → All] /.
 Graphics3D[gr_, opts___] :→ Graphics3D[
   {gr, Scale[gr, #, {-1, -1, -1}] & /@ (1 + 10^-3 - IdentityMatrix[3])}, opts]
```




```
1 + 10^-3 - IdentityMatrix[3] // N
```

```
{{0.001, 1.001, 1.001}, {1.001, 0.001, 1.001}, {1.001, 1.001, 0.001}}
```

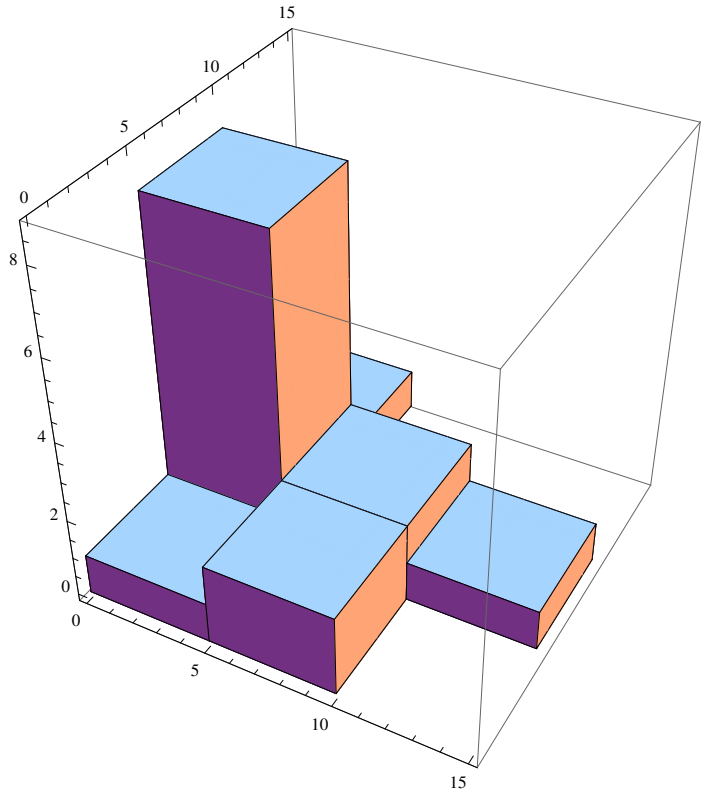
```
Plot3D[Abs[Sqrt[x + I y]], {x, -1, 1}, {y, -1, 1}, PlotRange → All] /.
 Graphics3D[gr_, opts___] :→ Graphics3D[{gr,
    Scale[gr, #, {-1, -1, 0}] & /@ (1 + 10^-3 - DiagonalMatrix[{1, 1, 0}])}, opts]
```




```
1 + 10^-3 - DiagonalMatrix[{1, 1, 0}] // N
```

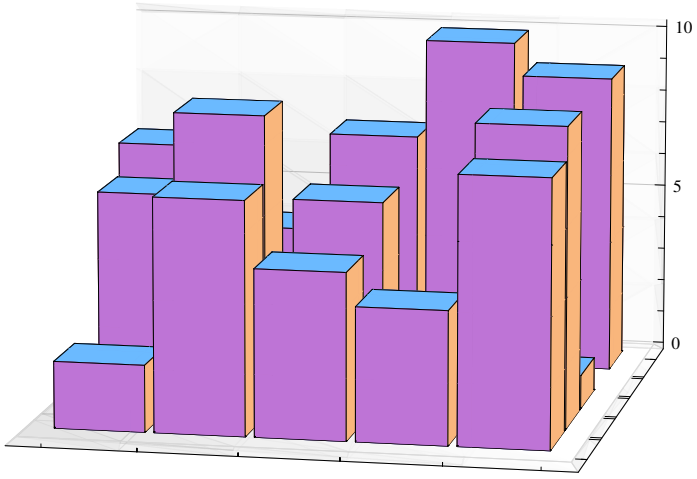
```
{{0.001, 1.001, 1.001}, {1.001, 0.001, 1.001}, {1.001, 1.001, 1.001}}
```

```
Histogram3D[{{7, 0}, {0, 8}, {0, 8}, {3, 7}, {1, 6}, {1, 10}, {4, 9}, {1, 5}, {10, 6},
   {0, 2}, {5, 0}, {7, 6}, {9, 7}, {0, 9}, {3, 9}}, BoxRatios → {1, 1, 1},
 ChartStyle → Directive[EdgeForm[Black], White], Lighting → Automatic]
```




```
(* Histogram zählt die Anzahl der vorkommenden Paare in {x,y}
 BarChart bildet die angegebenen Werte in einer Matrix ab *)
```

```
bild = BarChart3D[{{2, 1, 6, 7, 4}, {7, 9, 1, 0, 4},
    {5, 2, 6, 2, 7}, {4, 3, 0, 0, 10}, {8, 9, 1, 0, 9}} // Transpose,
  BarSpacing → {0.1, 0.1}, ChartElementFunction → "Cube", ChartLayout → "Grid",
  ChartStyle → Directive[EdgeForm[Black], White], Lighting → Automatic]
```



```
Export["histo3D.eps", bild]
```

```
histo3D.eps
```

```
$ExportFormats
```

```
{3DS, ACO, AIFF, AU, AVI, Base64, Binary, Bit, BMP, Byte, BYU, BZIP2, C, CDF,
 Character16, Character8, Complex128, Complex256, Complex64, CSV, CUR, DICOM,
 DIF, DIMACS, DOT, DXF, EMF, EPS, ExpressionML, FASTA, FASTQ, FCS, FITS, FLAC,
 FLV, GIF, Graph6, Graphlet, GraphML, GXL, GZIP, HarwellBoeing, HDF, HDF5, HTML,
 ICNS, ICO, Integer128, Integer16, Integer24, Integer32, Integer64, Integer8,
 JPEG, JPEG2000, JSON, JVX, KML, LEDA, List, LWO, MAT, MathML, Maya, MGF, MIDI,
 MOL, MOL2, MTX, MX, NASACDF, NB, NetCDF, NEXUS, NOFF, OBJ, OFF, Package, Pajek,
 PBM, PCX, PDB, PDF, PGM, PLY, PNG, PNM, POV, PPM, PXR, RawBitmap, Real128,
 Real32, Real64, RIB, RTF, SCT, SDF, SND, Sparse6, STL, String, SurferGrid,
 SVG, SWF, Table, TAR, TerminatedString, TeX, Text, TGA, TGF, TIFF, TSV,
 UnsignedInteger128, UnsignedInteger16, UnsignedInteger24, UnsignedInteger32,
 UnsignedInteger64, UnsignedInteger8, UUE, VideoFrames, VRML, VTK, WAV,
 Wave64, WDX, X3D, XBM, XHTML, XHTMLMathML, XLS, XLSX, XML, XYZ, ZIP, ZPR}
```

```
Export["histo3D.pdf", bild]
```

```
histo3D.pdf
```

```
Import["histo3D.pdf"]
```