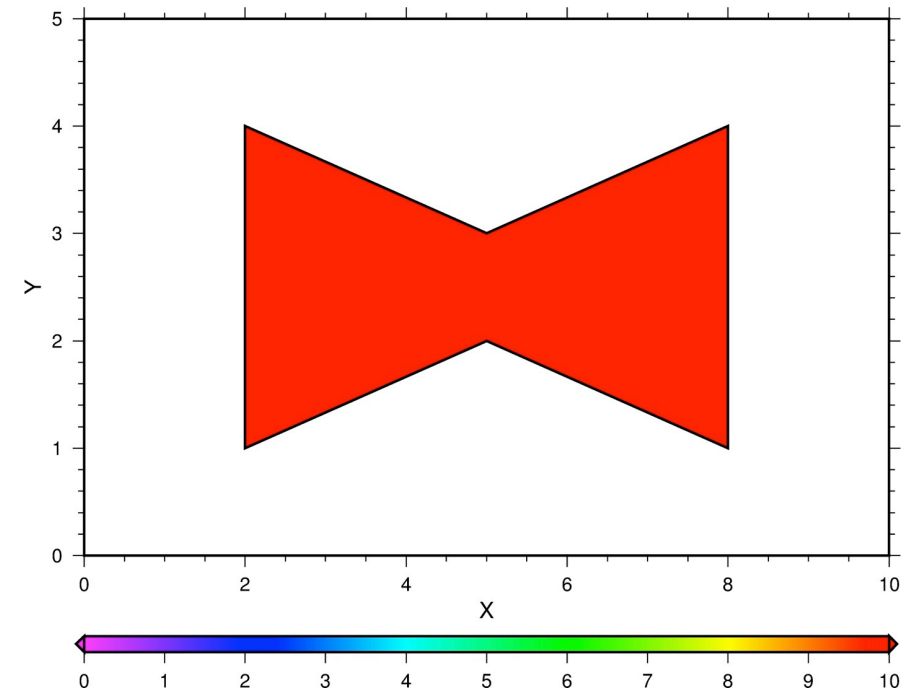


Abschließend noch das -G Flag. Damit lassen sich Polygone jeglicher Form mit der Farbe ausfüllen die danach angegeben wird:

```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 12
gmtset ANNOT_FONT_SIZE 10
gmtset LABEL_OFFSET 0.0
set psfile=line.ps
makecpt -Crainbow -T0/10/1 -Z -D > color.cpt
psbasemap -R0/10/0/5 -JX15/10 -Ba2f0.5:"X":/alf0.2:"Y":WSen -K -P -Y5 > $psfile
cat line.dat | psxy -R -J -O -K -V -W5/0/0/0 -G255/0/0 >> $psfile
psscale -Ccolor.cpt -D7.5/-1.5/15/0.3h -Ba1 -O -E >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Wobei line.dat folgenden Inhalt hat:

```
2 1
5 2
8 1
8 4
5 3
2 4
2 1
```

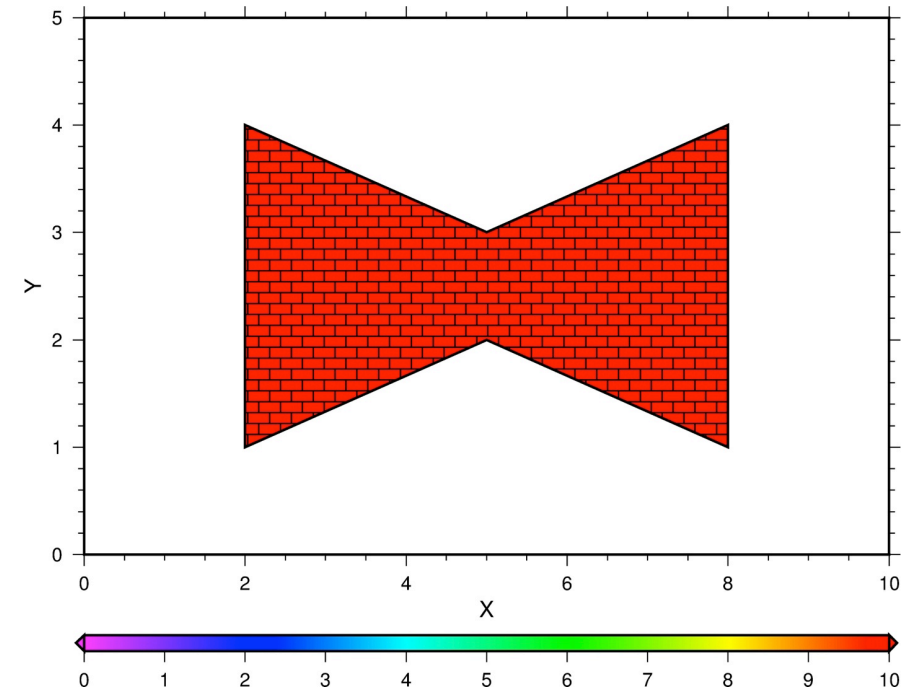


Abschließend noch das -G Flag. Damit lassen sich Polygone jeglicher Form mit der Farbe ausfüllen die danach angegeben wird. Es bietet auch die Möglichkeit als Füllung ein Pattern anzugeben:

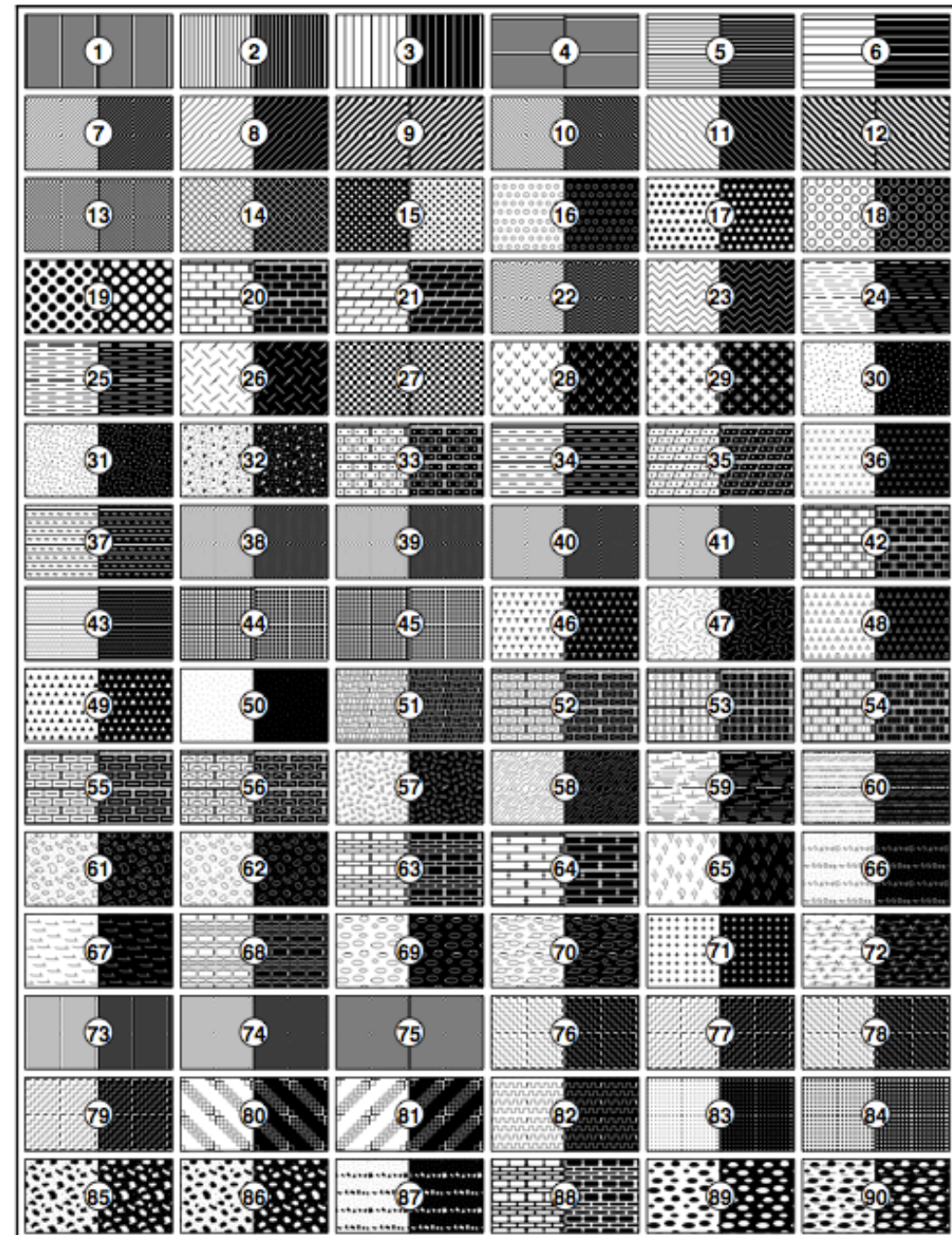
```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 12
gmtset ANNOT_FONT_SIZE 10
gmtset LABEL_OFFSET 0.0
set psfile=line.ps
makecpt -Crainbow -T0/10/1 -Z -D > color.cpt
psbasemap -R0/10/0/5 -JX15/10 -Ba2f0.5:"X":/a1f0.2:"Y":WSen -K -P -Y5 > $psfile
cat line.dat | psxy -R -J -O -K -V -W5/0/0/0 -Gp200/20:F0/0/0B255/0/0 >> $psfile
psscale -Ccolor.cpt -D7.5/-1.5/15/0.3h -Ba1 -O -E >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Wobei line.dat folgenden Inhalt hat:

```
2 1
5 2
8 1
8 4
5 3
2 4
2 1
```



In GMT stehen 90 verschiedene Pattern zur Verfügung.



- Unix – Befehle

- Verzeichniswechsel
- Erstellen, löschen, kopieren oder verschieben von Dateien/Verzeichnissen
- Grobe Bearbeitung von Dateien
- Ändern von Benutzerrechten

- AWK / SED

- Manipulieren von Dateien und Datensätzen

- SHELL – Skripte

- Zusammenfassen verschiedenster Befehle in ein Skript

- FORTRAN 95

- Programme zum Lösen gestellter Probleme

- GMT

- Plotten von Daten, Karten,...
- Manipulieren von Daten (z.B. interpolieren)

- LATEX

- Zusammenfassen von Ergebnissen in Textform (z.B. Bericht)
- Erstellen von Tabellen, Formeln
- Einbinden von Grafiken

Warum gerade diese Programme??

Alle sind kostenlos und universell einsetzbar !!!!

Zum Plotten von Seismogrammen bietet GMT zwei Programme: PSSEGY und PSWIGGLE. Mit PSSEGY kann direkt ein SEG Y-file geplottet wogegen PSWIGGLE einen ASCII-Input erwartet. Das Dateiformat SEG Y ist ein Standard-Dateiformat für seismische Daten und ist größtenteils Binär. Eine SEG Y-Datei besteht aus einem 3600 Byte Header (3200 Byte ASCII und 400 Byte binär) an dem sich dann die Spuren (Zeitreihen) anschließen, wobei jede Spur auch einen 240 Byte Header hat. In den binären Headern steht an fest vorgeschriebenen Positionen z.B. die Anzahl an Sample, Sampleintervall, Geophonposition, Schussposition,...

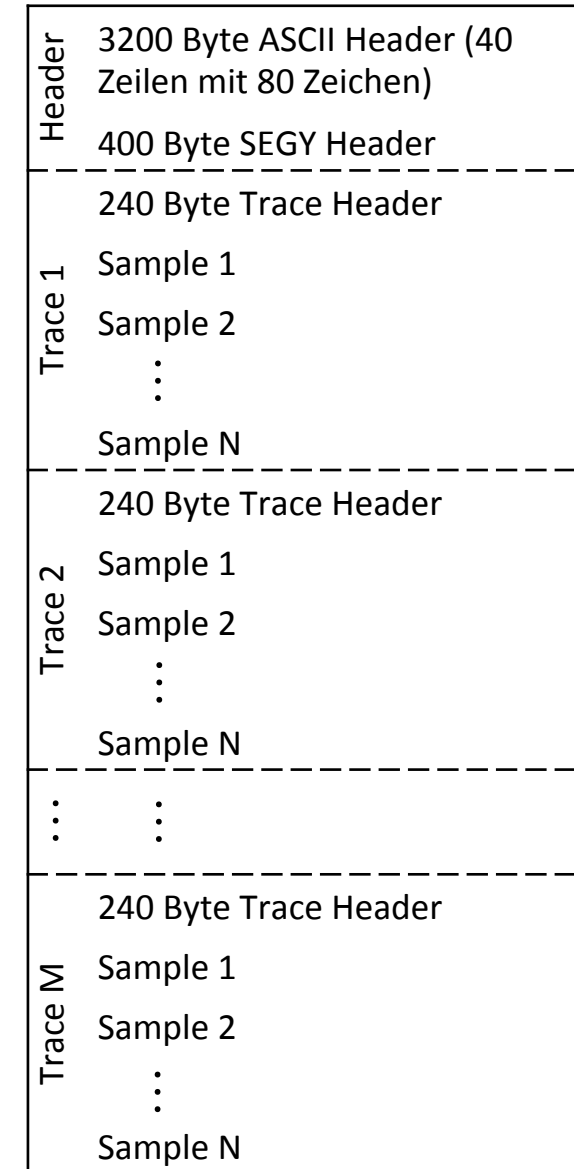
Erste Teil des 400 Byte SEG Y-Header:

```

001 - 004      Job identification number.
005 - 008 *    Line number.
009 - 012 *    Reel number.
013 - 014 *    Number of data traces per record.
015 - 016 *    Number of auxiliary traces per record.
017 - 018 *    Sample interval of this reel's data in microseconds.
019 - 020      Sample interval of original field recording in microseconds.
021 - 022 *    Number of samples per trace for this reel's data.
023 - 024      Number of samples per trace in original field recording.
025 - 026 *    Data sample format code:
                  1 = 32-bit IBM floating point
                  2 = 32-bit fixed-point (integer)
                  3 = 16-bit fixed-point (integer)
                  4 = 32-bit fixed-point with gain code (integer)
027 - 028 *    CDP fold (expected number of data traces per ensemble).
029 - 030      Trace sorting code:
                  1 = as recorded
                  2 = CDP ensemble
                  3 = single fold continuous profile
                  4 = horizontally stacked

```

⋮



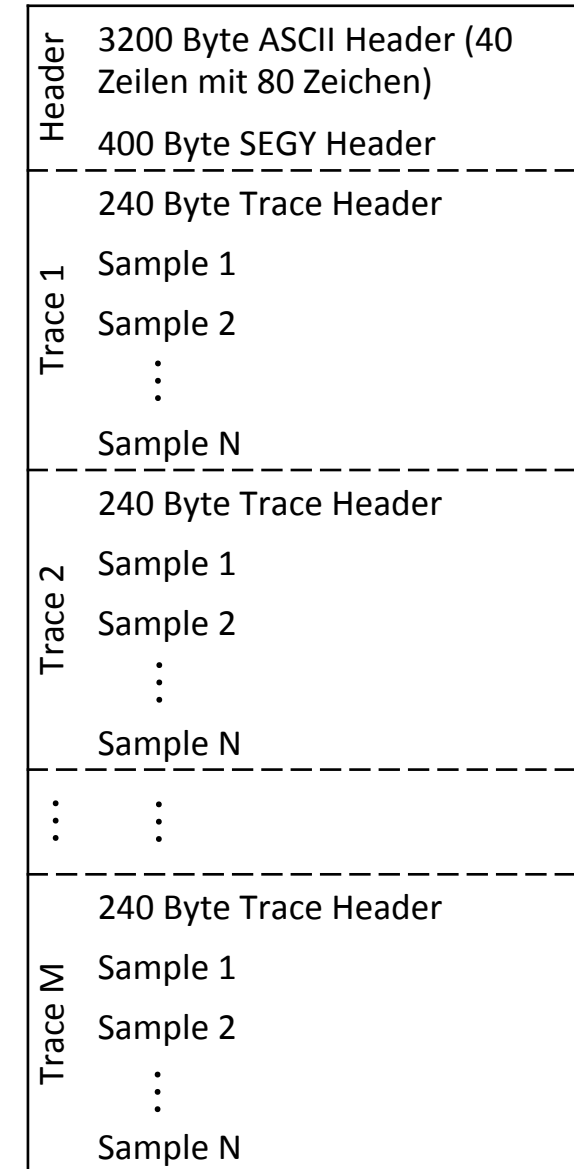
Zum Plotten von Seismogrammen bietet GMT zwei Programme: PSSEGY und PSWIGGLE. Mit PSSEGY kann direkt ein SEG-Y-File geplottet wogegen PSWIGGLE einen ASCII-Input erwartet. Das Dateiformat SEG-Y ist ein Standard-Dateiformat für seismische Daten und ist größtenteils Binär. Eine SEG-Y-Datei besteht aus einem 3600 Byte Header (3200 Byte ASCII und 400 Byte binär) an dem sich dann die Spuren (Zeitreihen) anschließen, wobei jede Spur auch einen 240 Byte Header hat. In den binären Headern steht an fest vorgeschriebenen Positionen z.B. die Anzahl an Sample, Sampleintervall, Geophonposition, Schussposition,...

Erste Teil des 240 Byte Trace-Header:

```

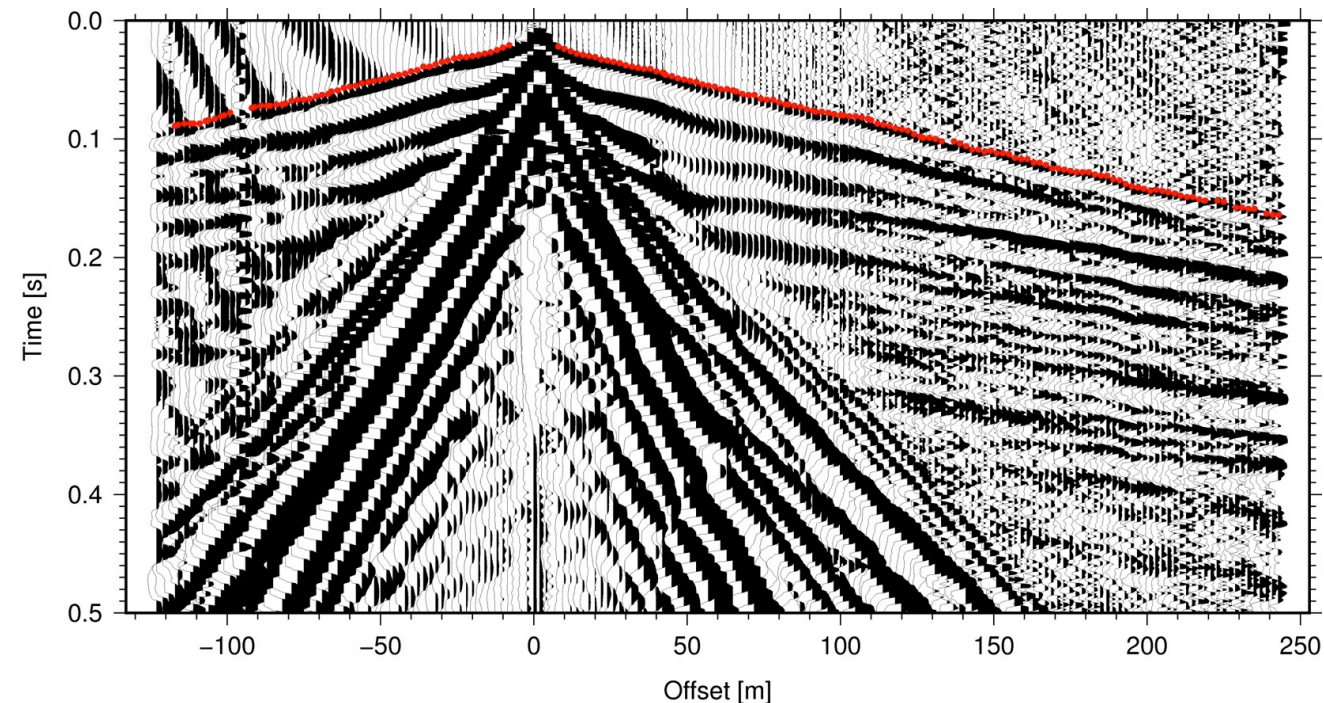
001 - 004 * Trace sequence number within line.
005 - 008   Trace sequence number within reel.
009 - 012 * Original field record number.
013 - 016 * Trace sequence number within original field record.
017 - 020   Energy source point number.
021 - 024   CDP ensemble number.
025 - 028   Trace sequence number within CDP ensemble.
      ⋮
041 - 044   Receiver group elevation.
045 - 048   Surface elevation at source.
      ⋮
069 - 070   Scalar for elevations and depths (+ = multiplier, - = divisor).
071 - 072   Scalar for coordinates (+ = multiplier, - = divisor).
073 - 076   X source coordinate.
077 - 080   Y source coordinate.
081 - 084   X receiver group coordinate.
085 - 088   Y receiver group coordinate.
      ⋮
115 - 116 * Number of samples in this trace.
117 - 118 * Sample interval of this trace in microseconds.
      ⋮

```



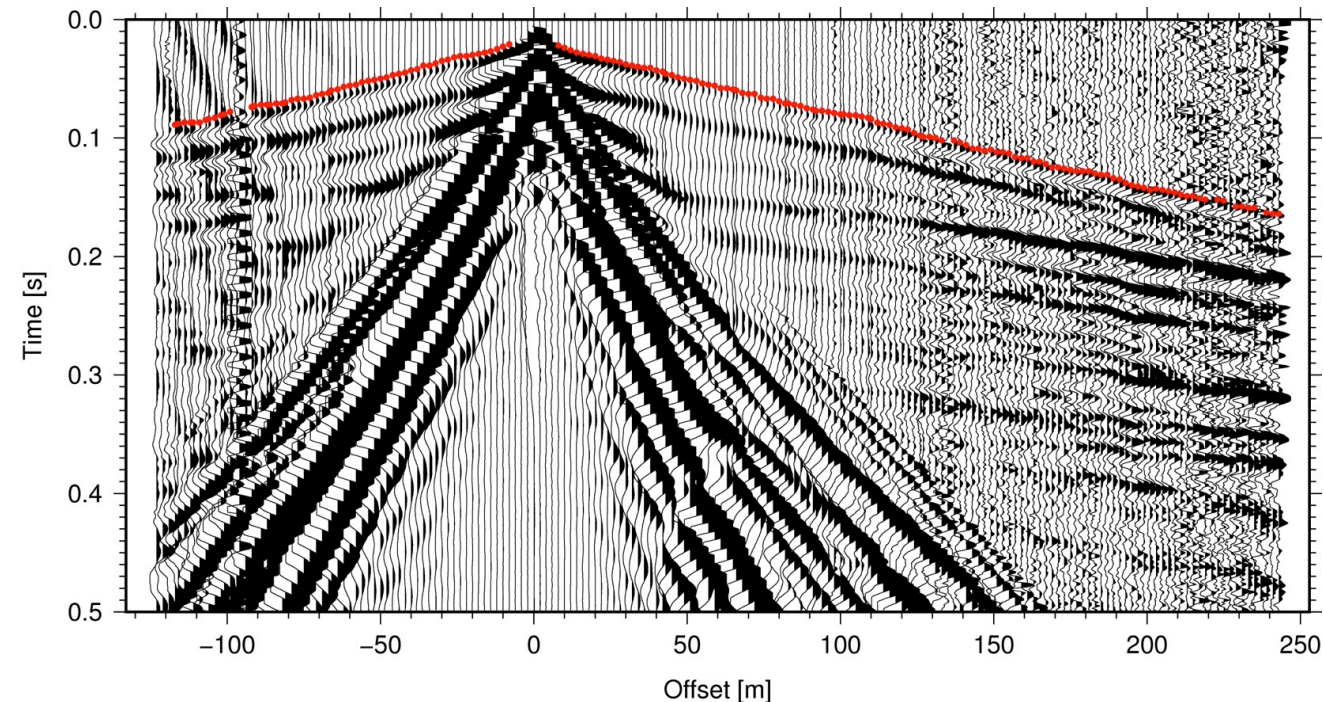
```
#!/bin/tcsh
set shot=10
set clip=2
segypread tape=kopanina.segy endian=0 | segyclean | suwind max=0 accept=$shot key=ep | sugain scale=-1.> tmp.su
segyhdrs < tmp.su
segypwrite tape=tmp.segy < tmp.su
set offmin=`surange < tmp.su | sed -n '/offset/p' | awk '{print $2-10}'`
set offmax=`surange < tmp.su | sed -n '/offset/p' | awk '{print $3+10}'`
set nt=`surange < tmp.su | sed -n '/tracl/p' | awk '{print $3-$2+1}'`
pssegy tmp.segy -D0.005 -C500 -JX20/-10 -R$offmin/$offmax/0/0.5 -So -K -F0/0/0 -M$nt -W -N > shot_var1.ps
cat sht_withdummies_sec.hdr | awk '{if($1=="'$shot'" && $4==1 && $9>0) print $6,$9}' | psxy -R -J -O -K -V -
Sc0.1 -W0.00001/255/0/0 -G255/0/0 >> shot_var1.ps
psbasemap -R -J -Ba50f10:"Offset [m]":/a0.1f0.01:"Time [s]":WSen -O >> shot_var1.ps
gv shot_var1.ps
```

Das Skript plottet alle Spuren, deren Offset (-So) zwischen offmin und offmax liegt in einem Zeitbereich zwischen 0 und 1 Sekunde. Ohne die -S Optionen würde anstatt des Offsets die Spurnummer das Auswahlkriterium sein. Mit -F wird angegeben, dass alle positiven Amplituden gefüllt werden sollen (Optional mit Farbe). Wenn kein -F gesetzt ist muss stattdessen -W gesetzt werden (Wiggle Plot). Die Linienstärke der Wiggles ist nicht einstellbar! Mit -D wird die Skalierung angegeben (keine Option!) und mit -C ab welcher Amplitude abgeschnitten werden soll. Mit psbasemap wird anschließend wieder ein Rahmen um die Seismogramme geplottet. Da mit pssegy keine Schüsse ausgewählt werden können, muss der für den Schuss entsprechende Bereich vorher z.B. mit Seismic Unix (SU) ausgeschnitten werden.



```
#!/bin/tcsh
set shot=10
set clip=2
segypread tape=kopanina.segy endian=0 | segyclean | suwind max=0 accept=$shot key=ep | \
sugain pbal=1 sugain scale=-1. |suascii bare=5 key=offset > tmp.xyz
set offmin=` sort -n -k1 tmp.xyz | head -1 | awk '{print $1-10}'`
set offmax=` sort -n -r -k1 tmp.xyz | head -1 | awk '{print $1+10}'`
cat tmp.xyz | awk '{if($3<='$clip' && $3>=-'$clip'){print $0}else{if($3<0){print $1,$2,-'$clip'}else{print
$1,$2,'$clip'}}}' | awk '{if($1!=old)print ">";print $0;old=$1}' | \
pswiggle -R$offmin/$offmax/0/0.5 -JX20/-10 -Ba50f10:"Offset [m]":/a0.1f0.01:"Time [s]":WSen -W2/0/0/0 \
-G0/0/0 -m">" -K > shot_var2.ps
cat sht_withdummies_sec.hdr | awk '{if($1=="'$shot'" && $4==1 && $9>0) print $6,$9}' \
| psxy -R -J -O -V -Sc0.1 -W0.00001/255/0/0 -G255/0/0 >> shot_var2.ps
gv shot_var2.ps &
```

Die zweite Möglichkeit seismische Daten zu Plotten ist PSWIGGLE. Vorteil von PSWIGGLE zu PSSEGY ist, dass für Spuren die Füllung und die Farben separat angegeben werden können und auch die Strichdicke der Wiggles angegeben werden kann. Nachteil hat PSWIGGLE nur einen: es ist kein Flag zum Abschneiden implementiert und muss daher vorher händisch gemacht werden (z.B.) mit AWK.



Bisher wurden "nur" Vektorgrafiken erzeugt. Auf den nächsten Folien soll gezeigt werden, wie mit GMT Rastergrafiken erzeugt werden.

Bevor ein Grid geplottet werden kann, muss dieses zuerst mit XYZ2GRD erstellt werden. Dieses Kommando wandelt z.B. ASCII-Daten in GMT-Grid-Dateien um.

Dabei gilt folgende Syntax:

```
xyz2grd datei.ascii -Rxmin/xmax/ymin/ymax -Idx/dy -Gdatei.grd
```

oder

```
cat datei.ascii | xyz2grd -Rxmin/xmax/ymin/ymax -Idx/dy -Gdatei.grd
```

wobei `datei.ascii` aus 3 Spalten bestehen muss (x,y,z)

Es besteht auch die Möglichkeit, Dateien mit nur einer Spalte einzulesen (-Z). Diese können als ASCII vorliegen oder auch binär sein. Obacht ist nur geboten bei der folgenden Angabe, mit welchem System das Grid mit den z-Werten belegt werden soll. Diese Problematik wird ausgiebig an einem Beispiel in der Vorlesung erläutert.

Das Kommando für die inverse Operation lautet GRD2XYZ. Die Syntax ist ähnlich der von XYZ2GRD

```
grd2xyz datei.grd -Rxmin/xmax/ymin/ymax > datei.ascii
```

Das `-I` Flag steht hier nicht zur Verfügung. Das ist aber auch nicht notwendig, weil das Intervall des Grids bekannt ist.

Jedes GMT-Grid hat einen binären Header den man sich mit `GRDINFO datei.grd` auf das Terminal ausgeben lassen kann:

```
sinc.grd: Title: datei.grd
sinc.grd: Command: xyz2grd -R-22/22/-22/22 -I0.05 -Gdatei.grd
sinc.grd: Remark:
sinc.grd: Gridline node registration used
sinc.grd: Grid file format: nf (# 18) GMT netCDF format (float) (COARDS-compliant) [DEFAULT]
sinc.grd: x_min: -22 x_max: 22 x_inc: 0.05 name: x nx: 881
sinc.grd: y_min: -22 y_max: 22 y_inc: 0.05 name: y ny: 881
sinc.grd: z_min: -0.217234000564 z_max: 0.999583005905 name: z
sinc.grd: scale_factor: 1 add_offset: 0
```

Dies ist ganz nützlich, wenn man z.B. für ein Skript die Modell-Dimensionen benötigt:

```
set xmin=`grdinfo datei.grd | awk '{if($2=="x_min:")print $3}'`
set xmax=`grdinfo datei.grd | awk '{if($4=="x_max:")print $5}'`
set ymin=`grdinfo datei.grd | awk '{if($2=="y_min:")print $3}'`
set ymax=`grdinfo datei.grd | awk '{if($4=="y_max:")print $5}'`
set r=$xmin/$xmax/$ymin/$ymax
```

```
#!/bin/tcsh
```

```
set xmin=-22
set xmax=22
set ymin=-22
set ymax=22
set dx=0.05
```

```
set r=$xmin/$xmax/$ymin/$ymax
set j=X15/15
```

```
set r2=$xmin/$xmax/-0.3/1.1
set j2=X15/5
```

```
makecpt -Cseis -T-0.2/1/0.1 -Z -D -I > amp.cpt
```

```
makecpt -Cseis -T-1/5/1 -Z> cont.cpt
```

```
echo $xmin $xmax $ymin $ymax $dx | \
```

```
awk '{nx=($2-$1)/$5;ny=($4-$3)/$5;for(i=-nx/2;i<=nx/2;i++){for(j=-ny/2;j<=ny/2;j++)\
```

```
{r=sqrt((i*$5)**2+(j*$5)**2);if(r!=0)print i*$5,j*$5,sin(r)/(r)}}}' | xyz2grd -Gsinc.grd -R$r -I$dx
```

```
echo $xmin $xmax $dx | awk '{nx=($2-$1)/$3;for(i=-nx/2;i<=nx/2;i++){print i*$3,0}}' | \
```

```
grdtrack -R$r -Gsinc.grd | awk '{print $1,$3}' > sinc.xyz
```

```
grdimage sinc.grd -Camp.cpt -Ba5f1WSen -R$r -J$j -K -P > sinc.ps
```

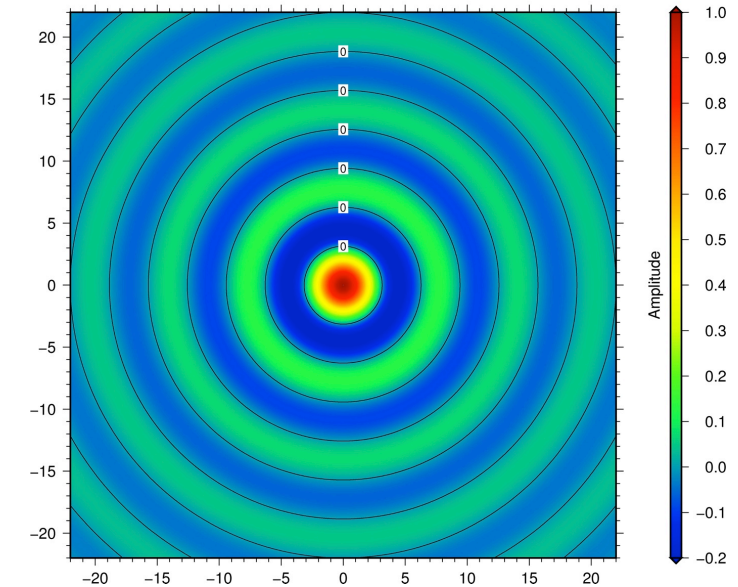
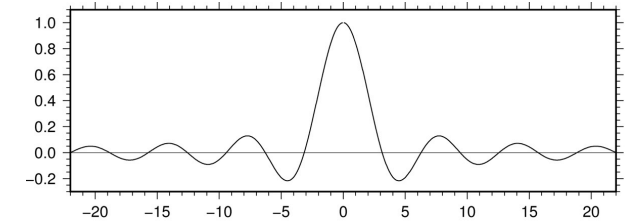
```
psscale -Camp.cpt -D16.5/7.5/15/0.3 -Ba0.1:"Amplitude": -A1 --LABEL_OFFSET=0.3 -O -K -E >> sinc.ps
```

```
grdcontour sinc.grd -J$j -R$r -Ccont.cpt -W3/0/0/0 -O -V -G10/20/0/0 -A+g255/255/255 -K >> sinc.ps
```

```
cat sinc.xyz | psxy -R$r2 -J$j2 -Ba5f1/a0.2f0.05g10000WSen -O -V -W5/0/0/0 -Y17>> sinc.ps
```

```
gv sinc.ps &
```

Zum Darstellen des Grids wird GRDIMAGE benötigt. Vor dem GRDIMAGE-Kommando muss auf jeden Fall eine Farbtabelle erzeugt worden sein (z.B. mit makecpt). Nach GRDIMAGE können z.B. Konturlinien mit GRDCONTOUR geplottet werden. Die z-Werte eines Grids entlang einer beliebigen Linie können mit GRDTRACK ausgegeben werden.



Mit GRDFILTER können Grids gefiltert werden. Die Länge des Filters wird mit $-Fl\ddot{a}nge$ angegeben. Vor der Länge kann auch noch der Filtertyp angegeben werden:

b Boxcar (Alle Wichtungen gleich)

c Cosine (Alle Wichtungen in Form eines Kosinus)

g Gaussian (Wichtungen gegeben durch Gauss-Funktion, wobei die Breite das 6fache der Standardabweichung ist)

m Median

p Maximum likelihood probability

l Lower (kleinsten Wert)

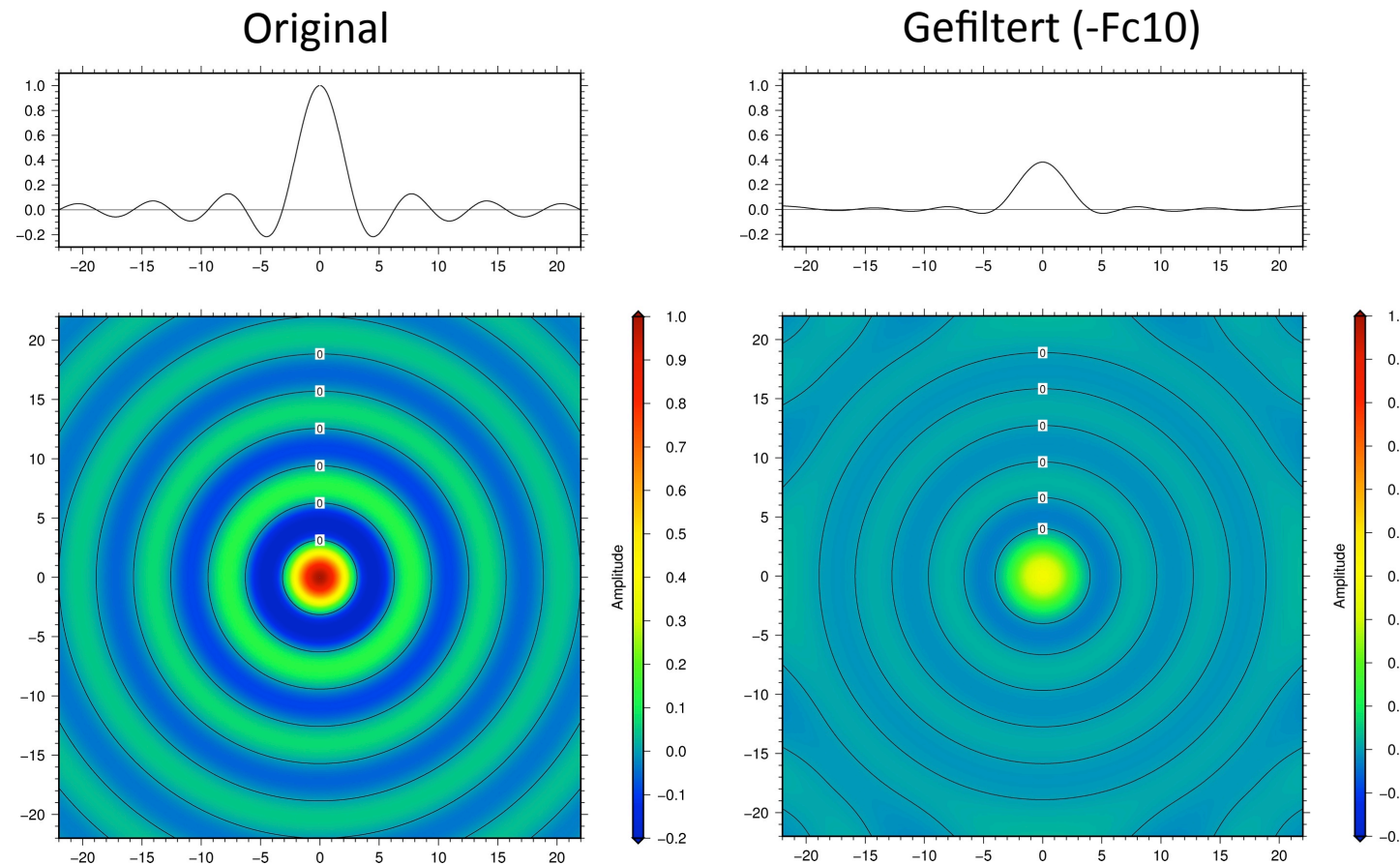
L Lower (kleinsten Wert aller positiven Werte)

u Upper (größten Wert)

U Upper (größten Wert aller negativen Werte)

Mit $-D$ muss noch angegeben werden, auf welches Koordinatensystem sich die Filterlänge bezieht (0 steht für kartesische Koordinaten).

```
grdfilter sinc.grd -Gdum.grd -Fc10 -D0
\mv dum.grd sinc.grd
```



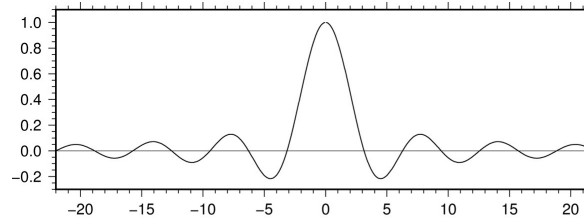
Mit GRDMATH können mathematische Operationen auf Grids durchgeführt werden. So können z.B. Grids miteinander multipliziert werden (keine Matrixmultiplikation!!!). Obacht gilt der Syntax:

```
grdmath Operand1 Operand2 Operator = neu.grd
```

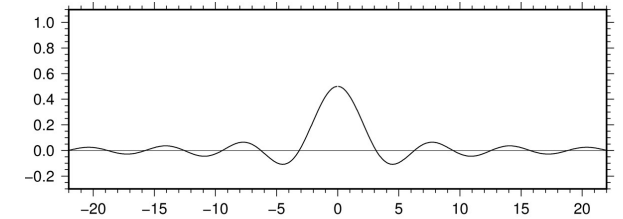
Es können auch mehrere Operationen aneinander gehängt werden:

```
grdmath Operand1 Operand2 Operator Operand3 Operator = neu.grd
```

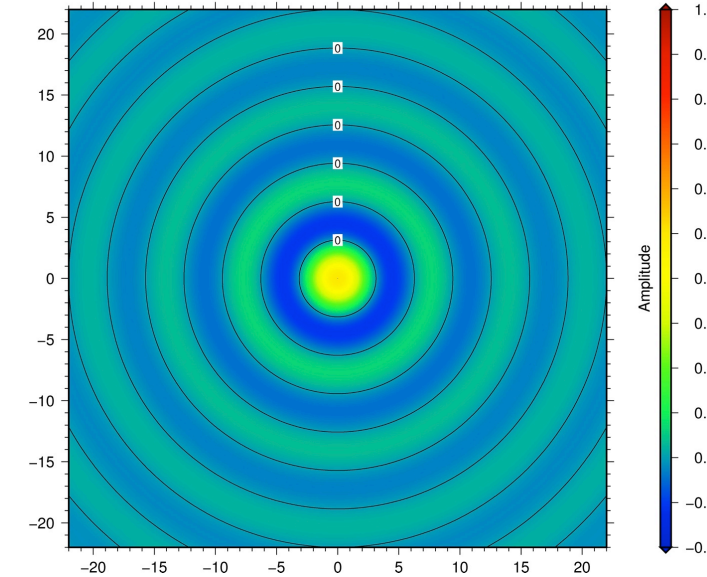
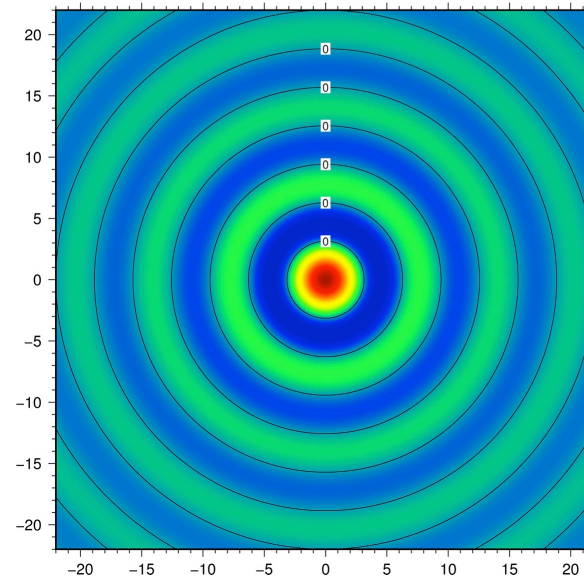
Original



Mit 2 dividiert



```
grdmath sinc.grd 2 DIV = dum.grd
\mv dum.grd sinc.grd
```



Beliebige Polygone können mittels GRDMASK in Grids umgewandelt werden:

```
cat polygon.xy | grdmask -R$r -I$dx -Gmask.grd
```

Mit *-Noutside/edge/inside* können den Bereichen ausserhalb, auf dem Rand und innerhalb des Polygons bestimmte werte zugeordnet werden.

Dazu ein Beispiel an der sinc-Funktion.

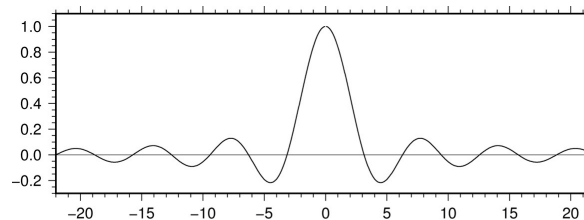
Alle Bereiche mit einem Radius größer 5π sollen ausgeblendet werden:

```
echo 5 1 | awk '{\
pi=atan2(0,-1);n=int(360/$2)+1;\
for(i=0;i<n;i++){\  
print $1*pi*cos(i*$2/180*pi),\  
$1*pi*sin(i*$2/180*pi)}}'\
| grdmask -R$r -I$dx -NNaN/1/1 -Gmask.grd
grdmath sinc.grd mask.grd MUL = dum.grd
```

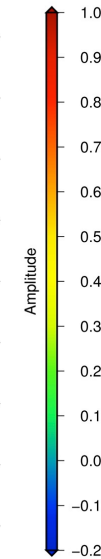
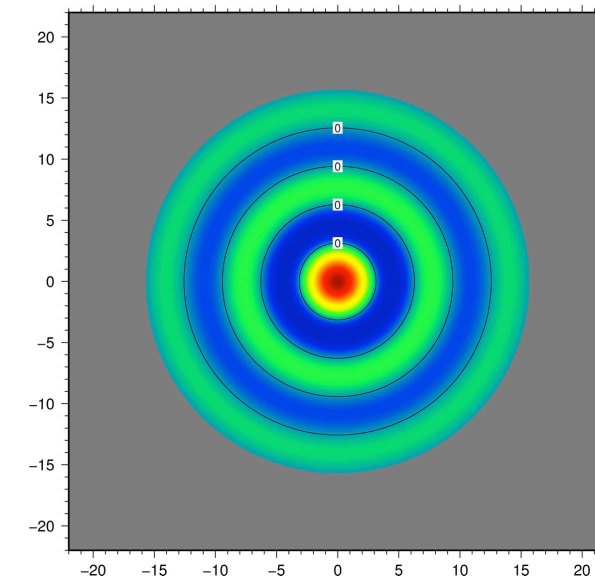
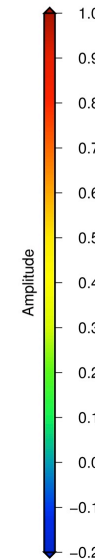
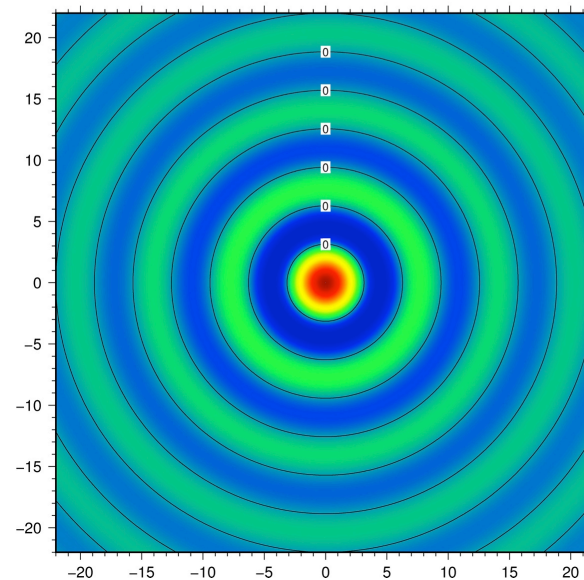
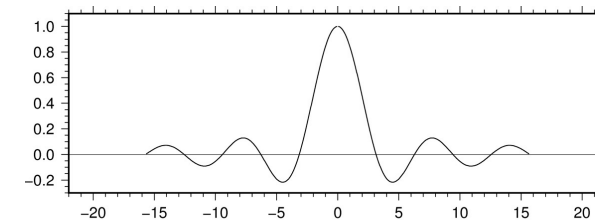
```
\mv dum.grd sinc.grd
```

Es besteht auch hier die Möglichkeit, mehrere Polygone getrennt durch ein bestimmtes Zeichen in GRDMASK einzulesen. Dafür muss bei GRDMASK wieder das *-m* Flag gesetzt werden (und ggf. das Trennzeichen in Gänsefüßchen angehängt werden).

Original



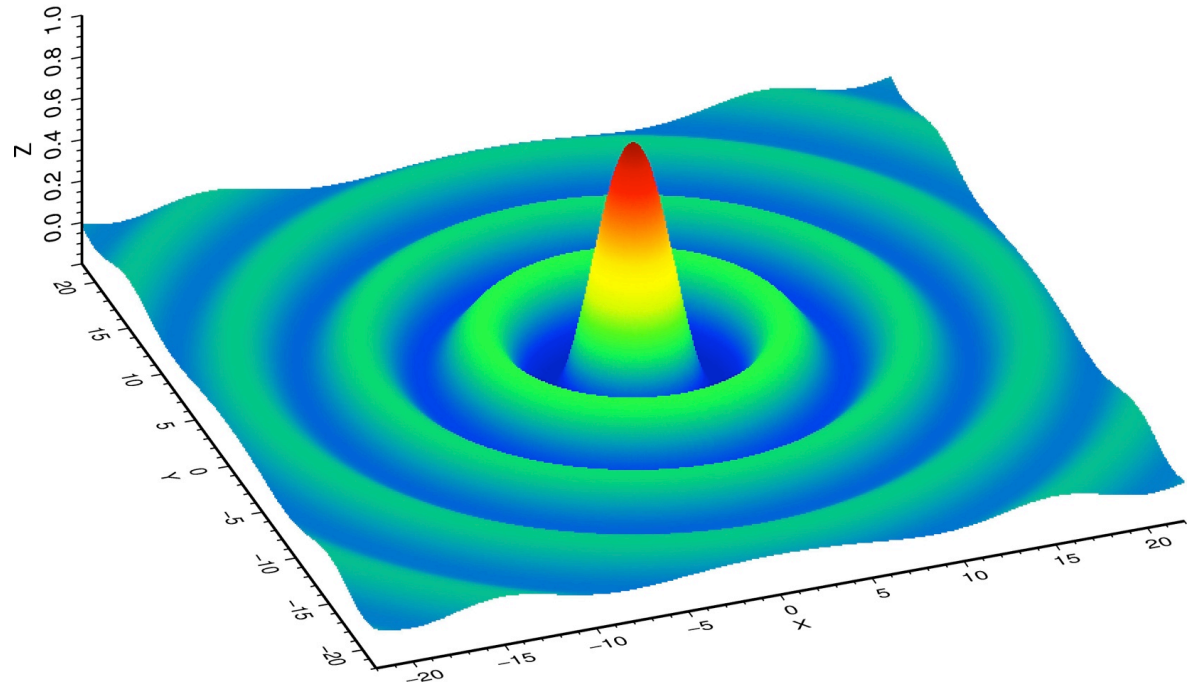
Maskiert



Möchte man aber nun einen Plot in einer 3D-Perspektive erstellen, dann kann dafür GRDVIEW verwendet werden. Es ändert sich nicht viel im Vergleich zu GRDIMAGE. Es muss nur zusätzlich eine Blickrichtung ($-E$ Azimuth/Höhe) und die Höhenskalierung ($-JZ$) angegeben werden.

```
#!/bin/tcsh
gmtset LABEL_OFFSET 0.25
set xmin=-22
set xmax=22
set ymin=-22
set ymax=22
set dx=0.05
set e=200/30
set r=$xmin/$xmax/$ymin/$ymax
set r2=$xmin/$xmax/$ymin/$ymax/-0.2/1
set jx=X15/15
set jz=Z5
makecpt -Cseis -T-0.2/1/0.1 -Z -D -I > amp.cpt
echo $xmin $xmax $ymin $ymax $dx | awk '{nx=($2-$1)/$5;ny=($4-$3)/$5;for(i=-nx/2;i<=nx/2;i++)\
{for(j=-ny/2;j<=ny/2;j++){r=sqrt((i*$5)**2+(j*$5)**2);if(r!=0)print i*$5,j*$5,sin(r)/(r)}}}' | \
xyz2grd -Gsinc.grd -R$r -I$dx
```

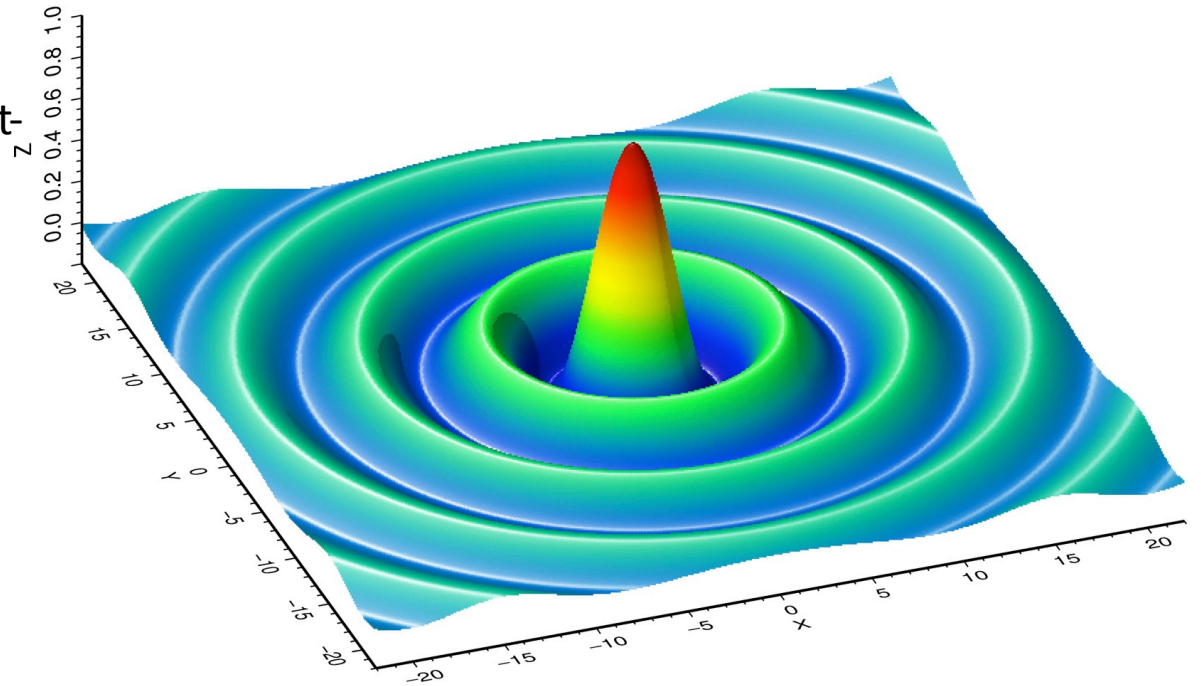
```
grdview sinc.grd -Camp.cpt -R$r2 -J$jx -J$jz -E$e -K -Qi > sinc_3D.ps
psbasemap -R$r2 -J$jx -J$jz -E$e -Ba5f1:"X":/a5f1:"Y":/a0.2f0.05:"Z":WSZ -O >> sinc_3D.ps
gv sinc_3D.ps
```



Möchte man aber nun einen Plot in einer 3D-Perspektive erstellen, dann kann dafür GRDVIEW verwendet werden. Es ändert sich nicht viel im Vergleich zu GRDIMAGE. Es muss nur zusätzlich eine Blickrichtung ($-E$ Azimuth/Höhe) und die Höhenskalierung ($-JZ$) angegeben werden.

Mit $-I$ intfile können auch Intensitäten dargestellt werden, was sich hervorragend für das Plotten von Schatten eignet. Das "Licht" und die Richtung des "Lichtes", welches den Schatten produziert wird mit GRDGRADIENT erzeugt. Mit GRDHISTEQ kann der Schatten noch geglättet werden.

```
#!/bin/tcsh
gmtset LABEL_OFFSET 0.25
set xmin=-22
set xmax=22
set ymin=-22
set ymax=22
set dx=0.05
set e=200/30
set r=$xmin/$xmax/$ymin/$ymax
set r2=$xmin/$xmax/$ymin/$ymax/-0.2/1
set jx=X15/15
set jz=Z5
makecpt -Cseis -T-0.2/1/0.1 -Z -D -I > amp.cpt
echo $xmin $xmax $ymin $ymax $dx | awk '{nx=($2-$1)/$5;ny=($4-$3)/$5;for(i=-nx/2;i<=nx/2;i++)\
{for(j=-ny/2;j<=ny/2;j++){r=sqrt((i*$5)**2+(j*$5)**2);if(r!=0)print i*$5,j*$5,sin(r)/(r)}}}' | \
xyz2grd -Gsinc.grd -R$r -I$dx
grdgradient sinc.grd -Gtmp -E245/30 -N -V
grdhisteq tmp -Gtmp.grd -N -V
grdmath tmp.grd 0.4 x = int.grd
grdview sinc.grd -Camp.cpt -Iint.grd -R$r2 -J$jx -J$jz -E$e -K -Qi > sinc_3D.ps
psbasemap -R$r2 -J$jx -J$jz -E$e -Ba5f1:"X":/a5f1:"Y":/a0.2f0.05:"Z":WSZ -O >> sinc_3D.ps
gv sinc_3D.ps
```



Einfache (Übersichts-) Karten können mit PSCOAST geplottet werden. Wichtige Flags sind:

- C Farbe von Seen
- S Farbe von "nassen" Flächen
- G Farbe von trockenen Flächen
- I Plotten von Flüssen
- N Plotten Grenzen
- D Auflösung der Daten (**full,high,intermediate,low,crude**)

```
#!/bin/tcsh
```

```
set r=5/16/47/55.5
```

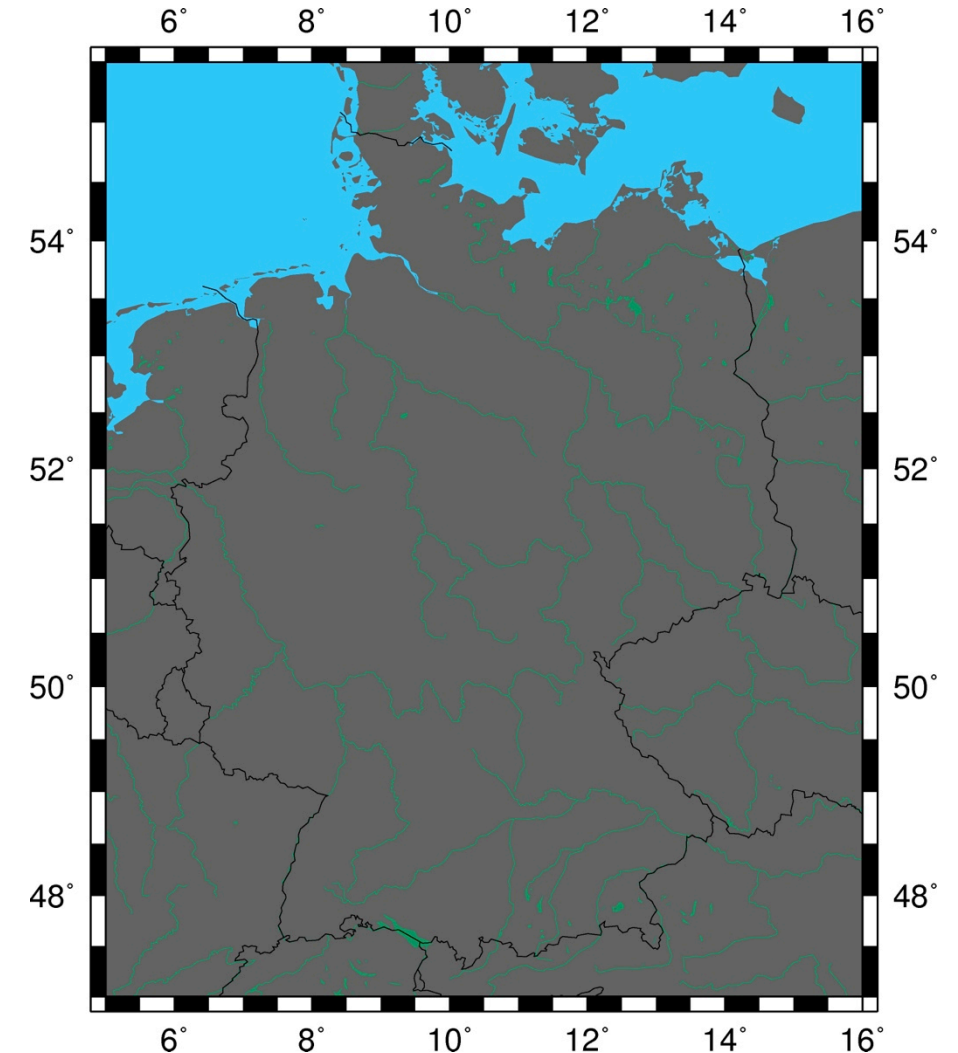
```
set j=M10
```

```
pscoast -R$r -J$j -K -V -S50/200/250 -C0/150/100 -Di -G100/100/100 -Ir/0/150/100 -N1 -P > pscoast.ps
```

```
psbasemap -R -J -Ba2f0.5/a2f0.5 -O -V >> pscoast.ps
```

```
gv pscoast.ps
```









```
ps2raster -A -E600 -Tj pscoast.ps
```



```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 14

cat << EOF > Legende.legend
#Ueberschrift
H 18 1 Legende
#Horizontale Linie der Staerke 1p
D 0 1p
#4 Spalten
N 4
#Anfang Senkrechte Linien zwischen den Spalten
V 0 1p
#####
#Spaltenueberschrift fuer Spalte1
L 16 1 C Spalte1
#Spaltenueberschrift fuer Spalte2
L 16 2 C Spalte2
#Spaltenueberschrift fuer Spalte3
L 16 3 C Spalte3
#Spaltenueberschrift fuer Spalte4
L 16 4 C Spalte4 @;255/0/0; @%34%\244@%% @;;
#Horizontale Linie der Staerke 1p
D 0 1p
#####
S 0.25 s 0.6 255/255/200 0.25p 0.7 Zelle 1.1
S 0.25 c 0.5 200/255/200 0.25p 0.7 Zelle 1.2
S 0.25 i 0.5 255/150/200 0.25p 0.7 Zelle 1.3
S 0.25 a 0.6 200/150/100 0.25p 0.7 Zelle 1.4
```

```
#####
S 255
S 0.25 s 0.6 100/255/100 0.25p 1 Zelle 2.2
S 255
S 255
#####
S 255
S 0.25 s 0.6 0/255/0 0.25p 0.7 Zelle 3.2
S 0.25 s 0.6 p300/20:F0/0/0B255/0/0 0.25p 0.7 Zelle
3.3
S 255
#Ende Senkrechte Linien zwischen den Spalten
V
EOF
pslegend Legende.legend -Dx7/14/14/4.5/TC -JX20/5 -
R0/1/0/1 -F -L1.5 -V -G255 > legend.ps
gv legend.ps
ps2raster -A -E600 -Tj legend.ps
```

Legende			
Spalte1	Spalte2	Spalte3	Spalte4 
 Zelle 1.1	 Zelle 1.2  Zelle 2.2  Zelle 3.2	 Zelle 1.3  Zelle 3.3	 Zelle 1.4

Beispiel Topo-Karte

Als Beispiel für die Zusammenarbeit zwischen Unix-Kommandos, AWK, SED und GMT soll eine Topographiekarte von Jena und Umgebung erstellt werden.

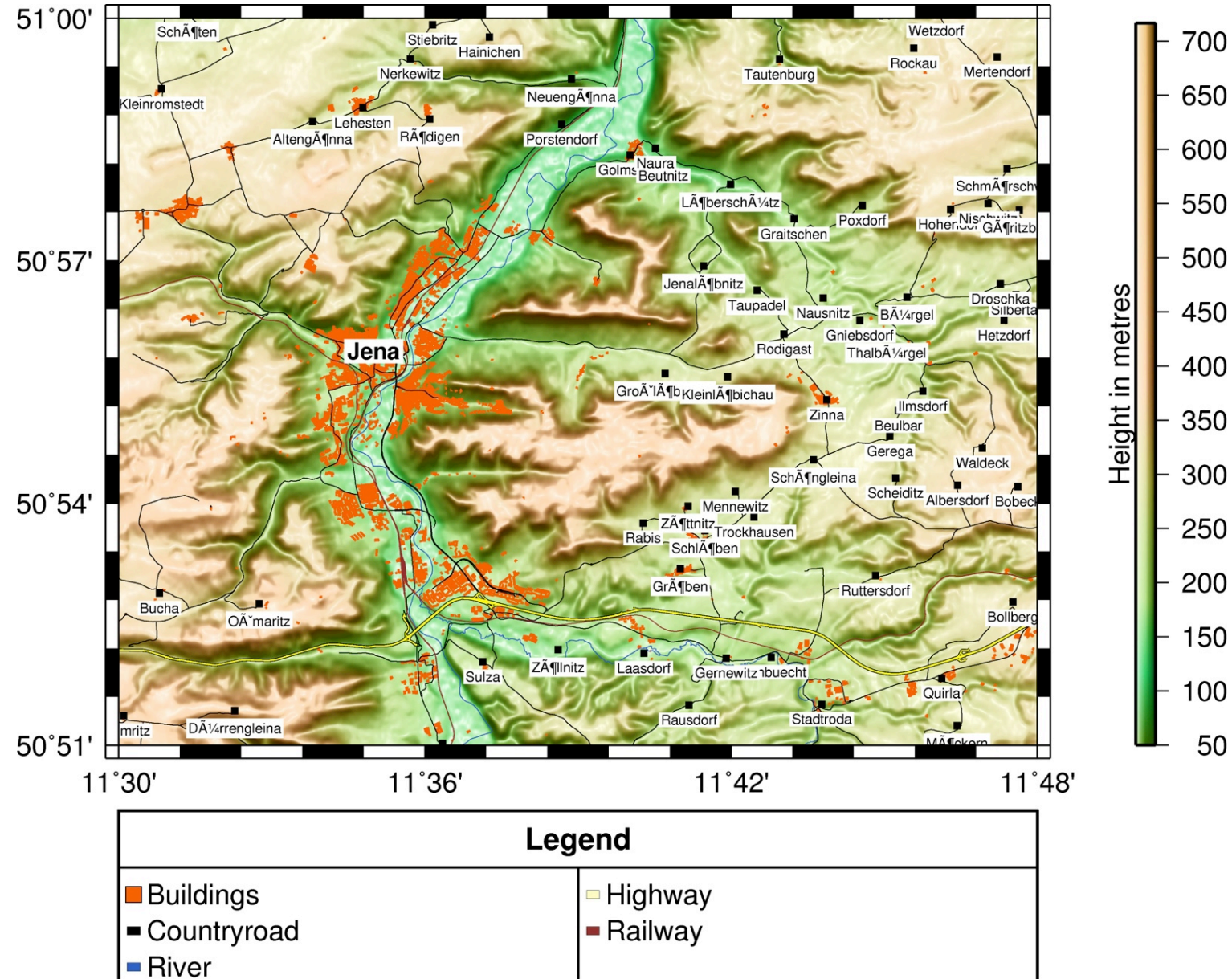
Das Topographie-Grid kann man sich unter <http://dds.cr.usgs.gov/srtm/version1/> herunterladen.

Daten für Straßen, Städte, usw. bekommt man als Shape-File unter:

<http://download.geofabrik.de/>

Zum Umwandeln von Shape-File zu ASCII-File kann man sich shp2txt runterladen:

<http://legacy.obviously.com/gis/shp2text/>



Beispiel Topo-Karte in 3D

Es ist natürlich auch möglich die Karte in 3D zu plotten. Dafür müssen nur folgende Änderungen durchgeführt werden:

1. GRDIMAGE -> GRDVIEW und PSXY -> PSXYZ
2. Flags `-EAzimuth/elev` und `-JZheight` bei jedem Kommando hinzufügen
3. Den Wertebereich bei `-R` um min. Höhe und max. Höhe erweitern
4. Mit `grdtrack` die Höhenwerte für die Position jeder Linie/Symbol extrahieren
5. Mit `AWK` den Wertebereich einschränken, weil sonst über die Ränder geplottet wird

