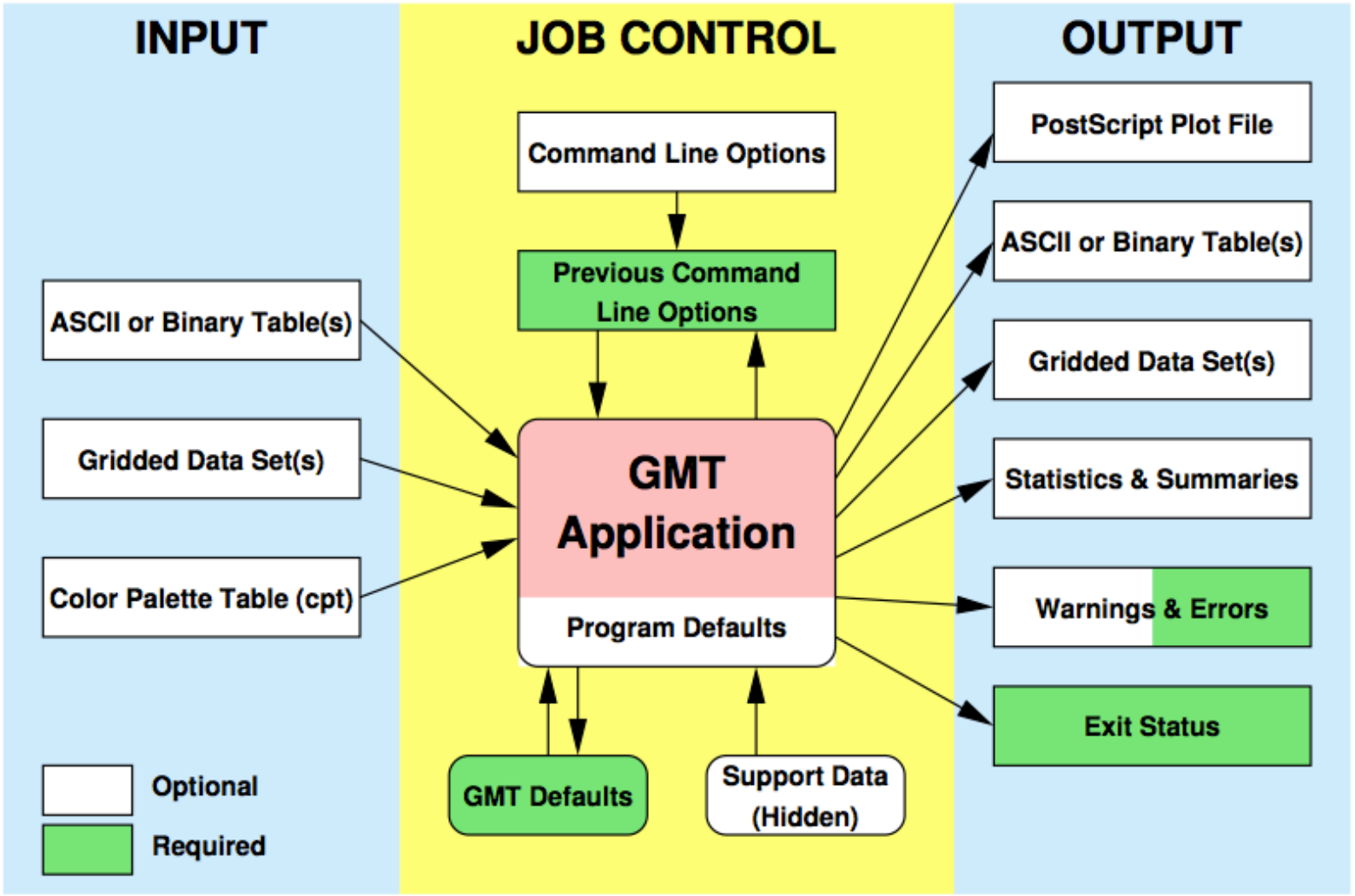
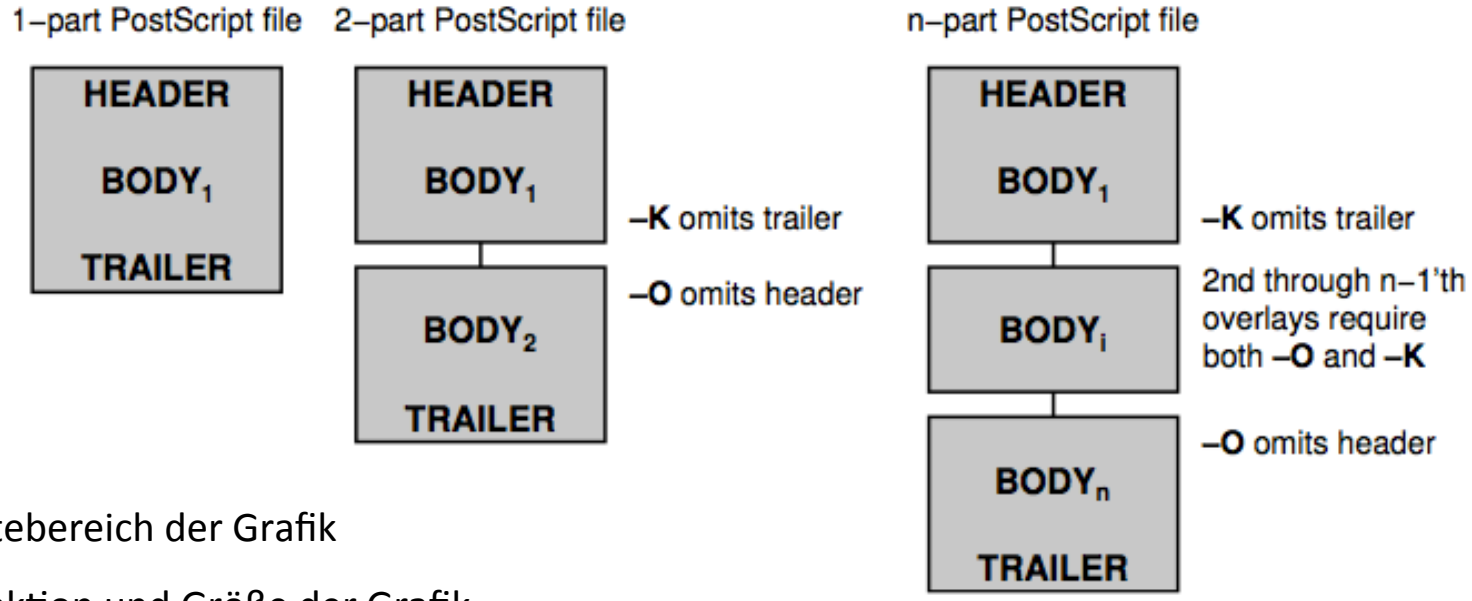
GMT (**G**eneric **M**apping **T**ools) ist ein kommandozeilenbasiertes open-source Softwarepaket, welches 1987 von Paul Wessel und Walter Smith entwickelt wurde. Es wurde seitdem ständig weiterentwickelt und ist heutzutage für Linux/Unix, Windows und Mac verfügbar (https://www.soest.hawaii.edu/gmt).



Mit GMT können Daten geplottet (Vektordaten, Rasterdaten, ...), manipuliert (Filtern, Resamplen, math. Operationen,...) und konvertiert (z.B. ASCII zu Binär) werden. GMT besteht aus über 60 individuellen Programmen die miteinander kombiniert werden können. Im Zusammenspiel mit UNIX-Kommandos und AWK/SED können die Programme in SHELL-Skripten aufgelistet werden und somit umfangreiche Grafiken erstellt werden.

Auf den nächsten Folien wird zuerst gezeigt, wie Vektordaten und Rasterdaten geplottet werden. Anschließend wird auf das Manipulieren eingegangen.

Die einzelnen GMT-Kommandos werden über sogenannte Flags gesteuert. Manche Flags haben meistens die gleiche Bedeutung:



-Rxmin/xmax/ymin/ymax[/zmin/zmax]	Wertebereich der Grafik
-Jmapproj	Projektion und Größe der Grafik
-Ix_inc[/y_inc]	Increment der Rasterdaten in x- und/oder y-Richtung
-Ggridname	Name des gridfiles mit den Rasterdaten oder Füllfarbe
-Bframeoptions	Rahmenbeschriftung
-O	Dieser GMT-Befehl folgt einem GMT-Befehl
-K	Es folgen weitere GMT-Befehle
-P	Portrait
-Xx_dist	Verschiebe Grafik in X-Richtung um x_dist Zentimeter
-Yy_dist	Verschiebe Grafik in Y-Richtung um y_dist Zentimeter

Default-Werte

In GMT sind diverse Formate per default eingestellt (z.B. Schriftgröße, Achsenabstand der Schrift,...). Die Default-Werte kann man sich mit `gmtdefaults -L` im Terminal anzeigen lassen:

```
#
#              GMT-SYSTEM 4.5.11 [64-bit] Defaults file
#
#-------- Plot Media Parameters -------------
PAGE_COLOR                     = white
PAGE_ORIENTATION               = landscape
PAPER_MEDIA                    = a4
#-------- Basemap Annotation Parameters ------
ANNOT_MIN_ANGLE                              = 20
ANNOT_MIN_SPACING              = 0
ANNOT_FONT_PRIMARY             = Helvetica
ANNOT_FONT_SIZE_PRIMARY        = 14p
ANNOT_OFFSET_PRIMARY           = 0.2c
ANNOT_FONT_SECONDARY           = Helvetica
ANNOT_FONT_SIZE_SECONDARY      = 16p
ANNOT_OFFSET_SECONDARY         = 0.2c
DEGREE_SYMBOL                  = ring
HEADER_FONT                    = Helvetica
HEADER_FONT_SIZE               = 36p
HEADER_OFFSET                  = 0.5c
LABEL_FONT                     = Helvetica
LABEL_FONT_SIZE                              = 24p
LABEL_OFFSET                   = 0.3c
OBLIQUE_ANNOTATION             = 1
PLOT_CLOCK_FORMAT              = hh:mm:ss
PLOT_DATE_FORMAT               = yyyy-mm-dd
PLOT_DEGREE_FORMAT             = ddd:mm:ss
Y_AXIS_TYPE                    = hor_text
#-------- Basemap Layout Parameters ---------
BASEMAP_AXES                   = WESN
BASEMAP_FRAME_RGB              = black
BASEMAP_TYPE                   = fancy
FRAME_PEN                      = 1.25p
FRAME_WIDTH                    = 0.2c
GRID_CROSS_SIZE_PRIMARY        = 0c
GRID_PEN_PRIMARY               = 0.25p
GRID_CROSS_SIZE_SECONDARY      = 0c
GRID_PEN_SECONDARY             = 0.5p
MAP_SCALE_HEIGHT               = 0.2c
POLAR_CAP                      = 85/90
TICK_LENGTH                    = 0.2c
TICK_PEN                       = 0.5p
X_AXIS_LENGTH                  = 25c
Y_AXIS_LENGTH                  = 15c
X_ORIGIN                       = 2.5c
Y_ORIGIN                       = 2.5c
UNIX_TIME                      = FALSE
UNIX_TIME_POS                  = BL/-2c/-2c
UNIX_TIME_FORMAT               = %Y %b %d %H:%M:%S
#-------- Color System Parameters -----------
COLOR_BACKGROUND               = black
COLOR_FOREGROUND               = white
COLOR_NAN                      = 128
COLOR_IMAGE                    = adobe
COLOR_MODEL                    = rgb
HSV_MIN_SATURATION             = 1
HSV_MAX_SATURATION             = 0.1
HSV_MIN_VALUE                  = 0.3
HSV_MAX_VALUE                  = 1
#-------- PostScript Parameters -------------
CHAR_ENCODING                  = ISOLatin1+
DOTS_PR_INCH                   = 300
GLOBAL_X_SCALE                 = 1
GLOBAL_Y_SCALE                 = 1
N_COPIES                       = 1
PS_COLOR                       = rgb
PS_IMAGE_COMPRESS              = lzw
PS_IMAGE_FORMAT                              = ascii
PS_LINE_CAP                    = butt
PS_LINE_JOIN                   = miter
PS_MITER_LIMIT                 = 35
PS_VERBOSE                     = FALSE
TRANSPARENCY                   = 0
#-------- I/O Format Parameters -------------
D_FORMAT                       = %.12lg
FIELD_DELIMITER                              = tab
GRIDFILE_FORMAT                              = nf
GRIDFILE_SHORTHAND             = FALSE
INPUT_CLOCK_FORMAT             = hh:mm:ss
INPUT_DATE_FORMAT              = yyyy-mm-dd
IO_HEADER                      = FALSE
N_HEADER_RECS                  = 1
NAN_RECORDS                    = pass
OUTPUT_CLOCK_FORMAT            = hh:mm:ss
OUTPUT_DATE_FORMAT             = yyyy-mm-dd
OUTPUT_DEGREE_FORMAT           = D
XY_TOGGLE                      = FALSE
#-------- Projection Parameters -------------
ELLIPSOID                      = WGS-84
MAP_SCALE_FACTOR               = default
MEASURE_UNIT                   = cm
#-------- Calendar/Time Parameters ----------
TIME_FORMAT_PRIMARY            = full
TIME_FORMAT_SECONDARY          = full
TIME_EPOCH                     = 2000-01-01T12:00:00
TIME_IS_INTERVAL               = OFF
TIME_INTERVAL_FRACTION         = 0.5
TIME_LANGUAGE                  = us
TIME_UNIT                      = d
TIME_WEEK_START                              = Sunday
Y2K_OFFSET_YEAR                              = 1950
#-------- Miscellaneous Parameters ----------
HISTORY                                      = TRUE
INTERPOLANT                    = akima
LINE_STEP                      = 0.025c
VECTOR_SHAPE                   = 0
VERBOSE                                      = FALSE
```

Für das Ändern dieser Default-Werte gibt es 2 Möglichkeiten. Wenn die Änderung ab einer bestimmten Stelle für das gesamte Skript gelten soll, dann an der Stelle die Wertzuweisung mit gmtset durchführen (z.B. `gmtset ANNOT_FONT_SIZE 12` oder `gmtset ANNOT_FONT_SIZE=12`). Soll die Änderung nur für ein Kommando gelten, dann kann die Änderung als Flag mit angegeben werden (z.B. `--ANNOT_FONT_SIZE=12`).

Es soll zuerst ein Rahmen mit psbasemap geplottet werden. Als Pflicht-Flags müssen **-R** (Wertebereich), **-J** (Projektion und Größe) und **-B** (Rahmenbeschriftung) gesetzt werden:

```
psbasemap -R0/1000/0/1000 -JX15/10 -Ba100f10:"X":/a100f10g50:"Y":."Erster Rahmen"::WSen -P > rahmen.ps
```

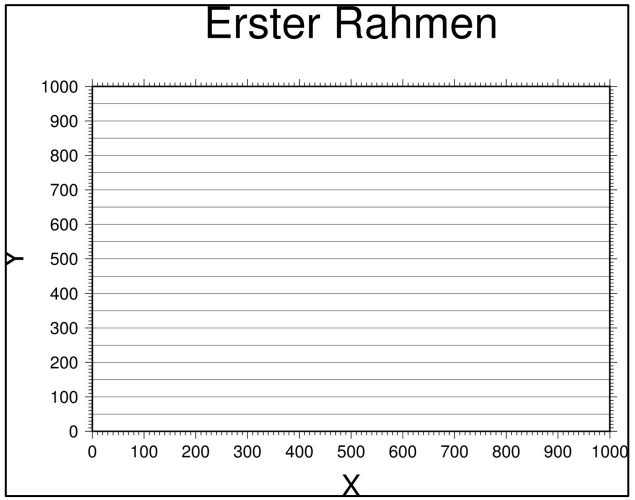
Die erzeugte Grafik mit GMT wird immer in ein PostScript-Format ausgegeben.
Die Grafik kann dann mit `gv rahmen.ps` angeschaut werden.

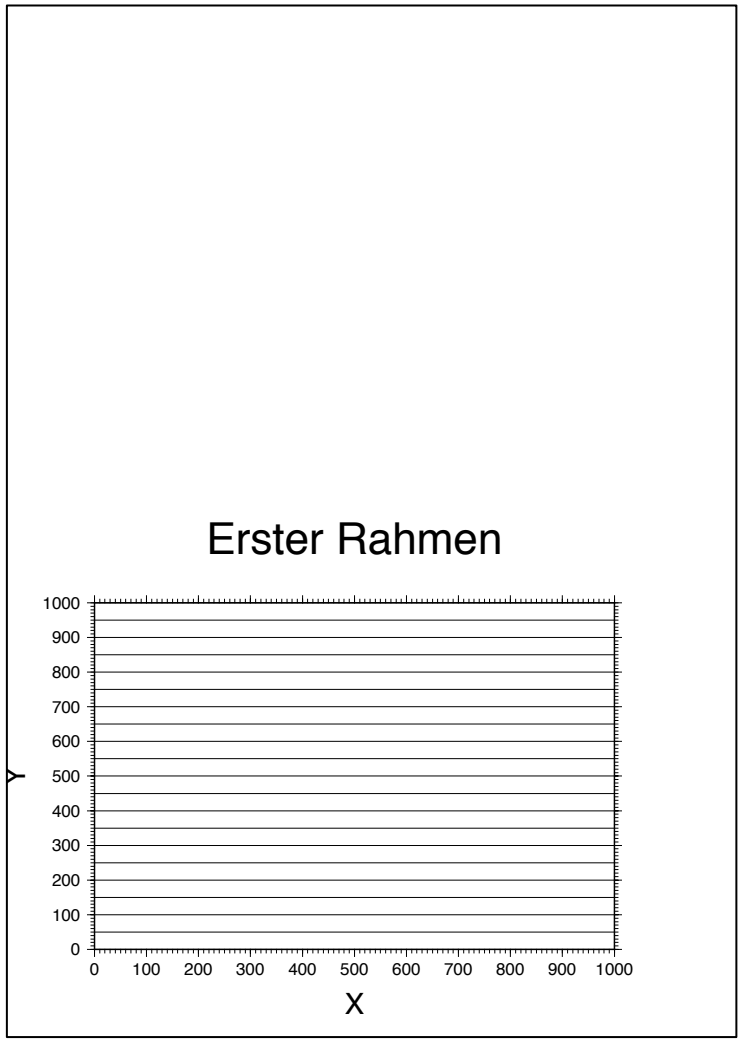
Das Papierformat ist per default A4. Um die Papergröße an die Grafik anzupassen kann ps2raster verwendet werden:

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

-A sorgt dafür, dass die Papiergröße an die Größe der Grafik angepasst wird
-E gibt die Auflösung in Dpi an
-T definiert das Ausgabeformat der Grafik:

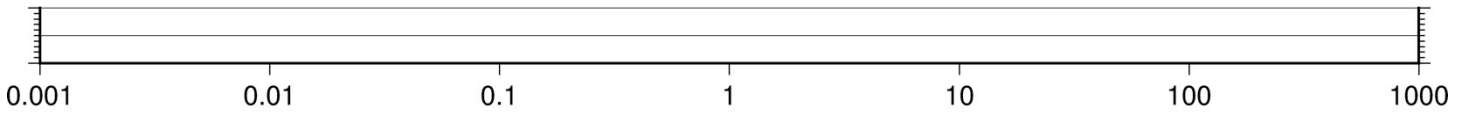
- -Tb -> .bmp
- -Tj -> .jpg
- -Te -> .eps
- -Tf -> .pdf
- -Tg -> .png
- -Tm -> .ppm
- -Tt -> .tif



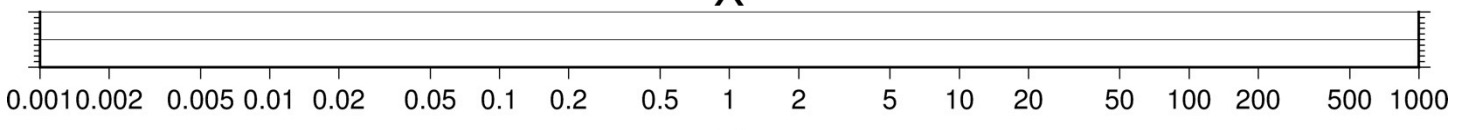


Die Achsen können, wie gerade gezeigt, einen linearen Verlauf haben. Es ist aber auch möglich, einen logarithmischen Verlauf zu wählen. Dafür muss bei dem -J Flag nach der Länge für die betreffende Achse noch ein l angehängt werden. Die Darstellungsart kann dann im -B Flag eingestellt werden.

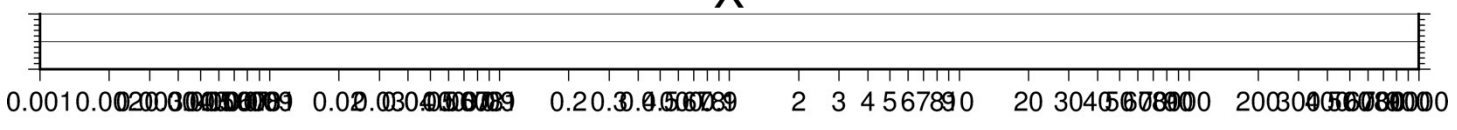
```
-JX15l/10 -Ba1:"X":/a100f10g50:...
```



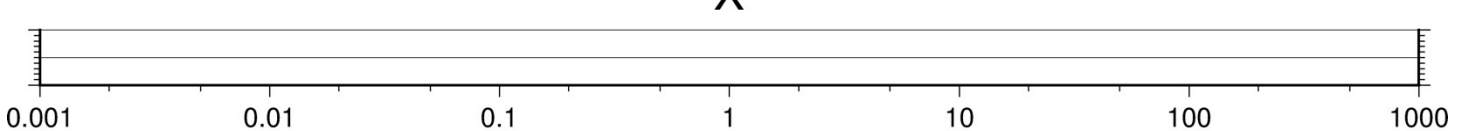

```
-JX15l/10 -Ba2:"X":/a100f10g50:...
```



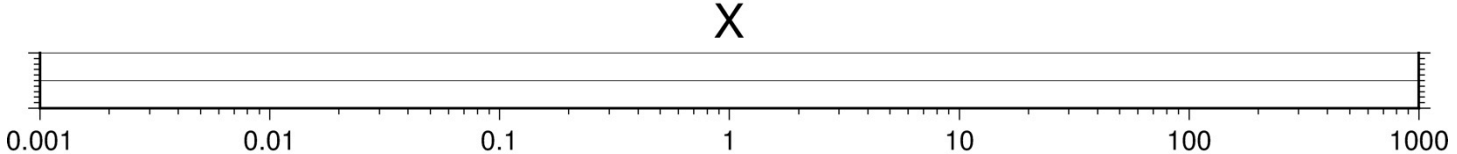

```
-JX15l/10 -Ba3:"X":/a100f10g50:...
```



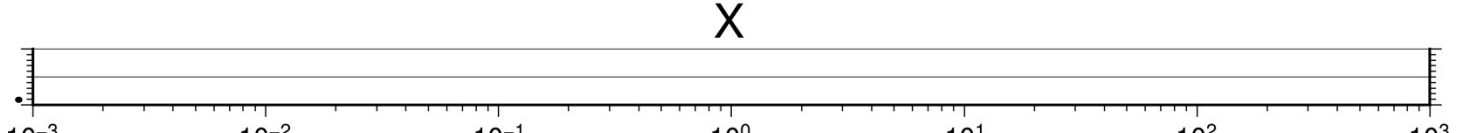

```
-JX15l/10 -Ba1f2:"X":/a100f10g50:...
```



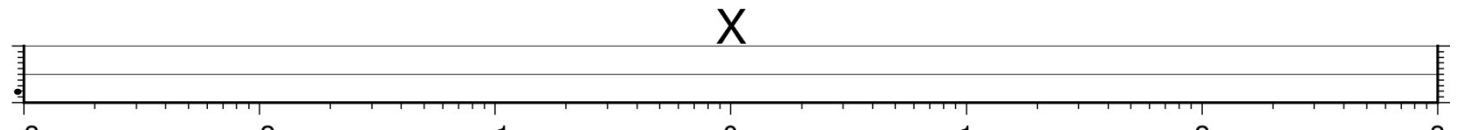

```
-JX15l/10 -Ba1f3:"X":/a100f10g50:...
```



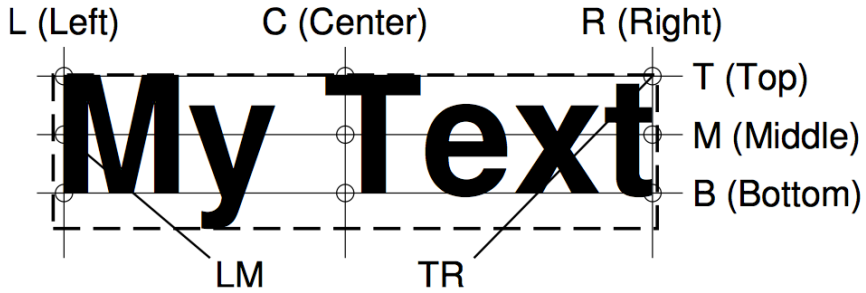

```
-JX15l/10 -Ba1pf3:"X":/a100f10g50:...
```



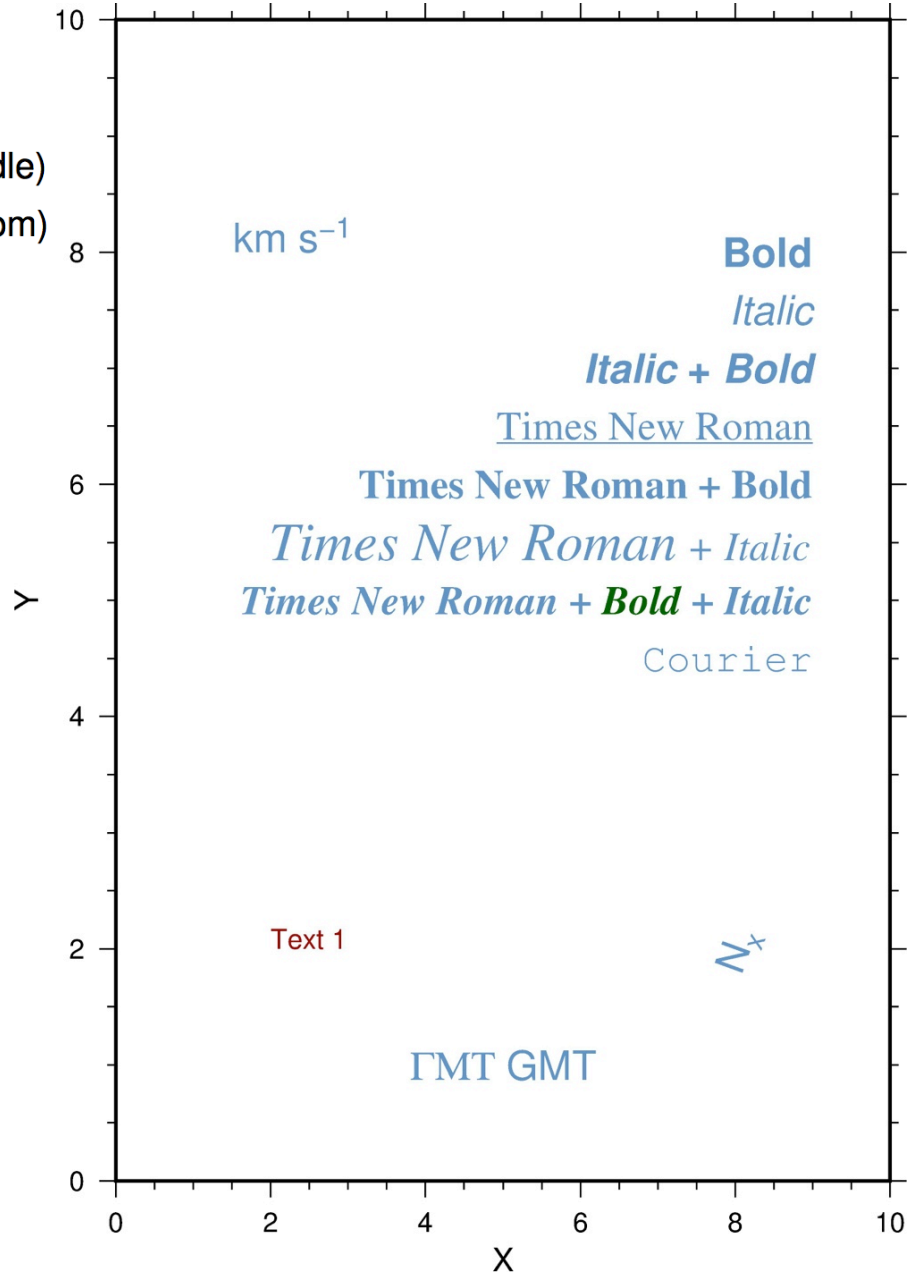

```
-JX15l/10 -Ba1lf3:"X":/a100f10g50:...
```



Mit PSTEXT können Schriften in Grafiken eingefügt werden. Jede Textzeile folgt auf einen festdefinierten Formatbeschreiber.
Als Beispiel dient das Skript text.gmt:




```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 12
gmtset ANNOT_FONT_SIZE 10
gmtset LABEL_OFFSET 0.0
set psfile=rahmen.ps
psbasemap -R0/10/0/10 -JX10/15 -Ba2f0.5:"X":/a2f0.5:"Y":WSen -K > $psfile
echo 2 2 10 0 0 LB Text 1 | pstext -R -J -O -K -V -G150/0/0 >> $psfile
pstext -R -J -O  -V -G100/150/200 << END >> $psfile
 3 8.0 15  0 0 RB km s@+-1@+
 8 2.0 15 70 0 CM N@-x@-
 9 8.0 15  0 1 RM Bold
 9 7.5 15  0 2 RM Italic
 9 7.0 15  0 3 RM Italic + Bold
 9 6.5 15  0 4 RM @_Times New Roman@_
 9 6.0 15  0 5 RM Times New Roman + Bold
 9 5.5 15  0 6 RM @:20:Times New Roman@:: + Italic
 9 5.0 15  0 7 RM Times New Roman + @;0/100/0;Bold@;; + Italic
 9 4.5 15  0 8 RM Courier
 5 1.0 15  0 0 CM @%12%\107\115\124@%% GMT
END
ps2raster -A -E600 -Tj $psfile
```



Die 5te Spalte des Formatbeschreibers repräsentiert die Schriftart. In GMT sind 34 Schriftarten verfügbar (siehe rechte Tabelle). Das Ändern der Schriftart innerhalb einer Zeile erfolgt durch ein voranstellen von @%fontno%. Die vorherige Schriftart kann dann wieder aktiviert werden mittels @%% (siehe linke Tabelle).

Code	*Effect*
@~	Turns symbol font on or off
@+	Turns superscript on or off
@-	Turns subscript on or off
@#	Turns small caps on or off
@_	Turns underline on or off
@%*fontno*%	Switches to another font; @%% resets to previous font
@:*size*:	Switches to another font size; @:: resets to previous size
@;*color*;	Switches to another font color; @;; resets to previous color
@!	Creates one composite character of the next two characters
@@	Prints the @ sign itself

#	Font Name	#	Font Name
0	Helvetica	**17**	**Bookman-Demi**
1	**Helvetica-Bold**	***18***	***Bookman-DemiItalic***
2	*Helvetica-Oblique*	19	Bookman-Light
3	***Helvetica-BoldOblique***	*20*	*Bookman-LightItalic*
4	Times-Roman	21	Helvetica-Narrow
5	**Times-Bold**	**22**	**Helvetica-Narrow-Bold**
6	*Times-Italic*	*23*	*Helvetica-Narrow-Oblique*
7	***Times-BoldItalic***	***24***	***Helvetica-Narrow-BoldOblique***
8	Courier	25	NewCenturySchlbk-Roman
9	**Courier-Bold**	*26*	*NewCenturySchlbk-Italic*
10	*Courier-Oblique*	**27**	**NewCenturySchlbk-Bold**
11	***Courier-BoldOblique***	***28***	***NewCenturySchlbk-BoldItalic***
12	Σψμβολ (Symbol)	29	Palatino-Roman
13	AvantGarde-Book	*30*	*Palatino-Italic*
14	*AvantGarde-BookOblique*	**31**	**Palatino-Bold**
15	**AvantGarde-Demi**	***32***	***Palatino-BoldItalic***
16	***AvantGarde-DemiOblique***	*33*	*ZapfChancery-MediumItalic*
		34	✺✪❒❄✣✻■✳✪✪▼▲ (ZapfDingbats)

PSTEXT

GMT

Symbol (@%12%)

octal	0	1	2	3	4	5	6	7
\04x		!	∀	#	∃	%	&	∋
\05x	(	)	*	+	,	−	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	≅	A	B	X	Δ	E	Φ	Γ
\11x	H	I	ϑ	K	Λ	M	N	O
\12x	Π	Θ	P	Σ	T	Y	ς	Ω
\13x	Ξ	Ψ	Z	[	∴	]	⊥	–
\14x	‾	α	β	χ	δ	ε	ϕ	γ
\15x	η	ι	φ	κ	λ	μ	ν	ο
\16x	π	θ	ρ	σ	τ	υ	ϖ	ω
\17x	ξ	ψ	ζ	{	|	}	~	

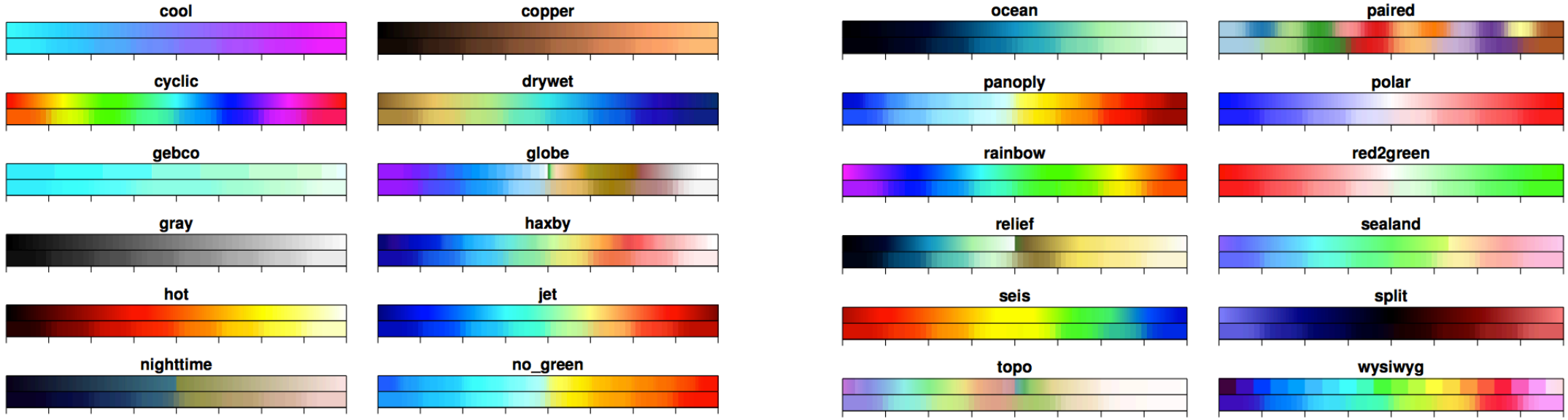
\24x	€	ϒ	′	≤	⁄	∞	ƒ	♣
\25x	♦	♥	♠	↔	←	↑	→	↓
\26x	°	±	″	≥	×	∝	∂	•
\27x	÷	≠	≡	≈	…	|	—	↵
\30x	ℵ	ℑ	ℜ	℘	⊗	⊕	∅	∩
\31x	∪	⊃	⊇	⊄	⊂	⊆	∈	∉
\32x	∠	∇	®	©	™	∏	√	·
\33x	¬	∧	∨	⇔	⇐	⇑	⇒	⇓
\34x	◊	〈	®	©	™	∑	⎛	⎜
\35x	⎝	⎡	⎢	⎣	⎧	⎨	⎩	⎪
\36x		〉	∫	⌠	⎮	⌡	⎫	⎪
\37x	⎭	⎤	⎥	⎦	⎫	⎬	⎭	

ZapfDingbats (@%34%)

octal	0	1	2	3	4	5	6	7
\04x		✁	✂	✃	✄	☎	✆	✇
\05x	✈	✉	☛	☞	✌	✍	✎	✏
\06x	✐	✑	✒	✓	✔	✕	✖	✗
\07x	✘	✙	✚	✛	✜	✝	✞	✟
\10x	✠	✡	✢	✣	✤	✥	✦	✧
\11x	★	☆	✪	✫	✬	✭	✮	✯
\12x	✰	✱	✲	✳	✴	✵	✶	✷
\13x	✸	✹	✺	✻	✼	✽	✾	✿
\14x	❀	❁	❂	❃	❄	❅	❆	❇
\15x	❈	❉	❊	❋	●	○	■	❑
\16x	❒	❑	❒	▲	▼	◆	❖	◗
\17x	|	❙	❚	❛	❜	❝	❞	

\24x		❡	❢	❢	❤	❥	❦	❧
\25x	♣	♦	♥	♠	①	②	③	④
\26x	⑤	⑥	⑦	⑧	⑨	⑩	❶	❷
\27x	❸	❹	❺	❻	❼	❽	❾	❿
\30x	①	②	③	④	⑤	⑥	⑦	⑧
\31x	⑨	⑩	❶	❷	❸	❹	❺	❻
\32x	❼	❽	❾	❿	→	→	↔	↕
\33x	↘	→	↗	→	→	→	→	→
\34x	➡	➡	➢	➢	➤	➥	➦	➧
\35x	➨	➩	➪	➫	➬	➭	➮	➯
\36x		➱	➲	➳	➴	➵	➶	➷
\37x	➸	➹	➺	➻	➼	➽	➾	

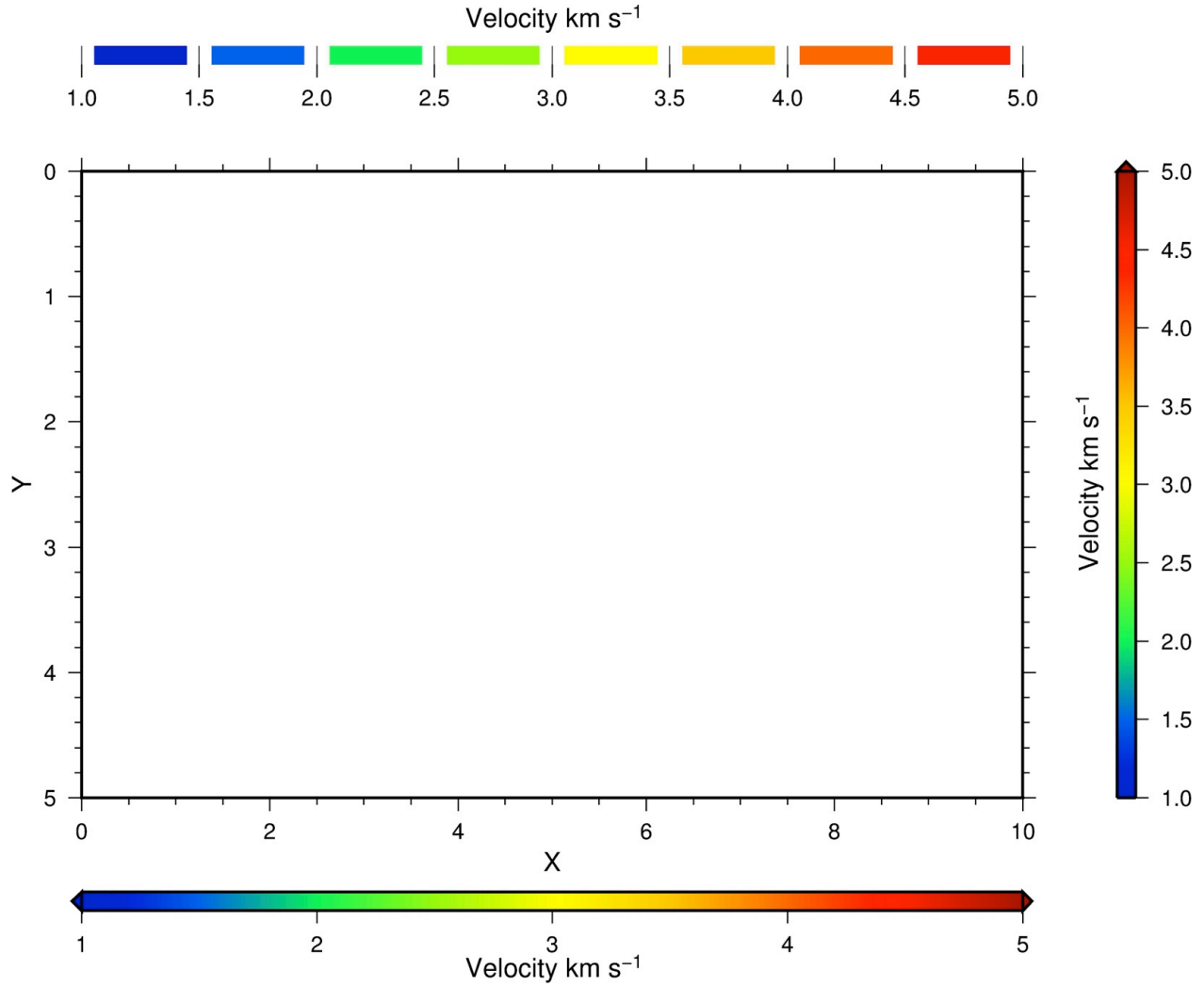
Für das plotten von Raster-Daten sind Farbtabellen unersetzlich. Beim Plotten von Vektorgrafiken sind sie nicht zwingend notwendig, können aber ab und zu ganz nützlich sein. Bei GMT stehen folgende vordefinierte Farbtabellen zur Verfügung:




```
makecpt -Cseis -T1/5/0.5 -D -I –Z > vel.cpt
```

```
#          cpt file created by: makecpt -I -Cseis -T1/5/0.5 -D -Z
#COLOR_MODEL = RGB
#
1          0          0          205        1.5        0          99         237
1.5        0          99         237        2          22         244        90
2          22         244        90         2.5        152        255        19
2.5        152        255        19         3          255        255        0
3          255        255        0          3.5        255        202        0
3.5        255        202        0          4          255        106        0
4          255        106        0          4.5        255        10         0
4.5        255        10         0          5          170        0          0
B          0          0          205
F          170        0          0
N          128        128        128
```

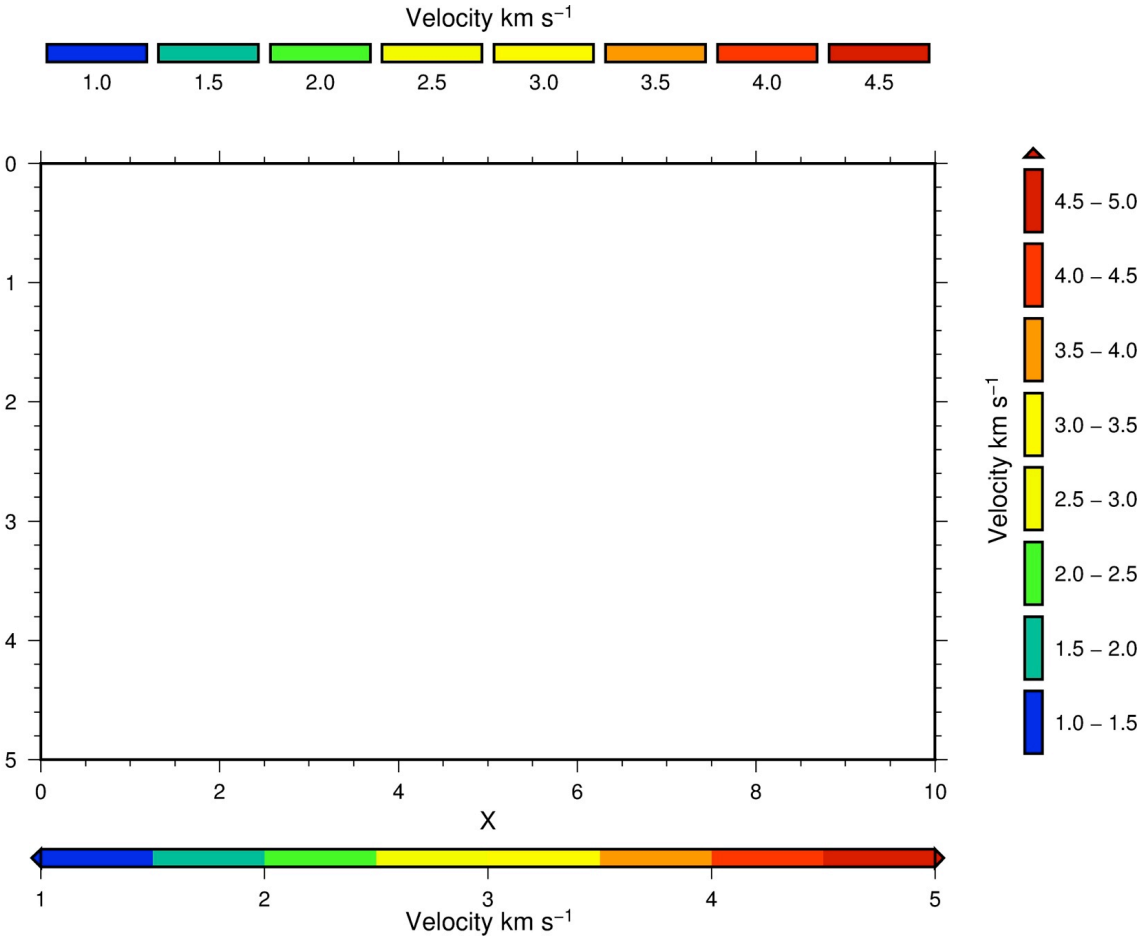
Um sich die Farbskala in der Grafik anzeigen zu lassen, wird das Kommando PSSCALE benötigt. Zum Plotten der erzeugten Farbtabelle auf der vorherigen Folie, kann z.B. folgendes Skript verwendet werden:




```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 12
gmtset ANNOT_FONT_SIZE 10
gmtset LABEL_OFFSET 0.0
set psfile=scale.ps
makecpt -Cseis -T1/5/0.5 -D -I -Z > vel.cpt
psbasemap -R0/10/0/5 -JX15/-10 -Ba2f0.5:"X":/a1f0.2:"Y":WSen -K -P -Y5 > $psfile
psscale -Cvel.cpt -D7.5/-1.5/15/0.3h -Ba1:"Velocity km s@+-1@+": -O -K -E >> $psfile
psscale -Cvel.cpt -D16.5/5/10/0.3  -Ba0.5:"Velocity km s@+-1@+": -Al --LABEL_OFFSET=0.3 -O -K -Ef >> $psfile
psscale -Cvel.cpt -D7.5/15/10/0.3h  -B:"Velocity km s@+-1@+":  -O -Al --LABEL_OFFSET=0.3  -L0.2 >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Wenn die Farbskala in Intervallen geplottet werden soll (-L), dann sollte sie mit makecpt nicht als kontinuierliche Farbtabelle erstellt werden (kein –Z).

```
#         cpt file created by: makecpt -I -Cseis -T1/5/0.5 -D
#COLOR_MODEL = RGB
#
1         0         45        233       1.5       0         45        233
1.5       0         190       156       2         0         190       156
2         73        252       45        2.5       73        252       45
2.5       245       255       2         3         245       255       2
3         255       250       0         3.5       255       250       0
3.5       255       154       0         4         255       154       0
4         255       58        0         4.5       255       58        0
4.5       218       0         0         5         218       0         0
B         0         45        233
F         218       0         0
N         128       128       128
```



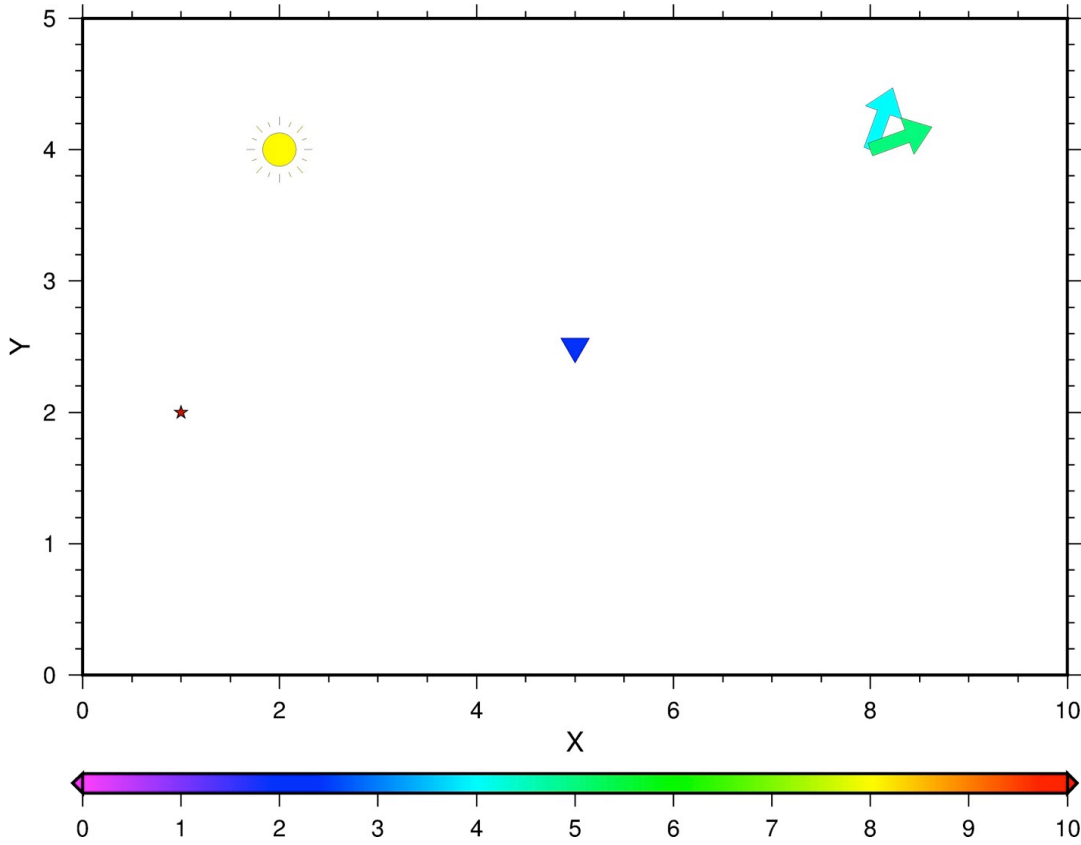

```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 12
gmtset ANNOT_FONT_SIZE 10
gmtset LABEL_OFFSET 0.0
set psfile=scale.ps
makecpt -Cseis -T1/5/0.5 -D -I  > vel.cpt
psbasemap -R0/10/0/5 -JX15/-10 -Ba2f0.5:"X":/a1f0.2:"Y":WSen -K -P -Y5 > $psfile
psscale -Cvel.cpt -D7.5/-1.5/15/0.3h -Ba1:"Velocity km s@+-1@+": -O -K -E >> $psfile
psscale -Cvel.cpt -D16.5/5/10/0.3   -B:"Velocity km s@+-1@+": -Al -K --LABEL_OFFSET=0.3 -O -Ef -Li0.2  >> $psfile
psscale -Cvel.cpt -D7.5/12/15/0.3h  -B:"Velocity km s@+-1@+":  -O  -Al --LABEL_OFFSET=0.3  -L0.2 >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Mit PSXY können Diagramme und Symbole geplottet werden. Der benötigte Input hängt ganz von den gesetzten Flags ab:
`x y [z] [size] [dx] [dy] [symbol]`

`x`	x-Position
`y`	y-Position
`z`	wenn als Flag -Cfarbtabelle.cpt gesetzt ist, dann wird jedes Symbol mit der entsprechenden Farbe geplottet die dem z-Wert entspricht
`size`	Größe des Symbols (Wenn Vektoren mit -Sv oder -SV geplottet werden, dann besteht `size` aus 2 Spalten mit Azimuth und Länge)
`dx`	wenn als Flag -Ex gesetzt wird, dann wird ein Fehlerbalken in X-Richtung geplottet
`dy`	wenn als Flag -Ey gesetzt wird, dann wird ein Fehlerbalken in Y-Richtung geplottet (wenn als Flag -Exy gesetzt wird, dann werden Fehlerbalken in beide Richtungen geplottet)
`symbol`	Symbol

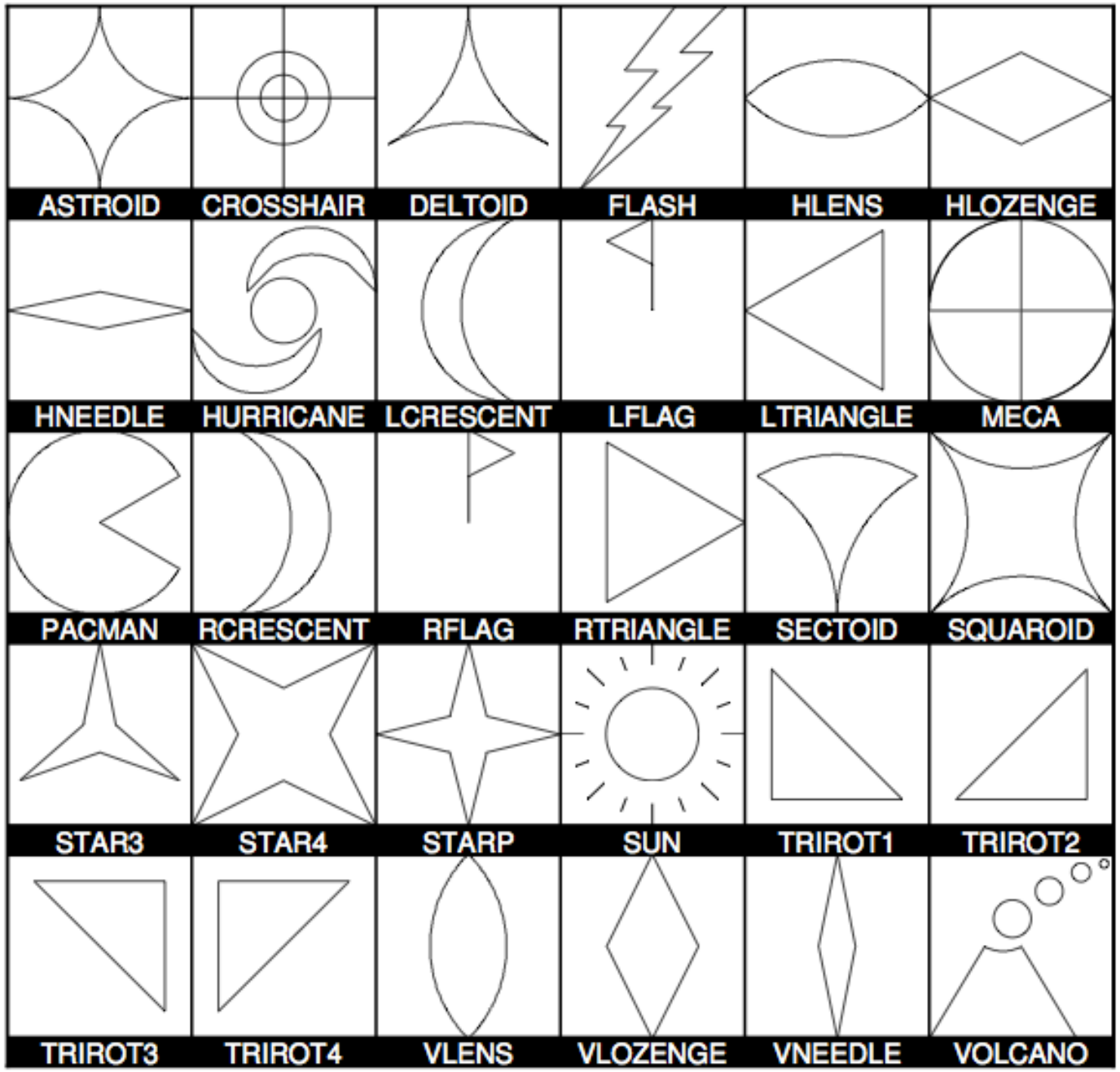
Dazu folgendes Beispiel-Skript:

```
#!/bin/tcsh
gmtset LABEL_FONT_SIZE 12
gmtset ANNOT_FONT_SIZE 10
gmtset LABEL_OFFSET 0.0
set psfile=psxy.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
echo 1 2 | psxy -R -J -O -V -Sa0.2 -W1/0/0/0 -G200/0/0 -K >> $psfile
psxy -R -J -O -K -V -S -W0.000001 -Ccolor.cpt << END >> $psfile
5 2.5 2   0.5 i
8 4   4    70 1 v0.2/0.4/0.3
8 4   5    70 1 V0.2/0.4/0.3
2 4   8  1  kSUN
END
psscale -Ccolor.cpt -D7.5/-1.5/15/0.3h -Ba1 -O -E >> $psfile
ps2raster -A -E600 -Tj $psfile
```



Es können in GMT auch Kustom-Symbole erstellt werden. Per default sind die nebenstehenden Symbole schon verfügbar. Aufgerufen werden diese mit

`-Sk``symbolname`

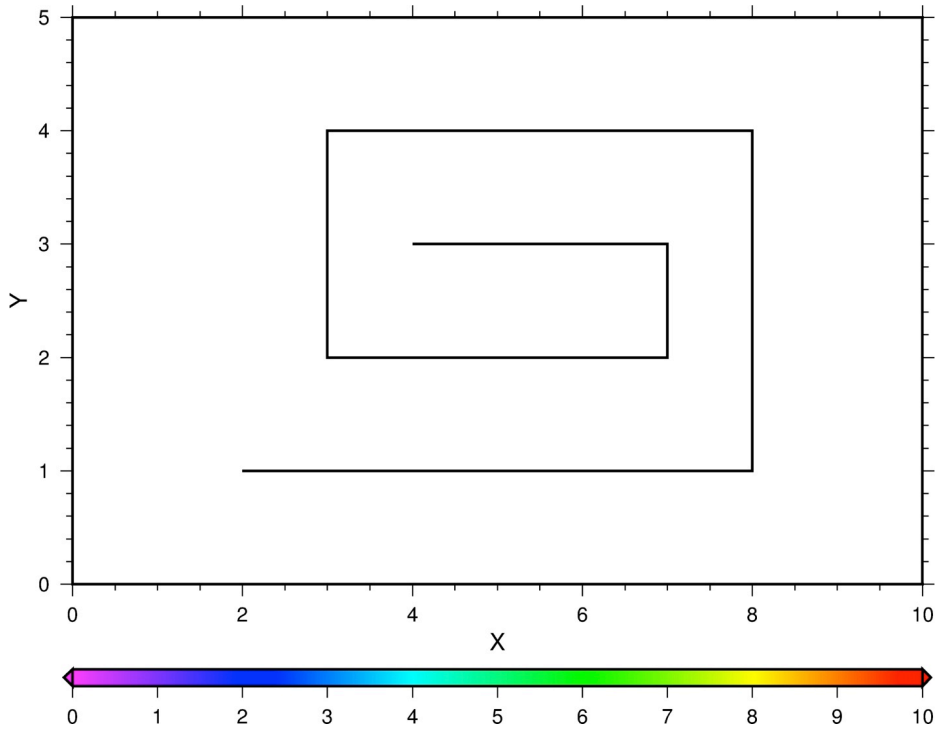


Es ist außerdem auch möglich, mit PSXY Linien zu plotten. Ein kleines Beispiel-Skript:

```
#!/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 >> $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
8 1
8 4
3 4
3 2
7 2
7 3
4 3
```

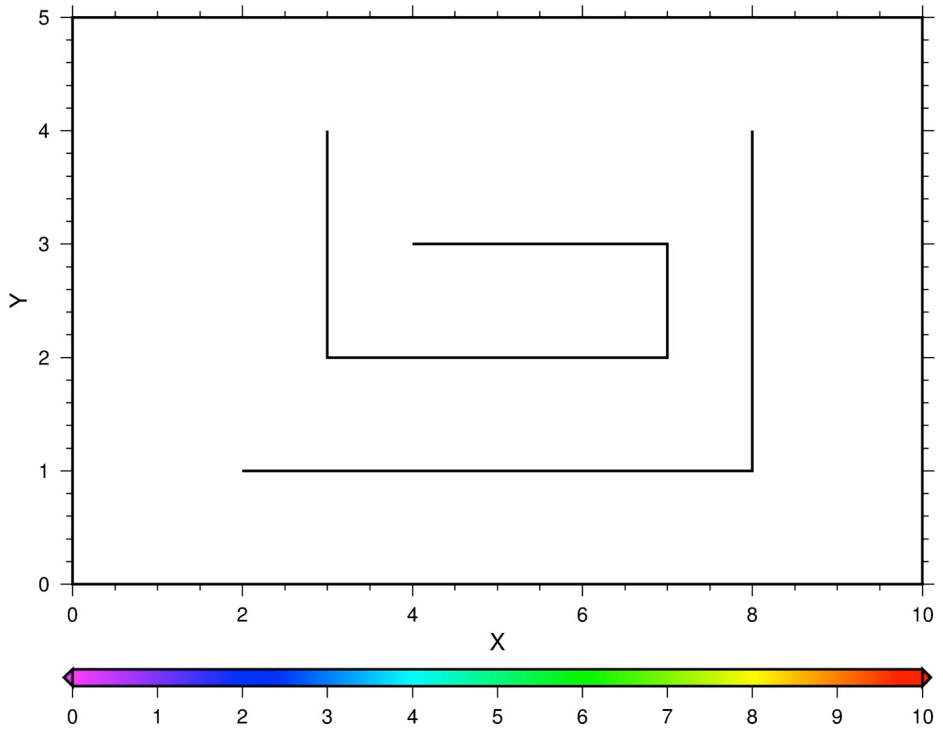


Möchte man mehrere Linien in einer Datei getrennt plotten, so hilft das –m Flag

```
#!/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 | awk '{if(NR==1 || NR==4)print ">";print $0}' | psxy -R -J -O -K -V -W5/0/0/0 -m">" >> $psfile
psscale -Ccolor.cpt -D7.5/-1.5/15/0.3h -Ba1 -O -E >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Wobei psxy aufgrund des AWK Kommandos folgenden Input bekommt:

```
>
2 1
8 1
8 4
>
3 4
3 2
7 2
7 3
4 3
```

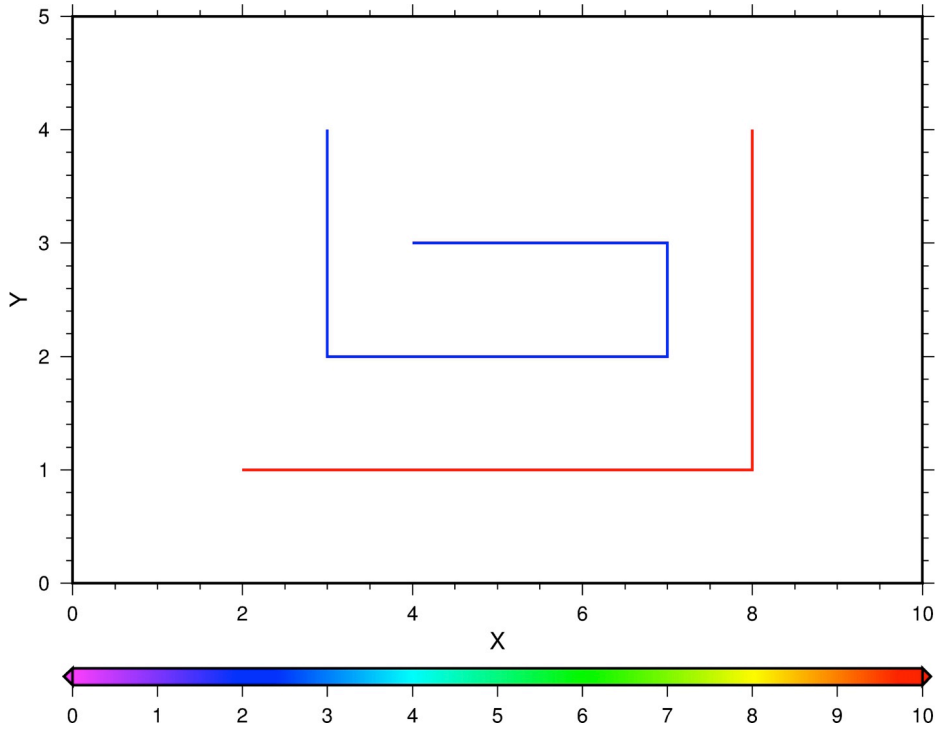


Möchte man mehrere Linien in einer Datei getrennt plotten, so hilft das –m Flag. Es ist auch möglich, nach dem Trennungs-zeichen Formate für die Liniensegmente anzugeben. So kann für jedes Segment Farbe und Stärke mit –W gesetzt werden

```
#!/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 | awk '{if(NR==1)print "> -W5/255/0/0";if(NR==4)print "> -W5/0/0/250";print $0}‘ \
| psxy -R -J -O -K -V -W5/0/0/0 -m">" >> $psfile
psscale -Ccolor.cpt -D7.5/-1.5/15/0.3h -Ba1 -O -E >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Wobei psxy aufgrund des AWK Kommandos folgenden Input bekommt:

```
> -W5/255/0/0
2 1
8 1
8 4
> -W5/0/0/250
3 4
3 2
7 2
7 3
4 3
```

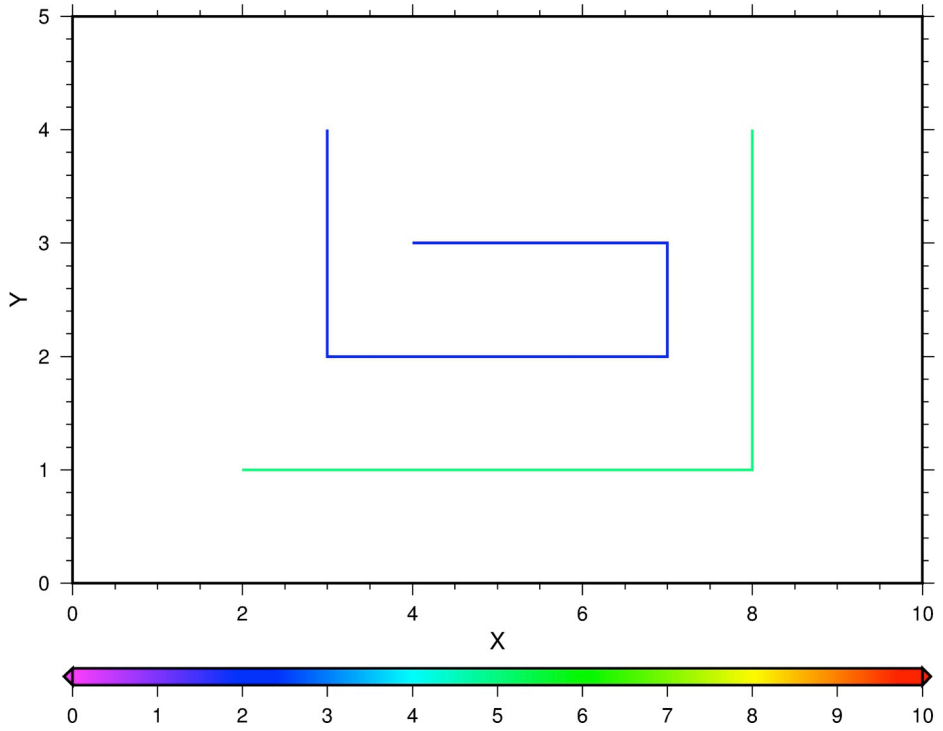


Möchte man mehrere Linien in einer Datei getrennt plotten, so hilft das –m Flag. Es ist auch möglich, nach dem Trennungs-zeichen Formate für die Liniensegmente anzugeben. Möchte man die Farbe in Abhängigkeit von einer Farbskala, dann –Z.

```
#!/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 | awk '{if(NR==1)print "> -Z5";if(NR==4)print "> -Z2";print $0}' \
| psxy -Ccolor.cpt -R -J -O -K -V -W5 -m">" >> $psfile
psscale -Ccolor.cpt -D7.5/-1.5/15/0.3h -Ba1 -O -E >> $psfile
ps2raster -A -E600 -Tj $psfile
```

Wobei psxy aufgrund des AWK Kommandos folgenden Input bekommt:

```
> -Z5
2 1
8 1
8 4
> -Z2
3 4
3 2
7 2
7 3
4 3
```



Zu Sinn und Zweck des Ganzen ein Beispiel. Oben ist ein Geschwindigkeitsmodell mit darüber geplotteten Triangulationsgrid zu sehen. Der Abstand der Knotenpunkte sollte gleich einer Wellenlänge sein (Wellenlänge=Geschwindigkeit/Frequenz). Für eine konstante Frequenz lässt sich das Geschwindigkeitsmodell direkt in ein Wellenlängenmodell transformieren. Die Verbindungsgerade zwischen den Knotenpunkten bekommt je nachdem ob der Abstand kleiner (weiß) oder größer (schwarz) einer Wellenlänge ist die entsprechende Farbe. Somit kann im Modell überprüft werden, ob es eine Systematik bei den zu kurzen oder zu langen Abständen gibt.