

GMT 中文手册 v5.4.5

GMT 中文社区

2019 年 02 月 18 日

保护环境，从阅读电子文档开始！

目 录

第 1 章 简介	21	2.4.1 ghostscript 的中文支持	38
1.1 GMT 简介	21	2.4.2 gsview 的中文支持	38
1.1.1 GMT 是什么	21	2.4.3 GMT 的中文支持	39
1.1.2 GMT 的历史	21	2.4.4 测试脚本	39
1.1.3 GMT 维护者	21	2.5 macOS 下安装 GMT	40
1.1.4 GMT 的特点	22	2.5.1 使用 homebrew 安装	40
1.1.5 其他制图软件	23	2.5.2 使用 GMT 安装包	41
1.2 GMT 版本	23	2.6 macOS 下的 GMT 中文支持	41
1.2.1 版本号	23	2.6.1 使 gs 支持中文	41
1.2.2 GMT 主流版本	23	2.6.2 使 GMT 支持中文	42
1.3 引用	24		
1.4 学习资源	25	第 3 章 GMT 初探	45
1.4.1 官方学习资源	25	3.1 命令行	45
1.4.2 社区学习资源	25	3.1.1 终端	45
1.4.3 其他学习资源	26	3.1.2 脚本	46
		3.1.3 脚本语言	46
第 2 章 安装	27	3.2 命令格式	46
2.1 Linux 下安装 GMT	27	3.3 绘制第一张图	47
2.1.1 解决依赖关系	27	3.3.1 图件分析与拆解	47
2.1.2 下载	28	3.3.2 绘制底图	48
2.1.3 安装 GMT	28	3.3.3 绘制震中和台站位置	49
2.1.4 修改环境变量	30	3.3.4 绘制射线路径	50
2.1.5 测试是否安装成功	31	3.3.5 添加台站名	52
2.2 Linux 下的 GMT 中文支持	31	3.3.6 图层的先后顺序	54
2.2.1 使 ghostscript 支持中文	31		
2.2.2 使 GMT 支持中文	33	第 4 章 绘图基础	57
2.2.3 对其他发行版的若干说明	34	4.1 画布	57
2.2.4 参考资料	36	4.1.1 画布尺寸	57
2.3 Windows 下安装 GMT	36	4.1.2 画布颜色	57
2.4 Windows 下的 GMT 中文支持	38	4.1.3 画布方向	57
		4.1.4 底图原点	58
		4.2 单位	59
		4.2.1 长度单位	59
		4.2.2 距离单位	60

4.2.3 小练习	60	4.10.2 线条平滑	78
4.3 颜色	61	4.10.3 端点箭头	78
4.3.1 颜色名	61	4.10.4 其他属性	79
4.3.2 RGB	61	4.11 锚点	79
4.3.3 HSV	62	4.12 绘制修饰物	80
4.3.4 CMYK	62	4.12.1 定位	81
4.3.5 灰色	62	4.12.2 背景面板	82
4.3.6 透明色	62	4.13 词汇表	83
4.3.7 颜色小结	63		
4.4 画笔	63	第 5 章 输入与输出	85
4.4.1 画笔线宽	63	5.1 数据格式	85
4.4.2 画笔颜色	64	5.1.1 地理坐标	85
4.4.3 画笔线型	64	5.1.2 绝对时间坐标	85
4.4.4 小练习	65	5.1.3 相对时间坐标	86
4.5 填充	65	5.1.4 一般坐标值	86
4.5.1 填充颜色	65	5.2 表数据	87
4.5.2 填充图案	66	5.2.1 ASCII 表	87
4.5.3 小练习	66	5.2.2 二进制表	89
4.6 文字	67	5.2.3 NetCDF 表	90
4.6.1 文字大小	67	5.3 NaN 数据	90
4.6.2 字体	67	5.4 兼容 OGR 的 GMT 矢量数	
4.6.3 填充色	68	据格式	91
4.6.4 描边	68	5.4.1 简介	91
4.6.5 小练习	69	5.4.2 OGR/GMT 格式	91
4.7 转义序列	69	5.4.3 OGR/GMT 元数据	92
4.7.1 转义字符	69	5.4.4 OGR/GMT 数据	93
4.7.2 小练习	70	5.4.5 示例	94
4.7.3 注意事项	71	5.5 PostScript	95
4.8 特殊字符	71	5.5.1 PS 是什么	95
4.8.1 八进制码指定特殊		5.5.2 PS 的优点	96
字符	71	5.5.3 PS 阅读器	96
4.8.2 特殊字体下的字符 .	73	5.5.4 将 PS 转换为其他格式 .	97
4.8.3 特殊字符小结 . . .	73		
4.8.4 小练习	74	第 6 章 网格文件	99
4.9 绘制矢量/箭头	74	6.1 网格文件简介	99
4.9.1 矢量分类	75	6.1.1 netCDF 网格文件	99
4.9.2 矢量箭头属性	75	6.1.2 Sun 光栅文件	99
4.9.3 矢量示例	77	6.1.3 Native 二进制格式	99
4.10 线条属性	77	6.1.4 用户自定义网格格式	100
4.10.1 端点偏移量	78	6.2 网格配准	100

6.2.1 网格线配准	101	8.3.3 地理底图	127
6.2.2 像素配准	101	8.3.4 笛卡尔线性轴	128
6.3 边界条件	101	8.3.5 笛卡尔 \log_{10} 轴	128
6.3.1 默认边界条件	102	8.3.6 笛卡尔指数轴	129
6.3.2 周期边界条件	102	8.3.7 时间轴	130
6.3.3 地理边界条件	102	8.3.8 自定义轴	132
6.4 读 netCDF 文件	102	8.4 -P 选项	134
6.4.1 多个二维变量的处理	102	8.5 -V 选项	135
6.4.2 三维变量的处理 . . .	103	8.6 -U 选项	135
6.4.3 四维变量的处理 . . .	104	8.7 -K 和 -O 选项	136
6.4.4 一维变量的处理 . . .	104	8.7.1 图层	136
6.4.5 修改坐标单位	104	8.7.2 PS 文件结构	137
6.5 写网格文件	105	8.7.3 -K 和 -O 的作用 . . .	137
6.5.1 网格文件格式	105	8.7.4 -K 和 -O 的使用 . . .	138
6.5.2 分块与压缩	107	8.8 -X 和 -Y 选项	139
6.6 网格文件后缀	108	8.9 -a 选项	140
6.7 查看 netCDF 文件	108	8.10 -b 选项	141
第 7 章 CPT 文件	109	8.11 -d 选项	142
7.1 CPT 简介	109	8.12 -e 选项	142
7.2 使用 CPT	109	8.13 -f 选项	142
7.3 CPT 文件格式	109	8.14 -g 选项	143
7.3.1 分类 CPT 文件 . . .	109	8.15 -h 选项	144
7.3.2 常规 CPT 文件 . . .	110	8.16 -i 和 -o 选项	144
7.3.3 周期性 CPT 文件 .	111	8.17 -n 选项	145
7.3.4 动态 CPT	112	8.18 -r 选项	145
7.4 调整 CPT	113	8.19 -p 选项	145
7.5 GMT 内置 CPT	114	8.20 -s 选项	146
7.6 其他 CPT	116	8.21 -t 选项	146
第 8 章 标准选项	117	8.22 -x 选项	147
8.1 -R 选项	118	8.23 -: 选项	147
8.1.1 四种方式	118		
8.1.2 五种坐标	119		
8.2 -J 选项	121		
8.2.1 GMT 投影代码 . . .	121		
8.2.2 Proj4 投影代码 . .	123		
8.3 -B 选项	124		
8.3.1 边框设置	125		
8.3.2 轴设置	126		
		第 9 章 投影方式	149
		9.1 -Jx: 笛卡尔变换	149
		9.1.1 笛卡尔线性坐标 . . .	149
		9.1.2 笛卡尔对数投影 . . .	152
		9.1.3 笛卡尔指数投影 . . .	152
		9.2 -Jp: 极坐标线性投影 . . .	153
		9.3 -Ja: Lambert 方位等面积投影	154
		9.3.1 矩形地图	154
		9.3.2 半球地图	155

9.3.3 震源辐射花样	156	10.4 FONT 参数	189
9.4 -Jb: Albers 圆锥等面积投影	156	10.5 FORMAT 参数	190
9.5 -Jc: Cassini 圆柱投影	157	10.6 IO 参数	193
9.6 -Jcyl_stere: 圆柱立体投影	158	10.7 MAP 参数	194
9.7 -Jd: 等距圆锥投影	159	10.8 PROJ 参数	197
9.8 -Je: 方位等距投影	160	10.9 PS 参数	198
9.9 -Jf: 球心方位投影	161	10.10 TIME 参数	201
9.10 -Jg: 正交投影	162	10.11 其他参数	202
9.11 -Jh: 等面积 Hammer 投影	164		
9.12 -Ji: 正弦曲线投影	165	第 11 章 地学数据集	205
9.13 -Jj: Miller 圆柱投影	166	11.1 地学数据集	205
9.14 -Jk: Eckert 投影	167	11.1.1 数据库原理	205
9.15 -Jl: Lambert 圆锥保角投影	168	11.1.2 建立 GMT 数据库 .	205
9.16 -Jm: Mercator 投影	169	11.2 GSHHG: 全球高精度海岸	
9.17 -Jn: Robinson 投影	170	线数据	206
9.18 -Jo: 倾斜 Mercator 投影	171	11.2.1 数据精度	206
9.19 -Jpoly: 多圆锥投影	172	11.2.2 数据内容	206
9.20 -Jq: 圆柱等距投影	173	11.2.3 使用示例	208
9.21 -Jr: Winkel Tripel 投影	174	11.2.4 备注	210
9.22 -Js: 立体等角投影	175	11.3 DCW: 世界数字图表	210
9.22.1 极区立体地图	175	11.3.1 区域代码	211
9.22.2 矩形立体地图	176	11.3.2 使用说明	212
9.22.3 一般立体地图	176	11.3.3 使用示例	213
9.23 -Jt: 横向 Mercator 投影	177	11.3.4 备注	215
9.24 -Ju: 通用横向 Mercator(UTM) 投影	179	11.4 earth_relief: 全球地形起伏	
9.25 -Jv: Van der Grinten 投影	180	数据	215
9.26 -Jw: Mollweide 投影	181	11.4.1 地形起伏数据简介 .	215
9.27 -Jy: 圆柱等面积投影	182	11.4.2 使用方法	216
第 10 章 配置 GMT	185	11.4.3 缓存空间问题	216
10.1 GMT 配置简介	185	11.4.4 技术细节	216
10.1.1 简介	185	11.4.5 数据来源及引用 . . .	217
10.1.2 设计思路	185	11.5 GADM: 全球行政区划数据库	217
10.1.3 修改 GMT 配置	185	11.5.1 数据下载	217
10.1.4 查看配置参数的值	187	11.5.2 数据格式及转换 . . .	218
10.1.5 删除配置文件	187	11.5.3 数据分级	219
10.1.6 GMT 配置示例	187	11.5.4 使用示例	219
10.2 COLOR 参数	188	11.5.5 2 级行政区划/市界 .	221
10.3 DIR 参数	189	11.5.6 许可协议	222
		11.5.7 备注	222

第 12 章 模块手册	223
12.1 backtracker	226
12.2 blockmean	226
12.3 blockmedian	226
12.4 blockmode	227
12.5 dimfilter	227
12.6 filter1d	227
12.6.1 必选选项	227
12.6.2 选项	228
12.6.3 示例	228
12.7 fitcircle	228
12.7.1 必选选项	229
12.7.2 选项	229
12.7.3 示例	230
12.8 gmt	231
12.9 gmt-config	232
12.10 gmt2kml	233
12.11 gmt5syntax	233
12.11.1 用法	233
12.12 gmtconnect	233
12.12.1 选项	233
12.12.2 示例	234
12.13 gmtconvert	234
12.14 gmtdefaults	234
12.14.1 必选选项	234
12.14.2 可选选项	234
12.14.3 示例	235
12.14.4 BUGS	235
12.15 gmtflexure	235
12.16 gmtget	235
12.16.1 必选选项	235
12.16.2 可选选项	235
12.17 gmtgravmag3d	236
12.18 gmtinfo	236
12.18.1 最小示例	236
12.18.2 可选选项	236
12.19 gmtlogo	237
12.19.1 必选选项	238
12.19.2 可选选项	238
12.19.3 示例	238
12.20 gmtmath	238
12.21 gmtpmodeler	238
12.22 gmtregress	239
12.23 gmtselect	239
12.23.1 七个空间准则	239
12.23.2 其他选项	241
12.23.3 示例	241
12.24 gmtset	242
12.24.1 必选选项	242
12.24.2 可选选项	243
12.24.3 FAQ	243
12.25 gmt simplify	243
12.25.1 选项	244
12.25.2 示例	244
12.25.3 参考文献	244
12.26 gmt spatial	244
12.27 gmt switch	244
12.28 gmt vector	244
12.29 gmt which	244
12.29.1 必选选项	244
12.29.2 可选选项	245
12.30 gpsgriddler	245
12.31 gravfft	245
12.32 grd2cpt	245
12.33 grd2rgb	245
12.33.1 必选选项	245
12.33.2 可选选项	246
12.33.3 示例	246
12.34 grd2xyz	246
12.34.1 必选选项	247
12.34.2 可选选项	247
12.34.3 示例	248
12.35 grdblend	248
12.35.1 必选选项	248
12.35.2 可选选项	249
12.35.3 示例	249
12.36 grdclip	250
12.36.1 必选选项	250
12.36.2 可选选项	250
12.36.3 示例	250

12.37	grdcontour	250	12.53.2 可选选项	263	
12.38	grdcut	250	12.53.3 示例	263	
12.38.1	必选选项	251	12.54	grdraster	263
12.38.2	可选选项	251	12.54.1	grdraster.info . . .	264
12.38.3	示例	252	12.54.2	必选选项	264
12.38.4	相关	252	12.54.3	可选选项	264
12.39	grdedit	252	12.54.4	示例	264
12.39.1	必选选项	252	12.55	grdredpol	264
12.39.2	可选选项	252	12.56	grdconvert	265
12.39.3	示例	254	12.56.1	必选选项	265
12.40	grdffft	254	12.56.2	示例	265
12.41	grdfilter	254	12.57	grdrotater	265
12.42	grdflexure	254	12.58	grdsample	265
12.43	grdgradient	254	12.58.1	必选选项	266
12.44	grdgravmag3d	254	12.58.2	可选选项	266
12.45	grdhisteq	254	12.58.3	示例	266
12.46	grdimage	254	12.59	grdseamount	267
12.46.1	必选选项	254	12.60	grdspotter	267
12.46.2	可选选项	254	12.61	grdtrack	267
12.46.3	示例	255	12.61.1	选项	267
12.47	grdinfo	255	12.61.2	相关	267
12.47.1	必选选项	256	12.62	grdtrend	267
12.47.2	可选选项	256	12.62.1	必选选项	267
12.48	grdlandmask	258	12.62.2	可选选项	268
12.48.1	必选选项	258	12.62.3	示例	268
12.48.2	可选选项	259	12.63	grdvector	268
12.48.3	注意事项	260	12.63.1	必选选项	268
12.48.4	示例	260	12.63.2	可选选项	268
12.49	grdmask	260	12.63.3	示例	269
12.49.1	必选选项	260	12.64	grdview	269
12.49.2	可选选项	261	12.65	grdvolume	269
12.49.3	注意事项	261	12.65.1	必选选项	270
12.49.4	示例	261	12.65.2	可选选项	270
12.50	grdmath	262	12.65.3	示例	270
12.51	grdpaste	262	12.65.4	引用	271
12.51.1	必选选项	262	12.66	greenspline	271
12.51.2	示例	262	12.67	gshhg	271
12.52	grdpmodeler	262	12.68	hotspotter	271
12.53	grdproject	262	12.69	img2google	271
12.53.1	必选选项	263	12.70	img2 grd	271

12.71 isogmt	271	12.89.1 必选选项	285
12.71.1 用法	271	12.89.2 可选选项	285
12.71.2 说明	271	12.89.3 示例	289
12.71.3 示例	272	12.89.4 FAQ	290
12.71.4 已知 BUGS	272	12.90 pscontour	290
12.72 kml2gmt	272	12.91 psconvert	290
12.72.1 必选选项	272	12.91.1 必选选项	291
12.72.2 可选选项	272	12.91.2 可选选项	291
12.72.3 示例	273	12.91.3 其他	294
12.73 makecpt	273	12.91.4 FAQ	294
12.73.1 选项	273	12.91.5 BUGS	294
12.74 mapproject	273	12.92 pscoupe	294
12.74.1 最小示例	274	12.92.1 剖面类型	295
12.74.2 -I 选项	274	12.92.2 选项	295
12.74.3 -Q 选项	274	12.92.3 示例	296
12.75 mgd77convert	274	12.93 pshistogram	300
12.76 mgd77header	274	12.93.1 必选选项	301
12.77 mgd77info	274	12.93.2 可选选项	301
12.78 mgd77list	274	12.94 psimage	302
12.79 mgd77magref	275	12.94.1 必选选项	302
12.80 mgd77manage	275	12.94.2 可选选项	303
12.81 mgd77path	275	12.94.3 示例	303
12.82 mgd77sniffer	275	12.95 pslegend	304
12.83 mgd77track	275	12.95.1 选项	304
12.84 nearneighbor	275	12.95.2 图例文件格式	304
12.85 originator	275	12.95.3 示例	306
12.86 project	275	12.96 psmask	310
12.86.1 必选选项	276	12.97 psmeca	310
12.86.2 可选选项	276	12.97.1 -S 选项	310
12.86.3 示例	276	12.97.2 选项	312
12.87 psbasemap	277	12.97.3 示例	313
12.87.1 必选选项	277	12.98 pspolar	317
12.87.2 可选选项	279	12.98.1 选项	317
12.87.3 示例	280	12.98.2 示例	318
12.87.4 BUGS	283	12.99 psrose	321
12.88 psclip	283	12.99.1 可选选项	321
12.88.1 必选选项	284	12.100 pssac	322
12.88.2 可选选项	284	12.100.1 必选选项	323
12.88.3 示例	284	12.100.2 可选选项	323
12.89 pscoast	284	12.100.3 示例	324

12.101 psscale	325	12.118 sphinterpolate	360
12.101.1 必选选项	325	12.119 sphtriangulate	360
12.101.2 选项	325	12.120 splitxyz	360
12.101.3 示例	326	12.121 surface	360
12.102 pssegy	326	12.122 talwani2d	360
12.103 pssegzy	327	12.123 talwani3d	360
12.104 pssolar	327	12.124 trend1d	360
12.104.1 必选选项	327	12.125 trend2d	360
12.104.2 可选选项	327	12.126 triangulate	360
12.104.3 示例	329	12.127 x2sys_binlist	361
12.104.4 BUGS	329	12.128 x2sys_cross	361
12.105 psternary	329	12.129 x2sys_datalist	361
12.105.1 可选选项	329	12.130 x2sys_get	361
12.106 pstext	329	12.131 x2sys_init	361
12.106.1 必选选项	330	12.132 x2sys_list	361
12.106.2 可选选项	330	12.133 x2sys_merge	361
12.106.3 示例	335	12.134 x2sys_put	361
12.107 psvelo	335	12.135 x2sys_report	361
12.107.1 -S 选项	335	12.136 x2sys_solve	361
12.107.2 选项	336	12.137 xyz2grd	361
12.107.3 示例	337	12.137.1 必选选项	362
12.108 pswiggle	340	12.137.2 可选选项	362
12.109 psxy	340	12.137.3 示例	364
12.109.1 可选选项	340	第 13 章 绘图示例	365
12.109.2 -S 选项	344	13.1 绘制底图	365
12.109.3 输入数据格式 . . .	352	13.1.1 线性坐标系	365
12.109.4 多段数据	353	13.1.2 对数坐标系	365
12.109.5 示例	353	13.1.3 区域地图	366
12.110 psxyz	356	13.1.4 全球地图	368
12.111 rotconverter	356	13.2 绘制双 Y 图	368
12.112 rotsmoothen	357	13.3 绘制间断 Y 轴	369
12.113 sample1d	357	13.4 绘制线条	369
12.113.1 可选选项	357	13.5 绘制符号	369
12.113.2 示例	357	13.6 绘制海岸线	369
12.114 segy2grd	358	13.7 绘制震源球	369
12.115 spectrum1d	358	13.8 绘制图例	369
12.115.1 选项	359	13.9 绘制地形起伏图	369
12.115.2 示例	359	13.10 绘制直方图	371
12.116 sph2grd	360	13.11 绘制图中图	371
12.117 sphdistance	360		

13.12 绘制 SAC 格式地震波形图	371	15.2 命令行及脚本基础	391
13.13 绘制三角图解	371	15.2.1 命令行基础	391
13.14 绘制等值线图	371	15.2.2 bash 简明教程	395
13.15 多子图	371	15.2.3 bat 简明教程	397
第 14 章 API	373	15.2.4 Unix 常用工具	399
14.1 GMT C API	373	15.3 GMT 颜色表	400
14.2 GMT Matlab API	373	15.4 GMT 预定义填充图案	400
14.2.1 简介	373	15.5 GMT 底层原理	402
14.2.2 安装	373	15.5.1 命令行历史	402
14.2.3 使用方法	375	15.6 目录结构	402
14.2.4 常见问题	376	15.6.1 bin 目录	403
14.2.5 附录	376	15.6.2 include 目录	403
14.3 GMT Julia API	378	15.6.3 lib 目录	403
14.3.1 简介	378	15.6.4 share 目录	404
14.3.2 安装	378	15.7 三种距离计算方式	404
14.3.3 使用	378	15.7.1 Flat Earth 距离	404
14.3.4 更多示例	381	15.7.2 大圆路径距离	405
14.3.5 附录	381	15.7.3 测地距离	405
第 15 章 附录	385	15.7.4 总结	405
15.1 风格指南	385	15.8 环境变量	405
15.1.1 使用脚本来执行		15.8.1 \$GMT_SHAREDIR	405
GMT 命令	385	15.8.2 \$GMT_DATADIR	406
15.1.2 不要跨平台写脚本	385	15.8.3 \$GMT_USERDIR	406
15.1.3 使用变量	385	15.8.4 \$GMT_TMPDIR	406
15.1.4 不要省略参数	386	15.8.5 \$GMT_CACHEDIR	406
15.1.5 开始与结束	387	15.9 自定义字体	406
15.1.6 命令中选项的顺序	388	15.10 获取 GMT 开发版	407
15.1.7 使用 SI 单位制	388	15.11 用 GMT 制作动画	407
15.1.8 不要依赖于 GMT		15.12 目录参数	408
的系统设置	388	15.13 等值线标注和“线条标注”	409
15.1.9 -P 选项的使用	389	15.13.1 标注的位置	409
15.1.10 不要滥用-B 选项	390	15.13.2 标注的属性	410
15.1.11 verbose 模式	390	15.13.3 等值线标注位置实例	412
15.1.12 慎用-X 和-Y	391	15.13.4 标注属性实例	415
15.1.13 网格文件后缀	391	15.13.5 综合实例	416
		15.14 通过 GDAL 读写网格文件	418

保护环境，从阅读电子文档开始！

图目录

1.1 GMT 的 5 个主要维护者	22
4.1 ISOLation1+ 和 Standard+ 编码下的八进制码	72
4.2 Symbol 和 Pifont 字体八进制码	73
4.3 GMT 中的三种矢量箭头	75
4.4 GMT 修饰物背景面板	83
6.1 GMT 网格配准方式	101
8.1 -R 选项指定数据范围	119
8.2 地理底图示例 1	128
8.3 地理底图示例 2	128
8.4 笛卡尔线性轴	128
8.5 对数坐标轴	129
8.6 指数投影坐标轴	129
8.7 时间轴示例 1	131
8.8 时间轴示例 2	131
8.9 时间示例 3	131
8.10 时间轴示例 4	131
8.11 时间轴示例 5	132
8.12 时间轴示例 6	132
8.13 时间轴示例 7	132
8.14 自定义坐标轴	134
8.15 -U 选项加时间截	136
8.16 -K 和 -O 选项的原理	137
9.1 笛卡尔坐标的线性变换	150
9.2 地理坐标的线性变换	151
9.3 日期时间坐标的线性变换	152
9.4 对数投影	152
9.5 指数变换	153
9.6 极坐标用法示例	154
9.7 使用 Lambert 方位等面积投影绘制矩形地图	155
9.8 使用 Lambert 方位等面积投影绘制半球地图	155
9.9 Albers 圆锥等面积投影	157

9.10 Cassini 投影绘制 Sardinia 岛	158
9.11 使用 Gall 立体投影绘制世界地图	159
9.12 等距圆锥地图投影	160
9.13 使用等距方位投影绘制全球图	161
9.14 球心方位投影	162
9.15 使用正交投影绘制半球	163
9.16 透视投影	164
9.17 使用 Hammer 投影绘制全球地图	165
9.18 使用正弦曲线投影绘制世界地图	165
9.19 使用间断正弦曲线投影绘制世界地图	166
9.20 使用 Miller 圆柱投影绘制世界地图	167
9.21 Eckert IV 投影绘制全球图	168
9.22 Eckert VI 投影绘制全球图	168
9.23 Lambert 保角圆锥投影	169
9.24 Mercator 投影	170
9.25 使用 Robinson 投影绘制全球地图	171
9.26 使用 -Joc 倾斜 Mercator 投影	172
9.27 使用 -J0a 倾斜 Mercator 投影	172
9.28 多圆锥投影	173
9.29 使用 Plate Carrée 投影绘制全球地图	174
9.30 使用 Winkel Tripel 投影绘制全球地图	175
9.31 极区立体保角投影	176
9.32 矩形边界下的极区立体保角投影	176
9.33 一般立体投影	177
9.34 矩形横向 Mercator 地图	178
9.35 全球横向 Mercator 地图	179
9.36 使用 Van der Grinten 投影绘制全球图	181
9.37 使用 Mollweide 投影绘制全球地图	182
9.38 使用 Behrman 圆柱等面积投影绘制地图	183
12.1 fitcircle 示例	231
12.2 方向玫瑰图	279
12.3 用 psbasemap 命令-D 选项绘制图中图	282
12.4 磁场玫瑰图	283
12.5 pscoupe 示例图 1	298
12.6 pscoupe 示例图 2	300
12.7 pslegend 示例图 1	307
12.8 pslegend 示例图 2	310
12.9 psmeca 示例图 1	314
12.10 psmeca 示例图 2	315

12.11 psmeca 示例图 3	317
12.12 pspolar 示例	320
12.13 曙暮光区的多种定义 (图片来自于 https://en.wikipedia.org/wiki/Twilight)	328
12.14 段落模式示意图	334
12.15 psvelo 示例图 1	338
12.16 psvelo 示例图 2	339
12.17 psxy -A 选项示意图	341
12.18 psxy -S 选项示意图	345
12.19 psxy -Sf 示意图	348
12.20 psxy -Sm 示意图	349
12.21 psxy -Sw 示意图。	352
12.22 psxy -F 选项示意图	355
12.23 psxy -L 选项示意图	356
13.1 线性 X-Y 图	365
13.2 对数 X-Y 图	366
13.3 区域地图	367
13.4 全球地图	368
14.1 Matlab PATH 设置	374
15.1 GMT 内置位图图案	401
15.2 通过指定 -Gd 选项的参数, 确定了标注的位置 (等值线上相距 1.5 英寸的点)	412
15.3 通过指定 -Gn 选项的参数, 确定了标注的位置 (每条长度超过 1 英寸的等值线的中心位置)	413
15.4 通过指定 -Gf 选项的参数, 确定了标注的位置 (等值线上与给定点距离最小的点)	413
15.5 通过指定 -GL 选项的参数确定了标注的位置 (大圆弧与等值线的交点)	414
15.6 通过指定 -GX 选项的参数 (多段数据文件 <i>cross.txt</i>), 确定了标注的位置	414
15.7 通过指定 -Sq 选项的参数控制标注属性.	415
15.8 另一个标注属性实例	416
15.9 标注的位置和内容来自不同的数据集	416
15.10 Canary Islands 到大西洋沿岸的海啸走时图, 特别是纽约。当发生灾难性滑坡时, 纽约将在 8 小时后遭遇大海啸。	418

保护环境，从阅读电子文档开始！

表目录

4.1	GMT 预定义画笔宽度名	63
4.2	Word 字号与 GMT 中字号对应关系	67
4.3	GMT 转义字符	69
4.4	欧洲特殊字母	70
6.1	GMT 自定义二进制网格文件结构	100
6.2	网格文件格式 ID	105
8.1	GMT 标准选项列表	117
8.2	GMT 投影代码	122
8.3	Proj4 投影代码	123
8.4	GMT 时间单位	130
10.1	GMT 预定义纸张大小	200
15.1	通配符	395

保护环境，从阅读电子文档开始！

欢迎来到 Generic Mapping Tools 的世界。

本文档是由 GMT 中文社区维护整理的 GMT 中文手册，其详尽地介绍了 GMT 的方方面面。其可以作为初学者的入门读物，也可以作为日常使用的参考书。

GMT 中文手册开始于 2014 年。在 2014 年，网络上能够找到的 GMT 中文文档大多基于 GMT4 甚至 GMT3。这些文档存在一些共同的缺点：

1. 排版质量差：大多是在 Word 里写好然后转换成 PDF
2. 缺乏维护：文档写好发布后就没有再持续更新
3. 不完整：很多有用的内容没有包含在文档中
4. 不严谨：很多文档声称是基于 GMT4，但实际上却在混用 GMT3 的语法
5. 命令过时：有些命令在新版本的 GMT 下无法执行，新用户的体验很糟糕
6. 重复劳动：多位作者独立翻译整理出了几份相似的文档，很多工作都是在重复劳动

相比于之前的文档，本文档具有如下优点：

1. 高质量排版：文档提供网页版和 PDF 版，PDF 由 LaTeX 制作得到，排版质量高
2. 持续维护：适配 GMT 的新版本，且不断修正文档中存在的错误，不定期发布新版本
3. 完整性：尽可能覆盖 GMT 的方方面面，供用户查找
4. 严谨性：尽可能保证所有命令均通过 GMT 最新版本的测试
5. 开源：原始文档为纯文本，在 GitHub 上开源
6. 协同合作：依托强大的 GitHub 和 Git，任何人均可参与到文档的编辑与维护中

文档的持续维护更新非一人之力所能完成，希望能够有更多的志愿者加入到文档的维护工作中来。

保护环境，从阅读电子文档开始！

第 1 章 简介

1.1 GMT 简介

1.1.1 GMT 是什么

GMT, 全称 Generic Mapping Tools, 中文一般译为“通用制图工具”, 是地球科学最广泛使用的制图软件。GMT 具有强大的绘图功能和数据处理功能。

绘图方面, GMT 支持绘制多种类型的底图, 包括 30 多种地图投影、笛卡尔线性坐标轴、对数轴、指数轴、极坐标系, 支持绘制统计直方图、等值线图、2D 网格图以及 3D 视角图等, 也支持绘制线段、海岸线、国界、多种符号、图例、色标、文字等。

数据处理方面, GMT 具有数据筛选、重采样、时间序列滤波、二维网格滤波、多项式拟合、线性回归分析等功能。

1.1.2 GMT 的历史

- 1988 年, Paul Wessel 和 Walter H.F. Smith 开发了 GMT 的最原始版本 GMT 1.0;
- 1991 年 8 月 10 日, GMT 2.0 发布;
- 1998 年 11 月 8 日, GMT 3.x 的第一个正式版发布;
- 2005 年 10 月 1 日, GMT 4.x 的第一个正式版发布; GMT4.x 系列的最后一个版本是 GMT 4.5.18, 发布于 2018 年 7 月 1 日;
- 2013 年 11 月 5 日, GMT 5.x 的第一个正式版发布; 目前最新版本 GMT 5.4.5 发布于 2019-01-04;

1.1.3 GMT 维护者

GMT 源码目前主要由 [Paul Wessel](#) 和 [Joaquim Luis](#) 维护。其源码公开在 [GitHub](#) 上, 任何用户均可贡献自己的代码。



图 1.1: GMT 的 5 个主要维护者

从左至右依次为 Joaquim Luis、Walter H.F. Smith、Remko Scharroo、Florian Wobbe 和 Paul Wessel。照片摄于 2016 年 8 月 15 日至 19 日在加州 La Jolla 举办的 GMT 开发者峰会。

1.1.4 GMT 的特点

为什么选择 GMT 作为绘图软件呢？因为 GMT 有如下特点：

1. 开源免费

GMT 是开源软件，其源码遵循 [LGPL](#) 协议。任何人均可自由复制、分发、修改其源代码，也可用于盈利。修改后的代码必须开源但可以使用其他开源协议。

2. 跨平台

GMT 的源码采用了高度可移植的 ANSI C 语言，其完全兼容于 POSIX 标准，可以运行在 Linux、macOS、Windows 及其他类 Unix 系统上。GMT 不公开了软件源代码，还提供了 Windows 和 macOS 下的安装包。各大 Linux 发行版中也提供了预编译的二进制包。

3. 模块化

GMT 遵循 UNIX 的模块化设计思想，将不同的绘图功能和数据处理功能划分到不同的模块中。这样的模块化设计有很多优点：

- 只需要少量的模块
- 各个模块之间相互独立且代码量少，易于更新和维护
- 每一步均独立于之前的步骤以及具体的数据类型，因而可以用于不同的应用中
- 可以在 shell 脚本中调用一系列程序，或通过管道连接起来，进而绘制复杂图片

4. 支持多种格式的高精度矢量图和位图

GMT 支持多种高精度的矢量图片格式和位图图片格式。矢量图片格式，如 PDF、PS、EPS 和 SVG，具有任意放大缩小而不失真的特性，可直接投稿到学术期刊；位图图片格式，如 BMP、JPG、PNG、PPM 和 TIFF 格式，可用于日常的文档及演示。

1.1.5 其他制图软件

除了 GMT 之外，还有很多其他的软件也可以实现各种制图功能。以下仅列出一些地学常用的制图软件。其中 √ 和 X 用于表示是否支持某一功能。

软件名称	二维图	三维图	地图
GMT	√	√ ¹	√
Matplotlib	√	√	√ ²
Microsoft Excel	√	√	√
Origin	√	√	X
Matlab	√	√	√ ³
ggplot2	√	X	√ ⁴
gnuplot	√	√	X

1.2 GMT 版本

1.2.1 版本号

GMT 的版本号遵循大多数开源软件使用的版本号规则。其版本号格式为:

major.minor.patch

其中 `major` 为主版本号, `minor` 为次版本号, `patch` 为补丁版本号。

一般而言, 版本号的更新规则如下:

- 当有极大的更新, 会增加主版本号 `major`, 因而 `major` 不同的两个版本在语法和功能上有很大的差异
- 当有较大的更新, 比如个别命令的语法有变化, 会更新次版本号 `minor`
- 若更新主要是修复错误, 则会增加 `patch` 的版本号

因而, `GMT 5.x.x` 与 `GMT 4.x.x` 之间有很大差异, 两个版本的语法是不完全兼容的, 切勿混在一起使用。`GMT 4.5.x` 和 `GMT 4.4.x`, 部分命令的语法和用法可能有一点区别。而 `GMT 4.5.13` 相对于 `GMT 4.5.12`, 主要是修复了一些 BUG。

1.2.2 GMT 主流版本

GMT 目前的主流版本有 `GMT4`、`GMT5` 和 `GMT6` 三个主版本。这几个版本有什么区别呢? 用户该如何选择呢?

¹ GMT 对三维图的支持很有限。

² 需要额外安装 `Cartopy` 或 `Basemap`

³ 需要额外安装 `M_Map`

⁴ 需要额外安装 `ggmap`

GMT4

优缺点:

- 已发布十年有余, 功能相对稳定, BUG 相对较少
- 最终版本为 4.5.18, 且不再增加新的功能也不再更新

GMT5

优缺点:

- 相对于 GMT4 有很多改进, 命令语法更统一, 选项的设计更加合理
- 增加了很多新功能, 其中非常有用的包括但不限于:
 - `-Bafg` 自动确定坐标轴的标注间隔
 - 支持透明色, 且支持透明图层
 - 引入修饰物的概念, 使得相关代码更具通用性
 - `-X` 和 `-Y` 支持多种指定坐标原点的方式, 画多子图的组合图时更加简单

GMT6

GMT6 的优缺点:

- 尚未正式发布, 新增的功能尚不稳定
- 几乎完全兼容 GMT5 语法
- 新增了现代模式, 大大简化了绘图脚本并且可以避免初学者常犯的错误
- 新增了 `movie` 模块, 可以更方便地制作动画

建议

综上, 对于用户的建议是:

- 新用户推荐学习 GMT5。
- 实验室可能有前人留下的 GMT4 的代码, 此时可以同时安装 GMT4 和 GMT5, 二者不冲突, 老脚本使用 GMT4 语法, 新脚本使用 GMT5 语法。

1.3 引用

若你发表的文章中包含了利用 GMT 制作的图件或利用 GMT 数据处理功能得到的结果, 可以考虑引用 GMT 的如下文章来回报 GMT 开发者:

GMT5 用户: Wessel, P., W. H. F. Smith, R. Scharroo, J. Luis, and F. Wobbe, Generic Mapping Tools: Improved Version Released, *EOS Trans. AGU*, 94(45), p. 409-410, 2013. doi:[10.1002/2013EO450001](https://doi.org/10.1002/2013EO450001).

GMT4/GMT3 用户: Wessel, P., and W. H. F. Smith, New, improved version of Generic Mapping Tools released, *EOS Trans. AGU*, 79(47), p. 579, 1998. doi:[10.1029/98EO00426](https://doi.org/10.1029/98EO00426).

GMT3.0 用户: Wessel, P., and W. H. F. Smith, New version of the Generic Map-

ping Tools released, *EOS Trans. AGU*, 76(33), 329, 1995. doi:10.1029/95EO00198.

GMT2 用户: Wessel, P., and W. H. F. Smith, Free software helps map and display data, *EOS Trans. AGU*, 72(41), 445-446, 1991. doi:10.1029/90EO00319.

GMT/MATLAB 工具箱用户: Wessel, P., and J. F. Luis The GMT/MATLAB Toolbox, *Geochem. Geophys. Geosyst.*, 18, 811-823, 2017. doi:10.1002/2016GC006723.

GMT 的某些模块基于 GMT 团队单独发展并发表的算法。算法相关文章包括：

dimfilter 模块 Kim, S.-S., and P. Wessel, Directional median filtering for regional-residual separation of bathymetry, *Geochem. Geophys. Geosyst.*, 9, Q03005, 2008. doi:10.1029/2007GC001850.

grdredpol 模块 Luis, J. F. and J. M. Miranda, Reevaluation of magnetic chronos in the North Atlantic between 35°N and 47°N: Implications for the formation of the Azores Triple Junction and associated plateau, *J. Geophys. Res.*, 113, B10105, 2008. doi:10.1029/2007JB005573.

surface 模块 Smith, W. H. F., and P. Wessel, Gridding with continuous curvature splines in tension, *Geophysics*, 55(3), 293–305, 1990. doi:10.1190/1.1442837.

x2sys 相关模块 Wessel, P., Tools for analyzing intersecting tracks: The x2sys package, *Computers & Geosciences*, 36, 348–354, 2010. doi:10.1016/j.cageo.2009.05.009.

greenspline 模块 Wessel, P., A General-purpose Green's function-based interpolator, *Computers & Geosciences*, 35, 1247–1254, 2009. doi:10.1016/j.cageo.2008.08.012.

Wessel, P. and J. M. Becker, Interpolation using a generalized Green's function for a spherical surface spline in tension, *Geophys. J. Int.*, 174, 21–28, 2008. doi:10.1111/j.1365-246X.2008.03829.x.

1.4 学习资源

GMT 学习资源列举如下，建议以官方学习资源为准。

1.4.1 官方学习资源

1. GMT 官方入门教程
2. GMT 官方参考手册
3. GMT 官方模块手册
4. GMT 官方示例
5. GMT 官方论坛
6. GMT 开发版源码 中的 test 目录下的测试脚本

1.4.2 社区学习资源

1. GMT 中文手册
2. GMT 中文社区示例

3. [GMT 中文社区博客](#)
4. 地学 GMT 中文社区 QQ 群: 1 群 (218905582, 已满); 2 群 (791856541)

1.4.3 其他学习资源

1. [SeisMan 博客](#)
2. [GMT Tutorial by Whyjay Zheng](#)

第 2 章 安装

2.1 Linux 下安装 GMT

2.1.1 解决依赖关系

GMT 在运行时依赖 fftw ($>=3.3$)、glib2 ($>=2.32$)、netCDF (>4.0 且支持 netCDF-4/HDF5)、ghostscript 等。GMT 在安装时主要依赖 GCC 编译器和 cmake ($>=2.8.5$)。因而，需要先安装 GMT 所依赖的软件包。

警告：由于 Linux 发行版众多，不同发行版下软件包的名称不同。因而，以下仅所列仅供参考，其他用户应自行根据关键词到 <https://pkgs.org> 上确认自己使用的发行版上软件包的具体名字。

对于 Ubuntu/Debian:

```
# 更新
$ sudo apt-get update

# 安装编译所需软件包
$ sudo apt-get install gcc g++ cmake make libc6

# 安装运行所需软件包
$ sudo apt-get install ghostscript
$ sudo apt-get install libnetcdf-dev
$ sudo apt-get install libgdal-dev python-gdal
$ sudo apt-get install liblapack3
$ sudo apt-get install libglib2.0-dev
$ sudo apt-get install libpcre3-dev
$ sudo apt-get install libfftw3-dev
```

对于 CentOS/RHEL/Fedora:

```
# CentOS 用户必须先安装 epel-release, RHEL/Fedora 用户无需安装
$ sudo yum install epel-release

# 安装编译所需软件包
$ sudo yum install gcc gcc-c++ cmake make glibc

# 安装运行所需软件包
$ sudo yum install ghostscript
$ sudo yum install netcdf-devel
```

(下页继续)

(续上页)

```
$ sudo yum install gdal-devel gdal-python  
$ sudo yum install lapack64-devel lapack-devel  
$ sudo yum install glib2-devel  
$ sudo yum install pcre-devel  
$ sudo yum install fftw-devel
```

安装完依赖包后，需要进一步确认 netCDF 是否支持 netCDF-4/HDF5 格式：

```
$ nc-config --has-nc4  
yes
```

若输出为 yes 则表示安装的 netCDF 支持 netCDF-4/HDF5 格式，则可继续安装 GMT，否则无法安装 GMT。

2.1.2 下载

Linux 下安装 GMT 需要下载如下三个文件（这里提供中科大提供的国内下载镜像）：

1. GMT 源码: <http://mirrors.ustc.edu.cn/gmt/gmt-5.4.5-src.tar.gz>
2. 全球海岸线数据 GSHHG: <http://mirrors.ustc.edu.cn/gmt/gshhg-gmt-2.3.7.tar.gz>
3. 全球数字图表 DCW: <http://mirrors.ustc.edu.cn/gmt/dcwgmt-1.1.4.tar.gz>

2.1.3 安装 GMT

将下载的三个压缩文件放在同一个目录里，按照如下步骤进行安装：

```
# 解压三个压缩文件  
$ tar -xvf gmt-5.4.5-src.tar.gz  
$ tar -xvf gshhg-gmt-2.3.7.tar.gz  
$ tar -xvf dcwgmt-1.1.4.tar.gz  
  
# 将 gshhg 和 dcw 数据复制到 gmt 的 share 目录下  
$ mv gshhg-gmt-2.3.7 gmt-5.4.5/share/gshhg  
$ mv dcwgmt-1.1.4 gmt-5.4.5/share/dcwgmt  
  
# 切换到 gmt 源码目录下  
$ cd gmt-5.4.5  
  
# 新建用户配置文件  
$ gedit cmake/ConfigUser.cmake
```

向 `cmake/ConfigUser.cmake` 文件中加入如下语句：

```
set (CMAKE_INSTALL_PREFIX "/opt/GMT-5.4.5")  
set (COPY_GSHHG TRUE)  
set (COPY_DCW TRUE)  
set (GMT_INSTALL_MODULE_LINKS FALSE)  
set (GMT_DATA_URL "http://mirrors.ustc.edu.cn/gmt/data/")  
set (GMT_USE_THREADS TRUE)
```

其中，

- `CMAKE_INSTALL_PREFIX` 用于设置 GMT 的安装路径, 上面的语句会将 GMT 安装在 `/opt/GMT-5.4.5` 目录下, 用户可以自行修改为其他路径。没有 root 权限的一般用户, 可以将安装路径设置为 `/home/xxx/software/GMT-5.4.5` 等有可读写权限的路径;
- `COPY_GSHHG` 和 `COPY_DCW` 设置为 `TRUE` 会将相关数据复制到 GMT 的 `share` 目录下
- `GMT_INSTALL_MODULE_LINKS` 为 `FALSE`, 表明不在 GMT 的 `bin` 目录下建立命令的软链接, 不建议设置为 `TRUE` (可选)
- `GMT_DATA_URL` 设置从中科大镜像下载 GMT 数据, 以加快数据下载速度 (可选)
- `GMT_USE_THREADS` 表示是否开启某些模块的并行功能 (可选)

警告: CentOS 6 用户需要将用户配置文件的最后一行改为 `set (GMT_USE_THREADS FALSE)`

小技巧: 此处为了便于一般用户理解, 只向 `cmake/ConfigUser.cmake` 中写入了必要的语句。

高级用户可以直接在 GMT 提供的模板配置文件的基础上进行更多配置。将 `cmake/ConfigUserTemplate.cmake` 复制为 `cmake/ConfigUser.cmake`, 然后根据配置文件中的大量注释说明信息自行修改配置文件。

继续执行如下命令以检查 GMT 的依赖是否满足:

```
# 注意, 此处新建的 build 文件夹位于 gmt-5.4.5 目录下, 不是 gmt-5.4.5/cmake 目录下
$ mkdir build
$ cd build/
$ cmake ..
```

`cmake ..` 会检查 GMT 对软件的依赖关系, 我的检查结果如下:

```
* Options:
* Found GSHHG database      : /home/user/GMT/gmt-5.4.5/share/gshhg (2.3.7)
* Found DCW-GMT database    : /home/user/GMT/gmt-5.4.5/share/dcwgmt
* NetCDF library            : /usr/lib64/libnetcdf.so
* NetCDF include dir        : /usr/include
* GDAL library              : /usr/lib64/libgdal.so
* GDAL include dir          : /usr/include/gdal
* FFTW library              : /usr/lib64/libfftw3f.so
* FFTW include dir          : /usr/include
* Accelerate Framework     :
* Regex support              : PCRE (/usr/lib64/libpcre.so)
* ZLIB library               : /usr/lib64/libz.so
* ZLIB include dir           : /usr/include
```

(下页继续)

(续上页)

```

* LAPACK library : yes
* License restriction : no
* Triangulation method : Shewchuk
* OpenMP support : enabled
* GLIB GTHREAD support : enabled
* PTHREAD support : enabled
* Build mode : shared
* Build GMT core : always [libgmt.so]
* Build PSL library : always [libpostscriptlight.so]
* Build GMT supplements : yes [supplements.so]
* Build GMT Developer : yes
* Build proto supplements : none
*
* Locations:
* Installing GMT in : /opt/GMT-5.4.5
* GMT_DATADIR : /opt/GMT-5.4.5/share
* GMT_DOCDIR : /opt/GMT-5.4.5/share/doc
* GMT_MANDIR : /opt/GMT-5.4.5/share/man
-- Configuring done
-- Generating done

```

正常情况下的检查结果应该与上面给出的类似。若出现问题，则需要检查之前的步骤是否有误，检查完成后删除原 build 目录再新建 build，继续执行 `cmake ..`，直到出现类似的检查结果。

警告： Anaconda 用户请注意！由于 Anaconda 中也安装了 FFTW、GDAL、netCDF 等库文件，GMT 在配置过程中通常会找到 Anaconda 提供的库文件，进而导致配置、编译或执行过程中出错。

解决办法是，在 `~/.bashrc` 中将 Anaconda 相关的环境变量注释掉，以保证 GMT 在配置和编译过程中找到的不是 Anaconda 提供的库文件。待 GMT 安装完成后，再将 Anaconda 相关环境变量改回即可。

检查完毕后，开始编译和安装：

```
$ make -j
$ sudo make -j install
```

注解： `-j` 选项可以实现并行编译以减少编译时间。但并行编译可能在个别发行版上无法使用。

2.1.4 修改环境变量

向 `~/.bashrc` 中加入如下语句以修改环境变量，并重启终端使其生效：

```
export GMT5HOME=/opt/GMT-5.4.5
export PATH=${GMT5HOME}/bin:$PATH
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}: ${GMT5HOME}/lib64
```

说明：

- 第一个命令添加了环境变量 `GMT5HOME`
- 第二个命令修改 `GMT5` 的 `bin` 目录加入到 `PATH` 中，使得终端可以找到 `GMT` 命令
- 第三个命令将 `GMT5` 的 `lib` 目录加入到动态链接库路径中。通常，32 位系统的路径为 `lib`, 64 位系统的路径为 `lib64`

2.1.5 测试是否安装成功

打开终端，键入如下命令，若正确显示 `GMT` 版本号，则表示安装成功：

```
$ gmt --version
5.4.5
```

2.2 Linux 下的 `GMT` 中文支持

`GMT` 原生并不支持中文。为了让 `GMT` 支持中文，需要首先修改 `ghostscript` 的配置文件使得其支持中文，再修改 `GMT` 的配置文件。

不同的 Linux 发行版下 `GMT` 中文支持的原理相同，但修改细节不同。这里先介绍 CentOS 7 上的修改方法，非 CentOS 7 用户可以先看 CentOS 7 的方法原理，再阅读文末关于其他发行版的进一步说明。

2.2.1 使 `ghostscript` 支持中文

ghostscript 中文配置文件

CentOS 7 下，`ghostscript` 的中文配置文件的路径为 `/usr/share/ghostscript/conf.d/cidfmap.zh_CN`。若该文件不存在，则表明系统中未安装 `ghostscript` 中文配置文件。

CentOS 7 下 `ghostscript` 简体中文配置文件可以通过如下命令安装：

```
$ sudo yum install ghostscript-chinese-zh_CN
```

ghostscript 中文配置文件

CentOS 7 中 `ghostscript` 中文配置文件的默认内容为：

```
/BousungEG-Light-GB <</FileType /TrueType /Path (/usr/share/fonts/wqy-zenhei/wqy-zenhei.
˓→ttc) /SubfontId 0 /CSI [(GB1) 4] >> ;
/GBZenKai-Medium <</FileType /TrueType /Path (/usr/share/fonts/wqy-zenhei/wqy-zenhei.
˓→ttc) /SubfontId 0 /CSI [(GB1) 4] >> ;
/MSungGBK-Light /BousungEG-Light-GB ;
```

(下页继续)

(续上页)

```
/Adobe-GB1      /BousungEG-Light-GB ;
```

其中的细节不管，其大致意义为：

- 第一行定义了字体名为 /BousungEG-Light-GB，对应的字体文件为 /usr/share/fonts/wqy-zenhei/wqy-zenhei.ttc，也就是文泉驿正黑；
- 第二行定义了字体名为 /GBZenKai-Medium，对应的字体文件也是文泉驿正黑；
- 第三行和第四行分别定义了字体名 /MSungGBK-Light 和 /Adobe-GB1，这两种都对应于 /BousungEG-Light-GB，相当于给字体定义了别名。

关于配置文件的几点说明：

- 字体名是任意的，比如字体名可以取为 /ABC；
- 字体文件似乎只能是 ttc 或 ttf 格式的，当然修改参数也有可能可以使用其他格式的字体；
- 要注意确认字体文件是否存在，比如 CentOS7 下的 wqy-zenhei.ttc 字体实际上位于软件包 wqy-zenhei-fonts 中。若字体不存在，则需要安装相应软件包。

添加 Windows 中文字体

Linux 的中文字体较少，所以这里使用 Windows 下中的中文字体，这里只考虑 Windows 下的宋体、仿宋、黑体和楷体四个基本字体。将这四个字体文件复制到 /usr/share/fonts/winfonts/ 目录下，然后对 ghostscript 的中文配置文件做如下修改：

```
% 原内容保持不变
/BousungEG-Light-GB <</FileType /TrueType /Path (/usr/share/fonts/wqy-zenhei/wqy-zenhei.
- ttc) /SubfontId 0 /CSI [(GB1) 4] >> ;
/GBZenKai-Medium    <</FileType /TrueType /Path (/usr/share/fonts/wqy-zenhei/wqy-zenhei.
- ttc) /SubfontId 0 /CSI [(GB1) 4] >> ;
/MSungGBK-Light     /BousungEG-Light-GB ;
/Adobe-GB1           /BousungEG-Light-GB ;

% 新增 Windows 字体的支持
/STSong-Light <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simsun.ttc) /
- SubfontId 0 /CSI [(GB1) 4] >> ;
/STFangsong-Light <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simfang.ttf) /
- SubfontId 0 /CSI [(GB1) 4] >> ;
/STHeiti-Regular <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simhei.ttf) /
- SubfontId 0 /CSI [(GB1) 4] >> ;
/STKaiti-Regular <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simkai.ttf) /
- SubfontId 0 /CSI [(GB1) 4] >> ;
```

这里仅以 Windows 下的常用四大字体为例。对于 Windows 下的其他中文字体、Linux 的其他中文字体甚至日韩字体来说，方法类似。

测试 ghostscript 对 Windows 中文字体的支持

下载 PS 测试文件 GMT_Chinese_Linux.ps，并打开终端用 gs GMT_Chinese_Linux.ps 命令查看该 PS 文件。若正确显示中文如下图，则表明

ghostscript 已支持 Windows 中文字体。

Song Typeface 宋体

Fangsong Typeface 仿宋体

Hei Typeface 黑体

Kai Typeface 楷体

注解: PS 文件本质上是一个纯文本文件, 可以用**编辑器**打开该 PS 文件以查看其内容。

PS 文件中要使用某个中文字体, 需要用 `FontName--CMap` 的格式来调用。其中 `FontName` 即 gs 中文配置文件中给定的字体名。CMap 可以取 `UniGB-UTF8-H` 和 `GB-EUC-H`, Linux 下一般用前者, Windows 下一般用后者, 用于指定汉字或中文字体的编码。

2.2.2 使 GMT 支持中文

配置 GMT 中文字体

新建 GMT 自定义字体配置文件 `~/.gmt/PSL_custom_fonts.txt` (若不存在 `~/.gmt` 目录则需新建该目录)。

向 GMT 自定义字体配置文件 `~/.gmt/PSL_custom_fonts.txt` 中加入如下语句:

```
STSong-Light--UniGB-UTF8-H 0.700 1
STFangsong-Light--UniGB-UTF8-H 0.700 1
STHeiti-Regular--UniGB-UTF8-H 0.700 1
STKaiti-Regular--UniGB-UTF8-H 0.700 1
```

第一列为字体名, 第二列为字母 A 的高度, 第三列与编码有关。

查看 GMT 当前支持的字体

用 `gmt pstext -L` 命令查看 GMT 当前的字体配置:

```
$ gmt pstext -L
Font # Font Name
-----
0 Helvetica
1 Helvetica-Bold
```

(下页继续)

(续上页)

```
...      ....  
39 STSong-Light--UniGB-UTF8-H  
40 STFangsong-Light--UniGB-UTF8-H  
41 STHeiti-Regular--UniGB-UTF8-H  
42 STKaiti-Regular--UniGB-UTF8-H
```

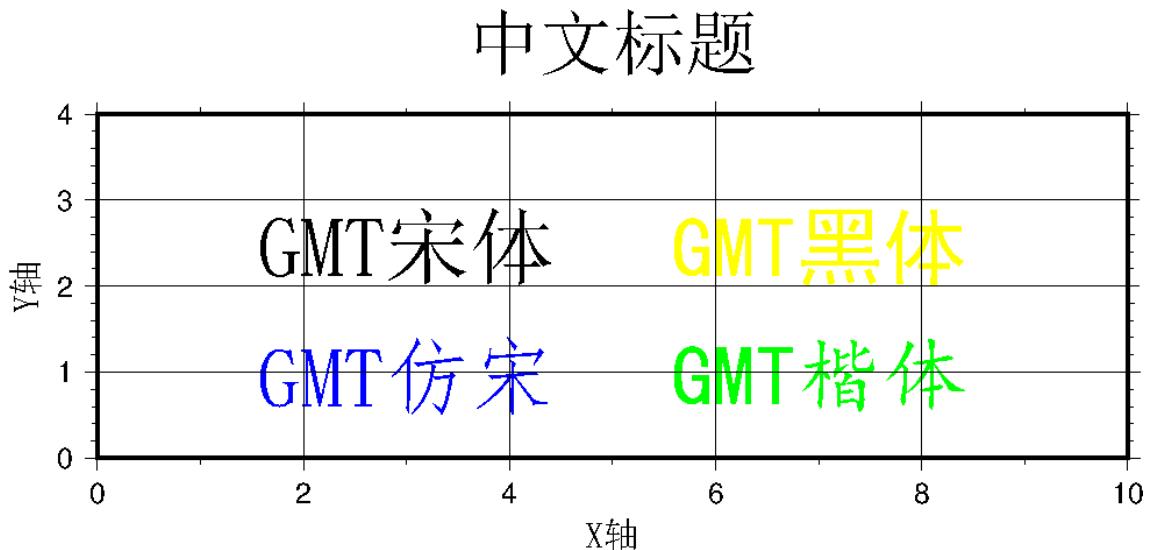
其中 0-38 为 GMT/gs 默认支持的西文字体, 39-42 为新添加的中文字体。以后要用中文字体时, 需要用这些编号来指定字体, 也许你的机器上的编号和这里不同。

GMT 中文测试

测试脚本:

```
#!/bin/bash  
gmt set FONT_TITLE 30p,39,black  
gmt set FONT_LABEL 15p,39,black  
  
gmt pstext -R0/10/0/4 -JX15c/5c -Bxafg+l"X 轴" -Byafg+l"Y 轴" \  
    -BWSen+t" 中文标题" -F+f -P > cn.ps << EOF  
3 2.5 35p,39,black GMT 宋体  
3 1.0 35p,40,blue GMT 仿宋  
7 2.5 35p,41,yellow GMT 黑体  
7 1.0 35p,42,green GMT 楷体  
EOF  
  
rm gmt.history gmt.conf
```

成图效果如下:



2.2.3 对其他发行版的若干说明

其他发行版与 CentOS 7 之间或多或少有一些区别, 列举如下。

CentOS 6

- ghostscript 中文配置文件需要用如下命令安装:

```
sudo yum install cjkuni-fonts-ghostscript
```

在安装配置文件的同时会安装中文字体 uming 和 ukai。

- ghostscript 中文配置文件中给定的字体路径: /usr/share/fonts/cjkuni/uming.ttc 和 /usr/share/fonts/cjkuni/ukai.ttc 是错误的。正确的字体路径是 /usr/share/fonts/cjkui-uming/uming.ttc 和 /usr/share/fonts/cjkuni-ukai/ukai.ttc, 要注意改正。

Ubuntu 14.04/15.04

- ghostscript 中文配置文件可以用如下命令安装(默认已安装):

```
sudo apt-get install poppler-data
```

- ghostscript 中文配置文件路径为: /etc/ghostscript/cidfmap.d/90gs-cjk-resource-gb1.conf
- ghostscript 中文配置文件中默认使用的 Linux 字体为 uming 和 ukai, 需要通过如下命令安装:

```
sudo apt-get install fonts-archic-uming fonts-archic-ukai
```

- gs 中文配置文件的默认内容为:

```
/BousungEG-Light-GB </FileType /TrueType /Path (/usr/share/fonts/truetype/archic/
↳uming.ttc) /SubfontId 0 /CSI [(GB1 4] >> ;
/GBZenKai-Medium </FileType /TrueType /Path (/usr/share/fonts/truetype/archic/
↳ukai.ttc) /SubfontId 0 /CSI [(GB1 4] >> ;
/Song-Medium /GBZenKai-Medium ;
/STSong-Light /BousungEG-Light-GB ;
/STFangsong-Light /BousungEG-Light-GB ;
/STHeiti-Regular /BousungEG-Light-GB ;
/STKaiti-Regular /BousungEG-Light-GB ;
/Adobe-GB1 /BousungEG-Light-GB ;
/Adobe-GB1-Bold /GBZenKai-Medium ;
```

需要将该文件改成:

```
% 原配置文件的内容, 与 STSong-Light 等相关的四行必须删除
/BousungEG-Light-GB </FileType /TrueType /Path (/usr/share/fonts/truetype/archic/
↳uming.ttc) /SubfontId 0 /CSI [(GB1 4] >> ;
/GBZenKai-Medium </FileType /TrueType /Path (/usr/share/fonts/truetype/archic/
↳ukai.ttc) /SubfontId 0 /CSI [(GB1 4] >> ;
/Song-Medium /GBZenKai-Medium ;
/Adobe-GB1 /BousungEG-Light-GB ;
/Adobe-GB1-Bold /GBZenKai-Medium ;

% 新增 Windows 字体的支持
```

(下页继续)

(续上页)

```
/STSong-Light <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simsun.ttc) /  
    ↳SubfontId 0 /CSI [(GB1) 4] >> ;  
/STFangsong-Light <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simfang.  
    ↳ttf) /SubfontId 0 /CSI [(GB1) 4] >> ;  
/STHeiti-Regular <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simhei.  
    ↳ttf) /SubfontId 0 /CSI [(GB1) 4] >> ;  
/STKaiti-Regular <</FileType /TrueType /Path (/usr/share/fonts/winfonts/simkai.  
    ↳ttf) /SubfontId 0 /CSI [(GB1) 4] >> ;
```

修改完 ghostscript 中文配置文件后, 必须要执行如下命令:

```
$ sudo update-gsfontmap
```

该命令会将 `/etc/ghostscript/cidfmap.d/*.conf` 合并成单独的文件 `/var/lib/ghostscript/fonts/cidfmap.gs` 在需要中文字体时会读取 `/var/lib/ghostscript/fonts/cidfmap` 而不是 `/etc/ghostscript/cidfmap.d/*.conf`。这是 Ubuntu/Debian 和 CentOS 的一个很大不同。

Ubuntu 12.04

- ghostscript 中文配置文件需要用如下命令安装:

```
sudo apt-get install gs-cjk-resource
```

- 其他部分未做测试, 估计跟 Ubuntu 15.05 差不多。

2.2.4 参考资料

- GMT 软件显示汉字的技术原理与实现, 赵桂儒, 《测绘通报》
- [ghostscript 中文打印经验](#)
- [GMT 中文支持](#)
- [维基词条: PostScript](#)
- [Debian Wiki](#)

2.3 Windows 下安装 GMT

GMT 为 Windows 用户提供了安装包, 可以直接安装使用。Windows 下需要安装 GMT、ghostscript, 查看 PS 文件需要使用 gsview。日常数据处理需要 Linux 下的小工具。

警告: 从 GMT 5.2.1 开始, GMT 提供的 Windows 下的安装包已经不再支持 Windows XP。

- 下载
到社区主页的 [下载页面](#) 下载所需要的安装包。
- 安装 GMT

直接双击安装包即可安装，直接点击下一步，使用默认的选项即可，无须做任何修改。在“选择组件”页面，建议将四个选项都勾选上，然后点击下一步安装完成。

注解： 安装过程中可能会出现 `Warning! PATH too long installer unable to modify PATH!` 的警告。出现此警告的原因是系统的环境变量 PATH 太长，GMT 安装包无法直接修改。

解决办法是，先忽略这一警告，待安装完成后按照如下步骤自行修改系统环境变量 PATH。

1. 点击“计算机”->“属性”->“高级系统设置”->“环境变量”打开“环境变量”编辑工具
 2. 在“系统变量”部分中，选中“Path”并点击“编辑”
 3. 在“变量值”的最后加上 GMT 的安装路径。需要注意“path”变量值中多个路径之间用英文分号分隔
-

安装完成后，“开始”->“所有程序”->“附件”->“命令提示符”以启动 cmd。在 cmd 窗口中执行：

```
C:\Users\xxxx> gmt --version
5.4.5
```

即表示安装成功。

3. 安装 ghostscript

安装的过程没什么可说的，在最后一步，记得勾选 `Generate cidfmap for Windows CJK TrueType fonts`。

4. 安装 gsview

双击直接安装即可。

5. 安装 UnixTools

解压压缩包，并将解压得到的 exe 文件移动到 GMT 的 bin 目录即可。

某些情况下，运行 GMT 脚本时会出现警告信息：

```
gmt: Unable to create GMT User directory : /Users/用户名/.gmt
gmt: Auto-downloading of earth_relief_##m|s.grd files has been disabled.
```

解决办法：

```
1、打开“我的电脑”->“属性”->“高级系统设置”->“环境变量”
2、新建“系统变量”
    变量名： HOME
    变量值： C:\Users\用户名
```

重新运行脚本检查问题是否得到解决。经过测试, HOME 变量的值可以取任意盘符中任何已存在的文件夹, 正常情况下, 执行脚本后会生成 %HOME%/.gmt/cache 空文件夹。

如果想要同时使用 GMT4 和 GMT5, 则需要在安装完 GMT4 和 GMT5 之后到 GMT5 的 bin 目录下, 将该目录下的所有文件按照大小排序, 所有大小为 6 KB 的都是“符号链接”, 直接选中删除就好。

2.4 Windows 下的 GMT 中文支持

2.4.1 ghostscript 的中文支持

通常, 在 C:\Program Files\gs\gs9.15\examples\cjk 目录下可以找到文件 gscjk_ag.ps。

注解: 如果找不到该文件, 请尝试重新安装 ghostscript。在安装的过程中, 会有一个生成 cidmap 的选项, 选中该选项则表示会为当前系统自动生成中文所需的 cidmap 文件。默认该选项是被选中的, 一定 **不要将该选项取消**;

启动 cmd, 键入如下命令:

```
cd "C:\Program Files\gs\gs9.16\bin"  
gswin64.exe ..\examples\cjk\gscjk_ag.ps
```

该命令用命令行版本的 gswin64c 打开 gscjk_ag.ps, 若能看到中文, 则说明 ghostscript 是可以正常支持中文的。

2.4.2 gsview 的中文支持

如果你需要用 gsview 查看 PS 文件, 则需要为 gsview 配置中文显示。否则, 则可以跳过这一部分。

安装好 gsview 之后, PS 格式会自动与 gsview 关联。一般情况下, 直接双击 PS 文件, 就会用 gsview 打开该 PS 文件。

双击打开 gscjk_ag.ps, 一般情况下不会正确显示汉字。这是因为 gsview 在打开 PS 文件时没有找到汉字所对应的字体文件。

在 gsview 的“选项”->“高级配置”中, 将 Ghostscript Options 由 -dNOPLATFONTS -sFONTPATH="c:\psfonts" 改成 -dNOPLATFONTS -sFONTPATH="C:\Windows\Fonts", 此时 gsview 在调用 gswin64 时会将选项传递给 gswin64, gswin64 则会在 FONTPATH 中搜索字体。

配置完毕后, 重新打开 gscjk_ag.ps, 若中文正常显示, 则表示 gsview 已支持中文。

2.4.3 GMT 的中文支持

新建 GMT 自定义字体配置文件 C:\Users\XXX\.gmt\PSL_custom_fonts.txt (若不存在 C:\Users\XXX\.gmt 目录则需新建该目录)。

向 GMT 自定义字体配置文件 C:\Users\XXX\.gmt\PSL_custom_fonts.txt 中加入如下语句:

```
STSong-Light--GB-EUC-H 0.700 1
STFangsong-Light--GB-EUC-H 0.700 1
STHeiti-Regular--GB-EUC-H 0.700 1
STKaiti-Regular--GB-EUC-H 0.700 1
```

用 gmt pstext -L 查看 GMT 字体, 可以看到, 新添加的四种中文字体对应的字体编号为 39 到 42。其中 STSong-Light--GB-EUC-H 即为宋体, GB-EUC 是文字编码方式, H 表示文字水平排列。

强烈建议在执行测试脚本前确认自己的中文字体编号。

2.4.4 测试脚本

注解: 请自行确认你的中文字体编号。如果编号不是 39 到 42, 请自行修改以下测试脚本。

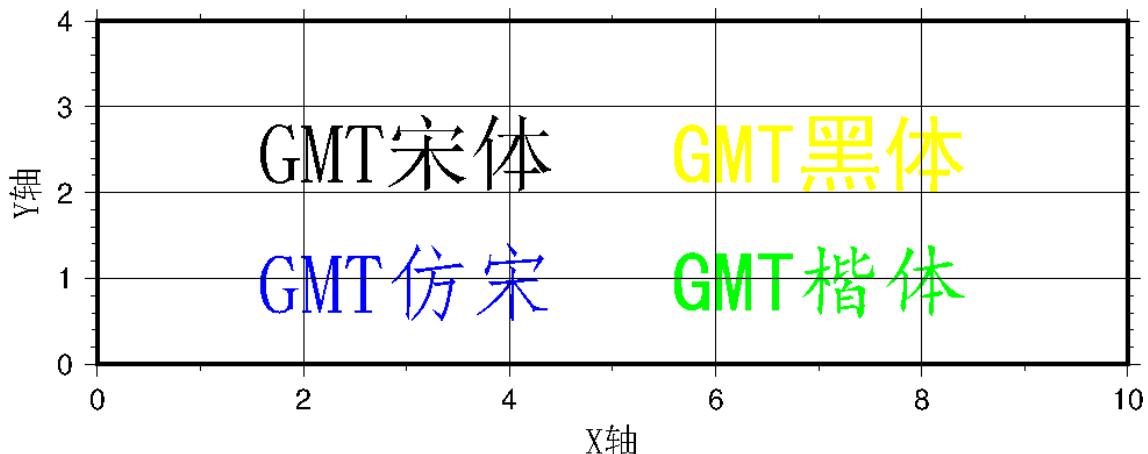
```
gmt set FONT_TITLE 30p,39,black
gmt set FONT_LABEL 15p,39,black

echo 3 2.5 35p,39,black GMT 宋体 > tmp
echo 3 1.0 35p,40,blue GMT 仿宋 >> tmp
echo 7 2.5 35p,41,yellow GMT 黑体 >> tmp
echo 7 1.0 35p,42,green GMT 楷体 >> tmp

gmt pstext tmp -R0/10/0/4 -JX15c/5c -Bxafg+l"X 轴" -Byafg+l"Y 轴" -BWSen+t"中文标题" -F+f
➥> GMT_chinese.ps
gmt psconvert GMT_chinese.ps -C-sFONTPATH=C:\windows\fonts -Tg -A -P -E300
del gmt.* tmp
```

成图效果如下:

中文标题



需要注意，若使用记事本编辑 bat 文件，则保存时应注意编码方式为 ANSI、Unicode 或 Unicode big endian，若使用 UTF-8 编码则会出现乱码；另外，很多编辑器默认将文本文件以 UTF-8 编码保存，因而可能需要修改编辑器的默认编码。

2.5 macOS 下安装 GMT

macOS 下 GMT 的安装方法有很多，可以直接使用安装包，也可以使用各种软件管理工具。

推荐使用 homebrew 方式安装。

2.5.1 使用 homebrew 安装

Homebrew 是 macOS 下的第三方软件包管理工具。

1. 安装 Homebrew:

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

2. 安装 GMT:

```
$ brew update && brew upgrade  
$ brew install gmt
```

3. 安装 ghostscript:

```
$ brew install ghostscript
```

4. 测试安装是否成功:

```
$ gmt --version  
5.4.5
```

如果想同时安装 GMT4 和 GMT5, 还需要执行以下步骤:

```
# 安装 GMT4
$ brew install gmt4

# 删除 GMT5 带的软链接
$ cd /usr/local/opt/gmt@5/bin/
$ find . -size -4c -delete      # 删除所有文件大小小于 4 字节的软链接
```

然后向 `~/.bashrc` 中写入如下语句以修改环境变量:

```
export GMT4HOME=/usr/local/opt/gmt@4
export PATH=${GMT4HOME}/bin:$PATH
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}: ${GMT4HOME}/lib64'
```

2.5.2 使用 GMT 安装包

GMT 为 macOS 用户提供了 dmg 安装包。

1. 到 [下载页面](#) 下载最新版本的 dmg 安装包。
2. 双击 dmg 包以解压, 将解压得到的 `GMT-5.4.5.app` 拖动到 Applications 目录即可。
3. GMT 默认会安装到 `/Applications/GMT-5.4.5.app/` 目录下, 将如下语句:

```
export PATH=${PATH}:/Applications/GMT-5.4.5.app/Contents/Resources/bin
```

加入到 `~/.bashrc` 中即可。

4. 测试安装是否成功:

```
$ gmt --version
5.4.5
```

2.6 macOS 下的 GMT 中文支持

本文介绍如何让 GMT 在 macOS 下支持中文。

2.6.1 使 gs 支持中文

首先需要使 gs 支持中文, 这可以通过 [cjk-gs-support](#) 项目提供的脚本 `cjk-gs-integrate.pl` 实现。

1. 下载脚本 [cjk-gs-integrate.pl](#)
2. 执行脚本:

```
$ sudo perl cjk-gs-integrate.pl
```

该脚本会自动搜索系统中自带的中文字体, 并生成 gs 支持中文所需的配置文件。

2.6.2 使 GMT 支持中文

修改 GMT 字体配置文件

在 `~/.gmt` (若无该文件夹, 请自行新建) 下创建字体配置文件:

```
$ touch ~/.gmt/PSL_custom_fonts.txt  
$ open ~/.gmt/PSL_custom_fonts.txt
```

打开 GMT 字体配置文件, 在文件中加入如下语句:

```
STSong-Light--UniGB-UTF8-H 0.700 1  
STFangsong-Light--UniGB-UTF8-H 0.700 1  
STHeiti-Regular--UniGB-UTF8-H 0.700 1  
STKaiti-Regular--UniGB-UTF8-H 0.700 1
```

这几句话分别添加了宋体、仿宋、黑体和楷体四种字体。

查看 GMT 当前支持的字体

用 `gmt pstext -L` 命令查看 GMT 当前的字体配置:

```
$ gmt pstext -L  
Font # Font Name  
-----  
0 Helvetica  
1 Helvetica-Bold  
... .....  
39 STSong-Light--UniGB-UTF8-H  
40 STFangsong-Light--UniGB-UTF8-H  
41 STHeiti-Regular--UniGB-UTF8-H  
42 STKaiti-Regular--UniGB-UTF8-H
```

其中 39-42 号字体为新添加的中文字体。以后要用中文字体时, 需要用这些编号来指定字体, 也许你的机器上的编号和这里不同。

GMT 中文测试

测试脚本:

```
#!/bin/bash  
gmt set FONT_TITLE 30p,39,black  
gmt set FONT_LABEL 15p,39,black  
  
gmt pstext -R0/10/0/4 -JX15c/5c -Bxafg+l"X 轴" -Byafg+l"Y 轴" \  
-BWSen+t" 中文标题" -F+f -P > cn.ps << EOF  
3 2.5 35p,39,black GMT 宋体  
3 1.0 35p,40,blue GMT 仿宋  
7 2.5 35p,41,yellow GMT 黑体  
7 1.0 35p,42,green GMT 楷体  
EOF  
gmt psconvert -A -P -Tf cn.ps
```

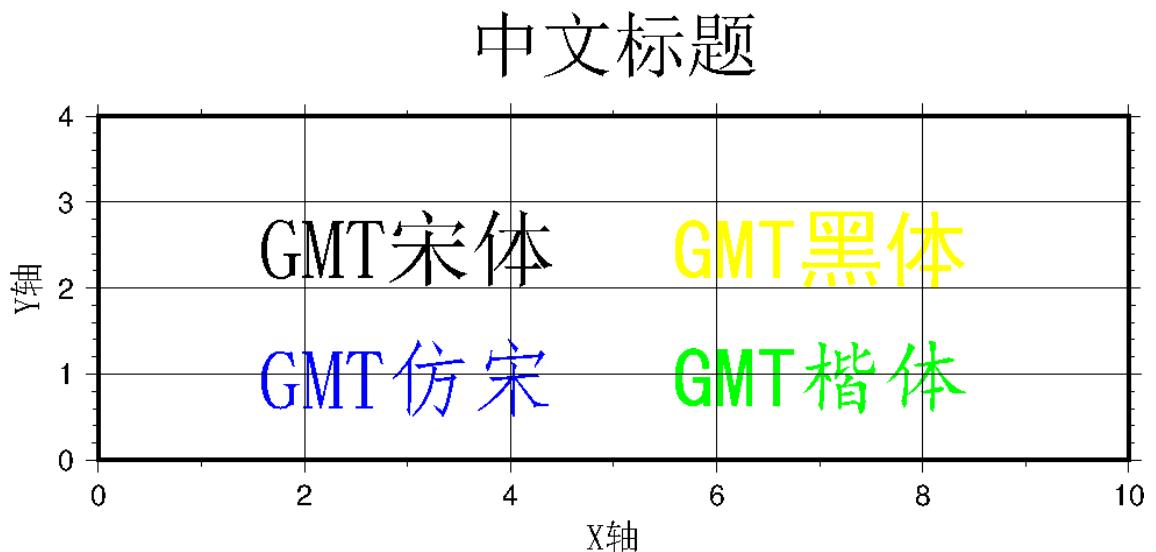
(下页继续)

(续上页)

```
gmt psconvert -A -P -Tg cn.ps  
gmt clear history conf
```

注解：生成的 PNG、JPG 格式的图片中可直接显示中文，而生成的 PDF 文件用 macOS 自带的 PDF 预览工具打开无法显示中文，使用 Adobe Reader 打开则可以正常显示中文。

成图效果如下：



保护环境，从阅读电子文档开始！

第 3 章 GMT 初探

3.1 命令行

与平常接触的 Adobe PhotoShop、Adobe Illustrator 之类的绘图工具不同，GMT 是没有图形界面的，即无法通过鼠标点击或拖动来执行绘图操作。GMT 是纯命令行的，所有绘图操作都是通过一系列命令来实现的。

命令行相对于图形界面的优势在于：

- 内存占用更低
- 精确控制图形的显示，比如线条宽度、圆的大小
- 方便写成脚本，可批量绘图、可重复、可自动化

其劣势在于：

- 不够直观
- 某些无法用坐标精确定义的曲线难以绘制

3.1.1 终端

终端，即 terminal，是用户键入命令与电脑进行交互的接口。

如何启动终端？

- Linux：终端是 Linux 系统的标配。如果你是 Linux 用户，相信你已经知道如何启动终端了。
- Windows：默认终端是“命令提示符”，也就是 cmd。直接点击“开始”->“附件”->“命令提示符”即可启动 cmd。也可以直接在开始按钮中的搜索框中搜索“cmd”并启动。
- macOS：可以使用“Command+ 空格”键启动 Spotlight，在弹出的搜索框中输入“terminal”再按回车即可。

启动终端后，默认情况下当前路径是自己的家目录。对于 Linux 用户，通常是 `/home/username`；对于 Windows 用户，通常是 `C:\Users\username`；对于 macOS 用户，通常是 `/Users/username`。

在终端中键入如下命令：

```
$ gmt pscoast -JM10c -R70/140/2/60 -Bafg -Ggray -Wthin > map.ps
```

则会在当前目录（即家目录）下生成名为 `map.ps` 的一张图，其中绘制了中国及其周

边的海岸线。

Windows 下可以直接双击以查看该 PS 文件；Linux 下可以用 `gs map.ps` 查看；macOS 下可以直接用 `open map.ps` 查看。

3.1.2 脚本

一张稍复杂的图通常都需要多个 GMT 命令来完成。你可以在终端键入如下几行命令来绘制海岸线并在某个点处加一个五角星：

```
gmt pscoast -JM10c -R70/140/2/60 -Bafg -W1/thin -Ggray -P -K > map.ps
echo 115 40 | gmt psxy -J -R -Sa0.5c -Gred -O >> map.ps
gmt clear history
```

直接在终端敲大一堆命令是不是很麻烦？一旦敲错了就得把光标移过去重新编辑，或者哪里不满意想修改，还得查找命令历史把命令找出来重新执行一遍，又或者一个不小心命令历史丢了，原来的画图命令再也找不到了。这一切都可以通过脚本来解决。

Linux 或 macOS 下，可以新建一个名为 `test.sh` 的文件，将上面的命令复制到文件中并保存就完成了一个脚本。然后在终端中键入如下命令即可执行该脚本：

```
$ sh ./test.sh
```

Windows 下，可以将上面的命令复制到名为 `test.bat` 的文件内并保存，然后双击该 bat 文件即可执行脚本。此时会看到一个黑框一闪而过。

3.1.3 脚本语言

脚本最简单的用法就是像上面的示例那样，把一堆要执行的命令复制进去即可，但脚本的功能可不仅仅如此，要学会善于利用脚本语言的强大功能，比如：

1. 定义变量，代替命令中重复的部分
2. 用脚本语言完成计算与数据处理
3. 利用脚本实现循环
4. 利用脚本的命令行参数实现脚本的可复用性

Linux 和 macOS 下最常见的脚本语言是 bash，Windows 下则是 bat。当然还有跨平台的 Perl 和 Python，本文档的所有示例都使用 bash 语法。

3.2 命令格式

GMT 的所有功能都是通过调用 `gmt` 这个命令来完成的。

一个 GMT 命令由“`gmt + 模块 + 选项 + 参数`”构成，写成如下形式：

```
gmt module -Axx+bxxxx -Bxx+axxxx
```

其中，

- `gmt` 是 GMT 中“唯一”的一个二进制文件, 所有 GMT 命令必须以 `gmt` 开头
- `module` 是用于完成某个特定操作的模块的名字
- `-A` 是模块 `module` 提供的选项, `xx` 为选项 `-A` 的参数
- `+b` 是选项 `-A` 的子选项, `xxxx` 为该子选项的参数

一个完整的示例:

```
gmt pscoast -R0/20/0/20 -JM6i -Ggray -Wthin -B5 -B+t"Title with spaces" -V -P > map.ps
```

其中,

- 命令以 `gmt` 开头
- `pscoast` 是用于绘制海岸线的模块
- `-R`、`-J`、`-G` 等都是 `pscoast` 模块的选项
- `-B+t"Title with spaces"` 中 `+t` 是选项 `-B` 的子选项, `"Title with spaces"` 是子选项 `+t` 的参数

几点说明:

- 若模块名以 `gmt` 开头, 则模块名中的 `gmt` 可省略。比如 `gmt gmtconvert xxx` `xxx` 可以简写为 `gmt convert xxx xxx`
- 模块名、选项等均区分大小写
- 每个模块可以使用哪些选项由模块自己定义, 具体参考每个模块的语法说明
- 选项以 `-` 开头, 后接单个字符表示某个选项, 字符后接选项的参数以及子选项
- 不以 `-` 开头的参数, 都会被当做文件, GMT 会尝试去读取
- 子选项以 `+` 开头, 后接单个字符以及子选项的参数
- 官方文档中子选项称为 modifier, 可以译为“修饰符”, 本文档中统一使用“子选项”
- 各选项间以空格分隔, 选项内部不能有空格
- 选项内部的字符串, 若存在空格, 应用单引号或双引号括起来
- 注意 `-A`、`-A` 以及 `- A` 的细微区别, 这通常是由于标点符号的全角和半角导致的。GMT 中只能使用第一种。GMT4 官方文档以及某些介绍 GMT 的博客, 都有此问题。GMT5、6 的官方文档以及本文档不存在该问题。

3.3 绘制第一张图

这一节将通过绘制一张地震学常见的射线路径分布图, 来展示 GMT 模块化作图的思想以及 GMT 中图层的概念。

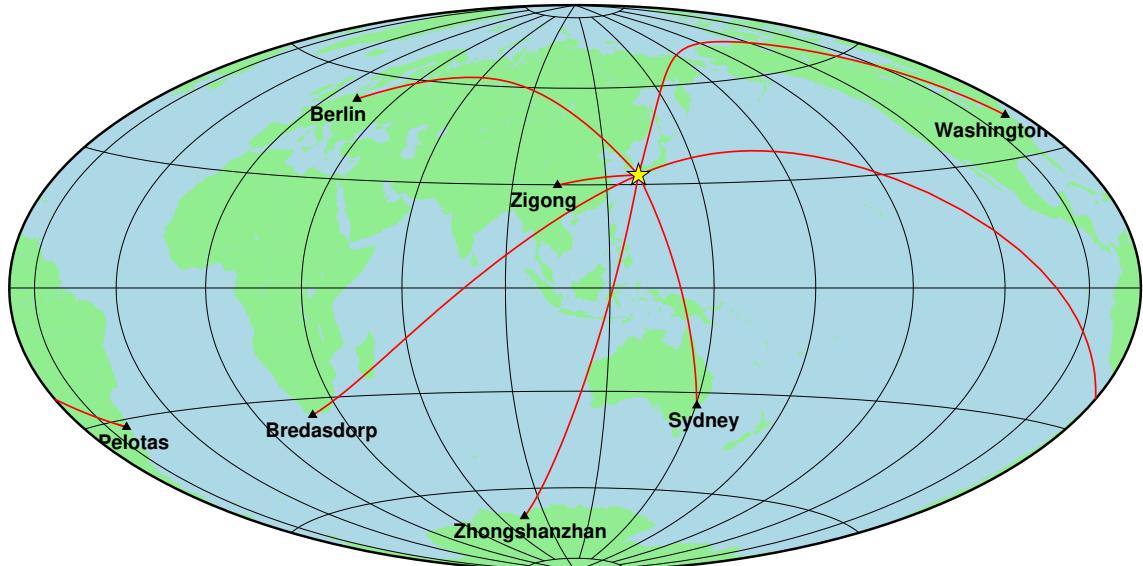
3.3.1 图件分析与拆解

一个完整的图件通常是由很多部分组成的。GMT 在制图时遵循模块化思想, 每个模块只绘制整张图的一部分, 因而通常一张图需要使用多个 GMT 命令才能绘制完成。绘制的过程中, 若想修改图中的某个部分, 只需修改绘制该部分所使用的命令即可, 而不会影响其他部分。这就是模块化作图。

下图展示了我们最终想要实现的绘图效果。整张图由如下几个部分构成:

1. 底图 (海岸线)
2. 震中位置 (五角星)
3. 台站位置 (三角形)、射线路径 (大圆弧线段)
4. 台站名 (文本)

Source Code

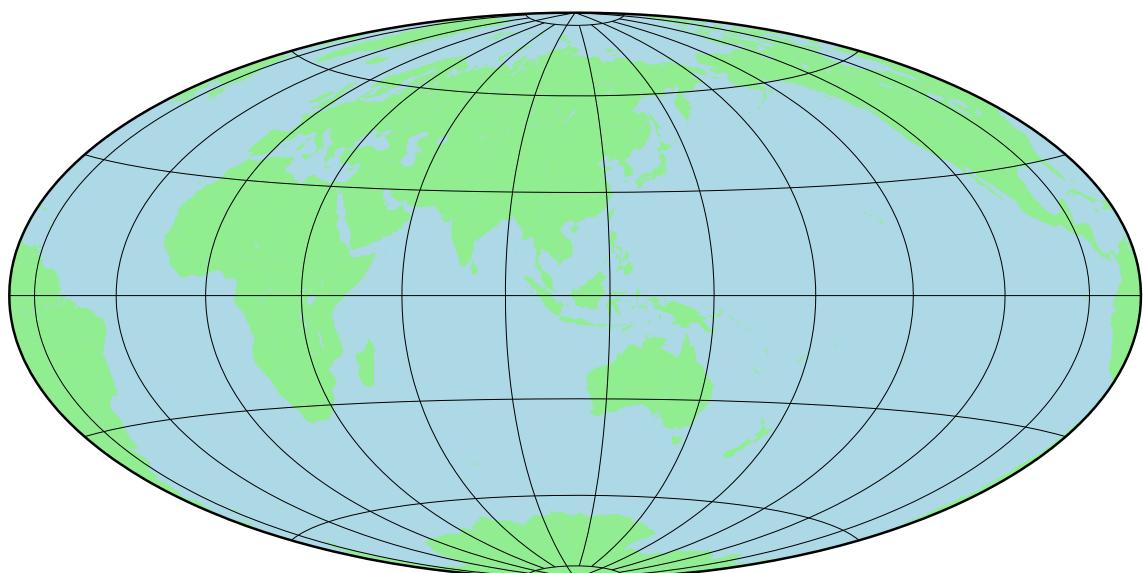


下面会一步步地把整张图绘制出来。

3.3.2 绘制底图

GMT 中的 `pscoast` 模块, 可以绘制带海岸线的底图, 并可以给陆地和海洋分别填充不同的颜色, 绘图命令及效果如下:

```
gmt pscoast -JH110/24c -Rg -Bg30 -Glightgreen -Slightblue -A1000 -Dc > first-plot.ps
```



解释以下几个选项：

1. `-JH110/24c` 表示使用 Hammer 投影，投影中心为 110 度，整张图宽度为 24 厘米
2. `-Rg` 相当于 `-R0/360/-90/90` 即绘制全球范围
3. `-Bg30` 表示网格线每 30 度显示一条
4. `-Glightgreen` 表示将陆地填充浅绿色，`-Slightblue` 表示将海洋填充浅蓝色

至此，即得到了我们需要的底图。

3.3.3 绘制震中和台站位置

`psxy` 模块可以用于绘制多种不同类型的符号，比如五角星和三角形。绘图脚本如下：

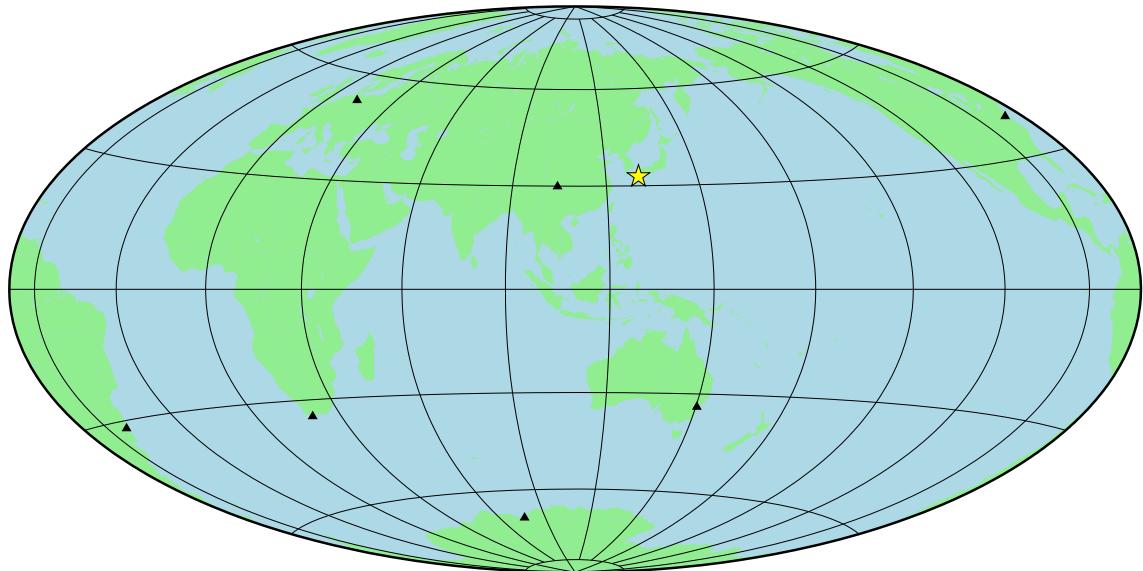
```
#!/bin/bash
J=H110/24c
R=g
PS=GMT_tutor2_2.ps

gmt pscoast -J$J -R$R -Bg30 -Glightgreen -Slightblue -A1000 -Dc -K > $PS

# 绘制震中位置
gmt psxy -J -R -Sa0.5c -W0.5p,black,solid -Gyellow -K -O >> $PS << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt psxy -J -R -St0.2c -W0.5p,black,solid -Gblack -K -O >> $PS << EOF
104.39 29.90
13.14 52.50
19.99 -34.52
-77.15 38.89
-52.47 -31.62
150.36 -33.79
76.22 -69.22
EOF

gmt psxy -J -R -T -O >> $PS
rm gmt.*
```



这里我们通过两次调用 `psxy` 命令分别绘制了五角星和三角形。由于绘图的命令比较多, 所以我们将绘图命令写成了一个脚本, 并定义了一些变量(比如 `$PS`)以避免重复输入类似的文本。

解释一下:

1. 第一个 `psxy` 命令用于绘制震中位置, 第二个则用于绘制台站位置
2. `-Sa0.5c` 表示绘制大小为 0.5 厘米的五角星, `-St0.2c` 表示绘制大小为 0.2 厘米的三角形
3. `-W` 表示画笔的属性, 这里用于指定用什么样的画笔绘制三角形或五角星的轮廓。
`-W0.5p,black,solid` 的意思是画笔 0.5p 宽, 黑色, 实心。
4. `-G` 表示颜色填充, 后面跟颜色的名字; `-Gblack` 表示填充黑色; `-Gyellow` 表示填充黄色。
5. 确定了画什么样的符号, 还要确定在哪里画。两个 `EOF` 之间的是命令的输入数据, 在这里就是位置信息, 一行代表一个位置。默认情况下, GMT 认为第一个数是经度, 第二个是纬度。
6. 最后一个 `psxy` 命令没有绘制任何东西, 其作用在于结束绘图, 这属于稍高级的技巧, 以后会介绍到。

注意到了吗? 相对于上一步的命令而言, `pscoast` 命令中多了一个 `-K` 选项, 中间的几个 `psxy` 命令使用了 `-K -O` 选项, 最后一个命令则使用了 `-O` 选项。具体原因此处不解释, 在[-K 和 -O 选项](#)一节会详细介绍。

3.3.4 绘制射线路径

`psxy` 模块还可以用于绘制线段, 默认情况下, `psxy` 会自动用大圆路径连接地球上的两个位置, 不需要额外设置。首先直接看代码:

```
#!/bin/bash
J=H110/24c
R=g
PS=GMT_tutor2_3.ps

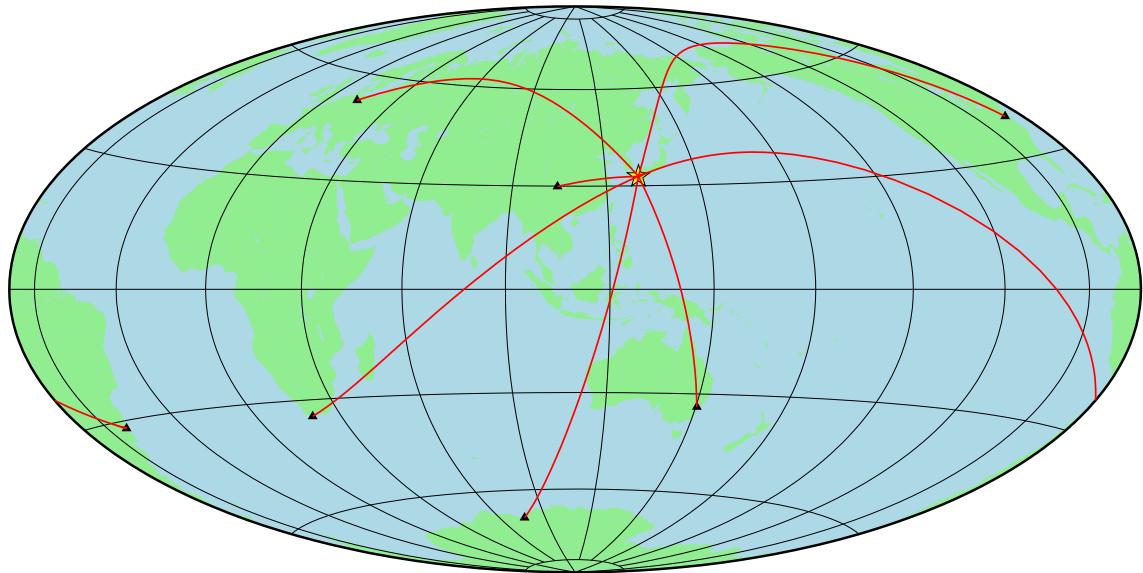
gmt pscoast -J$J -R$R -Bg30 -Glightgreen -Slightblue -A1000 -Dc -K > $PS

# 绘制震中位置
gmt psxy -J -R -Sa0.5c -W0.5p,black,solid -Gyellow -K -O >> $PS << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt psxy -J -R -St0.2c -W0.5p,black,solid -Gblack -K -O >> $PS << EOF
104.39 29.90
13.14 52.50
19.99 -34.52
-77.15 38.89
-52.47 -31.62
150.36 -33.79
76.22 -69.22
EOF

# 绘制大圆路径
gmt psxy -R -J -W1p,red -K -O >> $PS << EOF
>
130.72 32.78
104.39 29.90
>
130.72 32.78
13.14 52.50
>
130.72 32.78
19.99 -34.52
>
130.72 32.78
-77.15 38.89
>
130.72 32.78
-52.47 -31.62
>
130.72 32.78
150.36 -33.79
>
130.72 32.78
76.22 -69.22
EOF

gmt psxy -J -R -T -O >> $PS
rm gmt.*
```



用于绘制射线路径的 `psxy` 命令和之前的命令的区别在于没有了 `-S` 和 `-G` 选项。为了绘制一条线段，至少需要指定两个端点，输入数据中每个 `>` 之后的两行数据分别代表一条线条的两个端点位置。

3.3.5 添加台站名

最后还需要往图画里添加台站所在地区的名字。添加文字使用 `pstext` 模块：

```
#!/bin/bash
J=H110/24c
R=g
PS=GMT_tutor2_4.ps

gmt pscoast -J$J -R$R -Bg30 -Glightgreen -Slightblue -A1000 -Dc -K > $PS

# 绘制震中位置
gmt psxy -J -R -Sa0.5c -W0.5p,black,solid -Gyellow -K -O >> $PS << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt psxy -J -R -St0.2c -W0.5p,black,solid -Gblack -K -O >> $PS << EOF
104.39 29.90
13.14 52.50
19.99 -34.52
-77.15 38.89
-52.47 -31.62
150.36 -33.79
76.22 -69.22
EOF

# 绘制大圆路径
gmt psxy -R -J -W1p,red -K -O >> $PS << EOF
>
130.72 32.78
```

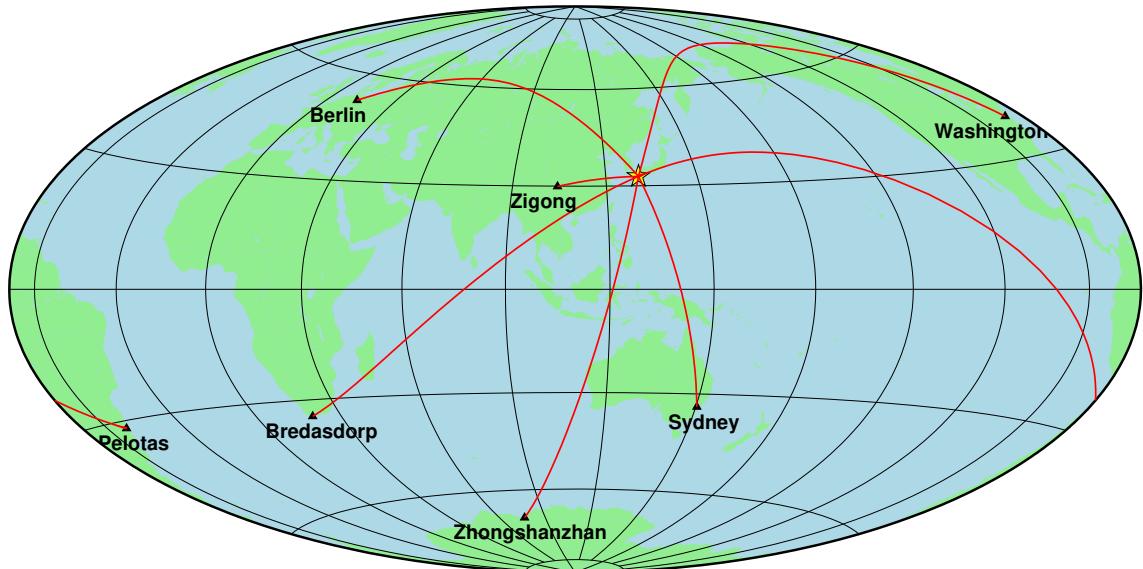
(下页继续)

(续上页)

```
104.39 29.90
>
130.72 32.78
13.14 52.50
>
130.72 32.78
19.99 -34.52
>
130.72 32.78
-77.15 38.89
>
130.72 32.78
-52.47 -31.62
>
130.72 32.78
150.36 -33.79
>
130.72 32.78
76.22 -69.22
EOF

# 添加文本
gmt pstext -J -R -F+f12p,1,black+jTL -D-1.5c/-0.15c -K -O >> $PS << EOF
-77.15 38.89 Washington
76.22 -69.22 Zhongshanzhan
EOF
gmt pstext -J -R -F+f12p,1,black+jTL -D-1c/-0.15c -K -O >> $PS << EOF
104.39 29.90 Zigong
13.14 52.50 Berlin
19.99 -34.52 Bredasdorp
EOF
gmt pstext -J -R -F+f12p,1,black+jTL -D-0.6c/-0.15c -K -O >> $PS << EOF
150.36 -33.79 Sydney
EOF
gmt pstext -J -R -F+f12p,1,black+jTL -D-0.6c/-0.15c -K -O >> $PS << EOF
-52.47 -31.62 Pelotas
EOF

gmt psxy -J -R -T -O >> $PS
rm gmt.*
```



解释如下：

1. EOF 之间的每一行依次是加入的文本的纬度、经度和内容，这与 psxy 类似，只是因为有内容，所以多了一列。之所以没有写在一个 gmt pstext 命令之下，是因为不同位置的文本的具体位置需要些许不同的调整，你可以看到每个 pstext 命令的 -D 参数些许不同，下面会详细解释。
2. -F 控制文本的字体、对齐方式等属性；+f10p,1,black 表示使用大小为 10p 的黑色 1 号字体；+j 是控制文本的对齐方式，TL 表示输入数据中的经纬度坐标是文本块的左上角，L 指左，T 指上，LT 和 TL 的含义完全一样。
3. -D 是在上述坐标的基础上微调。-D-1.5c/-0.15c 是说向左移动 1.5 厘米，向下移动 0.15 厘米。

3.3.6 图层的先后顺序

上面的绘图脚本已经基本绘制出我们最初想要的图件了。细细看会发现，还有一些不完美的地方：比如黄色五角星和三角形被线段盖住了。

这是因为，GMT 的每一个绘图命令都会产生一个图层，后绘制的图层会覆盖在先绘制的图层的上面，即后来者居上。解决办法也很简单，先绘制线段，再绘制三角形和五角星即可。

对脚本做微调，如下所示：

```
#!/bin/bash
J=H110/24c
R=g
PS=GMT_tutor2_5.ps

gmt pscoast -J$J -R$R -Bg30 -Glightgreen -Slightblue -A1000 -Dc -K > $PS
# 绘制大圆路径
```

(下页继续)

(续上页)

```

gmt psxy -R -J -W1p,red -K -O >> $PS << EOF
>
130.72 32.78
104.39 29.90
>
130.72 32.78
13.14 52.50
>
130.72 32.78
19.99 -34.52
>
130.72 32.78
-77.15 38.89
>
130.72 32.78
-52.47 -31.62
>
130.72 32.78
150.36 -33.79
>
130.72 32.78
76.22 -69.22
EOF

# 绘制震中位置
gmt psxy -J -R -Sa0.5c -W0.5p,black,solid -Gyellow -K -O >> $PS << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt psxy -J -R -St0.2c -W0.5p,black,solid -Gblack -K -O >> $PS << EOF
104.39 29.90
13.14 52.50
19.99 -34.52
-77.15 38.89
-52.47 -31.62
150.36 -33.79
76.22 -69.22
EOF

# 添加文本
gmt pstext -J -R -F+f12p,1,black+jTL -D-1.5c/-0.15c -K -O >> $PS << EOF
-77.15 38.89 Washington
76.22 -69.22 Zhongshanzhan
EOF
gmt pstext -J -R -F+f12p,1,black+jTL -D-1c/-0.15c -K -O >> $PS << EOF
104.39 29.90 Zigong
13.14 52.50 Berlin
19.99 -34.52 Bredasdorp
EOF
gmt pstext -J -R -F+f12p,1,black+jTL -D-0.6c/-0.15c -K -O >> $PS << EOF
150.36 -33.79 Sydney
EOF
gmt pstext -J -R -F+f12p,1,black+jTL -D-0.6c/-0.15c -K -O >> $PS << EOF

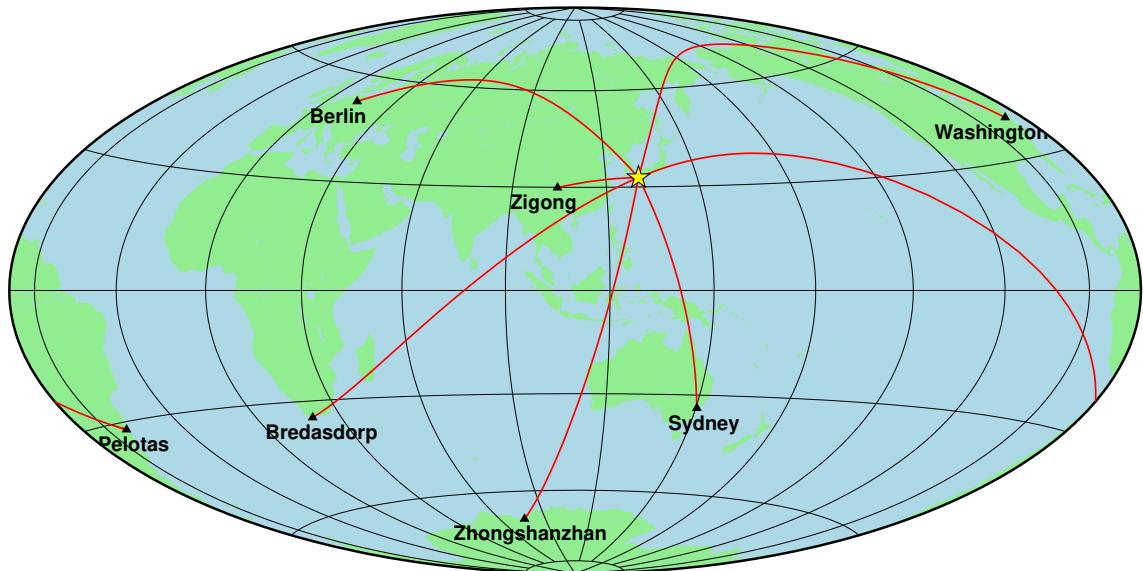
```

(下页继续)

(续上页)

```
-52.47 -31.62 Pelotas  
EOF
```

```
gmt psxy -J -R -T -O >> $PS  
rm gmt.*
```



需要注意的是，本节内容仅作演示用，实际绘图时通常会写脚本来处理自己的数据，以使得其符合 GMT 输入数据的要求，而不是像本节示例中这样手写所有的数据。

第 4 章 绘图基础

这一章介绍 GMT 中所有程序通用的基础知识。

4.1 画布

要画一张图，首先需要准备一个画布。GMT 的画布就是 PS 文件。GMT 默认的画布是 A4 大小的横置的白色 PS 文件。

注解：本节介绍的内容，涉及到后面的一些知识。因而这里只是做简单介绍，具体的操作看不懂也没关系。

4.1.1 画布尺寸

GMT 默认的画布尺寸为 A4，即 210mmx297mm。

通过修改 GMT 参数 *PS_MEDIA* 可以控制画布的默认尺寸。GMT 提供了几十种预定义的画布尺寸，还支持自定义尺寸，详情见 *PS_MEDIA* 的说明。

4.1.2 画布颜色

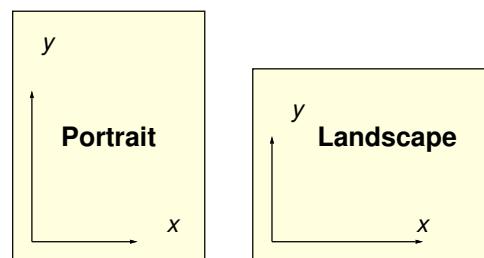
默认的画布颜色为白色，可以通过设置 *PS_PAGE_COLOR* 来修改画布颜色。

警告：若设置画布颜色为其他颜色，则使用 `psconvert` 将 PS 文件转化为其他图片格式时无法使用 `-A` 选项进行裁边。

4.1.3 画布方向

取一张 A4 纸放在桌面上，有两种放置的方式：竖着放和横着放。如图所示：

Source Code



图中左边把纸张竖着放的称为 Portrait (肖像) 模式, 右边把纸张横着放的方式称为 Landscape (风景) 模式。记起来也很简单, 人是瘦长的, 所以人的肖像照需要竖着拍, 风景是矮胖的, 就需要横着拍。

由于历史原因, GMT 中画布的默认放置方式是 Landscape 模式。有两种方式可以修改画布方向:

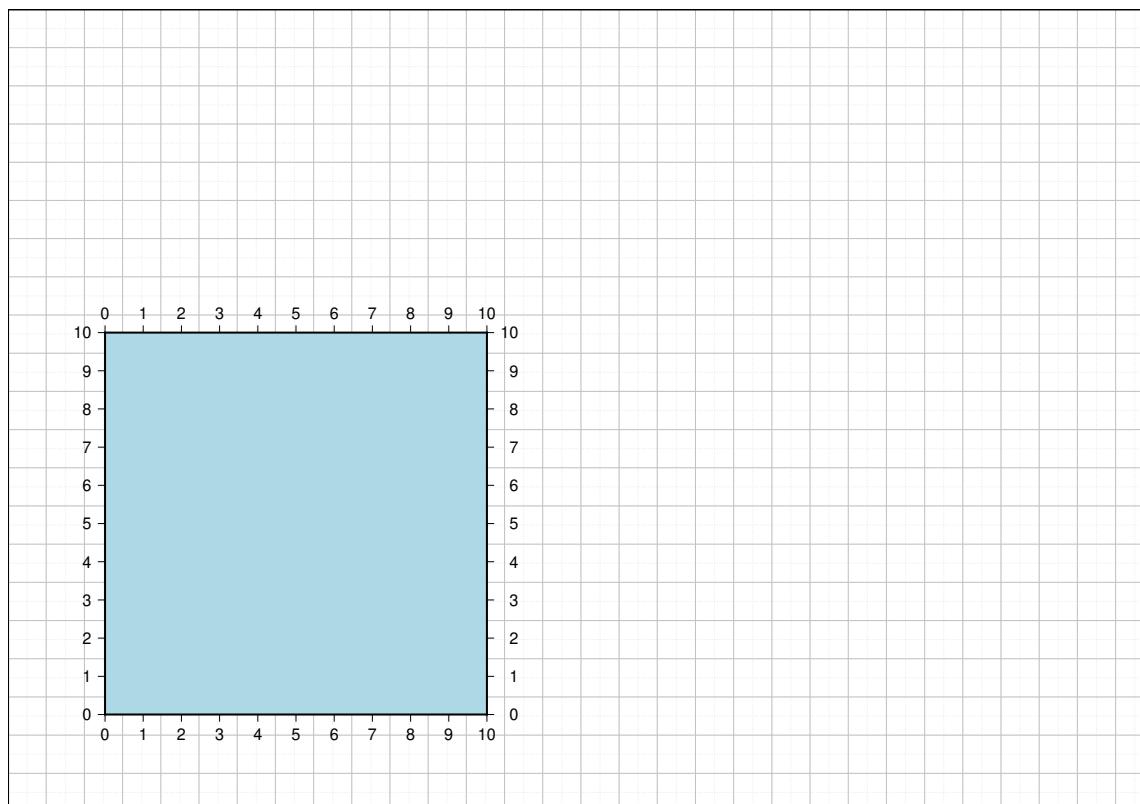
1. 在第一个绘图命令中使用 [-P 选项](#)
2. 设置 GMT 参数 `PS_PAGE_ORIENTATION` 为 `portrait`

小技巧: Linux 下默认的 PS 阅读器 evince 可以自动识别两种模式, Landscape 模式的 PS 文件横着放, Portrait 模式的文件竖着放, 所以看上去总是对的。而 gs 无法自动识别两种模式, 总是将文件竖着放, 因而用 gs 看 Landscape 模式的 PS 文件时就会很别扭。

4.1.4 底图原点

准备好画布之后, 可不能随便找一个点就开始画了, 还需要定义底图的原点。GMT 默认的底图原点距左下角 $(1i, 1i)$, 即坐标原点离纸张左下角的距离是水平方向偏移 1 英寸, 垂直方向偏移 1 英寸。

[Source Code](#)



图中所示是一个完整的 A4 画布。其中灰色实线的间距是 1 cm, 灰色虚线 (右键查看

大图) 的间距是 0.5 cm。图中的底图用如下命令得到:

```
gmt psbasemap -R0/10/0/10 -JX10c/10c -B1 -B+glightblue > map.ps
```

该命令中, `-JX10c/10c` 规定了底图的宽和高都是 10 cm, 可以看到, 底图的左边界位于 `1i`, 右边界位于 `1i+10c` 处, 下边界位于 `1i`, 上边界位于 `1i+10c` 处。

重要: 此处提到的底图原点是指矩形底图的坐标轴的左下角相对于画布左下角的位置, 而不包括坐标轴的标注或标签部分。

可以在命令中使用 [-X 和 -Y 选项](#) 来修改底图的原点位置。

4.2 单位

GMT 中的单位有两类: 长度单位和距离单位。

注解: 本手册中, 长度单位专用于度量纸张上的距离, 而距离单位则用于度量真实地球上距离。

4.2.1 长度单位

GMT 中的长度量可以使用厘米 (`cm`)、英寸 (`inch`) 或点 (`point`) 为单位。它们之间的关系如下:

```
1 inch = 2.54 cm = 72 point
```

厘米、英寸和点, 在 GMT 中分别用 `c`、`i` 和 `p` 表示。例如 `5c` 表示 5 厘米, `3i` 表示 3 英寸, `2p` 表示 2 点。

GMT 有两种方式可以指定长度量的单位, 即显式指定单位和隐式指定单位。

显式指定单位 在长度量后直接加上单位以显式地指定当前长度量的单位。比如 `-X4c` 意味着传给 `-X` 选项的长度量是 4 厘米

隐式指定单位 若长度量后没有给定长度单位, 则 GMT 会根据参数 `PROJ_LENGTH_UNIT` 所指定的默认长度单位对该长度量进行解释。比如 `-X4` 中的长度量 4 会根据 `PROJ_LENGTH_UNIT` 取值的不同而被解释为 4 厘米、4 英寸或 4 点。

使用长度单位时, 建议遵循如下几条:

- 始终显式指定长度量的单位, 不依赖于 `PROJ_LENGTH_UNIT` 的值, 以免导致同一脚本在不同机器上由于 GMT 参数配置的差异而导致结果不同
- 长度单位 `p` 用于指定较小的长度量, 比如线宽、文字大小等

- 长度单位 `c` 和 `i` 用于指定较大的长度量, 比如底图宽度、圆圈大小等
- 尽量使用 SI 单位制 (`c`) 而不用 US 单位制 (`i`), 因为国人对于 1 厘米要比 1 英寸更有概念

4.2.2 距离单位

对于真实地球上的距离量而言, 常用单位包括:

- `d`: 弧度 (degree of arc)
- `m`: 弧分 (minute of arc)
- `s`: 弧秒 (second of arc)
- `k`: 千米 (kilometer)
- `e`: 米 (meter) 默认单位

还有几个不常用的单位:

- `f`: 英尺 (foot)
- `M`: Statute mile
- `n`: Nautical mile
- `u`: US Survey foot

对于一个距离量而言, 若不指定单位, 则默认其单位为 `e` (即“米”)。当然还是建议为每个距离量显式指定其单位, 使得命令更加清晰。比如在地球上以某点为中心画一个特定半径的圆, 半径取 30 等效于 `30e` 表示 30 米, 半径取 `30k` 则表示 30 千米。

对于距离量而言, 还涉及到如何计算地球或任意星球上两点之间距离的问题。GMT 中提供了三种不同精度的距离计算方式, 分别是 Flat Earth 距离、大圆路径距离和测地距离。三种距离计算方式的详细介绍见附录中 “[三种距离计算方式](#)” 一节。本节只介绍如何告诉 GMT 你想要用哪种距离计算方式。

1. **大圆路径距离**: 将地球当做球体, 是 GMT 中默认使用的距离计算方式。比如距离量 `50k` 就默认用大圆弧距离来计算
2. **Flat Earth 距离**: 是大圆弧距离的一阶近似, 通过在距离量前加上减号来使用这种计算方式。比如距离量 `-50k` 则使用 Flat Earth 方式计算距离
3. **测地距离**: 计算距离时考虑了地球椭率, 可以通过在距离量前加上加号来使用这种计算方式。比如距离 `+50k` 则使用大地测量方式来计算距离

三种距离计算的方式, 从计算精度上看: 大地测量距离 > 大圆弧距离 > Flat Earth 距离; 从计算速度上看, Flat Earth 距离 > 大圆弧距离 > 大地测量距离。

4.2.3 小练习

1. 使用 `gmt get PROJ_LENGTH_UNIT` 查看 GMT 的默认长度单位。
2. 如下命令在一个长宽均为 10 厘米 (由 `-JX10c/10c` 控制) 的矩形底图中心, 绘制了一个直径为 3 厘米 (由 `-Sc3c` 控制) 的圆:

```
echo 5 5 | gmt psxy -JX10c/10c -R0/10/0/10 -Baf -Sc3c -W1p > test.ps
```

将 `-Sc3c` 改成 `-Sc3i` 或 `-Sc3` 并绘图查看图件的区别。

4.3 颜色

既然是绘图，丰富的颜色是必须的。GMT 中可以通过五种不同的方式来指定颜色，即：

1. 颜色名
2. RGB 值
3. HSV 值
4. CMYK 值
5. 灰度值

4.3.1 颜色名

通过颜色名指定颜色是最直观的方式了。常见的颜色如 `white`、`black`、`red`、`orange`、`yellow`、`green`、`cyan`、`blue`、`magenta`、`gray`（或 `grey`）和 `brown` 等。除 `white` 和 `black` 之外，其余的几种常见颜色名还可以加上前缀 `light` 或 `dark`，以表示浅色和深色。比如 `lightblue`、`blue`、`darkblue` 分别表示浅蓝、蓝色和深蓝。

所有的颜色名都是不区分大小写的，所以 `lightblue`、`LIGHTBLUE` 或者 `LightBlue` 都是合法的颜色名。

GMT 共支持 663 种颜色名，可以查找附录中的[GMT 颜色表](#) 获取 GMT 所支持的颜色及其对应的 RGB 值。

4.3.2 RGB

即三原色光模型，或又称 RGB 颜色模型，是一种加色模型，将红（Red）、绿（Green）、蓝（Blue）三原色的色光以不同的比例相加，以产生多种多样的色光。

GMT 中可以通过 `r/g/b` 的格式来指定 RGB 颜色，其中 `r`、`g`、`b` 的取值范围都是 0 到 255，三者用反斜线 / 分开。

RGB 颜色示例：

- `0/0/0`: 黑色；
- `255/255/255`: 白色；
- `255/0/0`: 红色；
- `0/255/0`: 绿色；
- `0/0/255`: 蓝色；

RGB 颜色除了可以用上面的表示法之外，还可以用 HTML 中常用的表示法 `#RRGGBB`，即分别用两位的十六进制数字表示每个颜色通道，0 对应的 16 进制是 `00`，255 对应的 16 进制是 `FF`。例如 `#000000` 即黑色，`#FF0000` 即红色。十六进制数用大小写表示均可。

4.3.3 HSV

通过 h-s-v 格式指定颜色, 其中 HSV 分别代表色相 (Hue)、饱和度 (Saturation) 和明度 (Value), 三者之间用破折号 - 分开。

- 色相 (H) 是色彩的基本属性, 就是平常所说的颜色名称 (如红色、黄色等), 取值范围为 0 到 360。
- 饱和度 (S) 是指色彩的纯度, 越高色彩越纯, 低则逐渐变灰, 取值范围为 0 到 1。
- 明度 (V) 是色彩的亮度, 取值范围为 0 (dark) 到 1 (light)。

例如 200-0.1-0.1。

4.3.4 CMYK

印刷四分色模式, 是彩色印刷时采用的一种套色模式, 利用色料的三原色混色原理, 加上黑色油墨, 共计四种颜色混合叠加, 形成所谓“全彩印刷”。四种标准颜色是:

- Cyan: 青色, 又称为天蓝色或是湛蓝
- Magenta: 品红色, 又称为洋红色
- Yellow: 黄色
- blacK: 定位套版色 (黑色)

四者的取值范围均为 0 到 1, 用反斜线 / 分开, 例如 0.2/0.3/0.4/0.4。

4.3.5 灰色

灰色是常见的一种颜色, 而灰色又可以根据灰的不同程度细分为不同的灰色。指定灰色的办法很简单, 用一个数值表示灰度即可, 其取值范围为 0 到 255。例如 0 表示黑色, 255 表示白色, 128 表示灰色。

除了用灰度表示之外, 灰色还可以用前面提到的几种形式表示:

1. 用 RGB 表示灰度

灰色本质上就是 R=G=B 的一种颜色。因而 128/128/128 代表灰度为 128, 200/200/200 代表灰度是 200。

2. 用 GMT 颜色名表示灰度

GMT 自定义了多个名字来表示不同程度的灰色。除了前面说过的 gray、lightgray 和 darkgray 之外, 还有 gray0、gray1 一直到 gray100。其中 gray0 即黑色, gray100 即白色。

4.3.6 透明色

在上面提到的几种指定颜色的方法的基础上, 还可以额外指定颜色的透明度。

GMT 中可以通过在颜色后加上 @ 再加上透明度来得到不同程度的透明色。透明度的取值范围是 0 到 100, 0 表示不透明, 100 表示全透明。

例如: red@25、30/25/128@60。

注解: PS 格式不支持透明色。

4.3.7 颜色小结

GMT 中可以用五种方法指定颜色, 分别是:

- 颜色名: red
- RGB 值: 30/25/128 或 #00FA84
- HSV 值: 200-0.1-0.1
- CMYK 值: 0.2/0.3/0.4/0.5
- 灰度: 30

4.4 画笔

有画笔才能画线条、三角形、圆形等各种复杂的形状。

GMT 中的画笔有三个属性: 笔宽、颜色和线型, 三者用逗号分隔, 即:

```
<width>,<color>,<style>
```

在 GMT 模块的语法介绍中, 一般用 `<pen>` 表示画笔属性, 读者在见到 `<pen>` 时应自行脑补成 `<width>,<color>,<style>`。

在指定画笔属性时, 可以指定三个属性中的任意一个或多个属性, 但要保证属性的相对顺序。

4.4.1 画笔线宽

GMT 中可以通过两种方式指定画笔宽度:

1. 宽度值 + 单位, 即 `<width>c|i|p`

在不指定线宽单位时, GMT 使用的默认线宽单位为 p。除了 p 之外, 还可以使用 c 或 i 指定线宽单位为厘米或英寸。

推荐只使用 p 作为线宽单位, 毕竟多数情况下线条的宽度都比较小, 用“小”单位 p 作为单位更方便些。且建议总是显式指定线宽单位, 以使得命令的参数更加易读, 比如 `1p`、`0.25p`。

2. 预定义画笔宽度名

对于一些常用的画笔宽度, GMT 将其定义为特定的名字, 以方便用户使用。下表中列出了 GMT 预定义画笔宽度名及其对应的线宽。

表 4.1: GMT 预定义画笔宽度名

线宽名	线宽	线宽名	线宽
faint	0	thicker	1.5p
default	0.25p	thickest	2p
thinnest	0.25p	fat	3p
thinner	0.50p	fatter	6p
thin	0.75p	fattest	12p
thick	1.0p	obese	18p

注解：指定线宽为 0p 或者 faint 并不表示不绘制线段，而是指定了画笔的宽度为最细宽度。画笔的实际最细宽度由具体的设备来决定。通常情况下，线宽为 0p 的线条在图片缩放时线条宽度总是保持最小值。

4.4.2 画笔颜色

见[颜色](#)一节。

4.4.3 画笔线型

画笔线型属性控制了线条的外观，可以用四种方式表示：

1. 简单符号

- . 表示点线
- - 表示虚线

2. 组合符号

通过对简单符号的任意组合可以获得更多的线型，比如 .- 表示点划线，..- 表示两个点号与一个破折号交替出现。

3. 预定义线型名

GMT 预定义了几种线型名，包括：

- solid: 即实线；
- dashed: 即虚线，等效于 -；
- dotted: 即点线，等效于 ..。

4. 复杂线型

通过 string:offset 的形式可以自定义任意复杂的线型。

string 是一系列由下划线 _ 分隔的数字组成。这一系列数字中，第奇数个数字表示实线的长度，第偶数个数字表示空白的长度。通过实线和空白的长度的不同组合，即可构成多种复杂的线型。offset 表示线段开始处整个线型的初始相位移动。

例如，4_8_5_8:2 表示线型首先是长度为 4p 的实线，然后是长度为 8p 的空白，

紧接着长度为 $5p$ 的实线和长度为 $8p$ 的空白, 然后按照该模式不断重复。此处的 `offset` 值为 $2p$, 因而线段的最开始处, 第一条实线的长度 $4p$ 经过相移后长度为 $2p$ 。

线型中的这些数值, 默认单位是 p , 也可以使用 c 或 i 。

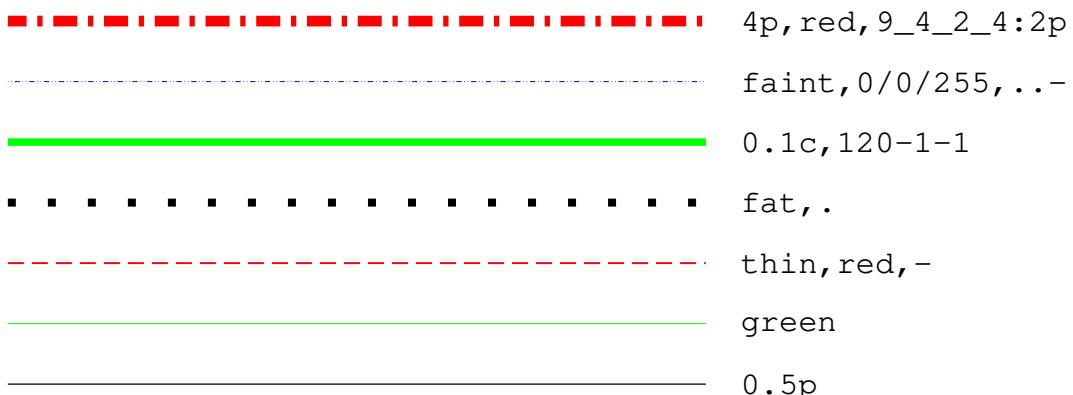
注解: `.` 和 `-` 的绝对长度由画笔宽度来决定。`.` 的长度等于画笔宽度; `-` 的长度为 8 倍画笔宽度; 点线或虚线中段间空白的长度为 4 倍画笔宽度。

4.4.4 小练习

下面的命令用 `psxy` 模块绘制了一条线段。读者可以将命令中 `-W` 选项中的 `<pen>` 修改为不同的值来理解如何指定不同的画笔属性:

```
gmt psxy -R0/10/0/5 -JX10c/5c -W<pen> > pens.ps << EOF
0 2
10 2
EOF
```

Source Code



4.5 填充

用画笔绘制了圆形或多边形之后, 还可以为其填充颜色。在 GMT 模块的语法介绍中, 一般用 `<fill>` 表示需要指定填充属性, 读者在见到 `<fill>` 时应自动联想到本节介绍的内容。

填充 `<fill>` 有两种方式/形式: 填充颜色和填充图案。GMT 的模块中通常使用 `-G<fill>` 选项来填充颜色, 本节以该选项为例。

4.5.1 填充颜色

给区域填充颜色很简单, 直接用 `-G<color>` 即可。颜色在[颜色](#)一节已经介绍过了。比如 `-Gred`、`-G230/200/0`。

4.5.2 填充图案

还可以给区域填充图案 (pattern), 比如地质填图里经常会给不同区域填充不同的图案以区分不同的地质结构。其语法为:

```
-G[p|P]<pattern>[+b<color>][+f<color>][+r<dpi>]
```

<pattern> 有两种取法:

- 取 1 到 90 内的整数, 表示使用 GMT 提供的 [90 种内置位图图案](#)
- 取文件名, 表示使用自定义的 1、8、24 位 Sun 光栅文件作为位图图案

+r<dpi> 设置了位图图案在当前页面上的分辨率。<dpi> 越大, 则区域内位图重复的次数越多。<dpi> 的默认值为 1200。

若使用 -GP 而不是 -Gp, 则图案会发生位反转, 即白色区域变成黑色, 黑色区域变成白色 (仅对 1 位位图或 GMT 预定义位图图案有效)。

对于 GMT 预定义的图案以及用户自定义的 1 位位图来说, 可以用 +b<color> 和 +f<color> 分别设置图案的前景色和背景色, 以分别替换默认的黑色和白色像素点。若设置前景色或背景色为 -, 则视为前景色或背景色为透明。

注解:

1. 由于 PostScript 实现的限制, -G 选项里使用的光栅图片必须小于 146x146 像素; 若要使用更大的图像, 需要使用 [psimage](#) 模块
 2. 若 GMT 在编译时链接了 GDAL, 则也可以使用其他格式的图片。
 3. 在 PostScript Level 1 下, 图案填充是通过使用多边形做路径裁剪实现的。复杂的裁剪路径会需要更多的内存, 因而可能导致某些 PS 解释器由于内存不足而退出。在这种情况下, 建议使用灰度填充区域。
-

4.5.3 小练习

下面使用 [psxy](#) 模块在底图中心绘制了一个直径为 4 厘米的圆, 并填充颜色:

```
echo 5 5 | gmt psxy -JX10c/10c -R0/10/0/10 -Baf -Sc4c -G<fill> > test.ps
```

请尝试将命令中的 <fill> 修改为以下不同的值并查看效果以理解这一节的内容:

- -G128
- -G127/255/0
- -G#00ff00
- -G25-0.86-0.82
- -GDarkOliveGreen1
- -Gp7+r300

- -Gp12+r300
- -Gp7+r1000
- -GP7+r1000
- -Gp7+bred+r300
- -Gp7+bred+f-+r300

4.6 文字

文字, 或者称文本或字符串, 主要由三个属性控制: 文字大小、字体、颜色。三个属性之间用逗号分隔, 即:

```
<size>,<fonttype>,<fill>
```

三者均是可选的, 但先后顺序不可乱。若其中任意一个属性被省略, 则使用该属性的默认值或上一次的设置值。

4.6.1 文字大小

文字大小, 可以理解成字号, 用数字加单位表示。在不指定单位的情况下默认单位为 p, 也可加上 c、p 或者 i 显式指定单位, 比如 15p。

有些中文期刊可以会有类似“图片标题是四号字”这样的要求, 这就需要知道 GMT 中的字体大小与 MS Word 中字号大小的对应关系。如下表所示:

表 4.2: Word 字号与 GMT 中字号对应关系

字号	p	字号	p
初号	42	小初	36
一号	26	小一	24
二号	22	小二	18
三号	16	小三	15
四号	14	小四	12
五号	10.5	小五	9
六号	7.5	小六	6.5
七号	5.5	八号	5

4.6.2 字体

GMT 默认支持 35 种标准字体。下图给出了 GMT 支持的 35 种字体的名字及对应的编号:

Source Code

#	Font Name	#	Font Name
0	Helvetica	17	Bookman-Demi
1	Helvetica-Bold	18	Bookman-Demibold
2	<i>Helvetica-Oblique</i>	19	Bookman-Light
3	Helvetica-BoldOblique	20	<i>Bookman-LightItalic</i>
4	Times-Roman	21	Helvetica-Narrow
5	Times-Bold	22	Helvetica-Narrow-Bold
6	<i>Times-Italic</i>	23	<i>Helvetica-Narrow-Oblique</i>
7	Times-BoldItalic	24	Helvetica-Narrow-BoldOblique
8	Courier	25	NewCenturySchlbk-Roman
9	Courier-Bold	26	<i>NewCenturySchlbk-Italic</i>
10	<i>Courier-Oblique</i>	27	NewCenturySchlbk-Bold
11	Courier-BoldOblique	28	NewCenturySchlbk-BoldItalic
12	Σψμβολ (Symbol)	29	Palatino-Roman
13	AvantGarde-Book	30	<i>Palatino-Italic</i>
14	AvantGarde-BookOblique	31	Palatino-Bold
15	AvantGarde-Demi	32	<i>Palatino-BoldItalic</i>
16	AvantGarde-DemiOblique	33	ZapfChancery-MediumItalic
		34	※□※♦※■※◆※◆▼▲ (ZapfDingbats)

GMT 中可以用字体名（区分大小写）或对应的字体编号来指定字体（比如 Helvetica-Bold 或者 1）。上图中给出了每种字体的字体编号以及字体名称。每个字体名称使用的是自己相对应的字体，所以可以从图中直观地看出不同字体的区别。

小技巧： 使用 `gmt pstext -L` 可以列出 GMT 所支持的字体及其编号。

图中大多数字体都很直观，比较特别的字体有两个，Symbol (12 号) 和 ZapfDingbats (34 号)，前者常用于输出希腊字母，后者则用于输出特殊图案，详情见[特殊字符](#)。

4.6.3 填充色

可以为文字指定颜色或图案，也就是常说的文字颜色，见[填充](#)一节。

4.6.4 描边

在给文字指定填充色的同时，可以在填充色 `<fill>` 后加上 `=<pen>` 来指定文本轮廓（即描边）的画笔属性。`<pen>` 的用法见[画笔](#)一节。比如 `red=2p,blue` 表示将文字填充为红色，并使用宽度为 `2p` 的蓝色线条给文字描边。若填充色 `<fill>` 为 `-`，则不对文字做填充，即实现空心文字的效果。

使用 `=<pen>` 语法绘制文本轮廓时，轮廓线条有一半宽度位于文字外部，另一半宽度会遮住字体。为了避免这一现象，可以使用 `=~<pen>` 语法，此时在绘制文字轮廓时只绘制

文字外部的半个线宽的线条。

4.6.5 小练习

pstext 模块可以用于绘制文字。下面的命令在图上写了文字 TEXT

```
echo 2.5 0.5 TEXT | gmt pstext -R0/5/0/1 -JX15c/2c -F+f<font> > text.ps
```

将其中的 修改为不同的值并查看绘图效果，以理解这一节的内容。

下图给出了一些 的示例：

Source Code

TEXT	30p,5,-=1p,black
TEXT	30p,5,blue=~1p,black
TEXT	30p,5,blue=1p,black
TEXT	30p,8,red
TEXT	30p,Times-Bold
TEXT	40p,blue
TEXT	30p,8
TEXT	30p

4.7 转义序列

前面一节已经介绍了如何设置文字的文字大小、字体和颜色。在此基础上，GMT 提供了转义字符，以实现上标、下标等功能，并可以在一个字符串内随意切换字体、字号和颜色，给文本加入了更丰富的效果。

4.7.1 转义字符

GMT 用 @ 符号实现转义的功能。将 @ 符号与某些字符结合起来，即构成了一系列可以实现特殊效果的转义字符。

GMT 可以识别的转义字符在下表列出：

表 4.3: GMT 转义字符

转义字符	说明
@+	打开/关闭上标
@-	打开/关闭下标
@_	打开/关闭下划线
@#	打开/关闭小型大写字母 (small caps)
@%<fontno>%	切换至另一字体; @%% 重置回前一字体
@:<size>:	切换至另一文本尺寸; @:: 重置回前一尺寸
@;<color>;	切换至另一文本颜色; @;; 重置回前一颜色
@~	打开/关闭 Symbol (12 号) 字体
@!	用接下来的两个字符创建组合字符
@@	输出 @ 符号自身

除此之外, GMT 还为一些常用的欧洲字母专门定义了转义字符。如下表:

表 4.4: 欧洲特殊字母

代码	效果	代码	效果
@E	Æ	@e	æ
@O	Ø	@o	ø
@A	Å	@a	å
@C	Ҫ	@c	ç
@N	Ñ	@n	ñ
@U	Ü	@u	ü
@S	ß	@i	í

4.7.2 小练习

下面给出了一些使用转义字符的示例, 左边 Input 是命令中的输入, 右边 Output 是图上显示的效果:

Source Code

Input	Output
abc@~def@~ghi	abcdδεφghi
2@~p@~r@+2@+h@-0@-	$2\pi r^2 h_0$
S@#mall@# C@#aps@#	SMALL CAPS
Thi is @_underline@_	This is <u>underline</u>
@%1%Use@%% @%23%different@%% @%8%fonts@%%	Use different fonts
@:10:Use@:: @:20:different@:: @:15:size@::	Use different size
@;red;Colorful@;; @;blue;text@;;	Colorful text
@!CV @@	¶ @
Stresses are @~s@~@+*@+@-xx@- MPa	Stresses are σ_{xx}^* MPa

读者可以将下面命令中的 TEXT 和 Label 修改为不同的值, 以体验转义字符的效果:

```
echo 5 2 TEXT | gmt pstext -R0/10/0/3 -Jx1c -Bxaf+l"Label" -BWSen -F+f20p > test.ps
```

4.7.3 注意事项

- 上标/下标不支持嵌套, 即只支持一层上标/下标
- pstext* 命令中有选项可以在文本周围加上文本框, 该选项对转义序列无效
- 转义序列需要成对存在, 与括号类似, 开启转义之后必须关闭转义
- 在 Windows 下, 由于 bat 脚本中 % 表示变量, 因此当你需要在 GMT 中使用百分号时, 应使用 %% 来表示一个百分号, 即 bat 脚本中的 %% 相当于字符 %; 切换字体时 @%%15%% 相当于正常情况下的 @%15%。

4.8 特殊字符

除了键盘上可以直接输入的一般字符之外, GMT 还可以打印一些常用的特殊字符, 如大小写的希腊字母、加减号、大于等于号等一些特殊符号。在 GMT 中, 可以通过输入八进制码的方式指定特殊字符。

4.8.1 八进制码指定特殊字符

GMT 中可以用一个三位的八进制码指定一个特殊字符。八进制码与特殊字符之间的对应关系, 可以通过查询下表得到:

ISOLatin1+									Standard+								
octal	0	1	2	3	4	5	6	7	octal	0	1	2	3	4	5	6	7
\03x	•	…	TM	—	—	fi	ž		\03x	¾	³	TM	²	ý	ÿ	ž	
\04x	!	"	#	\$	%	&	,		\04x	!	"	#	\$	%	&	,	
\05x	()	*	+	,	-	.	/	\05x	()	*	+	,	-	.	/
\06x	0	1	2	3	4	5	6	7	\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?	\07x	8	9	:	;	<	=	>	?
\10x	@	A	B	C	D	E	F	G	\10x	@	A	B	C	D	E	F	G
\11x	H	I	J	K	L	M	N	O	\11x	H	I	J	K	L	M	N	O
\12x	P	Q	R	S	T	U	V	W	\12x	P	Q	R	S	T	U	V	W
\13x	X	Y	Z	[\`]	^	-	\13x	X	Y	Z	[\`]	^	-
\14x	‘	a	b	c	d	e	f	g	\14x	‘	a	b	c	d	e	f	g
\15x	h	i	j	k	l	m	n	o	\15x	h	i	j	k	l	m	n	o
\16x	p	q	r	s	t	u	v	w	\16x	p	q	r	s	t	u	v	w
\17x	x	y	z	{		}	~	š	\17x	x	y	z	{		}	~	f
\20x	Œ	†	‡	Ł	/	⟨	ſ	⟩	\20x	Ã	Ҫ	Đ	Ł	Ñ	Ӯ	ӹ	پ
\21x	œ	Ŷ	Ž	ł	%o	”	“	”	\21x	Ý	Ŷ	Ž	ã	ı	ç	©	°
\22x	ı	‘	‘	^	~	-	ˇ	˙	\22x	÷	ð	¬	ł	–	μ	×	ñ
\23x	”	,	◦	›	‘	”	‘	ˇ	\23x	½	¼	¹	ő	±	®	š	þ
\24x	í	¢	£	¤	¥	¦	§		\24x	í	¢	£	/	¥	f	§	
\25x	”	©	a	«	¬	-	®	-	\25x	¤	’	“	«	⟨	⟩	fi	fl
\26x	◦	±	²	³	‘	µ	¶	·	\26x	Á	—	†	‡	·	Â	¶	•
\27x	›	ı	◦	»	¼	½	¾	¸	\27x	,	”	”	»	…	%o	Ä	¸
\30x	À	Á	Â	Ã	Ä	Å	Æ	Ç	\30x	À	‘	‘	^	~	-	ˇ	˙
\31x	È	É	Ê	Ë	Ì	Í	Î	Ï	\31x	”	É	◦	›	Ê	”	‘	ˇ
\32x	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	\32x	—	Ë	È	Í	Î	Ï	Ì	Ó
\33x	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	\33x	Ô	Ö	Ò	Ú	Û	Ü	Ù	á
\34x	à	á	â	ã	ä	å	æ	ç	\34x	â	Æ	ä	ª	à	é	ê	ë
\35x	è	é	ê	ë	ì	í	î	ï	\35x	è	Ø	Œ	º	í	î	ï	ì
\36x	ð	ñ	ò	ó	ô	õ	ö	÷	\36x	ó	æ	ô	ö	ò	í	ú	û
\37x	ø	ù	ú	û	ü	ý	þ	ÿ	\37x	ü	ø	œ	þ	ù	å	å	ÿ

图 4.1: ISOLation1+ 和 Standard+ 编码下的八进制码

下载: 八进制码表 PDF 版 | 八进制码表 PNG 版

左右两张表分别是什么？该用哪一张 GMT 支持多种字符编码，其中最常用的编码是 Standard+ 和 ISOLation1+。通常，GMT 的默认字符编码是 ISOLatin1+，所以只需要看左表即可；如果当前字符编码是 Standard+，则需要查询右表。可以通过 gmt get PS_CHAR_ENCODING 命令查到 GMT 当前的默认字符编码方式。

这张表如何查询呢？以 ISOLation1+ 编码为例，假如想要输入加减号 ±，查左表可知，加减号位于第 \26x 行、第 1 列，因而其八进制码为 \261。而如果你当前使用的是 Standard+ 编码，则对应的八进制码则是 \234。因而，当你需要输入某个特殊字符时，只需要输入其对应的八进制码即可。

4.8.2 特殊字体下的字符

如果文本使用了 12 号字体 (Symbol) 或 34 号字体 (ZapfDingbats)，则此时八进制码具有不同的含义。需要查询下表：

Symbol									ZapfDingbats								
octal	0	1	2	3	4	5	6	7	octal	0	1	2	3	4	5	6	7
\04x	!	\forall	#	\exists	%	&	\ni		\04x		\bowtie	\bowtie	\bowtie	\bowtie	\bowtie	\bowtie	\bowtie
\05x	()	*	+	,	-	.	/		\05x	\blacktriangleleft	\square	\blacktriangleleft	\square	\blacktriangleleft	\square	\blacktriangleleft	\square
\06x	0	1	2	3	4	5	6	7	\06x	\blacktriangleright	\bowtie	\bowtie	\checkmark	\checkmark	\times	\times	\times
\07x	8	9	:	;	<	=	>	?	\07x	\times	\oplus	\oplus	\oplus	\oplus	\dagger	\dagger	\dagger
\10x	\cong	A	B	X	Δ	E	Φ	Γ	\10x	\divideontimes	\star	\star	\star	\star	\diamond	\diamond	\diamond
\11x	H	I	\eth	K	Λ	M	N	O	\11x	\star	\star	\star	\star	\star	\star	\star	\star
\12x	Π	Θ	P	Σ	T	Y	ς	Ω	\12x	\star	$*$	$*$	$*$	$*$	$*$	$*$	$*$
\13x	Ξ	Ψ	Z	[\ddots]	\perp	-	\13x	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
\14x	-	α	β	χ	δ	ε	ϕ	γ	\14x	\circledast	\circledast	\circledast	$*$	$*$	$*$	$*$	$*$
\15x	η	ι	φ	κ	λ	μ	ν	\circ	\15x	$*$	$*$	$*$	$*$	\bullet	\circ	\blacksquare	\blacksquare
\16x	π	θ	ρ	σ	τ	υ	ω		\16x	\square	\square	\square	\blacktriangle	\blacktriangledown	\blacklozenge	\blacklozenge	\blacklozenge
\17x	ξ	ψ	ζ	{		}	\sim		\17x				'	,	"	"	
\24x	\texteuro	\textgamma	'	\leq	$/$	∞	f	\clubsuit	\24x	g	:	\heartsuit	\heartsuit	\heartsuit	\heartsuit	\heartsuit	\heartsuit
\25x	\spadesuit	\heartsuit	\clubsuit	\leftrightarrow	\leftarrow	\uparrow	\rightarrow	\downarrow	\25x	\clubsuit	\spadesuit	\heartsuit	\spadesuit	①	②	③	④
\26x	\circ	\pm	"	\geq	\times	\propto	∂	\bullet	\26x	⑤	⑥	⑦	⑧	⑨	⑩	①	②
\27x	\div	\neq	\equiv	\approx	\cdots		\dashv		\27x	③	④	⑤	⑥	⑦	⑧	⑨	⑩
\30x	\aleph	\beth	\aleph	\wp	\otimes	\oplus	\emptyset	\cap	\30x	①	②	③	④	⑤	⑥	⑦	⑧
\31x	\cup	\supset	\supseteq	\subset	\subset	\subseteq	\in	\notin	\31x	⑨	⑩	①	②	③	④	⑤	⑥
\32x	\angle	∇	\circledR	\circledC	\textTM	\prod	\vee	.	\32x	⑦	⑧	⑨	⑩	\rightarrow	\rightarrow	\leftrightarrow	\uparrow
\33x	\neg	\wedge	\vee	\Leftrightarrow	\Leftarrow	\Rightarrow	\Downarrow		\33x	\nwarrow	\rightarrow	\nearrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow
\34x	\diamond	\langle	\circledR	\circledC	\textTM	Σ	()	\34x	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow
\35x	(\lceil		L	\lceil	{	\lfloor		\35x	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow
\36x)	\rceil		J)			\36x	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow
\37x	J			J			J		\37x	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	\Rightarrow	

图 4.2: Symbol 和 Pifont 字体八进制码

当文本字体为 12 号字体 (Symbol 字符集) 时，应查询左表；当文本字体为 34 号字体 (Pifont ZapfDingbats) 时，应查询右表。

下载：[特殊字体八进制码 PDF 版](#) | [特殊字体八进制码 PNG 版](#)

4.8.3 特殊字符小结

GMT 中特殊字符可以通过八进制码的方式输入。想要输入特殊字符，首先需要找到该字符对应的八进制码。从上面四张表中找到自己需要的特殊符号的八进制码及其位于哪张表内。总共分四种情况：

1. 字符在 ISOLatin1+ 表内：可以直接使用该字符对应的八进制码
2. 字符在 Standard+ 表内：该表内的字符是与 ISOLatin1+ 中的字符完全重合的，

建议在 ISO Latin1+ 表中找到自己需要的字符，然后使用其对应的八进制码

3. 字符在 Symbol 表内：将文本字体设置为 12 号字体，并使用该字符对应的八进制码
4. 字符在 ZapfDingbats 表内：将文本字体设置为 34 号字体，并使用该字符对应的八进制码

4.8.4 小练习

使用 ISO Latin1+ 表中的八进制码输入字符 ± 234：

```
echo 5 2 \261 234 | gmt pstext -Jx1c -R0/10/0/4 -Baf > test.ps
```

试验之后会发现，图上显示的是 261 234 而不是 ± 234。产生这一现象的原因是因为反斜杠在很多脚本语言里有特殊的含义，此处反斜杠首先被 bash 所解释，真正传给 GMT 的是 5 2 261 234。针对这种情况，有四种解决办法：

```
# 1. 使用两个反斜杠
echo 5 2 \\261 234 | gmt pstext -Jx1c -R0/10/0/4 -Baf > test.ps

# 2. 使用单引号将字符串括起来
echo 5 2 '\261 234' | gmt pstext -Jx1c -R0/10/0/4 -Baf > test.ps

#. 3. 使用 EOF 输入字符串
gmt pstext -JX10c/10c -R0/10/0/10 -F+f15p,12 > test.ps << EOF
5 5 \141
EOF

#. 4. 从文本文件中读取字符串
gmt pstext input.txt -JX10c/10c -R0/10/0/10 -F+f15p,12 > test.ps
```

输入希腊字母，此时需要指定文本使用 12 号字体（即 -F+f15p,12 选项）：

```
echo 5 5 '\141' | gmt pstext -JX10c/10c -R0/10/0/10 -F+f15p,12 > test2.ps
```

如果需要在一堆正常的字符中使用 Symbol 表中的字符，可以使用转义字符 @~ 或 @%12% 临时将字体 Symbol 字体，例如：

```
gmt psbasemap -R0/10/0/10 -JX10c/10c -BWSne -Bx2+l"T@~\161@~t" > test.ps
```

4.9 绘制矢量/箭头

矢量是一个有大小和方向的量。

GMT 中能够绘制矢量的模块包括 `psxy`、`psxyz`、`grdvector` 和 `psvelo` 等。除此之外，大多数简单的线段也可以加上矢量箭头。

GMT 中绘制的矢量都可以分为两个独立的部分：

1. 矢量线
2. 矢量箭头

4.9.1 矢量分类

GMT 中的矢量可以分为三类：

- 笛卡尔矢量：矢量线是直线。可以通过如下几种方式指定
 1. 给定起点坐标和终点坐标
 2. 给定起点坐标、方向和长度（长度单位可以取 c|i|p）
 3. 给定起点坐标、方位角和长度（长度单位为 km）
- 弧形矢量：矢量线是圆弧。通过给定圆心坐标、半径以及圆弧起始和结束的角度来指定
- 地理矢量：矢量线是大圆弧路径。可以通过两种方式指定
 1. 给定起点坐标和终点坐标
 2. 给定起点坐标、方位角和长度（单位为 km）

以 psxy 模块为例，-Sv 或 -SV 用于绘制笛卡尔矢量，-Sm 用于绘制弧形矢量，-S= 用于绘制地理矢量。

下面的例子分别绘制了三种矢量箭头：

```
#!/bin/bash
# 笛卡尔矢量
echo 0.5 1.5 4.5 1.5 | gmt psxy -R0/5/0/5 -JX1.75i -Sv0.2i+s+b+e -W1.5p -Gred -K > GMT_arrows.ps

# 弧形矢量
echo 0.5 0.5 0.9i 0 90 | gmt psxy -R -J -Sm0.2i+b+e -W1.5p -Gred -X2i -K -O >> GMT_arrows.ps

# 地理矢量
echo 10 -35 90 8000 | gmt psxy -R0/90/-41.17/41.17 -JM1.75i -S=0.2i+b+e -W1.5p -Gred -Ox -X2i >> GMT_arrows.ps

rm gmt.*
```

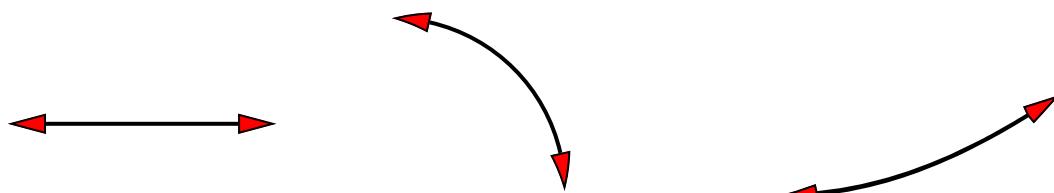


图 4.3: GMT 中的三种矢量箭头

(左) 笛卡尔矢量；(中) 弧形矢量；(右) 地理矢量

4.9.2 矢量箭头属性

注解: 读者可以修改上面示例中的 -S 选项后的参数并查看效果来理解接下来的内容。

绘制矢量时，默认只有矢量线没有矢量头，可以使用下面几个子选项为矢量加上矢量头：

- **+b[t|c|a|i|A|I][l|r]**: 在矢量线的首端加上矢量头
 - **t|c|a|i|A|I** 用于控制矢量头的形状，默认是 **a**
 - * **t** 矢量头为终点线（垂直于矢量线的短线）
 - * **c** 矢量头为圆圈
 - * **a** 矢量头为箭头
 - * **i** 矢量头为箭尾
 - * **A** 矢量头为简单箭头
 - * **I** 矢量头为简单箭尾
 - **l|r** 表示只绘制矢量的头的左半边或右半边（默认两边都绘制）
- **+e[t|c|a|i|A|I][l|r]**: 在矢量线的尾端加上矢量头。其他同上
- **+m[f|r][t|c|a][l|r]**: 在矢量线的中间加上矢量头。
 - **f|r** 表示矢量头沿着正方向或逆方向（默认值从首端指向尾端）
 - 其他同上

注解: **+m** 不能与 **+b** 或 **+e** 一起使用

下面的子选项可以进一步控制矢量头的外观：

- **+a<angle>** 控制矢量箭头顶端的夹角，默认值为 30 度
- **+l** 表示只绘制左半个箭头
- **+r** 表示只绘制右半个箭头
- **+g<fill>** 设置箭头填充色，**<fill>** 为 - 表示不填充，默认使用 **-G** 指定的填充色
- **+p[-] [<pen>]** 设置矢量的画笔属性，**-<pen>** 表示不绘制矢量箭头的轮廓，默认使用 **-W** 所指定的画笔属性

还有其他一些子选项：

- **+n<norm>** 默认情况下，矢量头的大小不随着矢量线的长度变化而变化，这可能会出现矢量长度太小时矢量头过大的情况。该选项使得矢量长度小于 **<norm>** 时，矢量头的属性（画笔宽度，箭头大小）会根据矢量长度按照 **length/<norm>** 缩放
- **+t[b|e]<trim>** 将矢量的首端或尾端偏移一定长度，**<trim>** 为正值表示矢量长度变短，为负值表示矢量长度变长。还可以使用 **+t<trim1>/<trim2>** 语法分别为首端和尾端指定偏移量。正值表示将矢量线缩短。
- **+o<plon>/<plat>** specifies the oblique pole for the great or small circles. 仅当与

`+q` 一起使用时有效

- `+q` 表明输入数据中的 `angle` 和 `length` 被解释为矢量的开始和结束的角度

除了弧形矢量外，其他矢量还可以使用如下子选项：

- `+j [b|c|e]` 默认情况下，输入数据中的 XY 坐标会作为矢量的首端坐标，该选项可以修改这一行为，`b|e|c` 分别代表输入数据中的 XY 坐标作为矢量的首端、尾端和中间
- `+s` 表明输入数据中的 `angle` 和 `length` 被解释为矢量的尾端 XY 坐标

对于笛卡尔矢量而言，还可以使用：

- `+z<scale>[<unit>]` expects input `dx,dy` vector components and uses the `scale` to convert to polar coordinates with length in given unit.

除此之外，GMT 配置参数中 `MAP_VECTOR_SHAPE` 也可以用于修改箭头的外观。

4.9.3 矢量示例

下图展示了部分 GMT 可以绘制的矢量，每个矢量右边给出了其对应的选项。需要注意的是，某些矢量设置了 `MAP_VECTOR_SHAPE` 的值为 0.5。

Source Code

	<code>v0.25i+s+bA+eA</code>
	<code>v0.25i+s+b+ei</code>
	<code>v0.25i+s+b+ec+a45</code>
	<code>v0.25i+s+e+bt</code>
	<code>v0.25i+s+et+bt</code>
	<code>v0.25i+s+bc</code>
	<code>v0.25i+s+et</code>
	<code>v0.25i+s+el+bl</code>
	<code>v0.25i+s+e+bi</code>
	<code>v0.25i+s+b+a45</code>
	<code>v0.25i+s+e+b</code>
	<code>v0.25i+s+e+b</code>
	<code>v0.25i+s+b</code>
	<code>v0.25i+s+e+a40</code>

4.10 线条属性

[画笔](#) 一节介绍了画笔的三个属性：线宽、颜色以及线型。对于线条而言，除了画笔的这三个属性之外，某些模块还可以为线条设置额外的线条属性，这些额外的属性可以通过在画笔属性后加上子选项来实现。

线条的额外属性包括：端点偏移量、线条平滑和端点箭头。

4.10.1 端点偏移量

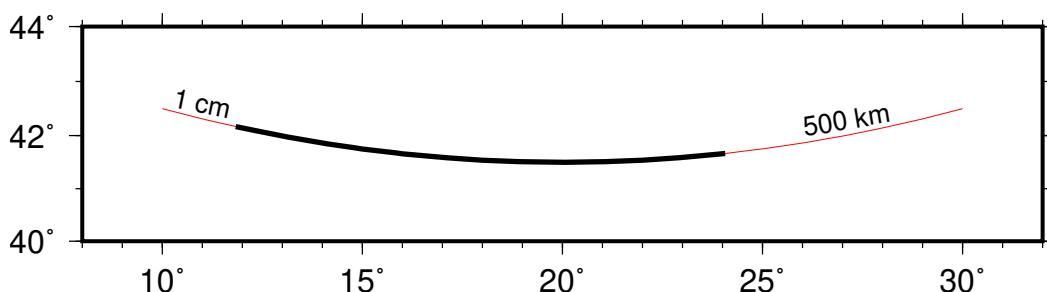
在给定若干个数据点绘制线条时，一般都是从起点（第一个点）一直画到终点（最后一个点）。可以使用 `+o<offset>[<u>]` 子选项为线段两端指定偏移量，使得绘制线段时的起点和终点与输入数据中指定的起点和终点间存在一定的偏移量。该子选项的语法是：

```
+o<offset>[<u>]
```

- 若只给了一个 `<offset>`，则表示起点和终点共用同一个偏移量
- 也可以用 `<offset>/<offset>` 分别为起点和终点指定不同的偏移量
- 对于每个偏移量，都可以使用长度单位或距离单位

图中，细线和粗线使用了完全相同的输入数据，其中细线没有使用 `+o` 选项的结果，此时线段的起点和终点与数据指定的点重合，粗线在绘制线条时使用了 `-W2p+o1c/500k` 选项，即在起点处偏移 1 厘米，在终点处偏移 500 千米。

Source Code

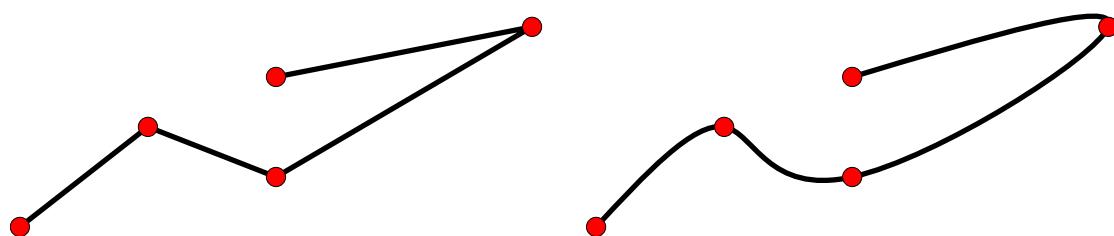


4.10.2 线条平滑

通常情况下，在绘制线条时，两点之间是用直线连接的（地图上两点之间默认用大圆弧连接）。使用 `+s` 子选项会使用 Bezier splines 在两点之间做样条插值以得到更光滑的曲线。

下图中，左图使用了 `-W2p` 选项，右边使用了 `-W2p+s`。

Source Code



4.10.3 端点箭头

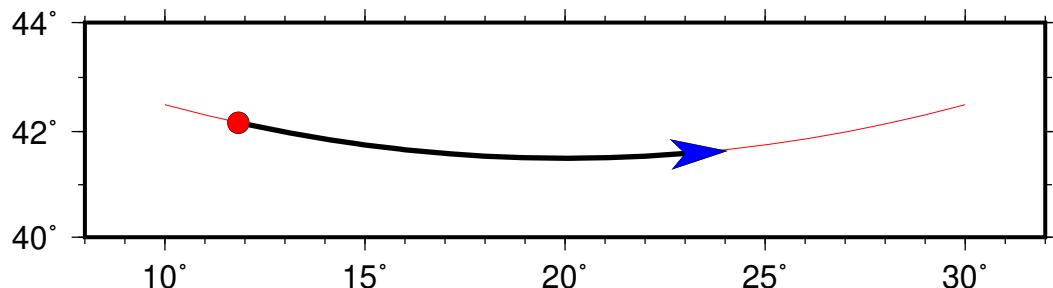
默认情况下，在绘制线段时，线段的两个端点并没有什么特别的。使用 `+v` 子选项，可以在线段的一端或两端绘制一个指向端点的箭头。该子选项的语法为：

```
+v[b|e]<vspecs>
```

- 默认会在线段两端都加上箭头, `b|e` 表示只绘制开头或结尾的箭头
- 箭头属性 `<vspecs>` 的细节参考[绘制矢量/箭头](#)一节的内容

下图中细线是通常绘制的线段, 粗线使用的选项是 `-W2p+o1c/500k+vb0.2i+gred+pfaht+bc+ve0.3i+gblue`。

Source Code



4.10.4 其他属性

除了上面提到的属性之外, GMT 中还有一些参数可以影响线段的外观。这些参数包括:

- `PS_LINE_CAP`: 控制线段顶端的绘制方式
- `PS_LINE_JOIN`: 控制线段拐点/交点的绘制方式
- `PS_MITER_LIMIT`: 控制线段拐点在 `miter` 模式下的阈值

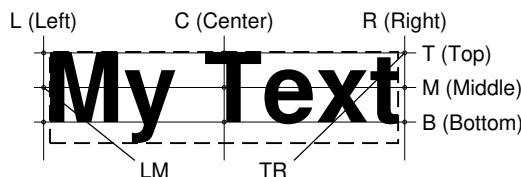
4.11 锚点

要将一个矩形物体放在指定的坐标处, 该怎么放呢? 是把矩形中心放在指定坐标处? 还是把矩形左上角放在坐标处? 最好能够控制将矩形的任意一点放在指定坐标处。

GMT 提供了这样一种机制, 可以很方便地将矩形的任意一点放在指定坐标处。这里所说的矩形, 并不是一个真正的矩形, 而是一串文本、比例尺、指南针、颜色条等等。这些东西都可以被抽象成一个矩形, 这一节仅以字符串为例, 介绍 GMT 中矩形的锚点。

任意一个字符串, 都有一个将其包起来的矩形。如图所示:

Source Code



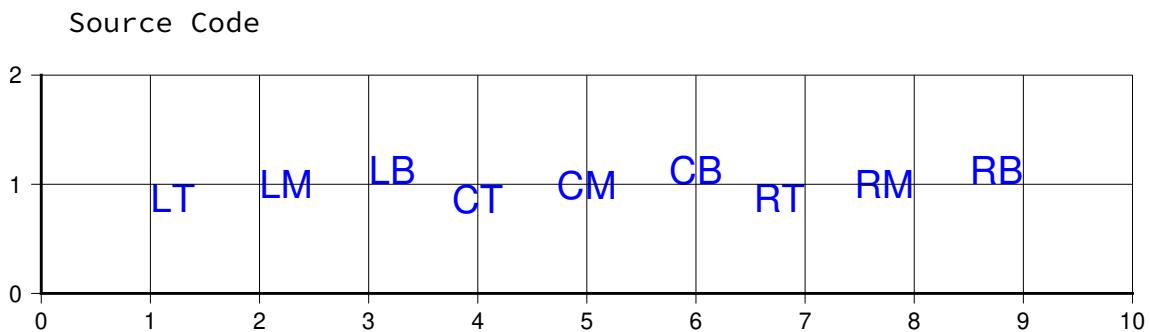
如上图所示, 在水平方向上, 定义 `L`、`C`、`R` 三个水平位置代码, 表示左、中、右三个水

平位置；在垂直方向上，定义 T、M、B 三个垂直位置代码，代表上、中、下三个垂直位置。

注解： 图中 B 代表的是字符 M 的底部，而不是字符 y 的底部，这在印刷行业里称为 baseline。

将任意一个水平位置代码与任意一个垂直位置组合起来，就构成了矩形的 9 个锚点（图中的 9 个圆圈），分别是 LT、LM、LB、CT、CM、CB、RT、RM、RB。先后顺序不重要，LT 和 TL 是一个意思。

要指定一串文本相对于特定坐标对齐方式，可以通过指定将文本的某个锚点放在特定坐标处来实现。下图展示了 9 种文本对齐方式的效果。图中将九个字符串按照相应的对齐方式放在坐标 (1,1)、(2,1) 到 (9,1) 处，即将字符 LT 的 LT 锚点放在 (1,1) 处，其他同理。



从图中可以直观得看到，CM 对齐方式将字符串的中心放在了指定坐标处，LT 对齐方式将字符串的左上角放在了指定坐标处，其他同理。

4.12 绘制修饰物

GMT 除了可以绘制常规的数据外，还可以绘制 8 种修饰物：

1. 地图比例尺 (`psbasemap -L pscoast -L`)
2. 方向玫瑰图 (`psbasemap -Td`)
3. 磁场玫瑰图 (`psbasemap -Tm`)
4. 颜色条 (`psscale`)
5. 图例 (`pslegend`)
6. Image overlay (`psimage`)
7. GMT logo (`gmtlogo`)
8. map inset (`psbasemap -D`)

8 种修饰物的绘制有各自不同的语法，但这些修饰物都可以抽象为一个矩形。对于这些抽象的矩形，GMT 设计了一套非常通用的机制，来决定这些修饰物在图上的位置，同时为这些矩形设计了统一的可定制的背景面板。

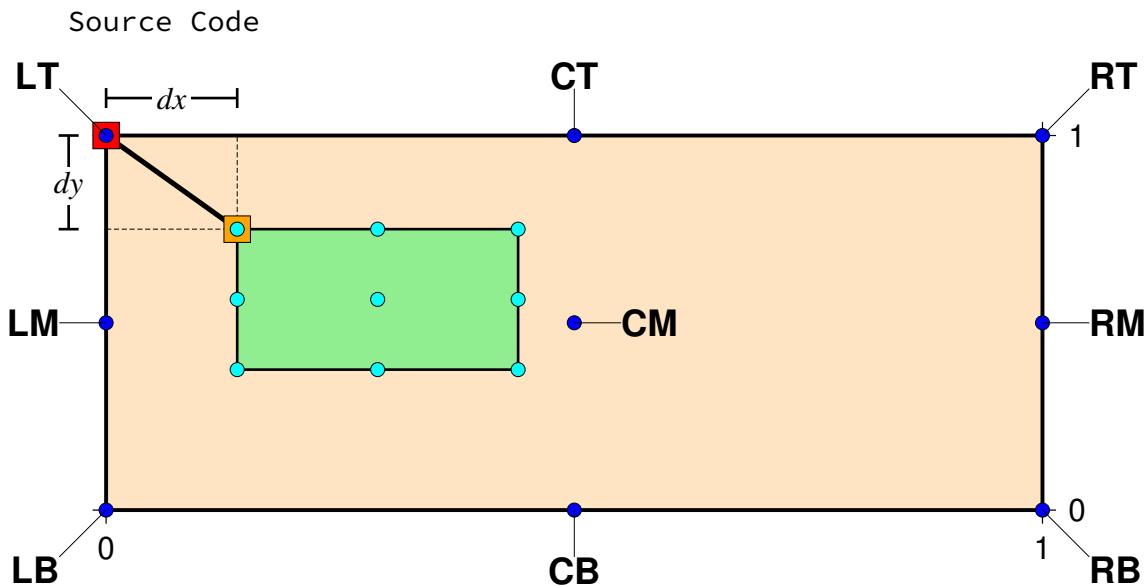
4.12.1 定位

GMT 通过 `-D[g|j|J|n|x]<refpoint>[+j<justify>][+o<dx>[/<dy>]]` 选项对修饰物进行定位。

GMT 为了将一个修饰物放在底图上, 做了四件事:

1. `[g|j|J|n|x]<refpoint>`: 在底图上指定一个参考点
2. `+j<justify>` 在修饰物上指定一个锚点
3. `+o<dx>[/<dy>]` 指定修饰物锚点的额外偏移量
4. 将底图参考点与偏移后的修饰物锚点对齐

下图中, 大矩形代表底图, 小矩形代表修饰物。



注解: 一个比较形象的比喻: 将底图看作是一片海洋, 将修饰物看作是一条小船。小船航行到海洋中的某个参考点, 然后将船上的一只锚(按照 GMT 的定义, 这条船有 9 只锚)从参考点处丢下, 即实现了船在海洋中的定位。

底图参考点

GMT 提供了 5 种方式用于指定底图的参考点, 5 种方法提供了极高的自由度, 使得用户可以指定底图内/外的任意一点作为参考点。其语法为:

```
[g|j|J|n|x]<refpoint>
```

1. `x` 用绘图坐标指定参考点, 即相对于绘图原点的偏移距离。例如 `x2.75i/2c`, 这种方法类似于使用 `-X` 和 `-Y` 选项
2. `n` 用归一化坐标指定参考点, X 方向和 Y 方向的坐标范围都是 0 到 1。比如 `n0.2/0.1`

3. g 用数据坐标指定参考点, 比如 `g135W/20N` 表示参考点是西经 135 度北纬 20 度。
当需要将修饰物放在某个特定坐标值时, 建议使用该方式
4. j 指定底图的 9 个锚点 中的一个作为参考点。与此同时, 也设置了修饰物默认使用相同的锚点。比如 `jBL` 指定底图左下角为参考点, 同时也指定了修饰物的锚点为 `BL`, 因而底图的左上角将与修饰物的左上角重合, 进而导致修饰物位于底图左上角的内部。当需要将修饰物放在底图内部某个角落、某条边的中心或图的中心时, 建议使用 `j`, 最常见的是绘制图例。
5. J 与 `j` 类似, 也是指定底图的某个锚点 为参考点。其与 `j` 的区别在于, 其同时设置了修饰物默认使用与之镜像相反的锚点。比如 `JTL` 将底图的左上角作为参考点, 同时修饰物的锚点是 `BR` 即右下角, 因而底图的左上角与修饰物的右下角重合, 进而导致修饰物位于底图左上角的外部。当需要将修饰物放在底图的外部时, 建议使用 `J`, 最常见的是绘制 colorbar

若未指定使用何种方式, 则默认使用 `x` 指定参考点。

修饰物锚点

为底图指定参考点之后, 还需要为修饰物指定锚点。修饰物的锚点用 `+j<anchor>` 来指定。锚点有 9 个, 见锚点 一节。

在未使用 `+j<anchor>` 指定锚点的情况下, 修饰物的锚点按如下规则取默认值:

1. 若使用 `j<refpoint>` 指定底图参考点, 则修饰物锚点与底图锚点相同
2. 若使用 `J<refpoint>` 指定底图参考点, 则修饰物锚点使用与底图参考点镜像相反的锚点
3. 若使用其他三种指定参考点的方式, 对于玫瑰图和比例尺而言, 锚点默认为 `MC`, 其他修饰物锚点默认为 `BL`

修饰物锚点偏移量

指定了修饰物锚点后, 还可以为其指定额外的偏移量。尤其是在使用 `j` 和 `J` 指定底图参考点时, 由于底图只能指定某个锚点作为参考点, 因而就需要为锚点指定额外的偏移量以增加定位的灵活性。

可以使用 `+o<dx>/<dy>` 选项指定参考点的额外偏移量。偏移量为正值表示沿着与指定锚点所使用的对齐代码同一方向作偏移。

比如使用 `jTL` 指定底图左上角为参考点, 同时修饰物的左上角锚点也被选中, 此时使用 `+o2c/1c` 表示将修饰物的左上角锚点向左移动 2 cm、向上移动 1 cm, 最后将底图参考点与偏移后的锚点对齐。这一特性常用在绘制图例中。

4.12.2 背景面板

解决了修饰物的精确定位后, GMT 还为修饰物设计了背景面板。背景面板是与修饰物相关联的, 因而背景面板的位置和大小由修饰物决定。除此之外, 背景面板还有一些其他属性。背景面板的属性由 `-F` 选项的子选项决定:

```
-F[+c<clearance(s)>][+g<fill>][+i[[<gap>/]<pen>]][+p[<pen>]][+r[<radius>]][+s[<dx>/<dy>/<fill>]]
```

1. `+g<fill>` 指定面板填充色, 默认不填充
2. `+p<pen>` 绘制面板边框。`<pen>` 为边框的画笔属性, 若不指定 `<pen>` 则默认使用默认使用 `MAP_FRAME_PEN` 的值
3. `+r<radius>` 绘制圆角边框, `<radius>` 为圆角的半径
4. `+i<gap>/<pen>` 在边框内部绘制一个内边框, `<gap>` 是内外边框的空白距离 (默认值为 `2p`), `<pen>` 为内边框的画笔属性, 比如 `+i0.1c/thin,dashed`, 默认使用 `MAP_DEFAULT_PEN`
5. `+c<clearance>` 设置修饰物与面板之间的空白距离。默认情况下面板的大小由修饰物的大小决定, 可以使用该子选项为面板增加额外的尺寸:
 - `+c<gap>` 为四个方向增加相同的空白距离
 - `+c<xgap>/<ygap>` 分别为 X 方向和 Y 方向指定不同的空白距离
 - `+c<lgap>/<rgap>/<bgap>/<tgap>` 分别为四个方向指定不同的空白距离
6. `+s<dx>/<dy>/<fill>` 下拉阴影区。`<dx>/<dy>` 是阴影区相对于面板的偏移量, `<shade>` 是阴影区的颜色, 默认值为 `4p/-4p/gray50`。

Source Code

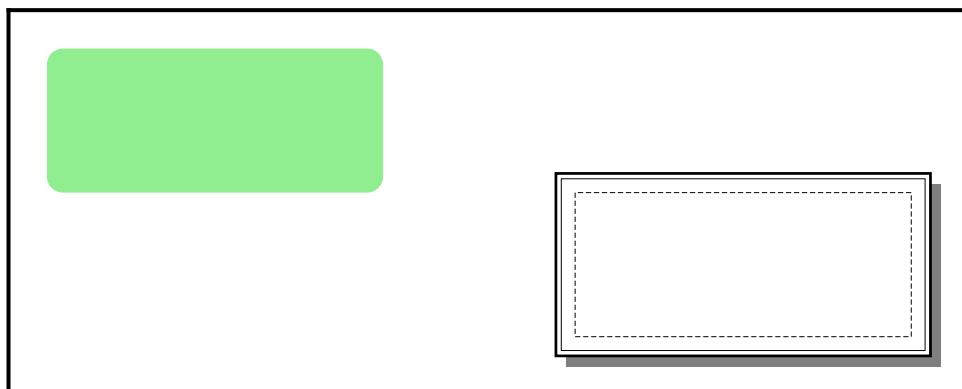


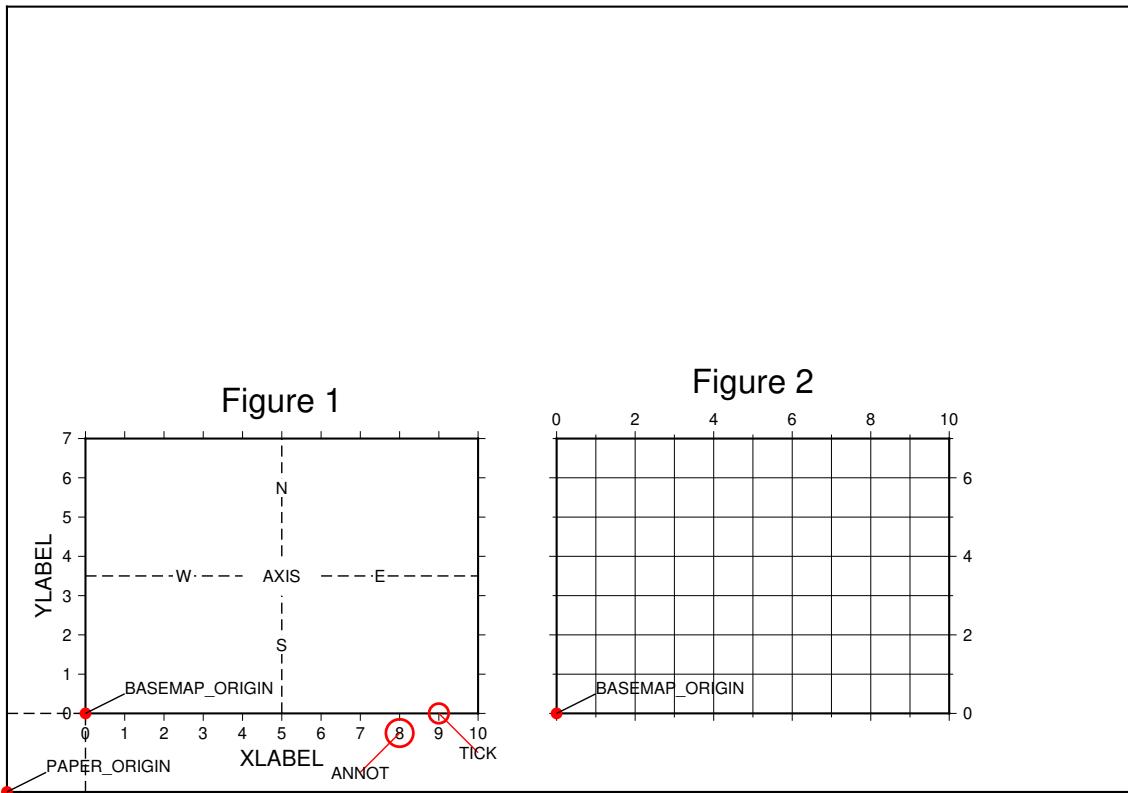
图 4.4: GMT 修饰物背景面板

左图使用了 `-F+glightgreen+r`, 右图使用了 `-F+p1p+i+s+gwhite+c0.1i`

4.13 词汇表

在 GMT 中有不少与绘图有关的专业词汇。这些英文词汇没有固定的中文翻译, 因而这里统一给出本文所使用的中文翻译, 并在图中指出这些词汇的含义。

Source Code



纸张原点 (PAPER_ORIGIN) 一张纸, 无论横着放还是竖着放, 纸张的原点都位于纸张的最左下角。图中最外部的黑色边框代表纸张的边界。左下角的红点即纸张原点的位置。

底图 (BASEMAP) 底图是一个抽象的概念, 可以认为图 1 中黑色边框就是一个底图。

底图原点 (BASEMAP_ORIGIN) 底图原点位于每一张底图的左下角, 如图中红点所示。通过 $-X$ 和 $-Y$ 选项可以控制底图原点相对于纸张原点的位置。

坐标轴 (AXIS) 每个底图都有四条边, 分别称为 W、E、S、N。四条边可以显示也可以不显示。

标签 (LABEL) 标签用于表明每个坐标轴的含义, 即图中的 XLABEL 和 YLABEL。

标题 (TITLE) 每张底图都可以有一个标题, 标题默认位于底图的上方, 即图中 Figure 1 所在位置。

标注 (ANNOT) 如图中所示, 即每个坐标轴上的数字

刻度 (TICK) 如图所示, 刻度指每个坐标轴上的刻度线

网格线 (GRID) 如图 2 所示。

子图 在一张纸上可以绘制多个底图, 每个底图称为一个子图, 通过使用 $-X$ 和 $-Y$ 选项可以调整每个子图相对于纸张原点的位置。图中图 1 和图 2 是整张图的子图。

第 5 章 输入与输出

GMT 每个模块都有很多输入和输出。总的来说，输入输出可以分为如下几类：

1. 1D 表数据
2. 2D 网格数据
3. CPT 文件
4. 警告和错误消息
5. 退出状态码

5.1 数据格式

GMT 作为一个画图工具，肯定要输入一些数据，同时作为一个数据处理工具，也会输出一些数据。这就牵涉到数据格式的问题了。

GMT 可以绘制一般的笛卡尔坐标轴、地图的经纬度轴以及绝对时间轴、相对时间轴。对于不同的坐标轴，需要的数据格式也不同。

5.1.1 地理坐标

地理坐标（即经纬度）有两种表示方式：

1. 以浮点型的度数表示。比如 -123.45 代表-123.45 度
2. 度分秒表示：

```
[±]ddd[:mm[:ss[.xxx[W|E|S|N]]]]
```

其中，**ddd**、**mm**、**ss**、**xxx** 分别表示弧度、弧分、弧秒、弧毫秒。**W**、**E**、**S**、**N** 分别代表西经、东经、北纬、南纬。例如 **123:27W** 代表西经 123 度 27 分，**123:27:15.120W** 表示西经 123 度 27 分 15.12 秒。

5.1.2 绝对时间坐标

绝对时间由两部分构成，即日期和时间，表示为：

```
<date>T<clock>
```

其中 **T** 是关键字，用于分隔日期和时间。

日期 **<date>** 可以是如下格式的一种：

1. **yyyy[-mm[-dd]]**：年-月-日，例如 **2013**、**2015-10**、**2015-01-02**
2. **yyyy[-jjj]**：年-一年中的第几日，例如 **2015-040**

3. `yyyy[-Www[-d]]`: 年-第几周-该周内第几天, 例如 `2014-W01-3`、`2014-W01`

时间 `<clock>` 是 24 小时制, 其格式为:

```
hh:[mm:[ss[.xxx]]]
```

例如 `10:10:35.120`。

使用过程中需要注意:

1. GMT 的时间数据的输入/输出格式默认为 `yyyy-mm-ddThh:mm:ss.xxx`。若想要输入其他格式的时间数据, 需要修改 `FORMAT_DATE_IN` 和 `FORMAT_CLOCK_IN`; 若想要输出其他格式的时间数据, 需要修改 `FORMAT_DATE_OUT` 和 `FORMAT_CLOCK_OUT`
2. 若未指定 `<date>` 则假定 `<date>` 为今日
3. 若未指定 `<clock>` 则认为是 `00:00:00`
4. 若指定了 `<clock>` 则必须要加 T, 比如 `T10:20:34` 表示今天的早晨 10 点多
5. 所有绝对时间在程序内部都会被转换成相对于特定时刻的秒数

下面举几个绝对日期的例子:

- `2014-02-10T10:00:00.000`
- `T10:20:44.234`
- `2014-040T23:23:54.330`

5.1.3 相对时间坐标

相对时间坐标即某个时间相对于参考时刻的秒数、小时数、天数或年数。因而在使用相对时间时, 首先要给定两个参数: 参考时刻以及相对时间所使用的单位。

GMT 参数 `TIME_EPOCH` 用于指定参考时刻, `TIME_UNIT` 用于指定相对时间的单位。也可以用参数 `TIME_SYSTEM` 同时指定这两个参数。默认的参考时刻为 1970 年 1 月 1 日午夜, 默认的相对时间单位为秒。

在指定了参考时刻后, 相对时间就跟一般的浮点数没什么区别了。那如何区分一般的浮点数与相对时间呢? 有两种方式:

1. 在数据后加上小写的 t, 比如 `30t` 表示相对于 `TIME_EPOCH` 间隔了 30 个 `TIME_UNIT` 时间单位的时刻
2. 在命令行中使用 `-ft` 选项表明当前数据是相对时间, 此时不需要在数字后加 t

5.1.4 一般坐标值

在绘制常规的笛卡尔坐标轴时, 即输入数据不是地理坐标、绝对时间或相对时间时, 输入数据可以直接用浮点数表示, 而不去在意其物理含义及单位。比如, 5 牛顿的力, 5 千克的质量, 在 GMT 看来都只是浮点数 5。

这些浮点数坐标可以用两种方式表示:

1. 一般表示: [±]xxx.xxx, 比如 123.45
2. 指数表示: [±]xxx.xx[E|e|D|d[±]xx], 比如 1.23E10

5.2 表数据

英文称为 table data, 也称为列数据或多列数据, 常用于表示点和线。表数据中有 N 个记录, 每个记录都有 M 个字段。

表数据可以有三种形式: ASCII 表、二进制表和 netCDF 表。最常用的表数据形式是 ASCII 表, 可以用编辑器直接编辑。ASCII 表中有 N 行 M 列, 每行是一个记录, 每列是一个字段。一个记录内的字段之间用空格、制表符、逗号或分号分隔。

5.2.1 ASCII 表

ASCII 表简介

ASCII 表是最常见的数据形式, 可以用编辑器直接编辑。数据中有 N 行 M 列, 每行是一个记录, 每列是一个字段。一个记录内的字段之间用空格、制表符、逗号或分号分隔。每个字段可以是整数 (12)、浮点数 (20.34)、地理坐标 (12:23:44.5W)、绝对时间 (2010-10-20T10:30:53.250)、相对时间 (30t), 同时, GMT 还可以处理大多数 CVS (Comma-Separated Values) 文件, 包括被双引号扩起来的数字。

例如:

```
# This is a comment line
# lon      lat    evdp
# This is another comment line
133.949  34.219  20
133.528  34.676  15
130.233  33.410  43
135.133  35.313  35
131.377  34.398  22
132.792  34.457  34
133.620  34.936  6
131.101  32.811  23
129.435  33.212  55
133.144  33.647  67
```

记录中以 # 开头的行, 即第一列是 # 的记录, 会被当做注释行直接忽略, 不算在 N 个记录之内。所以这个 ASCII 表可以认为有 10 行 3 列, 三列数据分别代表经度、纬度和深度。

不同的模块和选项的组合会对数据的列数以及每列的含义都有不同的定义, 因而需要根据具体情况去准备数据。准备数据的过程中可能会用到 GMT 的 -i、-o 选项以及 gawk 等工具。

在输出 ASCII 表时, 字段之间默认用 TAB 键分隔, 可以通过修改 *IO_COL_SEPARATOR* 来设置字段的分隔符,

文件头记录

在第一个记录前, 可以有一个或多个与数据无关的记录, 称为文件头记录 (file header records)。

记录中以 # 开头的行都被当做注释忽略, 所以不算是文件头记录。其他不以 # 开头但与数据无关的行, 则是文件头记录。要使用 -h 选项或设置参数 *IO_N_HEADER_RECS* 跳过这些文件头记录。

下面的 ASCII 表有一个文件头记录, 可以使用 -h1 选项跳过该文件头段记录:

```
# This is a comment line
# lon      lat    evdp
# This is another comment line
2015-01-05 10:20:30.456 15 45 60 6.0
133.949  34.219  20
133.528  34.676  15
130.233  33.410  43
135.133  35.313  35
131.377  34.398  22
132.792  34.457  34
133.620  34.936  6
131.101  32.811  23
129.435  33.212  55
133.144  33.647  67
```

数据段头记录

在绘制断层的时候, 可以将每个断层的经纬度信息分别放在各自的文件中, 但当断层数量很多时, 这样做可能导致目录下有太多数据文件而混乱不堪。为了解决类似的问题, GMT 引入了多段数据的概念。

多段数据, 顾名思义, 就是一个文件中包含了多个数据段。为了区分每个数据段, 需要在每段数据的开头加上一个额外的数据段头记录 (segment header records) 来标记一段新数据的开始。

数据段头记录可以是任意格式, 但所有数据段头记录的第一列字符必须相同。GMT 中默认的数据段头记录的首字符为 >, 也可以通过修改 *IO_SEGMENT_MARKER* 设置为其他字符。

IO_SEGMENT_MARKER 可以取两个特殊的值:

- 取 B 表示用空行作为数据段的分隔符
- 取 N 表示用一个所有列都是 NaN 的记录作为数据段分隔符

下面是一个包含两个数据段的多段数据, 每段数据是一个多边形:

```
>
10  20
```

(下页继续)

(续上页)

```
15 30
12 25
>
22 20
30 30
40 50
35 44
```

数据段头记录不仅仅用于标记数据段的开始，还可以额外指定该段数据的其他属性。比如：

- **-D** 指定某个距离值
- **-W** 指定当前数据段的画笔颜色
- **-G** 指定当前数据段的填充色
- **-Z** 设置当前数据对应的 Z 值，并从 CPT 文件中获取 Z 值对应的颜色
- **-L** 设置当前数据段的标签信息
- **-T** 设置当前数据段的一般描述信息
- **-Ph** 表明当前数据段构成的闭合多边形位于另一个闭合多边形的内部
- ...

这些数据段头记录中的选项参数将覆盖命令行中相应选项的值。

下面的多段数据，分别设置两段数据拥有不同的画笔颜色：

```
> -W2p,red
10 20
15 30
12 25
> -W2p,blue
22 20
30 30
40 50
35 44
```

5.2.2 二进制表

对于 IO 密集型任务，可以以二进制表的形式读写数据文件以加速。

简单地说，ASCII 表与二进制表的区别在于前者使用 `fprintf` 输出而后者使用 `fwrite` 输出。GMT 在读取二进制表数据时，无法直接判断数据中有多少个记录，每个记录有多少个字段。因而需要使用 **-bi** 选项指定二进制表数据的格式，详情见[-b 选项](#)。

二进制表中也可以有文件头记录，用 **-h** 选项可以指定要跳过的字节数。二进制表也可以表示多段数据，此时用一个值为 `NaN` 的记录作为数据段头记录来标记每段数据的开始。

5.2.3 NetCDF 表

表数据也可以用 NetCDF 格式保存, 该格式的好处在于通用。比如 GMT 自带的海岸线数据就是 NetCDF 的表数据。NetCDF 表数据中包含了一个或多个一维数组, 每个一维数组都有对应的变量名 (比如 `lon`、`lat`、`vel` 等等), 由于 NetCDF 格式的数据中包含了很多元数据 (meta data), 所以读取就变得很容易。

默认情况下, GMT 在读入 NetCDF 表时会从第一个一维数组开始读, 并将其作为输入的第一列, 然后再读入第二个一维数组, 将其作为输入的第二列, 依次循环下去, 直到读完自己所需要的字段数。

若需要手动指定要从 NetCDF 表中读入哪些变量, 可以在 netCDF 表文件名后加上后缀 `?<var1>/<var2>/...`, 也可以直接使用 `-bic<var1>/<var2>/...` 选项。其中 `<var1>` 等是要从 NetCDF 表中读入的变量名。比如 `file.nc?lon/lat` 表示要从文件中读入 `lon` 和 `lat` 两个一维数组作为输入数据。

目前, GMT 只支持读取 netCDF 表数据, 不支持写 netCDF 表数据。

5.3 NaN 数据

NaN, 全称 Not-a-Number。

在数据处理或绘图时, 经常会遇到某个数据点丢失或未赋值的情况。遇到这种情况, 以前的做法是给这一点赋一个特殊值 (比如 `-9999.99`) 来表明这个数据点有问题, 但这样的解决办法并不完美, 因为有时 `-9999.99` 或其他选定的数依然可能是一个合理值, 进而导致数据处理时出现问题。

更现代的解决办法时使用 IEEE 规定的 NaN 来表示某个数据点丢失或未赋值。从数学上来说, 当你进行一些未定义的数学操作 (比如 $0/0$) 的结果就是 NaN, 在 ASCII 文件中, 用字符 `NaN` 表示 NaN。

NaN 值在程序内部会以一种特殊的 bit pattern 进行保存, 因而程序可以很容易识别一个数据是否是 NaN。在 C 中, 标准库函数 `isnan` 即用于测试某一个浮点数是否是 NaN。

GMT 会在数据处理以及绘图时对 NaN 数据进行检测:

- 若值为 NaN 的数据出现在计算中, 则结果也是 NaN
- 值为 NaN 的数据, 则在绘图时通常不绘制该数据点, 或者为 NaN 数据指定专门的颜色

若输入数据的 X 或 Y 列中包含 NaN 值, 通常该记录会在读入过程中被跳过。而实际上, 这些 NaN 记录可以根据 `IO_NAN_RECORDS` 的取值不同而有两种解释。

5.4 兼容 OGR 的 GMT 矢量数据格式

5.4.1 简介

地理空间数据有多种格式，按照类型划分，可以大致分为光栅型（raster）和矢量型（vector）。

- 光栅型数据格式不完整列表: http://www.gdal.org/formats_list.html
- 矢量型数据格式不完整列表: http://www.gdal.org/ogr_formats.html

简单的说，在 GMT 中，netCDF 格式的网格文件属于光栅型地理空间数据，而一般的表数据则属于矢量型地理空间数据。

[GDAL](#) 是一个可以实现多种光栅型或矢量型地理空间数据格式间互相转换的库/工具，其全称为 Geospatial Data Abstraction Library。历史上，GDAL 仅用于处理光栅型数据格式，而 OGR 则仅用于处理矢量型数据格式。从 GDAL 2.0 开始，二者相互集成在一起，即 GDAL 已经具备了处理光栅型和矢量型地理空间数据格式的能力。本文中，提到 OGR 时，仅表示地理空间矢量数据格式。

一个矢量数据中，不仅仅有地理空间数据（地理坐标数据，点、线、多边形等），也可以有非地理空间数据（城市名等）。老版本的 GMT 只能处理地理空间数据，而不能利用非地理空间数据。GMT5 定义了一种兼容 OGR 的 GMT 矢量数据格式，通常称为 OGR/GMT 格式。这种格式中包含了地理空间和非地理空间数据，所有的非地理空间数据都以注释的形式写到文件中，因而 GMT4 也可以正常读取 OGR/GMT 格式的数据。OGR/GMT 格式中包含了非空间数据，使得 GMT 的输出可以很容易地被其他 GIS 或绘图软件所使用。

5.4.2 OGR/GMT 格式

OGR/GMT 格式的一些重要性质列举如下：

- 所有非空间数据都以注释行的形式写到文件中，这些注释行在 GMT4 中会被直接忽略
- 非空间数据的各个字段之间用空格分隔，每个字段均以字符 @ 作为前缀，紧接着一个用于表征该字段内容的字符。每个字段内部的多个字符串之间用字符 | 分隔
- 字符 \ 作为转义字符，比如字符串内 \n 表示换行
- 文件中，非空间数据均保存在空间数据之前。因而 GMT 在处理地理空间数据之前，以及解析了非地理空间信息，这些非地理空间信息可能会影响到地理空间数据的处理
- 数据文件的第一个注释行必须指定 OGR/GMT 格式的版本号，即 @VGMT1.0
- 为了兼容其他 GIS 格式（比如 shapefiles），OGR/GMT 格式中显式包含了一个字段，用于指定接下来的地理空间数据是点、线还是多边形
- 每个文件有一个头段注释，其中指定了当前文件所包含的地理特征，以及每个特征所对应的非地理属性（比如区域范围，投影方式等）
- 同一个 OGR/GMT 格式的文件中，所有数据段必须具有相同类型的特征（都是点或线或多边形）

5.4.3 OGR/GMT 元数据

在 OGR/GMT 格式的文件头部，需要包含一系列元数据信息。元数据用于描述整个文件的共同信息，比如版本号、几何类型、区域范围、投影方式、非空间数据的格式等信息。

格式版本号 @V

OGR/GMT 格式的版本号用 @V 来指定。因而 OGR/GMT 格式的文件的第一行的内容必须是：

```
# @VGMT1.0
```

其中 GMT1.0 是 OGR/GMT 格式的版本号。

几何类型 @G

@G 用于指定当前数据文件的几何类型，其后接的参数可以是：

- POINT: 包含多个数据点 (每个点都可以有自己的头段记录)
- MULTIPPOINT: 多点数据 (所有的点共用同一个头段记录)
- LINestring: 包含多个独立的线段 (即 GMT 中的多段数据，每条线段可以有自己的头段记录)
- MULTILINESTRING: 多线数据 (文件中的所有线段是一个特性，共用同一个头段记录)
- POLYGON: 包含多个闭合多边形 (每个多边形可以有自己的头段记录)
- MULTIPOLYGON: 多多边形数据 (所有多边形共用同一个头段记录)

例如：

```
# @VGMT1.0 @GPOLYGON
```

区域范围 @R

@R 用于指定区域范围，其格式与 -R 选项类似。例如：

```
# @R150/190/-45/-54
```

投影信息 @J

投影信息用四个可选的字符串表示，每个字符串以 @J 开头。

- @Je: 投影的 EPSG 代码
- @Jg: GMT 中所使用的投影参数
- @Jp: 投影参数的 Proj.4 表示
- @Jw: 投影参数的 OGR WKT (well known text) 表示

示例：

```
# @J@e4326 @JgX @Jp"+proj=longlat +ellps=WGS84+datum=WGS84 +no_defs"
# @Jw"GEOGCS["WGS84",DATUM["WGS_1984",SPHEROID["WGS84",6378137,1
298.257223563,AUTHORITY["EPSG","7030"]],TOWGS84[0,0,0,0,0,0,0],
AUTHORITY["EPSG","6326"]],PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],UNIT["degree",0.01745329251994328,
AUTHORITY["EPSG","9122"]],AUTHORITY["EPSG","4326"]]"
```

声明非空间字段 @N

@N 后接一个用于描述非空间字段名称的字符串，各个字段名称之间用 | 分隔。若字段名称中有空格，则必须用引号括起来。@N 必须有一个与之对应的 @T。其中 @T 用于指定每个字段名称的数据类型。可取的数据类型包括 `string`、`integer`、`double`、`datetime` 和 `logical`。

示例：

```
# @VGMT1.0 @GPOLYGON @Nname|depth|id @Tstring|double|integer
```

表明数据文件中包含了多个多边形，每个多边形都可以有独立的头段记录以指定非空间信息，非空间信息有三个，分别是 `name`、`depth` 和 `id`，三个字段分别是字符串、浮点型和整型。

5.4.4 OGR/GMT 数据

元数据之后即是真正的数据，包括非空间数据和空间数据。

非空间数据

非空间数据用 @D 表示，紧跟着一系列以 | 分隔的字符串，每个字段的含义以及格式由 @N 和 @T 决定。

非空间数据所在的注释行应放在每段数据的坐标数据前。对于几何类型为 `LINE`、`POLYGON`、`MULTILINE` 或 `MULTIPOLYGON` 的数据而言，每段数据之间用特定的字符分隔，默认分隔符是 >。非空间数据紧跟在 > 行之后。对于几何类型为 `POINT` 或 `MULTIPOINT` 的数据而言，则不需要分隔符。

@N 和 @D 中的字符串中若包含空格，则必须用双引号括起来。若字符串中本身包含双引号或 |，则需要使用转义字符进行转义。若两个 | 之间为空，则表示对应的字段为空值。

一个点数据的头段示例：

```
# @VGMT1.0 @GPOINT @Nname|depth|id @Tstring|double|integer
# @D"Point 1"-34.5|1
```

一个多边形数据的头段示例：

```
# @VGMT1.0 @GPOLYGON @Nname|depth|id @Tstring|double|integer
>
# @D"Area 1"|-34.5|1
```

多边形拓扑

旧版本的 GMT 只支持常规的多边形，不支持一个多边形内有个洞的情况。

GMT 通过在多边形数据前加上 @P 和 @H 来指定当前的数据段是外环还是内环，即是真正的多边形，还是多边形内的洞。@H 必须紧跟在对应的 @P 之后。

@H 所指定的洞不应该有任何 @D 值，因为非空间数据适用于整个特性，而 @H 所指定的多边形只是多边形的一部分，并不是一个新的多边形。

5.4.5 示例

点数据示例：

```
# @VGMT1.0 @GPOINT @Nname|depth|id
# @Tstring|double|integer
# @R178.43/178.5/-57.98/-34.5
# @Je4326
# @Jp"+proj=longlat +ellps=WGS84 +datum=WGS84+no_defs"
# FEATURE_DATA
# @D"point 1"|-34.5|1
178.5 -45.7
# @D"Point 2"|-57.98|2
178.43 -46.8
...
...
```

线数据示例：

```
# @VGMT1.0 @GLINESTRING @Nname|depth|id
# @Tstring|double|integer
# @R178.1/178.6/-48.7/-45.6
# @Jp"+proj=longlat +ellps=WGS84 +datum=WGS84+no_defs"
# FEATURE_DATA
> -W0.25p
# @D"Line 1"|-50|1
178.5 -45.7
178.6 -48.2
178.4 -48.7
178.1 -45.6
> -W0.25p
# @D"Line 2"|-57.98|$
178.43 -46.8
...
...
```

多边形数据示例：

```

# @VGMT1.0 @GPOLYGON @N"Polygon name" | substrate | id @Tstring | string | integer
# @R178.1/178.6/-48.7/-45.6
# @Jj@Jp" +proj=latlong +ellps=WGS84 +datum=WGS84+no_defs"
# FEATURE_DATA
> -Gblue -W0.25p
# @P
# @D"Area 1" | finesand|1
178.1 -45.6
178.1 -48.2
178.5 -48.2
178.5 -45.6
178.1 -45.6
>
# @H
# First hole in the preceding perimeter, so is technically still
# part of the same geometry, despite the preceding > character.
# No attribute data is provided, as this is inherited.
178.2 -45.4
178.2 -46.5
178.4 -46.5
178.4 -45.4
178.2 -45.4
>
# @P
...

```

5.5 PostScript

GMT 生成的图片为 PS 格式，全称是 PostScript。这一节将简单介绍一下 PS 语言与 PS 格式。

5.5.1 PS 是什么

PostScript 是一种用于描述矢量图形的页面描述语言。简单的说，用 PostScript 语言写成的文件就是 PS 格式的图片，一般文件后缀用 `ps`，简称为 PS 文件。

提到格式，很多人都会有畏惧感，总觉得格式是个很复杂的东西。其实，格式不过是一种定义，定义了将信息以何种方式保存到文件中，那些用于打开某种格式的软件，只不过恰好知道了格式的定义，并且根据定义将文件中的信息提取出来，然后呈现给用户，仅此而已。比如 `doc`，用 MS Word 可以打开，用 WPS 也可以打开，如果你愿意，你也可以自己读 `doc` 的格式定义，自己写代码读取 `doc` 格式的文件。

下面教你怎样从零创建一个 PS 格式的文件，以消除对 PS 格式的陌生感。

- 首先打开你最常用的一款文本编辑器（比如 `vim`、`gedit` 或 `notepad++`），新建一个空白文件；
- 将如下内容复制到该空白文件中：

```

%! PS-Adobe-3.0
/Helvetica findfont 20 scalefont setfont

```

(下页继续)

(续上页)

```
150 400 moveto  
(PostScript is not that hard!) show  
  
showpage  
%%Trailer  
%%EOF
```

3. 将该文件以文件名 `simple.ps` 保存
4. Linux 下用命令 `gs simple.ps` 查看该文件; Windows 下应该直接双击就可以看到解释一下这个文件:

- `%! PS-Adobe-3.0` 表明该文件是 PostScript 格式, 且是 PS3.0 版
- `Helvetica` 是 PS 内置的一种字体, `findfont` 命令用于寻找、缩放、设置字体
- `150 400 moveto` 将坐标原点移动到某位置
- `(text) show` 用于显示文字
- `showpage` 表示显示该页

是不是很简单? GMT 的绘图模块本质上就是生成一堆 PS 代码, 只要把这些 PS 代码按照一定的规则保存到纯文本文件中, 即可用 PS 阅读器查看绘图效果。

5.5.2 PS 的优点

GMT 所有的绘图模块都只能生成 PS 代码, 将这些 PS 代码保存到 PS 文件中即可完成绘图。

不管当初 GMT 是以什么理由选择 PS 作为图像格式的, 就今天来看, PS 文件具有如下优势:

1. 矢量图形格式

PS 是矢量图形格式, 即用点、直线或多边形等数学方程的几何元素来表示图像。因而可以任意旋转与缩放而不是出现图像失真。

2. 易于转换为其他格式

GMT 提供了 `psconvert` 模块, 可以很方便地将 PS 文件以任意精度转换为 jpeg、png、eps、pdf 等图片格式, 以满足不同情形下的需求。

5.5.3 PS 阅读器

PS 阅读器, 或称 PS 解释器, 是用于查看 PS 文件的软件。

- `ghostscript`
- `gsview`
- `evince`
- `zathura` (Linux only)
- `SumatraPDF` (Windows only)

5.5.4 将 PS 转换为其他格式

GMT 提供了 `psconvert` 模块，可以很方便地将 PS 文件以任意精度转换为 jpeg、png、eps、pdf 等图片格式，以满足不同情形下的需求。

比如，将 PS 文件转化为 JPG 文件：

```
gmt psconvert test.ps
```

转换为 JPG 格式的过程中对其进行裁剪并旋转：

```
gmt psconvert -A -P test.ps
```

转化为 PDF 格式：

```
gmt psconvert -Tf -A -P test.ps
```

保护环境，从阅读电子文档开始！

第 6 章 网格文件

6.1 网格文件简介

GMT 可以处理并绘制 2D 网格数据。通常，2D 网格文件的 X 方向代表经度、Y 方向代表纬度，Z 值可以表示高程、重力值、温度、速度等。

GMT 可以使用的 2D 网格数据有四种格式：

1. netCDF 格式
2. Sun 光栅文件
3. Native 二进制格式
4. 用户自定义网格格式

最常见也最推荐的网格数据格式是 netCDF 格式。

6.1.1 netCDF 网格文件

GMT 默认将 2D 网格保存为兼容 COARDS 的 netCDF 文件，一般以 .nc 或 .grd 作为文件后缀。

COARDS 是许多机构在分发网格文件时遵循的标准格式。GMT 兼容该格式，因而 GMT 可以直接读取其他机构或程序提供的网格文件，GMT 生成的网格文件也可以被其他程序读取。

netCDF 格式目前有两个版本，netCDF3 和 netCDF4。GMT 目前默认使用 netCDF4 版本的文件格式。netCDF 的具体格式比较复杂，此处不做介绍，一般用户也无需了解。

6.1.2 Sun 光栅文件

GMT 可以读取 Sun 光栅文件格式的网格文件。Sun 光栅文件包含了一个头段区以及一系列无符号一字节整型以表示 bit-pattern。所有的字节按照 scanline 方式排列，每行必须包含偶数个字节。

6.1.3 Native 二进制格式

在很久以前的 GMT 中，其自定义了一套二进制网格文件格式。但由于其通用性太差，已经被更现代的 netCDF 格式所取代。新版本的 GMT 依然支持这种二进制网格格式，但不建议使用。该文件格式由两部分组成：892 个字节的头段区和长度不定的数据区。

native 二进制网格格式包含了一个 892 字节的头段区以及长度不定的数据区。头段区中包含了用于描述网格文件的变量，这些变量在文件中的存储顺序及其含义如下表：

表 6.1: GMT 自定义二进制网格文件结构

参数	类型	说明
nx	int	X 方向节点数目
ny	int	Y 方向节点数目
registration	int	配准方式: 0 代表网格线配准, 1 代表像素配准
x_min	double	区域的 X 最小值
x_max	double	区域的 X 最大值
y_min	double	区域的 Y 最小值
y_max	double	区域的 Y 最大值
z_min	double	数据的 Z 最小值
z_max	double	数据的 Z 最大值
x_inc	double	X 方向的节点间隔
y_inc	double	Y 方向的节点间隔
z_scale_factor	double	读取 Z 值后要乘以的因子
z_add_offset	double	Z 值乘以因子后要加上的偏移量
x_units	char[80]	X 方向的单位
y_units	char[80]	Y 方向的单位
z_units	char[80]	Z 方向的单位
title	char[80]	对数据集的描述
command	char[320]	生成该数据的命令
remark	char[160]	额外的注释
z	TYPE [nx*ny]	Z 值数组

6.1.4 用户自定义网格格式

GMT 还可以读取用户自定义的网格文件, 只要用户写好自定义网格文件的读写子程序, 并将其与 GMT 函数库链接起来即可。详情参考源码中的 `gmt_customio.c` 文件。

6.2 网格配准

GMT 中的 2D 网格文件, 在确定了网格范围和网格间隔后, 网格线会出现在 $x = x_{min}, x_{min} + x_{inc}, x_{min} + 2 \cdot x_{inc}, \dots, x_{max}$ 和 $y = y_{min}, y_{min} + y_{inc}, y_{min} + 2 \cdot y_{inc}, \dots, y_{max}$ 处。而节点的位置有两种选择, 即网格线配准 (gridline registration) 和像素配准 (pixel registration)。GMT 默认使用的是网格线配准方式。

[Source Code](#)

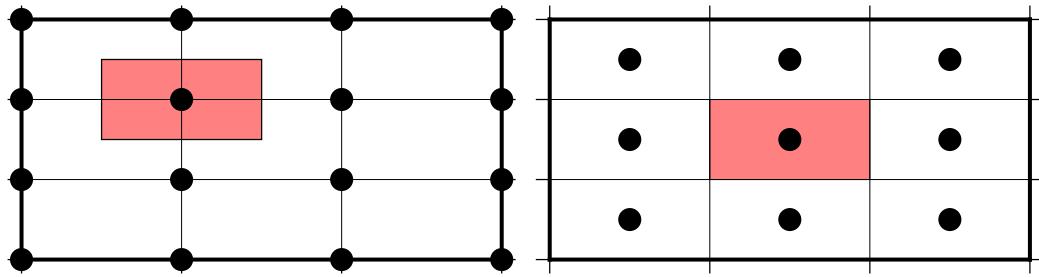


图 6.1: GMT 网格配准方式

(左) 网格线配准; (右) 像素配准。

注解: 大多数原始观测数据都采样网格线配准方式, 而有时经过处理的数据会以像素配准方式发布。尽管两种配准方式可以互相转换, 但转换过程中会降低 Nyquist 采样率, 阻尼一些高频信息。因而如果你可以控制, 应尽量避免配准转换。

6.2.1 网格线配准

在网格线配准方式下, 节点(图中黑色圆圈)中心位于网格线的交叉点处, 节点的值代表了长宽为 $x_{inc} \cdot y_{inc}$ 的单元(图中红色区域)内的平均值。这种情况下, 节点数目与网格范围和间隔的关系为:

$$\begin{aligned} nx &= (x_{max} - x_{min})/x_{inc} + 1 \\ ny &= (y_{max} - y_{min})/y_{inc} + 1 \end{aligned}$$

左图中 $nx=ny=4$ 。

6.2.2 像素配准

在像素配准方式下, 节点(图中黑色圆圈)位于网格单元的中心, 即网格点之间的区域, 节点的值代表了每个单元(图中红色区域)内的平均值。在这种情况下, 节点数目与网格范围和间隔的关系为:

$$\begin{aligned} nx &= (x_{max} - x_{min})/x_{inc} \\ ny &= (y_{max} - y_{min})/y_{inc} \end{aligned}$$

因而, 对于相同的网格区域和网格间隔而言, 像素配准比网格线配准要少一列和一行数据。右图中 $nx=ny=3$ 。

6.3 边界条件

GMT 中的某些模块在对网格文件做某些操作(比如插值或计算偏导)时, 在网格边界处需要指定网格的边界条件。边界条件的选取会影响到区域边界处的计算结果。GMT 中可以通过[-n 选项](#)选项指定网格的边界条件。

GMT 中网格文件的边界条件有三类:

6.3.1 默认边界条件

默认的边界条件是：

$$\nabla^2 f = \frac{\partial}{\partial n} \nabla^2 f = 0$$

其中 $f(x, y)$ 是网格文件内的值, $\partial/\partial n$ 是垂直于这个方向的偏导。

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$$

是二维 Laplace 操作符。

6.3.2 周期边界条件

X 方向的周期边界条件表明数据是以周期 $x_{max} - x_{min}$ 重复的, 数据每 $N = (x_{max} - x_{min})/x_{inc}$ 个点重复一次。Y 方向同理。

- 对于网格线配准的网格文件, 共 $N+1$ 列数据。第一列数据位于 $x = x_{min}$ 处, 最后一列 ($N+1$ 列) 数据位于 $x = x_{max}$ 处, 周期边界条件意味着数据的第一列和最后一列是完全相同的
- 对于像素配准的网格文件, 有 N 列数据, 第一列位于 $x_{min} + x_{inc}/2$, 最后一列 (第 N 列) 位于 $x_{max} - x_{inc}/2$, 第一列和最后一列的数据是不同的。

6.3.3 地理边界条件

地理边界条件表明:

1. 若 $(x_{max} - x_{min}) \geq 360$ 且 180 是 x_{inc} 的整数倍, 则在 X 方向使用周期为 360 的周期边界条件, 否则使用默认边界条件
2. 若条件 1 为真, 且 $y_{max} = 90$ 则 Y 方向上使用“北极边界条件”, 否则使用默认边界条件
3. 若条件 1 为真, 且 $y_{min} = -90$ 则 Y 方向上使用“南极边界条件”, 否则使用默认边界条件

6.4 读 netCDF 文件

netCDF 格式的设计相当灵活, 可以包含多个多维变量。而 GMT 中与网格相关的模块, 只能直接处理包含一个二维变量的 netCDF 文件。因而, GMT 在读取包含了多个多维变量的 netCDF 文件时, 可以做一些特殊的处理。

6.4.1 多个二维变量的处理

当 netCDF 网格文件中包含多个二维变量时, GMT 默认会读取第一个二维变量作为 Z 值, 并忽略其余的二维变量。如果用户想要自己指定读取某个特定的二维变量, 可以在网格文件名后加上后缀 ?<varname> 来实现, 其中 <varname> 是 netCDF 文件中包含的变量名。

比如想要从文件中获取名为 slp 的二维变量的信息, 可以用:

```
gmt grdinfo "file.nc?slp"
```

两点说明：

1. netCDF 中包含的变量名 <varname> 可以用 ncdump -c file.nc 得到
2. Linux 下问号会被解析为通配符，因而在命令行或 Bash 中使用时需要将问号转义，或者将整个文件名放在单引号或双引号内

6.4.2 三维变量的处理

偶尔会遇到三维网格文件，比如地球参考模型，三个维度分别是经度、纬度和深度，模型中的速度和密度等则是一个三维变量。

在遇到多维变量时，GMT 默认会读取第一层（即深度值最小的那一层）数据。可以通过如下两种方法来读取特定层的数据。

1. 文件名后加上 [<index>]

<index> 是第三维度变量（比如深度）的索引值，第一层的索引值为 0

2. 文件名后加上 (<level>)

<level> 是要获取数据的那一层的深度值。若 <level> 指定的深度与网格不重合，则 GMT 会找到离其最近的深度，而不会去做插值

假设有一个地球模型文件，ncdump -c file.nc 的结果为（只列出与深度有关的部分）：

```
dimensions:
  depth = 32 ;
variables:
  float depth(depth) ;
  depth:long_name = "depth below earth surface" ;
  depth:units = "km" ;
  depth:positive = "down" ;
data:
  depth = 50, 100, 200, 300, 400, 400, 500, 600, 600, 700, 800, 900, 1000,
  1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200,
  2300, 2400, 2500, 2600, 2700, 2800, 2850 ;
```

从中可以看到，该模型在深度方向上有 32 层，分别对应 50 千米、100 千米，一直到 2850 千米。file.nc?vp[1] 会读取第二层（即深度 100 km 处）的 P 波速度；而 file.nc?vp(200) 会读取深度 200 千米处的 P 波速度。

说明：

1. ncdump -c file.nc 命令可以查看 netCDF 网格文件中的信息
2. Linux 下中括号和小括号有特殊含义，因而在命令行或 Bash 中使用时需要进行转义，或者将整个文件名放在单引号或双引号内

6.4.3 四维变量的处理

对于四维变量，方法类似。假设有一个四维网格文件，四个维度分别是纬度、经度、深度、时间，变量为压强。利用 `ncdump` 可以查看四个维度的取值范围：

```
lat(lat): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9  
lon(lon): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9  
depth(depth): 0, 10, 20, 30, 40, 50, 60, 70, 80, 90  
time(time): 0, 12, 24, 36, 48  
pressure(time,depth,lat,lon): 共 10x10x10x5=5000 个值
```

为了得到 $\text{depth}=10$, $\text{time}=24$ 处的变量信息，可以用：

```
gmt grdinfo "file.nc?pressure[2,1]"
```

或者：

```
gmt grdinfo "file.nc?pressure(24,10)"
```

需要注意，时间在前，深度在后。

6.4.4 一维变量的处理

包含一维变量的 netCDF 文件，也就是前面所说的 netCDF 表。可以通过在文件名后加上变量名来使用一个一维变量，比如：

```
gmt psxy "file.nc?lon/lat" ...  
gmt convert "file.nc?time/lat/lon"
```

If one or more of the selected variables are two-dimensional, and have the same leading dimension as the other selected variables they will be plotted in their entirety. For example, if a netCDF file contains 6 time steps recording temperature at 4 points, and the variable `temp` is a 6 by 4 array, then the command `gmt convert "file.nc?time/temp"` can result in:

```
2012-06-25T00:00:00 20.1 20.2 20.1 20.3 2012-06-25T12:00:00 24.2 23.2 24.5  
23.5 2012-06-26T00:00:00 16.1 16.2 16.1 16.3 2012-06-26T12:00:00 22.1 23.0  
23.9 23.5 2012-06-27T00:00:00 17.5 16.9 17.2 16.8
```

If, for example, only the second temperature column is needed, use `gmt convert "file.nc?time/temp[1]"` (indices start counting at 0).

6.4.5 修改坐标单位

某些 GMT 工具要求网格中的两个维度的单位必须是米，若输入数据中的维度的单位不是米，则需要对网格坐标做一些变换。

1. 如果使用的是地理网格数据（即两个维度是经度和纬度），可以加上 `-fg` 选项，则网

格坐标会根据 Flat Earth 近似，自动转换成以米为单位。

2. 若使用的是笛卡尔坐标下的网格，但维度的单位不是米（比如是千米），则可以在网格文件名后加上 `+u<unit>` 选项来指定当前网格的维度单位，程序会在内部自动转换成以米为单位。比如，要读入一个维度单位为千米的网格文件，可以通过 `filename+uk` 将其转换成以米为单位。在输出网格时，会自动使用输入数据的原始单位，除非输出网格文件名中有额外的 `+u` 选项。也可以使用 `+U<unit>` 实现逆变换，将以米为单位的网格坐标变成以 `<unit>` 为单位。

6.5 写网格文件

6.5.1 网格文件格式

上面已经说过，GMT 可以读写三类格式的网格数据：netCDF 格式、二进制格式和 Sun 光栅文件。进一步细分，GMT 可以读取几十种不同格式的网格数据。

GMT 所支持的网格文件格式在下表列出，每种网格文件格式都有一个对应的两字符 ID。

表 6.2: 网格文件格式 ID

ID	说明
	<i>GMT 4 netCDF standard formats</i>
nb	GMT netCDF format (8-bit integer, COARDS, CF-1.5)
ns	GMT netCDF format (16-bit integer, COARDS, CF-1.5)
ni	GMT netCDF format (32-bit integer, COARDS, CF-1.5)
nf	GMT netCDF format (32-bit float, COARDS, CF-1.5)
nd	GMT netCDF format (64-bit float, COARDS, CF-1.5)
	<i>GMT 3 netCDF legacy formats</i>
cb	GMT netCDF format (8-bit integer, depreciated)
cs	GMT netCDF format (16-bit integer, depreciated)
ci	GMT netCDF format (32-bit integer, depreciated)
cf	GMT netCDF format (32-bit float, depreciated)
cd	GMT netCDF format (64-bit float, depreciated)
	<i>GMT native binary formats</i>
bm	GMT native, C-binary format (bit-mask)
bb	GMT native, C-binary format (8-bit integer)
bs	GMT native, C-binary format (16-bit integer)
bi	GMT native, C-binary format (32-bit integer)
bf	GMT native, C-binary format (32-bit float)
bd	GMT native, C-binary format (64-bit float)
	<i>Miscellaneous grid formats</i>
rb	SUN raster file format (8-bit standard)
rf	GEODAS grid format GRD98 (NGDC)
sf	Golden Software Surfer format 6 (32-bit float)
sd	Golden Software Surfer format 7 (64-bit float)
af	Atlantic Geoscience Center AGC (32-bit float)
ei	ESRI Arc/Info ASCII Grid Interchange format (ASCII integer)
ef	ESRI Arc/Info ASCII Grid Interchange format (ASCII float)
gd	Import/export via GDAL

除了上面列出的网格文件格式之外, 有 C 编程经验的高级用户还可以自己自定义网格文件格式, 并将读写该格式的子程序链接到 GMT 函数库中, 使得 GMT 可以支持自定义网格文件格式的读取。详情见源码中的 `gmt_customio.c`。

GMT 在读网格数据时, 可以自动检测网格文件的格式, 所以不用担心。而在写网格数据时, GMT 会默认使用 **nf** 格式 (可以修改 `IO_GRIDFILE_FORMAT` 以设置其他网格文件格式为默认值)。如果想要以其他格式写网格数据, 则可以在网格文件名后加上 `=<ID>` 参数以指定要使用的网格文件格式。

无论是读还是写网格文件, 都可以按照如下格式指定网格文件的文件名:

```
<name>[=<ID>][+s<scale>][+o<offset>][+n<nan>]
```

其中

- <name> 是网格文件名
- <ID> 是写网格文件时要使用的网格文件格式
- <scale> 将所有数据乘以比例因子 <scale>, 默认值为 1
- <offset> 将所有数据加上一个常数 <offset>, 默认值为 0
- <nan> 表明将文件中值为 <nan> 认为是 NaN

<scale> 和 <offset> 都可以取为 a, 表明由程序自动决定比例因子和偏移量的值。在读网格文件时, 会先乘以比例因子再加上偏移量; 在写网格文件时, 会先加上偏移量, 再乘以比例因子。

举几个例子:

1. 读入 Golden 软件公司的 surfer 软件生成的网格文件, GMT 可以自动识别, 故而直接用 `file.grd`
2. 读一个二进制短整型网格文件, 先将所有值为 32767 的值设置为 NaN, 再将数据乘以 10 并加上 32000, 可以用 `myfile.i2=bs+s10+o32000+n32767`
3. 将一个二进制短整数网格文件减去 32000 再除以 10, 然后写到标准输出, 可以用 `=bs+s0.1+o-3200`
4. 读一个 8 字节标准 Sun 光栅文件 (其原始范围为 0 到 255), 并将其归一化到正负 1 范围内, 可以用 `rasterfile+s7.84313725e-3+o-1`, 即先乘以因子使得数据范围从 0 到 255 变成 0 到 2, 再减去 1, 则数据范围变成 -1 到 1
5. 写一个 8 字节整型 netCDF 网格文件, 偏移距由 GMT 自动决定, 可以用 `=nb+oa`

GMT 还支持通过网格文件后缀自动识别网格文件格式, 详情见[网格文件后缀](#)一节。

6.5.2 分块与压缩

出于性能的考虑, GMT 在输出超过 16384 个网格单元的网格文件时, 会打开分块功能。所谓分块, 即数据不是按照一行一行序列存储的, 而是将整个网格分成若干个区块, 然后依次存储每个区块的数据。

在读取数据时, 若只需要读取数据的一部分, netCDF 只需要读取相应的区块即可, 不用先读取整个数据。

由于数据的压缩和解压比磁盘 IO 要快, 因而可以对 netCDF 数据进行压缩, 使得磁盘占用更少, IO 负载更少。netCDF 的压缩可以分为若干等级, 压缩级别越高, 文件越小, 读写数据越快, 但压缩/解压越耗时。通常, 压缩级别取 1 到 3 效果比较好。

GMT 参数 `IO_NC4_CHUNK_SIZE` 可以控制分块的大小, `IO_NC4_DEFLATION_LEVEL` 可以控制压缩等级。

6.6 网格文件后缀

GMT 中也可以将网格文件的后缀与网格文件格式关联起来，就像 Windows 下 `docx` 后缀的文件与 MS Word 关联一样，这样 GMT 就可以直接根据文件后缀确定网格文件的格式了，这样更加直观也更加易用。

这一特性通过一个叫 `gmt.io` 的文件来实现。GMT 会依次在当前目录、家目录或 `~/.gmt` 目录下，寻找 `gmt.io`。

`gmt.io` 的示例格式如下：

```
# GMT i/o shorthand file

# It can have any number of comment lines like this one anywhere
# suffix format_id scale offset NaN Comments
grd      nf      -      -      -      Default format
b        bf      -      -      -      Native binary floats
i2       bs      -      -      32767  2-byte integers with NaN value
ras      rb      -      -      -      Sun raster files
byte    bb      -      -      255   Native binary 1-byte grids
bit     bm      -      -      -      Native binary 0 or 1 grids
mask    bm      -      -      0      Native binary 1 or NaN masks
faa     bs      0.1    -      32767  Native binary gravity in 0.1 mGal
ns      ns      a      a      -      16-bit integer netCDF grid with auto-scale and
→auto-offset
```

要使用这一特性，需要将参数 `IO_GRIDFILE_SHORTHAND` 设置为 `true`。此时，文件名 `file.i2` 等效于 `file.i2=bs//32767`, `wet.mask` 等效于 `wet.mask=bm+n0`。

6.7 查看 netCDF 文件

某些软件可以直接用于查看 netCDF 文件的内容：

- [ncview](#)
- [Panoply](#)
- [ncBrowse](#)

更多相关工具，见 [netCDF 网站上的列表](#)。

注解： 尽管大多数程序都可以读取 netCDF 文件，但某些不支持 netCDF4 格式。

第 7 章 CPT 文件

7.1 CPT 简介

有些程序需要将用户数据用颜色、绘图或图案来表示。比如，在使用 2D 网格数据绘制彩图或灰度图时，需要为每个 Z 值或 Z 值范围指定要使用的颜色；再比如，有些程序需要根据 Z 值决定线条颜色、符号的填充色等。CPT 文件的作用就是为每个 Z 值或 Z 值范围定义其对应的颜色。CPT 全称是 color palette table，也称为调色板或色标文件。

CPT 文件可以在 `grdimage`、`psxy`、`psxyz` 等命令中使用。通常，你可以使用 `makecpt` 或 `grd2cpt` 对已有的 CPT（比如 GMT 内置的 CPT）文件进行重采样，并适应你目前所使用的数据范围。当然，也可以自己手写 CPT 文件，或使用 `awk`、`perl` 之类的文本处理工具自动生成 CPT 文件。

由于 GMT 中可以用多种方式来指定颜色，因而 GMT 自带的或用户自己的 CPT 文件中通常含有如下注释语句：

```
# COLOR_MODEL = model
```

其中 `model` 可以取 RGB、hsv、cmyk，其指定了 CPT 文件中的颜色要如何进行解释。

7.2 使用 CPT

命令行指定 CPT 文件名后，GMT 会依次在当前目录、`~/.gmt` 和 `$GMTHOME/share/cpt/` 目录下寻找 CPT 文件，如果找不到还会加上后缀 `.cpt` 寻找。

在文件名后加上后缀 `+u|U<unit>` 还可以对 CPT 文件中的 Z 值进行缩放。

- `filename.cpt+u<unit>` 可以将 Z 值从 `<unit>` 变换为以米为单位
- `filename.cpt+U<unit>` 可以将 Z 值从以米为单位转换成 `<unit>`

7.3 CPT 文件格式

CPT 文件有两种类型，一种适用于分类型数据，一种适用于常规数据。下面会逐一介绍。

7.3.1 分类 CPT 文件

分类 CPT 文件适用于有分类型数据，对于这些数据而言，常规的数值操作是未定义的。比如，将陆地分成不同的类型，沙漠、森林、冰川等等，定义沙漠对应的值是 1，森林对应的值是 2，冰川定义的值是 3。显然值取 1.5 是没有意义的。

对于这种分类型数据，需要给每个分类指定一个 `<key>`。CPT 文件中则规定了每个 `<key>` 所对应的颜色或填充图案，以及一个可选的标签（通常是类型名）。

分类 CPT 文件的格式为：

```
<key1>      <fill1>      [<label1>]
<key2>      <fill2>      [<label2>]
...
<keyn>      <filln>      [<labeln>]
```

几点说明：

1. `<key>` 必须单调递增但不必连续
2. `<fill>` 可以是颜色，也可以是图案，见[填充](#)一节
3. 标签名以分号开头

`<key>` 可以取几个特殊的值：

- B: 小于 `<key1>` 的值的颜色，即背景色，默认值由 `COLOR_BACKGROUND` 控制
- F: 大于 `<keyn>` 的值的颜色，即前景色，默认值由 `COLOR_FOREGROUND` 控制
- N: 值为 `Nan` 所对应的颜色，默认值由 `COLOR_NAN` 控制

下面是一个分类 CPT 文件的示例：

```
0 yellow ;desert
1 green ;forest
2 red ;iceland
```

7.3.2 常规 CPT 文件

对于连续变化的数据而言，可以为几个特定值指定颜色，其他值的颜色则通过插值计算得到。此时可以使用常规 CPT 文件。常规 CPT 文件的格式为：

```
<z0>      <color_min_1>  <z1>      <color_max_1>      [<A>]      [<label>]
<z1>      <color_min_2>  <z2>      <color_max_2>      [<A>]      [<label>]
...
<zn-1>    <color_min_n>  <zn>      <color_max_n>      [<A>]      [<label>]
B      <fill_back>
F      <fill_fore>
N      <fill_NaN>
```

以 CPT 文件中的第一行为例，其定义了一个 Z 值切片，切片范围为 `<z0>` 到 `<z1>`，切片内每个 Z 值的颜色由 `<color_min_1>` 线性渐变为 `<color_max_1>`。其他行同理。
几点说明：

1. `<z>` 值必须单调递增
2. 每行的最大 `<z>` 必须与下一行的最小 `<z>` 相同，即 Z 切片之间不能存在中断
3. 若 `<color_min_1>` 与 `<color_max_1>` 相等，则 `<z0>` 到 `<z1>` 范围内的所有 Z 值均使用相同的颜色

4. 若 `<color_min_1>` 使用了图案, 则 `<color_max_1>` 必须设置为 -
5. `<A>` 是可选的, 用于表明在使用 `psscale` 命令绘制色标时要如何标注。`<A>` 可以取 L、U、B, 表示选择每个 Z 切片的下限、上限或者上下限作为标注。
6. ;`<label>` 是切片的标签, 当 `psscale` 使用 -L 选项时会在用 `<label>` 作为标注
7. B|F|N 语句分别用于设置背景色 (默认值为 `COLOR_BACKGROUND`)、前景色 (默认值为 `COLOR_FOREGROUND`) 和 NaN 值的颜色 (默认值为 `COLOR_NAN`)
8. B|F|N 语句要放在 CPT 文件的开头或结尾

下面是一个常规 CPT 文件的示例:

```
30    p16+200  80   -
80    -        100  -
100   200/0/0  200  255/255/0
200   yellow   300  green
```

本例中

- $30 < z < 80$: 以 200dpi 分辨率的 16 号图案填充
- $80 < z < 100$: 直接跳过
- $100 < z < 200$: 从深红色线性变化成黄色
- $200 < z < 300$: 从黄色线性变化成绿色

7.3.3 周期性 CPT 文件

有一类特殊的 CPT 文件称之为周期性 CPT 文件。对于这类 CPT 文件而言, GMT 在根据 Z 值查找对应的颜色时, 会首先从 Z 值中减去 CPT 文件中 Z 值范围的整数倍, 使得 Z 值永远落在 CPT 文件所指定的范围内。这意味着此类 CPT 文件没有前景色和背景色。

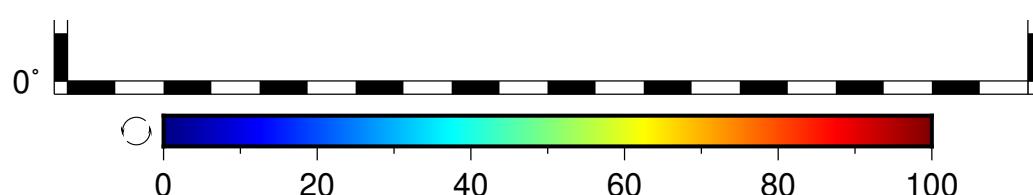
比如, 现有一个周期性 CPT 文件, 其定义了 Z 值范围在 $-\pi$ 到 π 之间的颜色表。若 Z 值等于 $3/2\pi$, 则会取 $-\pi/2$ 处的值作为其颜色。

在使用 `makecpt` 或 `grd2cpt` 生成 CPT 文件时加上 `-Ww` 选项即可生成周期性 CPT 文件。该选项本质上是在 CPT 文件中加上一个特殊的注释行:

```
# CYCLIC
```

GMT 在 CPT 中看到这一注释行时就会将该 CPT 当作周期性 CPT 文件。

Source Code



7.3.4 动态 CPT

GMT 内置的 CPT 文件都是动态的。

所有动态 CPT 文件都按照两种方式中的一种进行了归一化：

1. 如果为存在分界 (hinge) 的区域填色 (如地形颜色在高程为 0 的海岸线处发生非连续的分界) 那么 CPT 的 z 值在一侧为-1 到 0; 另一侧为 0 到 1; 分界线上为 0。CPT 文件中对分界线的值注释如下:

```
# HINGE = <hinge-value>
```

按照这种方式归一化的内置 CPT 包括 earth etopo1 geo globe polar red2green relief sealand split terra topo world

2. 如果不存在颜色的分界, 则 CPT 的 z 值归一化为 0 到 1 之间。实际使用中, CPT 的 z 值会拉伸到用户指定的范围, 包括两种拉伸模式:

1. 默认指定一个数值范围, 实际应用中 (如地形渲染) 再拉伸到真实值。见 CPT 文件中的 RANGE 注释:

```
# RANGE = <zmin/zmax>
```

2. 若未给定范围, 则扩展到数据的极限值

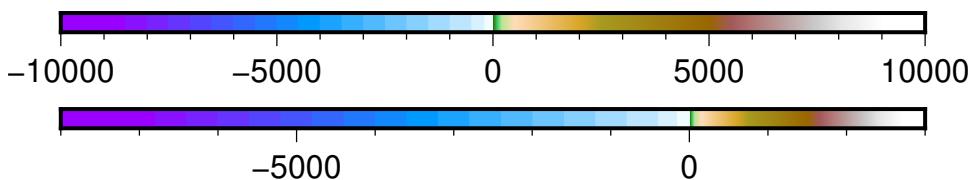
下图展示了动态的内置 CPT 如何拉伸适用于用户自定义数据, 绘制命令如下:

```
#!/bin/bash
PS=GMT_CPT_3.ps

gmt makecpt -Cglobe -T-8000/3000 > t.cpt

gmt psscale -Ct.cpt -Baf -Dx0/0+w4.5i/0.1i+h -K -P > $PS
gmt psscale -Cglobe -Baf -Dx0/0+w4.5i/0.1i+h -Y0.5i -O >> $PS

rm gmt.*
rm t.cpt
```



原始的内置 CPT (globe, 上图) 的初始范围为-10000 至 10000, 颜色不连续处的分界值 (hinge) 为 0, 小于 0 的部分为海蓝色系, 大于 0 的部分为大地色系。但实际上, 我们要绘制的区域高程值是非对称的, 从-8000 米到 3000 米 (下图)。由于 hinge 值的存在, 负高程依旧由海蓝色拉伸填充, 正高程则由大地色压缩填充。

注: 若要实现 hinge 分界效果, `makecpt` 命令中的 `-T` 选项不可设置 /<z_int>。

7.4 调整 CPT

GMT 模块 *makecpt* 和 *grd2cpt* 可以内置 CPT 文件为基础, 针对用户自己的数据制作专门的 CPT 文件; 也可以将用户自定义 CPT 重新调整为一个新的范围、颜色倒转甚至屏蔽部分颜色的新色标文件。比如某个内置 CPT 文件定义了从 0 到 1 颜色从蓝色变成红色, 用 *makecpt* 可以制作一个从 1000 到 3000 颜色从蓝色变成红色的 CPT 文件。不同的是, *makecpt* 常用于已知极值范围的表数据, 而 *grd2cpt* 常用于将 cpt 适用于一个或多个格网数据。此外, 两者均可通过以下方式翻转颜色:

1. 利用 **-Iz** 选项倒转 CPT 的 z 值指向(不包括前景色、背景色)。用于数据的正负颜色表示和惯例不符的情况(比如用正值表示海平面以下的深度, 而不是惯常的负高程)。
2. 利用 **-Ic** 选项翻转颜色的顺序, 包括前景色和背景色(与 **psscale** 的宽度设为负值效果一致)。
3. **-Icz** 表示以上两种效果的叠加, 即颜色顺序不变, 只调转了前景色和背景色
4. 利用 **-G** 选项提取主 CPT 的一部分颜色
5. 最后, 将初始的内置 CPT 或修改后的 CPT 拉伸为自定义数据范围

```
#!/bin/bash
J=X20c
R=1/10/1/10
PS=GMT_CPT_4.ps
SIZE=w15c/0.25c

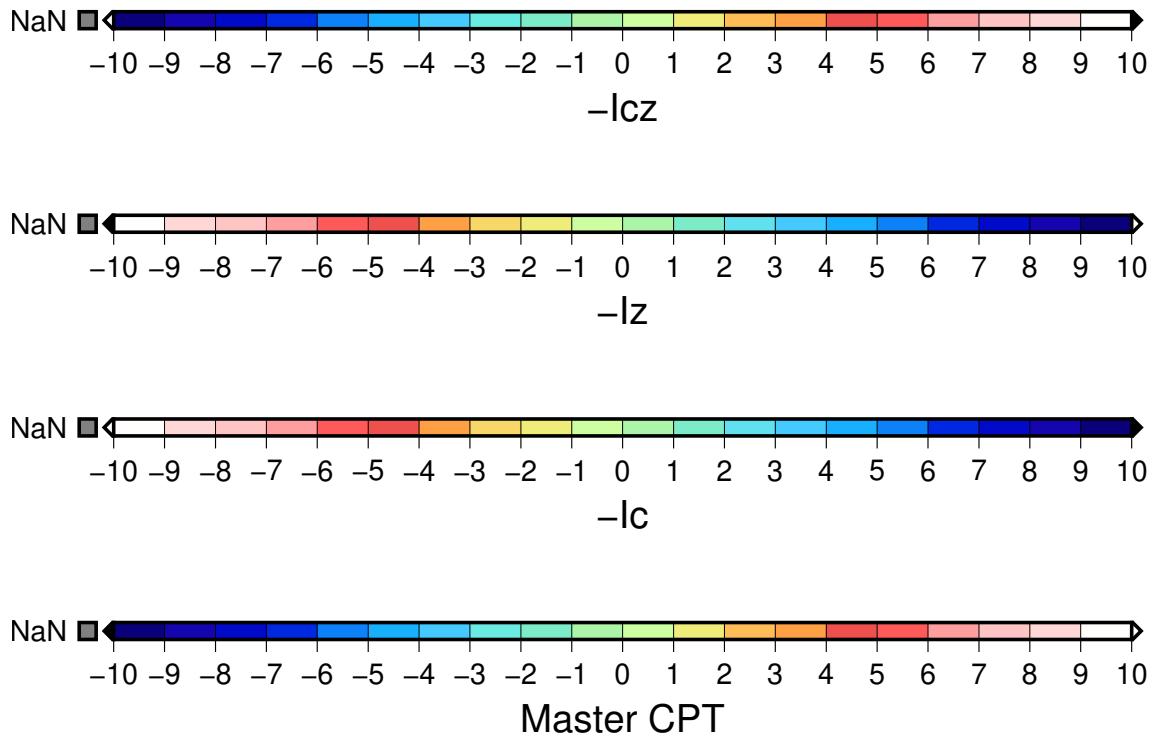
gmt makecpt -Chaxby -T-10/10/1 > Icpt.cpt
gmt makecpt -Chaxby -T-10/10/1 -Iz > Icpt_z.cpt
gmt makecpt -Chaxby -T-10/10/1 -Ic > Icpt_c.cpt
gmt makecpt -Chaxby -T-10/10/1 -Icz > Icpt_cz.cpt

gmt psxy -R$R -J$J -K -T > $PS

gmt psscale -R -J -CIcpt.cpt -Dn0.5/0.25+jCM+$SIZE+h+e+n -B+l"Master CPT" -K -O >> $PS
gmt psscale -R -J -CIcpt_c.cpt -Dn0.5/0.4+jCM+$SIZE+h+e+n -B+l"-Ic" -K -O >> $PS
gmt psscale -R -J -CIcpt_z.cpt -Dn0.5/0.55+jCM+$SIZE+h+e+n -B+l"-Iz" -K -O >> $PS
gmt psscale -R -J -CIcpt_cz.cpt -Dn0.5/0.7+jCM+$SIZE+h+e+n -B+l"-Icz" -K -O >> $PS

gmt psxy -R -J -O -T >> $PS

rm gmt.*
rm Icpt*.cpt
```



注: 这些 CPT 的调整选项顺序也是有优先级的。比如 **-lz** 使用后, 所有其他的更改都是基于 z 值反向的前提进行的。

7.5 GMT 内置 CPT

GMT 内置了 43 个常规 CPT 文件和一个分类 CPT 文件, 位于 `$GMTHOME/share/cpt` 目录中。

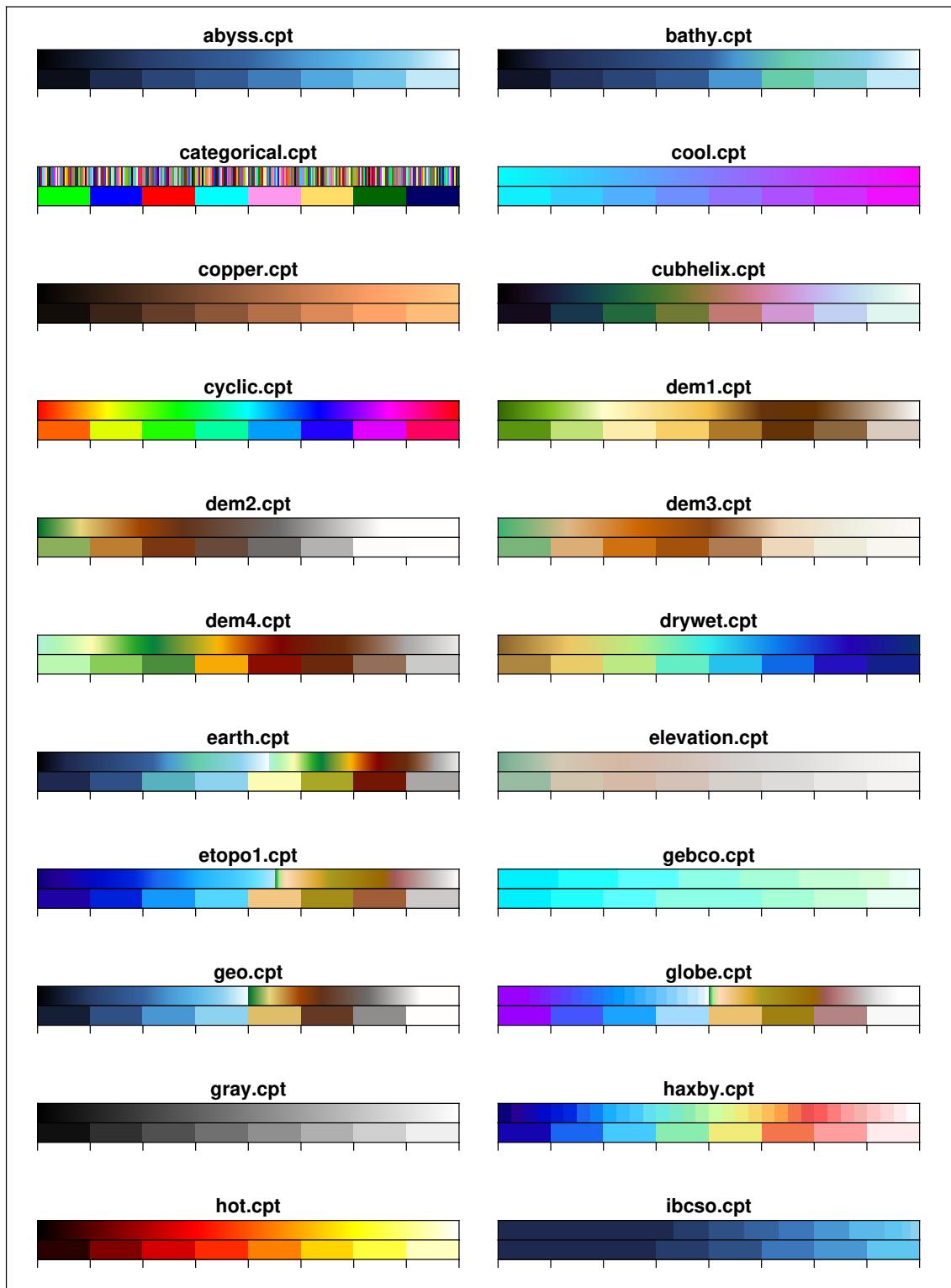
下图给出了 GMT 内置的 44 个 CPT 文件, 每张图上边的色标为原始 CPT, 用如下命令绘制:

```
gmt psscale -D5c/2c+w10c/1c+h+jTC -B0 -C<cpt> > test.ps
```

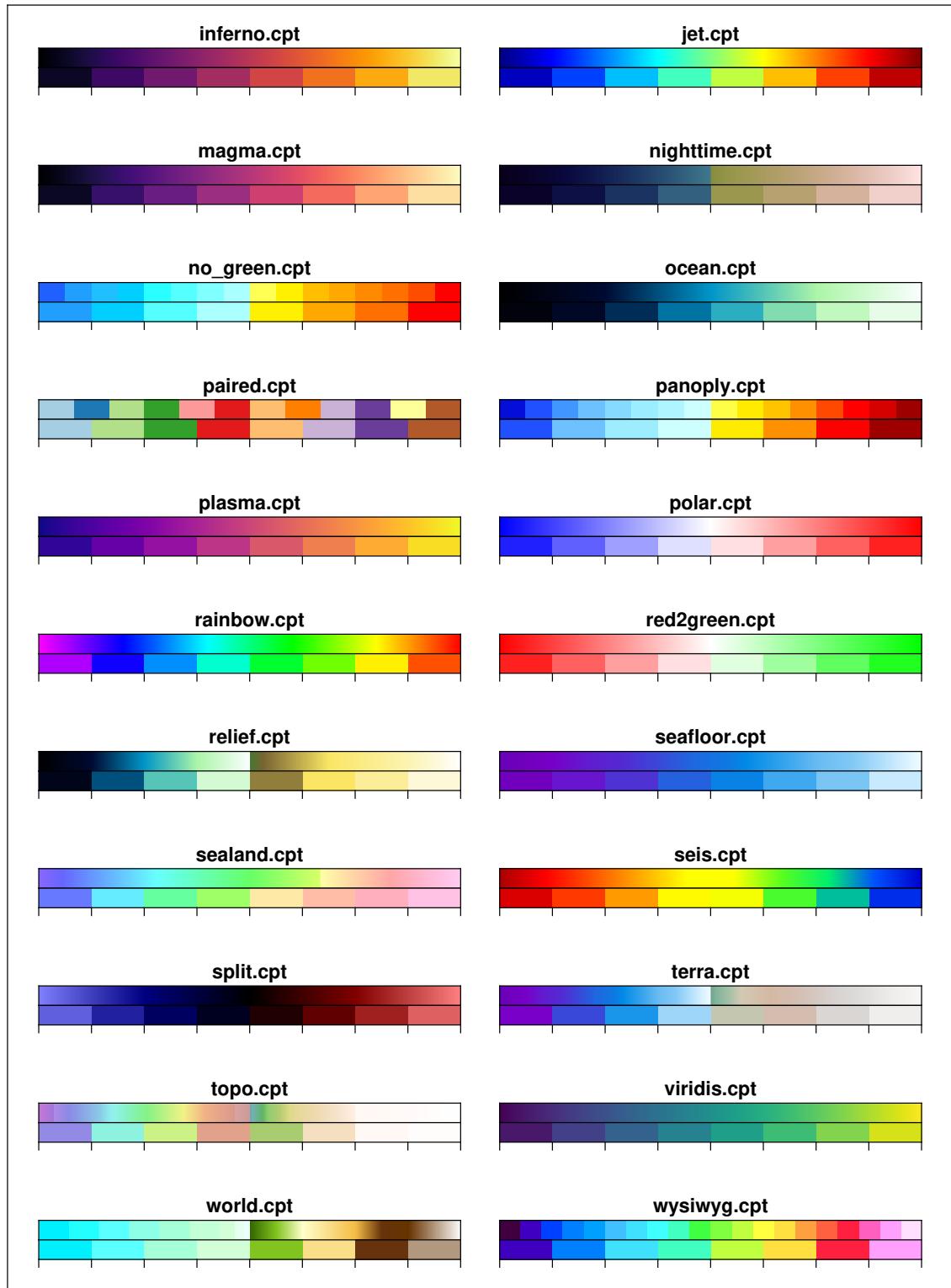
下边的色标是经过 `makecpt` 离散成 8 部分后的色标, 用如下命令绘制:

```
gmt makecpt -C<cpt> -T-1/1/0.25 > new.cpt
gmt psscale -D5c/2c+w10c/1c+h+jTC -Bf0.25 -Cnew.cpt > test.ps
```

[Source Code](#)



Source Code



7.6 其他 CPT

更多的 CPT 可以访问: <http://soliton.vm.bytemark.co.uk/pub/cpt-city/>

第 8 章 标准选项

GMT 模块众多，每个模块的具体效果由模块的众多选项来决定。不同的模块有不同的选项，这其中有一些选项是标准选项，即这些选项在所有的命令里都具有完全相同的意义。故而把这些通用的选项单独拿出来介绍。

GMT 中的选项都是以 - 加一个字符的形式构成，通常这个字符是经过精心挑选的，使得用户很容易根据字符记住该选项的作用。

本章将介绍 GMT 中标准选项的用法。由于标准选项在所有 GMT 模块中的用法是一样的，所以在模块手册中介绍每个模块的具体语法时，则不再介绍这些选项。

表 8.1: GMT 标准选项列表

选项	说明
-B	定义底图边框和轴的刻度、标注、标签等属性
-J	选择地图投影方式或坐标变换
-K	省略 PS 文件尾以追加更多 PS 代码
-O	省略 PS 文件头以将 PS 代码追加到已有的文件中
-P	设置纸张方向为 Portrait 模式
-R	指定区域范围
-U	在图上绘制时间戳
-V	详细报告模式
-X	移动 X 方向上的绘图原点
-Y	移动 Y 方向上的绘图原点
-a	将非空间数据与某些列联系在一起
-b	控制二进制的输入或输出
-d	将输入或输出中的 <i>nodata</i> 替换成 NaN
-e	仅处理与指定 pattern 匹配的数据
-f	设置 ASCII 输入或输出的格式
-g	根据数据间断对数据进行分段
-h	跳过数据的文件头段记录
-i	选择输入列
-n	设置网格插值方式
-o	选择输出列
-p	控制 3D 视角图
-r	设置网格配准方式
-s	控制 NaN 记录的处理方式
-t	设置图层透明度
-x	设置并行的核数 (仅限于支持并行的模块)
-:	输入数据是 y/x 而不是 x/y

8.1 -R 选项

-R 选项用于指定要绘制的数据范围或地图区域。该选项的参数可以用五种方式指定，同时支持六种不同类型的数据坐标。

8.1.1 四种方式

可以用四种方式指定数据范围。

1. `-R<xmin>/<xmax>/<ymin>/<ymax>`

通过给定 X 方向和 Y 方向的最大最小值来指定数据范围，是最常见的指定数据范围的方式。常用于指定笛卡尔数据的范围以及经线和纬线是直线的投影方式下的地理

区域。例如 $-R0/360/-90/90$ 表示 X 方向范围是 0 到 360, Y 方向范围是 -90 到 90。

2. $-R<xlleft>/<ylleft>/<xuright>/<yuright>+r$

通过给定矩形区域的左下角坐标 ($xlleft$, $ylleft$) 和右上角坐标 ($xuright$, $yuright$) 来指定数据范围。例如 $-R-90/20/-65/30+r$ 。这种形式通常用于倾斜的地图投影中, 此时不适合将经线和纬线作为地图边界。使用这种形式可以保证底图为矩形, 尽管经线和纬线可能不是直线。

3. $-R<gridfile>$

该方式会直接从 2D 网格文件 $<gridfile>$ 中提取数据范围信息。对于某些命令, 这种方式不仅会从网格中读取范围信息, 还会读入网格间隔和网格配准信息 (见 [-r 选项](#))。

4. $-R<anchor>/<x0>/<y0>/<nx>/<ny>$

该方法仅可用于创建新的网格文件。通过指定网格文件中某个特定点 (由 $<anchor>$ 决定, 见 [锚点](#) 一节) 的坐标 (由 $<x0>/<y0>$ 决定), X 和 Y 方向的网格点数 (由 $<nx>/<ny>$ 决定), 以及 X 和 Y 方向的网格间隔 (通常由 $-I<xinc>/<yinc>$ 选项决定), 即可指定网格文件的区域范围。

Source Code

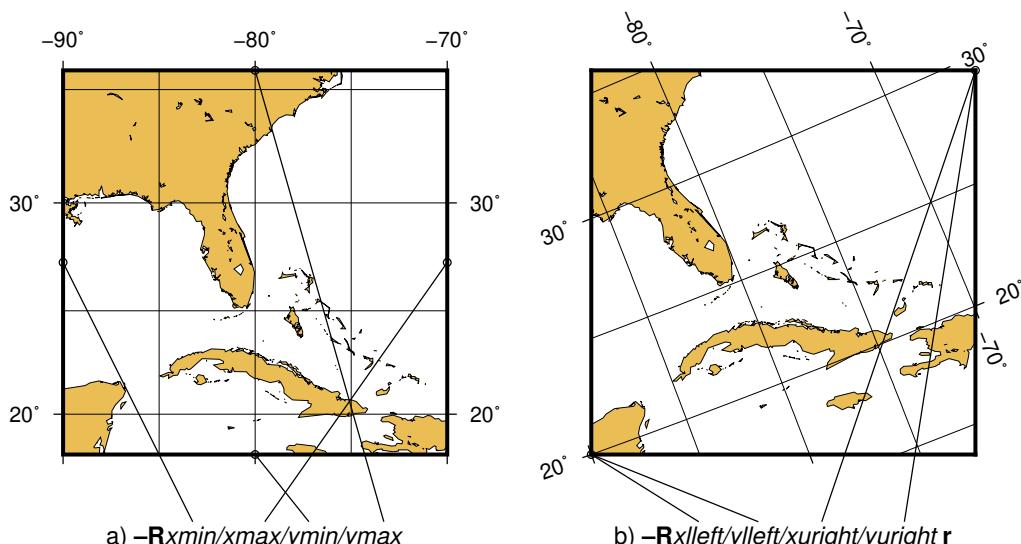


图 8.1: $-R$ 选项指定数据范围

(a) 指定每个维度的极值; (b) 指定左下角和右上角的坐标。

8.1.2 五种坐标

GMT 的坐标轴可以是多种类型, 比如地图的经纬度轴、一般的笛卡尔轴、时间轴等等。不同类型的轴需要用不同的方式来指定数据的范围。

地理坐标

地理坐标既可以用浮点数表示，也可以用度分秒表示：

1. 以浮点型的度数表示。比如 -123.45 代表-123.45 度

2. 度分秒表示：

```
[±]ddd[:mm[:ss[.xxx[W|E|S|N]]]]
```

其中，`ddd`、`mm`、`ss`、`xxx` 分别表示弧度、弧分、弧秒、弧毫秒。`W`、`E`、`S`、`N` 分别代表西经、东经、北纬、南纬。例如 `123:27W` 代表西经 123 度 27 分，`123:27:15.120W` 表示西经 123 度 27 分 15.12 秒。

对于两个常用的地图范围，GMT 提供了简写形式，`-Rg` 等效于 `-R0/360/-90/90`，`-Rd` 等效于 `-R-180/180/-90/90`。

GMT 对于地理投影和线性投影的默认设置有很大区别。有些时候数据是地理坐标，但是因为某些原因不能选择地理投影，只能选择线性投影 (`-JX` 或 `-Jx`)，此时可以通过如下几种方式表明当前数据是地理坐标下的数据，而不是简单的笛卡尔坐标：

1. 使用 `-Rg` 而不是 `0/360/-90/90`
2. 使用 `-Rd` 而不是 `-180/180/-90/90`
3. 使用 `-Rg<xmin>/<xmax>/<ymin>/<ymax>` 表明是某个有限范围的地理区域
4. 在范围后加后缀 `W|E|S|N` 或更通用的 `D|G`，比如 `-R0/360G/-90/90N`
5. 使用 `-fg` 选项表明输入数据是地理坐标，见[-f 选项](#)

注解： 推荐使用 `-fg` 这种方式来指定输入数据的类型。

投影后坐标

地理坐标可以通过选择投影方式投影成笛卡尔坐标，通过在区域范围后加上 `+u<unit>` 来表明这是一个经过投影的地理坐标。

例如 `-R-200/200/-300/300+uk` 表示位于投影中心 $(0, 0)$ 处的一个 $400\text{km} \times 600\text{km}$ 的矩形区域。这些坐标在 GMT 内部会被转换成区域左下角和右上角的地理坐标。当你想要用投影单位指定区域时用这种方式会比较方便（例如 UTM meters）。

绝对时间坐标

默认为 `yyyyymmddThh:mm:ss.xxx` 格式，见[数据格式](#) 一节。

相对时间坐标

见[数据格式](#) 一节。

笛卡尔坐标

既可以用浮点数表示也可以用指数表示，见[数据格式](#)一节。

8.2 -J 选项

-J 选项用于指定将数据投影到画布上所采用的函数，即坐标变换或投影方式。

8.2.1 GMT 投影代码

-J 选项有两种写法：

- `-J<δ>[<parameters>/]<scale>`
- `-J<Δ>[<parameters>/]<width>[h|+|-]`

此处 `<δ>` 是一个小写字母，`<Δ>` 是一个大写字母，代表某种投影方式。`<parameters>` 代表零个或多个由斜杠分隔的投影参数，具体的参数数目由投影方式决定。

投影方式使用大写字母时，其最后一个参数 `<width>` 表示地图宽度。投影方式使用小写字母时，最后一个参数 `<scale>` 表示比例尺，可以只跟一个数字 xxxx，也可以是 `1:xxxx` 格式。前者表示地球上的 1 度在图上为 xxxx 厘米（当然也可以是其他单位），而后者表示地图上的 1 厘米代表真实地球的 xxxx 厘米。

`h|+|-` 可进一步修改最后一个参数 `<width>` 的含义，其中：

- `h` 表示设置底图高度
- `+` 表示设置底图维度中最大的那个维度的长度
- `-` 表示设置底图维度中最小的那个维度的长度

即，可以使用 -J 选项选择投影方式，每个投影方式有大写和小写两种形式，其中小写形式指定的是地图的比例尺，大写形式指定的是图在画布上的宽度。

注解： 几乎所有地理投影方式，都只能指定其宽度或高度，而不能同时指定二者，因为地图高度会由地图宽度和投影方式自动决定。

画图时通常建议直接指定图片宽度，除非需要明确指定比例尺。

例如：

- `-Jm1c` 表示使用墨卡托投影，地图上的 1 度距离投影到画布上为 1 厘米
- `-Jm1:10000000` 表示使用墨卡托投影，画布上的 1 cm 代表实际距离中的 10000000 cm，即 100 km
- `-JM15c` 也表示使用墨卡托投影，整个地图的宽度是 15 厘米，地图的高度由 -R 和 -J 自动确定
- `-JM15ch` 表示整个地图的高度是 15 厘米

- **-JX10c/5c** 使用线性投影，地图的宽度是 10 厘米，高度为 5 厘米

下表列出了 GMT 所支持的全部投影方式，详细介绍见[投影方式](#)。

表 8.2: GMT 投影代码

-J 代码	说明
-JAlon₀/lat₀[/horizon]/width	Lambert azimuthal equal area
-JBlon₀/lat₀/lat₁/lat₂width	Albers conic equal area
-JClon₀/lat₀width	Cassini cylindrical
-JCyl_stere/[lon₀[/lat₀]]width	Cylindrical stereographic
-JDlon₀/lat₀/lat₁/lat₂width	Equidistant conic
-JElon₀/lat₀[/horizon]/width	Azimuthal equidistant
-JFlon₀/lat₀[/horizon]/width	Azimuthal gnomonic
-JGlon₀/lat₀[/horizon]/width	Azimuthal orthographic
-JGlon₀/lat₀alt/azim/tilt/twist/W/H/width	General perspective
-JH[lon₀]width	Hammer equal area
-JI[lon₀]width	Sinusoidal equal area
-JJ[lon₀]width	Miller cylindrical
-JKf[lon₀]width	Eckert IV equal area
-JKs[lon₀]width	Eckert VI equal area
-JLlon₀/lat₀/lat₁/lat₂width	Lambert conic conformal
-JM[lon₀[/lat₀]]width	Mercator cylindrical
-JN[lon₀]width	Robinson
-JOalon₀/lat₀azim/width	Oblique Mercator, 1: origin and azim
-JOblon₀/lat₀/lon₁/lat₁width	Oblique Mercator, 2: two points
-JOclon₀/lat₀/lon_p/lat_pwidth	Oblique Mercator, 3: origin and pole
-JP[a]width[/origin]	Polar [azimuthal] (θ, r) (or cylindrical)
-JPoly[lon₀[/lat₀]]width	(American) polyconic
-JQ[lon₀[/lat₀]]width	Equidistant cylindrical
-JR[lon₀]width	Winkel Tripel
-JSlon₀/lat₀[/horizon]/width	General stereographic
-JT[lon₀[/lat₀]]width	Transverse Mercator
-JUzone/width	Universal Transverse Mercator (UTM)
-JV[lon₀]width	Van der Grinten
-JW[lon₀]width	Mollweide
-JXwidth[l pexp T t][/height[l pexp T t]] d	Linear, $\log_{10}, x^a - y^b$, and time
-JYlon₀/lat₀width	Cylindrical equal area

8.2.2 Proj4 投影代码

GMT 用单个字母指定投影方式，但英文字母只有 26 个，而投影方式却不止 26 个，因而，从 GMT 4.3.0 开始，GMT 开始支持 Proj4 包的命名方式。与上面介绍的不同，Proj4 包不是使用单个字符指定投影方式，而是通过一个单词指定。比如墨卡托投影既可以用 `-Jm` 指定也可以用 `-Jmerc` 指定。

其语法为：

```
- ``-J<abbrev>[<parameters>/]<scale>``
- ``-J<Abbrev>[<parameters>/]<width>``
```

类似的，若投影代码的首字母为大写，则最后一个参数表示底图宽度；若投影代码为小写，则最后一个参数表述底图比例尺。

表 8.3: Proj4 投影代码

代码	说明
<code>-Jaea/lon₀/lat₀/lat₁/lat₂/scale</code>	Albers conic equal area
<code>-Jaeqd/lon₀/lat₀[/horizon]/scale</code>	Azimuthal equidistant
<code>-Jcass/lon₀/lat₀/scale</code>	Cassini cylindrical
<code>-Jcea/lon₀/lat₀/scale</code>	Cylindrical equal area
<code>-Jcyl_stere/[lon₀[/lat₀/]]scale</code>	Cylindrical stereographic
<code>-Jeqc/[lon₀[/lat₀/]]scale</code>	Equidistant cylindrical
<code>-Jeqdc/lon₀/lat₀/lat₁/lat₂/scale</code>	Equidistant conic
<code>-Jgnom/lon₀/lat₀[/horizon]/scale</code>	Azimuthal gnomonic
<code>-Jhammer/lon₀/scale</code>	Hammer equal area
<code>-Jeck4/lon₀/scale</code>	Eckert IV equal area
<code>-Jeck6/lon₀/scale</code>	Eckert VI equal area
<code>-Jlaea/lon₀/lat₀[/horizon]/scale</code>	Lambert azimuthal equal area
<code>-Jlcc/lon₀/lat₀/lat₁/lat₂/scale</code>	Lambert conic conformal
<code>-Jmerc/[lon₀[/lat₀/]]scale</code>	Mercator cylindrical
<code>-Jmill/lon₀/scale</code>	Miller cylindrical
<code>-Jmoll/[lon₀/]scale</code>	Mollweide
<code>-Jnsper/lon₀/lat₀alt/azim/tilt/twist/W/H/scale</code>	General perspective
<code>-Jomerc/lon₀/lat₀azim/scale</code>	Oblique Mercator, 1: origin and azimuth
<code>-Jomerc/lon₀/lat₀/lon₁/lat₁/scale</code>	Oblique Mercator, 2: two points
<code>-Jomercp::lon₀/lat₀/lon_p/lat_p/scale</code>	Oblique Mercator, 3: origin and pole
<code>-Jortho/lon₀/lat₀[/horizon]/scale</code>	Azimuthal orthographic
<code>-Jpolar/[a]scale[/origin]</code>	Polar [azimuthal] (θ, r) (or cylindrical)
<code>-Jpoly/[lon₀[/lat₀/]]scale</code>	(American) polyconic

下页继续

表 8.3 – 续上页

代码	说明
-Jrobin/[lon₀/]scale	Robinson
-Jsinu/lat₀/scale	Sinusoidal equal area
-Jstere/lon₀/lat₀[/horizon]/scale	General stereographic
-Jtmerc/[lon₀[/lat₀]])scale	Transverse Mercator
-Jutm/zone/scale	Universal Transverse Mercator (UTM)
-Jvandg/[lon₀/]scale	Van der Grinten
-Jwintri/[lon₀/]scale	Winkel Tripel
-Jxyxscale[l pexp[T t][/yscale[l pexp[T t]]][d]	Linear, log ₁₀ , $x^a - y^b$, and time

从 GMT6 开始, GMT 支持使用 PROJ.4 库来实现坐标和基准面的转换。这一特性是通过 GDAL 实现的, 因而需要 GMT 在安装时链接到 GDAL 库。详细的 PROJ.4 语法见 <http://proj4.org/apps/index.html>。

在 PROJ.4 中, 投影一般有很多参数, 多个参数之间用空格分隔。在 GMT 中, 可以将所有参数用双引号括起来:

```
-J" +proj=merc +ellps=WGS84 +units=m"
```

或直接将所有参数连在一起:

```
-J+proj=merc+ellps=WGS84+units=m
```

也可以直接使用 [EPSG codes](#), 但需要设置环境变量 `GDAL_DATA` 指向 GDAL 的 data 子目录。例如 `-JEPSG:4326` 表示使用 WGS-84 系统。

对于 `mapproject` 和 `grdproject` 模块, 可以直接使用 `+to` 关键字直接指定要将 A 参考系统转换为 B 参考系统, 而不需要中间步骤。例如:

```
-JEPSG:4326+to+proj=aeqd+ellps=WGS84+units=m
```

对于使用 `mapproject` 和 `grdproject` 进行点和网格文件的转换, GMT 可以使用所有的 [PROJ.4 投影](#)。但对于绘图而言, 其用处却很有限。一方面, 只有一部分 PROJ.4 的投影方式可以被映射到 GMT 的投影语法中。另一方面, 由于 PROJ.4 不是一个绘图库, 其不支持设置地图比例尺或地图大小。因而, GMT 为 PROJ.4 语法引入了两个扩展: `+width=` 和 `+scale=1:xxxx` 使得其与经典的 GMT 中的工作方式相似。也可以在投影参数的最后加上字符串 `/1:xxx` 来指定比例尺。

8.3 -B 选项

`-B` 选项用于控制底图边框的显示。

`-B` 选项有两套语法, 分别用于设置底图的边框以及每条轴的属性, 因而在一个命令中

可能需要多次使用 **-B** 选项。若命令中没有出现 **-B** 选项，则不绘制底图边框。

8.3.1 边框设置

-B 选项在设置边框属性时的语法为:

```
-B[<axes>][+b][+g<fill>][+n][+o<lon>/<lat>][+t<title>]
```

其中:

- **<axes>** 控制显示底图的哪几条边
- **+b** 在 3D 绘图中根据 **-R** 选项指定的范围绘制长方体的 12 条边
- **+g<fill>** 在底图内部填色, 见[填充](#)一节
- **+n** 表示不绘制边框和标注
- **+o<lon>/<lat>** 指定网格线的参考点。默认情况下, 网格线是以北极点作为参考的, 如果你想要以另一个点作为参考绘制倾斜的网格线, 则可以使用 **+o** 子选项
- **+t<title>** 指定当前底图的标题。该标题位于底图的上方中部。标题可以是任意字符串, 如果是字符串中有空格, 则必须用引号将字符串括起来

通常情况下, 只需要使用 **<axes>** 和 **+t<title>** 选项。

<axes>

<axes> 用于控制要绘制哪些边以及这些边是否有刻度或标注。**<axes>** 的格式为:

```
WSENZ[1234]wesez[1234]
```

对于二维图而言, 有上下左右四条边, 分别用东西南北 (WSEN) 的单词首字母表示。对于每条边都有三种状态:

1. 不出现某个字母 => 不绘制该字母所对应的边
2. 出现大写字母 WSEN => 绘制某条边, 该边有刻度、有标注
3. 出现小写字母 wesn => 绘制某条边, 该边有刻度、无标注

下面的命令分别使用了不同的 **-B** 选项, 可以自己执行, 查看绘图效果并试着理解 **<axes>** 的用法:

```
gmt psbasemap -R0/10/0/10 -JX5c -B2 -BWSN > test1.ps
gmt psbasemap -R0/10/0/10 -JX5c -B2 -BWsn > test2.ps
```

对于 3D 绘图来说, **<axes>** 还可以加上一个 Z 用于控制 Z 轴。同理, 大写的 Z 表示有刻度和标注, 小写的 z 表示有刻度无标注。表示默认情况下, 只会绘制一条 Z 轴, 可以使用 1234 的任意组合来表示要绘制哪些 Z 轴。其中 1 表示左下角的 Z 轴, 其他 Z 轴按逆时针顺序编号。加上 **+b** 子选项, 会绘制一个由 **-R** 选项范围决定的长方体的 12 条边, 即相当于一个 box。如果 Z 轴有指定网格间距, 则会在 xz 和 yz 平面内显示网格线。

下面的命令展示了 3D 绘图中 **-B** 选项的不同用法, 读者可以自己一一测试, 根据绘图

效果理解 -B 选项中各字母的含义。命令中的某些选项还没有介绍过，暂时可以不必理会其含义：

```
gmt psbasemap -R0/10/0/10/0/10 -JX5c -JZ5c -Bz2 -BWSENZ -p45/45 > test1.ps  
gmt psbasemap -R0/10/0/10/0/10 -JX5c -JZ5c -Bz2 -BWSENZ1234 -p45/45 > test2.ps  
gmt psbasemap -R0/10/0/10/0/10 -JX5c -JZ5c -Bz2 -BWSEN+b -p45/45 > test3.ps  
gmt psbasemap -R0/10/0/10/0/10 -JX5c -JZ5c -Bz2 -B+b -p45/45 > test4.ps  
gmt psbasemap -R0/10/0/10/0/10 -JX5c -JZ5c -Bz2 -BWESNZ+b -p45/45 > test5.ps  
gmt psbasemap -R0/10/0/10/0/10 -JX5c -JZ5c -B2 -Bz2 -BwSEnZ+b -p45/45 > test6.ps
```

示例

```
gmt psbasemap -R0/10/0/10 -JX5c -Ba2g2 -BWSen+lightblue+tttitle > test.ps  
gmt psbasemap -R0/10/0/10 -JX5c -Ba2g2 -BWS+lightblue+t"This is title" > test2.ps
```

8.3.2 轴设置

X 轴、Y 轴、Z 轴，每条轴都有很多属性，包括刻度间隔、网格线间隔、轴标签以及标注的间隔、前缀和单位。轴属性可以用如下语法控制：

```
-B[p|s][x|y|z]<intervals>[+l<label>][+p<prefix>][+u<unit>]
```

为了更加清晰，以上的语法也可以被分为两部分：

```
-B[p|s][x|y|z]<intervals>  
-B[p|s][x|y|z][+l<label>][+p<prefix>][+u<unit>]
```

其中，

- p|s 主属性或次属性
- x|y|z 设置哪一条轴的属性
- <interval> 设置刻度、网格线、标注的间隔
- +l<label> 用于给指定的轴加标签。
- +p<prefix> 选中的轴的标注加前缀
- +u<unit> 给选中的轴的标注加单位。对于地图而言，标注的单位为度，该符号是自动添加的，由 *FORMAT_GEO_MAP* 控制

p|s

对于每个轴来说，都有两个等级的属性可以设置，分别称为 p (Primary) 和 s (Secondary)。

对于地理坐标而言，通常只需要使用默认的 Primary 属性即可，而 Secondary 则主要用于坐标轴为时间轴的情况下，此时 p 和 s 分别用于指定不同尺度的时间间隔。在 GMT 默认的情况下，p 属性的标注比较靠近坐标轴，而 s 属性的标注离坐标轴稍远。p 和 s 的用法与区别，可以参考后面给出的例子。

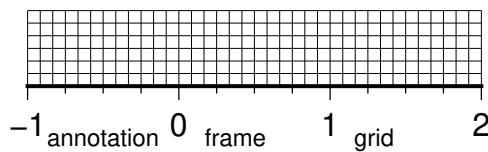
x|y|z

要设置哪些边的信息，默认值为 `xy`，即同时设置 X 轴和 Y 轴的信息。可以指定单个轴（比如只有 `x`），也可以同时指定多个轴（比如 `xy` 和 `xyz`）。如果想要不同轴有不同的设置，则需要多次使用 `-B` 选项，每个指定不同的轴。

<interval>

每个轴都有三个属性，分别是标注（annotation）、刻度（frame）和网格线（grid）。下图展示了这三个名词在绘图时的具体含义。

Source Code



`<interval>` 可以用于设置这三个属性的间隔，它是一个或多个 `[<t>]<stride>[±<phase>][<u>]` 的组合。

- `<t>` 可以取 `a`（标注）、`f`（刻度）、`g`（网格线），表明了要设置轴的哪部分的间隔
- `<stride>` 用于设置间隔，`stride` 为 0，表示不绘制
- $\pm<\text{phase}>$ 可以用于控制标注、刻度或网格线的起算点
- `<u>` 是间隔的单位，主要用于指定时间间隔

`-B` 选项还有一个可以自动计算间隔的功能，`-Bafg` 会根据当前的区域大小等信息自动计算合适的间隔，`-Bxafg -Byafg` 则会对 X 轴和 Y 轴分别计算合适的间隔。

读者可以将命令：

```
gmt psbasemap -JX10c/10c -R0/10/0/10 -Ba2f1g1 > test.ps
```

中的 `-B` 选项替换成如下不同的值并查看绘图效果以理解各个参数的含义：

- `-Ba2f1g1`
- `-Bxa2 -Bya1`
- `-Bxafg -Byafg`
- `-Ba2+1f1g1`

8.3.3 地理底图

地理底图与一般的坐标轴不同，其底图类型 `MAP_FRAME_TYPE` 使用 `fancy` 形式。

Source Code

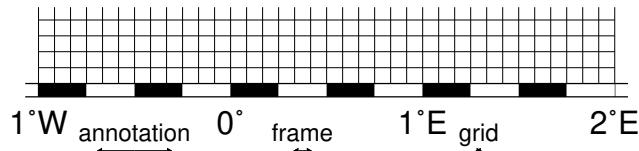


图 8.2: 地理底图示例 1

```
-Ba1f15mg5m -BS
```

下图同时使用了 p 和 s 两级属性。这里 p 属性用于显示弧度, s 属性用于显示弧分。

Source Code

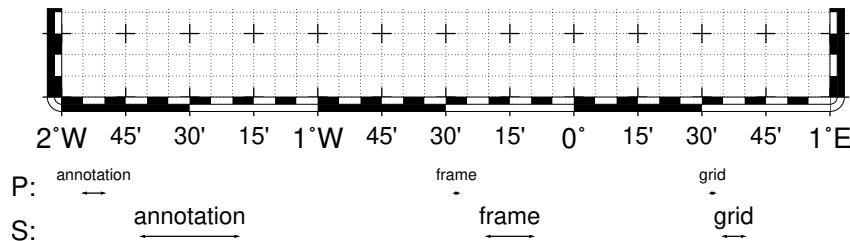


图 8.3: 地理底图示例 2

同时使用 P 和 S 两级属性 -Bpa15mf5mg5m -BwSe -Bs1f30mg15m

8.3.4 笛卡尔线性轴

对于一般的线性轴而言, 标注的格式由 *FORMAT_FLOAT_OUT* 决定, 其默认值为 %g, 即根据数据的大小决定用一般表示还是指数表示, 小数位的数目会根据 *<stride>* 自动决定。若设置 *FORMAT_FLOAT_OUT* 为其他值, 则会严格使用其定义的格式, 比如 %.2f 表示显示两位小数。

Source Code

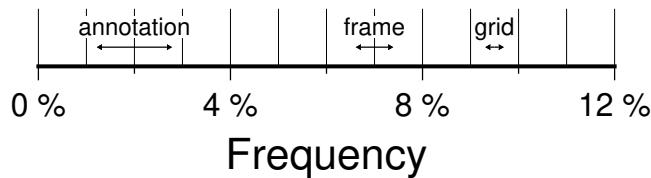


图 8.4: 笛卡尔线性轴

```
-R0/12/0/0.95 -JX3i/0.3i -Ba4f2g1+lFrequency+u" %" -BS
```

8.3.5 笛卡尔 \log_{10} 轴

由于对数坐标的特殊性, *<stride>* 参数具有特殊的含义。下面说明 *<stride>* 在对数坐标下的特殊性:

- *<stride>* 必须是 1、2、3 或负整数-n。

- 1: 每 10 的指数
- 2: 每 10 的指数的 1、2、5 倍
- 3: 每 10 的指数的 0.1 倍
- -n: 每 10 的 n 次方出现一次
- 在 `<stride>` 后加 `l`, 则标注会以 \log_{10} 的值显示, 比如 100 会显示成 2
- 在 `<stride>` 后加 `p`, 则标注会以 10 的 n 次方的形式显示, 比如 10^{-5}

Source Code

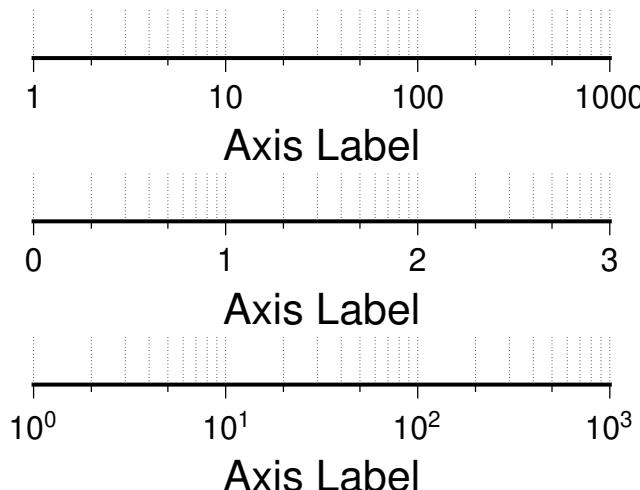


图 8.5: 对数坐标轴

(上) `-R1/1000/0/1 -JX3il/0.25i -Ba1f2g3`(中) `-R1/1000/0/1 -JX3il/0.25i -Ba1f2g3l`(下) `-R1/1000/0/1 -JX3il/0.25i -Ba1f2g3p`

8.3.6 笛卡尔指数轴

正常情况下, `<stride>` 用于生成等间隔的标注或刻度, 但是由于指数函数的特性, 这样的标注会在坐标轴的某一端挤在一起。为了避免这个问题, 可以在 `<stride>` 后加 `p`, 则标注会按照转换后的值等间隔出现, 而标注本身依然使用未转换的值。比如, 若 `stride=1`, `pow=0.5` (即 `sqrt`), 则在 1、4、处会出现标注。

Source Code

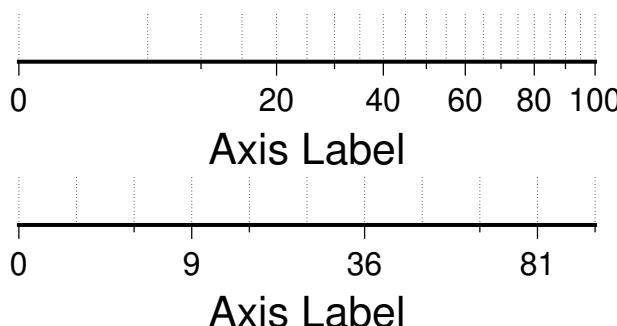


图 8.6: 指数投影坐标轴

(上) -R0/100/0/0.9 -JX3ip0.5/0.25i -Ba20f10g5 (下) -R0/100/0/0.9
 -JX3ip0.5/0.25i -Ba3f2g1p

8.3.7 时间轴

时间轴与其他轴不同的地方在于，时间轴可以有多种不同的标注方式。下面会用一系列示例来演示时间轴的灵活性。在下面的例子中，尽管只绘制了 X 轴（绘图时使用了 -BS），实际上时间轴标注的各种用法适用于全部轴。

在绘制时间轴时，需要指定时间间隔，时间间隔的单位可以取如下值：

表 8.4: GMT 时间单位

Flag	Unit	Description
Y	year	Plot using all 4 digits
y	year	Plot using last 2 digits
O	month	Format annotation using FORMAT_DATE_MAP
o	month	Plot as 2-digit integer (1–12)
U	ISO week	Format annotation using FORMAT_DATE_MAP
u	ISO week	Plot as 2-digit integer (1–53)
r	Gregorian week	7-day stride from start of week (see TIME_WEEK_START)
K	ISO weekday	Plot name of weekday in selected language
k	weekday	Plot number of day in the week (1–7) (see TIME_WEEK_START)
D	date	Format annotation using FORMAT_DATE_MAP
d	day	Plot day of month (1–31) or day of year (1–366) (FORMAT_DATE_MAP)
R	day	Same as d; annotations aligned with week (see TIME_WEEK_START)
H	hour	Format annotation using FORMAT_CLOCK_MAP
h	hour	Plot as 2-digit integer (0–24)
M	minute	Format annotation using FORMAT_CLOCK_MAP
m	minute	Plot as 2-digit integer (0–60)
S	seconds	Format annotation using FORMAT_CLOCK_MAP
s	seconds	Plot as 2-digit integer (0–60)

第一个例子展示了 2000 年春天的两个月，想要将这两个月的每周的第一天的日期标注出来。

```
gmt set FORMAT_DATE_MAP=-o FONT_ANNOT_PRIMARY +9p
gmt psbasemap -R2000-4-1T/2000-5-25T/0/1 -JX5i/0.2i -Bpa7Rf1d -Bsa10 -BS -P > GMT_-B_
˓→time1.ps
```



图 8.7: 时间轴示例 1

需要注意, `-Bsa10` 指定了次级标注的间隔为一个月, 由于此处使用的是大写的 O, 因而具体的显式方式由 `FORMAT_DATE_MAP` 决定。根据 `FORMAT_DATE_MAP` 的说明可知, 其值为 `-o` 表明以月份名格式显式。破折号表示要去掉日期前面的前置零(即 02 变成 2)。

下面的例子用两种不同的方式标注了 1969 年的两天。图中下面的例子使用周来标注, 上面的例子使用日期来标注。

```
gmt set FORMAT_DATE_MAP "o dd" FORMAT_CLOCK_MAP hh:mm FONT_ANNOT_PRIMARY +9p
gmt psbasemap -R1969-7-21T/1969-7-23T/0/1 -JX5i/0.2i -Bpa6Hf1h -Bsa1K -BS -P -K > GMT_-_
↪B_time2.ps
gmt psbasemap -R -J -Bpa6Hf1h -Bsa1D -BS -O -Y0.65i >> GMT_-B_time2.ps
```

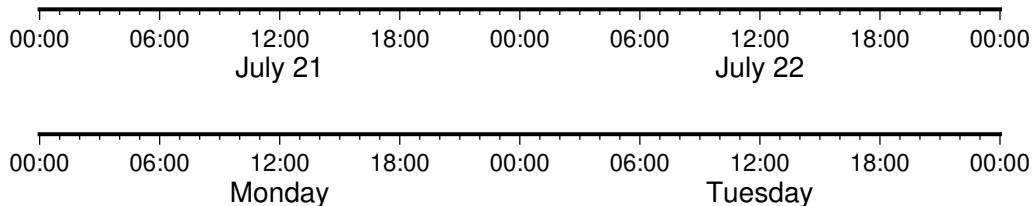


图 8.8: 时间轴示例 2

第三个例子展示了两年的时间, 并标注了每年以及每三个月。年标注位于一年间隔的中间, 月标注位于对应用月的中间而不是三个月间隔的中间。

```
gmt set FORMAT_DATE_MAP o FORMAT_TIME_PRIMARY_MAP Character FONT_ANNOT_PRIMARY +9p
gmt psbasemap -R1997T/1999T/0/1 -JX5i/0.2i -Bpa30f1o -Bsa1Y -BS -P > GMT_-B_time3.ps
```

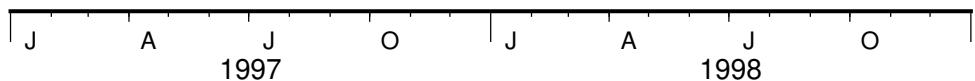


图 8.9: 时间示例 3

第四个例子展示了一天中的几个小时, 通过在 R 选项中指定 `t` 来使用相对时间坐标。这里使用了 `p` 属性和 `s` 属性, 12 小时制, 时间从右向左增加:

```
gmt set FORMAT_CLOCK_MAP=-hhm FONT_ANNOT_PRIMARY +9p TIME_UNIT d
gmt psbasemap -R0.2t/0.35t/0/1 -JX-5i/0.2i -Bpa15mf5m -Bsa1H -BS -P > GMT_-B_time4.ps
```

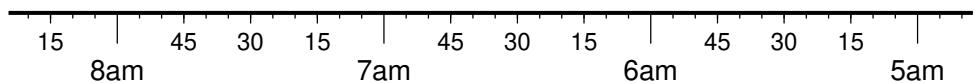


图 8.10: 时间轴示例 4

第五个例子用两种方式展示了几周的时间:

```
gmt set FORMAT_DATE_MAP u FORMAT_TIME_PRIMARY_MAP Character \
    FORMAT_TIME_SECONDARY_MAP full FONT_ANNOT_PRIMARY +9p
gmt psbasemap -R1969-7-21T/1969-8-9T/0/1 -JX5i/0.2i -Bpa1K -Bsa1U -BS -P -K > GMT_-B_ \
    ->time5.ps
gmt set FORMAT_DATE_MAP o TIME_WEEK_START Sunday FORMAT_TIME_SECONDARY_MAP Character
gmt psbasemap -R -J -Bpa3Kf1k -Bsa1r -BS -O -Y0.65i >> GMT_-B_time5.ps
```

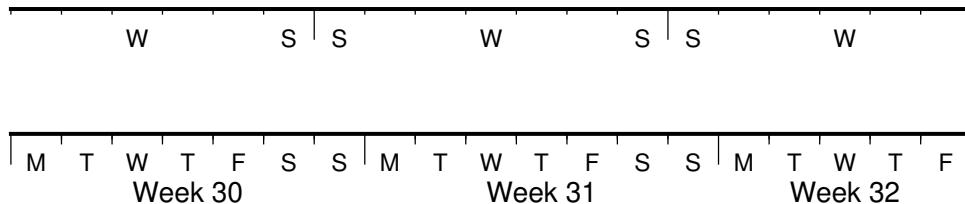


图 8.11: 时间轴示例 5

第六个例子展示了 1996 年的前 5 个月，每个月用月份的简写以及两位年份标注:

```
gmt set FORMAT_DATE_MAP "o yy" FORMAT_TIME_PRIMARY_MAP Abbreviated
gmt psbasemap -R1996T/1996-6T/0/1 -JX5i/0.2i -Ba1of1d -BS -P > GMT_-B_time6.ps
```



图 8.12: 时间轴示例 6

第七个例子:

```
gmt set FORMAT_DATE_MAP jjj TIME_INTERVAL_FRACTION 0.05 FONT_ANNOT_PRIMARY +9p
gmt psbasemap -R2000-12-15T/2001-1-15T/0/1 -JX5i/0.2i -Bpa5Df1d -Bsa1Y -BS -P > GMT_-B_ \
    ->time7.ps
```

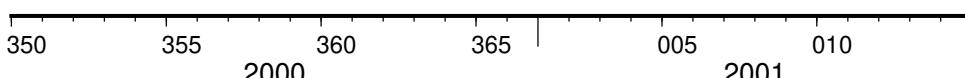


图 8.13: 时间轴示例 7

8.3.8 自定义轴

GMT 允许用户定义标注来实现不规则间隔的标注，用法是 `-Bc` 后接标注文件名。

标注文件中以“#”开头的行为注释行，其余为记录行，记录行的格式为:

```
coord type [label]
```

- `coord` 是需要标注、刻度或网格线的位置
- `type` 是如下几个字符的组合

- **a** 或 **i** 前者为 annotation, 后者表示 interval annotation
- 在一个标注文件中, **a** 和 **i** 只能出现其中的任意一个
- **f** 表示刻度, 即 frame tick
- **g** 表示网格线, 即 gridline
- **label** 默认的标注为 **coord** 的值, 若指定 **label**, 则使用 **label** 的值

需要注意, **coord** 必须按递增顺序排列。

下面的例子展示中展示了自定义标注的用法, **xannots.txt** 和 **yannots.txt** 分别是 X 轴和 Y 轴的标注文件:

```
cat << EOF > xannots.txt
416.0 ig Devonian
443.7 ig Silurian
488.3 ig Ordovician
542 ig Cambrian
EOF
cat << EOF > yannots.txt
0 a
1 a
2 f
2.71828 ag e
3 f
3.1415926 ag @~p@~
4 f
5 f
6 f
6.2831852 ag 2@~p@~
EOF
gmt psbasemap -R416/542/0/6.2831852 -JX-5i/2.5i -Bpx25f5g25+u" Ma" -Bpvcyannots.txt \
-BWS+glightblue -P -K > GMT_-B_custom.ps
gmt psbasemap -R416/542/0/6.2831852 -JX-5i/2.5i -Bsxcxannots.txt -BWS -O \
--MAP_ANNOT_OFFSET_SECONDARY=10p --MAP_GRID_PEN_SECONDARY=2p >> GMT_-B_\
custom.ps
rm -f [xy]annots.txt
```

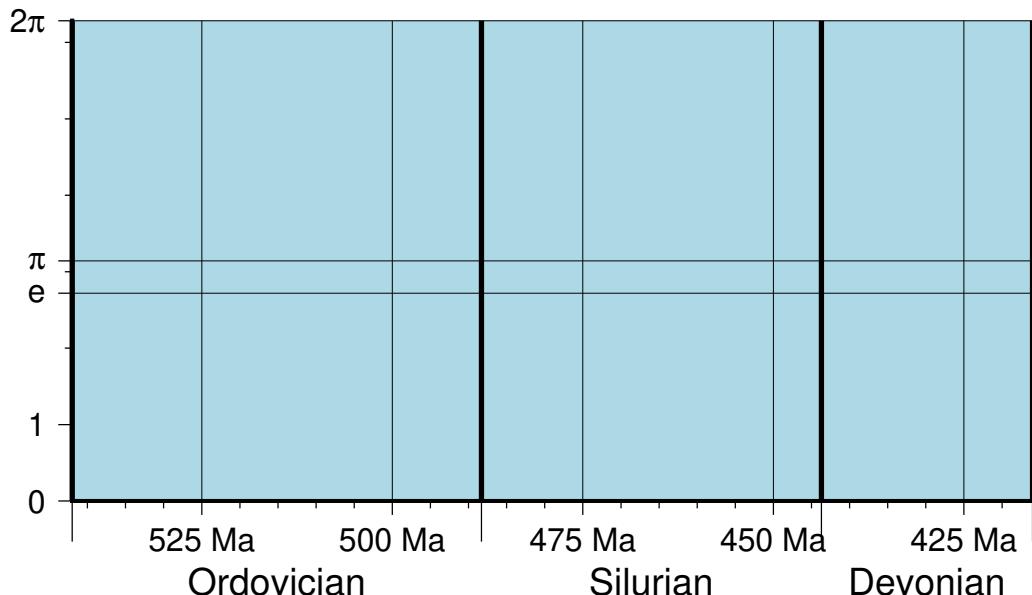


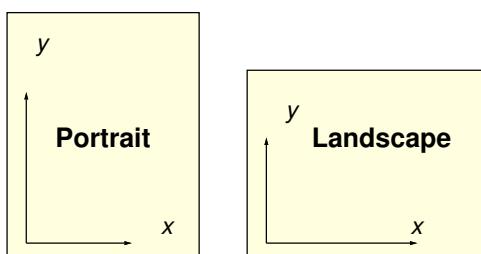
图 8.14: 自定义坐标轴

8.4 -P 选项

-P 选项用于设置纸张方向为 Portrait 模式。

取一张 A4 纸放在桌面上, 有两种放置的方式: 竖着放和横着放。如图所示:

Source Code



图中左边把纸张竖着放的称为 Portrait (肖像) 模式, 右边把纸张横着放的方式称为 Landscape (风景) 模式。记起来也很简单, 人是瘦长的, 所以人的肖像照需要竖着拍, 风景是矮胖的, 就需要横着拍。

由于历史原因, GMT 中画布的默认放置方式是 Landscape 模式。使用 -P 选项可以设置使用 Portrait 模式, 也可以设置 GMT 参数 *PS_PAGE_ORIENTATION* 为 *portrait* 来修改纸张方向。

注解: -P 选项设置的是画布的属性, 对于由多个命令绘制而成的图片来说, 只有第一个命令的 -P 选项是有效的。因而其他绘图命令中使用或不使用 -P 选项并不会影响绘图效果。

8.5 -V 选项

-V 选项可以控制命令的 verbose 等级以显示不同等级的命令执行信息，比如正在读取的文件名、读入的数据行数等。显示这些信息可以帮助用户判断命令是否执行正确。

verbose 共有 6 个等级，等级越高输出的信息越多，高等级会在低等级的基础上输出更多的信息。6 个等级从低到高分别为：

- **-Vq**: quiet 模式，不输出任何错误和警告信息
- **-Vn**: nomral 模式，仅输出致命错误信息和警告信息。当不使用 -V 选项时则默认为该等级
- **-Vc**: compatibility 模式，额外输出数据兼容性相关的警告信息。仅当 *GMT_COMPATIBILITY* 值为 4 即兼容模式时才默认为该模式
- **-Vv** 或 **-V**: verbose，即输出错误、警告以及数据处理的进程信息
- **-Vl**: long 模式，会额外输出详细的进程报告
- **-Vd**: debug 模式，会输出大量调试信息

-V 选项仅对当前命令有效，若希望所有命令都具有某个 verbose 级别，可以通过设置 GMT 参数 *GMT_VERBOSE* 的值来实现。

8.6 -U 选项

-U 选项用于在图上绘制一个带有 GMT 标识的时间戳。其语法为：

```
-U[<just>/<dx>/<dy>/] [c|<label>]
```

- -U 不加任何参数时会在当前图的左下角添加一个带 GMT 标识的时间戳
- -U<just>/<dx>/<dy> 用于设置时间戳的锚点（见[锚点](#)）及时间戳相对于当前坐标原点的位置，比如 -UBL/-1c/-1c
- -U<label> 会在时间戳后打印字符串 <label>，比如 -U"This is string"
- -Uc 会在时间戳后打印当前命令

GMT 参数中有一些与时间戳相关的参数：

- *MAP_LOGO* 控制是否绘制时间戳，默认值为 FALSE
- *MAP_LOGO_POS* 用于控制时间戳的位置
- *FORMAT_TIME_STAMP* 用于控制时间戳的显示格式
- *FONT_LOGO* 时间戳中文本的字体

```
gmt psbasemap -R0/10/0/5 -JX10c/3c -Bx1 -By1 -P \
-UBL/-1.5c/-1.5c/"This is a GMT logo" > GMT_-U.ps
```

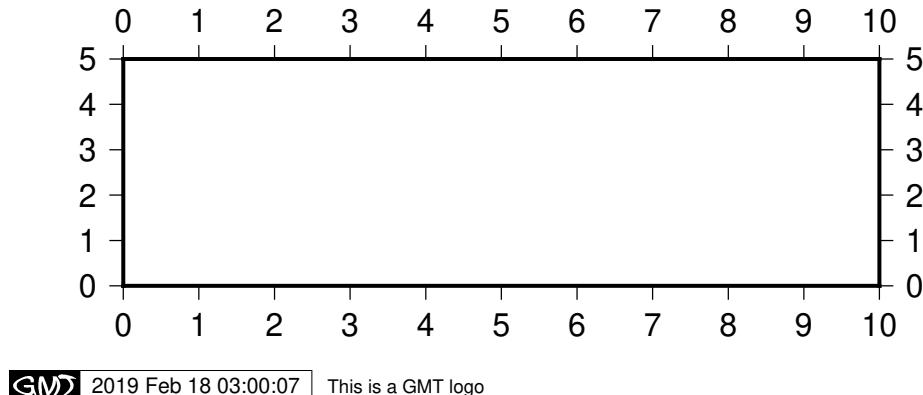


图 8.15: -U 选项加时间戳

重要: 每个带有 -U 选项的绘图命令都会绘制一遍时间戳, 因而只需要一个绘图命令中使用 -U 即可。

8.7 -K 和 -O 选项

-K 选项和 -O 选项是 GMT 绘图中常用的两个选项, 也是经常产生错误的两个选项。

8.7.1 图层

要理解 -K 选项和 -O 选项, 首先要理解图层的概念。如果你使用过 PhotoShop 或者在 PowerPoint 中画过图, 应该很容易理解这一概念。所谓图层, 可以理解成一张画有线条与文字的透明胶片, 将这些透明胶片按照顺序一张张重叠起来, 即构成了最终的图片。以 PhotoShop 中的图层为例, 使用图层的好处在于, 可以将某一图层单独拿出来修改, 而不影晌到其他图层。

图层的概念也适用于 GMT。通常来说, 用 GMT 绘图时是不可能用一个命令就画出想要的效果的。比如要画一个震中和台站的分布图, 通常需要将如下几个命令组合起来:

```
# 本示例中命令不完整, 仅供演示用途

# 绘制海岸线
gmt pscoast ... > map.ps
# 用五角星表示地震位置
gmt psxy event.dat -Sa0.5c ... >> map.ps
# 用三角形表示台站位置
gmt psxy station.in -St0.5c ... >> map.ps
```

每一个绘图命令本质上都是在向 PS 文件中新增了一个图层, 所以上面生成的 PS 文件中共有三个图层, 第一个图层仅包含了海岸线, 第二个图层仅包含一些五角星, 第三个图层仅包含一些三角形。这些图层按照顺序叠加起来, 就组成了我们最终看到的图片。在图层叠加过程中, 后面的图层中的非透明区域会覆盖前面的图层, 比如第二个图层中填充了颜

色的五角星就有可能把前一图层中的某一小段海岸线给覆盖掉。

8.7.2 PS 文件结构

在理解了图层的概念之后在来说说 PS 文件的结构。

任何一个 PS 文件都由 header、body 和 trailer 组成，可以理解成头、身体和尾巴。其中头部的作用是初始化 PS 文件，比如设置纸张大小、纸张方向、载入字体等；身体部分则是真正的绘图部分，包含了每个绘图命令绘制的图层；尾巴部分则用于控制 PS 文件的显示。

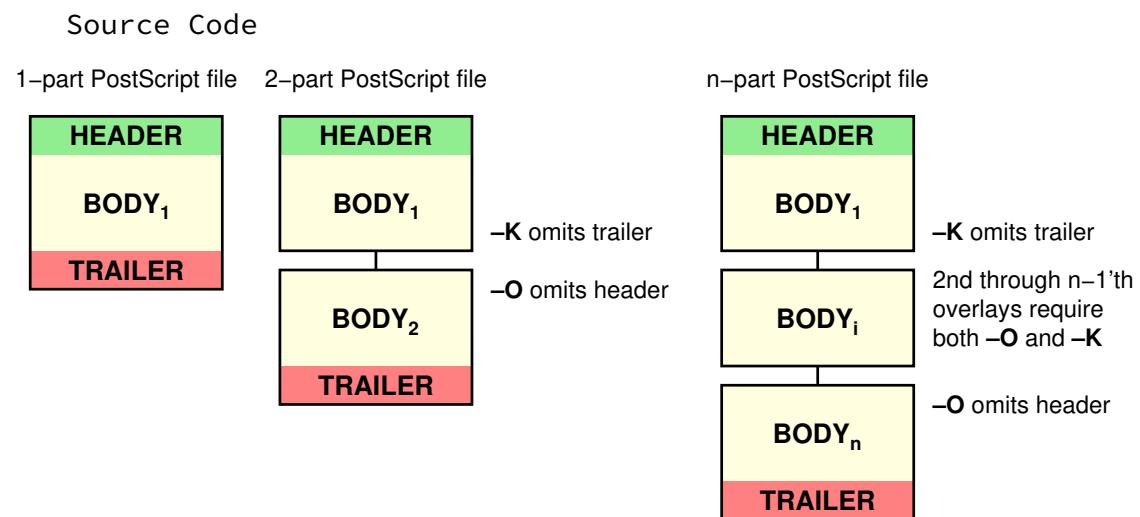


图 8.16: **-K** 和 **-O** 选项的原理

因而，一个 PS 文件中，只有一个头和尾巴，可以有零个、一个或多个身体。

8.7.3 -K 和 -O 的作用

在不使用 **-K** 和 **-O** 选项的情况下，GMT 的每个绘图命令都会产生完整的 PS 代码，即包含了头部、身体和尾部。而在使用多个命令绘制一张图片的多个图层时，则要保证第一个绘图命令必须省略尾部，中间的绘图命令必须省略头部和尾部，最后一个绘图命令必须省略头部。

-K 选项的作用是省略尾部，即意味着后面还有追加新的图层。**-O** 选项的作用是省略头部，即这个图层要覆盖在前一图层上。

因而，在一张图中绘制多个图层时，绘制第一个图层的命令需要使用 **-K** 选项省略尾部，绘制中间图层的命令需要使用 **-K -O** 选项，绘制最后一个图层的命令则需要使用 **-O** 选项。

因而，对于含多个图层的图片来说，绘图命令如下所示：

```
gmt pscmd1 ... -K > out.ps  
gmt pscmd2 ... -K -O >> out.ps  
...  
gmt pscmd3 ... -K -O >> out.ps  
gmt pscmd4 ... -O >> out.ps
```

8.7.4 -K 和 -O 的使用

总结一下 -K 和 -O 选项的用法。对于有 n 个图层的 PS 文件：

- 第 1 个绘图命令需要使用 -K 和重定向符号 >
- 第 2 到 n-1 个绘图命令需要使用 -K -O 和重定向符号 >>
- 第 n 个绘图命令需要使用 -O 和重定向符号 >>

上面总结的 -K 和 -O 选项的用法看上去很简单，但实际绘图中却非常容易出错。

想象一下这样一个场景，你需要画一张稍复杂的图，所以需要在脚本中写一堆命令来完成绘图。当你写脚本的时候正确地使用了 -K 和 -O 选项以及重定向符号，然后也初步完成了绘图。且不说绘图的过程有多艰辛，在完成初步绘图后你可能需要对这张图做一些微调，比如下面这些情形：

- 由于后面的图层会覆盖前面的图层，而有些覆盖是你不想要的，所以你可能会希望调换两个绘图命令的顺序。比如需要调整最开始的两个命令的顺序，或者要调整最后两个命令的顺序，你是否记得在调整顺序的时候要修改 -K 和 -O 选项以及重定向符号？
- 比如需要在第一个绘图命令前再加一个绘图命令，你是否记得要把原来的第一个绘图命令加上 -O 选项以及 >> ?
- 比如需要在最后一个绘图后面再加一个绘图命令，你是否记得要把原来的最后一个绘图命令加上 -K 选项？

上面列举的一些情形，即便是对于 GMT 比较熟悉的人，也偶尔会因为一时粗心而弄错。在被 -K 和 -O 的用法坑了几次之后，就得想一想，有没有办法可以避免这个问题呢？下面展示了一些小技巧：

```
#!/bin/bash  
  
PS=map.ps  
J=JX5c/5c  
R=0/10/0/10  
  
# 写入文件头  
gmt psxy -J$J -R$R -T -K > $PS  
  
# 真正的绘图命令  
gmt xxxx -J$J -R$R ... -K -O >> $PS  
gmt xxxx -J$J -R$R ... -K -O >> $PS  
  
# 写入文件尾  
gmt psxy -J$J -R$R -T -O >> $PS
```

解释一下:

- 对于需要用多个命令绘图的图片, 最好将命令写到脚本文件中, 这样方便记录和调试命令
- 上面的脚本是 bash 脚本, 并将常出现的值定义成变量, 以方便使用和修改
- `gmt psxy ... -T -K` 只向 PS 文件中写入头部
- `gmt psxy ... -T -O` 只向 PS 文件中写入尾部
- 中间的全部绘图命令统一用 `-K -O >>`。这样的统一使得, 任意调整命令顺序或删减命令, 都不需要修改 `-K`、`-O` 和重定向符号!
- 实际使用中, 第一个 `gmt psxy -T -K` 命令也可以考虑改成只绘制底图边框的 `gmt psbasemap -B ... -K`

注解: `-K` 和 `-O` 以及重定向符号是在 GMT 绘图过程中最容易出错的地方。GMT6 提供的现代模式中完全抛弃了 `-K` 和 `-O` 以及重定向, 极大地简化了绘图代码, 并减少了出错的可能性。推荐所有用户使用 GMT6 提供的现代模式。

8.8 -X 和 -Y 选项

绘图时需要指定绘图的原点在哪里。对于一个新图, GMT 默认的原点位于离纸张左下角 ($1i, 1i$) 处。留出 1 英寸是为了给边框外的标注和标签留下空间。

在一张图上绘制多个子图时, 通常需要控制底图原点的位置, `-X` 和 `-Y` 选项就用于移动底图原点。

`-X` 和 `-Y` 的用法是一样的, 所以下面仅以 `-X` 选项为例介绍其用法。其语法为:

```
-X[a|c|f|r][<xshift>[<u>]]
```

其中 `<xshift>` 新原点相对于当前原点的 X 方向偏移量, `<u>` 为偏移量的单位。

在偏移量之前加上不同的字符表示不同的含义:

- `-X2i` 或 `-Xr2i`: 在原底图原点的基础上沿 X 方向偏移 2 英寸得到新底图原点
- `-Xa5c`: 在原底图原点的基础上沿 X 方向偏移 5 厘米得到临时底图坐标, 当前命令执行完成后, 底图原点复原到原底图原点
- `-Xc`: 将底图绘制在整张纸的中心 (对于现代模式而言, 无意义)
- `-Xc3c`: 在纸张中心的基础上沿 X 方向偏移 3 厘米得到新底图原点
- `-Xf4c`: 在纸张左下角的基础上沿着 X 方向偏移 4 厘米得到新底图原点

若只使用了 `-X` 和 `-Y` 而不接任何额外参数, 则使用前一个 GMT 命令使用该选项时的参数值。

1. 第一个绘图命令的默认偏移量由 `MAP_ORIGIN_X` 和 `MAP_ORIGIN_Y` 控制
2. 当 PS 文件中有多个底图时才需要使用 `-X` 和 `-Y` 选项, 不要仅仅为了在某个地方加

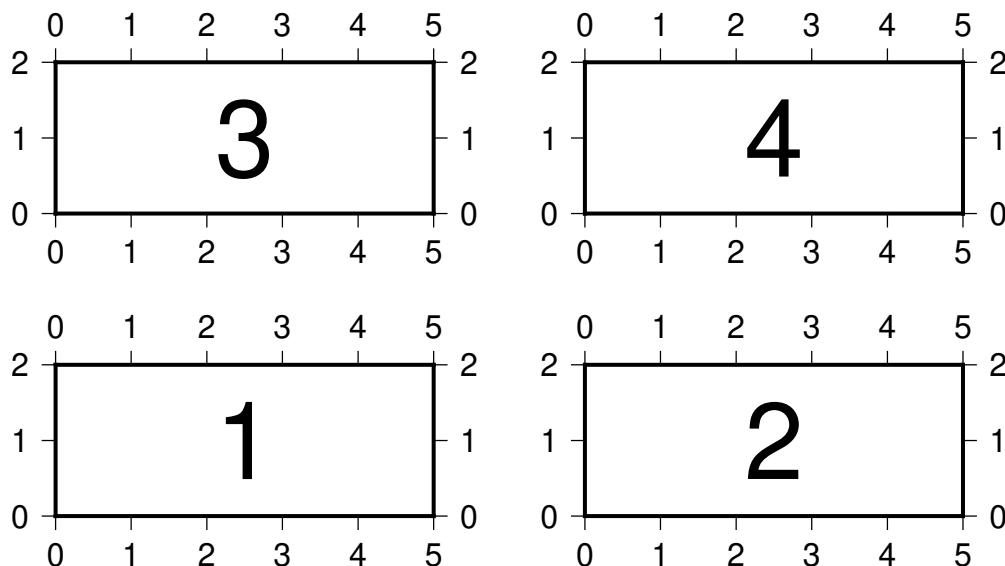
几个文字就使用这两个选项

-X 和 -Y 选项的用法介绍起来有些难度, 多试试就好, 下面举个简单的例子:

```
gmt psbasemap -Jx5c/2c -R0/5/0/2 -B1 -K > test.ps
gmt psbasemap -J -R -B1 -K -O -X7c >> test.ps
gmt psbasemap -J -R -B1 -K -O -X-7c -Y4c >> test.ps
gmt psbasemap -J -R -B1 -K -O -X7c >> test.ps
```

上图用四个 `psbasemap` 命令绘制了四张底图, 绘图效果如下:

Source Code



解释:

1. 第一个命令的绘图原点位于纸张左下角 ($1i, 1i$) 处, 绘制底图 1
2. 第二个将绘图原点右移了 7 cm, 绘制底图 2
3. 为了绘制底图 3, 第三个命令将底图左移了 7 cm, 并上移 4 cm
4. 第四个命令在底图 3 的基础上右移 7 cm, 绘制底图 4

8.9 -a 选项

GMT 自身是无法读取 shapefile 等格式的地理空间数据的。需要利用 [GDAL](#) 中的 `ogr2ogr` 工具将其他格式的地理空间数据转换为 GMT 可识别的 OGR/GMT 格式的数据文件, 转换过程中保留了地理空间数据的非空间元数据。关于该格式的介绍, 见[兼容 OGR 的 GMT 矢量数据格式](#)。

`-a` 选项用于建立非空间元数据与 GMT 输入/输出数据之间的联系。该选项的语法为:

```
-a[<col>=<name>[,...]
```

`-a` 选项后接一个或多个用逗号分隔的 `<col>=<name>`, 其作用在于将 OGR/GMT

格式的数据文件中非空间元数据 <name> 字段作为输入/输出数据的第 <col> 列。若不指定 <col>, 则默认列数为 2, 并依次增加。

例如 -a2=depth 会从数据文件中读取 X 和 Y 列信息, 并从非空间数据的 depth 字段中读取值作为输入的第三列。

也可以通过将 <col> 设置成 D|G|L|T|W|Z 来将非空间数据与其他属性联系起来, 比如标签、填充色、画笔属性、用于查找颜色的 Z 值等。该机制与在多段数据的段头记录中加上参数是等效的。

GMT 也可以用于输出 OGR/GMT 格式的数据文件, 此时可以使用 <col>=<name>[:<type>] 来指定将输出数据的第 <col> 列以 <type> 数据类型保存到非空间字段 <name> 中。与输入类似, <col> 也可以取 D|G|L|T|W|Z 中的一个。对于输出而言, 还需要加上 +g<geometry> 来指定数据的几何类型, <geometry> 可以取为 [M]POINT|LINE|POLY。若加上 +G, 则程序会自动将跨越国际日期变更线的线段或多边形分成多段。

8.10 -b 选项

GMT 中的很多程序都需要读入表数据, 表数据可以是 ASCII 格式、二进制格式, 也可以是 netCDF 格式。

ASCII 格式的数据很直观, 可以直接看到有多少列数据。而二进制格式的数据, GMT 在读入数据时是无法知道有多少列数据, 也无法知道每一列数据的格式的。因而就需要使用 -b 选项告诉 GMT 输入或输出数据的具体格式。

-b 的语法是:

```
-b[i|o][<ncols>][<type>][w][+L|+B]
```

-bi 表示对输入有效, -bo 表示对输出有效。其后可以跟一个或多个用逗号分隔的 <ncols><type>, 表示有 <ncols> 列类型为是 <type> 的数据。其中 <type> 可以取:

- c: 即 int8_t, 字符型
- u: 即 uint8_t, 无符号字符型
- h: 即 int16_t, 两字节有符号整型
- H: 即 uint16_t, 两字节无符号整型
- i: 即 int32_t, 四字节有符号整型
- I: 即 uint32_t, 四字节无符号整型
- l: 即 int64_t, 八字节有符号整型
- L: 即 uint64_t, 八字节无符号整型
- f: 四字节单精度浮点型
- d: 八字节双精度浮点型
- x: 不代表某种数据类型, 表示跳过 <ncols> 个字节

例如, 要读入的二进制数据中有 3 列, 前两列是单精度浮点型, 最后一列是 int32_t

型，则在读入时可以用 `-bi2f,1i`。

对于每一个 `<ncols><type>` 还可以加上 `w` 表示对数据强制做字节序转换。

`+L` 或 `+B` 表示整个数据按照 little-endian 或 big-endian 字节序读入。

若未指定 `<ncols>` 则假定 `<type>` 适用于所有列，读数据时 `<ncols>` 的值由数据期望得到的值决定。

可以使用 `-h<n>` 选项跳过二进制文件的前 `<n>` 个字节。

8.11 -d 选项

GMT 中缺失的值会用 `NaN` 表示。某些情况下，用户可能会用类似 `-99999` 这样的数据表示数据缺失，对于 GMT 而言，`-99999` 并不是一个特殊的值，因而就需要在将数据传递给 GMT 时告诉 GMT 某个特殊值表示数据缺失。

`-d` 选项的语法是：

```
-d[i|o]<nodata>
```

`-di` 和 `-do` 表示仅对输入或输出有效，其作用是：

- 对于输入数据，将数据中的 `<nodata>` 替换成 `NaN`；
- 对于输出数据，将 `NaN` 替换成 `<nodata>`。

8.12 -e 选项

通常，GMT 模块在读入 ASCII 表数据时会处理读入的所有数据记录。`-e` 选项提供了一个内置的模式匹配器，使得模块可以只处理匹配指定模式或正则表达式的数据记录。

`-e` 选项的完整语法是：

```
-e[~]"<pattern>"  
-e[~]<regexp>[i]
```

`-e~` 表示取反，即只处理不匹配指定模式或正则表达式的数据记录。如果数据记录中包含了字符 `~`，则需要使用 `\~` 进行转义。`i` 表示对大小写不敏感。

8.13 -f 选项

在某些情况下，需要显式指明当前输入/输出数据中每一列的含义，这可以通过 `-f` 选项来实现。

默认情况下，`-f` 选项对输入输出同时有效，`-fi` 表明该选项仅对输入数据有效，`-fo` 表明该选项仅对输出数据有效。

输入/输出数据中每一列的含义可以通过一系列以逗号分隔的字符串来表示。每个字

字符串包括两部分：列号以及数据类型。数据类型可以取 `x`、`y`、`T`、`t`、`f`，分别表示这一列是经度、纬度、绝对时间、相对时间或普通的浮点数。列号是一个从零开始起算的整数（比如第 6 列的列号为 5），当多个连续的列有相同的数据类型时，也可以指定一个列号的范围，列号范围的格式为 `<start>[:<inc>]:<stop>`，若未给定 `<inc>` 则默认为 1，比如 `1:5` 表示第 2 至 6 列数据。

例如，`-f10y,1x,3:4T` 表明输入数据中第一列是纬度，第二列是经度，第 3、4 列是绝对时间，其他列数据则假定是默认的浮点数类型。

`-fg` 是 `-f0x,1y` 的缩写，表明输入/输出是地理坐标。除此之外，还可以使用 `-fp<unit>`，该选项可以使用投影之后的坐标作为输入，在读入数据时投影后的坐标值会自动转换为经纬度值。

8.14 -g 选项

在处理多段数据时，GMT 提供了三种机制来决定文件中数据的分段情况：

1. 在[表数据](#) 中已经介绍了，使用数据段头记录来标记一段数据的开头
2. 若输入数据中，某个记录的某个关键列的值为 NaN，则也可以用于将该记录作为数据段的开始标识
 - 当 `IO_NAN_RECORDS` 为 `skip` 时，这些包含 NaN 值的记录会被自动跳过
 - 当 `IO_NAN_RECORDS` 为 `pass` 时，这些包含 NaN 值的记录会被当做数据段的开始标识
3. 也可以使用 `-g` 选项，通过判断两个相邻的数据点是否符合某个准则来决定数据分段

`-g` 选项的完整语法为：

```
-g[a]x|y|d|X|Y|D| [<col>]z[±]<gap>[u]
```

- `x` 两点的 X 坐标跳变超过 `<gap>`
- `y` 两点的 Y 坐标跳变超过 `<gap>`
- `d` 两点的距离超过 `<gap>`
- `<u>` 是单位
- `X|Y|D` 用于表示数据投影到纸上后 X 坐标、Y 坐标和纸上距离的跳变
- 若想要检查特定列是否满足分段准则，可以用 `[<col>]z`，`<col>` 的默认值为 2，即第三列
- `+<gap>` 表示前一数据减去当前数据超过 `<gap>` 则分段
- `-<gap>` 表示当前数据减去前一数据超过 `<gap>` 则分段
- `<gap>` 表示两个数据的绝对值超过 `<gap>` 则分段

该选项可重复多次，以指定多个分段准则，默认情况下，若符合任意一个准则则分段，可以使用 `-ga` 选项，表明仅当所有准则都满足时才分段。

8.15 -h 选项

-h 选项用于指定在读入数据时需要跳过的头段记录数。其语法为:

```
-h[i|o]<n>[+c][+d][+r<remark>][+t<title>]
```

其中 <n> 表示数据中的头段记录数。i 和 o 表示仅对输入/输出有效。

输入文件中的空行以及以“#”开头的行都会被自动当做头段记录，因而会被自动跳过，在指定 <n> 时不需要再考虑空行以及以“#”开头的行。

几种常见的用法：

1. -h: 使用 GMT 参数 *IO_N_HEADER_RECS* 的值作为头段记录数（默认值为 0）
2. -h3: 表示跳过 3 个头段记录

对于二进制输入文件，<n> 表示输入数据中要跳过的字节数，或输出数据中用空白字符补充的字节数。

对于输出文件而言，默认情况下，会将输入文件中的头段信息原样输出。使用 +d 以删除之前的头段信息，使用 +c 会将列名写到输出的头段记录中，+r<remark> 和 +t<title> 则可以分别加一个 remark 和 title 语句到输出的头段记录。

8.16 -i 和 -o 选项

经常遇到的情况是，已有的数据有很多列，而某个命令只需要其中的某几列；或者某个命令的输出有很多列，却只想要输出其中的某几列。

常见的做法是使用 gawk 以及管道对输入输出数据做处理。其实，GMT 提供的 -i 和 -o 选项可以满足日常的大多数需求了。

-i 选项的语法是：

```
-i<col>[+l][+s<scale>][+o<offset>][,...]
```

默认情况下，GMT 会读取输入数据中的全部列。-i 选项后接以逗号分隔的列号或列号范围，以指定输入数据中要使用哪些数据列以及这些数据列的顺序，例如 -i3,6,2 表示读入数据中的第 4、7、3 列。也可以多次使用某一列数据，比如 -o3,1,3 则会两次输出第四列数据。

列号范围的格式为 <start>[:<inc>]:[<stop>]，若省略 <inc> 则默认其值为 1，比如 1:3。

每个列号后还可以加上额外的字符以对每个数据做简单的处理。比如 +l 表示对当前列取 \log_{10} ，+s<scale> 表示将当前列乘以比例因子 <scale>，+o<offset> 表示将当前列的值加上 <offset>。比如 -i2+s2+o10,6,3 表示读入数据的第 3、7、4 列，并对第 3 列数据乘以 2 再加上 10。

-o 选项的语法为:

```
-o<col>[,...]
```

即 -o 只能控制输出哪些列, 不能对输出的数据做进一步处理。

8.17 -n 选项

-n 选项用于控制二维网格数据重采样过程中的插值算法。其语法为:

```
-n[b|c|l|n][+a][+b<BC>][+c][+t<threshold>]
```

- b 表示 B-spline 平滑算法
- c 表示使用 bicubic 插值算法 (默认值)
- l 表示 bilinear 插值算法
- n 表示 nearest-neighbor 值。
- +a: 关闭抗混淆 (仅在算法支持的前提下有效)
- +b<BC>: 设置网格的边界条件。<BC> 可以取 g、p、n, 分别代表地理边界条件、周期边界条件和自然边界条件。对于后两种边界条件, 可以在后面加上 x 或 y 表示边界条件仅对一个方向有效。比如 -nb+bnxpy 表明 X 方向使用自然边界条件, Y 方向使用周期边界条件
- +c: 假设原网格的 Z 值范围为 zmin 到 zmax, 插值后的网格范围可能为超过这一范围, 使用 +c 则将超过 zmin 和 zmax 的部分裁剪掉
- +t<threshold> 见官方文档

8.18 -r 选项

GMT 中的 2D 网格文件有网格线配准和像素配准两种配准方式, 详情见[网格配准](#)一节。

默认情况下, GMT 认为输入/输出的网格文件都采用网格线配准方式, 使用 -r 选项, 则 GMT 会认为输入/输出的网格文件为像素配准方式。

对于相同的网格区域 (-R) 和网格间隔 (-I) 而言, 像素配准会比网格线配准少一列和一行数据。

8.19 -p 选项

在一张纸上画一个 2D 矩形, 当从不同的方向去看这个矩形时, 可以看到不同的形态。-p 选项可以用于选择从怎样的视角去看一张图。该选项的语法为:

```
-p[x|y|z]<azim>/<elev>[/<zlevel>][+w<lon0>/<lat0>][/<z0>][+v<x0>/<y0>]
```

<azim> 和 <elev> 指定视角的方位角和海拔, 默认值是 180/90, 即视角在正上方。此方位角是相对于北方向顺时针旋转的角度, <elev> 是相对于纸张平面旋转的角度,

向上为正向下为负。

对于 3D 绘图而言(使用 $-JZ|z$), 默认情况下会在 Z 轴底部绘制 XY 平面内边框。使用 $-p$ 选项再加上 $<zlevel>$ 可以指定在某个 $Z=<zlevel>$ 处绘制 2D 边框及标注。也可以用 $-px|y$ 在 $X=<xlevel>$ 或 $Y=<ylevel>$ 处绘制 YZ 平面或 XZ 平面。默认使用 $-pz$, 即在 $Z=<zlevel>$ 处绘制 XY 屏幕。

对于使用 $-p$ 选项画出来的图, 如果是作为动画的一帧, 可以加上 $+$ 以规定数据域的中心, 也可以用 $+w<lon0>/<lat0>/<z>$ 指定一个特殊的世界坐标点, 该点会被投影到纸张的中心; 也可以用 $+v<x0>/<y0>$ 指定投影后视点的坐标。

8.20 -s 选项

$-s$ 选项用于压制 Z 值等于 NaN 的记录的输出。其语法为:

```
-s[<cols>][a|r]
```

默认情况下, 命令会输出所有记录, 包括那些 Z 值等于 NaN 的记录。使用该选项可以抑制 Z 值为 NaN 的记录的输出。

$<\text{cols}>$ 是一系列用逗号分隔的列号或者列号范围, 其中列号范围的格式为 $<\text{start}>[:<\text{inc}>]:<\text{stop}>$, 若省略 $<\text{inc}>$ 则默认其值为 1。

比如 $2,5,7$ 和 $2:4$, 表示检测这些列数据是否是 NaN 。

r 表示 reverse, 即只输出满足 NaN 的记录行。 $-sa$ 表示所有列中只要有一列为 NaN 即抑制其输出。

8.21 -t 选项

$-t$ 选项用于设置当前图层的透明度, 其语法为:

```
-t<transp>
```

其中 $<\text{transp}>$ 表示透明度, 可以取 0 到 100 内的数字。默认值为 0, 即不透明, 50 表示半透明, 100 则表示完全透明。

PS 格式不支持透明效果。在使用 $-t$ 选项后, PS 文件中看不到透明效果, 需要将 PS 格式转换为 PDF、PNG 等格式中才能看到效果。

使用如下命令将 PS 文件转换为 PDF:

```
gmt psconvert -A -P -Tf xxxx.ps
```

8.22 -x 选项

在编译 GMT 时若打开了 OpenMP 选项，则 GMT 中某些模块在运行时可以通过 OpenMP 加快计算速度。默认情况下，这些模块会尝试使用所有可用的核。**-x** 选项用于限制所使用的核数，其语法为：

```
-x[[-]<n>]
```

例如：

- **-x8** 表示仅使用 8 个核，若计算机的最大核数小于 8，则使用计算机的全部核
- **-x-4** 表示使用 all-4 个核，也就是说给其他程序留下四个核。若 all-4<1，则使用 1 个核

支持该选项的模块包括：[grdfilter](#)、[grdlandmask](#)、[grdmask](#)、[grdmath](#)、[grdsample](#)、[greenspline](#)、[sph2grd](#)、[gpsgriddler](#)、[talwani2d](#)、[talwani3d](#)、[surface](#)、[x2sys_solve](#)。

8.23 -: 选项

GMT 在读入数据时，会默认第一列是 X 值，第二列是 Y 值。对于地理数据而言，即第一列是经度、第二列是纬度。若要读入的数据中第一列是纬度、第二列是经度，则需要先交换第一列和第二列再读入，可以使用 **-:** 选项实现前两列数据的交换。

默认情况下，该选项同时对输入和输出生效：

- **-:i** 表明该选项仅交换输入数据的前两列，等效于使用 **-i1,0** 选项
- **-:o** 表明该选项仅交换输出数据的前两列，等效于使用 **-o1,0** 选项

注解： 该选项仅用于交换输入/输出的前两列数据，**-R** 选项中始终是经度在前，网格文件中经度始终是第一维度。

保护环境，从阅读电子文档开始！

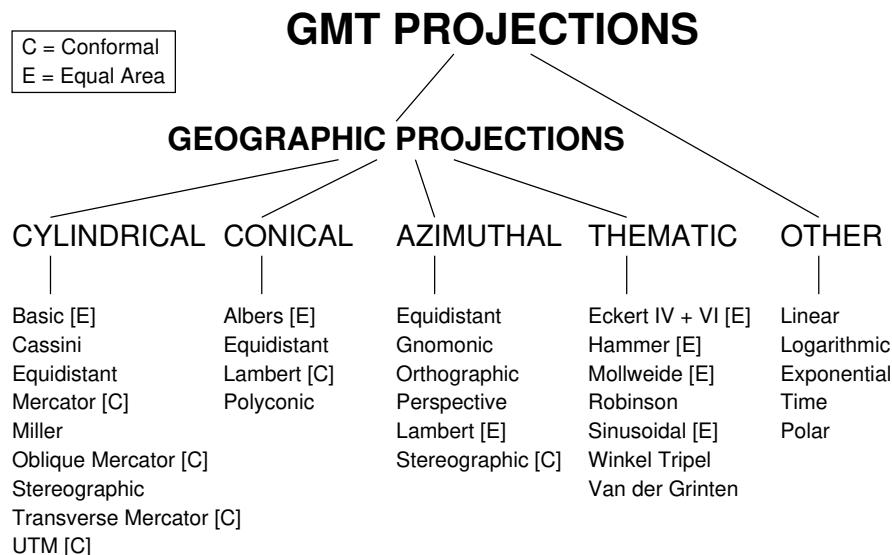
第 9 章 投影方式

GMT 支持几十种不同的投影方式，这些投影方式可以分为三类：

1. 笛卡尔线性投影
2. 极坐标投影
3. 地图投影

其中地图投影又可以细分为几十种不同的投影方式。

Source Code



9.1 -Jx: 笛卡尔变换

GMT 中笛卡尔坐标变换分为三类：

- 线性坐标
- \log_{10} 坐标
- 指数坐标

在开始之前，先用 `gmtmath` 生成两个数据以供接下来示例使用：

```
gmt gmtmath -T0/100/1 T SQRT = sqrt.txt
gmt gmtmath -T0/100/10 T SQRT = sqrt10.txt
```

9.1.1 笛卡尔线性坐标

笛卡尔线性坐标可以通过四种方式指定：

- $-Jx<scale>$ X 轴和 Y 轴拥有相同的比例尺 $<scale>$
- $-JX<width>$ X 轴和 Y 轴拥有相同的长度 $<width>$
- $-Jx<xscale>/<yscale>$ 分别为 X 轴和 Y 轴指定不同的比例尺
- $-JX<width>/<height>$ 分别为 X 轴和 Y 轴指定不同的长度

笛卡尔线性坐标的使用场景可以分为三类：

1. 常规的浮点数坐标
2. 地理坐标
3. 日期时间坐标

常规浮点数坐标

对于常规的浮点型数据而言，选择笛卡尔线性坐标意味着对输入坐标做简单的线性变换 $u' = au + b$ ，即将输入坐标 u 投影到纸张坐标 u' 。

下面的命令将函数 $y = \sqrt{x}$ 用笛卡尔线性变换画在图上。

```
gmt psxy -R0/100/0/10 -JX3i/1.5i -Bag -BWSne+gsnow -Wthick,blue,- -P -K sqrt.txt > GMT_linear.ps
gmt psxy -R -J -St0.1i -N -Gred -Wfaint -O sqrt10.txt >> GMT_linear.ps
```

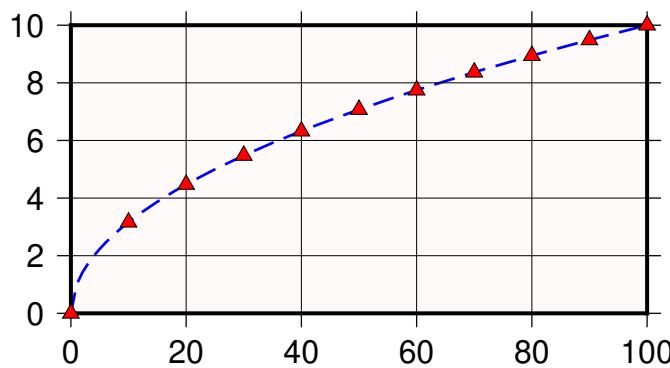


图 9.1：笛卡尔坐标的线性变换

说明：

- 正常情况下，X 轴向右递增，Y 轴向上递增。有些时候可能需要 X 轴向左递增或者 Y 轴向下递增（比如 Y 轴是深度时），只要将轴的比例尺或者轴长度设置为负值即可。
- 若指定 X 轴的长度，并设置 Y 轴的长度为 0，则会根据 X 轴的长度和范围计算出 X 轴的比例尺，并对 Y 轴使用相同的比例尺，进而计算出 Y 轴的长度，即 $-JX10c/0c$ ， $-JX0c/10c$ 同理。

地理坐标

理论上地理坐标应该用地理投影画，而不应该用线性投影，但是有时候可能的确需要使用线性投影。用线性投影绘制地理坐标时会碰到一个问题，即经度有一个 360 度的周期性。因而在使用线性投影时需要通知 GMT 数据实际上是地理坐标。有三种办法：

1. 在 -R 后、数据范围前加上 g 或 d, 比如 -Rg-55/305/-90/90
2. 在 -Jx 或 -JX 选项的最后加上 g 或 d, 比如 -JX10c/6cd
3. 使用 -fg 选项

下面的例子用线性投影绘制了一个中心位于 125°E 的世界地图。

```
gmt set MAP_GRID_CROSS_SIZE_PRIMARY 0.1i MAP_FRAME_TYPE FANCY FORMAT_GEO_MAP ddd:mm:ssF
gmt pscoast -Rg-55/305/-90/90 -Jx0.014i -Bagf -BWSen -Dc -A1000 -Glightbrown -Wthinnest \
-P -Slightblue > GMT_linear_d.ps
```

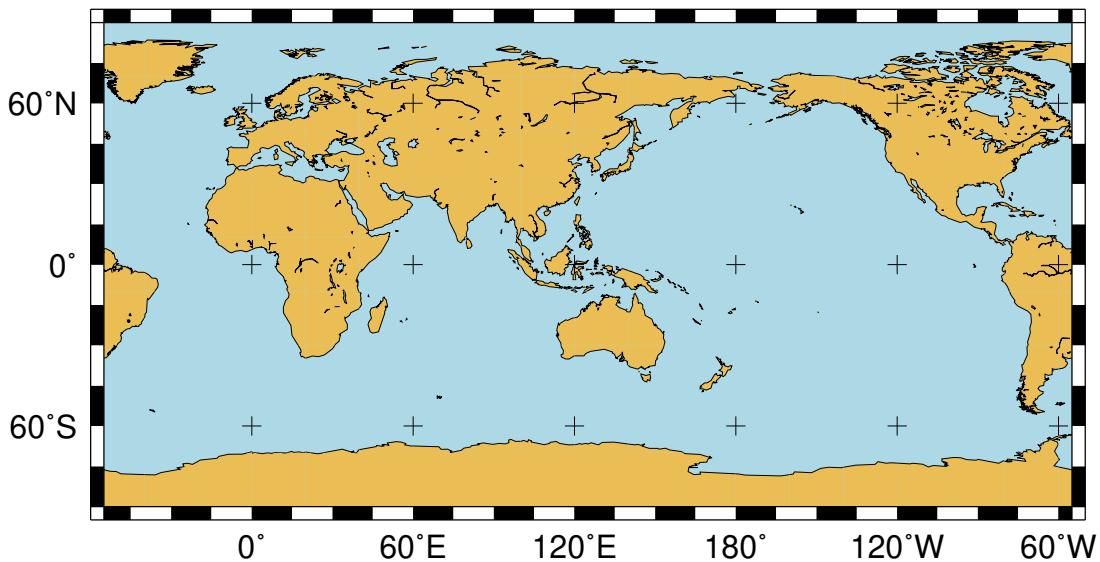


图 9.2: 地理坐标的线性变换

日期时间坐标

时间日期坐标也可以用线性投影绘制, 此时需要告诉 GMT 输入坐标是绝对时间还是相对时间。

可以通过在 -Jx 或 -JX 的最后加上 T 或 t, 不过实际上 -R 选项中已经指定了时间范围, 所以没有必要在 -J 和 -R 选项中都指定。当 -R 和 -J 选项给出的坐标类型相冲突时, GMT 会给出警告, 并以 -JX 选项为准。

```
gmt set FORMAT_DATE_MAP o TIME_WEEK_START Sunday FORMAT_CLOCK_MAP=-hhmm \
FORMAT_TIME_PRIMARY_MAP full
gmt psbasemap -R2001-9-24T/2001-9-29T/T07:0/T15:0 -JX4i/-2i -Bxa1Kf1kg1d \
-Bya1Hg1h -BWsNe+glightyellow -P > GMT_linear_cal.ps
```



图 9.3: 日期时间坐标的线性变换

9.1.2 笛卡尔对数投影

对数变换 \log_{10} 的数学表示是 $u' = a \log_{10}(u) + b$ ，可以通过在比例尺或轴长度后加上 l 指定。

下面的命令绘制了一个 X 轴为对数轴 Y 轴为线性轴的图。

```
gmt psxy -R1/100/0/10 -Jx1.5il/0.15i -Bx2g3 -Bya2f1g2 -BWSne+gbisque \
-Wthick,blue,- -P -K -h sqrt.txt > GMT_log.ps
gmt psxy -R -J -Ss0.1i -N -Gred -W -O -h sqrt10.txt >> GMT_log.ps
```

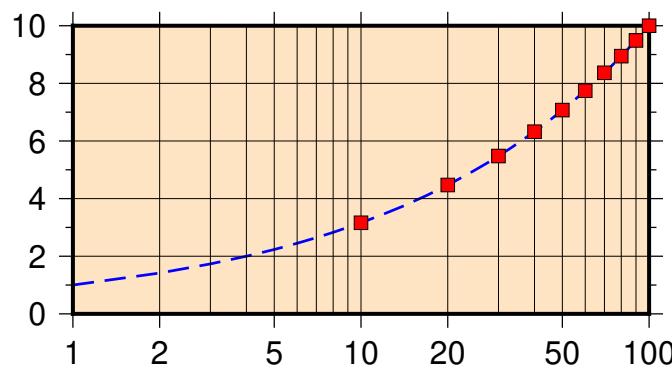


图 9.4: 对数投影

注意：若想要 X 轴和 Y 轴都使用对数投影，且 X 轴和 Y 轴比例尺不同，则必须在指定每个轴的比例尺时分别加上 l ，例如 $-JX10cl/6cl$ 。

9.1.3 笛卡尔指数投影

指数投影的函数表示是 $u' = au^b + c$ ，使得用户可以绘制类似 $x^p - y^q$ 这样的函数关系。如果选 $p=0.5$ 、 $q=1$ 则相对于绘制 x 与 \sqrt{x} 的函数曲线。

要使用指数投影，需要在比例尺或轴长度后加上 $p<\exp>$ ，其中 $<\exp>$ 是要使用的指数。

```
gmt psxy -R0/100/0/10 -Jx0.3ip0.5/0.15i -Bxa1p -Bya2f1 -BWSe+givory \
-Wthick -P -K sqrt.txt > GMT_pow.ps
gmt psxy -R -J -Sc0.075i -Ggreen -W -O sqrt10.txt >> GMT_pow.ps
```

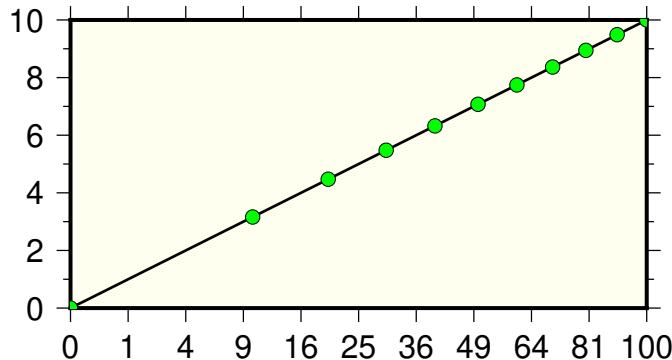


图 9.5: 指数变换

9.2 -Jp: 极坐标线性投影

-Jp 投影用于绘制极坐标数据 (即角度 θ 和半径 r)。

指定极坐标投影的语法为:

```
-Jp[a]<scale>[/<theta0>][r][z]
-JP[a]<width>/<theta0>[r][z]
```

其中

- -Jp<scale> 表示指定比例尺, -JP<width> 表示指定整张图的宽度
- 默认情况下, 角度 θ 是指相当于水平方向逆时针旋转的角度 (标准定义); 在 p 或 P 后插入 a 表明输入数据是相对于北方向顺时针旋转的角度 (地理学中的方位角)
- <theta_0> 表示对坐标轴进行旋转的角度, 默认值为 0
- z 表示将 r 轴标记为深度而不是半径
- r 表示将径向方向反转, 此时 r 轴范围应在 0 到 90 之间

下面给出了一些极坐标的示例以展示极坐标的用法:

```
gmt set FORMAT_GEO_MAP +D FONT_TITLE 14p,1,red
gmt psbasemap -JP5c -R0/360/0/1 -Bxa45f -B+t"-JP5c -R0/360/0/1" -Y10c -K > Jp.ps
gmt psbasemap -JPa5c -R0/360/0/1 -Bxa45f -B+t"-JPa5c -R0/360/0/1" -X8c -K -O >> Jp.ps
gmt psbasemap -JPa5c -R0/90/0/1 -Bxa45f -Ba0.2 -BWNse+t"-JPa5c -R0/90/0/1" -X8c -K -O >> Jp.ps
gmt psbasemap -JPa5c/45 -R0/90/0/1 -Bxa45f -Ba0.2 -BWNse+t"-JPa5c/45 -R0/90/0/1" -X-16c
->-Y-6.5c -K -O >> Jp.ps
gmt psbasemap -JPa5c/45 -R0/90/3480/6371 -Bxa45f -Ba -BWNse+t"-JPa5c/45 -R0/90/3480/6371
->" -X8c -Y1c -K -O >> Jp.ps
gmt psbasemap -JPa5c/45z -R0/90/3480/6371 -Bxa45f -Ba -BWNse+t"-JPa5c/45z -R0/90/3480/
->6371" -X8c -O >> Jp.ps
```

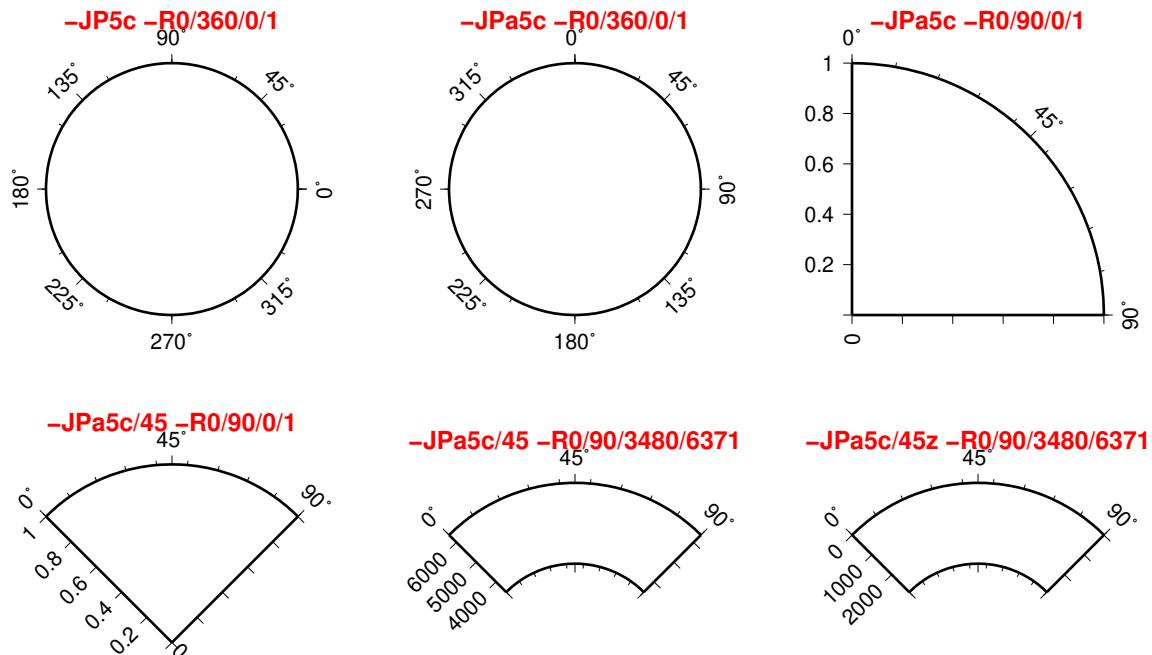


图 9.6: 极坐标用法示例

9.3 -Ja: Lambert 方位等面积投影

维基链接: https://en.wikipedia.org/wiki/Lambert_azimuthal_equal-area_projection

该投影由 Lambert 于 1772 年发展得到, 通常用于绘制大区域地图 (例如整个大洲或半球), 该投影是方位等面积投影。在投影的中心畸变为 0, 离投影中心距离越远, 畸变越大。该投影的参数为:

```
-JA<lon>/<lat>/[<distance>/]<width>
-Ja<lon>/<lat>/[<distance>/]<scale>
```

- <lon>/<lat> 投影中心坐标
- <distance> 投影中心到边界的角度, 默认值为 90, 即距离投影中心各 90 度, 即整个半球
- <width> 地图宽度
- <scale> 地图比例尺 1:xxxx 或 <radius>/<latitude> (<radius> 是纬线 <latitude> 与投影中心在图上的距离)

9.3.1 矩形地图

对于此投影而言, 经线和纬线通常不是直线, 因而不适合用于指定地图边界。因而本例中通过指定区域的左下角 (0° E/ 40° S) 和右上角 (60° E/ 10° S) 的坐标来指定区域范围。

```
gmt set FORMAT_GEO_MAP ddd:mm:ssF MAP_GRID_CROSS_SIZE_PRIMARY 0
gmt pscoast -R0/-40/60/-10r -JA30/-30/4.5i -Bag -Dl -A500 -Gp300/10 \
-Wthinnest -P > GMT_lambert_az_rect.ps
```

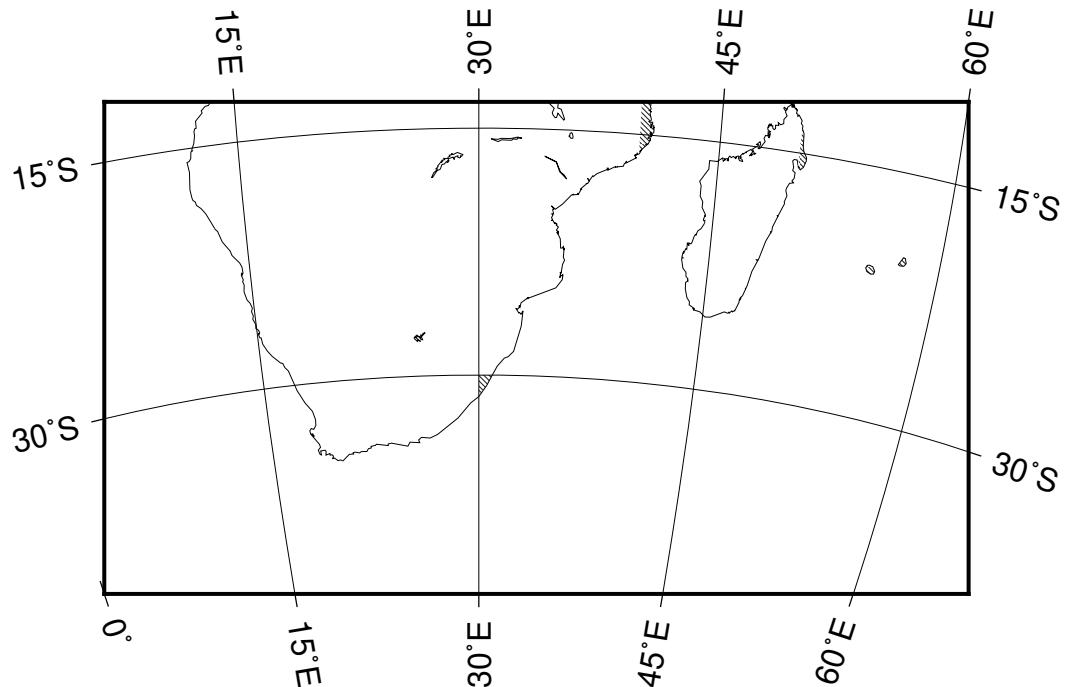


图 9.7：使用 Lambert 方位等面积投影绘制矩形地图

9.3.2 半球地图

要绘制半球地图，需要指定区域范围为整个地球。下图绘制了以南美洲为中心的半球图。

```
gmt pscoast -Rg -Ja280/30/3.5i -Bg -Dc -A1000 -Gnavy -P > GMT_lambert_az_hemi.ps
```

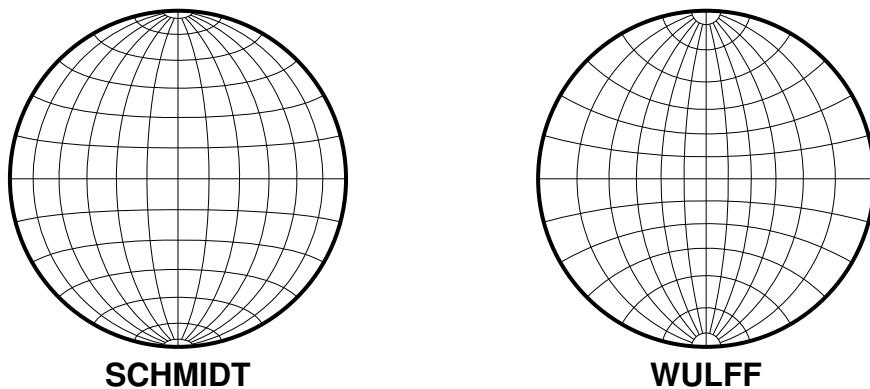


图 9.8：使用 Lambert 方位等面积投影绘制半球地图

9.3.3 震源辐射花样

地震学在绘制震源机制解时，就是将三维的辐射花样信息投影到一个水平面内。投影的方式有两种：Schmidt 网和 Wulff 网。其中 Schmidt 网使用的是 Lambert 方位等面积投影（中心经纬度为 0/0），Wulff 网使用的则是等角度的立体投影。两种震源球投影方式如下图所示：

Source Code



9.4 -Jb: Albers 圆锥等面积投影

维基链接: https://en.wikipedia.org/wiki/Albers_projection

此投影由 Albers 于 1805 年提出，主要用于绘制东西方向范围很大的地图，尤其是美国地图。

此投影是圆锥、等面积投影。纬线是不等间隔分布的同心圆，在地球南北极处分布较密。经线则是等间隔分隔，并垂直切割纬线。

在两条标准纬线处，比例尺和形状的畸变最小；在两者之间，沿着纬线的比例尺太小；在两者外部，沿着经线的比例尺则太大。沿着经线，则完全相反。

该投影方式的参数为：

```
-JB<lon>/<lat>/<lat1>/<lat2>/<width>  
-Jb<lon>/<lat>/<lat1>/<lat2>/<scale>
```

- <lon> 和 <lat> 是投影中心的位置
- <lat1> 和 <lat2> 是两条标准纬线

下图绘制了台湾附近的区域，投影中心位于 125°E/20°N，25 度和 45 度纬线是两条标准纬线。

```
gmt set MAP_GRID_CROSS_SIZE_PRIMARY 0  
gmt pscoast -R110/140/20/35 -JB125/20/25/45/5i -Bag -Dl -Ggreen -Wthinnest \  
-A250 -P > GMT_albers.ps
```

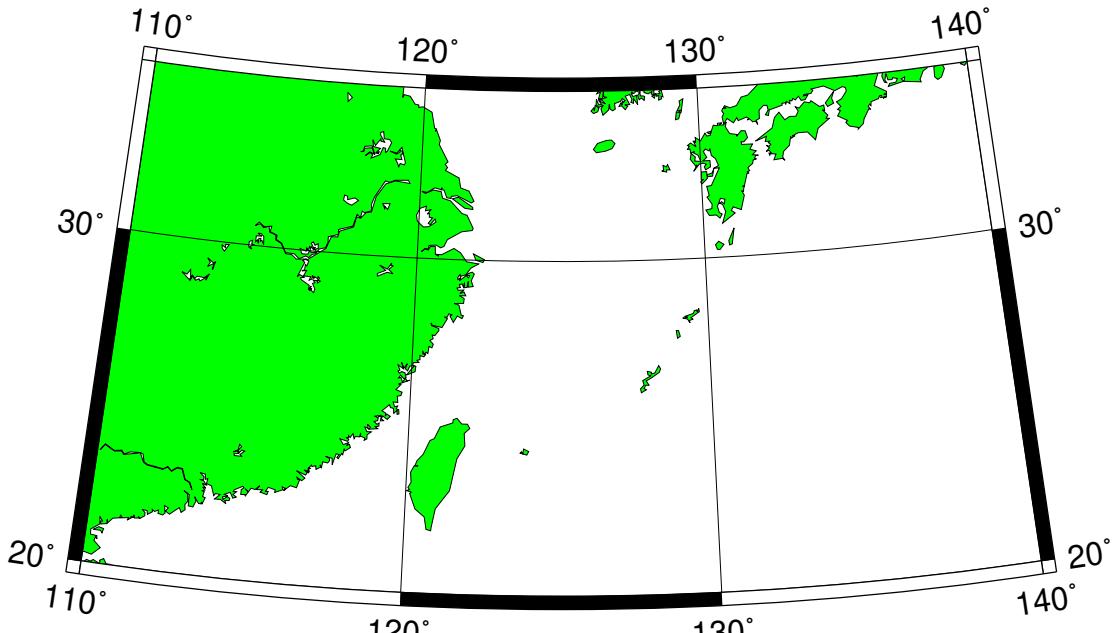


图 9.9: Albers 圆锥等面积投影

9.5 -Jc: Cassini 圆柱投影

维基链接: https://en.wikipedia.org/wiki/Cassini_projection

此圆柱投影由 César-François Cassini de Thury 于 1745 年在调查法国时提出。其偶尔也被称为 Cassini-Soldner 投影, 因为后者提供了更加精确的数学分析得到了椭球下的公式。

此投影既不保角也不等面积, 而是介于二者之间的一种投影方式。沿着中心经线的畸变最小, 适合绘制南北方向区域范围较大的地图。其中, 中心经线、距离中心经线 90 度的两条经线以及赤道是直线, 其余经线和纬线都是复杂的曲线。

该投影方式的参数为:

```
-JC<lon>/<lat>/<width>
-JC<lon>/<lat>/<scale>
```

其中, <lon>/<lat> 为中心的经纬度。

```
gmt pscoast -R7:30/38:30/10:30/41:30r -JC8.75/40/2.5i -Bafg -LjBR+c40+w100+f+o0.15i/0.
→2i \
-Gspringgreen -Dh -Sazure -Wthinnest -Ia/thinner -P --FONT_LABEL=12p > GMT_cassini.ps
```

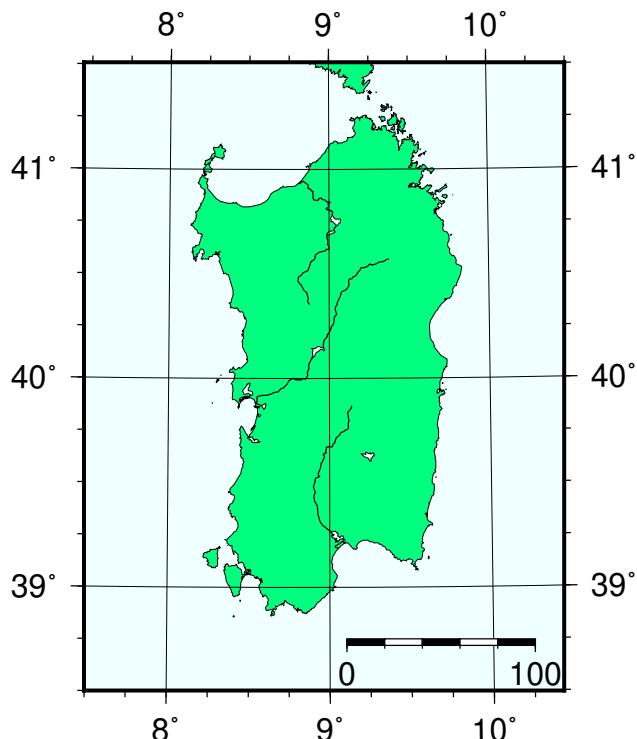


图 9.10: Cassini 投影绘制 Sardinia 岛

9.6 -Jcyl_stere: 圆柱立体投影

维基链接: https://en.wikipedia.org/wiki/Gall_stereographic_projection

圆柱立体投影不像其他的圆柱投影那样显著,但由于其相对简单且能够克服其他圆柱投影的缺点(比如高纬度的畸变),故而仍然被使用。立体投影是透视投影,将整个球沿着赤道上的对跖点投影到一个圆柱上。该圆柱于两条距赤道等间距的标准纬线处穿过球体。

该投影的参数为:

```
-JCyl_stere/[<lon>[/<lat>]]/<width>
-Jcyl_stere/[<lon>[/<lat>]]/<scale>
```

- <lon> 中心经线,若省略则使用区域的中心经线
- <lat> 标准纬线,默认是赤道。若使用,则必须指定中心经线

一些比较流行的标准纬线的选择如下:

Miller' s modified Gall	66.159467°
Kamenetskiy' s First	55°
Gall' s stereographic	45°
Bolshoi Sovietskii Atlas Mira or Kamenetskiy' s Second	30°
Braun' s cylindrical	0°

```
gmt set FORMAT_GEO_MAP dddA
gmt pscoast -R-180/180/-60/80 -JCyl_stere/0/45/4.5i -Bxa60f30g30 -Bya30g30 -Dc -A5000 \
-Wblack -Gseashell4 -Santiquewhite1 -P > GMT_gall_stereo.ps
```

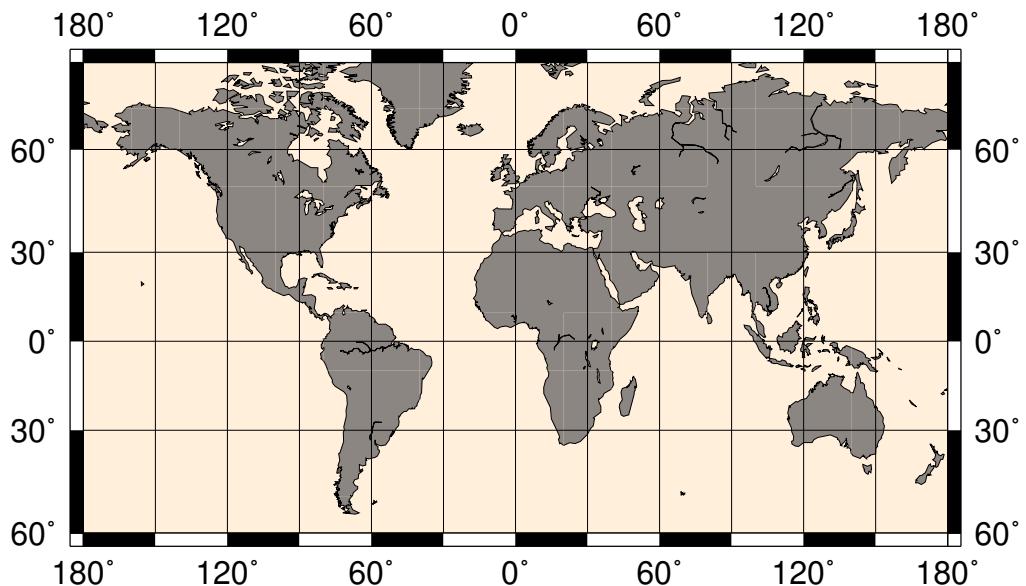


图 9.11: 使用 Gall 立体投影绘制世界地图

9.7 -Jd: 等距圆锥投影

维基链接: https://en.wikipedia.org/wiki/Equidistant_conic_projection

等距圆锥投影由希腊哲学家 Claudius Ptolemy 于公元 150 年提出。其既不是保角也不是等面积，而是两种的折衷。在所有经线以及标准纬线上比例尺没有畸变。

该投影的参数为:

```
-JD<lon>/<lat>/<lat1>/<lat2>/<width>
-Jd<lon>/<lat>/<lat1>/<lat2>/<scale>
```

- <lon>/<lat> 投影中心位置
- <lat1>/<lat2> 两条标准纬线

等距圆锥投影常用于绘制小国家的地图集。

```
gmt set FORMAT_GEO_MAP ddd:mm:ssF MAP_GRID_CROSS_SIZE_PRIMARY 0.05i
gmt pscoast -R-88/-70/18/24 -JD-79/21/19/23/4.5i -Bag -Di -N1/thick,red \
-Glightgreen -Wthinnest -P > GMT_equidistant_conic.ps
```



图 9.12: 等距圆锥地图投影

9.8 -Je: 方位等距投影

维基链接: https://en.wikipedia.org/wiki/Azimuthal_equidistant_projection

方位投影最显著的特征是在图上测量的从中心到任意一点的距离是真实的。因而，地图上以投影中心为圆心的圆在真实地球上与投影中心是等距离的。同时，从中心出发的任意方向也是真实的。该投影常用于展示多个地理位置与指定点的距离图。

该投影的参数为:

```
-JE<lon>/<lat>[/<distance>]/<width>
-Je<lon>/<lat>[/<distance>]/<scale>
```

- <lon>/<lat> 投影中心的经纬度
- <distance> 是边界距离投影中心的度数，默认值为 180，即绘制全球图
- <scale> 可以取 1:xxxx 格式，也可以是 <radius>/<latitude>（表示从投影中心到纬线 <latitude> 在图上的距离为 <radius>）

下图中，投影中心为 100°W/40°N，离投影中心 180 度的点在图中的最外边界处。

```
gmt pscoast -Rg -JE-100/40/4.5i -Bg -Dc -A10000 -Glightgray -Wthinnest > GMT_az_
-equitistant.ps
```



图 9.13: 使用等距方位投影绘制全球图

9.9 -Jf: 球心方位投影

维基链接: https://en.wikipedia.org/wiki/Gnomonic_projection

此投影是一个从中心投影到与表面相切的一个平面的透视投影。此投影既不等面积也不保角, 且在半球的边界处有很大畸变, 但从投影中心出发的方向是真实的。大圆会被投影成直线。

该投影的参数为:

```
-JF<lon>/<lat>[/<distance>]/<width>
-Jf<lon>/<lat>[/<distance>]/<scale>
```

- <lon>/<lat> 投影中心的经纬度
- <distance> 地图边界到投影中心的角度, 默认值为 60 度
- <scale> 可以是 1:xxxx 也可以是 <radius>/<latitude> (<radius> 是投影中心到纬线 <latitude> 在图上的距离)

```
gmt pscoast -Rg -JF-120/35/60/4.5i -B30g15 -Dc -A10000 -Gtan -Scyan -Wthinnest \
-P > GMT_gnomonic.ps
```

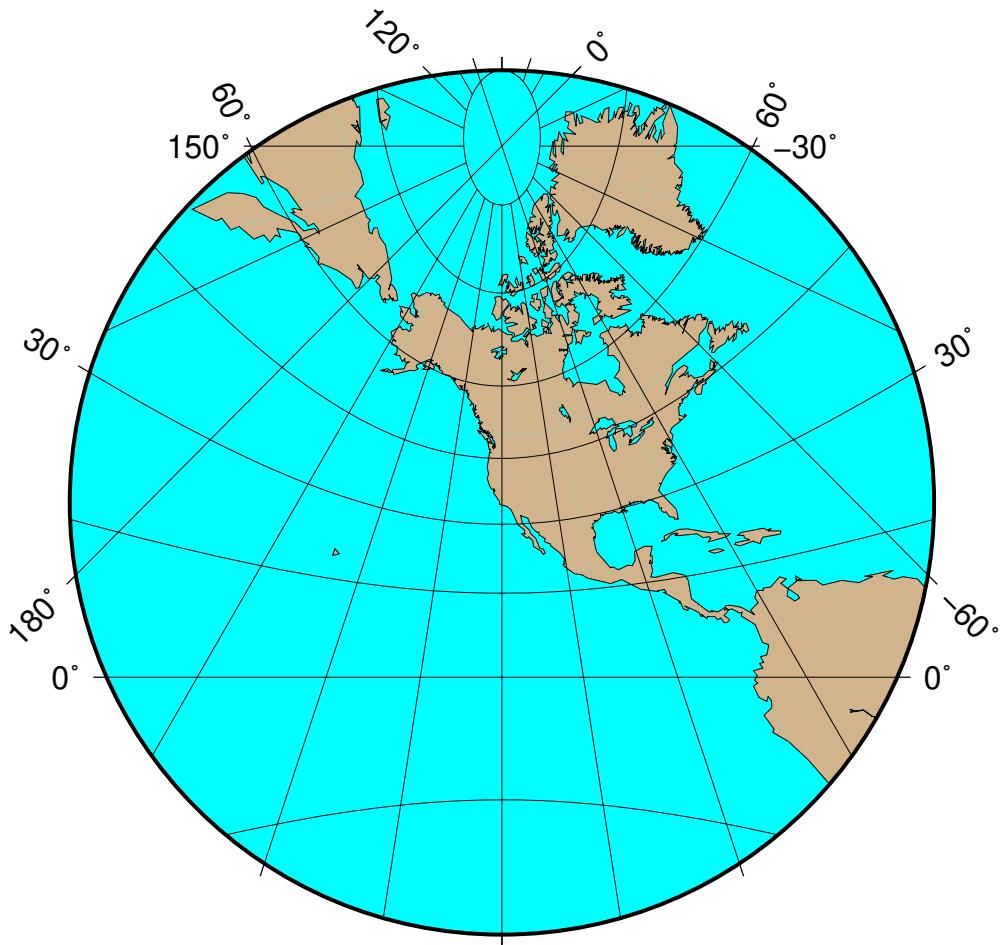


图 9.14: 球心方位投影

9.10 -Jg: 正交投影

正交方位投影是一种从无穷远距离处的透视投影，因而常用于绘制从外太空看地球。与 Lambert 等面积投影以及立体投影类似，一次只能看到一个半球。该投影既不是等面积也不是保角，在半球边界处有较大的畸变。从投影中心出发的任意方向是真实的。

该投影的参数为：

```
-JG<lon>/<lat>[/<distance>]/<width>
-Jg<lon>/<lat>[/<distance>]/<scale>
```

- <lon>/<lat> 是投影中心位置
- <distance> 是边界离投影中心的度数，默认值为 90
- <scale> 地图比例尺 1:xxxx 或 <radius>/<latitude> (<radius> 是纬线 <latitude> 与投影中心在图上的距离)

```
gmt pscoast -Rg -JG-75/41/4.5i -Bg -Dc -A5000 -Gpink -Sthistle -P > GMT_orthographic.ps
```



图 9.15: 使用正交投影绘制半球

`-Jg` 加上更多的参数时还可以用于绘制透视投影，以在二维平面内模拟从太空看三维的地球。具体的参数为：

```
-JG<lon>/<lat>/<alt>/<az>/<tilt>/<twist>/<width>/<height>
```

- `<lon>/<lat>` 投影中心的经纬度
- `<alt>` 是观察者所处的海拔，单位为 km。若该值小于 10，则假定是观察者相对于地心的距离，若距离后加了 `r`，则表示观察者与地心的距离（单位为 km）。
- `<az>` 观察者的方位角，默认值为 90 度，即从东向观测
- `<tilt>` 倾角（单位为度），默认值为 60 度。若值为 0 则表示在顶点直接向下看，值为 60 则表示在顶点处沿着水平方向 30 度角的方向观察
- `<twist>` 扭转角度，默认值为 180 度。This is the boresight rotation (clockwise) of the image. The twist of 180° in the example mimics the fact that the Space Shuttle flies upside down.
- `<width>/<height>` 是视角的角度，单位为度，默认值为 60。This number depends on whether you are looking with the naked eye (in which case you view is about 60° wide), or with binoculars, for example.
- `<scale>` as 1:xxxxx or as radius/latitude where radius is distance on map in inches from projection center to a particular oblique latitude

```
gmt pscoast -Rg -JG4/52/230/90/60/180/60/60/5i -Bx2g2 -By1g1 -Ia -Di -Glightbrown \
-Wthinnest -P -Slightblue --MAP_ANNOT_MIN_SPACING=0.25i > GMT_perspective.ps
```

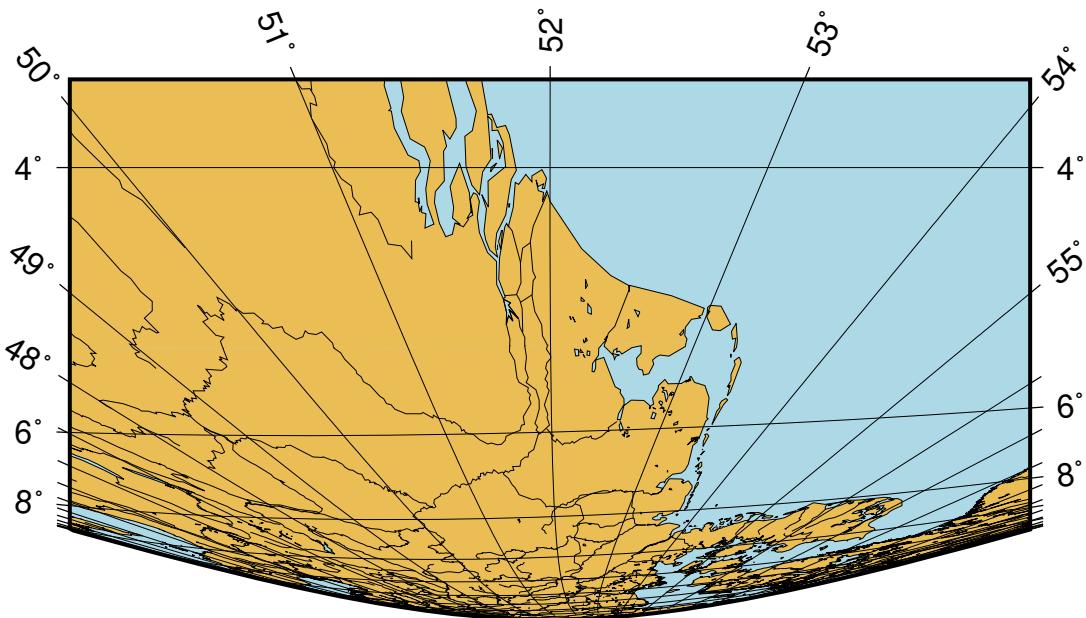


图 9.16: 透视投影

9.11 -Jh: 等面积 Hammer 投影

维基链接: https://en.wikipedia.org/wiki/Hammer_projection

等面积 Hammer 投影由 Ernst von Hammer 于 1892 年提出, 也被称为 Hammer-Aitoff 投影 (Aitoff 投影与之看起来相似, 但不等面积)。投影后的边界是一个椭圆, 赤道和中心经线是直线, 其余纬线和经线都是复杂曲线。

该投影的参数为:

```
-JH[<lon>/]<width>
-Jh[<lon>/]<scale>
```

<lon> 是中心经线, 默认位于地图区域的中心。

```
gmt pscoast -Rg -JH4.5i -Bg -Dc -A10000 -Gblack -Scornsilk -P > GMT_hammer.ps
```



图 9.17: 使用 Hammer 投影绘制全球地图

9.12 -Ji: 正弦曲线投影

维基链接: https://en.wikipedia.org/wiki/Sinusoidal_projection

正弦曲线投影是等面积投影，是已知的最古老的投影之一，也被称为等面积 Mercator 投影。其中心经线是直线，其余经线是正弦曲线，纬线是等间距的直线。在所有纬线和中心经线处比例尺是真实的。

该投影的参数为:

```
-JI[<lon>/]<width>
-Ji[<lon>/]<scale>
```

<lon> 是中心经线，默认值为地图区域的中心。

```
gmt pscoast -Rd -JI4.5i -Bxg30 -Byg15 -Dc -A10000 -Ggray -P > GMT_sinusoidal.ps
```

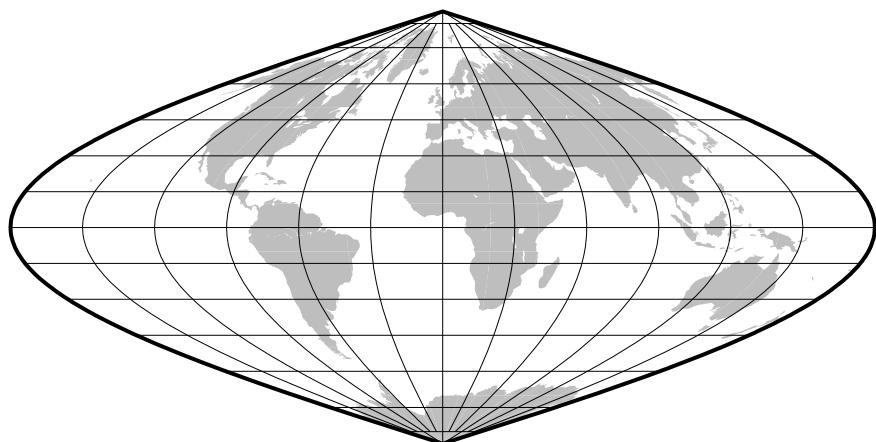


图 9.18: 使用正弦曲线投影绘制世界地图

为了减少形状的畸变，1927 年引入了间断正弦曲线投影，即用三个对称的段来覆盖全

球。传统上，间断出现在 160°W 、 20°W 和 60°E 处。为了生成间断地图，必须调用 `pscoast` 三次以分别绘制每段地图并叠加起来。间断正弦曲线投影一般仅用于显示全球不连续数据分布。

为了生成一个宽度为 5.04 英寸的间断世界地图，需要设置比例尺为 $5.04/360 = 0.014$ ，并将每段图沿水平方向偏移其对应的宽度 (140·0.014 and 80·0.014)。

```
gmt pscoast -R200/340/-90/90 -Ji0.014i -Bxg30 -Byg15 -A10000 -Dc -Gblack -K -P > GMT_
˓→sinus_int.ps
gmt pscoast -R-20/60/-90/90 -Ji0.014i -Bxg30 -Byg15 -Dc -A10000 -Gblack -X1.96i -O -K >>
˓→ GMT_sinus_int.ps
gmt pscoast -R60/200/-90/90 -Ji0.014i -Bxg30 -Byg15 -Dc -A10000 -Gblack -X1.12i -O >> ④
˓→GMT_sinus_int.ps
```

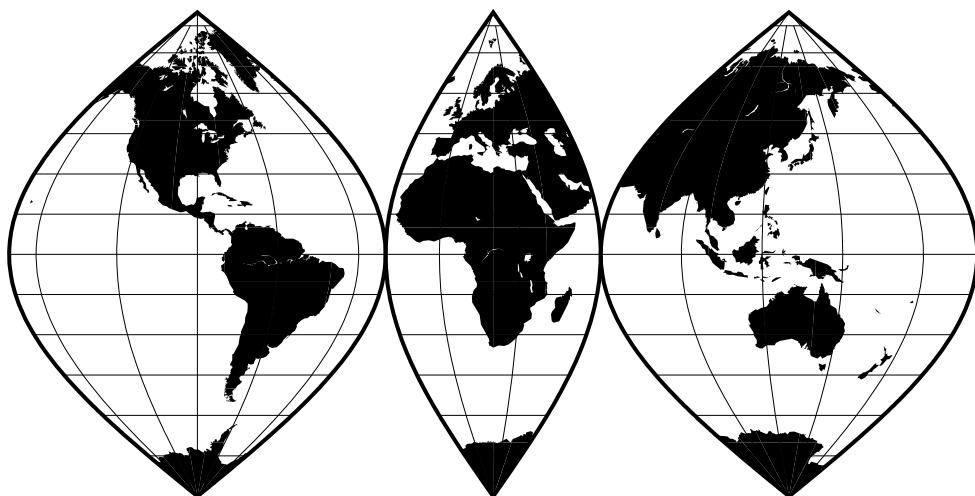


图 9.19：使用间断正弦曲线投影绘制世界地图

9.13 -Jj: Miller 圆柱投影

维基链接: https://en.wikipedia.org/wiki/Miller_cylindrical_projection

此投影由 Osborn Maitland Miller 于 1942 年提出，该投影既不是保角也不是等面积。所有的经线和纬线都是直线。该投影是 Mercator 与其他圆柱投影之间的折衷。在此投影中，纬线之间的间距使用了 Mercator 公式并乘以 0.8 倍的真实纬度，因而避免了极点的奇点，然后再将结果除以 0.8。

该投影的参数为：

```
-JJ<lon>/<width>
-Jj<lon>/<scale>
```

<lon> 为中心经度，默认为地图区域的中心。

```
gmt pscoast -R-90/270/-80/90 -Jj1:400000000 -Bx45g45 -By30g30 -Dc -A10000 \
-Gkhaki -Wthinnest -P -Sazure > GMT_miller.ps
```



图 9.20：使用 Miller 圆柱投影绘制世界地图

9.14 -Jk: Eckert 投影

维基链接：

- https://en.wikipedia.org/wiki/Eckert_IV_projection
- https://en.wikipedia.org/wiki/Eckert_VI_projection

Eckert IV 和 VI 投影由 Max Eckert-Greifendorff 于 1906 年提出，是伪圆柱等面积投影。中心经线以及所有的纬线都是执行，其余经线是等间隔分布的椭圆弧 (IV) 或正弦曲线 (VI)。比例尺在纬线 $40^{\circ}30'$ (IV) 和 $49^{\circ}16'$ (VI) 是真实的。 $-JKf$ (f 代表 four) 表示使用 Eckert IV 投影， $-JKs$ (s 代表 six) 表示使用 Eckert VI 投影。若不指定 f 或 s ，则默认使用 Eckert VI 投影。

该选项的参数为：

```
-JK[f|s][<lon>/]<width>
-Jk[f|s][<lon>/]<scale>
```

$<\text{lon}>$ 为中心经线，默认值为地图区域的中心。

Eckert IV 示例：

```
gmt pscoast -Rg -JKf4.5i -Bg -Dc -A10000 -Wthinnest -Givory -Sbisque3 -P > GMT_eckert4.
↪ps
```



图 9.21: Eckert IV 投影绘制全球图

Eckert VI 示例:

```
gmt pscoast -Rg -JKs4.5i -Bg -Dc -A10000 -Wthinnest -Givory -Sbisque3 -P > GMT_eckert4.  
→ps
```

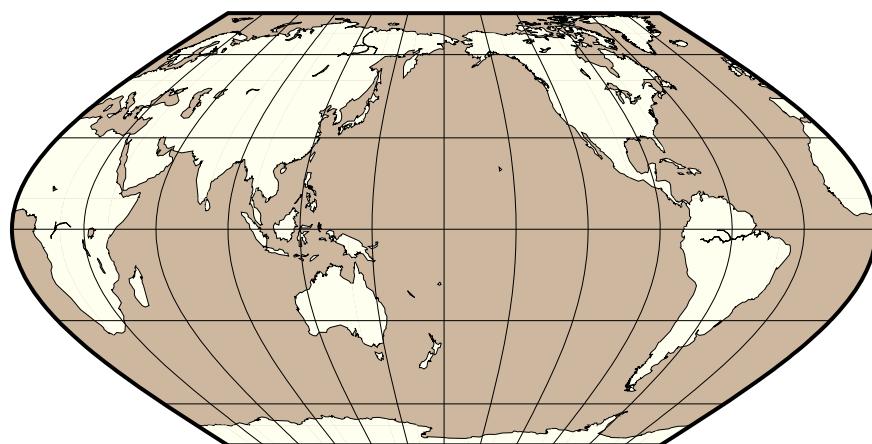


图 9.22: Eckert VI 投影绘制全球图

9.15 -Jl: Lambert 圆锥保角投影

维基链接: https://en.wikipedia.org/wiki/Lambert_conformal_conic_projection

此投影由 Heinrich Lambert 于 1772 年提出, 主要用于绘制东西方向范围很大的地图。与 Albers 投影不同的是, Lambert 投影不是等面积的。纬线是共圆心的圆弧, 经线是这些圆的等间隔分布的半径。与 Albers 投影类似, 只有两条标准纬线是无畸变的。

该投影的参数为:

```
-JB<lon>/<lat>/<lat1>/<lat2>/<width>  
-Jb<lon>/<lat>/<lat1>/<lat2>/<scale>
```

- <lon> 和 <lat> 是投影中心的位置

- <lat1> 和 <lat2> 是两条标准纬线

Lambert 保角投影场用于绘制美国地图，两个固定的标准纬线是 33°N 和 45°N。

```
gmt set MAP_FRAME_TYPE FANCY FORMAT_GEO_MAP ddd:mm:ssF MAP_GRID_CROSS_SIZE_PRIMARY 0.05i
gmt pscoast -R-130/-70/24/52 -Jl-100/35/33/45/1:50000000 -Bag -Dl -N1/thick,red \
-N2/thinner -A500 -Gtan -Wthinnest,white -Sblue -P > GMT_lambert_conic.ps
```

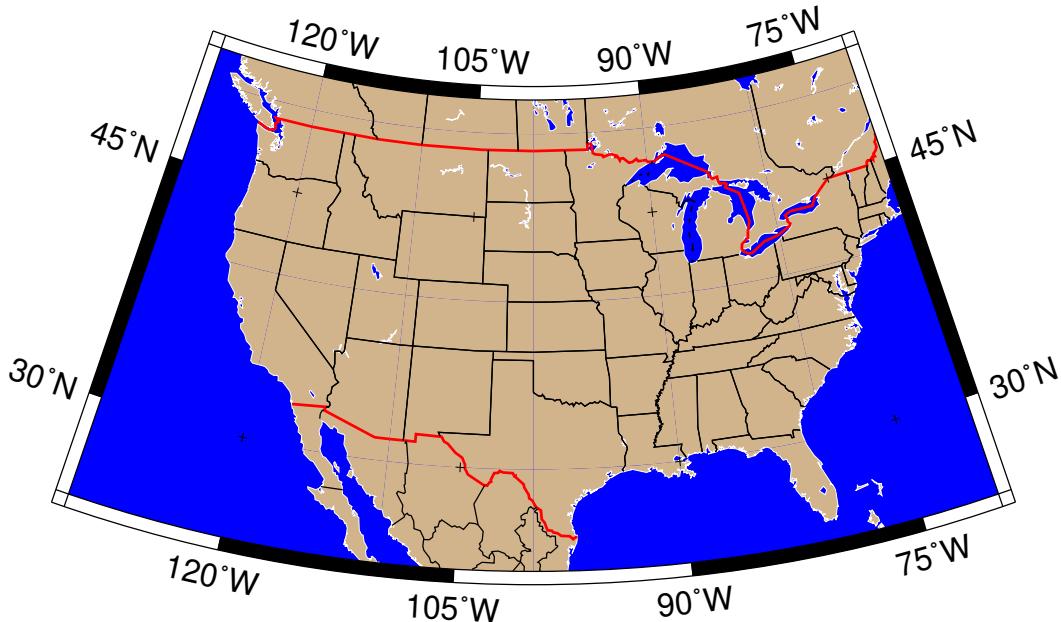


图 9.23: Lambert 保角圆锥投影

投影中心的选取并不影响投影，但其指定了哪一条经线垂直于地图。

9.16 -Jm: Mercator 投影

维基链接: https://en.wikipedia.org/wiki/Mercator_projection

此投影是圆柱保角投影，沿着赤道无畸变，但两极畸变严重。此投影的主要特点是等方位角的线是一条直线，这样一条线称为 rhumb 线或 loxodrome。

在常规 Mercator 投影中，圆柱与赤道相切。若圆柱沿着其他方向与地球相切，则称为横向 Mercator 投影或倾斜 Mercator 投影。

常规的 Mercator 投影需要的参数如下:

```
-JM[<lon>[/<lat>]]<width>
-Jm[<lon>[/<lat>]]<scale>
```

- <lon> 中心经线，默认为地图区域的中心
- <lat> 标准纬线，默认值为赤道。若要指定标准纬线，则必须同时指定中心经线

```
gmt set MAP_FRAME_TYPE fancy  
gmt pscoast -R0/360/-70/70 -Jm1.2e-2i -Bxa60f15 -Bya30f15 -Dc -A5000 -Gred \  
-P > GMT_mercator.ps
```

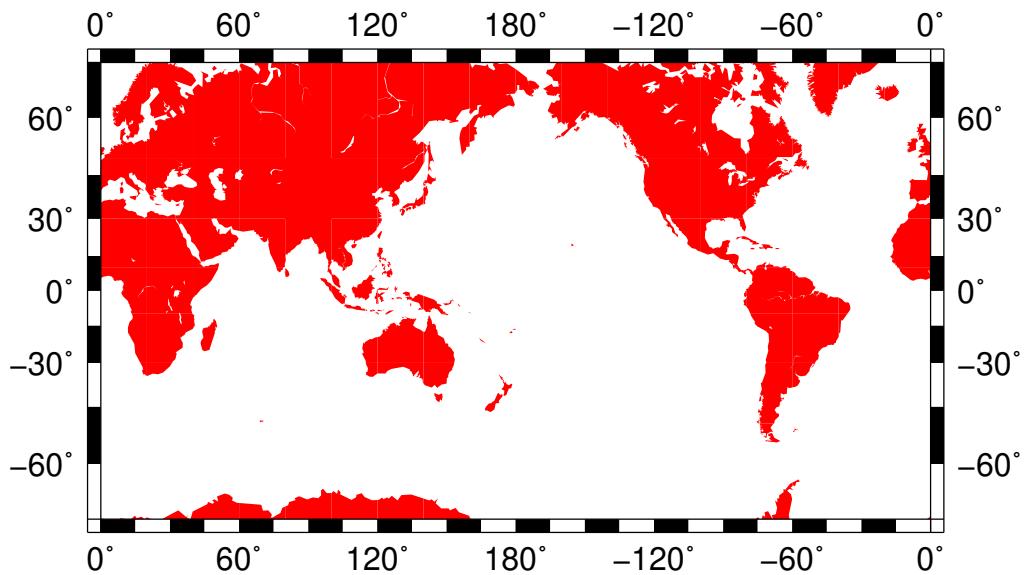


图 9.24: Mercator 投影

9.17 -Jn: Robinson 投影

维基链接: https://en.wikipedia.org/wiki/Robinson_projection

此投影 Arthur H. Robinson 于 1963 年提出，是一个修改后的圆柱投影，既不是保角也不是等面积。中心经线以及所有纬线都是直线，其余经线都是曲线。其使用查找表的方式而不是解析表达式来使得全球看上去比较正常。比例尺在经线 38 度是真实的。

该投影的参数为:

```
-JN[<lon>/]<width>      -Jn[<lon>/]<scale>
```

<lon> 是中心经线，默认值为地图区域的中心。

```
gmt pscoast -Rd -JN4.5i -Bg -Dc -A10000 -Ggoldenrod -Ssnow2 -P > GMT_robinson.ps
```

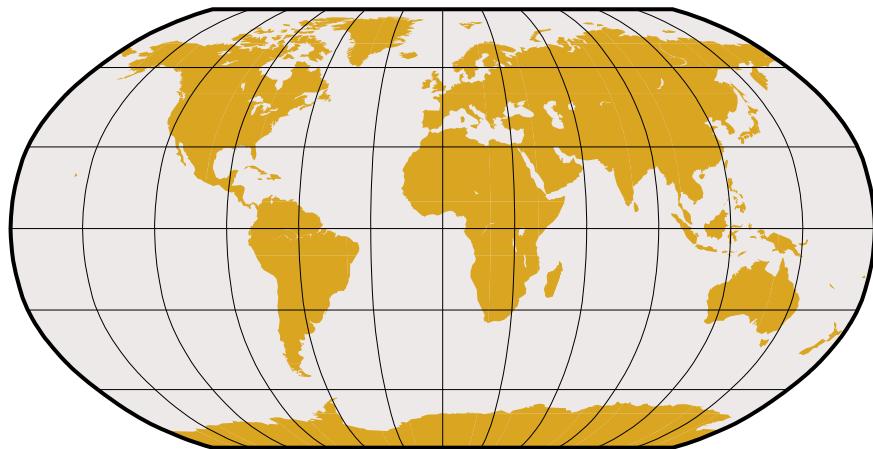


图 9.25: 使用 Robinson 投影绘制全球地图

9.18 -Jo: 倾斜 Mercator 投影

维基链接: https://en.wikipedia.org/wiki/Space-oblique_Mercator_projection

倾斜 Mercator 投影常用于绘制沿着倾斜方向横向范围较大的地图，其经线和纬线都是复杂曲线。

其有多种定义方式:

```
-J0[a|A]<lon>/<lat>/<azi>/<width>
-Jo[a|A]<lon>/<lat>/<azi>/<scale>

-J0[b|B]<lon>/<lat>/<lon2>/<lat2>/<width>
-Jo[b|B]<lon>/<lat>/<lon2>/<lat2>/<scale>

-J0[c|C]<lon>/<lat>/<lonp>/<latp>/<width>
-Jo[c|C]<lon>/<lat>/<lonp>/<latp>/<scale>
```

- <lon>/<lat> 投影中心的经纬度
- <azi> 倾斜赤道的方位角
- <lon2>/<lat2> 倾斜赤道另一个点的经纬度
- <lonp>/<latp> 投影极点的经纬度

在三种定义中, 大写的 A|B|C 表示允许投影极点位于南半球。

```
gmt pscoast -R270/20/305/25r -J0c280/25.5/22/69/4.8i -Bag -Di -A250 -Gburlywood \
-Wthinnest -P -TdjTR+w0.4i+f2+l+o0.15i -Sazure --FONT_TITLE=8p \
--MAP_TITLE_OFFSET=@0.05i > GMT_obl_merc.ps
```

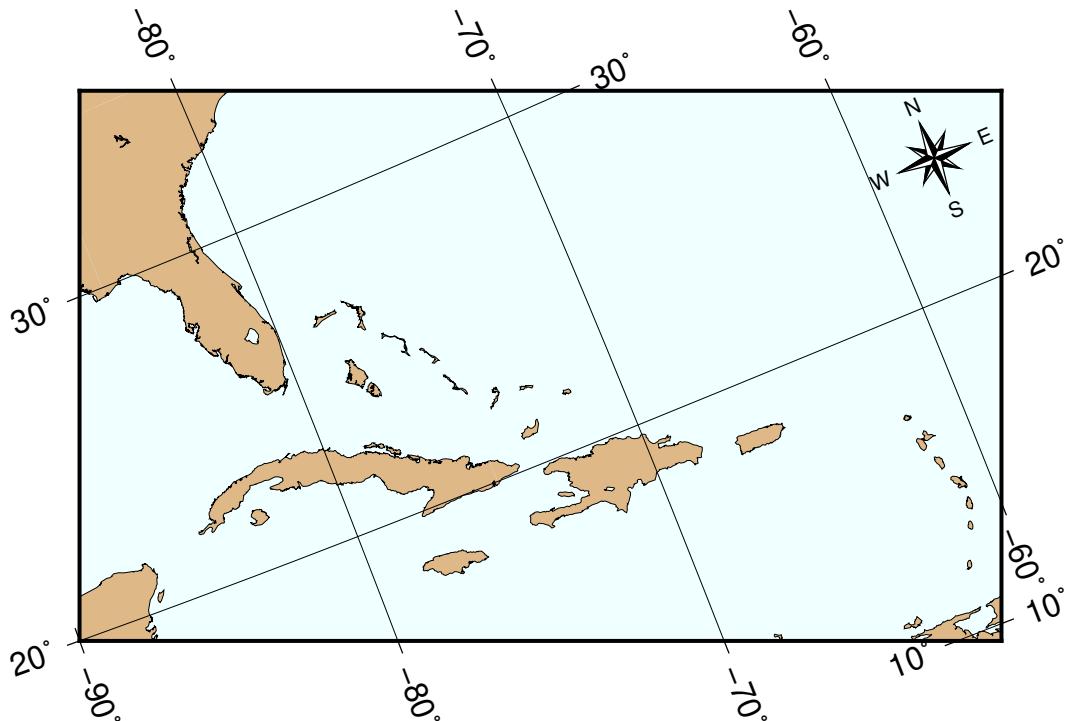


图 9.26: 使用 -Joc 倾斜 Mercator 投影

在使用倾斜投影时，直接指定整个区域相对地图中心的相对投影坐标更为方便，下面的示例中使用了 -Rk-1000/1000/-500/500 来指定相对投影坐标。

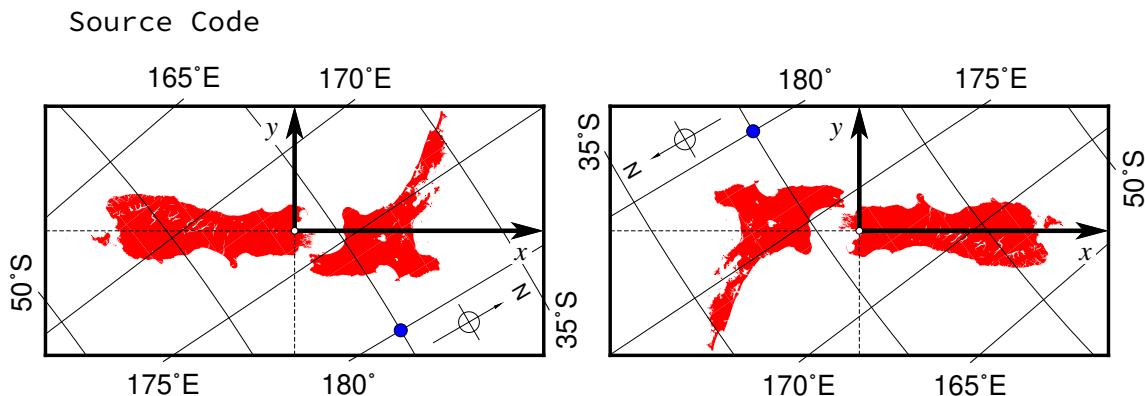


图 9.27: 使用 -JOa 倾斜 Mercator 投影

(左) `-JOa173:17:02E/41:16:15S/35/3i` (右) `-JOA173:17:02E/41:16:15S/215/3i`

9.19 -Jpoly: 多圆锥投影

维基链接: https://en.wikipedia.org/wiki/Polyconic_projection

此投影既不是等面积也不是保角投影，沿着中心经线处畸变为 0。所有纬线的比例尺都是真实的，但其余经线则存在畸变。

```
gmt pscoast -R-180/-20/0/90 -JPoly/4i -Bx30g10 -By10g10 -Dc -A1000 -Glightgray \
-Wthinnest -P > GMT_polyconic.ps
```

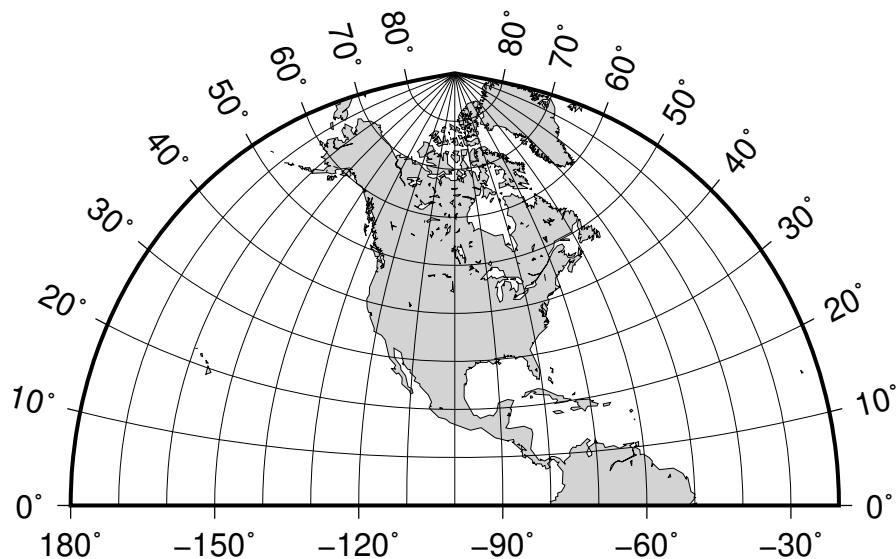


图 9.28: 多圆锥投影

9.20 -Jq: 圆柱等距投影

维基链接: https://en.wikipedia.org/wiki/Equirectangular_projection

这个简单的圆柱投影是一个经度和纬度的线性缩放。最常用的形式是 Plate Carrée 投影, 其中对经线和纬线的缩放比例是相同的。所有的经纬线都是直线。

该投影的参数为:

```
-JQ[<lon>/[<lat>]/]<width>
-Jq[<lon>/[<lat>]/]<scale>
```

- **<lon>** 是中心经线, 默认认为地图区域的中心
- **<lat>** 是标准纬线, 默认认为赤道, 若指定了标准纬线, 则必须指定中心经线

```
gmt pscoast -Rg -JQ4.5i -B60f30g30 -Dc -A5000 -Gtan4 -Slightcyan -P > GMT_equi_cyl.ps
```

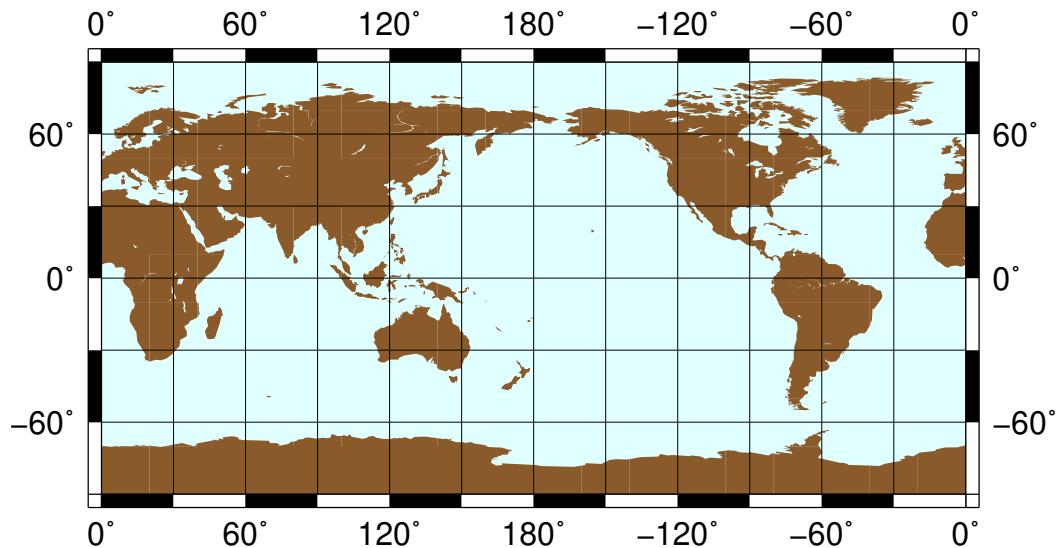


图 9.29: 使用 Plate Carrée 投影绘制全球地图

选择不同的标准纬线，则可以获取经度和纬度的不同缩放比例。流行的几个标准纬线如下：

Grafarend and Niermann, minimum linear distortion	61.7°
Ronald Miller Equirectangular	50.5°
Ronald Miller, minimum continental distortion	43.5°
Grafarend and Niermann	42°
Ronald Miller, minimum overall distortion	37.5°
Plate Carrée, Simple Cylindrical, Plain/Plane	0°

9.21 -Jr: Winkel Tripel 投影

维基链接: https://en.wikipedia.org/wiki/Winkel_tripel_projection

1921 年 Oswald Winkel 设计了该投影，以在三个元素（面积、角度、距离）之间折衷，在绘制全球地图时，这三个元素的畸变最小。此投影不是保角也不是等面积投影。中心经线和赤道是直线，其他经线和纬线是曲线。该投影取等距圆柱投影和 Aitoff 投影的坐标的平均值。极点处投影为 0.4 倍赤道长度的直线。

该投影的参数为：

```
-JR[<lon>/]<width>
-Jr[<lon>/]<scale>
```

<lon> 是中心经线，默认值为地图区域的中心。

```
gmt pscoast -Rd -JR4.5i -Bg -Dc -A10000 -Gburlywood4 -Swheat1 -P > GMT_winkel.ps
```

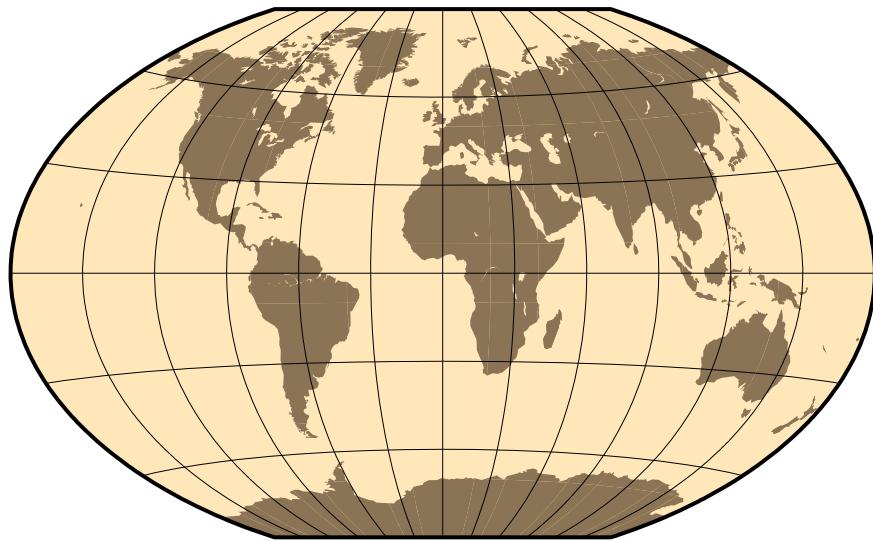


图 9.30: 使用 Winkel Tripel 投影绘制全球地图

9.22 -Js: 立体等角投影

维基链接: https://en.wikipedia.org/wiki/Stereographic_projection

此投影是保角方位投影, 主要用于绘制南北极区域。在两极, 所有经线都是直线, 纬线则是圆弧。

该投影的参数:

```
-JS<lon>/<lat>[<distance>]/<width>
-Js<lon>/<lat>[<distance>]/<scale>
```

- <lon>/<lat> 投影中心的经纬度
- <distance> 地图边界到投影中心的角度, 默认值为 90 度
- <scale> 可以是 1:xxxx 也可以是 <radius>/<latitude> (<radius> 是投影中心到纬线 <latitude> 在图上的距离), 还可以是 <slat>/1:xxxx (指定在标准纬线 <slat> 处的比例尺)

9.22.1 极区立体地图

下面的示例中, 投影中心为北极, 地图边界与经线和纬线完全重合。

```
gmt pscoast -R-30/30/60/72 -Js0/90/4.5i/60 -B10g -Dl -A250 -Groalblue \
-Sseashell -P > GMT_stereographic_polar.ps
```

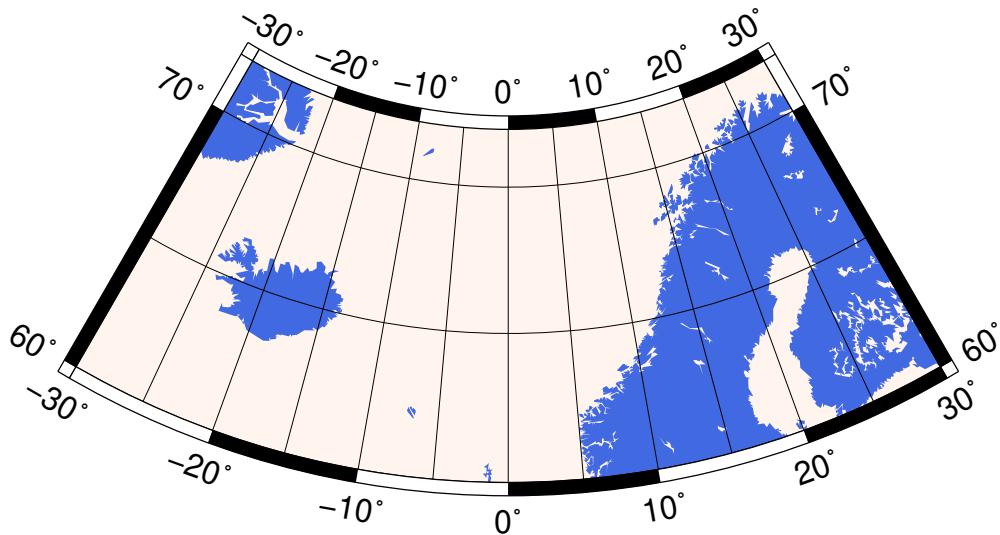


图 9.31: 极区立体保角投影

9.22.2 矩形立体地图

与 Lambert 方位等面积投影类似, 也可以通过指定地图区域左下角和右上角的坐标来绘制一个矩形区域。

```
gmt set MAP_ANNOT_OBLIQUE 30  
gmt pscoast -R-25/59/70/72r -JS10/90/11c -B20g -Dl -A250 -Gdarkbrown -Wthinnest \  
-Slightgray -P > GMT_stereographic_rect.ps
```

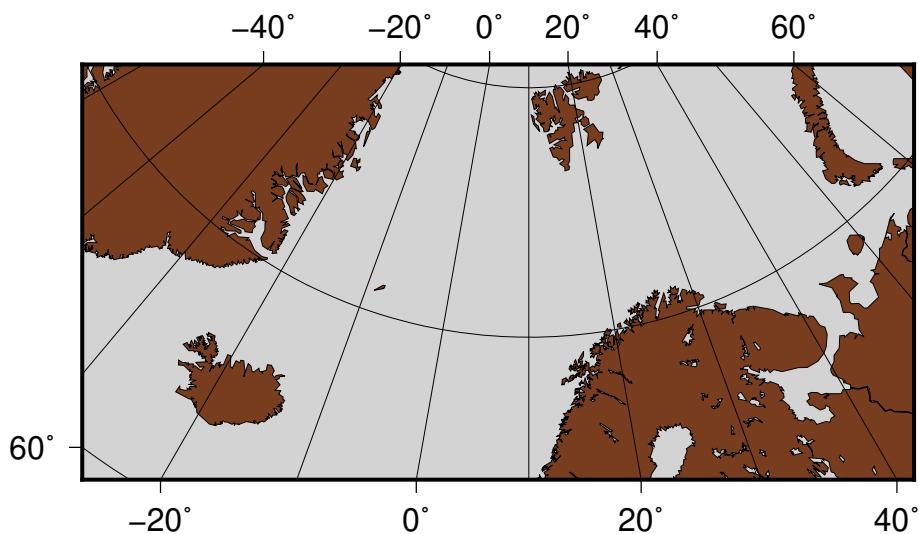


图 9.32: 矩形边界下的极区立体保角投影

9.22.3 一般立体地图

```
gmt set MAP_ANNOT_OBLIQUE 0  
gmt pscoast -R100/-42/160/-8r -JS130/-30/4i -Bag -Dl -A500 -Ggreen -Slightblue \  
-Wthinnest -P > GMT_stereographic_general.ps
```

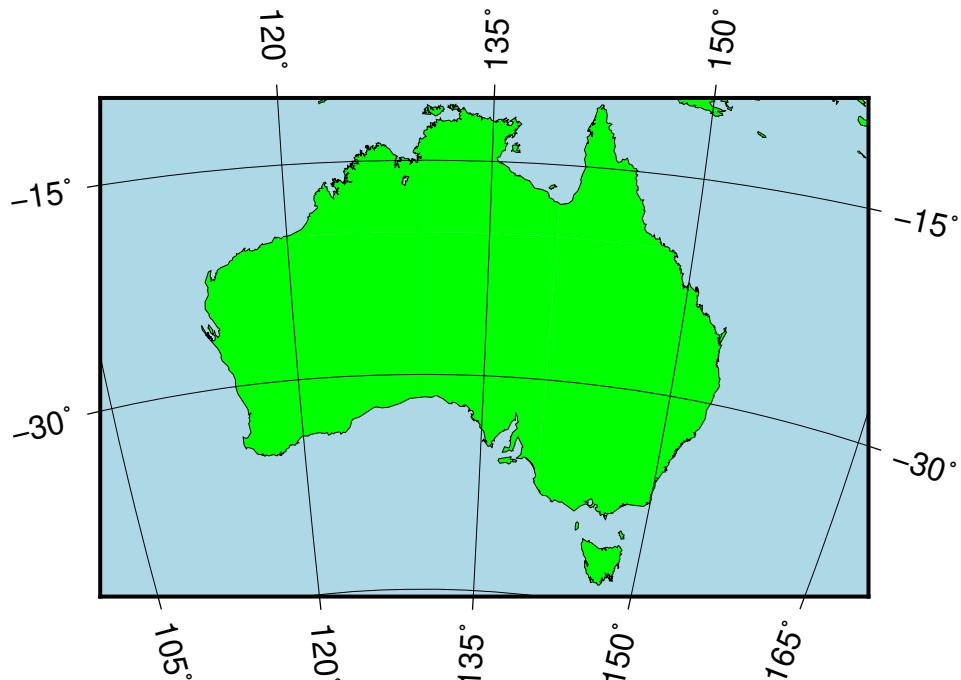


图 9.33: 一般立体投影

9.23 -Jt: 横向 Mercator 投影

维基链接: https://en.wikipedia.org/wiki/Transverse_Mercator_projection

此投影由 Lambert 于 1772 年提出。该投影中，圆柱与某条经线相切。在该经线处无畸变。离中心经线越远畸变越大，距离中心经线 90 度处的经线畸变达到无穷。中心经线和赤道都是直线，其余经线和纬线则是复杂曲线。

该投影的参数:

```
-JT<lon> [<lat>] /<width>
-Jt<lon> [<lat>] /<scale>
```

<lon> 中心经线, <lat> 原点的纬度, 默认值为赤道。

地图缩放因子默认值为 1, 可以通过修改参数 *PROJ_SCALE_FACTOR* 以实现自定义。

```
gmt pscoast -R20/30/50/45r -Jt35/0.18i -Bag -Dl -A250 -Glightbrown -Wthinnest \
-P -Sseashell > GMT_transverse_merc.ps
```

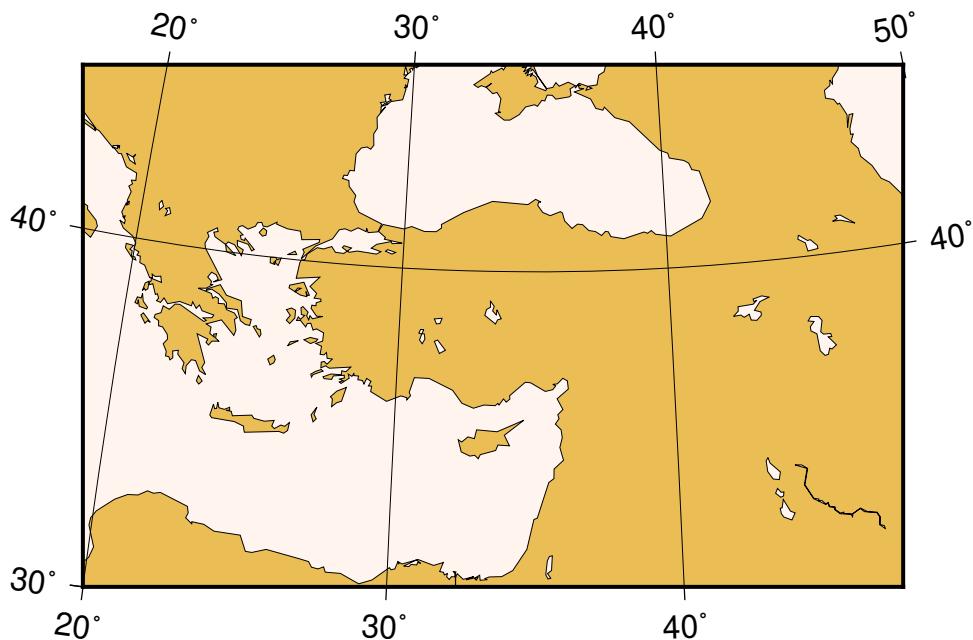


图 9.34: 矩形横向 Mercator 地图

```
gmt pscoast -R0/360/-80/80 -JT330/-45/3.5i -Ba30g -BWSne -Dc -A2000 \
-Slightblue -G0 -P > GMT_TM.ps
```



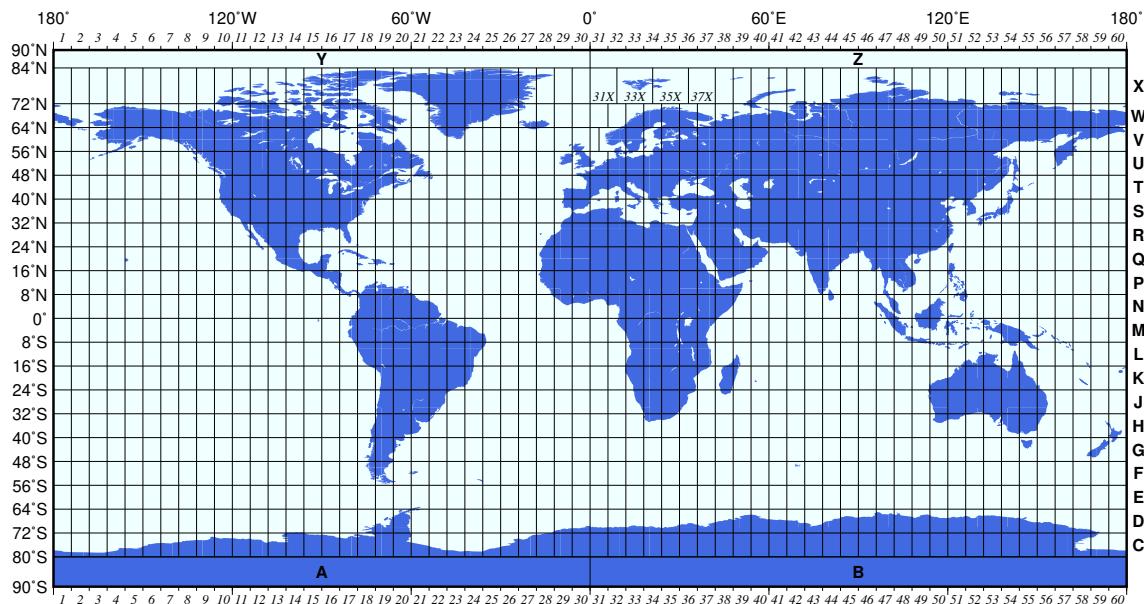
图 9.35: 全球横向 Mercator 地图

9.24 -Ju: 通用横向 Mercator(UTM) 投影

维基链接: https://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system

通用横向 Mercator(UTM) 投影是横向 Mercator 投影的一个特殊子集。此处, 全球在南北纬 84 度之间被划分为 60 个区域, 大多数区域的宽度都是 6 度。每一个区域都有各自位移的中心经线。进一步, 每个区域都被划分为纬度带。

[Source Code](#)



该投影的参数为:

```
-JU<zone>/<width>  
-Ju<zone>/<scale>
```

其中 <zone> 可以取 1–60、A、B、Y、Z，负值表示南半球的区域，也可以加上 C–H 以及 J–N 来指定纬度带。

为了让任意指定区域的畸变最小化，公式中乘以了比例因子 0.9996，这个值可以通过修改 *PROJ_SCALE_FACTOR* 以自定义。这是的 UTM 投影是割线投影而不是切线投影，在赤道处比例尺的畸变只有千分之一。在中心经线附近 10 度范围内的椭球投影表达式都是精确的。对于更大的区域，则在一般球状公式中使用保角纬度作为代替。

9.25 -Jv: Van der Grinten 投影

维基链接: https://en.wikipedia.org/wiki/Van_der_Grinten_projection

此投影由 Alphons J. van der Grinten 于 1904 年提出，其既不等面积也不保角。中心经线和赤道都是直线，其余经线则是圆弧，仅在赤道处比例尺是真实的，主要用于在一个圆内展示整个世界地图。

该投影的参数为:

```
-JV<lon>/<width>  
-Jv<lon>/<scale>
```

<lon> 是投影中心经线，默认值为地图区域的中心。

```
gmt pscoast -Rg -JV4i -Bxg30 -Byg15 -Dc -Glightgray -A10000 -Wthinnest -P > GMT_grinten.  
→ps
```

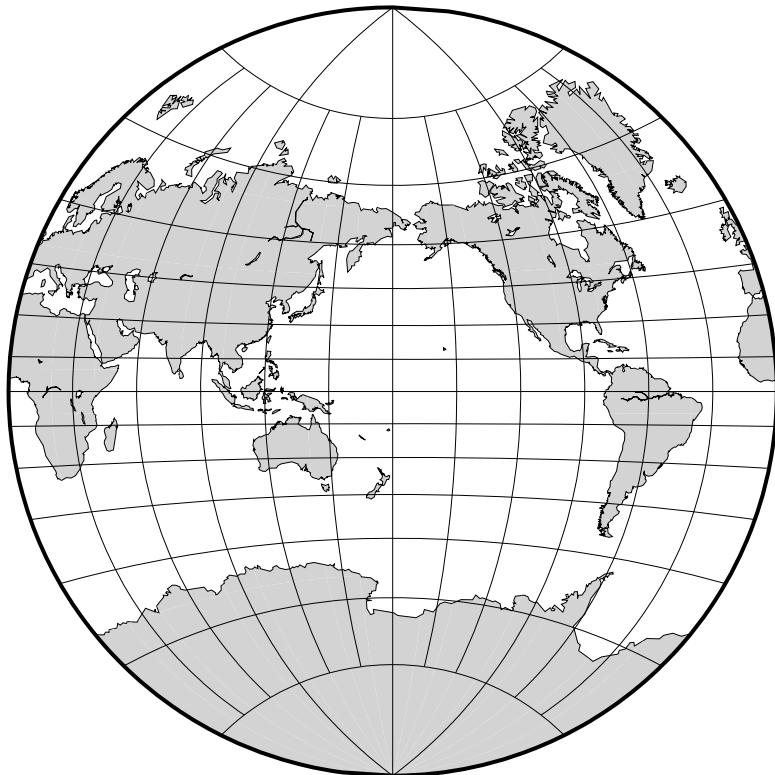


图 9.36: 使用 Van der Grinten 投影绘制全球图

9.26 -Jw: Mollweide 投影

维基链接: https://en.wikipedia.org/wiki/Mollweide_projection

此投影是伪圆柱等面积投影, 由 Karl Brandan Mollweide 于 1805 年提出。纬线是不等间隔分布的直线, 经线是等间隔分布的椭圆弧。比例尺仅在南北纬 40 度 44 分纬线上才是真实的。此投影主要用于绘制全球的数据分布图。

该投影的参数为:

```
-JW[<lon>/]<width>
-Jw[<lon>/]<scale>
```

<lon> 为中心经线, 默认值为地图区域的中心。

```
gmt pscoast -Rd -JW4.5i -Bg -Dc -A10000 -Gtomato1 -Sskyblue -P > GMT_mollweide.ps
```

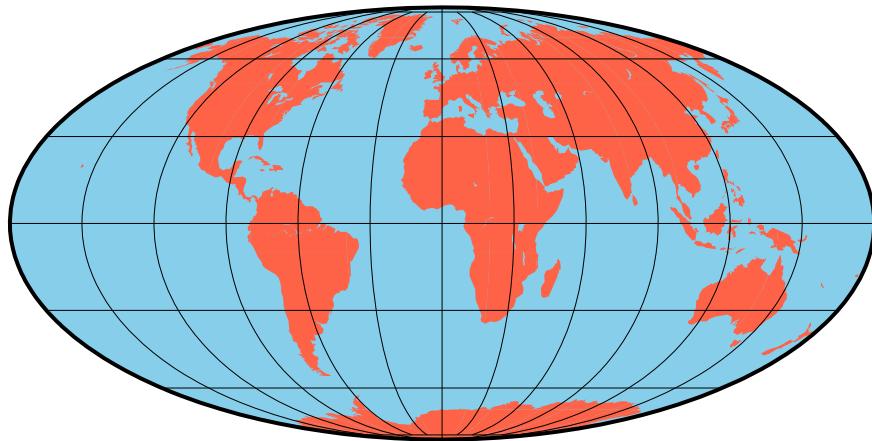


图 9.37: 使用 Mollweide 投影绘制全球地图

9.27 -Jy: 圆柱等面积投影

维基链接: https://en.wikipedia.org/wiki/Cylindrical_equal-area_projection

选择不同的标准纬线，则对应不同的圆柱投影。所有的这些圆柱投影都是等面积且不保角的。所有经线和纬线都是直线。在高纬度处畸变很大。

该投影的参数为:

```
-JY<lon>/<lat>/<width>
-Jy<lon>/<lat>/<scale>
```

<lon> 是中心经线, <lat> 是标准纬线。

标准纬线可以取任意值，下面列出了一些比较流行的标准纬线的选择：

Balthasar	50°
Gall	45°
Hobo-Dyer	37°30' (= 37.5°)
Trystan Edwards	37°24' (= 37.4°)
Caster	37°04' (= 37.0666°)
Behrman	30°
Lambert	0°

```
gmt pscoast -R-145/215/-90/90 -JY35/30/4.5i -B45g45 -Dc -A10000 -Sdodgerblue \
-Wthinnest -P > GMT_general_cyl.ps
```

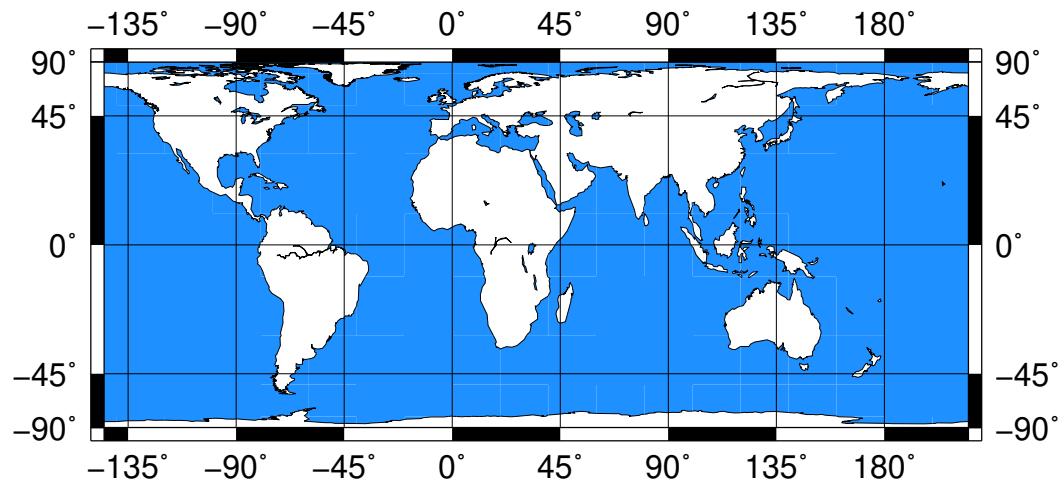


图 9.38: 使用 Behrman 圆柱等面积投影绘制地图

保护环境，从阅读电子文档开始！

第 10 章 配置 GMT

10.1 GMT 配置简介

10.1.1 简介

GMT 中还有 150 多个配置参数，用于控制图像的外观（如底图边框的画笔粗细、颜色，文字标注的字体、大小和颜色等）和数据的处理方式（如默认的插值方式、地图投影使用的椭球等）。

当执行一个 GMT 模块时，GMT 首先会将所有配置参数初始化为 GMT 的默认值，然后去搜索 `gmt.conf` 文件。若找到该文件，则会读取该文件中的参数值并覆盖 GMT 配置参数默认值。

GMT 参数配置文件 `gmt.conf` 的搜索路径按优先级排序为：`./gmt.conf > ~/.gmt/gmt.conf > ~/gmt.conf`。即 GMT 程序首先会在当前目录寻找参数配置文件；若未找到，则到 `~/.gmt` 目录下寻找；若仍未找到，则在家目录下寻找。

10.1.2 设计思路

GMT 将这 100 多个配置参数单独放置在配置文件 `gmt.conf` 中，主要是基于两方面的考虑：

1. 让每个模块的命令行语法中都涵盖如此多的参数是不实际的，这些参数有些很少或几乎不需要改变（比如地图投影过程中使用的地球椭率）
2. 不同的期刊可能要求不同的字体和字号，用于 PowerPoint 演示的图件也可能与期刊的要求不同。可以针对不同的应用环境设计不同的 GMT 配置参数，保存到不同的 `gmt.conf` 文件中。对于特定的应用环境，可以将对应的配置文件 `gmt.conf` 复制到 GMT 脚本所在目录再执行脚本即可，可以尽可能的避免由于环境不同而导致的大量参数微调。

10.1.3 修改 GMT 配置

GMT 系统自带的 `gmt.conf` 文件中对每个参数都给了一个合理的默认值，但有时用户需要修改这些参数值，GMT 提供了多种方法来实现修改。

1. 用 `gmtset` 模块在脚本执行的过程中显式地修改 GMT 参数值

比如，需要设置主标注的字体为 `12p,Times-Bold,red`：

```
gmt set FONT_ANNOT_PRIMARY 12p,Times-Bold,red
```

`gmtset` 会修改当前目录下的 `gmt.conf` 文件中的相应参数值，进而影响到接下来其它 GMT 程序的执行。若当前目录没有 `gmt.conf` 文件，则 `gmtset` 会先复制系统自带的 `gmt.conf` 文件，再修改之。该命令修改的参数将一直生效，直到被新值覆盖。

2. 若你需要修改某些参数值，使得其在执行单个命令时有效，而不影响其他命令的执行效果，可以考虑在该命令行上使用 `--PAR=value` 语法。

比如，针对某个 GMT 命令，为了临时设置浮点数的输出格式包含更多的小数位，而不影响其他命令的浮点数输出格式，可以在该命令中加上 `--FORMAT_FLOAT_OUT=%.16lg`。

3. 在单个 GMT 命令中使用 `+<defaultfile>` 的语法，来指定使用特定配置文件 `<defaultfile>`，该方法仅对单个命令有效：

```
gmt psxy ... +custom_gmt.conf
```

4. 在脚本开始执行之前，将要使用的配置文件 `gmt.conf` 复制到当前目录，待脚本执行完毕后，删除该配置文件。该方法要求写脚本时要非常小心，因为若脚本因为错误出现中断，会影响到配置文件的修改和删除，进而导致难以解释的效果。

通常，仅推荐使用方法一和方法二。

注解： 在使用方法一时，应在脚本的最后删除配置文件，以避免一个潜在的问题。

假如一个绘图脚本由三个命令构成：

```
gmt psxy ...
gmt set ...
gmt psxy ...
```

第一个命令 `psxy` 绘制了一张图，第二个命令 `set` 修改了 GMT 配置参数，例如将默认字体 A 修改为字体 B，第三个命令 `psxy` 绘制了第二张图。此时，第一个命令 `psxy` 用的是默认字体 A，而第三个命令 `psxy` 用的是字体 B。

当执行完一次该脚本时，在当前目录下会生成配置文件 `gmt.conf`，其中的配置参数使用的是字体 B。

若再次执行该脚本，由于当前目录下已经有了上一次执行生成的 `gmt.conf` 文件，且文件中配置参数使用的是字体 B，则第一个命令 `psxy` 受到该参数文件影响而使用了字体 B，而第二个命令 `set` 将字体 B 修改为字体 B（相当于没修改），第三个命令 `psxy` 则使用字体 B。这导致了执行同一个脚本出现了不同的结果，且难以调试和拍错。

因而，在脚本结束时应删除当前目录下的参数配置文件，甚至删除其他一些中间文件。

10.1.4 查看配置参数的值

使用:

```
gmt defaults -D
```

即可查看所有 GMT 配置参数的默认值。

使用:

```
gmt get FORMAT_GEO_MAP
```

可以查看单个配置参数 FORMAT_GEO_MAP 的当前值。

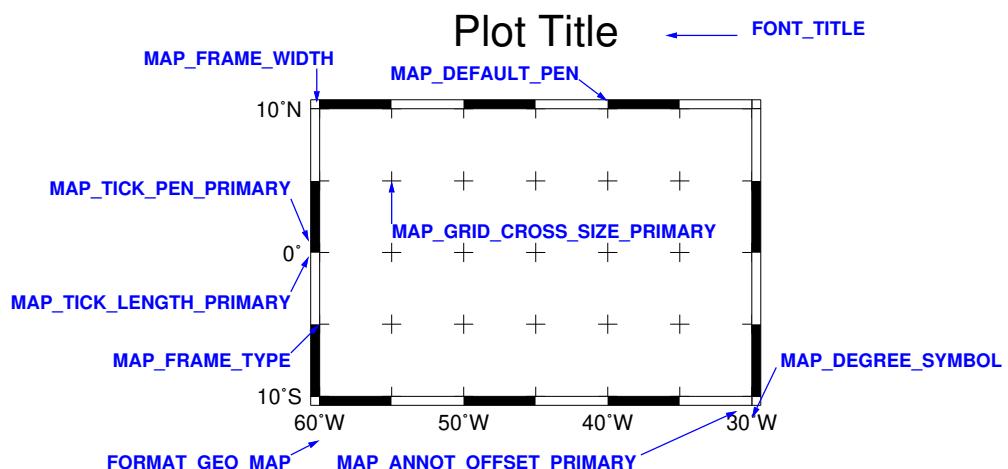
10.1.5 删除配置文件

可以使用 gmt clear conf 命令删除配置文件 gmt.conf。

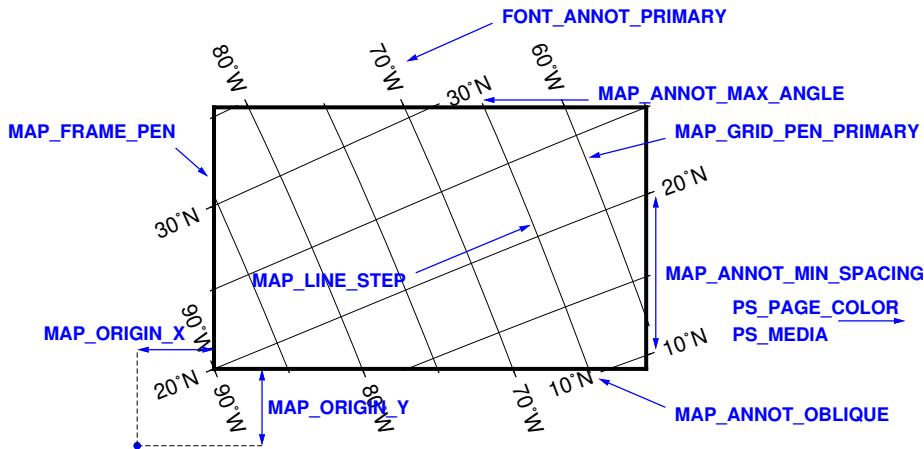
10.1.6 GMT 配置示例

下面列出部分会影响到绘图效果的 GMT 参数。

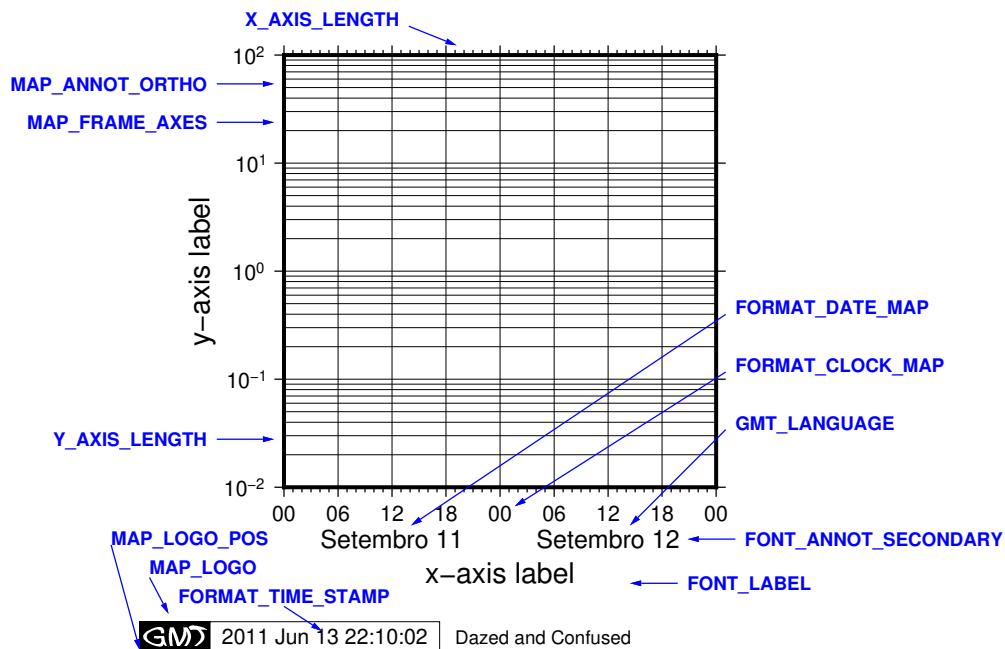
Source Code



Source Code



Source Code



10.2 COLOR 参数

下面列出所有与颜色相关的参数，参数的默认值在中括号内列出。

COLOR_BACKGROUND 图片背景色，数据 Z 值小于 CPT 文件中最小值所对应的背景色 [black]

COLOR_FOREGROUND 图片前景色，数据 Z 值大于 CPT 文件中最大值所对应的前景色 [white]

COLOR_NAN Z 值等于 NaN 时所使用的颜色 [127.5]

某些绘图命令（比如 `grdimage`、`colorbar`）可以使用强度文件（intensity file）来模拟光照效果。光照效果的实现，本质上是先将任意颜色转换成 HSV 模型，然后根据强度的正负，增大/减小 HSV 模型中的 S（饱和度）和 V（明度），以达到模拟光照的效果。下面

的四个参数控制了模拟光照过程中 S 和 V 变化的极限值, 估计是为了避免过亮或过暗的情况。

COLOR_HSV_MAX_S 最大正强度对应的 S 值, 取值范围 0 到 1 [0.1]

COLOR_HSV_MIN_S 最小负强度对应的 S 值, 取值范围 0 到 1 [1.0]

COLOR_HSV_MAX_V 最大正强度对应的 V 值, 取值范围 0 到 1 [1.0]

COLOR_HSV_MIN_V 最小负强度对应的 V 值, 取值范围 0 到 1 [0.3]

COLOR_MODEL 对 CPT 文件做插值时所使用的颜色空间 [none]

可以取如下值:

- **none**: 使用 CPT 文件的注释行中指定的 COLOR_MODEL
- **rgb**: 在 RGB 颜色空间中插值
- **hsv**: 在 HSV 颜色空间中插值
- **cmyk**: 假定文件是 CMYK 颜色, 但在 RGB 空间插值

在 RGB 颜色空间内插值可能会产生异常的色调, 而直接在 HSV 颜色空间插值则可以更好地保留色调值。

10.3 DIR 参数

下面列出所有与目录相关的参数, 参数的默认值在中括号内列出。

DIR_DCW DCW 数据文件的路径, 默认值为空, GMT 会自动猜测合理的路径值。

DIR_GSHHG 海岸线数据所在路径 [`$GMT_SHAREDIR/coast`]

DIR_DATA 存放数据文件的目录, 默认值为空。

GMT 在命令中遇到文件名时, 会首先在当前目录下寻找, 若找不到, 则会到参数 **DIR_DATA** 指定的目录中寻找, 若找不到, 则到环境变量 `${GMT_DATADIR}` 所指定的目录中寻找。

DIR_CACHE GMT 用户存储从外部 URL 链接中下载得到的数据文件的目录。

10.4 FONT 参数

本页面列出所有字体相关参数, 参数的默认值在中括号内列出。

FONT 一次性设置所有 FONT 类参数 (**FONT_LOGO** 除外) 的默认字体, 该参数不在 `gmt.conf` 中。

FONT_ANNOT 同时设置 **FONT_ANNOT_PRIMARY** 和 **FONT_ANNOT_SECONDARY** 的值。

FONT_ANNOT_PRIMARY 一级 (Primary) 标注的字体 [12p, Helvetica,

black]

若在该参数的值前加上 +，则其他字体、偏移量、刻度长度等参数值会相对于 **FONT_ANNOT_PRIMARY** 成比例缩放。

FONT_ANNOT_SECONDARY 二级 (Secondary) 标注所使用的字体 [14p, Helvetica, black]

FONT_LABEL 轴标签所使用的字体 [16p, Helvetica, black]

FONT_TITLE 图上方标题所使用的字体 [24p, Helvetica, black]

FONT_LOGO GMT 时间戳中字符串所使用的字体 [8p, Helvetica, black]

该参数中仅字体 ID 有效，字号及颜色均无效。

10.5 FORMAT 参数

下面列出所有与格式相关的参数，通常包括输入格式、输出格式和绘图格式三类，参数的默认值在中括号内列出。

日期数据的输入/输出/绘图格式：

FORMAT_DATE_IN 输入数据中日期字符串的格式模板 [yyyy-mm-dd]

日期字符串可以用公历表示，也可以用 ISO 周历表示。

对于公历而言，可以将如下模板做合理组合：

- **yyyy**: 四位年份
- **yy**: 两位年份
- **mm**: 两位月份
- **o**: 月份的名称简写
- **dd**: 两位日期
- **jjj**: 一年中的第几天

比如 **ddmmmyy**、**yy-mm-dd**、**dd-o-yyyy**、**yyyy/dd/mm**、**yyyy-jjj** 等。

对于 ISO 周历而言，其格式为 **yyyy[-]W[-]ww[-]d** 模板，表示某年的第 **ww** 周的第 **d** 天。比如 **yyyyWwwd** 或 **yyyy-Www** 等。

FORMAT_DATE_OUT 输出日期字符串时所使用的格式 [yyyy-mm-dd]

参考 **FORMAT_DATE_IN** 的相关说明。除此之外：

- 若模板开头有一个 -，则所有的整数年月日在输出时会省略前置零。比如若使用模板 **-yyyy-mm-dd** 则输出类似于 **2012-1-3** 而不是 **2012-01-03**
- 若模板为 -，则输出时省略日期，日期和时间中的 T 也会被省略

FORMAT_DATE_MAP 绘制日期字符串时所使用的格式 [yyyy-mm-dd]

参考 *FORMAT_DATE_IN* 和 *FORMAT_DATE_OUT* 中的相关说明。除此之外,

- 绘制月份名时的 mm 可以用 o 替代, 即图上显示 Jan 而不是 01
- 用 u 代替 W[-]ww, 即图上显示 Week 10 而不是 W10

所有的日期文本字符串都由 *GMT_LANGUAGE* 、 *FORMAT_TIME_PRIMARY_MAP* 和 *FORMAT_TIME_SECONDARY_MAP* 控制。

时间数据的输入/输出/绘图格式:

FORMAT_CLOCK_OUT 输出时间字符串时所使用的格式 [hh:mm:ss]

默认使用 24 小时制。若要使用 12 小时制, 可以在字符串的最后加上 am、AM、a.m.、A.M.。比如 hh:mm:ss.xxx、hh:mm、hh:mm.xxx、hha.m. 等等。

- 若时间格式模板以 - 开头, 则输出时间字符串时不会输出前置 0
- 若时间格式模板为 -, 则在输出日期时间时不输出时间字符串

FORMAT_CLOCK_IN 输入数据中时间数据的格式 [hh:mm:ss]

默认使用 24 小时制, 即 hh:mm:ss, 若要使用 12 小时制, 则在参数后加上 am|pm|AM|PM。比如 hh:mm 或 hh:mm:ssAM

FORMAT_CLOCK_MAP 图上绘制时间字符串时所使用的格式 [hh:mm:ss]

地理坐标的输出/绘图格式:

FORMAT_GEO_OUT 控制地理坐标数据的输出格式 [D]

格式的通用形式有两类:

- [±]D: 表示将地理数据以浮点数的形式输出, 浮点数的格式由 *FORMAT_FLOAT_OUT* 决定
 - D: 经度输出范围为 -180 到 180
 - +D: 经度输出范围为 0 到 360
 - -D: 经度输出范围为 -360 到 0
- [±]ddd[:mm[:ss]][.xxx][F|G]
 - ddd: 固定格式的整型弧度
 - :: 分隔符
 - mm: 固定格式的整型弧分
 - ss: 固定格式的整型弧秒
 - .xxx: 前一个量的小数部分
 - F: 用 WSEN 后缀来表示正负号
 - G: 与 F 相同, 但后缀前有一空格
 - ±: 默认经度范围为 -180 到 180, 若加正号则范围为 0 到 360, 加负号则范围为 -360 到 0

示例及效果:

- `ddd:mmF` => `35:45W`
- `ddd:mmG` => `35:45 W`
- `ddd:mm:ss` => `40:34:24`
- `ddd.xxx` => `36.250`

FORMAT_GEO_MAP 绘图时地理坐标的显示格式 [`ddd.mm.ss`]

格式的具体定义参考 `FORMAT_GEO_OUT` , 但具体格式会进一步由 `-B` 选项中的值控制。除此之外, 还可以在格式后面加上 `A` 以表示绘制坐标的绝对值。

浮点数的输出/绘图:

FORMAT_FLOAT_OUT 双精度浮点数在输出时所使用的格式 [`%.12lg`]

具体的格式遵循 C 语言 `printf` 函数的格式定义, 比如 `%.3lf` 。

若需要为不同列指定不同的输出格式, 可以使用多个逗号分隔的 `cols:format` 形式。其中, `cols` 可以是列号 (比如 5 代表数据的第六列), 也可以是列范围 (比如 3-7 表示第 4 到 8 列), 不指定 `cols` 的格式将用于其他余下的列。比如 `0:%.3lf`, `1-3: %.12lg, %lf`

也可以列出 N 个用空格分隔的格式, 这些格式分别应用到数据的前 N 列中, 比如 `%.3lf %.2lf %lf` 。

注解:

1. 由于 GMT 内部将所有数字以浮点型保存, 因而若使用整型格式 `%d` 显示则会出现错误
 2. 若设置为 `%'lg`, 则 `10000` 会显示成 `10,000` 。由于单引号的特殊意义, 因而可能需要转义, 即写成 `%\'lg`
-

FORMAT_FLOAT_MAP 以双精度浮点数形式绘制地图边框标注或等值线标注时所使用的格式 [`%.12lg`]

见 `FORMAT_FLOAT_OUT` 中的相关说明。

FORMAT_TIME_MAP 同时设置 `FORMAT_TIME_PRIMARY_MAP` 和 `FORMAT_TIME_SECONDARY_MAP` 的值

FORMAT_TIME_PRIMARY_MAP 一级标注中月份、周名的格式 [`full`]

可以取如下值:

- `full`: 显示全称, 比如 `January`
- `abbreviate`: 显示简称, 比如 `Jan`

- **character**: 显示单个字符, 比如 J

还可以使用 **Full**、**Abbreviate**、**Character** 表示所有名字均大写。

全称、简称以及单字符的定义, 见 `$GMTHOME/share/localization/gmt_us.locale`

FORMAT_TIME_SECONDARY_MAP 二级标注中月份、周名的格式 [full]

见 **FORMAT_TIME_PRIMARY_MAP** 中的相关说明。

FORMAT_TIME_STAMP GMT 时间戳中时间信息的显示格式 [%Y %b %d %H:%M:%S]

该选项的值用 C 函数 `strftime` 解析, 故而理论上可以包含任意文本。

10.6 IO 参数

IO_HEADER 指定输入/输出的表文件中是否有文件头记录 [false]

可以取 `true|false`。若值为 `true`, 则相当于使用了 `-h` 选项

IO_N_HEADER_RECS 在使用 `-h` 选项时, 要跳过的文件头记录的数目 [0]

见 [-h 选项 和表数据](#)。

IO_LONLAT_TOGGLE 数据的前两列是纬度、经度而不是经度、纬度 [false]

该参数的作用与 `-:` 选项相同, 见 [-: 选项](#)。其可以取如下值:

1. `false` 默认值, 输入/输出数据均为 (x, y)
2. `true` 输入/输出数据均为 (y, x)
3. `IN` 仅输入数据为 (y, x)
4. `OUT` 仅输出数据为 (y, x)

IO_NAN_RECORDS 控制当读入的记录中的 X、Y 或 Z 包含 NaN 记录时的处理方式 [pass]

可以取如下值:

- `skip`: 直接跳过 NaN 记录, 并报告 NaN 记录的数目
- `pass`: 将所有记录传递给程序

IO_COL_SEPARATOR GMT 输出 ASCII 表数据时列与列之间的分隔符 [tab]

可以取 `tab`、`space`、`comma` 和 `none`

IO_SEGMENT_MARKER 多段数据中每段数据开始的标识符 [>]

见 [表数据](#) 中的相关介绍。若希望输入和输出数据中使用不同的数据段标识符, 则可以使用逗号分隔输入和输出数据的段标识符, 比如 `>, :.`。

有两个特殊的标识符：

1. **B** 表示将空行作为数据段开始的标识符
2. **N** 表示将一个 NaN 记录作为数据段开始的标识符

To use B or N as regular segment markers you must escape them with a leading backslash.

IO_FIRST_HEADER 若整个数据中只有一个数据段时，是否要写这个数据段的文件头记录。默认情况下，只有当这个单独段的头段记录中有额外的内容时才会写该头记录。可选的值包括 `always`、`never` 和 `maybe`。

IO_NC4_CHUNK_SIZE 控制写 netCDF 文件时的分块大小 [auto]

IO_NC4_DEFLATION_LEVEL 输出 netCDF4 格式的数据时所使用的压缩等级 [3]

可以取 0 到 9 的整数，0 表示不压缩，9 表示最大压缩。低压缩率可以提高性能并减少文件尺寸，而高压缩率虽然可以进一步减小文件尺寸，但却需要更多的处理时间。

IO_SEGMENT_BINARY 二进制数据中，某个记录的所有值都是 NaN 时该如何解释 [2]

默认情况下，当二进制数据中某个记录的值为 NaN 的列数超过了 **IO_SEGMENT_BINARY** 的值时，则将该记录解释为二进制数据中的数据段头记录。将该参数赋值为 0 或 off 可以关闭这一特性。

IO_GRIDFILE_SHORTHAND 是否支持自动识别网格文件后缀的功能 [false]

见[网格文件后缀](#)一节。若设置为 `true`，则会检测每个网格文件的后缀是否在用户自定义文件后缀中；若为 `false`，则不检测用户自定义文件后缀。

IO_GRIDFILE_FORMAT GMT 默认使用的网格文件格式 [nf]

见[读 netCDF 文件](#)一节。

10.7 MAP 参数

MAP_DEFAULT_PEN 设置所有与 -W 选项相关的画笔属性的默认值 [default,pen]

在参数值的前面加上 + 可以覆盖其他 PEN 相关参数中的颜色。

MAP_FRAME_AXES 要绘制/标注的轴 [WSEN]

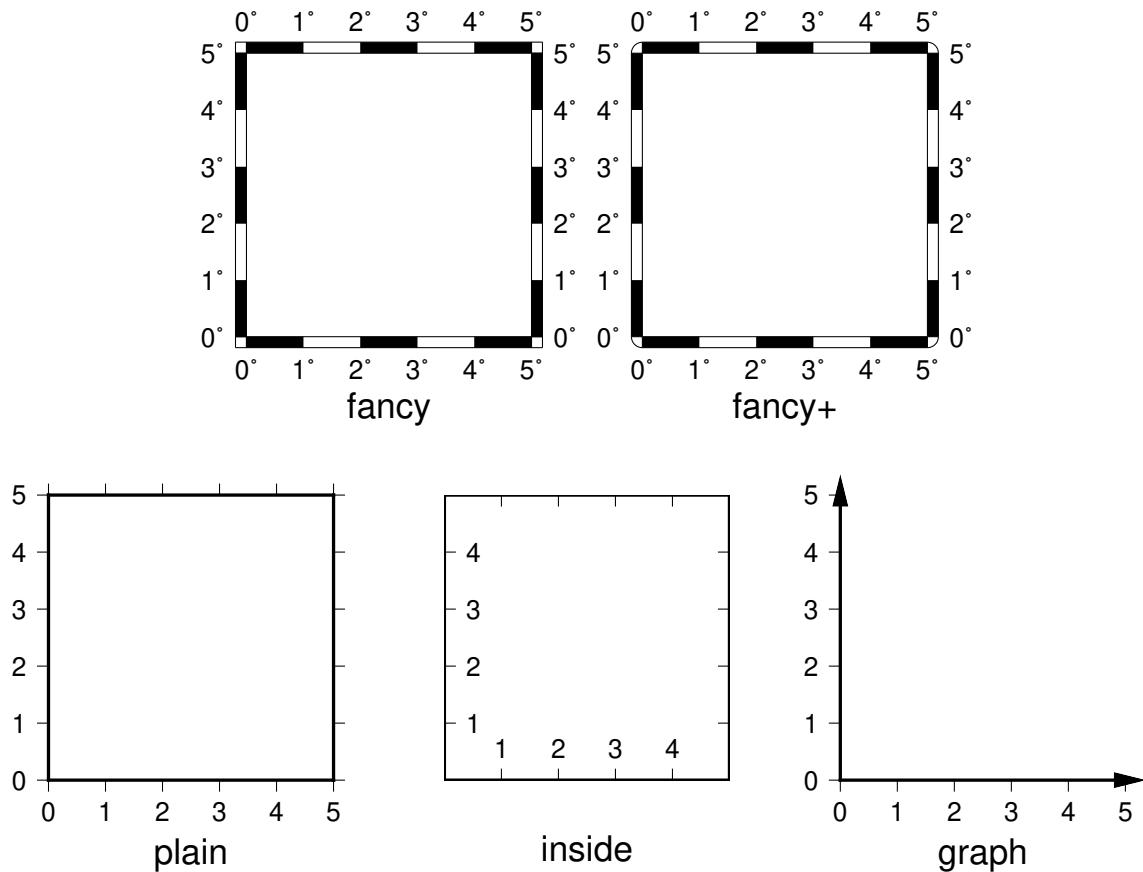
默认值为 WSEN，绘制并标注四条边，可以通过 -B 选项控制实际绘制的边。详情见[-B 选项](#) 选项。

MAP_FRAME_TYPE 底图边框类型 [fancy]

可选值包括 `inside|plain|graph|fancy|fancy+`。一般情况下, `fancy` 边框类型仅适用于投影后的 X、Y 方向平行于经度纬度方向的情况, 比如 rectangular 投影、polar 投影。对于某些投影, 只能使用 `plain` 底图, 即便 `MAP_BASEMAP_TYPE` 被设置为 `fancy`。

下图给出了不同的底图边框类型的效果:

Source Code



`MAP_FRAME_PEN` 绘制底图类型为 `plain` 时边框的画笔属性 [`thicker, black`]

`MAP_FRAME_WIDTH` 设置底图类型为 `fancy` 时的边框宽度 [`5p`]

`MAP_ORIGIN_X` 新绘图在纸张上的原点的 X 坐标 [`1i`]

`MAP_ORIGIN_Y` 设置新绘图在纸张上的原点的 Y 坐标 [`1i`]

`MAP_LOGO` 是否在左下角绘制 GMT 时间戳 [`false`]

可以取 `true|false`, 等效于在命令行中使用 `-U` 选项。

`MAP_LOGO_POS` GMT 时间戳相对于当前绘图原点的对齐方式与位置 [`BL/-54p/-54p`]

`MAP_TITLE_OFFSET` 图标题的底部与轴标注 (或轴标签) 的顶部之间的距离

[14p]

MAP_HEADING_OFFSET 子图标题的顶部与图总标题的底部之间的距离
[18p]

MAP_SCALE_HEIGHT 地图比例尺的高度 [5p]

MAP_TICK_PEN 同时设置 **MAP_TICK_PEN_PRIMARY** 和
MAP_TICK_PEN_SECONDARY 的值

MAP_TICK_PEN_PRIMARY 一级刻度的画笔属性 [thinner,black]

MAP_TICK_PEN_SECONDARY 二级刻度的画笔属性 [thinner,black]

MAP_TICK_LENGTH 同时设置 **MAP_TICK_LENGTH_PRIMARY** 和
MAP_TICK_LENGTH_SECONDARY 的值

MAP_TICK_LENGTH_PRIMARY 一级刻度的主刻度和次刻度的长度
[5p/2.5p]

若只给定一个长度值，则次刻度的长度假定为主刻度的一半

MAP_TICK_LENGTH_SECONDARY 二级刻度的主刻度和次刻度的长度
[15p/3.75p]

若只给定一个长度值，则次刻度的长度假定为主刻度的 25%

MAP_LINE_STEP 绘制线段时所使用的最大步长 [0.75p]

地理投影下，两点之间会用大圆路径连接，因而 GMT 需要先计算大圆路径上的其他中间点的坐标，并将这些点用直线连起来。若该步长太大，会导致大圆路径看上去很不光滑。

MAP_GRID_PEN 同时设置 **MAP_GRID_PEN_PRIMARY** 和
MAP_GRID_PEN_SECONDARY 的值

MAP_GRID_PEN_PRIMARY 一级网格线的线条属性 [default,black]

MAP_GRID_PEN_SECONDARY 二级网格线的线条属性 [thinner,black]

MAP_GRID_CROSS_SIZE 同时设置 **MAP_GRID_CROSS_SIZE_PRIMARY** 和
MAP_GRID_CROSS_SIZE_SECONDARY 的值

MAP_GRID_CROSS_SIZE_PRIMARY 一级网格交叉线的大小，0 表示绘制连续的网格线 [0p]

MAP_GRID_CROSS_SIZE_SECONDARY 二级网格交叉线的大小，0 表示绘制连续的网格线 [0p]

MAP_ANNOT_OFFSET 同时设置 **MAP_ANNOT_OFFSET_PRIMARY** 和

MAP_ANNOT_OFFSET_SECONDARY 的值

MAP_ANNOT_OFFSET_PRIMARY 一级标注的开始位置与刻度尾端间的距离 [5p]

MAP_ANNOT_OFFSET_SECONDARY 二级标注的底部与 secondary 标注的顶部之间的距离 [5p]

MAP_LABEL_OFFSET 轴标注底部与轴标签顶部间的距离 [8p]

MAP_VECTOR_SHAPE 矢量箭头的形状 [0]

取值范围为 -2 到 2。0 表示矢量头为三角形, 1 表示箭头形状, 2 表示打开的 V 字。其他的中间值代表了两种形状的中间值。

MAP_DEGREE_SYMBOL 在地图上绘制“度”时所使用的符号 [ring]

可以取 ring|degree|colon|none

MAP_ANNOT_MIN_ANGLE 对于某些倾斜投影方式而言, 如果标注的基线与地图的边界之间的夹角小于该值, 则不绘制标注。合理的取值范围为 [0,90] [20]

MAP_ANNOT_MIN_SPACING 在某些倾斜投影中, 相邻两个标注之间的最小距离, 若标注的距离小于该值, 则不绘制 [0p]

MAP_ANNOT_ORTHO 控制哪些轴的标注垂直于轴 [we]

该参数可以将 wesnz 做任意组合

MAP_ANNOT_OBLIQUE 见官方文档

MAP_POLAR_CAP 控制网格线在两极附近的显示, 见官方文档

10.8 PROJ 参数

本节列出投影相关参数, 参数的默认值在中括号内列出。

PROJ_LENGTH_UNIT 长度量的默认单位 [c]

见[单位](#)一节。

PROJ_ELLIPSOID 地图投影中使用的地球椭球标准 [WGS-84]

GMT 支持几十种地球椭球标准, 具体可取值见官方文档。除此之外, GMT 还支持自定义椭球, 用户只需按照固定的格式对椭球命名, GMT 会从椭球名字中提取半长轴以及扁率。可用的格式如下:

- a: 球半径为 a, 单位为 m, 扁率为零。比如 6378137
- a, inv_f: inv_f 为扁率的倒数, 比如 6378137, 298.257223563
- a, b=semi_minor: semi_minor 为半短轴长度, 单位为 m。比如 6378137,

b=6356752.3142

- a, f=flattening: flattening 为扁率, 比如 6378137, f=0.0033528

需要注意, 对于某些全球投影, GMT 会对选中的地球椭球做球状近似, 将扁率设为零, 并使用其平均半径。当 GMT 做此类近似时, 会给出警告信息。

PROJ_AUX_LATITUDE 球体近似时的辅助纬线 [authalic]

在使用大圆弧距离计算方式时, 需要将真实地球近似为一个半径为 *PROJ_MEAN_RADIUS* 的球体, 在做球体近似时需要选择合适的辅助纬线。可选值包括

- authalic
- geocentric
- conformal
- meridional
- parametric
- none

当设置为除 *none* 外的其他值时, GMT 会在计算距离前, 将大圆弧距离计算时使用的两点中任意一点的纬度转换成辅助纬度。

PROJ_MEAN_RADIUS 地球/行星的平均半径 [authalic]

在计算两点间的大圆弧距离或区域的表面积时才会被使用。可选值包括

- mean (R_1)
- authalic (R_2)
- volumetric(R_3)
- meridional
- quadratic

PROJ_SCALE_FACTOR 修改某些投影的地图缩放因子以减小面积失真

- Polar Stereographic: 默认值为 0.9996
- UTM: 默认值为 0.9996
- Transverse Mercator: 默认值为 1

PROJ_GEODESIC 指定大地测量距离中所使用的算法 [Vincenty]

可以取:

1. Vincenty 默认值, 精确到 0.5mm
2. Rude given for legacy purpose
3. Andoyer 精度为 10 米量级, 比 Vincenty 快 5 倍

10.9 PS 参数

本节列出所有与 PS 相关的参数, 参数的默认值在中括号内列出。

PS_CHAR_ENCODING 字符集编码方式 [ISOLatin1+|Standard+]

在处理文本中的八进制码时所使用的字符集编码方式，可选值包括：**Standard**、**Standard+**、**ISOLatin1**、**ISOLatin1+** 和 **ISO-8859-x**（其中 x 取值为 [1-10] 或 [13-15]）。若安装 GMT 时使用 SI 单位制，则默认值为 ISOLatin1 编码；否则使用 Standard+ 编码。

PS_LINE_CAP 控制线段的端点的绘制方式 [butt]

可以取如下值：

- **butt**: 不对端点做特殊处理，即端点是矩形（默认值）
- **round**: 端点处为直径与线宽相等的半圆弧
- **square**: 端点处为边长与线宽相等的半个正方形

下图展示了 **PS_LINE_CAP** 取不同值时线段端点的区别。需要注意，图中三条线段的长度是相同的，但因为参数设置不同而导致实际线段长度看上去有些不一样。

Source Code



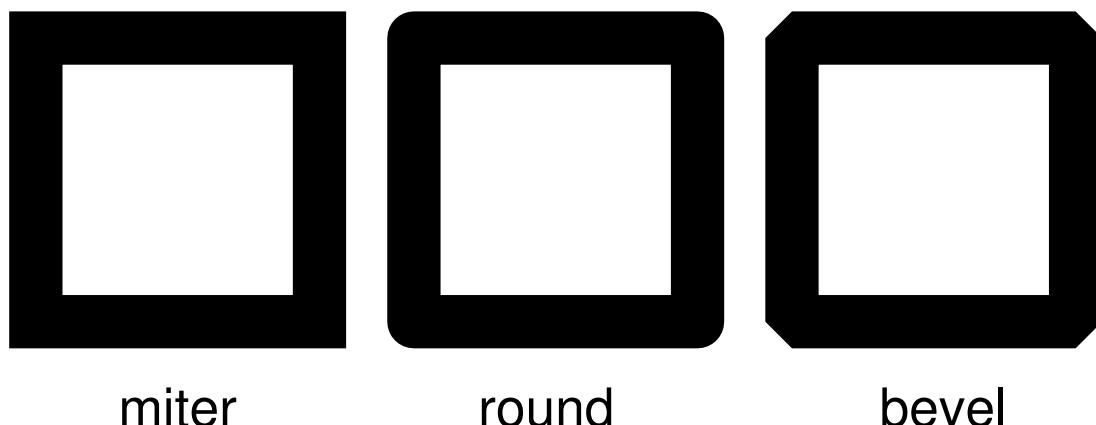
说明：当 **PS_LINE_CAP=round** 时，若线段长度为零，则线段将以圆的形式存在，这可以用于创建圆点线，此时将同一条线绘制两次，每次使用不同的相移和颜色，则可以创建颜色变化的线条。

PS_LINE_JOIN 控制线段拐点的绘制方式 [miter]

可以取 **miter**、**round**、**bevel**

下图展示了 **PS_LINE_JOIN** 取不同值时线段拐点的绘图效果。当线宽较小时，几乎看不出来区别，这里为了显示的效果，将线宽设置为 20p。

Source Code



PS_MITER_LIMIT 设置 mitered 拐点的角度阈值 [35]

当两个相交的线段之间的夹角小于该阈值时，则该拐角会被 bevelled 而不是被 mitered。该参数的取值范围为 0 到 180。若设置为 0，则使用 PS 的默认值（11 度），若设置为 180，则所有拐角都会被 beveled。

PS_MEDIA 设置当前纸张的尺寸 [a4|letter]

下表列出了 GMT 预定义的若干种纸张尺寸及其对应的宽度和高度（单位为 points）。

表 10.1: GMT 预定义纸张大小

Media	width	height	Media	width	height
A0	2380	3368	archA	648	864
A1	1684	2380	archB	864	1296
A2	1190	1684	archC	1296	1728
A3	842	1190	archD	1728	2592
A4	595	842	archE	2592	3456
A5	421	595	flsa	612	936
A6	297	421	halfletter	396	612
A7	210	297	statement	396	612
A8	148	210	note	540	720
A9	105	148	letter	612	792
A10	74	105	legal	612	1008
B0	2836	4008	11x17	792	1224
B1	2004	2836	tabloid	792	1224
B2	1418	2004	ledger	1224	792
B3	1002	1418			
B4	709	1002			
B5	501	709			

用户还可以用 WxH 的格式完全自定义纸张尺寸，其中 W 和 H 分别为纸张的宽度和高度。比如 12cx12c 表示纸张为宽度和高度都为 12 厘米。

若某些尺寸经常使用，用户还可以自定义纸张格式，只需要新建或编辑 `~/.gmt/gmt_custom_media.conf` 即可，文件格式也很简单：

```
# 纸张格式名 宽度 高度
paper1      2000  3000
paper2      3000  0
```

纸张高度为 0，表示纸张可以向上无限延展。

PS_PAGE_COLOR 设置纸张的背景色 [white]

PS_PAGE_ORIENTATION 设置纸张方向 [landscape]

可以取 `portrait` 或 `landscape`。

PS_SCALE_X 绘图时 X 方向的全局比例 [1.0]

用于实现图像的整体缩放

PS_SCALE_Y 绘图时 Y 方向的全局比例 [1.0]

用于实现图像的整体缩放

PS_TRANSPARENCY 设置生成 PS 文件所使用的透明模式 [Normal]

可取值包括 `Color`、`ColorBurn`、`ColorDodge`、`Darken`、`Difference`、`Exclusion`、`HardLight`、`Hue`、`Lighten`、`Luminosity`、`Multiply`、`Normal`、`Overlay`、`Saturation`、`SoftLight`、`Screen`

PS_COLOR_MODEL 设置生成 PS 代码时颜色所使用的颜色模型 [rgb]

可以取 `RGB`、`HSV`、`CMYK` 或 `GRAY`。若设置为 `HSV`，其不会影响绘图过程中使用 `RGB` 指定的颜色；若设置为 `GRAY`，则所有的颜色都将使用 `YIQ` 方法转换成灰度。

PS_COMMENTS 生成的 PS 代码中是否包含注释信息 [false]

若为 `true`，则生成的 PS 文件中会包含注释，用于解释文件中操作的逻辑，当你需要手动编辑 PS 文件时比较有用。默认情况下，其值为 `false`，即 PS 文件中不会包含注释，此时生成的 PS 文件更小。

PS_CONVERT 现代模式下调用 `psconvert` 命令时默认加上的参数，多个参数之间用逗号分隔 [A,P]

PS_IMAGE_COMPRESS 设置 PS 中的图像压缩算法 [deflate,5]

可以取值为

- `rle`: Run-Length Encoding scheme
- `lzw`: Lempel-Ziv-Welch compression
- `deflate[,level]`: DEFLATE compression, `level` 可以取 1 到 9;
- `none`: 不压缩，相当于 `deflate,5`。

10.10 TIME 参数

本节列出所有时间相关参数，参数的默认值在中括号内列出。

TIME_EPOCH 指定所有相对时间的参考时刻 [1970-01-01T00:00:00]

其格式为 `yyyy-mm-ddT[hh:mm:ss]` 或 `yyyy-Www-ddTT[hh:mm:ss]`。

TIME_UNIT 指定相对时间数据相对于参考时间的单位 [s]

可以取:

1. y: 年; 假定一年 365.2425 天;
2. o: 月; 假定所有月是等长的;
3. d: 天;
4. h: 时;
5. m: 分钟;
6. s: 秒;

TIME_SYSTEM TIME_EPOCH 和 **TIME_UNIT** 的合并版

即指定 **TIME_SYSTEM** 相当于同时指定了 **TIME_EPOCH** 和 **TIME_UNIT**。可取如下值:

1. JD: 等效于 -4713-11-25T12:00:00 d
2. MJD: 等效于 1858-11-17T00:00:00 d
3. J2000: 等效于 2000-01-01T12:00:00 d
4. S1985: 等效于 1985-01-01T00:00:00 s
5. UNIX: 等效于 1970-01-01T00:00:00 s
6. RD0001: 等效于 0001-01-01T00:00:00 s
7. RATA: 等效于 0000-12-31T00:00:00 d

该参数并不存在于 **gmt.conf** 中, 当指定该参数时, 其会被自动转换为 **TIME_EPOCH** 和 **TIME_UNIT** 对应的值。

TIME_WEEK_START 指定周几是一周的第一天, 可取值为 Monday 或 Sunday。

TIME_Y2K_OFFSET_YEAR 当用两位数字表示四位数字的年份时, **TIME_Y2K_OFFSET_YEAR** 给定了 100 年序列的第一年 [1950]

比如, 若 **TIME_Y2K_OFFSET_YEAR**=1729, 则数字 29 到 99 分别表示 1729 到 1799, 而数字 00 到 28 则表示 1800 到 1828。默认值为 1950, 即 00 到 99 表示的年份范围为 1950 到 2049。

TIME_INTERVAL_FRACTION 见官方文档中的说明

TIME_REPORT 见官方文档中的说明

TIME_IS_INTERVAL 见官方文档中的说明

10.11 其他参数

本节介绍 GMT 中的其他参数, 参数的默认值在中括号内列出。

GMT_AUTO_DOWNLOAD 是否允许 GMT 自动从 GMT 服务器 (由 **GMT_DATA_URL** 控制) 下载数据文件到缓存目录 [on]

GMT_DATA_URL GMT 数据服务器地址, 默认使用 SOEST 官方镜像。

GMT_DATA_URL_LIMIT 从远程下载的文件的大小上限, 即 GMT 缓存目录的空间上限, 默认无限制

GMT_COMPATIBILITY 是否开启兼容模式 [4]

- 若值为 4, 表示兼容 GMT4 语法并给出警告
- 若值为 5, 则表示不兼容 GMT4 语法, 严格遵守 GMT5 语法, 遇到 GMT4 语法时直接报错

GMT_VERBOSE 控制 GMT 命令的 verbose 级别 [c]

可选值包括:

- quiet
- normal
- compat
- verbose
- long
- debug

也可以直接使用每个级别的第一个字母。每个级别的具体含义见 [-V 选项](#) 一节。

GMT_TRIANGULATE 选择 triangulate 命令中源码的来源 [Watson]

`triangulate` 命令有两个版本的源码, Watson 的版本遵循 GPL, Shewchuk 的版本不遵循 GPL。该选项用于控制要使用哪个版本, Shewchuk 版本拥有更多功能。

GMT_LANGUAGE 绘制月份、星期几等时所使用的语言 [US]

该参数的默认值为 US, 所以在绘图时一月会显示 January 而不是 一月。可以通过修改该参数设置 GMT 所使用的语言。该参数可以取值包含 CN1、CN2、UK、US 等。详细列表见 [GMT 官方文档 gmt.conf](#)

每种语言的定义位于 `$GMTHOME/share/localization` 中。读者可以自定义自己的语言, 并将其放在该目录或 `~/.gmt` 目录下。

实际使用时, 还涉及到编码和字体问题, 所以想做出中文效果还需要一些特殊处理。

GMT_HISTORY GMT 历史文件 `gmt.history` 的处理方式 [true]

- true 可以读写
- readonly 只能读不能写
- false 不显示历史文件

GMT_INTERPOLANT 程序中一维插值所使用的算法 [akima]

1. linear: 线性插值
2. akima: akima's spline

3. `cubic`: natural cubic spline
4. `none`: 不插值

GMT_EXPORT_TYPE 控制表数据的数据类型, 仅被外部接口使用 [double]

可以取 `double`、`single`、`[u]long`、`[u]int`、`[u]short`、`[u]char`

GMT_EXTRAPOLATE_VAL 外插时超过数据区时如何处理 [NaN]

可选值包括:

1. `NaN`: 区域范围外的值一律为 NaN
2. `extrap`: 使用外插算法计算的区域外的值
3. `extrapval, val`: 设置区域外的值为 `val`

GMT_CUSTOM_LIBS 自定义的 GMT 共享库文件, 默认值为空

GMT 支持自定义模块。用户可以自己写一个 GMT 模块, 并将其编译成 GMT 兼容的动态函数库, 并告知 GMT 该函数库的信息, 就可以通过 `gmt xxx` 的语法调用自定义的模块, 以实现扩充 GMT 功能的目的。

该参数用于指定自定义动态库函数的路径, 多个路径之间用逗号分隔。路径可以是共享库文件的绝对路径, 也可以是其所在的目录。若路径是一个目录名, 该目录必须需斜杠或反斜杠结尾, 表明使用该目录下的全部共享库文件。在 Windows 下, 若目录名是 /, 则在 `${GMTHOME}/bin/gmt_plugins` 目录下寻找库文件。

GMT_FFT 要使用的 FFT 算法 [auto]

可以取:

1. `auto`: 自动选择合适的算法
2. `fftw[, planner]`: FFTW 算法, 其中 `planner` 可以取
`measure|patient|exhaustive`
3. `accelerate` OS X 下使用 Accelerate Framework
4. `kiss`: kiss FFT
5. `brenner`: Brenner Legacy FFT

第 11 章 地学数据集

11.1 地学数据集

在使用 GMT 绘图时, 经常会用到一些特定的地学数据, 比如国界线、地形起伏数据等等。这些数据具有两个特性:

1. 固定性: 几乎不需要改动就可以直接使用;
2. 复用性: 经常用于不同的图件。

GMT 自带了一些地学数据, GMT 中文社区也提供了一些地学数据, 用户也可以整理自己的数据集。对于这些非 GMT 自带的数据, 用户可以自行建立一个 GMT 数据库以方便地在 GMT 绘图中使用这些数据集。

11.1.1 数据库原理

当 GMT 命令行中出现某个文件时, GMT 会按照如下顺序依次寻找文件, 直到找到文件为止:

```
当前目录 -> GMT 用户目录 -> GMT 数据目录 -> GMT 缓存目录
```

GMT 用户目录由环境变量 `GMT_USERDIR` 决定。若该环境变量未定义, 则使用默认用户目录。Linux 和 macOS 下默认 GMT 用户目录是 `~/.gmt`, Windows 下默认 GMT 用户目录是 `C:\Users\用户名\.gmt`。

GMT 数据目录由环境变量 `GMT_DATADIR` 决定, 默认值为空。因而, 可以将自己需要的地学数据集放在某个特定的目录下, 然后修改环境变量 `GMT_DATADIR` 使得其包含该特定目录即可。此时, 在 GMT 命令中即可直接使用数据对应的文件名, 而无需指定绝对路径。

11.1.2 建立 GMT 数据库

将数据集放在某个特定目录下, 然后修改环境变量 `GMT_DATADIR` 使得其包含该目录即可。

比如 Linux 和 macOS 下数据库可以放在 `~/GMTDB`, 然后向 `~/.bashrc` 中加入如下语句:

```
export GMT_DATADIR=~/GMTDB
```

对于 Windows 用户, 可以把数据放在 `D:\GMTDB` 下, 然后打开“我的电脑”->“属性”

-> “高级”->“环境变量”，然后添加环境变量，变量名为 `GMT_DATADIR`，值为 `D:\GMTDB`，最后重启电脑使得环境变量生效。

如果喜欢将不同的数据分类放在不同的目录下，则可以向 `GMT_DATADIR` 添加多个目录。Linux 和 macOS 下多个目录之间用冒号 : 分隔，Windows 下多个目录之间用 ; 分隔。例如：

```
export GMT_DATADIR=~/GMTDB/data1:~/GMTDB/data2
```

11.2 GSHHG: 全球高精度海岸线数据

GSHHG 数据主页: <http://www.soest.hawaii.edu/wessel/gshhg/>

GSHHG，全称为 A Global Self-consistent, Hierarchical, High-resolution Geography Database。GMT 提供的 GSHHG 数据中包含了海岸线、河流和国界等数据。

GMT 的 `pscoast` 模块可以直接绘制 GSHHG 中的数据，也可以使用 `pscoast` 的 -M 选项将数据导出为纯文本文件供其他程序使用。

11.2.1 数据精度

GSHHG 提供了五种不同精度的数据，以满足不同的需求。五种精度从高到低分别为：

```
full > high > intermediate > low > crude
```

`pscoast` 的 -D 选项加上每种精度的单词首字母即可指定使用何种精度的数据。在绘制全球地图时，可以用 -Dc 指定使用最低精度的数据，以避免绘制了大量细节而导致文件太大；在绘制几度范围的小区域地图时，则可以使用 -Df 指定使用最高精度的数据。当然，也可以使用 -Da 选项，此时 GMT 会根据当前绘图区域的大小自动选择合适的数据精度。

下面的示例绘制了一个小区域的海岸线边界，可以看到 -D 取不同精度时边界精细程度的差异：

Source Code



11.2.2 数据内容

GSHHG 数据中包含了海岸线数据、河流数据和国界数据。

海岸线

海岸线数据可以进一步细分为 4 个不同的等级：

- 1: 陆地和海洋的分界线, 即真正意义上的海岸线
- 2: 陆地与湖泊的分界线
- 3: 湖泊中的岛屿与湖泊的分界线
- 4: 湖泊中的岛屿里的池塘与岛屿的分界线

pscoast 模块中有以下几个与海岸线相关的选项：

- **-W[<level>/]<pen>** 绘制不同等级的海岸线
- **-G<fill>** 设置陆地、岛屿等陆区的填充色
- **-S<fill>** 设置海洋、湖泊等水区的填充色
- **-Cl<fill>** 设置湖泊的填充色
- **-Cr<fill>** 设置河流湖的填充色

河流

河流进一步可以细分为 10 个等级：

- 0: Double-lined rivers (river-lakes).
- 1: Permanent major rivers.
- 2: Additional major rivers.
- 3: Additional rivers.
- 4: Minor rivers.
- 5: Intermittent rivers - major.
- 6: Intermittent rivers - additional.
- 7: Intermittent rivers - minor.
- 8: Major canals.
- 9: Minor canals.
- 10: Irrigation canals.

pscoast 模块的 **-I** 选项可以用于绘制不同等级的河流, 其基本语法为 **-I<level>/<pen>**。其中 **<level>** 除了可以取 1 至 10 之外, 还可以取：

- **a:** 所有河流和运河, 即包含 0-10 等级的所有河流
- **A:** 除了河流湖之外的所有河流和运河, 即包含 1-10 等级的河流
- **r:** 所有永久河流, 即 0-4 等级
- **R:** 除了河流湖之外的永久河流, 即 1-4 等级
- **i:** 所有间歇性河流, 即 5-7 等级
- **c:** 所有运河, 即 8-10 等级

该选项可以重复多次使用, 为不同等级的河流设置不同的画笔属性。

国界线

国界线进一步细分为三个等级

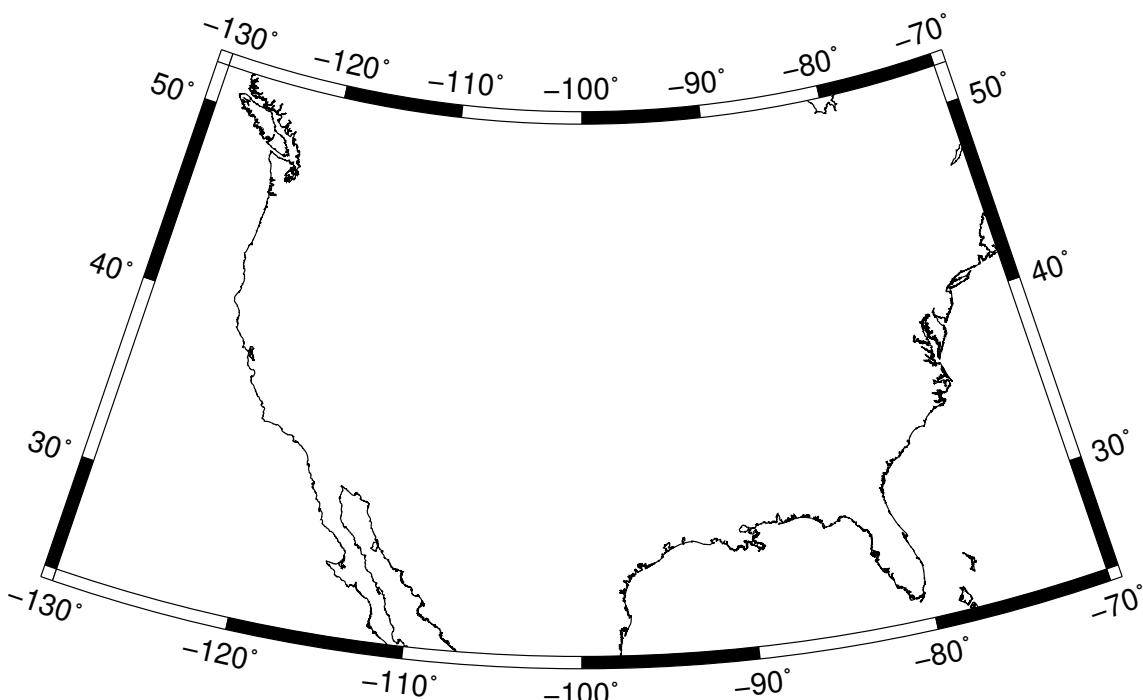
- 1: 国界
- 2: 美国州界
- 3: 海洋边界

pscoast 模块的 `-N` 选项可以用于绘制不同等级的国界线, 其基本语法为 `-N<level>/<pen>`。其中 `<level>` 可以取 1 至 3, 也可以取 `a` (表示所有边界)。该选项可以重复多次使用, 为不同等级的国界设置不同的画笔属性。

11.2.3 使用示例

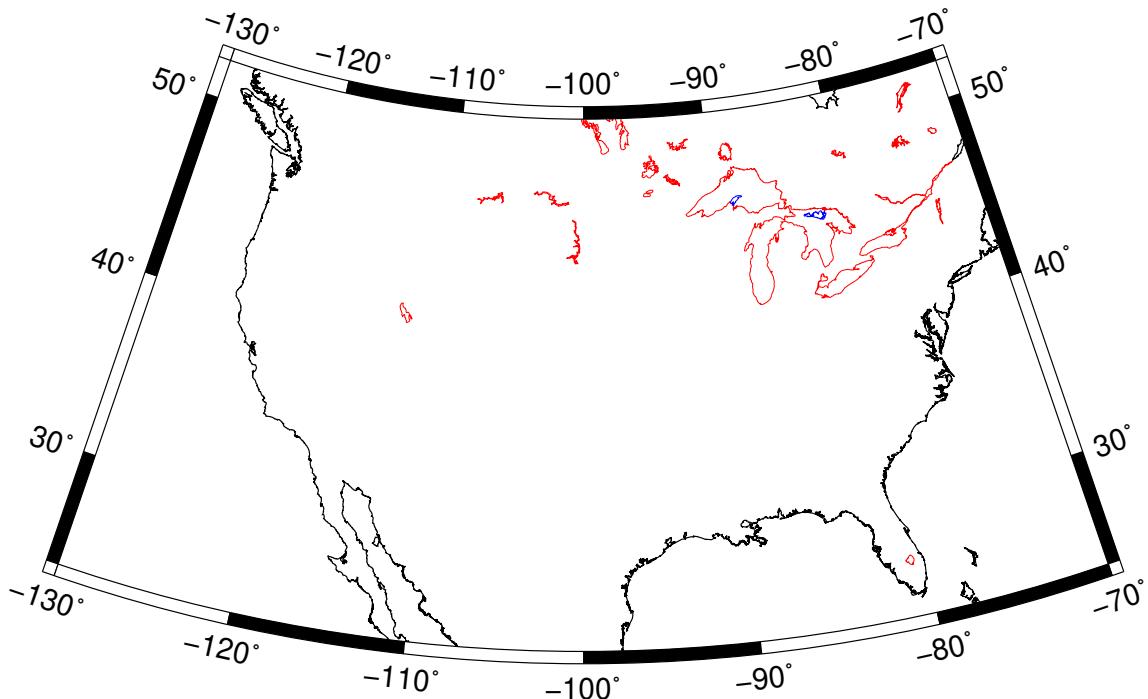
绘制 1 级海岸线:

```
gmt pscoast -R-130/-70/24/52 -JL-100/35/33/45/15c -Ba -Dh -A1000 -W1/0.5p -P > map.ps
```



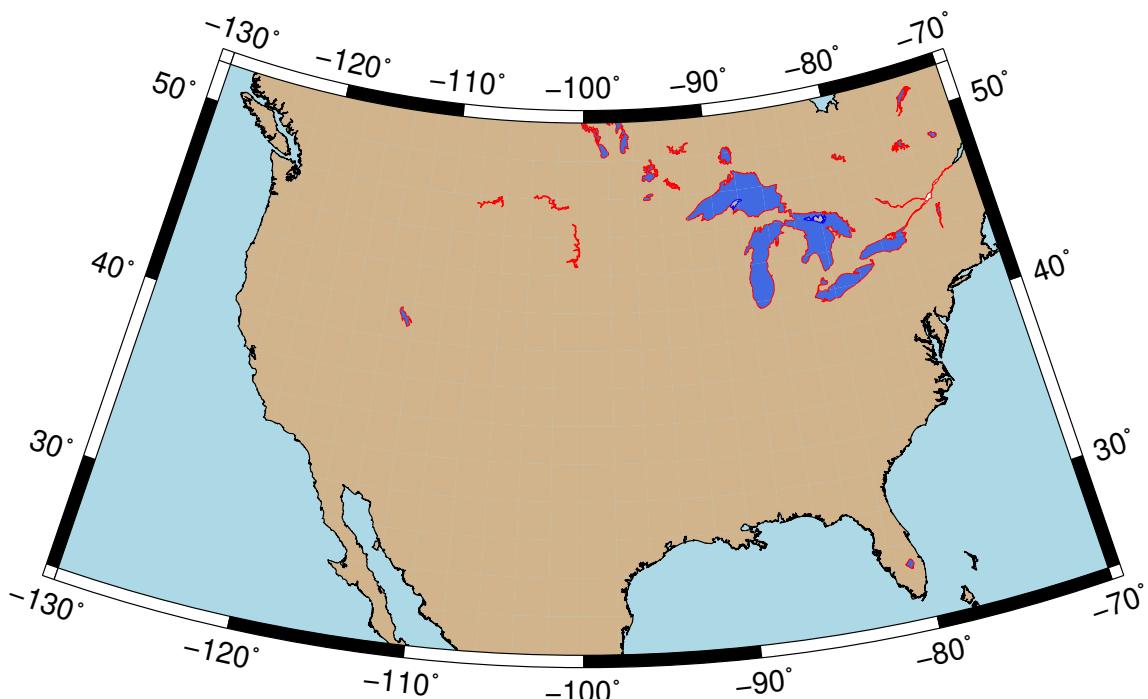
同时绘制 1-3 级海岸线, 黑色的为 1 级海岸线, 红色的为 2 级湖泊线 (图中的大面积红色区域为五大湖), 蓝色的为 3 级岛屿线 (即五大湖内部的岛屿):

```
gmt pscoast -R-130/-70/24/52 -JL-100/35/33/45/15c -Ba -Dh -A1000 \
-W1/0.5p -W2/0.3p,red -W3/0.2p,blue -P > map.ps
```



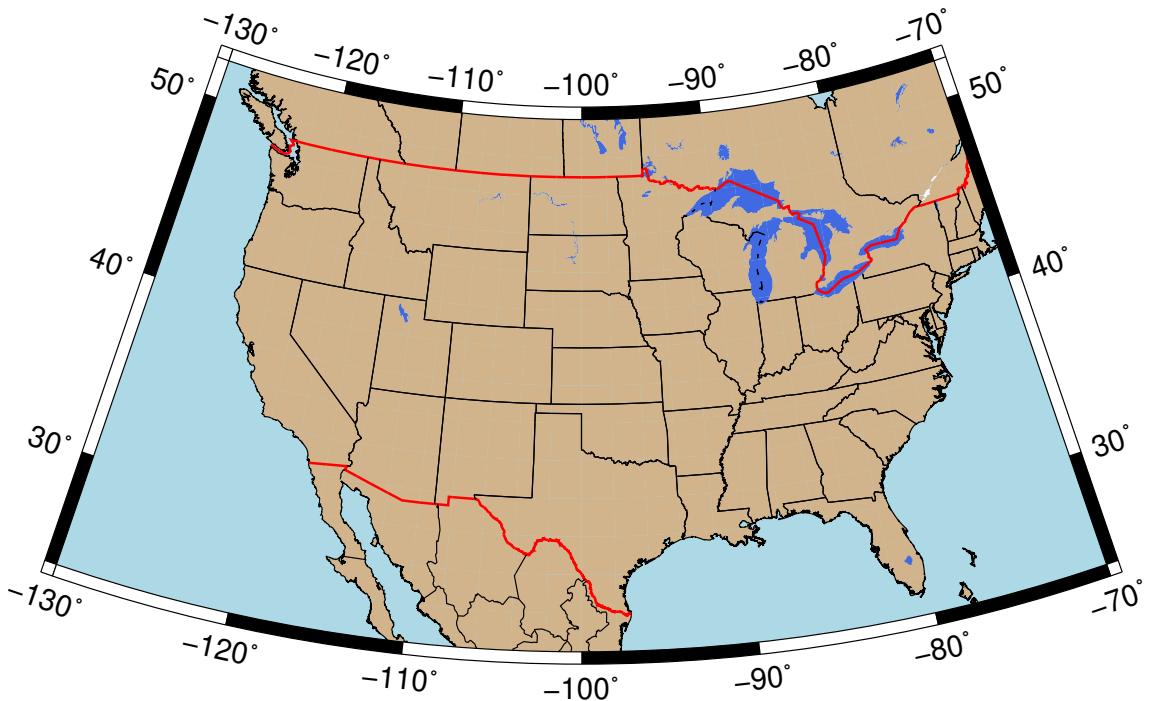
绘制 1-3 级海岸线，并为陆地、还有、湖泊填充不同的颜色：

```
gmt pscoast -R-130/-70/24/52 -JL-100/35/33/45/15c -Ba -Dh -A1000 \
-W1/0.5p -W2/0.3p,red -W3/0.2p,blue \
-Gtan -Slightblue -Cl/royalblue -P > map.ps
```



绘制海岸线、国界和美国州界：

```
gmt pscoast -R-130/-70/24/52 -JL-100/35/33/45/15c -Ba -Dh -A1000 \
-W1/0.5p -N1/thick,red -N2/thinner \
-Gtan -Slightblue -Cl/royalblue -P > map.ps
```



11.2.4 备注

GSHHG 提供的中国国界数据不符合中国的领土主张，在正式刊物中发表使用此类国界数据的图件时都可能存在问题。

11.3 DCW: 世界数字图表

DCW 数据主页: <http://www.soest.hawaii.edu/wessel/dcw/>

DCW, 全称为 Digital Chart of the World, 即世界数字图表。GMT 提供的 DCW 数据是在原始 DCW 数据的基础上修改得到的，其中包含了如下行政边界数据：

1. 七大洲的洲界
2. 全球 250 个国家或地区的边界
3. 8 个大国的省界/州界

GMT 的 `pscoast` 模块可以直接绘制 DCW 数据中提供的行政边界数据，也可以使用 `pscoast` 的 `-M` 选项将边界数据导出为纯文本文件供其他程序使用。

GMT 提供的 DCW 数据默认位于 GMT 安装目录下的 `share/dcw` 下，其中主要包含了三个文件：

- `dcw-gmt.nc`: netCDF 格式的 DCW 数据
- `dcw-countries.txt`: 辅助文件，内含国家代码

- dcw-states.txt: 辅助文件, 内含省界代码

11.3.1 区域代码

为了绘制某个特定行政区域的边界, 首先需要知道这些行政区域的代码。

洲代码

七大洲都有各自的代码, 其代码分别为:

- AF: 非洲 (Africa)
- AN: 南极洲 (Antarctica)
- AS: 亚洲 (Asia)
- EU: 欧洲 (Europe)
- OC: 大洋洲 (Oceania)
- NA: 北美洲 (North America)
- SA: 南美洲 (South America)

国家代码

每个国家也有各自的编码。国家代码可以从 DCW 的辅助文件 dcw-countries.txt 中查找, 其文件格式为:

洲代码	国家代码	国家名
-----	------	-----

该文件共计 250 个国家。文件内容大致如下:

AS BH Bahrain
AS BN Brunei
AS BT Bhutan
AS CN China
AS CX Christmas Island
AS GE Georgia
AS HK Hong Kong
AS HM Heard Island and McDonald Islands
AS ID Indonesia
AS IL Israel
AS IN India

其中可以看到, 中国的国家代码为 CN。

省/州代码

目前有如下八个国家的省界/州界数据:

- AR: 阿根廷
- AU: 澳大利亚
- BR: 巴西
- CA: 加拿大
- US: 美国

- CN: 中国
- IN: 印度
- RU: 俄罗斯

省代码可以从 DCW 辅助文件 `dcw-states.txt` 中查找到, 其文件格式为:

国家代码	省代码	省名
------	-----	----

以中国的数据为例, 其包括全部 34 个省级行政区域: 23 个省(包括台湾省), 5 个自治区, 4 个直辖市, 以及香港, 澳门 2 个特别行政区。中国的省代码是数字, 和中国居民身份证号码相同:

CN 11 Beijing
CN 50 Chongqing
CN 31 Shanghai
CN 12 Tianjin
CN 34 Anhui
CN 35 Fujian
CN 62 Gansu
CN 44 Guangdong
CN 52 Guizhou
CN 46 Hainan
CN 13 Hebei
CN 23 Heilongjiang
CN 41 Henan
CN 42 Hubei
CN 43 Hunan
CN 32 Jiangsu
CN 36 Jiangxi
CN 22 Jilin
CN 21 Liaoning
CN 63 Qinghai
CN 61 Shaanxi
CN 37 Shandong
CN 14 Shanxi
CN 51 Sichuan
CN 71 Taiwan
CN 53 Yunnan
CN 33 Zhejiang
CN 45 Guangxi
CN 15 Nei Mongol
CN 64 Ningxia
CN 65 Xinjiang
CN 54 Xizang
CN 91 Xianggang (Hong Kong)
CN 92 Aomen (Macao)

11.3.2 使用说明

洲代码、国家代码和省代码都是字母, GMT 通过如下方式区分:

- 在洲代码前加上 = 号表示某个大洲, 比如 =AS 表示亚洲
- 国家代码不需要做任何处理格式, 比如 GB 表示英国

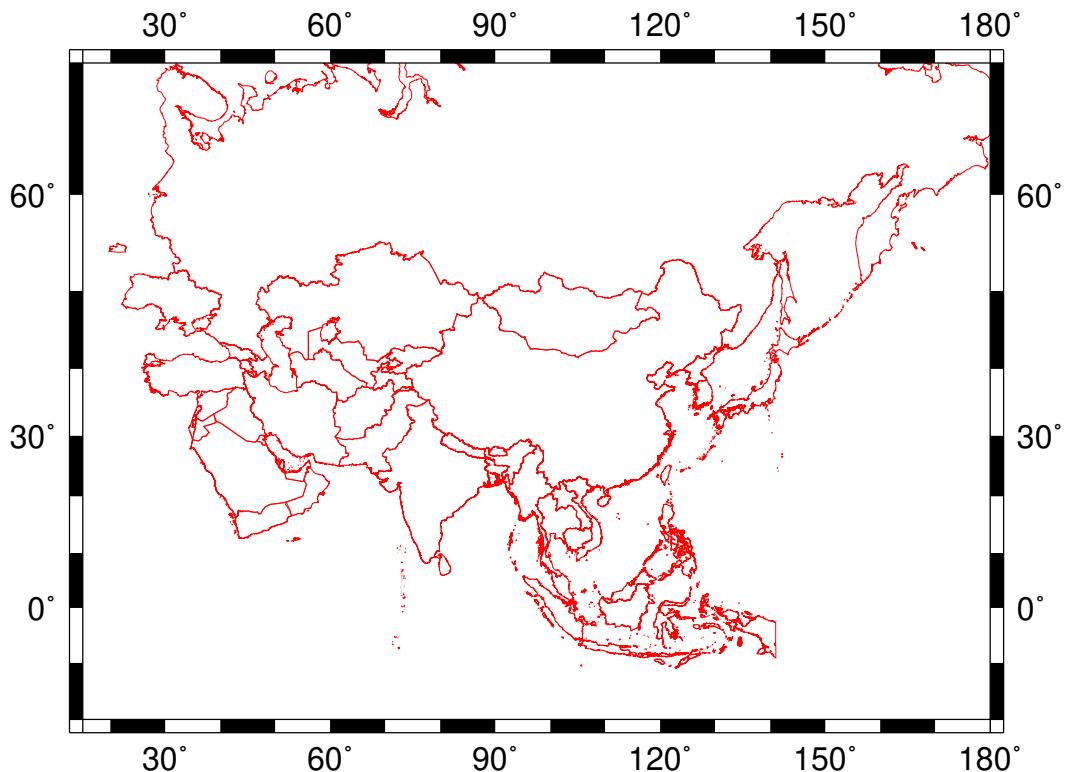
- 省代码的格式为 `country.state`, 即必须在省代码前加上国家代码才可以, 比如 `US.TX` 表示美国 Texas 州

11.3.3 使用示例

绘制洲界

绘制全部亚洲国家的边界:

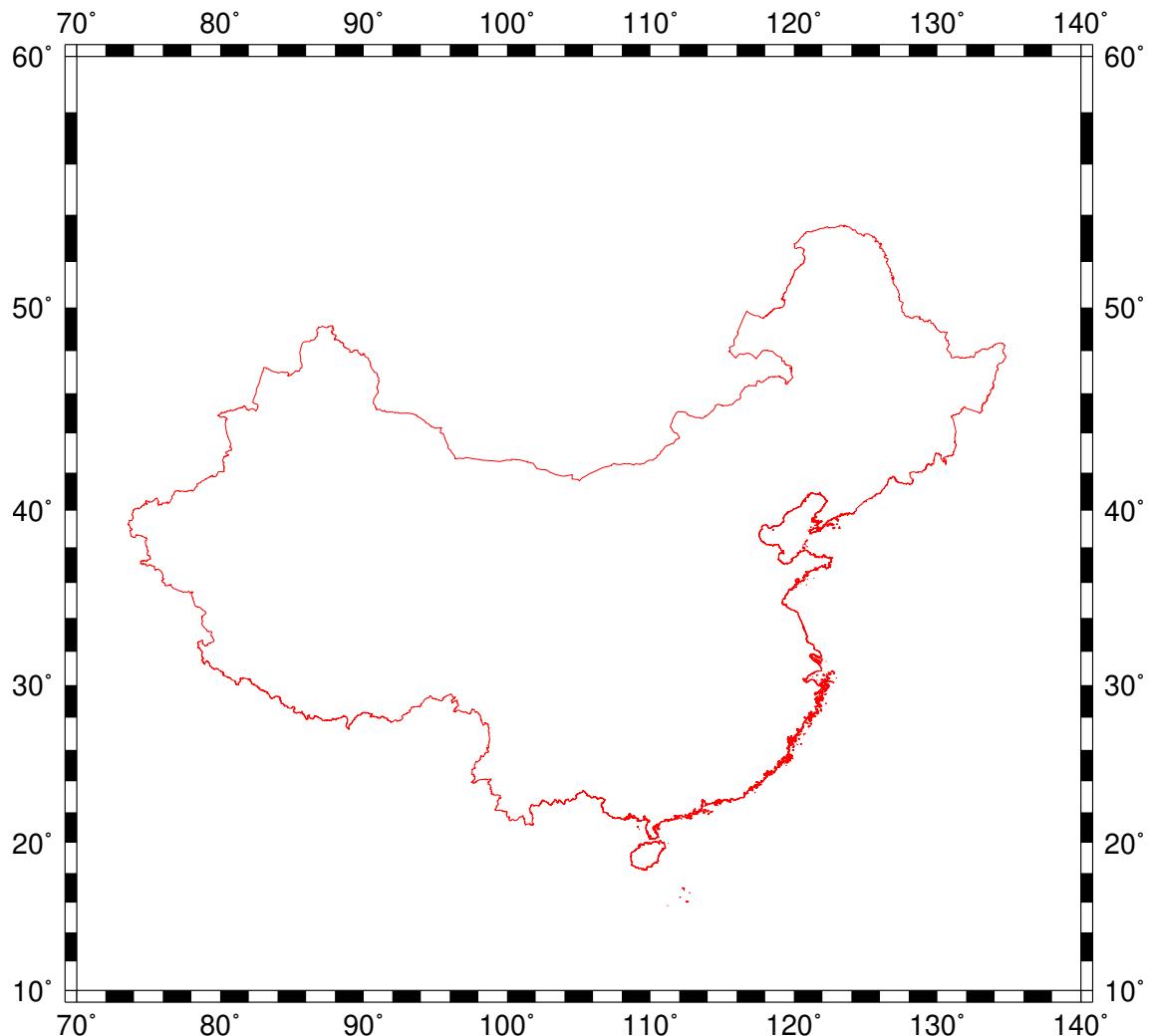
```
gmt pscoast -R15/180/-20/70 -JM12c -Baf -E=AS+p0.25p,red > dataset_dcw_01.ps
```



绘制国界

绘制中国大陆边界(不含台湾、香港、澳门):

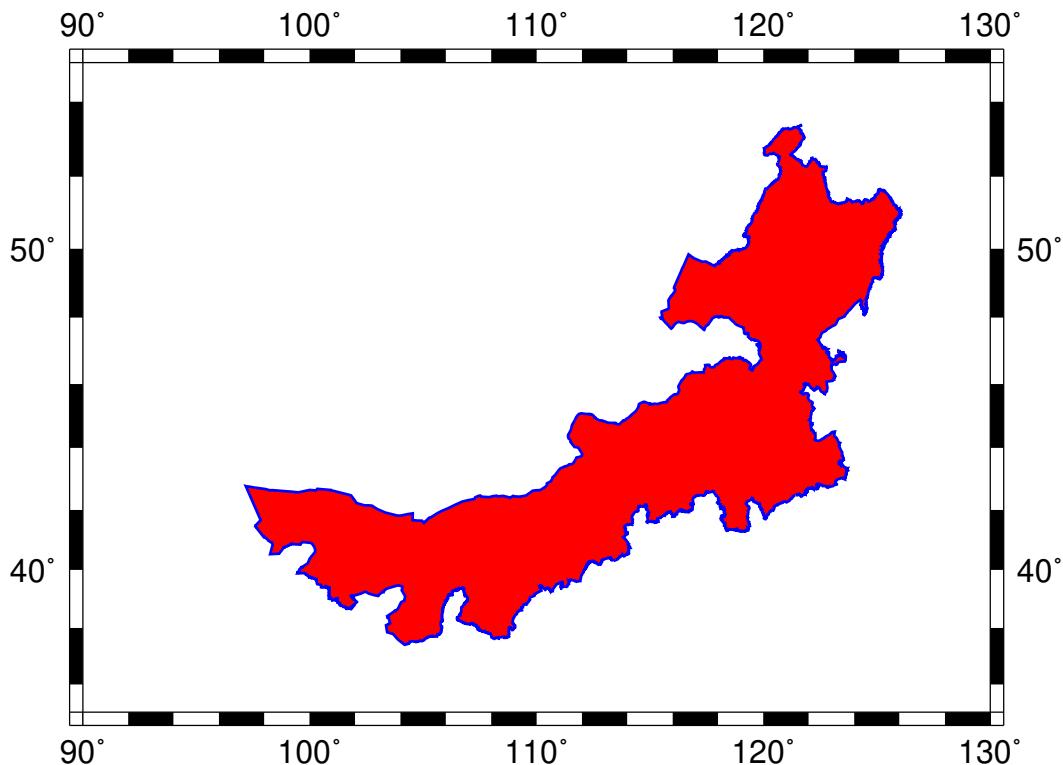
```
gmt pscoast -JM15c -Baf -ECN+p0.25p,red -R70/140/10/60 > dataset_dcw_02.ps
```



绘制省界

绘制内蒙古，并设置边界颜色和填充颜色：

```
gmt pscoast -JM12c -Baf -ECN.15+p1p,blue+gred -R90/130/35/55 > dataset_dcw_03.ps
```



导出省界数据

导出内蒙古的边界数据:

```
gmt pscoast -ECN.15 -M > neimenggu.dat
```

这里只需要使用 `-M` 选项即可。

11.3.4 备注

DCW 数据提供的中国国界数据不符合中国的领土主张，在正式刊物中发表使用此类国界数据的图件时都可能存在问题。此处展示的国界仅用于展示如何使用数据。

11.4 earth_relief: 全球地形起伏数据

11.4.1 地形起伏数据简介

GMT 提供了从 15 弧秒到 60 弧分的多种不同精度的全球地形起伏网格数据供用户使用。

地形数据名称	精度	大小	说明
earth_relief_60m	60 弧分	112 Kb	ETOPO1 高斯球面滤波得到
earth_relief_30m	30 弧分	377 Kb	ETOPO1 高斯球面滤波得到
earth_relief_20m	20 弧分	783 Kb	ETOPO1 高斯球面滤波得到
earth_relief_15m	20 弧分	1.4 Mb	ETOPO1 高斯球面滤波得到
earth_relief_10m	10 弧分	2.9 Mb	ETOPO1 高斯球面滤波得到
earth_relief_06m	6 弧分	7.5 Mb	ETOPO1 高斯球面滤波得到
earth_relief_05m	5 弧分	11 Mb	ETOPO1 高斯球面滤波得到
earth_relief_04m	4 弧分	16 Mb	ETOPO1 高斯球面滤波得到
earth_relief_03m	3 弧分	28 Mb	ETOPO1 高斯球面滤波得到
earth_relief_02m	2 弧分	58 Mb	ETOPO2v2 冰层表面
earth_relief_01m	1 弧分	214 Mb	ETOPO1 冰层表面
earth_relief_30s	30 弧秒	778 Mb	SRTM30+
earth_relief_15s	15 弧秒	2.6 Gb	SRTM15+

这些数据保存在 GMT 的 **FTP 服务器** 上 (国内镜像 <http://mirrors.ustc.edu.cn/gmt/data/>)。当用户第一次使用某个精度的地形起伏数据时, GMT 会自动从服务器上下载该数据文件, 并保存到 GMT 的缓存文件夹下 (由 *DIR_CACHE* 控制, 默认为 `~/.gmt/` 目录), 然后再读取该文件。以后再使用该数据时, GMT 会自动从缓存文件夹下读取该数据文件, 而无需再次从服务器下载。

11.4.2 使用方法

当需要使用地形数据时, 可以直接通过 `@earth_relief_<res>` 的形式调用这些地形起伏数据, 其中 `<res>` 表示网格文件的精度。如果命令中使用了 `-R` 选项, 则只会读取该区域内的地学起伏数据。

使用示例:

```
gmt grdinfo @earth_relief_60m
gmt grdimage -Rg -JH15c @earth_relief_05m -Cglobe > map.ps
```

11.4.3 缓存空间问题

你可以使用多种方式来控制你的缓存目录所占用的空间大小:

1. 通过参数 `GMT_DATA_URL_LIMIT` 设置缓存目录空间大小的上限, 默认无限制;
2. 可以通过 `gmt clear cache` 命令清理整个缓存目录

11.4.4 技术细节

如上表所示, 3 弧分及更低分辨率的全球地形数据均是 NOAA ETOPO1 全球地形起伏网格数据的衍生产品。GMT 利用球面高斯滤波对其进行重采样以防止混叠现象。使用:

```
gmt grdinfo @earth_relief_60m
```

可以在输出中看到生成这些网格文件所使用的滤波命令。

2 弧分的数据直接使用了 NOAA 提供的 ETOPO2v2 文件 (冰层表面版本)。30 弧秒和 15 弧秒的数据来自于 SRTM30+ 和 SRTM15+。

所有的网格文件都是网格线配准的。网格文件采用了更高效的文件格式，使得其文件大小远小于原始文件的大小，且完全保持数据精度。

11.4.5 数据来源及引用

1. ETOPO2v2: <https://dx.doi.org/10.7289/V5J1012Q>.
2. ETOPO1: Amante, C., and B. W. Eakins (2008), ETOPO1 1 arc-minute global relief model: Procedures, data sources and analysis, National Geophysical Data Center, Boulder, CO. [原始数据下载链接](#)
3. SRTM30+: Becker, J. J., et al. (2009), Global Bathymetry and Elevation Data at 30 Arc Seconds Resolution: SRTM30_PLUS, Marine Geodesy, 32, 355-371. [原始数据下载链接](#)
4. SRTM15+: Olson, C. L., J. J. Becker, and D. T. Sandwell (2014), A new global bathymetry map at 15 arcsecond resolution for resolving seafloor fabric: SRTM15_PLUS, in Eos Trans. AGU, edited, pp. Abstract OS34A-03 [原始数据下载链接](#)

11.5 GADM: 全球行政区划数据库

GADM 主页: <https://gadm.org/>

GADM, 全称 Database of Global Administrative Areas, 是一个高精度的全球行政区划数据库。其包含了全球所有国家和地区的国界、省界、市界、区界等多个级别的行政区划边界数据。

11.5.1 数据下载

GADM 提供了两种下载方式：

1. 下载全球所有国家和地区的所有数据 https://gadm.org/download_world.html
2. 按国家下载 https://gadm.org/download_country_v3.html

推荐只下载自己需要的国家的数据。

需要说明的是, GADM 中对 country 的定义为“any entity with an ISO country code”。因而如果想要下载完整的中国数据, 实际上需要下载 China、Hong Kong、Macao 和 Taiwan 四个数据。

11.5.2 数据格式及转换

对于每个数据, GADM 提供了 5 种不同的格式:

- Geopackage: 可以被 GDAL/OGR、ArcGIS、QGIS 等软件读取
- Shapefile: 可直接用于 ArcGIS 等软件
- KMZ: 可直接在 Google Earth 中打开
- R (sp): 可直接用于 R 语言绘图
- R (sf): 可直接用于 R 语言绘图

如果在安装 GMT 时, GMT 已经正确链接了 GDAL 库, 则 Shapefile 格式的数据可以直接用于绘图。实际绘图时, 可能只想要一小部分数据(比如某个省的省界), 这种情况下, 则需要将数据转换成纯文本文件, 以方便从数据中提取出需要的部分。

GDAL 的 `ogr2ogr` 可以实现多种地理数据格式之间的互相转换。该软件的安装及介绍见 [GDAL/OGR: 地理空间数据格式转换神器](#)。本文使用的是 GDAL 2.4.0, 其他版本的 GDAL 可能用法略有不同。

Geopackage 转 GMT

以 China 数据为例, 解压得到文件 `gadm36_CHN.gpkg`。使用如下命令查看文件的信息:

```
$ ogrinfo gadm36_CHN.gpkg
INFO: Open of `gadm36_CHN.gpkg'
      using driver `GPKG' successful.
1: gadm36_CHN_0 (Multi Polygon)
2: gadm36_CHN_1 (Multi Polygon)
3: gadm36_CHN_2 (Multi Polygon)
4: gadm36_CHN_3 (Multi Polygon)
```

可以看到 Geopackage 文件中包含了四个文件, 使用如下命令(注意其中的一对单引号)将其转换为 GMT 可识别的格式:

```
ogr2ogr -f OGR_GMT '' gadm36_CHN.gpkg gadm36_CHN_0
ogr2ogr -f OGR_GMT '' gadm36_CHN.gpkg gadm36_CHN_1
ogr2ogr -f OGR_GMT '' gadm36_CHN.gpkg gadm36_CHN_2
ogr2ogr -f OGR_GMT '' gadm36_CHN.gpkg gadm36_CHN_3
```

同理, 对 Hong Kong、Macao 和 Taiwan 的数据做同样的处理即可。最终得到以 `.gmt` 结尾的数据 12 个, 其中 CHN 四个、HKG 两个、MAC 三个、TWN 三个。

Shapefile 转 GMT

以 China 数据为例, 将下载的 ZIP 压缩包解压会得到一堆文件, 其中 `gadm36_CHN_[0123].shp` 是真正需要的 4 个 Shapefile 的数据文件。

使用如下命令即可将 Shapefile 转换为 GMT 可识别的格式:

```
ogr2ogr -f OGR_GMT gadm36_CHN_0.gmt gadm36_CHN_0.shp
ogr2ogr -f OGR_GMT gadm36_CHN_1.gmt gadm36_CHN_1.shp
ogr2ogr -f OGR_GMT gadm36_CHN_2.gmt gadm36_CHN_2.shp
ogr2ogr -f OGR_GMT gadm36_CHN_3.gmt gadm36_CHN_3.shp
```

对于 Hong Kong、Macao、Taiwan 的数据做类似操作。最终得到以 .gmt 结尾的数据 12 个，其中 CHN 四个、HKG 两个、MAC 三个、TWN 三个。

11.5.3 数据分级

提取得到的数据文件的文件名类似 gadm36_CHN_0.gmt，其中 CHN 为国家/地区代码，0 表示行政等级。

以中国数据为例，其数据包含了四个等级：

- 0 级：即国界
- 1 级：即省界
- 2 级：即市界
- 3 级：即区界

对于 Hong Kong 而言，则只包含 0 级和 1 级边界。

11.5.4 使用示例

中国全图

绘制中国全图需要前面提取出的四个 0 级数据。

```
#!/bin/bash
PS=gadm_level0.ps

gmt psxy -JM15c -R72/136/15/54 -T -K -P > $PS
gmt psxy -J -R gadm36_CHN_0.gmt -K -O >> $PS
gmt psxy -J -R gadm36_HKG_0.gmt -K -O >> $PS
gmt psxy -J -R gadm36_MAC_0.gmt -K -O >> $PS
gmt psxy -J -R gadm36_TWN_0.gmt -K -O >> $PS
gmt psxy -J -R -T -O >> $PS
```

绘图效果如下：



1 级行政区划/省界

代码与上面的代码几乎一样，此处使用了中国的一级数据和香港、台湾、澳门的 0 级数据。

```
#!/bin/bash
PS=gadm_level1.ps

gmt psxy -JM15c -R72/136/15/54 -T -K -P > $PS
gmt psxy -J -R gadm36_CHN_1.gmt -K -O >> $PS
gmt psxy -J -R gadm36_HKG_0.gmt -K -O >> $PS
gmt psxy -J -R gadm36_MAC_0.gmt -K -O >> $PS
gmt psxy -J -R gadm36_TWN_0.gmt -K -O >> $PS
gmt psxy -J -R -T -O >> $PS
```

绘图效果如下：



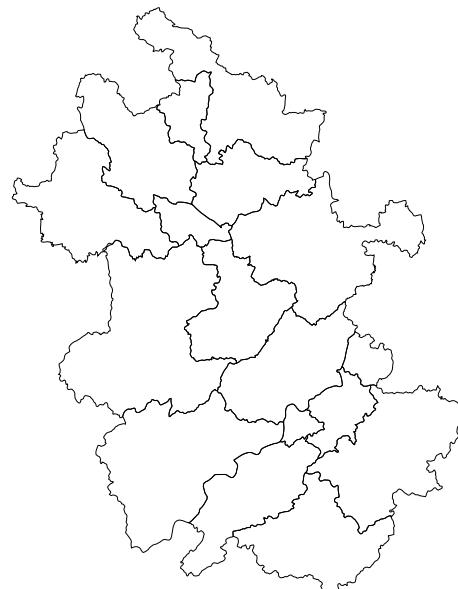
此处绘制了全国所有省的省界数据。如果只想要绘制某个省的省界数据，可以用文本编辑器打开 CHN 的一级数据文件，在注释行中有清晰地标记出每段数据是哪个省的边界，因而可以很方便地提取出来。

11.5.5 2 级行政区划/市界

2 级数据中包含了全国所有的市级边界。此处以安徽省为例，用文本编辑器打开 gadm36_CHN_2.gmt，从中提取安徽相关的数据保存到文件 gadm36_CHN_Anhui_2.gmt 中，绘图效果如下：

```
gmt psxy -JM10c -R114/120/29/35 gadm36_CHN_Anhui_2.gmt -P > gadm_level2.ps
```

绘图效果如下：



11.5.6 许可协议

GADM 的[许可协议](#) 表明该数据可以免费用于学术和非商业用途，可以利用该数据绘制学术出版物的地图，但禁止重新分发或商业用途。

11.5.7 备注

GADM 提供的中国国界数据不一定符合中国的领土主张，省界、市界、区界等数据也不一定能够是最新的版本，在正式刊物中发表使用此类数据的图件时需要谨慎。

第 12 章 模块手册

主程序/脚本

- *gmt* : GMT 主程序
- *isogmt* : 在“隔离模式”下运行 GMT 命令或脚本
- *gmt5syntax* : 将旧的 GMT 脚本修改成新的 *gmt <module>* 语法
- *gmt-config* : 返回 GMT 动态函数库的基本信息

参数设置

- *gmtset* : 修改单个或多个 GMT 参数的值
- *gmtget* : 列出单个或多个 GMT 参数的当前值
- *gmtdefaults* : 列出所有 GMT 参数的当前值

信息提取

- *grdinfo* : 从网格文件中提取基本信息
- *gmtinfo* : 从表数据中提取信息
- *gmtwhich* : 返回指定文件的完整路径

格式转换

- *psconvert* : 将 GMT 生成的 PS 文件转换为其他图片格式
- *grdconvert* : 在不同的网格格式之间互相转换
- *xyz2grd* : 将 XYZ 数据或 Z 数据转换成网格文件
- *grd2xyz* : 将网格文件转换成表数据
- *kml2gmt* : 将 Google Earth 的 KML 文件转换为 GMT 表数据

绘制 1D 数据

- *pssolar* : 计算或/和绘制晨昏线以及民用、航海用以及天文用曙暮光区域
- *psbasemap* : 绘制底图
- *psclip* : 打开或关闭多边形裁剪路径
- *pshistogram* : 统计并绘制直方图
- *pstext* : 在图上写文本
- *pscoast* : 在地图上绘制海岸线、河流、国界线
- *pslegend* : 绘制图例
- *psrose* : 绘制极坐标下的直方图 (sector 图、rose 图或 windrose 图)
- *psxy* : 在图上绘制线段、多边形和符号
- *pssac* : 在地图上绘制 SAC 格式的地震波形数据
- *psmeca* : 在地图上绘制震源机制解

- *pspolar* : 在震源球上绘制台站极性
- *psvelo* : 在地图上绘制速度矢量、十字线、楔形图
- *pscoupe* : 绘制震源机制解的剖面图
- *psternary* : 绘制三角图解

绘制 2D 数据

- *gmtlogo* : 在图上绘制 GMT 图形 logo
- *psimage* : 将 EPS 或光栅图片放在图上
- *psscale* : 在图上绘制灰色或彩色色标
- *grdvector* : 根据两个网格文件绘制矢量场
- *grdimage* : 在图上绘制网格数据

1D 数据处理

- *gmtselect* : 根据多个准则筛选数据
- *fitcircle* : 拟合球面上数据点的平均位置及圆弧
- *gmtsimplify* : 使用 Douglas-Peucker 算法对线段做简化
- *filter1d* : 对 1D 表数据做时间域滤波
- *gmtconnect* : 将端点接近的线段连接起来
- *project* : 将数据点投影到线或大圆路径上, 生成测线, 坐标转换
- *sample1d* : 对 1D 表数据进行重采样
- *spectrum1d*: 计算一个时间序列的自功率谱, 或两个时间序列的互功率谱

2D 数据处理

- *grdclip* : 对网格文件的 Z 值做裁剪
- *grdconvert* : 在不同的网格格式之间互相转换
- *grdcut* : 从一个网格文件中裁剪出一个矩形子区域生成新的网格文件
- *grredit* : 修改网格文件的头段或内容
- *grdlandmask* : 根据海岸线数据创建陆地-海洋的 mask 网格文件
- *grdinfo* : 从网格文件中提取基本信息
- *grdpaste* : 将两个网格沿着其共同边界拼接成一个文件
- *grdraster* : 从二进制数据中提取子区域并保存为 GMT 网格文件
- *grdsample* : 对网格文件做重采样
- *grdvolume* : 计算网格数据中某个等值线所包围的表面积和体积
- *grd2xyz* : 将网格文件转换成表数据
- *grd2rgb* : 将网格文件、原始 RGB 文件、SUN 光栅文件转换成 R/G/B 三个网格文件
- *grdblend* : 将多个部分重叠的网格文件合并成一个网格文件
- *grdtrend* : 拟合网格的趋势面并计算残差
- *grdproject* : 对网格数据做地图变换和逆变换
- *grdmask* : 根据多边形数据或点数据创建 mask 网格文件

其他

1. *grdtrack*
2. *makecpt*

-
- 3. *grd2cpt*
 - 4. *gmtconvert*
 - 5. *trend1d*
 - 6. *grdcontour*
 - 7. *psmask*
 - 8. *grdview*
 - 9. *grdgradient*
 - 10. *grdmath*
 - 11. *gmt2kml*
 - 12. *trend2d*
 - 13. *mapproject*
 - 14. *surface*
 - 15. *grdfilter*
 - 16. *grdffft*
 - 17. *splitxyz*
 - 18. *gmtmath*
 - 19. *blockmean*
 - 20. *blockmedian*
 - 21. *blockmode*
 - 22. *psxyz*
 - 23. *pscontour*
 - 24. *pssegy*
 - 25. *pssegyz*
 - 26. *pswiggle*
 - 27. *grdgravmag3d*
 - 28. *grdhisteq*
 - 29. *gmtpmodeler*
 - 30. *grdpmodeler*
 - 31. *grdrredpol*
 - 32. *grdrotater*
 - 33. *dimfilter*
 - 34. *triangulate*
 - 35. *nearneighbor*
 - 36. *mgd77convert*
 - 37. *mgd77header*
 - 38. *mgd77info*
 - 39. *mgd77list*
 - 40. *mgd77magref*
 - 41. *mgd77manage*
 - 42. *mgd77path*
 - 43. *mgd77sniffer*
 - 44. *mgd77track*
 - 45. *x2sys_binlist*
 - 46. *x2sys_cross*

47. *x2sys_datalist*
48. *x2sys_get*
49. *x2sys_init*
50. *x2sys_list*
51. *x2sys_merge*
52. *x2sys_put*
53. *x2sys_report*
54. *x2sys_solve*
55. *gmtswitch*
56. *backtracker*
57. *gmtgravmag3d*
58. *gmtspatial*
59. *gmtvector*
60. *gravfft*
61. *grdseamount*
62. *gpsgridder*
63. *grdspotter*
64. *greenspline*
65. *gshhg*
66. *hotspotter*
67. *img2google*
68. *img2 grd*
69. *originator*
70. *rotconverter*
71. *rotsmooth*
72. *seg y2 grd*
73. *sph2 grd*
74. *sphdistance*
75. *sphinterpolate*
76. *sphtriangulate*

12.1 backtracker

[官方文档](#) [backtracker](#)
[简介](#)

12.2 blockmean

[官方文档](#) [blockmean](#)
[简介](#)

12.3 blockmedian

[官方文档](#) [blockmedian](#)
[简介](#)

12.4 blockmode

[官方文档 blockmode](#)

简介

12.5 dimfilter

[官方文档 dimfilter](#)

12.6 filter1d

[官方文档 filter1d](#)

简介 对 1D 表数据做时间域滤波

该模块用于对多列时间序列数据做时间域滤波。用户需要指定哪一列数据代表时间（即自变量）。若输入的时间序列是等间隔且无间断或 outliers 则滤波速度较快。对于有间断的不等间隔数据，需要使用 -L、-Q、-S 选项。

12.6.1 必选选项

-F<type><width> [<mode>] [+h] 设置滤波器类型

滤波器分为两大类，卷积滤波器和非卷积滤波器。<type> 用于指定滤波器类型，<width> 指定滤波器宽度（单位与时间数据相同）。

对于卷积滤波器，<type> 可以取：

- b Boxcar: All weights are equal
- c Cosine Arch: Weights follow a cosine arch curve
- g Gaussian: Weights are given by the Gaussian function
- f Custom: Instead of *width* give name of a one-column file with your own weight coefficients.

对于非卷积滤波器，<type> 可以取：

- m Median: Returns median value
- p Maximum likelihood probability (a mode estimator): Return modal value. If more than one mode is found we return their average value. Append - or + to the filter width if you rather want to return the smallest or largest of the modal values.
- l Lower: Return the minimum of all values.
- L Lower: Return minimum of all positive values only.
- u Upper: Return maximum of all values.
- U Upper: Return maximum of all negative values only.

大写的 B|C|G|M|P|F will use robust filter versions, i.e., replace outliers (2.5 L1 scale off median) with median during filtering.

In the case of **L|U** it is possible that no data passes the initial sign test; in that case the filter will return 0.0.

该模块默认对数据进行低通滤波，加上 **+h** 选项则对数据进行高通滤波。

12.6.2 选项

-D<increment> 当输入的时间序列是不等间隔采样时，需要使用该选项。
<increment> 为输出数据的分辨率，所有横坐标（时间）都会被 rounded off 到 **<increment>** 的整数倍。当然，也可以使用 [sample1d](#) 对时间序列做重采样。

-E 输出时间序列的首尾端数据。默认情况下，首尾两端都会丢失半滤波器宽度的数据点

-I<ignore_val> 若输出数据等于 **<ignore_val>**，则设置其值为 NaN

-L<lack_width> 检查数据间断。若输入数据存在超过 **<lack_width>** 的间断，则该数据点不输出值。

-N<t_col> 指定哪一列数据包含自变量（即时间）。默认值为 0，即第一列。

-Q<q_factor> 检查输出值的质量因子。

<q_factor> 的取值为 0 到 1，若某点的卷积的平均权重小于 **<q_factor>**，则该点的输出会被压制。

-S<symmetry_factor> 检查数据关于时间窗中心的对称性。

<symmetry_factor> 的取值范围为 0 到 1。若 $((\text{abs}(\text{n_left} - \text{n_right})) / (\text{n_left} + \text{n_right})) > \text{factor}$ ，则该点不输出值。

-T<t_min>/<t_max>/<t_inc>[+n] 限定时间范围为 **<t_min>** 到 **<t_max>**，时间间隔为 **<t_inc>**。
+n 表示 **<t_inc>** 代表的是等间隔的时间点数目。

12.6.3 示例

数据 **cruise.gmt** 中包含了撕裂数据，分别是重力、磁场、地形和距离（单位为 m），现对其做 10 km Gaussian 滤波，去除 outliers，并在 0 到 100 km 内每 2 km 输出一个滤波后的数据点：

```
gmt filter1d cruise.gmt -T0/1.0e5/2000 -FG10000 -N3 -V > filtered_cruise.gmt
```

```
gmt filter1d v3312.dt -FM50 -T0/100000/25 -L10 -S0.3 > v3312_filt.dt
```

12.7 fitcircle

官方文档 [fitcircle](#)

简介 拟合球面上数据点的平均位置及圆弧

该命令从输入数据的前两列读取经纬度数据，将其转换为单位球面上的笛卡尔三维矢量，然后计算输入坐标数据的平均位置以及可以拟合这些坐标点的大圆路径的 pole。

12.7.1 必选选项

<table> 输入数据

-L<norm>

- -L1 解法 1
- -L2 解法 2
- -L 或 -L3 同时输出解法 1 和解法 2 的结果

在计算输入坐标的均值和大圆弧 pole 时有两种计算方法，分别用 -L1 和 -L2 表示。当数据沿着某个大圆弧且距离接近时，两组解结果相似。若数据较分散，则大圆弧的 pole 要比均值不精确得多。此时可以比较两组结果以作为相互的验证。

The **-L1** solution is so called because it approximates the minimization of the sum of absolute values of cosines of angular distances. This solution finds the mean position as the Fisher average of the data, and the pole position as the Fisher average of the cross-products between the mean and the data. Averaging cross-products gives weight to points in proportion to their distance from the mean, analogous to the “leverage” of distant points in linear regression in the plane.

The **-L2** solution is so called because it approximates the minimization of the sum of squares of cosines of angular distances. It creates a 3 by 3 matrix of sums of squares of components of the data vectors. The eigenvectors of this matrix give the mean and pole locations. This method may be more subject to roundoff errors when there are thousands of data. The pole is given by the eigenvector corresponding to the smallest eigenvalue; it is the least-well represented factor in the data and is not easily estimated by either method.

12.7.2 选项

-Ff|m|n|s|c 控制输出格式。

正常情况下，该命令会将计算结果以较复杂的形式输出。使用 -F 选项，则只返回简单的坐标。-F 后可以加上其他修饰符以指定要返回的坐标：

- f Flat Earth mean location
- m mean location
- n north pole of great circle
- s south pole of great circle
- c pole of small circle and its colatitude, which requires -S

-S[<lat>] 拟合小圆弧而不是大圆弧

The pole will be constrained to lie on the great circle connecting the pole of the best-

fit great circle and the mean location of the data. Optionally append the desired fixed latitude of the small circle [Default will determine the latitude].

12.7.3 示例

如下命令, 用两种计算方法拟合了数据的大圆弧路径和小圆弧路径, 并借助[project](#) 生成路径坐标。

```
#!/bin/bash
# test/fitcircle/circles.sh
#
# Tests gmt fitcircle on spherical data points

ps=fitcircle_ex1.ps

# Fit great circle to red points and small circle to green points
# Plot the best-fitting great and small circles as well as the
# location of the mean locations and pole locations for both
# the -L1 and -L2 options (i.e., -L3).

gmt fitcircle gcircle.txt -L3 > g.txt
gmt fitcircle scircle.txt -L3 -S > s.txt
gpole1=`grep "L1 N Hemisphere" g.txt | gawk '{printf "%s/%s\n", $1, $2}'` 
gpole2=`grep "L2 N Hemisphere" g.txt | gawk '{printf "%s/%s\n", $1, $2}'` 
spole1=`grep "L1 Small Circle Pole" s.txt | gawk '{printf "%s/%s\n", $1, $2}'` 
spole2=`grep "L2 Small Circle Pole" s.txt | gawk '{printf "%s/%s\n", $1, $2}'` 
slat1=`grep "L1 Small Circle" s.txt | gawk '{print 90-$NF}'` 
slat2=`grep "L2 Small Circle" s.txt | gawk '{print 90-$NF}'` 
gmt psxy -Rg -JG-30/40/7i -P -Bg -K gcircle.txt -Sc0.04i -Gred -Xc -Yc > $ps
gmt psxy -R -J -O -K scircle.txt -Sc0.04i -Ggreen >> $ps
gmt project -G1 -T$gpole1 -L-180/180 | gmt psxy -R -J -O -K -W3p >> $ps
gmt project -G1 -T$gpole2 -L-180/180 | gmt psxy -R -J -O -K -W1p,- >> $ps
gmt project -G1/$slat1 -T$spole1 -L-180/180 | gmt psxy -R -J -O -K -W3p >> $ps
gmt project -G1/$slat2 -T$spole2 -L-180/180 | gmt psxy -R -J -O -K -W1p,- >> $ps
grep "Great Circle Pole" g.txt | gmt psxy -R -J -O -K -Sa0.2i -Gred -W0.25p >> $ps
grep "Small Circle Pole" s.txt | gmt psxy -R -J -O -K -Sa0.2i -Ggreen -W0.25p >> $ps
grep "L1 Average" g.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
grep "L2 Average" g.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
grep "L1 Average" s.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
grep "L2 Average" s.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
gmt psxy -R -J -O -T >> $ps
rm g.txt s.txt gmt.*
```

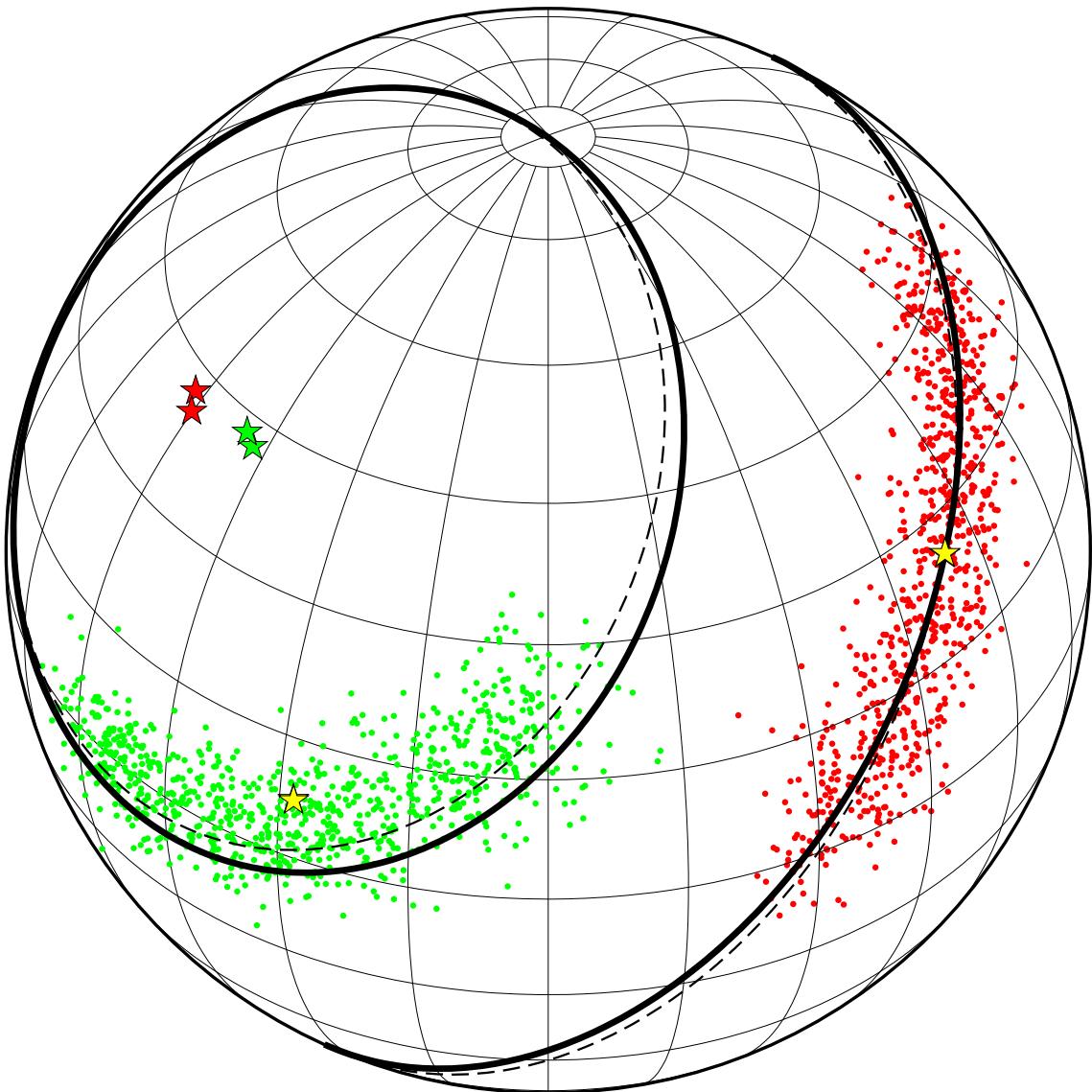


图 12.1: fitcircle 示例

12.8 gmt

官方文档 [gmt](#)

说明 GMT 的主程序，也是 GMT5 中唯一的一个二进制文件，GMT 的所有模块都需要通过该命令调用

GMT 的所有功能通过如下语法调用：

```
gmt <module> <module-options>
```

其中，<module> 是 GMT 模块名，<module-options> 是模块所支持的选项。

比如，想要调用 GMT 的 psbasemap 模块：

```
gmt psbasemap -JX10c/10c -R0/20/0/5 -Bafg > test.ps
```

gmt 还提供了一个特殊的模块, 名为 `clear`

```
gmt clear history|conf|cache|all
```

- `gmt clear history`: 删除当前目录下的 `gmt.history` 文件
- `gmt clear conf`: 删除当前目录下的 `gmt.conf` 文件
- `gmt clear cache`: 删除用户目录 (`~/.gmt`) 下的缓存文件夹
- `gmt clear all`: 清理 `history`、`conf` 和 `cache`

除此之外, 还有一些其他选项可以使用:

- `gmt --help` : 列出 GMT 提供的所有模块名及其功能
- `gmt --version` : 显示 GMT 版本
- `gmt --show-bindir` : 显示 GMT 的 bin 目录
- `gmt --show-datadir` : 显示 GMT 的数据目录, 默认为空
- `gmt --show-sharedir` : 显示 GMT 的 share 目录
- `gmt --show-plugindir` : 显示 GMT 的插件目录
- `gmt --show-modules` : 列出 GMT 的所有模块名
- `gmt --show-cores` : 显示当前计算机可以使用的核数
- `gmt <module> =` : 检测模块 `<module>` 是否存在, 若存在则返回 0, 否则返回非零值

12.9 gmt-config

官方文档 无

说明 返回 GMT 动态函数库的基本信息

`gmt-config` 是位于 GMT 的 bin 目录下的一个 bash 脚本, 主要用于返回 GMT 动态函数库相关的信息, 以供其他程序使用。

其提供了如下选项:

```
$ gmt-config

--help      显示此帮助信息并退出
--bits      库文件是 32 位还是 64 位
--cflags    GMT 的 CFLAGS: ``-I/opt/GMT/include/gmt``
--datadir   GMT 数据目录, 默认为空
--dcw       DCW 数据的位置(可能为空)
--dep-libs  GMT 函数库依赖的其他函数库
--gshhg    GSHHG 数据的位置
--has-fftw  编译过程中是否使用了 FFTW
--has-gdal  编译过程中是否使用了 GDAL
--has-pcre  编译过程中是否使用了 PCRE
--has-lapack 编译过程中是否使用了 LAPACK
--has-openmp 编译过程中是否开启了 OpenMP 支持
```

(下页继续)

(续上页)

```
--includedir include 目录所在位置
--libs      链接 GMT 函数库所需的信息 ``-L/opt/GMT/lib64 -lgmt``
--plugindir GMT 插件目录
--prefix    GMT 安装目录
--version   GMT 版本
```

12.10 gmt2kml

[官方文档 gmt2kml](#)

12.11 gmt5syntax

[官方文档 gmt5syntax](#)

简介 将旧的 GMT 脚本修改成新的 gmt <module> 语法。

gmt5syntax 是一个 Perl 脚本, 作用是在命令前加上 gmt 使得旧版本的语法变成新版本语法。该脚本位于 \${GMTHOME}/share/tools 目录下。

12.11.1 用法

```
gmt5syntax old_script > new_script
```

12.12 gmtconnect

[官方文档 gmtconnect](#)

简介 将端点接近的线段连接起来

该命令会读入一个或多个多段数据, 并检查所有线段的两个端点。若某对端点完全相同或者二者的距离在可容忍的范围内, 则将两段线段连接成一条线段。该进程会一直重复, 直到剩余的端点均不在可容忍的范围内。最终连接得到的线段会被写到终端或特定的输出文件中。

若不清楚该如何确定可容忍范围值, 可以使用 -L 选项, 获得所有端点之间的距离的列表, 通过分析该列表来得到合适的值。

12.12.1 选项

-C[<closed>] 将所有闭合多边形写到文件 <closed> (默认文件名为 gmtconnect_closed.txt) 中并将其他数据段写到标准输出。

使用该选项不会对线段做连接。

-D[<template>] 对于多段数据文件, 将每段数据分别输出到不同的数据文件中。

<template> 是文件名的模板, 该模板中必须包含一个整型参数的格式, 比如 %d 或 %08d, 也可以在整型参数格式前加上字符参数格式 %c, 实际输出时会被 C 或 O 代替, 分别表示 closed 和 open。

默认的模板为 `gmtconnect_segment_%d.txt`。

-L[<linkfile>] 将连接信息写到指定的文件中， 默认文件名为 `gmtconnect_link.txt`。

对于每段数据而言，会写入原始的数据段 ID；对于线段的起始点和终点而言，会报告离得最近的线段的 ID，以及两个线段端点之间的距离。

-Q[<template>] Used with **-D** to a list file with the names of the individual output files. Optionally, append a filename template for the individual file names; this template **may** contain a C format specifier that can format an character (C or O for closed or open, respectively). [Default is `gmtconnect_list.txt`].

-T[<cutoff> [<unit>]] [/<nn_dist>] Specifies the separation tolerance in the data coordinate units [0]; append distance unit (see UNITS). If two lines has endpoints that are closer than this cutoff they will be joined. Optionally, append `/nn_dist` which adds the requirement that a link will only be made if the second closest connection exceeds the `nn_dist`. The latter distance must be given in the same units as `cutoff`. However, if no arguments are given then we close every polygon regardless of the gap between first and last point.

12.12.2 示例

```
gmt connect segment_*.txt -Tf0.1 > new_segments.txt
```

```
gmt connect my_lines.txt -T150e -DMap_segment_%04d.dat
```

12.13 gmtconvert

官方文档 [gmtconvert](#)

12.14 gmtdefaults

官方文档 [gmtdefaults](#)

简介 列出所有 GMT 参数的当前值或系统默认值

12.14.1 必选选项

无

12.14.2 可选选项

-D[u|s] 打印系统默认参数值。若不使用该选项，则打印参数当前值。

1. **-D** : 列出 GMT 系统默认参数值
2. **-Du** : 列出 GMT 的 US 单位制下的默认参数值
3. **-Ds** : 列出 GMT 的 SI 单位制下的默认参数值

12.14.3 示例

在当前目录生成一份 GMT 系统默认配置:

```
gmt gmtdefaults -D > gmt.conf
```

搜索与 LABEL 有关的参数的值:

```
gmt gmtdefaults | grep LABEL
```

12.14.4 BUGS

- -D 选项无效 (v5.1.2)

12.15 gmtflexure

官方文档 [gmtflexure](#)

12.16 gmtget

官方文档 [gmtget](#)

简介 列出单个或多个 GMT 参数的当前值

12.16.1 必选选项

<PARAMETER> GMT 的参数名, 见[配置 GMT](#)

`gmtset` 后直接跟一个或多个参数名:

```
$ gmt gmtget PS_MEDIA
a4

$ gmt get MAP_GRID_CROSS_SIZE_PRIMARY MAP_GRID_CROSS_SIZE_SECONDARY
24p,Helvetica,black 16p,Helvetica,black fancy
```

12.16.2 可选选项

-G<defaultsfile> 读取指定的 GMT 配置文件

默认情况下, 该模块会按照如下优先级寻找配置文件 `gmt.conf`

```
./gmt.conf > ~/gmt.conf > ~/.gmt/gmt.conf > 系统默认参数
```

-L 输出时一行只输出一个返回值。

一次指定多个参数名时, 默认会将所有返回值输出在一行, 各值之间以空格分隔。该选项会一行只输出一个返回值:

```
$ gmt get FONT_TITLE FONT_LABEL MAP_FRAME_TYPE -L  
24p,Helvetica,black  
16p,Helvetica,black  
fancy
```

12.17 gmtgravmag3d

[官方文档 gmtgravmag3d](#)

12.18 gmtinfo

[官方文档 gmtinfo](#)

简介 从表数据中提取极值信息

12.18.1 最小示例

提取文件中各列的极值:

```
$ gmt info input.dat  
input.dat: N = 6 <1/2> <1/3> <0/3.2>
```

从输出中可以看到, 输入文件中有 6 行 3 列, 三列数据的范围分别是 1 到 2、1 到 3 和 0 到 3.2。

12.18.2 可选选项

-Aa|f|s 输入数据为多段数据或多文件时的处理方式(默认为 -Aa):

1. **-Aa** : 报告所有文件的总最值
2. **-Af** : 分别报告各个文件自己的最值
3. **-As** : 分别报告各段数据自己的最值

-C 输出时, 每列的最大最小值分别占一列

在输出每一列的最大最小值时, 默认会用 `<min/max>` 的格式输出, 不易于脚本处理。该选项会将每列的最大最小值分别输出为一列:

```
$ gmt info input.dat -C  
1 2 1 3 0 3.2
```

该选项经常与 **-i** 及 **-o** 选项一起使用, 以提取部分列的最值信息。

-D 对 **-I** 选项的结果做整体偏移, 使得区域范围的中心与原数据的中心对齐。

比如数据的真实范围是 `0/3` , 使用 **-I2** 选项会得到 `0/4` , 在此基础上使用 **-D** 选项, 则变成 `-0.5/3.5` 。

-EL|l|H|h<col> 返回第 `<col>` 列最值所在的记录行。

1. **l|H** 表示返回最小、最大值所在的记录行
2. **L|H** 表示返回绝对值最小、最大值所在的记录行

若有多行均满足要求，则只返回第一行。若 `<col>` 未指定，则默认是最后一列。

-F[i|d|t] 返回记录统计数据

- `-Fi` 返回总文件数、总数据段数、总数据记录数、总头段记录数、总记录数
- `-Fd` 返回每个数据段的信息：表号、段号、行数、开始记录号、结束记录号
- `-Ft` 与 `-Fd` 类似，但会对每个输入文件重置段号、开始记录号和结束记录号

-I[p|f|s]<dx>[/<dy>[/<dz>...]] 调整最大/小值使得其是给定的增量的最近倍数，返回的格式为 `-Rw/e/s/n`

```
$ gmt info input.dat -I2/2
-R0/2/0/4
```

如果设置 `<dx>` 的值为 `-`，则表示以 `-Rw/e/s/n` 的格式输出真实的 `-R` 范围：

```
$ gmt info input.dat -I-
-R1/2/1/3
```

当 `-I` 选项和 `-C` 选项一起使用时，输出不再是 `-Rw/e/s/n` 格式：

```
$ gmt info input.dat -C -I2
0  2  0  4  0  3.2
```

上面的例子中，`-I` 选项后只接了一个增量，此时第二列数据也会使用同样的增量（出于向后兼容性考虑）。若只想要为第一列指定增量，第二列不指定，可以使用 `-Ip` 选项：

```
$ gmt info input.dat -C -Ip2
0  2  1  3  0  3.2
```

对于某些算法，比如 FFT 或者 surface 命令，`-R` 选项的不同取值会影响到运算的速度，`-If` 会输出优化后的 `-R` 以得到最快 FFT 结果，`-Is` 会输出优化后的 `-R` 以得到最快的 surface 结果。

-L 确定多个表数据 (`-Af`) 或多段数据 (`-As`) 的共同范围。若与 `-I` 选项连用，则最终的结果会向内近似以使得极值位于真实数据范围内。

-S[xy] 为绘制误差棒增加额外的列，与 `-I` 选项一起使用生成数据并直接用于接下来 `psxy -E` 绘制误差棒。

- `-Sx` 将数据的第三列 (2) 作为水平误差棒的数据
- `-Sy` 将数据的第四列 (3) 作为垂直误差棒的数据
- `-Sxy` 将数据的第三 (2) 和第四 (3) 列作为水平和垂直误差棒的数据

-T<dz>[+c<col>] 以 `-Tzmin/zmax/dz` 的形式报告第 `<col>` 列的最值，`<col>` 的默认值为 0。该选项不能与 `-I` 选项连用。

12.19 gmtlogo

官方文档 [gmtlogo](#)

说明 在图上绘制 GMT 的图形 logo

该模块将 GMT 的图形 logo 绘制在图上。默认情况下, GMT 的图形 logo 默认宽 2 英寸, 高 1 英寸, 将放在当前的绘图原点处。

12.19.1 必选选项

无

12.19.2 可选选项

-D[g|j|J|n|x]<refpoint>[+w<width>][+j<justify>][+o<dx>[/<dy>]]

设置 logo 在图中的位置

- [g|j|J|n|x]<refpoint> 设置底图上的参考点, 见[绘制修饰物](#)一节
- +j<justify> 设置 logo 上的锚点, 见[绘制修饰物](#)一节
- +o<dx>[/<dy>] 设置参考点的额外偏移量, 见[绘制修饰物](#)一节
- +w<width> 设置 logo 的宽度

-F[+c<clearances>][+g<fill>][+i[[<gap>/]<pen>]][+p[<pen>]][+r[<radius>]][+s[<

在 logo 后加一个背景面板, 见[绘制修饰物](#)一节

- +p<pen> 面板边框的画笔属性 (默认画笔属性由参数 MAP_FRAME_PEN 决定)
- +g<fill> 面板填充色 (默认不填充颜色)
- +c<clearances> 设置 logo 与面板边框之间空白区域的大小
- +i<gap>/<pen> 为背景面板加上额外的内边框 (默认的 <gap> 值为 2p, 内边框的画笔属性为 MAP_DEFAULTS_PEN)
- +r<radius> 面板使用圆角矩形边框 (默认圆角半径为 6p)
- +s<dx>/<dy>/<fill> 为面板增加背景阴影区 (默认值为 4p/-4p/gray50)

12.19.3 示例

单独绘制一个 2 英寸宽的 GMT logo:

```
gmt logo -P -Dx0/0+w2i > logo.ps
```

将 GMT logo 作为一个图层放在当前底图的左上角:

```
gmt logo -R -J -DjTR+o0.1i/0.1i+w3i -F+glightblue+s -K -O >> bigmap.ps
```

12.20 gmtmath

官方文档 [gmtmath](#)

12.21 gmtpmodeler

官方文档 [gmtpmodeler](#)

12.22 gmtregress

[官方文档 gmtregress](#)

12.23 gmtselect

[官方文档 gmtselect](#)

简介 基于多种准则筛选数据

该命令会从输入文件中读取前两列作为经度和纬度，并判断该点是否符合一定的空间准则，以筛选出符合条件的记录。输入文件中仅前两列会被使用。

七个空间准则包括：

1. 在矩形区域内 (-R 和 -J)
2. 与点文件中的每个点的距离在一定范围之内
3. 与线文件中的每条线的距离在一定范围之内
4. 在多边形文件所指定的多边形内
5. 在某个地理区域内 (需要海岸线数据)
6. Z 值在某个范围内
7. 该点所在的网格单元内具有有效值 (即非零和非 NaN 的值)

12.23.1 七个空间准则

准则 1

使用 -R 和 -J 筛选出在该区域内的点：

```
gmt select points.xy -R0/5/0/5
```

准则 2

筛选所有与点文件中的每个点的距离在一定范围内的点。

-C<pointfile>+d<dist>[<unit>] 该准则会筛选出与文件 <pointfile> 中的每个点的距离在 <dist> 之内的记录。

若 <dist> 等于 0，则 <pointfile> 中的第三列是每个数据点各自的影响半径，即筛选出不在任何一个数据点的影响半径内的点。默认情况下 <dist> 是笛卡尔坐标系下的距离，单位为用户单位。若指定 -fg 选项，则表明 <dist> 为球面距离。若使用了 -R 和 -J，则 <dist> 表示投影后的纸面距离，单位由参数 PROJ_LENGTH_UNIT 决定。

准则 3

筛选所有与线文件中的每条线的距离在一定范围之内的点。

-L<linefile>+d<dist>[<unit>][+p] <linefile> 中包含了一系列线段，该

准则会筛选出与这些线段的距离不超过 `<dist>` 的记录。

若 `<dist>` 等于零, 则可以在 `<linefile>` 中每段数据的段头记录中使用 `-D<dist>` 参数, 为每个线段分别指定距离值。

默认情况下 `<dist>` 是笛卡尔坐标系下的距离, 单位为用户单位。若指定 `-fg` 选项, 则表明 `<dist>` 为球面距离。若使用了 `-R` 和 `-J`, 则 `<dist>` 表示投影后的纸面距离, 单位由 `PROJ_LENGTH_UNIT` 决定。

使用 `+p` 则会将数据点垂直投影到线段上, 只有投影位置在线段的两个端点内的记录才会被保留, 即只有线段左右一定距离内的点才会被保留, 超过线段两端点的点不会被保留。

准则 4

筛选出在某个多边形内的点。

`-F<polygonfile>` `<polygonfile>` 中可以包含一个或多个多边形, 该选项筛选出所有在多边形内的记录。

准则 5

根据地理特征信息筛选数据。

`-N<wet>/<dry>` 跳过或保留陆地 (`dry`) /海湖 (`wet`) 区域内的点。

`<wet>` 和 `<dry>` 可以取 `s` 或 `k`, 分别表示 skip 和 keep。默认值为 `-Ns/k`, 即保留所有位于陆地上的记录, 并跳过所有海洋、湖泊中的记录。

`-N<ocean>/<land>/<lake>/<island>/<pond>` 进一步细分地理特征, 五项分别表示海洋、陆地、湖泊、岛屿、池塘 (?). 每一项均可以取 `s` 或 `k`, 分别表示 skip 和 keep。默认值为 `-Ns/k/s/k/s`, 等效于 `-Ns/k`, 即仅保留所有陆地上的记录。

`-D[a|f|h|i|l|c][+]` 选择海岸线数据的精度, 仅与 `-N` 选项一起使用有效。见 [pscoast](#) 中 `-D` 选项的介绍。

`-A<min_area>[/<min_level>/<max_level>][+ag|i|s|S][+r|l][+p<percent>]`

不绘制面积过小的区域 (湖泊或岛屿), 或不绘制某个级别的湖泊边界。

在绘制湖泊时, 若不管湖泊的面积大小而把所有湖泊的边界都画上去, 可能导致图看上去比较乱, 该选项用于对湖泊进行筛选。面积小于 `<min_area>` 平方千米或者湖泊级别不在 `[min_level,max_level]` 范围的边界都不会被绘制。默认值为 `0/0/4`, 即绘制所有湖泊边界, 即绘制 0 到 4 级所有级别的面积大于 0 的湖泊。

对于 `level=2`, 即湖岸线, 包括常规的湖以及很宽的河流。加上 `+r` 则只绘制河流, 加上 `+l` 则只绘制常规湖。

对于南极洲而言, 因为有冰层的存在, 所以海岸线有多种处理方式:

1. `+ai` 用 ice shell boundary 作为南极洲的海岸线, 默认值

2. +ag 以 ice grounding line 作为海岸线
3. +as 忽略南纬 60 度以南的海岸线, 用户可以使用 psxy 绘制自己的南极洲海岸线
4. +aS 忽略南纬 60 度以北的海岸线

+p<percent> : 一个多边形, 降低精度后, 边数减少, 面积变化, 当面积变化过大时再绘制这个多边形就不合适了, 该子选项用于去除那些面积与最高精度面积之比小于 <percent> 的多边形。

准则 6

筛选 Z 值在某个范围内的点

-Z<min>[/<max>][+c<col>] 判断记录的 Z 值是否在 <min> 到 <max> 之间或等于 NaN。

若省略 <max> 则判断 Z 值是否等于 <min>。若不限制范围的上限或下限, i 可以使用 - 代替。

若第三列 Z 值代表时间, 想要判断 Z 值是否在某个时间范围内, 需要使用 -f2T 选项。

可以使用 +c<col> 指定记录中的某一列作为 Z 值, 默认以第三列 (col=2) 作为 Z 值。若想要对多列做类似的测试, 可以重复使用 -Z 选项, 每次指定不同的列号。注意: 当多次使用 -N 选项时, 不可使用 -Iz 选项。

准则 7

根据数据点所在的网格单元内具有有效值 (即非零和非 NaN 的值) 来筛选数据。

-G<gridmask> 使用 -G<gridmask> 指定一个网格文件。对于每个数据点而言, 判断其对应的网格单元是否具有有效值 (即非零或非 NaN 的值), 若该网格单元有有效值, 则保留该数据点。

12.23.2 其他选项

-E[fn] 在判断点是否在一个多边形内时, 默认会将恰好在多边形边界线上的点也认为是在多边形内, 使用该选项会将多边形上的点认为是在多边形外部。f 和 n 分别修改 -F 和 -N 选项的行为。

-I[cfglsrz] 对七个准则取反, 即筛选出不符合准则的记录。c, f, g, l, r, s, z 分别对应于 -C、-F、-G、-L、-R、-S 和 -Z。

12.23.3 示例

筛选与 pts.txt 中所有点的距离在 300 km 以内, 与 lines.txt 中线段的距离在 100 km 以外的点:

```
gmt select lonlatfile -fg -Cpts.txt+d300k -Llines.txt+d100k -Il > subset
```

此处需要使用 `-fg` 以告知程序正在处理地理数据。

筛选某个区域内所有不在陆地上的点:

```
gmt select data.txt -R120/121/22/24 -Dh -Nk/s > subset
```

筛选 `quakes.txt` 中所有位于多边形区域内的点:

```
gmt select quakes.txt -Flonlatpath.txt -fg > subset
```

`stations.txt` 中的点投影后与 `origin.txt` 的距离在 5 cm 之内的点:

```
gmt select stations.txt -Corigin.txt+d5 -R20/50/-10/20 -JM20c \
--PROJ_LENGTH_UNIT=cm > subset
```

12.24 gmtset

[官方文档 gmtset](#)

简介 修改单个或多个 GMT 参数的值

该命令会修改当前目录下 GMT 配置文件 `gmt.conf` 中指定参数的值。若配置文件不存在，则新建一个并设置对应的参数值。

由于该命令会修改当前目录下的配置文件 `gmt.conf`，因而该命令修改的参数将对接下来的命令一直有效，直到 GMT 参数再次被修改或覆盖。若想要参数修改仅对某个命令有效，可以在该命令中使用 `--<PARAMETER>=<value>` 语法。

注解: GMT 5.4.0 之后的版本中，只有与 GMT 系统默认值不同的参数才会被写到 `gmt.conf` 中。

12.24.1 必选选项

<PARAMETER>=<value> 要修改的 GMT 参数名 `<PARAMETER>` 以及想要设置的值 `<value>`

参数名和值必须成对存在，二者可以用等号 = 连接，也可以省略等号。

修改主标注字体为 Helvetica，字号为 12p，设置网格交叉线的尺寸为 0.2 厘米:

```
gmt gmtset FONT_ANNOT_PRIMARY 12p,Helvetica MAP_GRID_CROSS_SIZE_PRIMARY 0.2c
```

也可以用等号将参数名和参数值连接起来:

```
gmt gmtset FONT_ANNOT_PRIMARY=12p,Helvetica MAP_GRID_CROSS_SIZE_PRIMARY=0.2c
```

12.24.2 可选选项

- C 将当前目录下的 GMT4 配置文件 .gmtdefaults4 转换为 GMT5 所使用的 gmt.conf 文件，并保留原 GMT4 配置文件。
- D[s|u] 基于系统配置文件修改参数值。

1. -D : 使用 GMT 编译过程中指定的默认参数文件
2. -Du : 使用 US 单位制下的默认参数文件
3. -Ds : 使用 SI 单位制下的默认参数文件

使用 -D 选项而不指定其他参数则会在当前目录下生成一个系统配置文件：

```
gmt set -D
```

- G<defaultsfile> 指定要读取并修改的配置文件名 <defaultsfile>
- [BJRXxYycop]<value> 修改 GMT 命令历史文件 gmt.history 中的值。该选项用于在不执行其他绘图命令的前提下强行修改命令历史，但用处不大。

例如：

```
$ gmt psbasemap -JX10c/5c -B0/10/0/10 -B1 > test.ps
$ gmt gmtset -JX5c/5c
```

该命令会将 GMT 命令历史文件 gmt.history 中 -J 选项的值从 X10c/5c 改成 X5c/5c 。

12.24.3 FAQ

1. 错误消息：

```
gmtset: Warning: parameter xxxx is deprecated. Use xxx instead.
```

出现该错误是因为 GMT5 重命名了几乎所有参数，而当前命令使用的是 GMT4 的老参数名。解决办法在警告信息中已经说了，使用 GMT5 的新参数名即可。

12.25 gmt simplify

[官方文档 gmt simplify](#)

简介 使用 Douglas-Peucker 算法对线段做简化

该命令可以将一个复杂多边形进行简化，用一条直线代替一系列点，并保证每个点与直线的偏离都在可容忍的范围内。

12.25.1 选项

-T<tolerance>[<unit>] 指定最大所能容忍的误差。默认单位为用户单位，也可以指定其他距离单位。

12.25.2 示例

将线段简化, 可容忍误差为 2 千米:

```
gmt simplify segment.d -T2k > new_segment.d
```

将笛卡尔线段简化, 可容忍误差为 0.45:

```
gmt simplify xyline.d -T0.45 > new_xyline.d
```

12.25.3 参考文献

Douglas, D. H., and T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line of its caricature, *Can. Cartogr.*, **10**, 112-122, 1973.

12.26 gmtspatial

官方文档 [gmtspatial](#)

12.27 gmtswitch

官方文档 [gmtswitch](#)

12.28 gmtvector

官方文档 [gmtvector](#)

12.29 gmtwhich

官方文档 [gmtwhich](#)

简介 返回指定文件的完整路径

GMT 会依次在如下目录中去寻找命令行中指定的文件:

```
当前目录 > $GMT_USERDIR > $GMT_DATADIR > $GMT_CACHEDIR
```

该命令会报告文件的完整路径, 使得用户可以确认自己在使用的究竟是哪个数据文件。

12.29.1 必选选项

<files> 任意数据文件名

12.29.2 可选选项

- A** 仅考虑用户有读权限的文件
- C** 不报告完整路径, 只打印 Y 或 N 以表示是否找到文件
- D** 不报告完整路径, 仅打印包含该文件的目录名
- G** GMT 可以自动下载某些文件:
 - 以 URL 形式给出的文件会自动下载
 - 以 @filename 形式给出的文件会从 GMT 网站上下载
 - GMT 提供的全球地形起伏数据 earth_relief_*.grd

使用该选项, 若 GMT 没有在本地缓存目录中找到这些文件, 则会尝试下载。

12.30 gpsgriddeder

官方文档 [gpsgriddeder](#)

12.31 gravfft

官方文档 [gravfft](#)

12.32 grd2cpt

官方文档 [grd2cpt](#)

12.33 grd2rgb

官方文档 [grd2rgb](#)

简介 将网格文件、原始 RGB 文件、SUN 光栅文件转换成 R/G/B 三个网格文件

该命令的作用是:

1. 读取 8、24 或 32 位 Sun 光栅文件, 并将其中的 red、green、blue 分量 (0-255) 分别写到三个网格文件中, 由于光栅文件的头段信息有限, 可以使用 -R、-I 和 -r 选项设置完整的头段信息
2. 读取 2D 网格文件, 并根据指定的 CPT 文件, 将 Z 值转换为对应的 RGB 值, 然后将 red、green、blue 分量分别写到单独的网格文件中
3. 读取 RGB 或 RGBA 原始光栅文件, 并将 red、green、blue 分量分别写到单独的网格文件中, 此时需要使用 -W 选项

12.33.1 必选选项

<infile> 输入文件, 可以是 Sun 光栅文件、2D 网格文件或原始光栅文件

-G<template> 要生成的三个网格文件的文件名模板, 模板中必须包含 %c

例如, -Gout_%c.nc 会生成三个网格文件 out_r.nc、out_g.nc 和 out_b.nc

12.33.2 可选选项

-C<cpt> 若输入文件是 2D 网格文件, 则需要指定 CPT 文件, 以将 Z 值转换为 RGB 颜色

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- <xinc> X 方向的网格间隔
- <yinc> Y 方向的网格间隔
- <unit> 网格间隔的单位。对于地理坐标系而言默认值为度, 可以设置单位为 m|s|e|f|k|M|n|u。
- +e 微调 X 和 Y 方向范围的最大值, 使得其是网格间隔的整数倍 (默认会微调网格间隔以适应给定的数据范围)
- +n 表明 <xinc> 和 <yinc> 不是网格间隔, 而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意:

1. 若 <yinc> 设置为 0, 则表示其与 <xinc> 相同
2. 若使用 -R<grdfile> 选项, 则网格网格已经根据网格文件自动初始化, 此时依然可以使用 -I 覆盖网格文件中的间隔值

-L<layer> 只输出某一分量, <layer> 可以取 r|g|b。默认会输出三个分量。

-W<width>/<height>[/<n_bytes>] 设置原始光栅文件中的尺寸信息。

原始光栅文件中不包含图片的尺寸信息, 因而需要单独的选项来设置。默认情况下, 假定光栅文件是 RGB 原始文件, 即每个像素 3 个字节; 对于 RGBA 文件, 每个像素 4 个字节, 此时 <n_bytes>=4。

使用 -W=/=4 可以自动猜测 RGBA 光栅文件的尺寸信息。

12.33.3 示例

使用 CPT 文件将 2D 网格文件转换为 R、G、B 三个网格文件:

```
gmt grd2rgb hawaii_grv.nc -Ctopo.cpt -Ghawaii_grv_%c.nc
```

从 Sun 光栅文件中输出 R 分量:

```
gmt grd2rgb radiation.ras -Lr -Gcomp_%c.nc
```

12.34 grd2xyz

官方文档 [grd2xyz](#)

简介 将网格文件转换成表数据

该命令读取一个或多个 2D 网格文件, 并将 XYZ 数据以 ASCII 或二进制格式写到标

准输出中。ASCII 输出的格式由参数 FORMAT_FLOAT_OUT 控制。

12.34.1 必选选项

<grid> 要转换的 2D 网格文件

12.34.2 可选选项

-C[f|i] 输出的 XY 坐标值用对应的列、行号替代

默认输出的三列数据是：X 坐标、Y 坐标和 Z 值。使用该选项，则输出的三列数据为：列号、行号和 Z 值。其中，行号和列号从 0 开始算起。

- 使用 -Cf 则行号和列号从 1 开始算起

- -Ci 会输出两列数据：索引和 Z 值。索引相当于是将二维数组用一维数组表示

-R<west>/<east>/<south>/<north> 指定要转换的网格文件的数据范围

-W[a|<weight>] 默认输出 XYZ 三列数据，使用该选项则输出四列 XYZW，其中 W 即 <weight>

若使用 -Wa 则权重为每个节点所占据的面积。

-Z[flags] 以 ASCII 或二进制形式输出表数据

使用该选项，则输出时只有 Z 值，没有 XY 信息。输出 Z 值的顺序由 <flags> 决定。若是行优先，<flags> 的第一个字符可以取：

- T 表示第一行是 y=ymax
- B 表示第一行是 y=ynin

<flags> 的第二个字符可以取：

- L 表示每一行的第一个元素是 x=xmin
- R 表示每一行的第一个元素是 x=xmax

若是列优先，则 L|R 为第一个字符，B|T 为第二个字符。

对于网格线配准的网格文件而言：

- 若网格在 X 方向是周期的，输出数据时不需要包含 x=xmax 所在的列，则加上 x
- 若网格在 Y 方向是周期的，输出数据时不需要包含 y=ymax 所在的行，则加上 y

若数据需要做字节交换，则加上 w。最后需要指定，数据以何种数据类型保存：

- a ASCII 表，每行输出一个 Z 值
- c int8_t, signed 1-byte character
- u uint8_t, unsigned 1-byte character
- h int16_t, short 2-byte integer
- H uint16_t, unsigned short 2-byte integer

- `i` int32_t, 4-byte integer
- `I` uint32_t, unsigned 4-byte integer
- `l` int64_t, long (8-byte) integer
- `L` uint64_t, unsigned long (8-byte) integer
- `f` 4-byte floating point single precision
- `d` 8-byte floating point double precision

默认值为 `-ZTLa`。

12.34.3 示例

```
gmt grd2xyz hawaii_grv.nc > hawaii_grv.xyz
```

```
gmt grd2xyz raw_data.nc -ZTLf > hawaii_grv.b
```

12.35 grdbblend

[官方文档 grdbblend](#)

简介 将多个部分重叠的网格文件合并成一个网格文件

该模块会读取多个网格文件以及一个合并参数文件，并将多个网格文件合并成一个网格文件。

12.35.1 必选选项

-G<outgrid> 输出的网格文件名

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- `<xinc>` X 方向的网格间隔
- `<yinc>` Y 方向的网格间隔
- `<unit>` 网格间隔的单位。对于地理坐标系而言默认值为度，可以设置单位为 `m|s|e|f|k|M|n|u`。
- `+e` 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- `+n` 表明 `<xinc>` 和 `<yinc>` 不是网格间隔，而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意：

1. 若 `<yinc>` 设置为 0，则表示其与 `<xinc>` 相同
2. 若使用 `-R<grdfile>` 选项，则网格网格已经根据网格文件自动初始化，此时依然可以使用 `-I` 覆盖网格文件中的间隔值

-R<west>/<east>/<south>/<north> 数据区域范围

12.35.2 可选选项

<grid1> <grid2> 要合并的网格文件列表

<blendfile> 文件中包含了要合并的网格文件的文件名列表, 其格式为:

```
网格文件名(必须) -R 选项(可选) 相对权重(可选)
```

- 在考虑权重的情况下, **-R** 范围外的部分会给零权重, 范围内的部分则会使用指定的相对权重, 在边界处会加上 2D 余弦 taper 权重。
- 若相对权重为负值, 则整个反过来, 即范围内的会给零权重, 范围外的会给权重的绝对值。
- **-R** 选项可以取值 **-**, 此时会直接使用网格文件的数据范围。
- 若未指定权重, 则默认权重为 1
- 也可以不指定 **<blendfile>**, 只是将所有网格文件名列在命令行中, 此时会使用网格数据的真实范围, 且所有网格文件的权重都是 1.

-C[f|l|o|u] 在合并时, 若多个网格出现重叠, 则忽略权重并按照下面的规则设置重叠区域的值:

- **-Cf** 将第一个访问该节点的网格数据作为该节点的值
- **-Co** 将最后一个访问该节点的网格数据作为该节点的值
- **-Cl** 将所有网格文件在该节点的值的最小值作为该节点的值
- **-Cu** 将所有网格文件在该节点的值的最大值作为该节点的值

-N<nodata> 将无数据的节点的值设置为 **<nodata>**, 默认值为 **Nan**

-Q 创建一个无头段的二进制网格文件, 使得其可以被 **grdraster** 所用。

-W[z] 不合并数据, 仅输出每个节点所使用的权重。**-Wz** 表示输出 Z 值乘以权重的和。

-Z<scale> 在输出前先将数据乘以比例因子 **<scale>**, 默认值为 1

12.35.3 示例

假设要合并几个网格文件, 则可以设置 **blendfile** 的内容为:

```
piece_1.nc -R0/30/-90/90 1
piece_2.nc -R25/50/-90/90 1.5
piece_3.nc -R45/80/-90/90 0.9
piece_4.nc -R80/160/-90/90 1
```

执行如下命令即可实现数据合并:

```
gmt grdbblend blend.job -Gblend.nc -R0/160/-90/90 -I1m/1m -V
```

将所有网格文件 **MB_*.nc** 以相同权重合并:

```
gmt grdbblend MB_*.nc -Gblend.nc -R0/360/-90/90 -I1m/1m -V
```

12.36 grdclip

官方文档 [grdclip](#)

简介 对网格文件的 Z 值范围做裁剪

12.36.1 必选选项

<ingrid> 输入网格文件名

-G<outgrid> 输出网格文件名

12.36.2 可选选项

-R<west>/<east>/<south>/<north>[+r][+u<unit>] 指定要截取的网格区域。若该选项指定的范围超过了网格文件的边界，则仅提取二者公共的区域。

-S 裁剪参数，该选项可以多次使用。

- **-Sa<high>/<above>** 将所有大于 **<high>** 的值设置为 **<above>**
- **-Sb<low>/<below>** 将所有小于 **<low>** 的值设置为 **<below>**
- **-Si<low>/<high>/<between>** 将所有在 **<low>** 和 **<high>** 范围内的值设置为 **<between>**，该选项可多次使用
- **-Sr<old>/<new>** 将所有等于 **<old>** 的值设置为 **<new>**，该选项可以多次使用

12.36.3 示例

将所有大于 70 的值设置为 NaN，并将小于 0 的值设置为 0:

```
gmt grdclip data.nc -Gnew_data.nc -Sa70/NaN -Sb0/0 -V
```

将所有 25 到 30 范围内的值设置为 99, 35 到 39 范围内的值设置为 55, 将 17 换成 11, 将所有小于 10 的值设置为 0:

```
gmt grdclip classes.nc -Gnew_classes.nc -Si25/30/99 -Si35/39/55 -Sr17/11 -Sb10/0 -V
```

12.37 grdcontour

官方文档 [grdcontour](#)

12.38 grdcut

官方文档 [grdcut](#)

简介 从一个网格文件中裁剪出一个子区域的网格文件

该模块可以从一个网格文件中根据条件裁剪出一个新的网格文件:

- **-R** 选项: 直接指定子区域的范围
- **-Z** 选项: 通过检查网格节点的值间接限定子区域的范围

- **-S** 选项：通过指定与特定点的距离间接限定子区域的范围

12.38.1 必选选项

<ingrid> 输入网格文件名
-G<outgrid> 输出网格文件名

12.38.2 可选选项

-N[<nodata>] 允许新网格的区域范围超过原网格的区域范围。

一般情况下，若指定的区域范围大于输入网格的区域范围，超出的部分会被自动忽略，实际的输出网格的区域范围会自动适应输入网格的区域范围。使用 **-N** 选项，则超出的区域范围内，网格节点会被赋以指定的值，默认赋值为 NaN，也可以自己指定其值为 **<nodata>**

-R<west>/<east>/<south>/<north> 直接指定要截取的网格子区域的范围
-S[n]<lon>/<lat>/<radius>[<unit>] 指定圆心位置及其半径，程序会自动计算一个矩形区域，该矩形区域包含了圆上及圆内所有网格点

1. **<lon>/<lat>** 圆心位置
2. **<radius>** 圆半径
3. **<unit>** 圆半径所使用的距离单位，见[单位](#)一节
4. **-Sn** 将所有矩形区域内但不在圆内的节点的值为 NaN

-Z[n|r]<min>/<max> 定义一个 Z 值范围 **<min>/<max>** 并据此确定一个新的矩形子区域，使得在该子区域**外**的所有节点值都位于给定的 Z 值范围外。**<min>** 和 **<max>** 默认值为正负无穷，可以用 - 表示无穷。

为了理解这个选项，需要了解该选项是如何具体实现的。为了得到满足要求的矩形子区域，代码中首先将矩形子区域的四条边界设置为原始网格文件的四条边界，然后逐步向内收缩，当某条边界上的某个网格点的值在给定的 Z 值范围内，则此边界停止收缩。需要格外注意的是，该选项只保证了子区域**外**的所有节点值都位于给定的 Z 值范围外，矩形子区域**内的**的网格点的值并不一定都在 Z 值范围内。

在收缩四个边界时，正常情况下，值为 NaN 的记录会被直接跳过：

- **-Zn** 认为 NaN 记录位于 Z 值范围之外，故而保证子区域内是没有 NaN 记录的
- **-Zr** 认为 NaN 记录位于 Z 值范围之内，因而当遇到 NaN 记录时，则停止收缩边界

待处理：

1. 查看源码确认 **-Z** 选项的解释是否正确
2. 确认 **-Z** 选项的示例是否正确

12.38.3 示例

使用 -R 选项直接指定子区域的范围:

```
gmt grdcut input.nc -Goutput.nc -R0/30/-30/30
```

使用 -S 选项生成一个包含了原点 (45,30) 周围 500 km 以内的所有点的矩形区域，并设置矩形区域内圆外的节点值为 NaN:

```
gmt grdcut input.nc -Goutput.nc -Sn45/30/500k -v
```

使用 -Z 选项使得子区域内的所有节点值都小于 0:

```
gmt grdcut abc.nc -Gdef.nc -Z-/0 -V
```

12.38.4 相关

- [grdpaste](#)
- [grdsample](#)

12.39 grdedit

官方文档 [grdedit](#)

简介 修改网格文件的头段或内容

该模块具有如下功能:

1. 从 2D 网格文件中读入头段信息，并使用命令行中的值替换头段信息
2. 对全球地理网格文件还可以在东西方向旋转
3. 可以用 x, y, z 值替换网格文件中特定结点处的值

12.39.1 必选选项

<grid> 要修改的 2D 网格文件

12.39.2 可选选项

-A 如有必要，则对网格间隔做微调使得其与数据的范围相兼容。仅用于处理 GMT 3.1 之前版本生成的网格文件。

-C 清除网格文件头段区中生成该网格所使用的命令历史

-D[+x<xname>][+y<yname>][+z<zname>][+s<scale>][+o<offset>][+n<invalid>][+t<ti

修改网格文件头段中的基本信息:

- +x<xname> X 变量名及其单位，格式为 varname [unit]，比如 "distance [km]"
- +s<scale> 读入网格数据后要乘以的因子，默认值为 1
- +o<offset> 读入数据后并乘以因子后要加入的常数，默认值为 0

- **+n<invalid>** 指定特定的数用于表示该节点处无有效值, 默认为 NaN
- **+t<title>** 网格文件的标题
- **+r<remark>** 网格文件的注释信息

其他说明:

1. 未指定的项其值保持不变
2. 可以给一个空值以重置某一项, 比如使用 **+t** 而不指定标题则设置标题为空
3. 若文本中包含空格则需要用引号括起来
4. 对于地理数据(比如 **-fg**)而言, <xname> 和 <yname> 会自动设置

假设数据的范围是 **300/310/10/30**, 现修改数据的范围以及标题:

```
gmt grdedit data.nc -R-60/-50/10/30 -D+t"Gravity Anomalies"
```

-E[a|h|l|r|t|v] 对网格进行处理, 该选项与除 **-G** 外的其他选项不兼容

- **-Ea** rotate the grid around 180 degrees
- **-Eh** 水平翻转网格(从左到右)
- **-El** 逆时针将网格旋转 90 度
- **-Er** 顺时针将网格旋转 90 度
- **-Et** 对网格进行转置(想象成一个二维矩阵), 默认使用该选项
- **-Ev** 垂直旋转网格(从上到下)

将网格数据逆时针旋转 90 度, 并将旋转后的网格写到新网格文件中:

```
gmt grdedit oblique.nc -El -Goblique_rot.nc
```

-G<outgrid> 默认情况下, 该命令会直接修改并覆盖原始网格文件, 使用该选项则将修改后的网格写到新的文件中。

-J<params> 使用该选项, 则会将地理相关信息以 CF-1 兼容的元数据形式(可被 GDAL 识别)保存到 netCDF 文件中。

-N<table> 从文件 <table> 中读入 XYZ 数据, 并用这些 XYZ 数据替换网格中对应节点的值。

-R<w>/<e>/<s>/<n> 修改网格文件的范围。同时, 网格间隔会做相应修改。

-S 仅用于全球地理网格数据。将网格沿着经度范围整体偏移, 使得其满足 **-R** 定义的新范围。

原数据范围是 **0/360/-72/72**, 现将数据整体偏移 180 度使得数据范围是 **-180/180/-72/72**

```
gmt grdedit world.nc -R-180/180/-72/72 -S
```

-T 修改网格文件的头段, 将一个网格线配准的文件变成像素配准的文件, 或反之。

使用该选项后, 网格线配准的数据的范围将在四个方向上扩大半个网格间隔, 像素点配置的数据的范围将在四个方向上缩小半个网格间隔。

12.39.3 示例

12.40 grdfft

官方文档 [grdfft](#)

12.41 grdfilter

官方文档 [grdfilter](#)

12.42 grdflexure

官方文档 [grdflexure](#)

12.43 grdgradient

官方文档 [grdgradient](#)

12.44 grdgravmag3d

官方文档 [grdgravmag3d](#)

12.45 grdhisteq

官方文档 [grdhisteq](#)

12.46 grdimage

官方文档 [grdimage](#)

简介 在地图上绘制网格数据

12.46.1 必选选项

<grid> 该命令的输入文件必须是 2D 网格文件, 可以是一个只包含 Z 数据的网格文件, 也可以是三个分别包含 R、G、B 值的网格文件。

12.46.2 可选选项

-A<out_img>[=<driver>] 将图片以光栅格式保存

默认图片会以 PostScript 代码的形式输出, 使用此选项可以以其他图片格式保存。文件名中使用后缀 .ppm 则会以 Portable Pixel Map 格式保存。

若 GMT 支持 GDAL, 则可以以更多的光栅格式保存。

1. <out_img> 为要保存的文件名
 2. <driver> 图片格式, 见 GDAL 的文档
- C<cpt>** 绘图时要使用的 CPT 文件名。

也可以直接使用 GMT 自带的 CPT 文件名, 此时 GMT 会自动根据网格文件的 Z 值范围将自带的 CPT 采样成 16 级的连续 CPT 文件。也可以通过 **-C<color1>, <color2>[,<color3>,...]** 的语法构建一个线性连续 CPT 文件。

-D[r] 表明输入的网格文件是需要通过 GDAL 读取的图片文件, 见官方文档。

-E[i|<dpi>] 设置投影后网格的精度, 默认值为 100。

-G[f|b]<color> 该选项仅当生成的图片是黑白图时才可用。

This option will instead use the image as a transparent mask and paint the mask (or its inverse, with **-Gb**) with the given color combination.

-I[<intensfile>|<intensity>|<modifiers>] 增加光照效果

可以通过三种方式设置光照效果:

1. 给定一个 Z 值范围为 (-1,1) 的网格文件, 该文件可以用 **grdgradient** 生成
2. 给定一个常数作为光照强度
3. 不指定光照强度文件, 只使用 **-I+** 则会自动调用 **grdgradient** 并使用参数 **-A-45 -Nt1** 计算输入网格数据的梯度作为光照强度文件。用户可以使用 **+a<azimuth>+n<args>** 以自定义 **grdgradient** 的 **-A** 和 **-N** 选项

-M 使用 YIQ 转换强制将其转换为灰度图。

-N 对于非矩形地图, 在地图边界处不对图片做裁剪。

-Q 将值为 NaN 的节点处设置为透明色

12.46.3 示例

使用默认的光照效果:

```
gmt grdimage stuff.nc -JX6i+ -I+ > quick.ps
```

12.47 grdinfo

官方文档 [grdinfo](#)

简介 从 2D 网格文件中提取基本信息

能提取的信息包括:

- X、Y、Z 的最大和最小值
- 最大/最小 Z 值所在的位置
- X、Y 的网格间隔
- X 和 Y 方向节点数目
- 均值、标准差
- 中位数、绝对中位差 (median absolute deviation)
- the mode (LMS), LMS scale of z
- 值为 NaN 的节点数
- 网格配准方式

12.47.1 必选选项

grdfile 一个或多个网格文件名

下面展示了一个简单的示例:

```
$ gmt grdinfo test.nc
test.nc: Title: ETOP05 global topography
test.nc: Command: grdraster -R100/150/-30/30 1 -Gtest.nc
test.nc: Remark: /opt/GMT-4.5.13/share/dbase/etopo5.i2
test.nc: Gridline node registration used [Geographic grid]
test.nc: Grid file format: nf = GMT netCDF format (32-bit float), COARDS, CF-1.5
test.nc: x_min: 100 x_max: 150 x_inc: 0.083333333333 name: longitude [degrees_
↪east] nx: 601
test.nc: y_min: -30 y_max: 30 y_inc: 0.083333333333 name: latitude [degrees_
↪north] ny: 721
test.nc: z_min: -10376 z_max: 6096 name: m
test.nc: scale_factor: 1 add_offset: 0
test.nc: format: classic
```

从输出中可以看到很多信息:

- 网格文件中的标题信息;
- 生成该网格文件的命令;
- 网格文件的配准方式, 此处为 Gridline 配准;
- 数据格式为 nf 即 netCDF 32 位 float;
- 数据中 X 维度的最小值 x_min、最大值 x_max、网格间隔 x_inc 以及数据点数 nx;
- 数据中 Y 维度的最小值 y_min、最大值 y_max、网格间隔 y_inc 以及数据点数 ny;
- 数据中 Z 值的最小值 z_min 和最大值 z_max 以及其他信息;

12.47.2 可选选项

-C 将输出信息以 Tab 分隔显示在一行中。输出格式为:

```
name w e s n z0 z1 dx dy nx ny [x0 y0 x1 y1] [med scale] [mean std rms] [n_nan]
```

默认只输出前 11 列, 其含义看名字就可以猜到了。若使用了 -M、-L1、-L2、-M 选项, 则会输出对应的方括号内的项。

若与 -I 选项一起使用, 则输出格式为:

```
NF w e s n z0 z1
```

其中 NF 是总网格数目。

```
$ gmt grdinfo test1.nc -C
test1.nc    100 150 -30 30 -10376 6096  0.083333333333 0.083333333333 601 721
```

-D[<xoff>[/<yoff>]][+i] 将单个网格区域划分为多个特定大小的子区域范围,

子区域大小由 $-I<dx>/<dy>$ 控制, $<xoff>/<yoff>$ 用于指定多个子区域之间的重叠量。 $+i$ 子选项表明若该子区域内无数据则忽略该区域。

```
$ gmt grdinfo earth_relief_01m.grd -D -I120/90
-R-240/-120/-90/0
-R-120/0/-90/0
-R0/120/-90/0
-R120/240/-90/0
-R-240/-120/0/90
-R-120/0/0/90
-R0/120/0/90
-R120/240/0/90
```

-F 以每行输出一个信息的方式显示上例中的输出信息, 与 **-C** 选项不兼容。

-I[<dx> [<dy>] | b | i | r]

1. **-I**: 以 $-I<xinc>/<yinc>$ 的格式输出网格间隔
2. **-Ir**: 以 $-R<w>/<e>/<s>/<n>$ 格式输出网格文件的真实范围
3. **-Ii**: 以 $-R<w>/<e>/<s>/<n>$ 格式输出 img 补充包生成的网格文件的精确范围
4. **-Ib**: 输出网格区域范围的四个顶点对应的坐标
5. $-I<dx>/<dy>$ 会先获取网格的区域范围, 并对该范围做微调使得其是 $<dx>$ 和 $<dy>$ 的整数倍, 并以 $-R<w>/<e>/<s>/<n>$ 的格式输出

示例:

```
$ gmt grdinfo test1.nc -I
-I0.083333333333/0.083333333333

$ gmt grdinfo test1.nc -Ir
-R100/150/-30/30

$ gmt grdinfo test1.nc -Ib
> Bounding box for test1.nc
100    -30
150    -30
150     30
100     30

$ gmt grdinfo test1.nc -I3/3
-R99/150/-30/30
```

当 $-I<dx>/<dy>$ 与 **-C** 选项连用时:

```
$ gmt grdinfo test1.nc -I3/3 -C
1   99  150 -30 30  -10376  6096
```

-L[0|1|2|p|a] 报告 Z 值的其他信息, 该选项可多次使用。

- **-L0** : 扫描整个数据并报告 Z 值的范围, 而不仅仅只是从网格的头段中读取 Z 值范围

- **-L1** : 输出中位数以及 L1 scale (L1 scale= 1.4826*Median Absolute Deviation)
- **-L2** : 输出均值、标准差以及均方根
- **-Lp** : 输出 mode (LMS) 和 LMS scale
- **-La** : 输出以上全部信息

注意, 对于像素配准的地理网格数据而言, 每个节点代表的区域面积随着纬度的增加而减小, 此时 GMT 报告的是网格文件在球面平均下的统计值。

-M 寻找并报告 Z 最小和最大值所对应的坐标, 以及值为 NaN 的网格点的数目

-R<w>/<e>/<s>/<n> 从网格文件中取出一个子区域, 并报告该子区域的信息。

若指定的区域范围超过了网格边界, 则只提取公共区域内的网格信息。

-T[<dz>][+a[<alpha>]][+s] 以 **-T<zmin>/<zmax>/<dz>** 的格式输出 Z 值范围

1. **-T<dz>** : 提取 Z 的最小最大值, 并做微调使得最值是 **<dz>** 的整数倍, 然后以 **-T<zmin>/<zmax>/<dz>** 的格式输出
2. **+a<alpha>** 使用该子选项, 则会对网格文件中的值进行排序, 并排除两端的部分数据。**<alpha>** 代表要排除的数据的百分比, 默认值为 2, 即排除最小的 1% 以及最大的 1% 之后再输出 Z 值范围
3. **+s** : 根据 Z 的绝对最大值, 输出一个关于 0 对称的范围

```
$ gmt grdinfo test1.nc -T0.1  
-T-10376/6096/0.1  
  
$ gmt grdinfo test1.nc -T0.1+s  
-T-10376/10376/0.1
```

12.48 grdlandmask

官方文档 [grdlandmask](#)

简介 根据海岸线数据创建陆地-海洋的 mask 网格文件

该命令会根据海岸线数据, 确定指定网格内的每个节点是在陆地上还是在水中, 并给节点赋予不同的值。

12.48.1 必选选项

-G<mask_grd_file> 生成的 mask 网格文件的文件名

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- **<xinc>** X 方向的网格间隔
- **<yinc>** Y 方向的网格间隔
- **<unit>** 网格间隔的单位。对于地理坐标系而言默认值为度, 可以设置单位为 **m|s|e|f|k|M|n|u**。
- **+e** 微调 X 和 Y 方向范围的最大值, 使得其是网格间隔的整数倍 (默认会微调)

网格间隔以适应给定的数据范围)

- **+n** 表明 **<xinc>** 和 **<yinc>** 不是网格间隔, 而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意:

1. 若 **<yinc>** 设置为 0, 则表示其与 **<xinc>** 相同
2. 若使用 **-R<grdfile>** 选项, 则网格已经根据网格文件自动初始化, 此时依然可以使用 **-I** 覆盖网格文件中的间隔值

12.48.2 可选选项

-A<min_area>[/<min_level>/<max_level>][+ag|i|s|S][+r|l][+p<percent>]

不绘制面积过小的区域(湖泊或岛屿), 或不绘制某个级别的湖泊边界。

在绘制湖泊时, 若不管湖泊的面积大小而把所有湖泊的边界都画上去, 可能导致图看上去比较乱, 该选项用于对湖泊进行筛选。面积小于 **<min_area>** 平方千米或者湖泊级别不在 **[min_level,max_level]** 范围的边界都不会被绘制。默认值为 **0/0/4**, 即绘制所有湖泊边界, 即绘制 0 到 4 级所有级别的面积大于 0 的湖泊。

对于 **level=2**, 即湖岸线, 包括常规的湖以及很宽的河流。加上 **+r** 则只绘制河流, 加上 **+l** 则只绘制常规湖。

对于南极洲而言, 因为有冰层的存在, 所以海岸线有多种处理方式:

1. **+ai** 用 ice shell boundary 作为南极洲的海岸线, 默认值
2. **+ag** 以 ice grounding line 作为海岸线
3. **+as** 忽略南纬 60 度以南的海岸线, 用户可以使用 **psxy** 绘制自己的南极洲海岸线
4. **+aS** 忽略南纬 60 度以北的海岸线

+p<percent>: 一个多边形, 降低精度后, 边数减少, 面积变化, 当面积变化过大时再绘制这个多边形就不合适了, 该子选项用于去除那些面积与最高精度面积之比小于 **<percent>** 的多边形。

-D<resolution>[+] 选项海岸线数据的精度, 见 [pscoast](#) 中的介绍。

-E 默认情况下会将恰好位于多边形边界上的节点当作是在多边形的内部, 使用该选项则会将其认为是在多边形的外部。

-N<maskvale>[o] 设置网格节点的值, 可以是数字, 也可以是 NaN。该选项可以取两种格式:

1. **-N<wet>/<dry>**
2. **-N<ocean>/<land>/<lake>/<island>/<pond>**

默认值为 **0/1/0/1/0** (即 **0/1**), 即将水域内的网格设置为 0, 将陆地内的网格设置为 1。

12.48.3 注意事项

`grdlandmask` 生成的网格文件属于 **分类型** 数据，即所有数据只能取几个固定的值，比如 `-N0/1` 会将水域内的网格值设置为 0，将陆地内的网格值设置为 1。在这种情况下，对这种数据用标准方法（比如样条）进行插值通常会得到无意义的结果，使用时需要小心。

然而，当你使用该网格文件绘制地图时，网格数据会被重新投影使得在投影后的坐标下变成一个矩形。这个过程中涉及到了网格插值，默认使用的插值算法是样条插值，因而可能会在图中产生假象。因而建议在使用 `grdimage` 绘制此类数据时使用 `-nn` 选项即 nearest neighbor 插值算法以避免这一问题。

12.48.4 示例

将所有陆地上的节点设置为 NaN，水域上的节点设置为 1：

```
gmt grdlandmask -R-60/-40/-40/-30 -Dh -I5m -N1/NaN -Gland_mask.nc -V
```

生成全球 1x1 度的网格，并将不同性质的区域设置成不同的值：

```
gmt grdlandmask -R0/360/-90/90 -Dl -I1 -N0/1/2/3/4 -Glevels.nc -V
```

12.49 grdmask

[官方文档 `grdmask`](#)

简介 根据多边形数据或点数据创建 mask 网格文件

12.49.1 必选选项

<pathfiles> 一个或多个 ASCII 数据文件，其中包含了多边形或数据点

-G<mask_grd_file> 生成的 mask 网格文件的文件名

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- **<xinc>** X 方向的网格间隔
- **<yinc>** Y 方向的网格间隔
- **<unit>** 网格间隔的单位。对于地理坐标系而言默认值为度，可以设置单位为 `m|s|e|f|k|M|n|u`。
- **+e** 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- **+n** 表明 **<xinc>** 和 **<yinc>** 不是网格间隔，而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意：

1. 若 **<yinc>** 设置为 0，则表示其与 **<xinc>** 相同
2. 若使用 **-R<grdfile>** 选项，则网格网格已经根据网格文件自动初始化，此时

依然可以使用 **-I** 覆盖网格文件中的间隔值

12.49.2 可选选项

- A[m|p|x|y]** 控制两点之间的连接方式, 见[psxy](#) 命令中对 **-A** 选项的介绍
- N[z|Z|p|P]<values>** 设置位于多边形外部、边界和内部的节点值, 默认值为 0/0/1, 即多边形内部节点值为 1, 其他节点值为 0。

<values> 的形式为 <out>/<edge>/<in>, 可以是任意数值, 也可以是 NaN。

- **-Nz** 将多边形内的节点设置为从多段数据的段头记录中获取的 Z 值, 比如多边形段头记录中的 **-Z<zval>**、**-L<header>** 或 **-aZ=<name>**
- **-NZ** 与 **-Nz** 类似, 只是其会将多边形的边界也当做是多边形的内部
- **-Np** 使用一个从 0 递增的数字作为多边形的 ID, 也可以在其后加上一个数字以指定序列的起始值
- **-NP** 与 **-Np** 类似, 只是其会将多边形的边界当做多边形的内部

需要注意, **-Nz|Z|p|P** 不能与 **-S** 连用。

- S<search_radius>[<unit>]** 对所有数据点设置一个搜索半径, 设置圆内、圆边界、圆外部的节点值。

若 <search_radius> 为 z, 则取输入数据的第三列作为半径。对于地理苏沪剧而言, 可以在 **-Sz** 后加上距离单位。

若未使用 **-S** 选项, 则认为输入数据是一个或多个闭合多边形。

12.49.3 注意事项

grdlandmask 生成的网格文件属于 分类型数据, 即所有数据只能取几个固定的值, 比如 **-N0/1** 会将水域内的网格值设置为 0, 将陆地内的网格值设置为 1。在这种情况下, 对这种数据用标准方法(比如样条)进行插值通常会得到无意义的结果, 使用时需要小心。

然而, 当你使用该网格文件绘制地图时, 网格数据会被重新投影使得在投影后的坐标下变成一个矩形。这个过程中涉及到了网格插值, 默认使用的插值算法是样条插值, 因而可能会在图中产生假象。因而建议在使用 **grdimage** 绘制此类数据时使用 **-nn** 选项即 nearest neighbor 插值算法以避免这一问题。

12.49.4 示例

多边形内和边界上的节点值为 0, 外部值为 1:

```
gmt grdmask coastline_*.xy -R-60/-40/-40/-30 -I5m -N1/0/0 -Gland_mask.nc=nb -V
```

数据点周围 50 千米范围的节点值为 1, 其余为 NaN:

```
gmt grdmask data.xyz -R-60/-40/-40/-30 -I5m -NNaN/1/1 -S50k -Gdata_mask.nc=nb -V
```

将多边形的 ID 作为多边形内部节点的值:

```
gmt grdmask plates.gmt -R-40/40/-40/40 -I2m -Nz -Gplate_IDs.nc=ns -aZ=POL_ID -V
```

将多边形的 ID 作为多边形内部节点的值，但多边形 ID 从 100 开始：

```
gmt grdmask plates.gmt -R-40/40/-40/40 -I2m -Np100 -Gplate_IDs.nc=ns -V
```

12.50 grdmath

[官方文档](#) [grdmath](#)

12.51 grdpaste

[官方文档](#) [grdpaste](#)

简介 将两个网格文件沿着其共同边界拼接成一个文件

要合并的两个网格文件必须拥有相同的网格间隔以及一条共同的边。若不满足，则需要使用 [grdcut](#)、[grdsample](#) 命令对网格数据做处理再拼接。对于地理网格数据而言，可能需要使用 **-f** 选项以正确处理经度的周期性。

12.51.1 必选选项

<file_a.nc> <file_b.nc> 要进行拼接的两个网格文件名
-G<outfile.nc> 拼接后生成的网格文件名

12.51.2 示例

一个简单的示例：

```
gmt grdpaste file_a.nc file_b.nc -Goutfile.nc
```

对于地理网格而言，可能需要使用 **-f** 选项处理网格的经度周期性问题：

```
gmt grdpaste file_a.nc file_b.nc -Goutfile.nc -V -fg
```

12.52 grdpmodeler

[官方文档](#) [grdpmodeler](#)

12.53 grdproject

[官方文档](#) [grdproject](#)

简介 对网格数据做地图变换和逆变换

该命令可以将地理坐标下的网格数据投影到一个矩形网格中，也可以将一个矩形坐标系下的网格数据反投影到地理坐标下。

12.53.1 必选选项

<in_grdfile> 要进行变换的 2D 网格数据
-G<out_grdfile> 输出的网格文件名

12.53.2 可选选项

-C[<dx>/<dy>] 默认投影后的坐标是相对于区域的左下角, 该选项使得投影后的坐标相对于投影的中心。<dx>/<dy> 是要加到投影后坐标的偏移量。
-D<xinc>[<unit>][+e|n] [/<yinc>[<unit>][+e|n]] 指定新网格的网格间隔。
-E<dpi> 设置新网格的分辨率, 即每英寸的点数。
-F[c|i|p|e|f|k|M|n|u] 强制 1:1 比例, 即输出数据的单位是真实的投影长度, 默认单位为 m。也可以指定为其他单位。
-I 逆变换, 将矩形区域转换成地理区域。
-Mc|i|p 指定投影后的测量单位, 默认值由参数 PROJ_LENGTH_UNIT 决定。

12.53.3 示例

将地理网格数据转换成 Mercator 网格, 分辨率为 300dpi:

```
gmt grdproject dbdb5.nc -R20/50/12/25 -Jm0.25i -E300 -r -Gdbdb5_merc.nc
```

将网格数据逆变换为地理网格:

```
gmt grdproject topo_tm.nc -R-80/-70/20/40 -Jt-75/1:500000 -I -D5m -V -Gtopo.nc
```

将 UTM (以米为单位) 下的网格数据逆变换为地理网格:

```
gmt grdproject topo_utm.nc -R203/205/60/65 -Ju5/1:1 -I -Mm -Gtopo.nc -V
```

12.54 grdraster

[官方文档 grdraster](#)

简介 从二进制数据中提取子区域并保存为 GMT 网格文件

该命令会读取根据 `grdraster.info` 的内容, 读取指定的二进制数据, 从中提取出一个区域里的数据, 并保存为网格文件。

GMT 会依次在当前目录、环境变量 `$GMT_USERDIR` 和 `$GMT_DATADIR` 指定的目录、`$GMT_SHARED_DIR/dbase` 目录下寻找 `grdraster.info` 文件。

注解: 使用该模块的前提是用户已经自行备好了二进制数据文件, 对于一般用户而言相对复杂, 不建议使用该模块。

12.54.1 grdraster.info

`grdraster.info` 定义了一个数据库，其作用在于指定了本地二进制文件的基本信息，使得 GMT 可以精确地知道二进制数据的格式，进而可以准确读取二进制文件的内容。

文件内容的格式为：

```
文件号 文件标题 Z 值单位 -R 范围 -I 间隔 配准方式 数据格式 scale offset NaN 文件名
```

12.54.2 必选选项

<filenumber> `grdraster.info` 中某文件所对应的文件号。

也可以使用 `text pattern` 找到 `grdraster.info` 中数据描述中匹配的行。

12.54.3 可选选项

-G<grdfile> 默认会直接将数据以 XYZ 格式输出到标准输出流。使用该选项，则会将数据写到网格文件中。

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- `<xinc>` X 方向的网格间隔
- `<yinc>` Y 方向的网格间隔
- `<unit>` 网格间隔的单位。对于地理坐标系而言默认值为度，可以设置单位为 `m|s|e|f|k|M|n|u`。
- `+e` 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- `+n` 表明 `<xinc>` 和 `<yinc>` 不是网格间隔，而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意：

1. 若 `<yinc>` 设置为 0，则表示其与 `<xinc>` 相同
2. 若使用 `-R<grdfile>` 选项，则网格网格已经根据网格文件自动初始化，此时依然可以使用 `-I` 覆盖网格文件中的间隔值

12.54.4 示例

从 1 号数据文件中提取数据，数据分辨率为 30 弧分。为了方便后面进行网格滤波，此处区域范围左右各增加了 4 度：

```
gmt grdraster 1 -R-4/364/-62/62 -I30m -Gdata.nc
```

12.55 grdredpol

官方文档 [grdredpol](#)

12.56 grdconvert

官方文档 [grdconvert](#)

简介 在不同的网格格式之间互相转换

12.56.1 必选选项

<ingrdfile>[=<id>[+s<scale>][+o<offset>][+n<invalid>]] 要读入的网格文件。

<outgrdfile>[=<id>[+s<scale>][+o<offset>][+n<invalid>]][:<driver>[/<datatype>]] 要写入的网格文件。

-N 如果你想要将一个网格文件转换为 native 二进制文件供 [grdraster](#) 模块直接使用，则需要使用 -N 选项，以保证不将 GMT 头段结构写到文件中。

12.56.2 示例

从一个三维网格文件中提取第二层数据:

```
gmt grdconvert climate.nc?temp[1] temp.nc -V
```

将网格文件转换成四字节 native 浮点型网格:

```
gmt grdconvert data.nc ras_data.b4=bf -V
```

将网格文件转换成二字节短整型文件，将其乘以 10 并减去 32000，并设置无数据节点的值为-9999:

```
gmt grdconvert values.nc shorts.i2=bs/10/-32000/-9999 -V
```

将网格文件转换为 [grdraster](#) 可直接使用的二进制文件:

```
gmt grdconvert etopo2.nc etopo2.i2=bs -N -V
```

To create a dumb file saved as a 32 bits float GeoTiff using GDAL:

```
gmt grdmath -Rd -I10 X Y MUL = lixo.tif=gd:GTiff
```

12.57 grdratater

官方文档 [grdratater](#)

12.58 grdsample

官方文档 [grdsample](#)

简介 对网格文件做重采样

该命令读取一个网格文件，并对其做插值以生成一个新的网格文件。新旧网格文件的

区别在于：

1. 不同的配准方式 (-r 或 -T)
2. 不同的网格间隔或网格节点数 (-I)
3. 不同的网格范围 (-R)

默认使用 bicubic 插值方式，可以使用 -n 选项设置其他插值方式。该命令可以很安全地将粗网格插值为细网格；反之，将细网格插值为粗网格时，则可能存在混淆效应，因而需要在插值前使用 grfft 或 grdfilter 对网格文件做滤波。

12.58.1 必选选项

<in_grdfile> 要重采样的 2D 网格文件
-G<out_grdfile> 重采样生成的网格文件

12.58.2 可选选项

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- <xinc> X 方向的网格间隔
- <yinc> Y 方向的网格间隔
- <unit> 网格间隔的单位。对于地理坐标系而言默认值为度，可以设置单位为 m|s|e|f|k|M|n|u 。
- +e 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- +n 表明 <xinc> 和 <yinc> 不是网格间隔，而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意：

1. 若 <yinc> 设置为 0，则表示其与 <xinc> 相同
2. 若使用 -R<grdfile> 选项，则网格网格已经根据网格文件自动初始化，此时依然可以使用 -I 覆盖网格文件中的间隔值

-R<w>/<e>/<s>/<n> 指定新网格的数据范围。

若只使用 -R 选项，则等效于使用 grdcut 或 grdedit -S 。

-T 交换网格文件的配准方式。即若输入是网格线配准，则输出为像素点配准；若输入是像素点配准，则输出为网格线配准。

-n[b|c|l|n][+a][+b<BC>][+c][+t<threshold>] 重采样时使用的插值算法，见 [-n 选项](#) 一节。

12.58.3 示例

将 5x5 弧分的数据采样成 1x1 弧分：

```
gmt grdsample hawaii_5by5_topo.nc -I1m -Ghawaii_1by1_topo.nc
```

将网格线配准的网格文件修改为像素配准的网格文件:

```
gmt grdsample surface.nc -T -Gpixel.nc
```

12.59 grdseamount

[官方文档](#) [grdseamount](#)

12.60 grdspotter

[官方文档](#) [grdspotter](#)

12.61 grdtrack

[官方文档](#) [grdtrack](#)

简介 沿着指定的 (x,y) 位置采样网格点

该命令会读取一个或多个网格文件以及一个表数据。表数据中前两列分别是经纬度，其余列可忽略。该命令会提取指定位置处网格的值，若有必要，会对网格进行插值。

12.61.1 选项

-G<gridfile> 要采样的 2D 网格文件，若需要同时采样多个网格文件，可以重复使用 -G 选项。

<xyfile> 表数据，前两列为 (x,y) 位置信息

-Z 仅输出采样后的 Z 值，默认会输出所有列数据。

12.61.2 相关

- [project](#)

12.62 grdtrend

[官方文档](#) [grdtrend](#)

简介 拟合网格的趋势面并计算残差

该命令会读取一个 2D 网格文件，并用最小二乘方法拟合一个低阶多项式趋势面。多项式趋势面的定义为：

$$\begin{aligned} m1 + m2*x + m3*y + m4*x*y + m5*x*x + m6*y*y + m7*x*x*x + m8*x*x*y \\ + m9*x*y*y + m10*y*y*y. \end{aligned}$$

12.62.1 必选选项

<gridfile> 2D 网格文件名

-N<model>[+r] 指定要拟合的模型。

<model> 指定要拟合的模型的参数个数。例如 -N3 表示 bilinear 趋势, -N6 表示 quadratic 趋势面。加上 +r 表示 robust 拟合, 此时, 程序会根据 robust scale estimate 多次迭代, 给数据重新赋予权重, 以得到一个对 outliers 不敏感的解。

12.62.2 可选选项

-D<diff.nc> 将残差 (输入减去拟合) 结果写到网格文件中

-T<trend.nc> 将拟合得到的趋势文件写到网格文件 <trend.nc> 中

-W<weight.nc> 若 <weight.nc> 存在, 则读取该文件, 并求解一个有权重的最小二乘问题。默认为常规的最小二乘拟合。若 -N 选项中指定了 robust 拟合, 则 robust 拟合中所使用的权重会写到文件 <weight.nc> 中。

12.62.3 示例

从网格文件中移除线性趋势, 并将结果写到残差文件中:

```
gmt grdtrend hawaii_topo.nc -N3 -Dhawaii_residual.nc
```

对网格文件做 bicubic 面的 robust 拟合:

```
gmt grdtrend hawaii_topo.nc -N10r -Thawaii_trend.nc -Whawaii_weight.nc -V
```

12.63 grdvector

[官方文档 grdvector](#)

简介 根据两个网格文件绘制矢量场

该命令会读取两个 2D 网格文件并绘制矢量场。两个网格文件分别代表平面内矢量场的 X 分量和 Y 分量, 最终矢量场用不同长度和方向的箭头表示。两个网格文件, 也可以分别是极坐标下 r 方向和 theta 方向的分量。

12.63.1 必选选项

<compx.nc> 矢量场的 X 分量网格

<compy.nc> 矢量场的 Y 分量网格

12.63.2 可选选项

-A 输入数据为极坐标表示。即网格文件包含的是 (r, theta) 分量而不是 (x, y) 分量

-C[<cpt>] 根据矢量的长度决定矢量的颜色。

有三种方式:

1. 指定用户自己的 CPT 文件
2. 使用 GMT 自带的 CPT 文件 (默认是 rainbow), 该命令会根据网格的 Z 值范围, 自动生成一个 16 级的连续 CPT 文件

3. 指定 `-C<color1>,<color2>[,<color3>,...]` , 根据这些颜色自动构建一个线性连续 CPT 文件

-G<fill> 设置矢量内部的填充色

-I[x]<dx>[/<dy>] 每隔 `<dx>` 和 `<dy>` 绘制一个矢量。其中 `<dx>` 和 `<dy>` 必须是原始网格间隔的整数倍, `<dx>` 和 `<dy>` 后加上 m 表示弧分, 加上 s 表示弧秒。

也可以使用 `-Ix<dx>/<dy>` , 此时 `<dx>` 和 `<dy>` 表示倍数。

-N 不裁剪地图边界外的矢量。

-Q<parameters> 修改矢量的属性。见[绘制矢量/箭头](#)一节。

-S[i|l]<scale> 设置矢量长度的缩放比例。

对于笛卡尔数据而言, 缩放比例即图上单位距离所对应的实际数据的多少, 默认值为 1。可以加上 c|i|p 以指定测量单位。

-Sl<scale> 表示所有矢量拥有固定的长度。

对于地理数据而言, 缩放比例表示每千米所对应的数据单位。使用 `-Si<scale>` 则缩放比例表示每数据单位所对应的千米数。

-T 该选项意味着笛卡尔数据的方位角会根据 X 和 Y 方向缩放比例的符号而改变。

-W<pen> 设置矢量轮廓的画笔属性。

-Z 与 `-A` 选项一起使用, 表明输入的 theta 分量数据表示方位角而不是方向信息。

12.63.3 示例

对矢量长度进行缩放, 使得图上 1 英寸代表实际的 10 个数据单位:

```
gmt grdvector r.nc theta.nc -Jx5c -A -Q0.1i+e+jc -S10i > gradient.ps
```

绘制地理数据, 缩放比例为每个数据单位对应 200 km, 每隔 3 个网格点绘制一个矢量:

```
gmt grdvector comp_x.nc comp_y.nc -Ix3 -JH0/20c -Q0.1i+e+jc -S200 > globe.ps
```

12.64 grdview

[官方文档](#) `grdview`

12.65 grdvolume

[官方文档](#) `grdvolume`

简介 计算网格数据中某个等值线所包围的表面积和体积

该命令读取一个 2D 网格文件, 通过指定某条等值线确定某个 Z 值平面, 并计算由该等值线约束的区域网格表面积、网格表面到该平面所包围的体积, 以及最大平均高度 (体积/面积)。也可以指定一系列等值线, 此时该命令会分别计算每个等值线范围内的表面积和体积。

12.65.1 必选选项

<grdfile> 输入的 2D 网格文件名

12.65.2 可选选项

-C 指定等值线 (Z 值平面), 以计算由该等值线所决定的表面积、体积以及平均高度。若不使用该选项, 则返回整个网格文件的表面积、体积和平均高度。

该选项有四种不同的语法:

- **-C<cval>** 指定单个等值线
- **-C<low>/<high>/<delta>** 指定多条等间距的等值线
- **-Cr<low>/<high>** 计算两个 Z 值平面之间的体积
- **-Cr<cval>** 指定网格最小值和 $z=<cval>$ 两条等值线

-L<base> 同时也计算从等值线到 <base> 的体积

-S[<unit>] 对于地理网格, 默认会将角度转换为 “Flat Earth” 下的距离, 默认单位为米。使用 -S 选项以指定其他的单位, 则输出的表面积单位为此 $<\text{unit}>^2$, 而输出的体积单位则是 $<\text{unit}>^2 * z_{\text{unit}}$ 。

-T[c|h]

- **-Th** 确定最大平均高度所对应的等值线
- **-Tc** 确定最大曲率 (高度 vs 等值线值) 所对应的等值线

-Z<fact>[/<shift>] 将数据减去 <shift> 再乘以比例因子 <fact>。

12.65.3 示例

计算网格表面之下的体积, 网格数据的单位是 km, 结果的单位为 km^3 :

```
gmt grdvolume hawaii_topo.nc -Sk
```

计算网格表面与等值线 $Z=250\text{m}$ 之间的体积:

```
gmt grdvolume peaks.nc -Se -C250
```

在等值线 100 到 300 范围内, 以 10 为间隔, 计算所有等值线所约束的表面积和体积:

```
gmt grdvolume peaks.nc -Sk -C100/300/10 > results.d
```

在等值线 100 到 300 范围内, 以 10 为间隔, 搜索最大平均高度 (即体积与表面积的比) 所对应的等值线值:

```
gmt grdvolume peaks.nc -Sk -C100/300/10 -Th > results.d
```

计算湖内从表面到 300 米深度范围内水的体积:

```
gmt grdvolume lake.nc -Cr-300/0
```

12.65.4 引用

如果本命令对你非常有用，可以引用如下文章：

Wessel, P., 1998, An empirical method for optimal robust regional-residual separation of geophysical data, *Math. Geol.*, **30**(4), 391-408.

12.66 greenspline

[官方文档 greenspline](#)

12.67 gshhg

[官方文档 gshhg](#)

12.68 hotspotter

[官方文档 hotspotter](#)

12.69 img2google

[官方文档 img2google](#)

12.70 img2grd

[官方文档 img2grd](#)

12.71 isogmt

[官方文档 isogmt](#)

简介 在“隔离模式”下运行 GMT 命令或脚本

12.71.1 用法

```
isogmt command
```

12.71.2 说明

该命令本质上是一个 Bash 脚本，其脚本内容为：

```
export GMT_TMPDIR=`mktemp -d ${TMPDIR:-/tmp}/gmt.XXXXXX`  
gmt $1 =  
if test $? -eq 0; then  
    gmt "$@"  
else
```

(下页继续)

(续上页)

```
"$@"
fi
rm -rf $GMT_TMPDIR
unset GMT_TMPDIR
```

该脚本做了如下几件事情：

1. 定义环境变量 \${GMT_TMPDIR} 该环境变量指定了 GMT 的临时文件目录, 接下来执行的 GMT 命令所生成的临时文件都会保存在该临时目录下
2. 检查 *isogmt* 后接的是一个 GMT 模块名还是一个脚本, 并根据不同情况分别执行该模块/脚本
3. 待命令/脚本执行完毕后, 再删除临时目录和环境变量

12.71.3 示例

隔离模式下运行一个 GMT 命令:

```
isogmt psbasemap -R0/10/0/10 -JX10c/10c -B1 > test.ps
```

隔离模式下运行一个脚本:

```
isogmt sh run.sh
```

12.71.4 已知 BUGS

1. 无法在隔离模式下执行脚本 (v5.2.1)

12.72 kml2gmt

官方文档 [kml2gmt](#)

简介 将 Google Earth 的 KML 文件转换为 GMT 表数据

该模块可以将 KML 文件中包含的点、线或多边形导出为 GMT 可用的格式。

注解: KMZ 文件本质上是一个 ZIP 压缩包, 其中包含了一个 KML 文件以及若干个辅助文件。可以将 KMZ 文件解压得到 KML 文件, 再使用该模块进行转换。

12.72.1 必选选项

<kmlfiles> 要转换的 KML 文件

12.72.2 可选选项

-F[s|l|p] 指定要输出的数据类型。默认会输出 KML 中包含的所有点、线或多边形

1. **-Fs** 只输出点
 2. **-Fl** 只输出线
 3. **-Fp** 只输出多边形
- Z** 默认只输出经纬度信息, 若使用 **-Z** 选项, 则输出坐标的高程信息作为 GMT 的 Z 值

12.72.3 示例

从 KML 文件中提取所有经纬度信息:

```
gmt kml2gmt google.kml -V > google.txt
```

从一个 KML 文件中分别提取点和多边形到不同的文件:

```
gmt kml2gmt google.kml -Fp -V > polygons.txt
gmt kml2gmt google.kml -Fs -V > points.txt
```

也可以直接用外部命令 `ogr2ogr` 实现转换:

```
ogr2ogr -f "GMT" somefile.gmt somefile.kml
```

12.73 makecpt

[官方文档 makecpt](#)

简介 制作 GMT CPT 文件

12.73.1 选项

- A[+]<transparency>** 为所有的颜色切片设置相同的透明度, <transparency> 的取值范围是 0 到 100。加上 + 则该透明度同时应用于前景色、背景色和 NaN 颜色。
- C<cpt>** 指定要进行插值的主 CPT 文件, 默认值是 `rainbow`。

可以是 GMT 自带的 CPT 文件, 也可以是用户自定义的 CPT 文件, 也可以是通过 **-C<color1>,<color2>[,<color3>]** 语法指定一个线性连续变化的颜色列表。

- T<z_min>/<z_max>[/<z_inc>[+]]** 或 **-T<ztable>** 定义要生成的 CPT 文件的 Z 值范围及 Z 值间隔

- 若使用了 **-C** 选项且 **<z_inc>** 未指定, 则 Z 值间隔的数目与输入的主 CPT 文件相同。
- 若 **<z_inc>** 后有 +, 则将 **<z_inc>** 解释为 Z 值间隔的数目而不是 Z 值间隔
- 也可以指定文件 **<ztable>**, 其中每行一个 Z 值
- 若不使用 **-T** 选项, 则默认使用主 CPT 文件中的 Z 值范围

- Z** 生成连续 CPT 文件。默认生成不连续 CPT 文件, 即每个 Z 值切片内为同一颜色

12.74 mapproject

[官方文档 mapproject](#)

- 投影正变换: 读入经纬度数据, 将其转换为指定投影类型下的 XY 坐标
- 投影逆变换: 读入 XY 坐标, 将其转换为经纬度数据

12.74.1 最小示例

最简单的, mapproject 可以用于将计算任意一点相对于当前底图左下角的偏移量:

```
$ echo 5 5 | gmt mapproject -R0/10/0/10 -JX10c/5c  
5  2.5
```

-JX10c/5c 指定了投影方式, -R0/10/0/10 指定了数据范围, 心算一下可知坐标 (5, 5) 相对于底图左下角的偏移距是 (5c, 2.5c), 也就是 mapproject 的输出。

对于地理投影, 投影公式很复杂, 心算就不可能了。mapproject 的用处就体现出来了:

```
$ echo 121 42 | gmt mapproject -R100/160/0/80 -JM10c  
3.5 7.6841597874
```

由输出可知, 坐标 (121, 42) 相对于当前底图原点的偏移距是 (3.5c, 7.68c)。

12.74.2 -I 选项

-I 选项表明做投影逆变换, 即从 XY 坐标中得到经纬度信息:

```
$ echo 3.5 7.6841597874 | gmt mapproject -R100/160/0/80 -JM10c -I  
121  42
```

12.74.3 -Q 选项

- -Qd: 列出 GMT 支持的全部大地水准面模型
- -Qe: 列出 GMT 支持的全部椭球模型
- -Q: 列出 GMT 支持的全部椭球模型和大地水准面模型

12.75 mgd77convert

官方文档 [mgd77convert](#)

12.76 mgd77header

官方文档 [mgd77header](#)

12.77 mgd77info

官方文档 [mgd77info](#)

12.78 mgd77list

官方文档 [mgd77list](#)

12.79 mgd77magref

官方文档 [mgd77magref](#)

12.80 mgd77manage

官方文档 [mgd77manage](#)

12.81 mgd77path

官方文档 [mgd77path](#)

12.82 mgd77sniffer

官方文档 [mgd77sniffer](#)

12.83 mgd77track

官方文档 [mgd77track](#)

12.84 nearneighbor

官方文档 [nearneighbor](#)

12.85 originator

官方文档 [originator](#)

12.86 project

官方文档 [project](#)

简介 将数据点投影到线或大圆路径上, 生成测线, 坐标转换

不管是要生成测线还是要将数据投影到测线上, 首先要定义测线。

测线可以用如下三种方式中的任意一种来定义:

1. 用 -C 选项定义测线的起点, 并用 -A 定义测线方位角
2. 用 -C 选项定义测线的起点, 并用 -E 选项定义测线的结束点
3. 用 -C 选项定义测线的中心, 并用 -T 选项定义 rotation pole 的位置

在指定测线后, 该命令可以将输入的 (x, y) 数据投影到该测线上, 默认的输出格式为:

x	y	z	p	q	r	s
---	---	---	---	---	---	---

其中:

- x 和 y 是数据在原坐标系下的位置,

- z 是输入数据中的其余所有列
- p 和 q 是数据 (x,y) 在测线坐标系下的位置
- r 和 s 是测线上与 (x,y) 最近的点在 XY 坐标系下的坐标

关于测线坐标系，在指定了测线后，会创建一个倾斜的坐标系。该倾斜坐标系的赤道线沿着测线，零经度线则穿过测线的中心，此时 p 是倾斜坐标系下的经度， q 是倾斜坐标系下的纬度。

12.86.1 必选选项

-C<cx>/<cy> 对于定义 1 和 2 而言，该选项指定测线的起点坐标；对于定义 3 而言，该选项指定了旋转坐标下零经线所穿过的点

12.86.2 可选选项

-A<azimuth> 定义 1 中用于指定测线的方位角

-E<bx>/<by> 定义 2 中用于指定测线的终点

-F<flags> 指定输出格式， $flags$ 可以是 $xyzpqrs$ 的任意组合。

默认输出格式为 $xyzpqrs$ 。若使用 **-G** 选项，则输出格式为 rsp

-G<dist>[/<colat>][+h] 生成测线模式。

该选项用于生成测线，此时不需要输入文件。 $<dist>$ 的单位规定见 **-Q** 选项。输出数据有三列：经度、纬度以及当前点离测线起点的距离。默认情况下，会按照大圆路径生成测线，当使用定义 2 指定测线的两个端点时，可以通过指定 $<colat>$ 来生成小圆。

若使用 $+h$ 则会在段头记录中报告 pole 的位置。

-L[w]<l_min>/<l_max> 仅坐标 p 在 $<l_{min}>$ 和 $<l_{max}>$ 之间的数据会被投影到测线上。 $<l_{min}>$ 和 $<l_{max}>$ 的单位规定见 **-Q** 选项。如果是负数则表示反方向的数据范围。若使用了 **-E** 选项，则可以使用 **-Lw** 来限制只绘制能投影到测线起点和终点之间的数据。

-N Flat Earth。在平面内使用笛卡尔坐标变换。默认使用球面三角。

-Q 使用地图单位。即投影假设 x 、 y 、 r 、 s 的单位是弧度， p 、 q 、 $<dist>$ 、 $<l_{min}>$ 、 $<l_{max}>$ 、 $<w_{min}>$ 、 $<w_{max}>$ 的单位是千米。若不使用该选项，则所有量都假设为相同的单位。

-S 将输出按照 p 增序排列。

-T<px>/<py> 定义 3 中用于指定 rotation pole 的位置

-W<w_min>/<w_max> 仅坐标 q 在 $<w_{min}>$ 和 $<w_{max}>$ 范围内的数据才会被投影到测线上。

12.86.3 示例

指定测线的起点和终点，在测线上每隔 10 千米生成一个点：

```
gmt project -C-50/10 -E-10/30 -G10 -Q > great_circle_points.xy
```

指定测线的起点和终点, 沿着 colatitude=60 的小圆上, 每隔 10 千米生成一个点:

```
gmt project -C-50/10 -E-10/30 -G10/60 -Q > small_circle_points.xy
```

利用 -F 选项指定输出哪些坐标来得到某点在某测线上的投影点:

```
echo 102 30 | gmt project -C103/31 -A225 -L0/500 -Fr -Q
```

已知某点, 根据方位角和大圆距离计算另一点。已知一点 (120, 25), 根据方位角 45 度, 大圆距离 123 千米的点位置

```
gmt project -C120/25 -A45 -L0/123 -G123 -Q | tail -1
```

12.87 psbasemap

[官方文档 psbasemap](#)

简介 绘制底图

该命令用于绘制:

- **-B** 选项绘制底图边框 (标注、刻度、标签等) 和标题
- **-L** 选项绘制比例尺
- **-T** 选项绘制方向玫瑰、磁场玫瑰图

12.87.1 必选选项

-B **-L** **-T** 三个选项必须至少存在一个。

-B 绘制底图边框。**-B** 选项是 GMT 的通用选项, 见 GMT 中文手册中相关内容。

示例:

```
gmt psbasemap -R-180/180/-70/70 -JM10c -Bx60 -By30 > test.ps
```

-L[g|j|J|n|x]<refpoint>+c[<slon>/]<slat>+w<length>[e|f|k|M|n|u][+a<align>]
在地图上指定位置绘制比例尺。

1. **[g|j|J|n|x]<refpoint>** 指定地图上的参考点, 详情见[绘制修饰物](#)一节
2. **+j<justify>** 指定比例尺上的锚点 (默认锚点为比例尺的中心), 详情见[绘制修饰物](#)一节
3. **+o<dx>/<dy>** 为比例尺设置额外的偏移量, 详情见[绘制修饰物](#)一节
4. **+c<slon>/<slat>** 要绘制哪一个点的比例尺, 对于倾斜投影, **slon** 默认取中央经线
5. **+w<length>[e|f|M|n|k|u]** 比例尺的长度, 默认单位为 km, 也可使用其他长度单位[单位](#)

6. `+a<align>` 修改比例尺标签的位置, 默认位于比例尺上方中部, 可以取 `l`、`r`、`t`、`b` 分别代表左右上下
7. `+l<label>` 为比例尺加标签; 若不指定 `<label>`, 默认的标签是比例尺所使用的长度单位
8. `+u` 比例尺的标注默认只有值没有单位, 该选项会给所有标注加上单位
9. `+f` 默认是简单的比例尺, 使用该选项则绘制 fancy 比例尺, 即黑白相间的火车轨道比例尺

除此之外还有一些 GMT 参数可以控制比例尺的外观:

1. `FONT_LABEL` 控制比例尺的标签字体
2. `FONT_ANNOT_PRIMARY` 控制比例尺的标注字体
3. `MAP_SCALE_HEIGHT` 控制比例尺的高度
4. `MAP_TICK_PEN_PRIMARY` 控制比例尺的线及刻度属性

示例:

```
gmt psbasemap -R90/110/30/40 -JM10c -Bx5 -By5 -Lg95/35+c35+w800k+lscale+u+f >
↪ test.ps

gmt psbasemap -R90/180/-50/50 -Jm0.025i -Bafg -B+tMercator -Lx1i/
↪ 1i+c0+w5000k > mercator.ps
```

-Td[g|j|J|n|x]<refpoint>+w<width>[+f[<level>]][+j<justify>][+l<w,e,s,n>][+o<d>]

在指定位置绘制方向玫瑰图

1. `[g|j|J|n|x]<refpoint>` 指定地图上的参考点, 见[绘制修饰物](#)
2. `+j<justify>` 指定玫瑰图上的锚点(默认为 MC), 见[绘制修饰物](#)
3. `+o<dx>/<dy>` 指定参考点的偏移量, 见[绘制修饰物](#)
4. `+w<width>` 玫瑰图的宽度
5. `+f<level>` 绘制 fancy 玫瑰图, `<level>` 指定了 fancy 玫瑰图的不同类型。
`<level>` 可以取:

- 1 绘制 E-W 和 N-S 四个方向
- 2 绘制 8 个方向
- 3 绘制 16 个方向

6. `+l<w>,<e>,<s>,<n>` 为四个方向分别指定标签, 默认标签是四个方向的单字母代码(英文语言下是 W、E、S、N, 具体值由参数 `GMT_LANGUAGE`), 四个方向的标签之间用逗号分隔, 比如 `+lw,e,s,n` 或 `+l",,Down,Up"`

下图展示了方向玫瑰图的效果图:

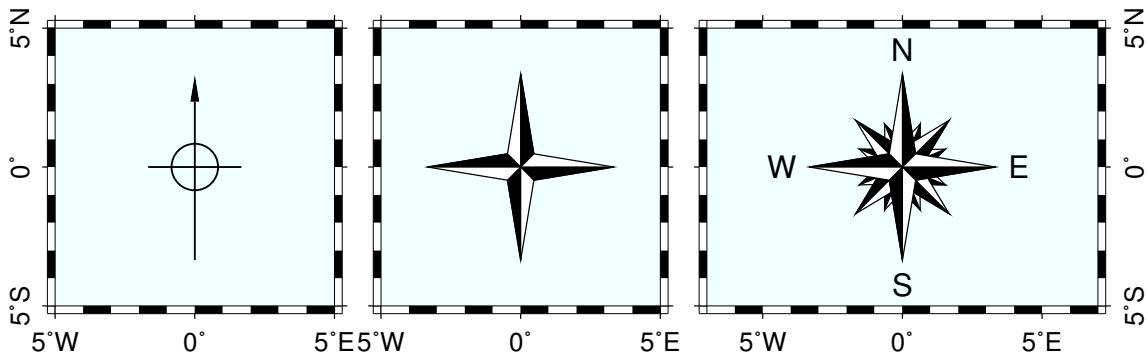


图 12.2: 方向玫瑰图

(左) -Tdg0/0+w1i+jCM (中) -Tdg0/0+w1i+f1+jCM (右) -Tdg0/0+w1i+f3+l+jCM

-Tm[g|j|J|n|x]<refpoint>+w<width>[+d<dec>[/<dlabel>]]][+i<pen>][+j<justify>]
绘制磁场玫瑰图, 用于展示磁场方向。

磁场玫瑰包括两个同心圆环, 其中外环用于展示方向信息, 内环用于显示磁场方向。

说明:

1. [g|j|J|n|x]<refpoint> 指定地图上的参考点, 见[绘制修饰物](#)
2. +j<justify> 指定玫瑰图上的锚点(默认为 MC), 见[绘制修饰物](#)
3. +o<dx>/<dy> 指定参考点的偏移量, 见[绘制修饰物](#)
4. +w<width> 玫瑰图的宽度
5. +p<pen> 绘制外环并设置其画笔属性
6. +i<pen> 绘制内环并设置其画笔属性
7. +d<dec>/<dlabel> 设置磁倾角以及罗盘指针上的磁倾角标签。若 <dlabel> 为空, 则使用默认标签 d = <dec>; 若 <dlabel> 为 -, 则不绘制标签。当使用 +d 子选项时, 会同时绘制地理方向和磁场方向
8. +l<w>,<e>,<s>,<n> 为四个方向的标签, 默认标签是四个方向的单字母代码(英文语言下是 W、E、S、N, 具体值由参数 GMT_LANGUAGE; 若 <n> 取值为 *, 则会在北方向标签处绘制星代表北极星, 也可以使用 +l,,,N 只指定 N 方向的标签)
9. 内外环都可以设置标注、刻度和网格的间隔。内外环的间隔默认值都是 30/5/1。可以使用 +t<ints> 选项, 后面接 6 个斜杠分隔的值, 以分别指定两个圆环的 3 种刻度值, 其中前三个值控制内环属性, 后三个值控制外环属性

12.87.2 可选选项

-A[<file>] 不绘制图形, 仅输出矩形底图的边框坐标。

该选项会将矩形底图的边框坐标输出到标准输出或文件中。使用该选项时, 必须通过 -J 和 -R 指定绘图区域, 且不能同时使用其他选项。若不指定 <file> 则默认输出到标准输出, 否则输出到文件 <file> 中。

说明:

1. 该选项似乎仅适用于矩形底图边框，非矩形边框会输出一堆 NaN
2. 尚不清楚该选项存在的意义
3. 边框的采样间隔由参数 MAP_LINE_STEP 决定

示例：

```
gmt psbasemap -R0/10/0/10 -JX10c/5c -Aoutline.txt
```

-D[<unit>]<xmin>/<xmax>/<ymin>/<ymax>[r] [+s<file>][+t]

-D[g|j|J|n|x]<refpoint>+w<width>[/<height>][+j<justify>][+o<dx>/<dy>][+s<file>]

在底图中绘制图中图 (inset map box)，需要 -F 选项，该选项有两种语法。

先解释图中图 (inset box) 的概念。在绘制一个小区域时，为了表明该小区域在地球上的位置，通常需要在一张更大范围（比如整个中国或整个地球）的底图上标出小区域所在的位置，这种绘图形式通常称为 inset map。而 inset box 就是 inset map 的边框。

指定 inset map box 的范围有三种方法：

1. -D<xmin>/<xmax>/<ymin>/<ymax>[r] : 类似 -R 的语法，直接指定 inset box 在地图上的范围
2. -D<unit><xmin>/<xmax>/<ymin>/<ymax> : 类似 -R 的语法，指定投影后的坐标范围，<unit> 为投影后的坐标单位
3. 指定参考点和锚点，见[绘制修饰物](#)一节

1. [g|j|J|n|x]<refpoint> 指定大区域底图上的参考点

2. +j<justify> 指定 inset box 上的锚点

3. +o<dx>/<dy> 指定参考点的额外偏移量

4. +w<width>[/<height>] 指定 inset box 的宽度或/和高度

4. +s<file> : 将 inset box 的左下角位置以及其长宽写到文件中，坐标值以及长度值均使用当前地图单位

5. +t 将绘图原点移动到 inset box 的左下角

-F[d|l|t][+c<clearances>][+g<fill>][+i[[<gap>/]<pen>]][+p[<pen>]][+r[<radius>]]

控制 inset box、比例尺和方向玫瑰的背景边框的属性

说明：

1. 该选项用于给 inset box、比例尺和方向玫瑰绘制背景边框，默认同时控制三者的属性
2. d|l|t 表示该选项定义的属性仅适用于 -D、-L 或 -T 选项
3. 直接使用 -F 选项，则绘制背景边框，边框属性由参数 MAP_FRAME_PEN 控制
4. 其他子选项的含义见[绘制修饰物](#)一节的详细介绍

12.87.3 示例

下面的脚本展示了一种 inset box 的用法：

```
#!/bin/bash
R=115/123/35/40          # 小区域地图的范围-R
J=M20c                     # 小区域的投影方式-J
PS=psbasemap_ex1.ps

# 打开 GMT
gmt psxy -R$R -J$J -T -K > $PS

# 绘制小区域的地图
gmt pscoast -R$R -J$J -B1 -Df -W2p -I2p -N1/2p -A1000 -K -O >> $PS
# ...
# 此处可以任意添加其他绘图命令
# ...

# 开始绘制 inset map
Rg=72/135/16/55          # 大区域地图的范围-R
Jg=M7c                     # 大区域地图的投影方式-J
# 绘制大区域地图的海岸线及边框
gmt pscoast -R$Rg -J$Jg -B0 -B+gwhite -Df -N1 -W -A5000 -K -O --MAP_FRAME_TYPE=plain >>
$PS
# 在大区域地图内绘制小区域所在的方框
gmt psbasemap -R$Rg -J$Jg -D$R -F+p2p,blue -K -O >> $PS

# 结束 GMT 绘图
gmt psxy -R$R -J$J -T -O >> $PS
rm gmt.*
```

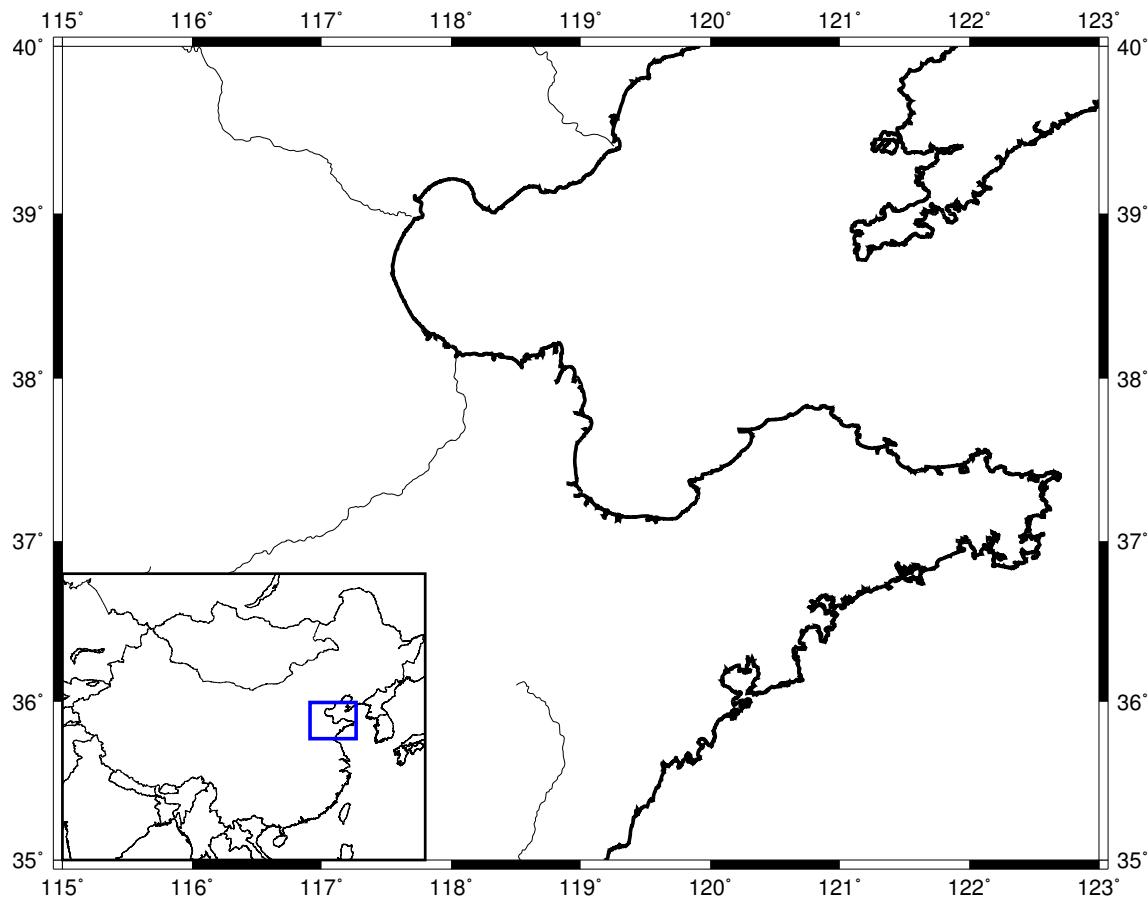


图 12.3: 用 psbasemap 命令-D 选项绘制图中图

下面的脚本绘制了磁场玫瑰图：

```
#!/bin/bash
#      $Id$
ps=psbasemap_ex4.ps
# Magnetic rose with a specified declination
gmt psbasemap -R-10/-2/12.8812380332/0.661018975345r -J0c0/0/50/60/7i -Baf -BWSne -P -K \
--X1.25i \
--MAP_ANNOT_OBLIQUE=34 --FONT_ANNOT_PRIMARY=12p > $ps
gmt psbasemap -R -J -Tmg-2/0.5+w2.5i+d-14.5+t45/10/5+i0.25p,blue+p0.25p,red+l+jCM -O -K \
--FONT_ANNOT_PRIMARY=9p,Helvetica,blue --FONT_ANNOT_SECONDARY=12p,Helvetica,red \
--FONT_LABEL=14p,Times-Italic,darkgreen --FONT_TITLE=24p --MAP_TITLE_OFFSET=7p \
--MAP_FRAME_WIDTH=10p --COLOR_BACKGROUND=green --MAP_DEFAULT_PEN=2p,darkgreen \
--COLOR_BACKGROUND=darkgreen --MAP_VECTOR_SHAPE=0.5 --MAP_TICK_PEN_ \
--SECONDARY=thinner,red \
--MAP_TICK_PEN_PRIMARY=thinner,blue >> $ps
gmt psbasemap -R -J -O -K -DjTR+w2.9i/3.9i+o0.05i -F+p+ggray95 >> $ps
echo "5.5 3.8 GMT DEFAULTS" | gmt pstext -R0/7/0/5 -Jx1i -O -K -F+f14p,Helvetica- \
Bold+jCM >> $ps
gmt pstext -R -J -O -K -F+f12p+jLM << EOF >> $ps
4.1 3.50 FONT_TITLE
4.1 3.25 MAP_TITLE_OFFSET
```

(下页继续)

(续上页)

```

4.1 3.00 MAP_DEGREE_SYMBOL
4.1 2.75 @;blue;FONT_ANNOT_PRIMARY@;;
4.1 2.50 @;blue;MAP_TICK_PEN_PRIMARY@;;
4.1 2.25 @;blue;MAP_ANNOT_OFFSET_PRIMARY@;;
4.1 2.00 @;blue;MAP_TICK_LENGTH_PRIMARY@;;
4.1 1.75 @;red;FONT_ANNOT_SECONDARY@;;
4.1 1.50 @;red;MAP_TICK_PEN_SECONDARY@;;
4.1 1.25 @;red;MAP_ANNOT_OFFSET_SECONDARY@;;
4.1 1.00 @;red;MAP_TICK_LENGTH_SECONDARY@;;
4.1 0.75 @;darkgreen;FONT_LABEL@;;
4.1 0.50 @;darkgreen;MAP_DEFAULT_PEN@;;
4.1 0.25 @;darkgreen;COLOR_BACKGROUND@;;
EOF
gmt psxy -R -J -O -T >> $ps
rm gmt.*

```

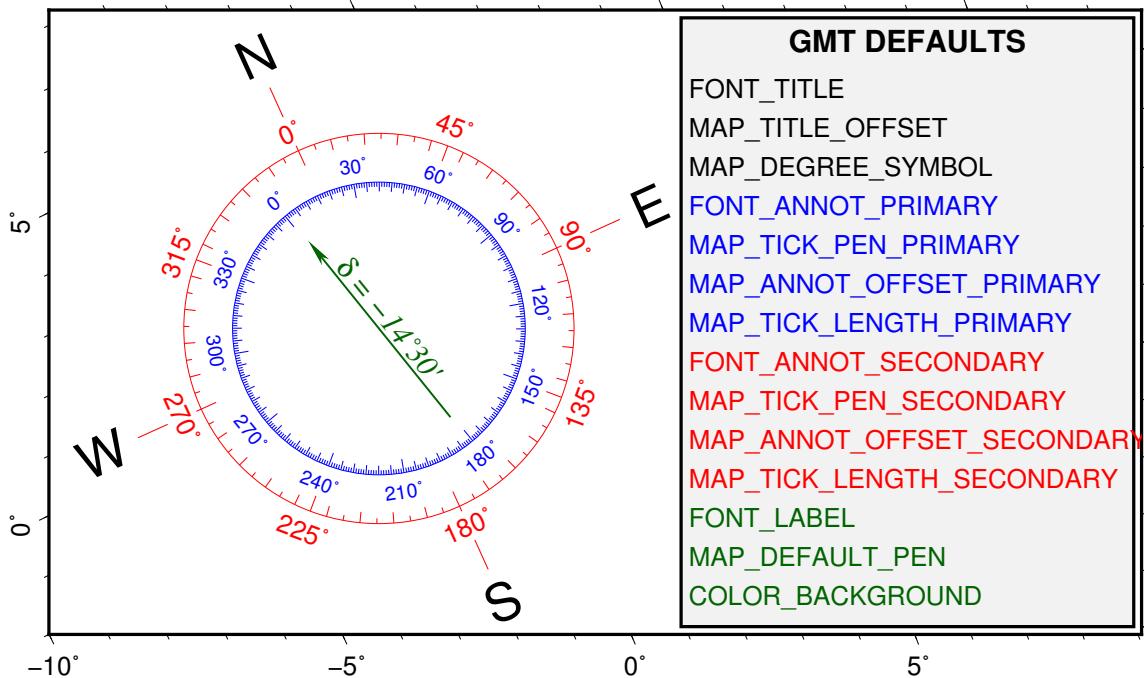


图 12.4: 磁场玫瑰图

12.87.4 BUGS

- 使用 -A 选项可以正常输出结果, 但会出现 `double free` 的错误 (v5.2.1)

12.88 psclip

官方文档 [psclip](#)

简介 打开或关闭多边形裁剪路径

该模块会从输入文件中读取 XY 数据, 由此构成一个或多个多边形, 进而构建出一个或多个裁剪路径。接下来的所有绘图命令中, 只有在多边形内部的部分才会被绘制。

为了判断某个点是在裁剪区域内还是在裁剪区域外，`psclip` 使用了“奇偶规则”。从任意一点绘制一条任意方向的射线，若该射线穿过裁剪路径线段奇数次，则该点位于裁剪区域内；若穿过偶数次，则该点位于裁剪区域外。

最后，记得再次调用 `psclip -C` 以关闭裁剪区域。

12.88.1 必选选项

`-J -R`

12.88.2 可选选项

-A[m|p|x|y] 修改两点间的连接方式，详情见 [psxy](#) 中 `-A` 选项的介绍

地理投影下，两点之间默认沿着大圆弧连接。

1. `-A`：忽略当前的投影方式，直接用直线连接两点
2. `-Am`：先沿着经线画，再沿着纬线画
3. `-Ap`：先沿着纬线画，再沿着经线画

笛卡尔坐标下，两点之间默认用直线连接。

1. `-Ax` 先沿着 X 轴画，再沿着 Y 轴画
2. `-Ay` 先沿着 Y 轴画，再沿着 X 轴画

-C[<n>] 结束当前裁剪路径。

- `-C`: 关闭所有裁剪路径
- `-C<n>`: 仅关闭当前所有处于激活状态下的裁剪路径中的其中 `<n>` 个

若在开启裁剪后有使用 `-X|-Y` 移动过坐标原点，则在关闭裁剪路径时也需要使用 `-X|-Y` 选项。

-N 反转“区域内”和“区域外”的概念，即只有在多边形外的部分才是裁剪区域，绘图时只有在多边形外的才会被绘制。该选项不能与 `-B` 选项连用。

-T 不需要任何输入数据。根据 `-R` 选项将整个地图区域裁剪出来，该选项不能与 `-B` 选项连用。

12.88.3 示例

```
# 打开裁剪路径
gmt psclip my_region.xy -R0/40/0/40 -Jm0.3i -K > clip_mask_on.ps
# 其他绘图命令
# ...

# 关闭裁剪路径
gmt psclip -C -O >> complex_plot.ps
```

12.89 pscoast

官方文档 [pscoast](#)

简介 在地图上绘制海岸线、河流、国界线

该命令除了可以用于绘制海岸线、河流、政治边界，还可以裁剪陆地区域或水域，也可以将数据导出到文件中。

12.89.1 必选选项

-J -R

12.89.2 可选选项

-A<min_area>[/<min_level>/<max_level>][+ag|i|s|S][+r|l][+p<percent>]

不绘制面积过小的区域(湖泊或岛屿)，或不绘制某个级别的湖泊边界。

在绘制湖泊时，若不管湖泊的面积大小而把所有湖泊的边界都画上去，可能导致图看上去比较乱，该选项用于对湖泊进行筛选。面积小于 `<min_area>` 平方千米或者湖泊级别不在 `[min_level,max_level]` 范围的边界都不会被绘制。默认值为 `0/0/4`，即绘制所有湖泊边界，即绘制 0 到 4 级所有级别的面积大于 0 的湖泊。

对于 `level=2`，即湖岸线，包括常规的湖以及很宽的河流。加上 `+r` 则只绘制河流，加上 `+l` 则只绘制常规湖。

对于南极洲而言，因为有冰层的存在，所以海岸线有多种处理方式：

1. `+ai` 用 ice shell boundary 作为南极洲的海岸线，默认值
2. `+ag` 以 ice grounding line 作为海岸线
3. `+as` 忽略南纬 60 度以南的海岸线，用户可以使用 `psxy` 绘制自己的南极洲海岸线
4. `+aS` 忽略南纬 60 度以北的海岸线

`+p<percent>`：一个多边形，降低精度后，边数减少，面积变化，当面积变化过大时再绘制这个多边形就不合适了，该子选项用于去除那些面积与最高精度面积之比小于 `<percent>` 的多边形。

-C[l|r/]<fill> 设置湖泊与河流湖的颜色。

默认情况下，湖泊与河流湖会被当做 wet 区域，直接使用 `-S` 指定的填充值。该选项可以为湖泊和河流湖单独指定颜色，也可以多次使用该选项分别为湖泊和河流湖指定颜色。

1. `-C<fill>` 同时指定湖泊和河流湖的颜色
2. `-Cl/<fill>` 指定湖泊(lake)颜色
3. `-Cr/<fill>` 指定河流湖(river-lake)颜色

-D[a|f|h|i|l|c][+] 选择海岸线数据精度。

GMT 自带的 GSHHG 海岸线数据有 5 个不同精度的版本，从高到低依次为：full、high、intermediate、low 和 crude。GMT 默认使用低精度数据。该选项可以指定要使用的数据精度，其中 `f|h|i|l|c` 分别代表 5 种不同的数据精度，当然也可以用 `-Da`

选项，此时 GMT 会根据当前绘图区域的大小自动选择合适的数据精度。

默认情况下，若找不到指定精度的海岸线数据，程序会自动报错退出。该选项中加上 + 则命令在找不到当前指定的精度数据时，自动寻找更低精度的数据。

-E<code1>, <code2>, ... [+l|L][+g<fill>][+p<pen>][+r|R[<incs>]]

绘制或导出行政区划边界（洲界、国界、省界）。

除了海岸线数据 GSHHG 之外，GMT 还自带了 DCW (Digital Chart of World) 数据，即全球的行政区划数据。DCW 数据位于 \${GMTHOME}/share/dcw 目录下，包含了全球各国国界和省界数据。该数据独立于 GSHHG 数据，因而 -A 和 -D 选项对该数据无效。

说明：

1. <code> 是要绘制或提取的边界数据的代码，多个代码之间用逗号分隔
2. 具体的代码参考 dcw 目录下的文档。代码有如下几种形式
 1. 洲代码前加上 = 则绘制整个洲内所有国家边界，比如 =AS 会绘制所有亚洲国家的边界
 2. 直接使用国界代码，则绘制国界边界，比如 US 绘制美国边界
 3. 使用 国家代码. 州代码，则绘制州(省)边界，比如 US.TX 绘制美国 Texas 州的边界
 3. +l 仅列出所有国家及其对应代码，不绘制边界也不提取数据
 4. +L 列出部分国界的省及其代码
 5. +g<fill> 设置多边形的填充色
 6. +p<pen> 绘制多边形的轮廓
 7. +r 获取多边形所对应的区域范围，以便于直接从数据中提取 -R 选项的范围。<incs> 可以是 <inc>、<xinc>/<yinc>、<winc>/<einc>/<sinc>/<ninc> 以调整区域范围使得范围是这些步长的整数倍
 8. +R 与 +r 类似，只是之后的 <incs> 等参数被解释为区域范围向外扩展的增量

额外的说明：

1. 除非使用了 +r、+R 或 -M 选项，否则必须指定 +p|+g 中的一个
2. -E 选项可以重复出现多次，以分别为不同的多边形设置不同的属性
3. 若使用了 +r|R 但未使用 -J|-M，则直接输出 -R<w>/<e>/<s>/<n> 格式的字符串

-F 控制比例尺和玫瑰图的背景边框，见[绘制修饰物](#)一节

-G<fill>|c 设置 dry 区域的填充色或裁剪 dry 区域

1. -G<fill> 设置 dry 区域(一般指陆地)的填充色
2. -Gc 将 dry 区域裁剪出来，使得接下来的绘图只有 dry 区域内的才会被绘制

-I<river>[/<pen>] 绘制河流。

河流 <river> 可以取：

- 0 = Double-lined rivers (river-lakes)

- 1 = Permanent major rivers
- 2 = Additional major rivers
- 3 = Additional rivers
- 4 = Minor rivers
- 5 = Intermittent rivers - major
- 6 = Intermittent rivers - additional
- 7 = Intermittent rivers - minor
- 8 = Major canals
- 9 = Minor canals
- 10 = Irrigation canals
- a = All rivers and canals (0-10)
- A = All rivers and canals except river-lakes (1-10)
- r = All permanent rivers (0-4)
- R = All permanent rivers except river-lakes (1-4)
- i = All intermittent rivers (5-7)
- c = All canals (8-10)

<pen> 的默认值为 default, black, solid , 该选项可以重复使用多次。

-L[g|j|J|n|x]<refpoint>+c[<slon>/]<slat>+w<length>[e|f|k|M|n|u][+a<align>]
在地图上指定位置绘制比例尺。

1. [g|j|J|n|x]<refpoint> 指定地图上的参考点, 详情见[绘制修饰物](#) 一节
2. +j<justify> 指定比例尺上的锚点 (默认锚点为比例尺的中心), 详情见[绘制修饰物](#) 一节
3. +o<dx>/<dy> 为比例尺设置额外的偏移量, 详情见[绘制修饰物](#) 一节
4. +c[<slon>/]<slat> 要绘制哪一个点的比例尺, 对于倾斜投影, slon 默认取中央经线
5. +w<length>[e|f|M|n|k|u] 比例尺的长度, 默认单位为 km, 也可使用其他长度单位[单位](#)
6. +a<align> 修改比例尺标签的位置, 默认位于比例尺上方中部, 可以取 l、r、t、b 分别代表左右上下
7. +l<label> 为比例尺加标签; 若不指定 <label> , 默认的标签是比例尺所使用的长度单位
8. +u 比例尺的标注默认只有值没有单位, 该选项会给所有标注加上单位
9. +f 默认是简单的比例尺, 使用该选项则绘制 fancy 比例尺, 即黑白相间的火车轨道比例尺

除此之外还有一些 GMT 参数可以控制比例尺的外观:

1. FONT_LABEL 控制比例尺的标签字体
 2. FONT_ANNOT_PRIMARY 控制比例尺的标注字体
 3. MAP_SCALE_HEIGHT 控制比例尺的高度
 4. MAP_TICK_PEN_PRIMARY 控制比例尺的线及刻度属性
- M** 将边界数据以多段 ASCII 表的形式导出到标准输出

使用该选项，则只导出数据而不绘图，该选项需要与 **-E|-I|-N|-W** 选项一起使用。
-N<border>[/<pen>] 绘制政治边界。

该选项在某些地方与 **-E** 选项有重叠。边界类型 **<border>** 可以取：

- 1 : 国界
- 2 : 州界；(目前只有美国、加拿大、澳大利亚以及南美各国的数据)
- 3 : Marine boundaries
- a : 1-3 的全部边界；

说明：

1. **<border>** 是必须值, **<pen>** 是可选值
2. 可以多次重复使用 **-N** 选项, 指定不同级别的边界
3. **<pen>** 的默认值是 `default,black,solid`

-Q 关闭区域裁剪。

使用 **-Gc** 和 **-Sc** 可以分别裁剪出 dry 区域和 wet 区域, 接下来的其他绘图命令中只有在裁剪区域内的部分才会被绘制。在绘图结束后, 需要关闭裁剪, 就需要再次调用 **pcoast**, 并加上 **-Q** 选项。若在开启裁剪后使用了 **-X** 和 **-Y** 选项, 则在关闭时也要记得使用 **-X** 和 **-Y**。

-S<fill>|c 设置 wet 区域的填充色或裁剪 wet 区域

1. **-S<fill>** 设置 wet 区域(一般指海洋或湖泊)的填充色
2. **-Sc** 将 wet 区域裁剪出来, 使得接下来的绘图只有 wet 区域内的才会被绘制

-Td[g|j|J|n|x]<refpoint>+w<width>[+f[<level>]][+j<justify>][+l<w,e,s,n>][+o<dx>/<dy>]

在指定位置绘制方向玫瑰图

1. **[g|j|J|n|x]<refpoint>** 指定地图上的参考点, 见[绘制修饰物](#)
2. **+j<justify>** 指定玫瑰图上的锚点(默认为 MC), 见[绘制修饰物](#)
3. **+o<dx>/<dy>** 指定参考点的偏移量, 见[绘制修饰物](#)
4. **+w<width>** 玫瑰图的宽度
5. **+f<level>** 绘制 fancy 玫瑰图, **<level>** 指定了 fancy 玫瑰图的不同类型。
<level> 可以取：

- 1 绘制 E-W 和 N-S 四个方向
 - 2 绘制 8 个方向
 - 3 绘制 16 个方向
6. **+l<w>,<e>,<s>,<n>** 为四个方向分别指定标签, 默认标签是四个方向的单字母代码(英文语言下是 W、E、S、N, 具体值由参数 **GMT_LANGUAGE**), 四个方向的标签之间用逗号分隔, 比如 **+lw,e,s,n** 或 **+l",,Down,Up"**

-Tm[g|j|J|n|x]<refpoint>+w<width>[+d<dec>[/<dlabel>]][+i<pen>][+j<justify>][+l<w>,<e>,<s>,<n>]

绘制磁场玫瑰图, 用于展示磁场方向。

磁场玫瑰包括两个同心圆环, 其中外环用于展示方向信息, 内环用于显示磁场方向。

说明:

1. `[g|j|J|n|x]<refpoint>` 指定地图上的参考点, 见[绘制修饰物](#)
 2. `+j<justify>` 指定玫瑰图上的锚点(默认为 MC), 见[绘制修饰物](#)
 3. `+o<dx>/<dy>` 指定参考点的偏移量, 见[绘制修饰物](#)
 4. `+w<width>` 玫瑰图的宽度
 5. `+p<pen>` 绘制外环并设置其画笔属性
 6. `+i<pen>` 绘制内环并设置其画笔属性
 7. `+d<dec>/<dlabel>` 设置磁倾角以及罗盘指针上的磁倾角标签。若 `<dlabel>` 为空, 则使用默认标签 `d = <dec>`; 若 `<dlabel>` 为 -, 则不绘制标签。当使用 `+d` 子选项时, 会同时绘制地理方向和磁场方向
 8. `+l<w>,<e>,<s>,<n>` 为四个方向的标签, 默认标签是四个方向的单字母代码(英文语言下是 W、E、S、N, 具体值由参数 `GMT_LANGUAGE`; 若 `<n>` 取值为 *, 则会在北方向标签处绘制星代表北极星, 也可以使用 `+l,,,N` 只指定 N 方向的标签)
 9. 内外环都可以设置标注、刻度和网格的间隔。内外环的间隔默认值都是 `30/5/1`。可以使用 `+t<ints>` 选项, 后面接 6 个斜杠分隔的值, 以分别指定两个圆环的 3 种刻度值, 其中前三个值控制内环属性, 后三个值控制外环属性
- W[<level>/]<pen>** 绘制湖岸线(shoreline)。

shore 指水与陆地交界的“岸”(如: 海岸、湖岸、河岸等), 是一个较为笼统的说法。

GMT 将 shoreline 分成四个等级(<level> 取 1-4):

1. coastline: 海岸线
2. lakeshore: 湖泊与陆地的岸线
3. island-in-lake shore: 首先要有陆地, 陆地中有个湖, 湖里有个岛。即岛的岸线
4. lake-in-island-in-lake shore: 首先有陆地, 陆地中有个湖, 湖中有个岛, 岛里又有湖。这里指的是湖的岸线

使用时需要注意:

1. 不使用 `-W` 选项, 则不绘制任何 shore
2. 使用 `-W`, 给定画笔属性 `<pen>`, 但不给出 `<level>`, 则绘制四个 level 的 shore
3. 可以用 `-W<level>/<pen>` 的方式指定要绘制哪一个 level 的 shore, 并指定线条属性, 在同一个命令中可以多次使用 `-W`, 以指定不同 level 的 shore 的画笔属性
4. `-W` 选项中 `<level>` 是可选的, 而 `<pen>` 是必须的! 因而 `-W2` 会被解释为所有 level 的画笔属性, 而不是 level 2

12.89.3 示例

```
gmt pscoast -R-30/30/-40/40 -Jm0.1i -B5 -I1/1p,blue -N1/0.25p,- \
-I2/0.25p,blue -W0.25p,white -Ggreen -Sblue -P > africa.ps
```

```
gmt pscoast -R-30/-10/60/65 -Jm1c -B5 -Gp100/28 > iceland.ps
```

将非洲区域裁剪出来，并在其中的陆地部分绘制地形：

```
gmt pscoast -R-30/30/-40/40 -Jm0.1i -B5 -Gc -P -K > africa.ps
gmt grdimage -Jm0.1i etopo5.nc -Ccolors.cpt -O -K >> africa.ps
gmt pscoast -Q -O >> africa.ps
```

绘制部分国家的国界线（似乎有 BUG）：

```
gmt pscoast -JM6i -P -Baf -EGB,IT,FR+gblue+p0.25p,red+r -EES,PT,GR+gyellow > map.ps
```

提取冰岛的海岸线数据：

```
gmt pscoast -R-26/-12/62/68 -Dh -W -M > iceland.txt
```

12.89.4 FAQ

1. 错误消息：

```
pscoast: low resolution shoreline data base not installed.
```

出现该错误的原因有如下几种：

1. 未安装 GSHHG 海岸线数据
2. 安装了但路径不正确（建议的做法是把所有 GSHHG 的文件放在 \$GMTHOME/share/coast 目录下）
3. 安装的 netCDF 版本号为 3.x 而不是 4.x
4. 自行编译了 netCDF 4.x，且编译时使用了 --disable-netcdf4 选项

12.90 pscontour

[官方文档 pscontour](#)

12.91 psconvert

[官方文档 psconvert](#)

简介 将 GMT 生成的 PS/EPS 文件转换为其他图片格式

注解： 本模块原名为 `ps2raster`，v5.2.1 版本起更名为 `psconvert`

该模块通过调用 GhostScript 将 PS/EPS 文件转换为其他图片格式，包括 BMP、EPS、JPEG、PDF、PNG、PPM、SVG、TIFF 格式。

12.91.1 必选选项

psfiles 要转换格式的 PS 文件名

默认情况下, 转换后的文件与转换前的文件有相同的文件名, 文件后缀由文件格式决定。

如下命令会将 PS 文件转换成 JPG 格式:

```
gmt psconvert test.ps
gmt psconvert test1.ps test2.ps map*.ps
```

12.91.2 可选选项

-A[u][<margins>][-][+p<paint>][+p[<pen>]][+r][+s[m]|S<width>[<u>]/<height>]
对输出的图片做裁边

默认情况下, 转换得到的图片的大小由 PS 文件的纸张尺寸决定。通常画图的时候是不会把一张 A4 纸画满的, 所以在图片周围就会出现多余的白色部分。

-A 选项会对 PS 文件进行裁剪, 仅保留其中有绘图的部分:

```
gmt psconvert -A test.ps
```

-Au 会先将 GMT 的时间戳去掉再裁剪。

默认的裁剪方式会将图片裁剪到尽可能小。如果想要图片周围有额外的白色区域, 可以加上额外的边距 <margins>。其中 <margins> 可以取:

- 一个数, 四条边的额外边距相同, 如 -A0.5c
- 两个数字, 分别指定 X 和 Y 方向的额外边距, 如 -A0.5c/1c
- 四个数字, 分别指定上下左右四条边的边距, 如 -A0.5c/0.5c/0.5c/0.5c

-A+s 可以指定最终图片的尺寸:

- A+s<width> 指定最终生成的图片的宽度, 高度自动决定。程序会对图片做插值以保证 -E 的值
- A+S<scale> 指定图片的缩放比例
- A+sm<width>/<height> 设置图片所允许的最大尺寸。若原始图片的宽度不大于 <width> 则使用图片的原始尺寸, <height> 同理。

如下命令会生成一个宽为 5 厘米的图片:

```
gmt psconvert -A+s5c test.ps
```

- A+r 使得在计算边界时使用 round 函数而不是 ceil 函数, 这会对裁剪造成极其微小的区别, 仅当要处理非常小的图片时才需要使用。
- A+g<paint> 为 BoundingBox 指定填充色

- **-A+p<pen>** 为 BoundingBox 指定边框颜色, 默认颜色为 `0.25p,black`
- **-A-** 覆盖 **-W** 选项中自动设置的 **-A** 值

-C<gs_options> 传递给 GhostScript 的选项

该选项用于在调用 GhostScript 时传给 GhostScript 额外的选项, 若要额外给 GhostScript 增加多个选项, 可重复使用 **-C** 命令。

在 Windows 下, 若 PS 文件中含中文, 则可能需要使用 **-C** 选项告诉 `ghostscript` 字体路径:

```
gmt psconvert -C-sFONTPATH=C:\Windows\Fonts chinese.ps
```

-D<outdir> 设置输出目录

默认情况下, 会在 PS 文件同一目录中生成其他图片文件, 使用 **-D<outdir>** 选项可以指定输出目录, **-D.** 表示在当前目录输出。需注意, 输出目录必须已存在, 否则会报错。

-E<dpi> 设置图片精度

值越大, 图片越清晰, 文件也越大。PDF 格式默认值为 720, 其他格式默认值为 300, 单位为 dpi。

```
gmt psconvert -Tj -E600 test.ps
```

说明:

1. EPS 是矢量图片格式, **-E** 选项对其无效
2. PDF 是矢量图片格式, **-E** 选项对其中的 pattern 和字体有效

-F<out_name> 指定输出的文件名

默认情况下, 该命令会直接使用输入的 PS 文件名, 并修改其后缀作为输出文件的文件名。比如 `test.ps` 转换出的图片则为 `test.jpg`、`test.png` 之类的。

-F 选项可强制指定输出文件名, 文件后缀由输出的文件格式自动决定:

```
gmt psconvert -Tf -Fout test.ps
```

-G<ghost_path> 指定 GhostScript 可执行文件的路径

`psconvert` 底层调用了 GhostScript 来实现 PS 到其他格式的转换, 因而成功转换的前提是必须能够找到 GhostScript 的可执行文件。**-G** 选项即用于显式指定 `ghostscript` 可执行文件的路径。

说明:

1. Linux 下一般不需要设置 `ghostscript` 的路径, 除非你自己重新编译了 `gs` 并安装到了非标准路径下
2. Windows 下, 一般也不需要使用该选项, 程序会自动从注册表里获取路径信息
3. 如果从注册表中获取路径失败, 则必须指定 **-G** 选项, 例如

`-Gc:\programs\gs\gs9.02\bin\gswin64c`

-I Enforce gray-shades by using ICC profiles. GhostScript versions ≥ 9.00 change gray-shades by using ICC profiles. GhostScript 9.05 and above provide the ‘-dUseFastColor=true’ option to prevent that and that is what **psconvert** does by default, unless option **-I** is set. Note that for GhostScript ≥ 9.00 and < 9.05 the gray-shade shifting is applied to all but PDF format. We have no solution to offer other than upgrade GhostScript.

-L<listfile> <listfile> 中包含要转换的所有 PS 文件名

-P 强制转换后的图片为 Portrait 模式。

若 PS 文件在绘图时使用了 **-P** 选项，则 **psconvert** 的 **-P** 选项无效；若 PS 文件在绘图时未使用 **-P** 选项，则 **psconvert** 的 **-P** 选项会将图片旋转 90 度。

-Q[g|t][1|2|4] Set the anti-aliasing options for graphics or text. Append the size of the subsample box (1, 2, or 4) [4]. Default is no anti-aliasing (same as *bits* = 1).

-S 在执行 `ghostscript` 命令后，将具体的命令打印到标准错误流中，且保留转换过程中的所有临时文件。该选项主要用于调试。

-Tb|e|E|f|F|j|g|G|m|s|t 指定要转换的图片格式。可以接如下值：

- **b** : BMP;
- **e** : EPS;
- **E** : 带有 `PageSize` 命令的 EPS;
- **f** : PDF;
- **F** : 多页 PDF;
- **j** : JPEG (默认值);
- **g** : PNG;
- **G** : 透明 PNG;
- **m** : PPM;
- **s** : SVG;
- **t** : TIFF;

说明：

1. **g** 和 **G** 的区别在于前者背景色为白色，后者背景色为透明；
2. 对于 `bjgt`` 格式可以在其后加```-` 将 PS 文件转换为灰度图；
3. EPS 格式可以与其他格式合在一起使用。比如 **-Tef** 会同时生成 EPS 和 PDF 文件。除此之外，该命令一次只能转换一种格式，比如 **-Tbf** 则只会生成 PDF 格式（即以 **-T** 选项中的最后一个格式为准）
4. **-TF** 会将多个 PS/PDF 文件转换并合并成一个多页的 PDF 文件，需要使用 **-F** 选项指定输出的文件名

转换为 PDF 格式：

```
gmt psconvert -Tf test.ps
```

转换为 JPG 格式的灰度图:

```
gmt psconvert -Tj test.ps
```

利用一堆 PS 文件生成一个多页 PDF:

```
gmt psconvert -TF -Fout.pdf *.ps
```

-W 见官方文档

-Z 转换完成后删除输入的 PS 文件。若转换失败，输入的 PS 文件不会被删除。

12.91.3 其他

1. 转换为 PDF、PNG 时使用 DEFALTE 压缩；转换为 TIFF 时使用 LZW 压缩；
2. 对于小 dpi 的图片而言，可以使用 **-Qt4 -Qg4** 使得文字和线条看上去更加尖锐；
3. psconvert 还可以用于其他命令生成的 PS 文件；
4. 新版本的 Word 已经支持直接插入 PDF 格式的图片，高清且文件小，推荐使用；

12.91.4 FAQ

1. 错误信息:

```
GMT PS format detected but file is not finalized. Maybe a -K in excess? No output  
↳ created.
```

出现这种错误的最常见原因是最后一个绘图命令里多了一个 **-K**，即本该只使用 **-O** 选项关闭 PS 文件的，但却使用了 **-K -O** 选项。

12.91.5 BUGS

1. Windows 下会自动为文件名加双引号，导致无法找到要转换的 PS 文件 (v5.1.1)
2. 使用 **-A** 选项时，**-C** 选项无效，导致无法在 Windows 转换含中文的 PS 文件 (v5.1.1)
3. 不能使用 **-F** 选项，即不能生成多页 PDF (v5.1.2)
4. **-Z** 选项无效 (v5.2.1)

12.92 pscoupe

官方文档 [pscoupe](#)

简介 绘制震源机制解的剖面图

psmeca 在绘制震源球时，本质上是取了一个水平剖面，并将三维震源球的下半球投影到该水平剖面上。而 *pscoupe* 则更灵活一些，可以将三维震源球投影到任意一个剖面上。

- 对于一个水平剖面，会将下半球投影到平面上（即 *psmeca* 的做法）
- 对于一个垂直剖面，会将垂直平面后的半球投影到平面上
- 对于任意一个非水平的平面而言：

- 北方向为平面的最速下降方向
- 东方向为平面的走向方向
- 下方向则根据右手定则确定

12.92.1 剖面类型

-A 选项用于确定剖面。

`-Aa<lon1>/<lat1>/<lon2>/<lat2>/<dip>/<p_width>/<dmin>/<dmax>[f]`

- `<lon1>/<lat1>` 剖面起点的经纬度
- `<lon2>/<lat2>` 剖面终点的经纬度
- 起点和终点确定了剖面的长度
- `<dip>` 剖面所在平面的倾角 (0 表示水平剖面, 90 表示垂直剖面)
- `<p_width>` 剖面的宽度 (即剖面不是一个平面, 而是一个有厚度的长方体)
- `<dmin>/<dmax>` 是沿着最速下降方向 (“北” 方向) 的最小、最大距离
- `f` 表示根据剖面的参数自动计算边框的范围
- TODO: `<p_width>/<dmin>/<dmax>` 的具体含义尚不明确

`-Ab<lon1>/<lat1>/<strike>/<p_length>/<dip>/<p_width>/<dmin>/<dmax>[f]`

- `<lon1>/<lat1>` 剖面起点的经纬度
- `<strike>` 是剖面的走向
- `<p_length>` 是剖面的长度
- 其他参数与 -Aa 相同

`-Ac<x1>/<y1>/<x2>/<y2>/<dip>/<p_width>/<dmin>/<dmax>[f]`

与 -Aa 选项相同, 只是 `<x>/<y>` 为笛卡尔坐标而不是地理坐标

`-Ad<x1>/<y1>/<strike>/<p_length>/<dip>/<p_width>/<dmin>/<dmax>[f]`

与 -Ab 选项相同, 只是 `<x>/<y>` 为笛卡尔坐标而不是地理坐标

12.92.2 选项

-E<color> 扩张部分的填充色, 默认为白色

-F<mode>[<args>] 设置多个属性, 可重复使用多次

- `-Fs<symbol>[<size>/<fontsize>[/<offset>[<u>]]]` 见官方文档
- `-Fa[<size>] [/<P_axis_symbol>[<T_axis_symbol>]]` 计算并绘制 P 和 T 轴的符号。`<size>` 是符号大小, 符号与 `psxy` 类似, 可以取 `c|d|h|i|p|s|t|x`, 默认值为 `6p/cc`
- `-Fe<fill>` 设置 T 轴符号的填充色
- `-Fg<fill>` 设置 P 轴符号的填充色

- **-Fp<pen>** P 轴轮廓的画笔属性
- **-Ft<pen>** T 轴轮廓的画笔属性
- **-Fr<fill>** 在标签后加一个方框

-G<color> 指定压缩部分的填充色, 默认值为黑色

-L[<pen>] 设置震源球外部轮廓的线条属性

-M 所有震级使用相同的大小, 具体大小由 **-S** 选项的 **<scale>** 参数决定。

-N 地图区域外的震源球也要绘制, 默认不绘制。

-Q 默认会生成一些临时文件, 其中包含了剖面和剖面上的震源机制的信息, 使用该选项, 则不会生成这些临时文件。

-T<num_of_planes>[/<pen>] 绘制断层平面。**<num_of_planes>** 可以取:

- 0 绘制两个断层面
- 1 绘制第一个断层面
- 2 绘制第二个断层面

-S 选择震源机制解的格式。与 *psmeca* 中 **-S** 选项用法相同

唯一的不同在于, 输入文件中 **newX** 和 **newY** 对该命令无效, 这里加上 **newX** 和 **newY** 只是为了是的该命令可以使用与 *psmeca* 相同的输入文件。

-W<pen> 设置断层边界的画笔属性

-Z<cpt> 指定 CPT 文件, 根据数据文件中第三列的值 (即地震深度) 确定震源球的压缩部分的颜色。

12.92.3 示例

下面的示例展示了 **-Aa** 选项的用法, 指定 **P1** 和 **P2** 为剖面的起点和终点, 并展示了不同倾角时的震源球的效果。

```
#!/bin/bash
ps=pscoupex1.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i \
MAP_FRAME_WIDTH 0.1i MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i

# Plotting 2 mechanisms on map
gmt psmeca -R128/130/10/11.1 -JX2i -Fa0.1i/cc -Sc0.4i -B1 -Y8.5i -P -K << EOF > $ps
# lon  lat  dep  str  dip  rake  str  dip  rake  m  ex  nx  ny
129.5  10.5 10  0   90   0   90   90  180  1 24  0   0  10km
128.5  10.5 40  0   45   90  180   45  90   1 24  0   0  40km
EOF
(echo 128 11; echo 130 11) | gmt psxy -R -J -K -O -W0.25p,red >> $ps
gmt pstext -R -J -N -F+f14p,Helvetica-Bold+j -K -O << EOF >> $ps
128 11 ML P1
130 11 MR P2
EOF

# Represent cross-sections between points P1(128E11N) and P2(130E/11N)
# on a plane the dip of which varies
# from quasi horizontal to vertical.
# y dimension is counted along steepest descent on the plane
# so the values of depth are only in the vertical cross-section.
```

(下页继续)

(续上页)

```

# Variation of dip for cross-section plane
# (WE azimuth, between points (128E,11N) and (130E,11N))

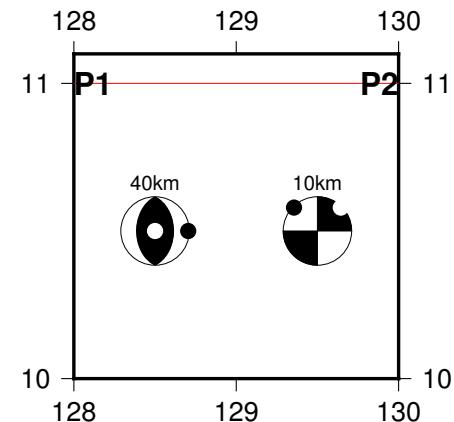
plots () {
y_offset=-2.5i
for d in $1 $2 $3 ; do
gmt pscoupe -R0/200/0/100 -Jx1.5i/-1.5i -Bxa100f10 -Bya50f10 -BWesN \
-Q -L -Sc0.4 -Aa128/11/130/11/$d/60/0/100f -Ggrey -Fa0.1i/cc $4 $5 \
-Y$y_offset -X$x_offset -O -K << EOF
# lon    lat   dep str dip rake str dip rake m ex nx ny
129.5 10.5 10  0    90   0   90   90 180  1 24   0   0 10km
128.5 10.5 40  0    45   90 180   45 90   1 24   0   0 40km
EOF
gmt pstext -R -J -F+f18p,Helvetica-Bold+jBL -O -K <<< "10 15 $d"
y_offset=0i
x_offset=2.5i
done
x_offset=-5i
}

x_offset=0i

plots 10 20 30 >> $ps
plots 40 50 60 >> $ps
plots 70 80 90 -N >> $ps

gmt pstext -X-5i -R0/10/0/15 -Jx1i -F+jBL+fHelvetica-Bold+f -O << EOF >> $ps
3 8.5 24 Variation of dip
3 8.0 20 W-E cross-section
EOF
rm gmt.*

```



Variation of dip W-E cross-section

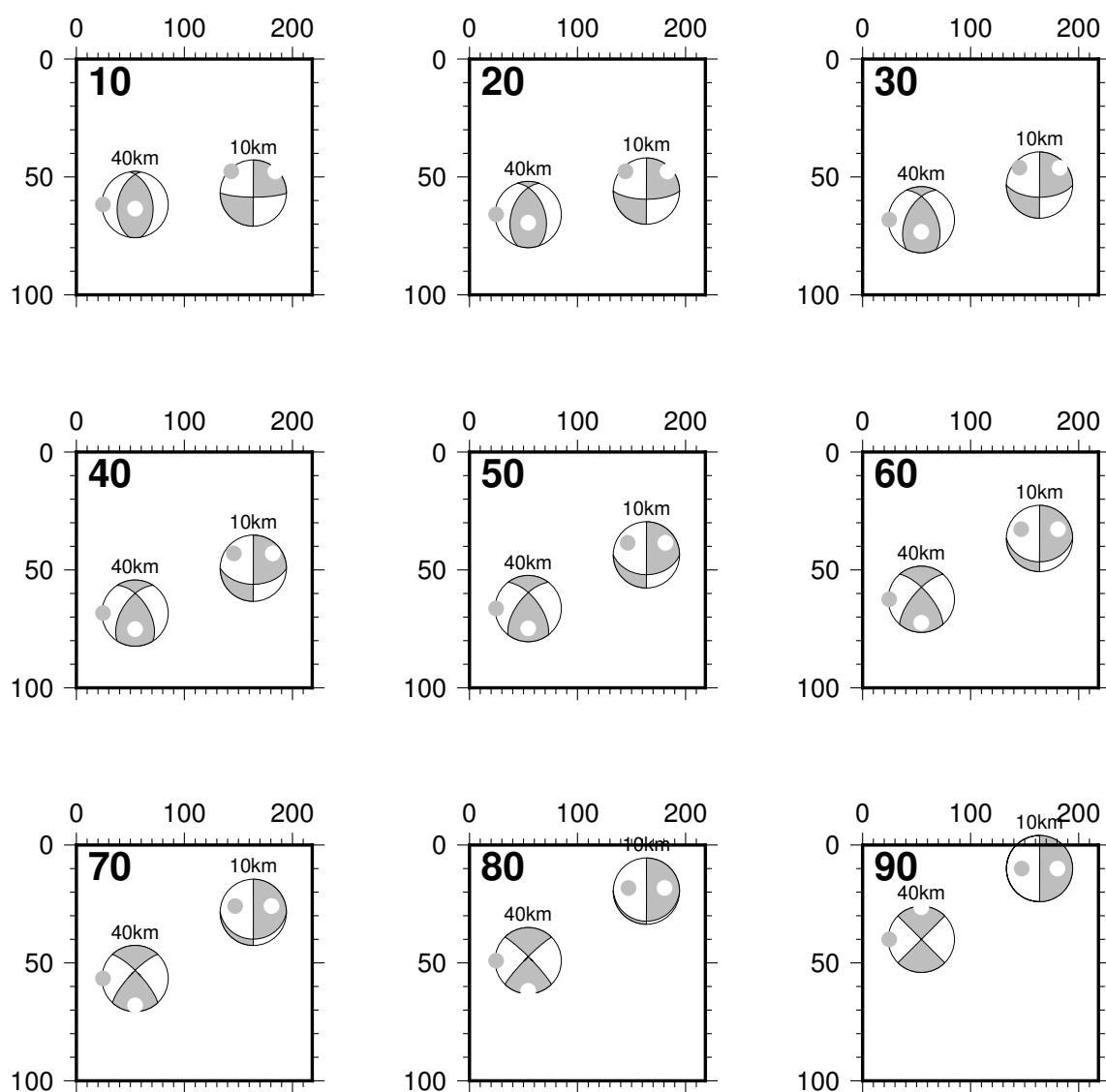


图 12.5: pscoupe 示例图 1

下面的示例将震源球投影到不同方位角的剖面上的效果：

```

#!/bin/bash
# test/meca/meca_4.sh

ps=pscoupex2.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i MAP_FRAME_WIDTH 0.1i \
MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i

# Plotting 2 mechanisms on map
gmt psmeca -R128/130/10/11.1 -JX2i -Fa0.1i/cc -Sc0.4i -B1 -Y8.5i -P -K << EOF > $ps
# lon lat dep str dip rake str dip rake m ex nx ny
129.5 10.5 10 0 90 0 90 90 180 1 24 0 0 10km
128.5 10.5 40 0 45 90 180 45 90 1 24 0 0 40km
EOF
(echo 128 11; echo 130 11) | gmt psxy -R -J -K -O -W0.25p,red >> $ps
gmt pstext -R -J -N -F+f14p,Helvetica-Bold+j -K -O << EOF >> $ps
128 11 ML P1
130 11 MR P2
EOF

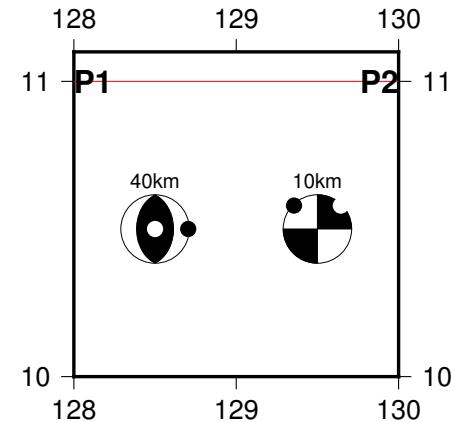
plots () {
y_offset=-2.5i
for a in $1 $2 $3 ; do
gmt pscoupe -R0/250/0/100 -JX1.5i/-1.5i -Bxa100f10 -Bya50f10 -BWesN \
-Q -L -Sc0.4 -Ab$4/$5/$a/250/90/$6/0/100f -Ggrey -Fa0.1i/cc $7 $8 \
-Y$y_offset -X$x_offset -O -K << EOF
# lon lat dep str dip rake str dip rake m ex nx ny
129.5 10.5 10 0 90 0 90 90 180 1 24 0 0 10km
128.5 10.5 40 0 45 90 180 45 90 1 24 0 0 40km
EOF
gmt pstext -R -J -F+f18p,Helvetica-Bold+jBR -O -K <<< "240 90 $a"
y_offset=0i
x_offset=2.5i
done
x_offset=-5i
}

x_offset=0i

plots 0 40 80 128 10.0 200 >> $ps
plots 120 160 200 128 11.0 400 -N >> $ps
plots 240 280 320 130 10.5 200 -N >> $ps

gmt pstext -X-5i -R0/10/0/15 -Jx1i -F+jBL+fHelvetica-Bold+f -O << EOF >> $ps
3 8.5 24 Variation of azimuth
3 8.0 20 vertical cross-section
EOF
rm gmt.*

```



Variation of azimuth vertical cross-section

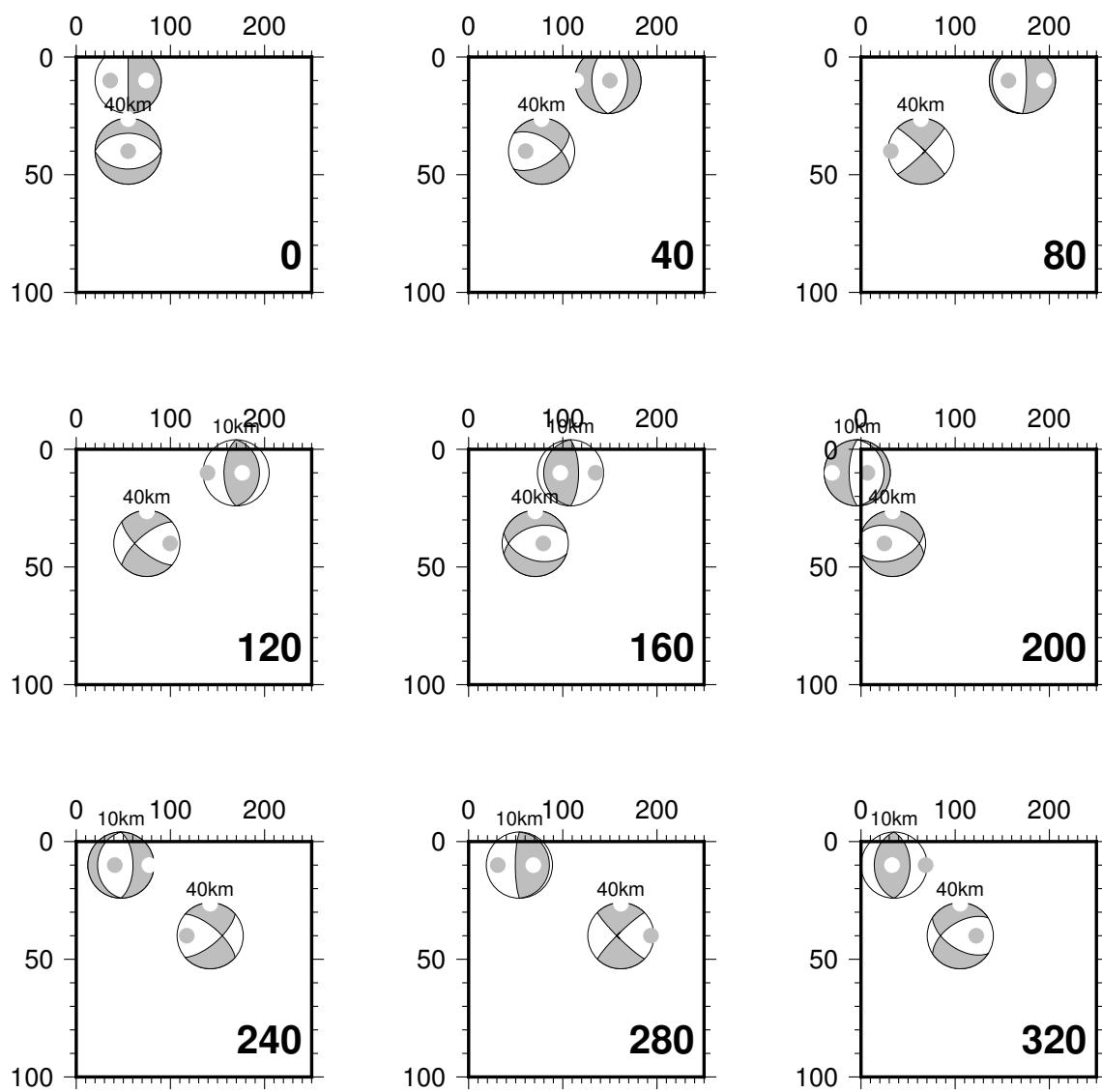


图 12.6: pscoupe 示例图 2

12.93 pshistogram

官方文档 [pshistogram](#)

简介 统计并绘制直方图

该命令会读取数据中的第一列, 对其进行统计, 并绘制直方图或累积直方图。

12.93.1 必选选项

-W<bin_width>[+l|h|b] 设置直方图计算时的 bin 宽度

[+l|h|b] 用于控制落在 -R 范围外的数据点的处理方式。默认情况下, 这些数据会被忽略

- +b 将落在范围外的数据包含在第一个或最后一个 bin 中
- +l 仅将小于第一个 bin 的数据放在第一个 bin 中
- +h 仅将大于最后一个 bin 的数据放在最后一个 bin 中

12.93.2 可选选项

-A 绘制水平直方图, 默认绘制垂直直方图

-C<cpt> 指定 CPT 文件, 将每个 bar 的中间值作为 Z 值查询 CPT 中的颜色

-D[+b][+f][+o<offset>][+r] 为每个 bar 添加标注, 标注内容是每个 bar 的统计数目

- +b 将标注放在 bar 的下边 (默认放在上边)
- +f 设置标注的字体
- +o<offset> 修改 bar 与标注的距离 (默认值为 6p)
- +r 将标注从水平方向旋转为垂直方向

-F center bin on each value (默认是左边界)

假设数据范围是 0 到 100, bin 宽度为 10。默认情况下, 会将 0 到 10 作为第一个 bin, 10 到 20 作为第二个 bin, 以此类推。若使用该选项, 则第一个 bin 以 0 为中心, 即 0 到 5 是第一个 bin, 5 到 15 是第二个 bin, 以此类推。

-G<fill> 设置 bar 的填充色

-I[o|O] 返回计算结果不绘图。

- -I 返回 xmin xmax ymin ymax
- -Io 返回 bin 的 X 和 Y 值
- -IO 返回 bin 的 X 和 Y 值, 即使 Y=0

-L<pen> 设置 bar 的边框的画笔属性

-N[<mode>][+p<pen>] 绘制等效的正态分布曲线

<mode> 用于设定正态分布的中间位置及比例:

- mode=0: mean and standard deviation (default)
- mode=1: median and L1 scale
- mode=2: LMS mode and scale

<pen> 用于指定曲线的属性。该选项可以使用多次以绘制多条曲线。

-Q 绘制累积直方图

-S 绘制阶梯状直方图。

默认的直方图中, 每个 bin 都用一个 bar 表示。使用该选项, 则去除 bar 内部的线条。

-Z[<type>][+w] 选项直方图的种类

- type=0: counts (默认值)
- type=1: frequency_percent
- type=2: log (1.0 + count)
- type=3: log (1.0 + frequency_percent)
- type=4: log10 (1.0 + count)
- type=5: log10 (1.0 + frequency_percent).

若要使用第二列数据而不是 count 数作为权重, 可以加上 **+w** 选项。

若需要绘制其他种类的直方图, 则只能先自己对输入数据做处理再绘图。

12.94 psimage

[官方文档 psimage](#)

简介 将图片或 EPS 文件放在地图上

该命令可以读取一个 EPS 文件或光栅图片格式, 并将其画在地图上。

该命令可以用于:

1. 将多张图合并到一张图上
2. 将自己单位的 logo 放在 GMT 生成的图上
3. 将一般图片放在图上

12.94.1 必选选项

<imagefile> EPS 文件或其他光栅图片格式 (GIF、PNG 等) 的文件

- EPS 文件必须包含合适的 BoundingBox
- 光栅文件的颜色深度可以是 1、8、24、32 位
- 光栅文件是通过 GDAL 读入的, 若安装 GMT 时未配置 GDAL, 则该命令只支持 EPS 文件

注解: `psimage` 并不支持 PS 文件, 建议使用如下命令将 PS 文件转化为 EPS 文件:

```
gmt psconvert -A -P -Te xxx.ps
```

12.94.2 可选选项

-D[g|j|J|n|x]<refpoint>+r<dpi>+w[-]<width>[/<height>][+j<justify>][+n<nx>]
指定图片的尺寸和位置

1. [g|j|J|n|x]<refpoint> 指定底图上的参考点, 见[绘制修饰物](#)一节
2. +r<dpi> 指定图片的 DPI 以间接指定图片的尺寸
3. +w[-]<width>[/<height>] 直接指定图片的尺寸。若未给定 <height> 则按照 <width> 以及原图的纵横比进行缩放; 若 <width> 为负值, 则使用其绝对值作为宽度, 并使用 PS 的图片操作符将图片插值到设备的分辨率
4. +j<justify> 指定图片的锚点, 默认锚点是 BL, 见[绘制修饰物](#)一节
5. +o<dx>[/<dy>] 指定参考点的额外偏移量, 见[绘制修饰物](#)
6. +n<nx>/<ny> 使图片在水平方向重复 <nx> 次, 垂直方向重复 <ny> 次, 若省略 <ny> 则默认其与 <nx> 相等, 默认值为 1/1

-F[+c<clearance(s)>][+g<fill>][+i[[<gap>/]<pen>]][+p[<pen>]][+r[<radius>]]

为图片加上背景边框, 见[绘制修饰物](#)一节

- +p<pen> 面板边框的画笔属性
- +g<fill> 面板填充色
- +c<clearance> 设置 logo 与面板边框之间空白区域的大小
- +i<gap>/<pen> 为背景面板加上额外的内边框
- +r<radius> 面板使用圆角矩形边框
- +s<dx>/<dy>/<fill> 为面板增加阴影区

-M 使用 YIQ 变换将彩图转换成灰度图

-G[b|f|t]<color> 对光栅图片设置颜色属性

1-bit 图片默认为黑色和白色, 可以通过如下选项进行修改:

- -Gb<color> 设置背景色, 即将白色替换成其他颜色
- -Gf<color> 设置前景色, 即将黑色替换成其他颜色
- <color> 可以取 - , 表示透明色

对于 8、24、32 位图片而言:

- -Gt<color> 将某个特定颜色设置为透明

-I 绘图前对 1-bit 图片进行反转, 即黑色变白色, 白色变黑色

12.94.3 示例

```
gmt psimage logo.jpg -Dx0/0+w1i -F+pthin,blue > image.ps
```

```
gmt psimage tiger.eps -Dx2i/1i+jTR+w3i > image.ps
```

```
gmt psimage 1_bit.ras -Gbbrown -Gfred -Dx0/0+w1c+n5 > image.ps
```

12.95 pslegend

官方文档 [pslegend](#)

简介 在图上添加图例

12.95.1 选项

-C<dx>/<dy> 设置图例边框与内部图例之间的空白, 默认值为 4p/4p

-D[g|j|J|n|x]<refpoint>+w<width>[/<height>][+j<justify>][+l<spacing>][+o<dx>]

设置图例的位置和大小

- [g|j|J|n|x]<refpoint> 指定底图上的参考点, 见[绘制修饰物](#)
- +w<width>/<height> 用于指定图例框的尺寸, 若 <height> 未指定或取 0, 则 GMT 会自动估算所需的高度
- +j<justify> 指定图例上的锚点, 默认锚点是 BL, 见[绘制修饰物](#) 一节
- +o<dx>/<dy> 指定参考点的额外偏移量
- +l<spacing> 行间距因子, 默认值为 1.1, 则当前字体大小的 1.1 倍

该选项几个比较有用的用法是:

1. 将图例放在左下角: -DjBL+w4c+o0.2c/0.2c
2. 将图例放在左上角: -DjTL+w4c+o0.2c/0.2c
3. 将图例放在右下角: -DjBR+w4c+o0.2c/0.2c
4. 将图例放在右上角: -DjTR+w4c+o0.2c/0.2c

-F 控制图例的背景属性, 见[绘制修饰物](#) 一节

默认图例无边框, 使用该选项则会给图例绘制边框。

12.95.2 图例文件格式

图例文件用于控制图例中各项的布局。图例文件中的每个记录对应图例中的一项, 图例中每项的顺序由记录的先后顺序决定。每个记录的第一个字符决定了当前记录的图例类型。GMT 中共有 14 种图例类型, 列举如下:

comment 以 # 开头的行或空行都会被跳过

A cptname 指定 CPT 文件, 使得某些记录可以通过指定 Z 值来设定颜色, 可以多次使用该记录以指定不同的 CPT 文件

B cptname offset height [optional arguments] 绘制水平 colorbar

1. **offset** 是 colorbar 相对于图例框左边界的距离

2. **height** 是 colorbar 高度

3. 还可以加上子选项 +e[b|f][<length>][+h][+ma|c|l|u][+n[<txt>]]以及其他一些参数: -B、-I、-L、-M、-N、-S、-Z、-p 等, 详情见[psyscale](#) 命令

C textcolor 接下来的所有文本所使用的颜色。

可以直接指定颜色, 也可以用 **z=<val>** 指定 Z 值, 以从 CPT 文件中查找相应的颜色 (CPT 文件由 A 记录指定), 若 **textcolor** 为 - , 则使用默认颜色

D [offset] pen [-|+|=] 绘制一条水平线

1. offset 为线条左右顶端与图例边框的空白距离
2. pen 为线条属性
 1. 默认值为 0
 2. 若未指定 pen, 则使用 MAP_GRID_PEN_PRIMARY
 3. 若 pen 设置为 - , 则绘制一条不可见的线
 3. 默认情况下, 线条上下各留出四分之一的行间距, -|+|= 分别表示线条上方无空白、线条下方无空白和线条上下均无空白。

F fill1 fill2 ... filln 指定单元的填充色。

可以直接指定颜色, 也可使用 z=<value> 形式指定从 CPT 文件中查找颜色。若只给定了一个 fill, 则整行都使用相同的填充色, 否则依次为当前行的每列应用不同的 fill, 若 fill 为 - , 则不填充。

G gap 给定一个垂直空白

空白的高度由 gap 决定, gap 可以用 i|c|p 单位, 也可以用 l 作为单位表示多少行空白, gap 也可以取负值, 表示将当前行上移。

H fontsize|- font|- header 为图例指定一个居中的标题。

1. header 为标题
2. fontsize 为字号
3. font 为字体号
4. - 表示使用默认的文字大小以及默认字体 FONT_TITLE

I imagefile width justification 将 EPS 或光栅文件放在图例中

1. width 为图片宽度
2. justification 为图片的对齐方式

L fontsize|- font|- justification label 在图例中某列增加指定的文字

1. label 为显示的文本
2. fontsize 为字号
3. font 为字体号
4. justification 为对齐方式, 可以取 L|C|R , 分别表示左对齐、居中对齐和右对齐
5. - 表示使用默认的文字大小以及默认字体 FONT_TITLE

M slon|- slat length [+f][+l[label]][+u] [-Fparam] [-Rw/e/s/n -Jparam]

在图例中绘制比例尺, 在 [psbasemap](#) 命令中有详细介绍

N ncolumns 或 N relwidth1 relwidth2 ... relwidthn 修改图例中的列数

默认只有一列, 该记录仅对 S 和 L 记录有效。该记录指定的列数会一直有效直到再次使用 N 记录。ncolumns 用于指定若干个等宽的列, relwidth1 relwidth2 ... relwidthn 用于指定每列所占的相对宽度, 所有宽度的和应等于 -D 选项所设置的宽度相等。

P paragraph-mode-header-for-pstext 在图例中添加段落, 参考 [pstext](#) 命令中的段落模式

S [dx1 symbol size fill pen] [dx2 text] 在图例中绘制符号

1. **symbol** 指定要绘制的符号类型, 见 [psxy](#) 命令的 **-S** 选项。若 **symbol** 设置为 **-** 则表示绘制线段
2. **dx1** 是符号中心与左边界的距离
3. **dx2** 是 **text** 与左边界的距离
4. **text** 的字体由参数 **FONT_ANNOT_PRIMARY** 控制
5. **fill** 和 **pen** 控制符号的填充和轮廓颜色, 设置为 **-** 则表示不填充或无轮廓颜色
6. **fill** 可以使用 **z=<val>** 的形式, 从 CPT 文件中查找颜色
7. **dx1** 除了可以指定距离, 还可以使用 **L|C|R** 表示符号在当前列的对齐方式
8. 若 **S** 记录中无其他参数, 则直接调至下一列
9. 若 **symbol** 取为 **f|q|v**, 则可以在 **symbol** 后加上额外的子选项
10. 某些符号可能需要指定多个 **size**, 将多个 **size** 用逗号分隔作为 **size** 即可

T paragraph-text 用参数 **FONT_ANNOT_PRIMARY** 打印一段文本

V [offset] pen 在两列之间绘制垂直的线条

12.95.3 示例

```
#!/bin/bash
gmt pslegend -R-10/10/-10/10 -JM6i -F+gazure1 -Dx0.5i/0.5i+w5i/3.3i+jBL+l1.2 \
-C0.1i/0.1i -B5f1 << EOF > pslegend_ex1.ps
# Legend test for pslegend
# G is vertical gap, V is vertical line, N sets # of columns, D draws horizontal line.
# H is header, L is label, S is symbol, T is paragraph text, M is map scale.
#
G -0.1i
H 24 Times-Roman My Map Legend
D 0.2i 1p
N 2
V 0 1p
S 0.1i c 0.15i p300/12 0.25p 0.3i This circle is hachured
S 0.1i e 0.15i yellow 0.25p 0.3i This ellipse is yellow
S 0.1i w 0.15i green 0.25p 0.3i This wedge is green
S 0.1i f0.1i+l+t 0.25i blue 0.25p 0.3i This is a fault
S 0.1i - 0.15i - 0.25p,- 0.3i A dashed contour
S 0.1i v0.1i+a40+e 0.25i magenta 0.25p 0.3i This is a vector
S 0.1i i 0.15i cyan 0.25p 0.3i This triangle is boring
V 0 1p
D 0.2i 1p
N 1
M 5 5 600+u f
G 0.05i
#I SOEST_logo.ras 3i CT
G 0.05i
B rainbow.cpt 0.2i 0.2i
G 0.05i L 9 4 R Smith et al., @%5%J. Geophys. Res., 99@%%, 2000
G 0.1i
```

(下页继续)

(续上页)

P

T Let us just try some simple text that can go on a few lines.
 T There is no easy way to predetermine how many lines will be required,

T so we may have to adjust the box height to get the right size box.

EOF

rm gmt.*

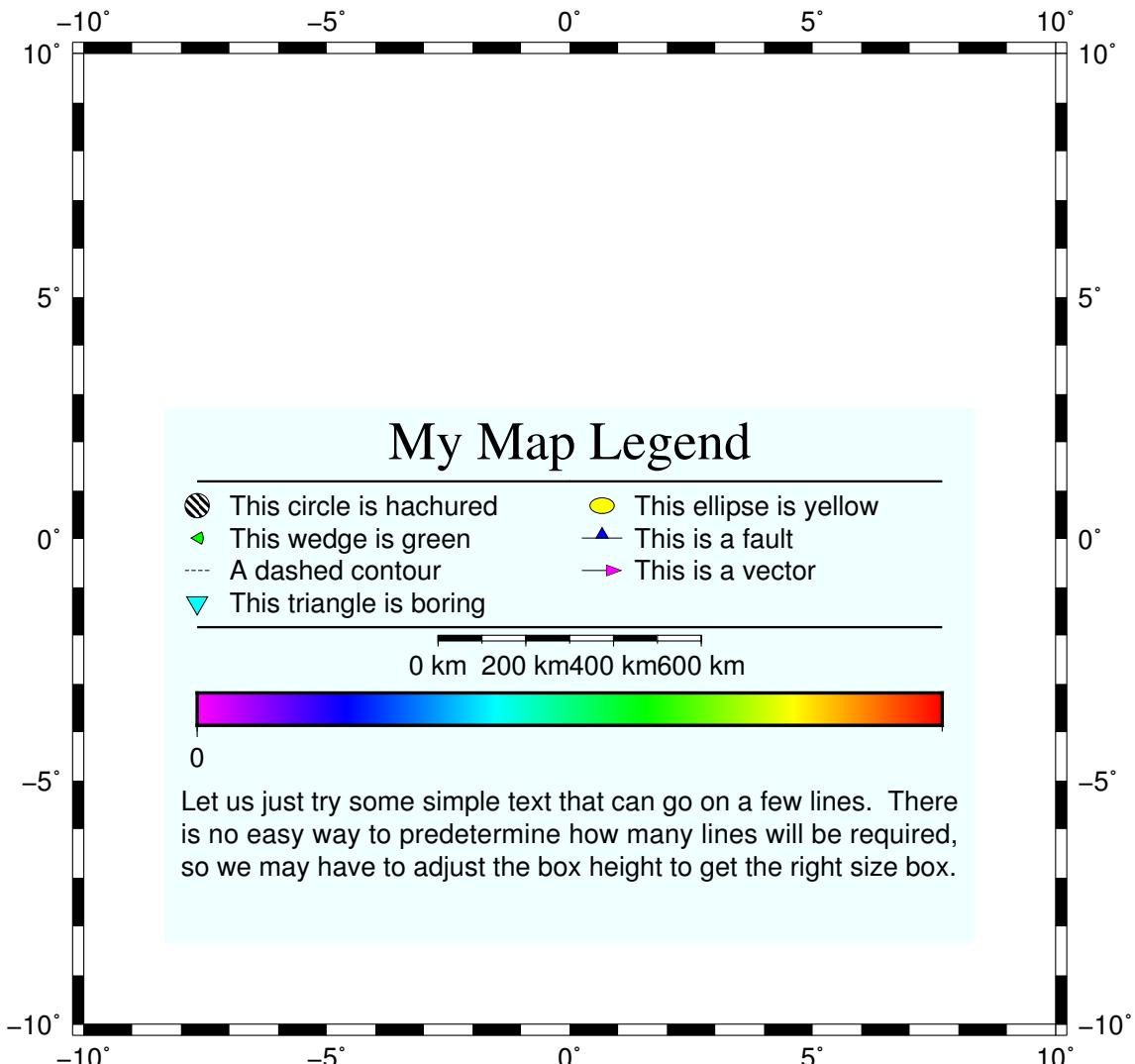


图 12.7: pslegend 示例图 1

```
#!/bin/bash
ps=pslegend_ex2.ps

gmt gmtset FONT_ANNOT_PRIMARY 12p FONT_LABEL 12p

cat <<EOF > table.txt
#G 0.04i
H 24 Times-Roman Eight Largest Cities in North America
D 1p
N 6 22 16 20 20 8 8
EOF
```

(下页继续)

(续上页)

```
V 0.25p
S 0.15i c 0.1i snow1 -
L -- C City Name
L -- C Country
L -- C Population
L -- C Climate
L -- C WC?
L -- C OL?
D 0 1p
F lightgreen
S 0.15i c 0.1i red 0.25p
L -- R Mexico City
L -- R Mexico
L -- R 8,851,080
L -- R Tropical
L -- C Y
L -- C Y
F -
S 0.15i c 0.1i orange 0.25p
L -- R New York City
L -- R USA
L -- R 8,405,837
L -- R Tempered
L -- C Y
L -- C N
S 0.15i c 0.1i yellow 0.25p
L -- R Los Angeles
L -- R USA
L -- R 3,904,657
L -- R Subtropical
L -- C Y
L -- C Y
F lightblue
S 0.15i c 0.1i green 0.25p
L -- R Toronto
L -- R Canada
L -- R 2,795,060
L -- R Tempered
L -- C N
L -- C N
F -
S 0.15i c 0.1i blue 0.25p
L -- R Chicago
L -- R USA
L -- R 2,714,856
L -- R Tempered
L -- C Y
L -- C N
S 0.15i c 0.1i cyan 0.25p
L -- R Houston
L -- R USA
L -- R 2,714,856
L -- R subtropical
L -- C N
```

(下页继续)

(续上页)

```

L -- C N
F lightred
S 0.15i c 0.1i magenta 0.25p
L -- R Havana
L -- R Cuba
L -- R 2,106,146
L -- R Tropical
L -- C N
L -- C N
F lightblue
S 0.15i c 0.1i white 0.25p
L -- R Montreal
L -- R Canada
L -- R 1,649,519
L -- R Tempered
L -- C N
L -- C Y
D 1p
V 1p
F -
N 1
L 9 4 R Information from Wikipedia
G 0.05i
T Many of these cities have hosted World Cup Soccer (WC) and some
T have hosted the Olympics (OL). The rest is just some basic information
T about each city, such as climate and population. Of course, this is all
T an excuse to demonstrate variable-width tables and row coloring.
#G 0.1i
EOF
cat << EOF > t.cpt
1      red
2      orange
3      yellow
4      green
5      blue
6      cyan
7      magenta
8      white
EOF
gmt psLegend -Dx0/0+w5.6i+jBL+l1.1 -C0.05i -F+p+gsnow1 -B0 table.txt -P --FONT_ANNOT_
↪PRIMARY=12p \
--FONT_LABEL=12p > $ps
rm -f table.txt t.cpt gmt.*

```

	City Name	Country	Population	Climate	WC?	OL?
●	Mexico City	Mexico	8,851,080	Tropical	Y	Y
○	New York City	USA	8,405,837	Tempered	Y	N
■	Los Angeles	USA	3,904,657	Subtropical	Y	Y
●	Toronto	Canada	2,795,060	Tempered	N	N
●	Chicago	USA	2,714,856	Tempered	Y	N
●	Houston	USA	2,714,856	subtropical	N	N
●	Havana	Cuba	2,106,146	Tropical	N	N
○	Montreal	Canada	1,649,519	Tempered	N	Y

Information from Wikipedia

Many of these cities have hosted World Cup Soccer (WC) and some have hosted the Olympics (OL). The rest is just some basic information about each city, such as climate and population. Of course, this is all an excuse to demonstrate variable-width tables and row coloring.

图 12.8: pslegend 示例图 2

12.96 psmask

官方文档 [psmask](#)

12.97 psmeca

官方文档 [psmeca](#)

简介 在地图上绘制震源机制解

12.97.1 -S 选项

-S 确定了输入文件的格式, 其语法为:

```
-S<format><scale>[/<fontsize>[/<offset>][u]]
```

- <format> 可以取 a、c 等, 在下面会介绍
- <scale> 指定了 5 级地震(地震矩为 4.0E23 dynes-cm) 的震源球的直径。默认情况下, 震源球的直径与震级大小成正比, 实际直径由如下公式决定:

```
size = M / 5 * <scale>
```

- <fontsize> 是震源球的标题的字体大小
- <offset> 震源球标题相对于震源球的偏移量
- u 震源球标题位于震源球的下方

下面介绍各种震源机制解的格式, 使用下面这个地震作为示例:

```
Date: 1976/ 1/ 1 Centroid Time: 1:29:53.4 GMT
Lat= -29.25 Lon=-176.96
Depth= 47.8 Half duration= 9.4
Centroid time minus hypocenter time: 13.8
Moment Tensor: Expo=26 7.680 0.090 -7.770 1.390 4.520 -3.260
Mw = 7.3 mb = 6.2 Ms = 0.0 Scalar Moment = 9.56e+26
Fault plane: strike=202 dip=30 slip=93
Fault plane: strike=18 dip=60 slip=88
```

-Sa Aki and Richards 约定的震源机制格式, 输入文件的具体格式为:

```
X Y depth strike dip rake mag newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度, 单位为 km
- strike、dip、rake 断层的三个参数, 单位为度
- mag 为震级
- newX 和 newY 放置震源球的经纬度, 这样可以避免震源球覆盖震源位置。若值为 0, 则表示将震源球放在震源位置
- title 出现中震源球上方/下方的字符串(可选)

```
gmt psmeca -Rg -JN15c -Sa2c > meca.ps << EOF
-176.96 -29.25 47.8 202 30 93 7.3 0 0 197601010129
EOF
```

-Sc Harvard CMT(现在的 Global CMT) 定义的格式:

```
X Y depth strike1 dip1 rake1 strike2 dip2 rake2 mantissa exponent newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度, 单位为 km
- 两组 strike、dip、rake 分别是两个断层面的参数
- mantissa 和 exponent 是地震标量矩的尾数和指数部分。例如, 地震标量矩为 9.56e+26 dyne-cm, 则 mantissa=9.56、exponent=26

```
gmt psmeca -Rg -JN15c -Sc2c > meca.ps << EOF
-176.96 -29.25 47.8 202 30 93 18 60 88 9.56 26 0 0 197601010129
EOF
```

-Sm|d|z GCMT 的零迹矩张量解, 输入数据格式为:

```
X Y depth mrr mtt mff mrt mrf mtf exp newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度, 单位为 km
- mrr 等是地震矩的 6 个分量, 单位是 10^{exp} dyne-cm
- exp 地震矩的指数部分(比如, mrr=2.0、exp=26, 则真实的 mrr=2.0e26)

地震矩张量可以分解成各向同性部分 (ISO)、双力偶部分 (DC) 和补偿线性向量偶极

部分 (CLVD)。

- **m** 表示绘制地震矩的零迹部分 (DC+CLVD)
- **d** 表示仅绘制地震矩的双力偶部分 (DC)
- **z** 表示仅绘制地震矩的零迹各向异性部分 (DC+CLVD)

说明：

- **m** 和 **z** 是一个东西？还是我的理解有误？TODO
- GMCT 所使用的 rtf 坐标，实际上就是 USE 坐标系

```
gmt psmeca -Rg -JN15c -Sm2c > meca.ps << EOF
-176.96 -29.25 47.8 7.680 0.090 -7.770 1.390 4.520 -3.260 26 0 0 title
EOF
```

-Sp 由两个断层平面的部分数据构成的机制解，输入数据格式为：

```
X Y depth strike1 dip1 strike2 fault mag newX newY title
```

- **X** 和 **Y** 为震源经纬度
- **depth** 是地震深度，单位为 km
- **strike1** 和 **dip1** 平面 1 的断层参数
- **strike2** 平面 2 的断层参数
- **fault** 取-1 或 +1，表示正断层和逆断层
- **mag** 震级

-Sx|y|t 指定主坐标轴的方位，即 T、N、P 轴，输入数据格式为：

```
X Y depth Tvalue Tazim Tplunge Nvalue Nazim Nplunge Pvalue Pazim Pplunge exp newX
→newY title
```

- **X** 和 **Y** 为震源经纬度
- **depth** 是地震深度，单位为 km
- **Tvalue** 等 9 个量定义了 T、N、P 轴的大小和方向
- **exp** 是 **Tvalue** 等的指数部分

对这种表示不太熟悉，应该是跟地震矩等效的：

- **x** 绘制标准的 GCMT 解
- **y** 只绘制地震矩的 double couple 部分 (DC)
- **z** 绘制 0 迹地震矩

12.97.2 选项

<table> 输入文件，文件中震源机制信息，具体格式由 **-S** 选项决定。

-C[<pen>][P<pointsizes>] 绘制 (X,Y) 和 (newX,newY) 的连线

将震源球放在输入文件的 **newX** 和 **newY** 所指定的位置，在震源位置处绘制一个小圆，并绘制一条直线连接小圆和震源球。

<pen> 指定直线的属性, <pointsize> 指定圆的大小。

-D<depmin>/<depmax> 只绘制震源深度在 <depmin> 和 <depmax> 之间的地震。

-E<fill> 扩张部分的填充色, 默认为白色

-F<mode>[<args>] 设置多个属性, 可重复使用多次。

- **-Fa[<size>] [/<P_axis_symbol> [<T_axis_symbol>]]** 在震源球上 P 轴和 T 轴处绘制符号。<size> 是符号大小; 符号可以取 c|d|h|i|p|s|t|x (具体含义见 *psxy* 的 -S 选项), 默认值为 6p/cc, 即在 P 轴和 T 轴处分别绘制一个大小为 6p 的圆

- **-Fe<fill>** 设置 T 轴符号的填充色

- **-Fg<fill>** 设置 P 轴符号的填充色

- **-Fp<pen>** 设置 P 轴符号的画笔属性

- **-Ft<pen>** 设置 T 轴符号的画笔属性

- **-Fo** 使用旧版本的 *psvelomeca* 命令的输入数据格式, 即不需要第三列的深度信息

- **-Fr<fill>** 在标签后加一个方框

- **-Fz<pen>** 覆盖零迹矩张量的画笔属性

-G<fill> 指定压缩部分的填充色, 默认值为黑色

-L<pen> 设置震源球外部轮廓的线条属性。

-M 所有震级使用相同的大小, 具体大小由 -S 选项的 <scale> 参数决定。

-N 地图区域外的震源球也要绘制, 默认不绘制。

-T<num_of_planes>[/<pen>] 绘制断层平面。<num_of_planes> 可以取:

- 0 绘制两个断层面
- 1 绘制第一个断层面
- 2 绘制第二个断层面

-W<pen> 同时设置所有线条以及符号轮廓的画笔属性以及标题颜色。

该选项设置的属性可以被 -C、-L、-T、-Fz 指定的属性替代。

-Z<cpt> 指定 CPT 文件, 根据数据文件中第三列的值 (即地震深度) 确定震源球的压缩部分的颜色。

12.97.3 示例

```
#!/bin/bash
PS=psmeca_ex1.ps

gmt pscoast -Rg -JQ10i -Bx60 -By30 -A10000 -Wthin -G200 -K > $PS
gmt psmeca -R -J -Sc1/12/0.2 -C0.5pP2p -Egrey -Gred -L1p -W1p -O << EOF >> $PS
-176.96 -29.25 47.8 202 30 93 18 60 88 9.56 26 -160 -10 010176A
EOF
rm gmt.*
```

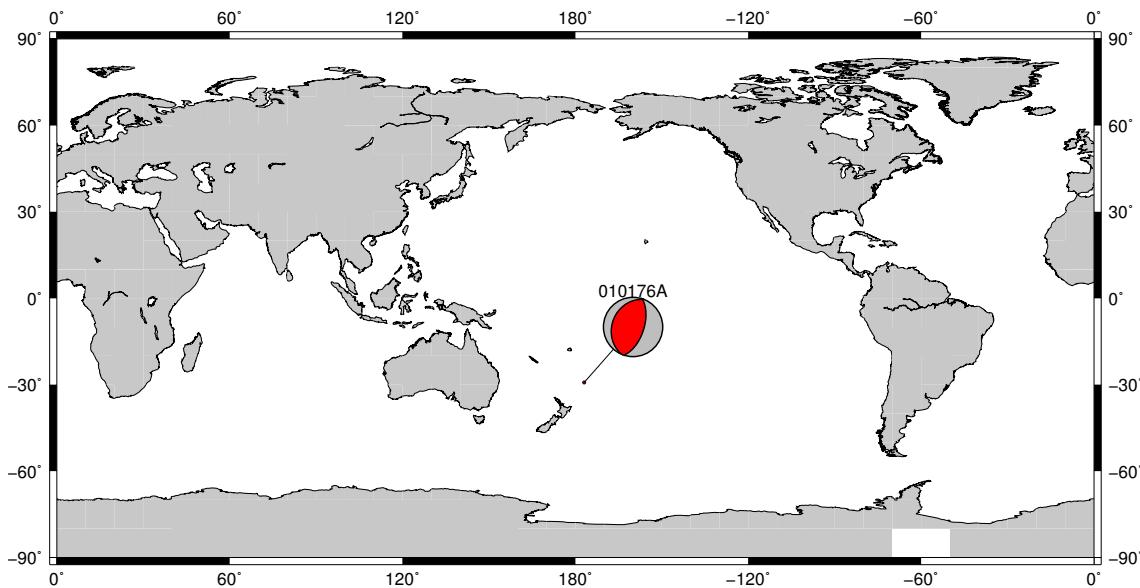


图 12.9: psmeca 示例图 1

```
#!/bin/bash
# test/meca/meca_1.sh
# Check gmt psmeca for plotting beach balls

ps=psmeca_ex2.ps

# Right lateral Strike Slip
echo 0.0 5.0 0.0 0 90 0 5 0 0 Right Strike Slip | gmt psmeca -Sa2.5c -Gblack \
-R-1/4/0/6 -JM14c -P -B2 -K > $ps

# Left lateral Strike Slip
echo 2.0 5.0 0.0 0 90 180 5 0 0 Left Strike Slip | gmt psmeca -Sa2.5c -Gblack -R -J -K - \
-K >> $ps

# Thrust
echo 0.0 3.0 0.0 0 45 90 5 0 0 Thrust | gmt psmeca -Sa2.5c -Gblack -R -J -K -0 >> $ps
echo 2.0 3.0 0.0 45 45 90 5 0 0 Thrust | gmt psmeca -Sa2.5c -Gblack -R -J -K -0 >> $ps

# Normal
echo 0.0 1.0 0.0 0 45 -90 5 0 0 Normal | gmt psmeca -Sa2.5c -Gblack -R -J -K -0 >> $ps
echo 2.0 1.0 0.0 45 45 -90 5 0 0 Normal | gmt psmeca -Sa2.5c -Gblack -R -J -K -0 >> $ps

# Mixed
echo 3.4 0.6 0.0 10 35 129 5 0 0 Mixed | gmt psmeca -Sa2.5c -Gblack -R -J -0 >> $ps
rm gmt.*
```

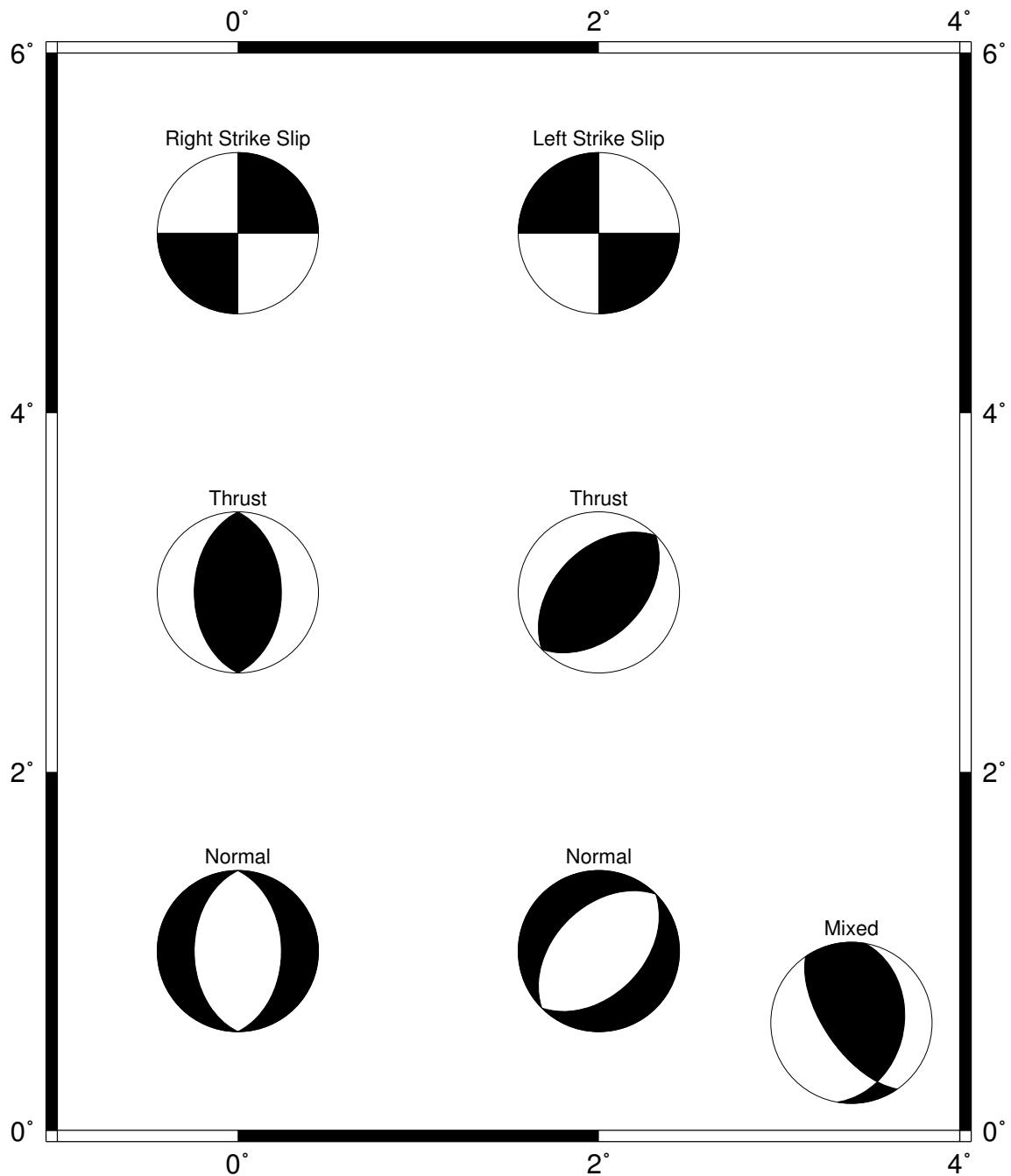


图 12.10: psmeca 示例图 2

```

#!/bin/bash
# test/meca/meca_2.sh

ps=psmeca_ex3.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i MAP_FRAME_WIDTH 0.1i \
MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i

# This is a strike-slip CMT mechanism in red
# Best double couple overlays moment tensor
gmt psmeca -X2i -Y5i -R238.5/242/32.5/35.5 -Jm1.3i -B2 -BWeSn+tpsmeca -Sm0.4i/12pu -T0 \

```

(下页继续)

(续上页)

```
-P -C0.5pP5p -Gred -K -W0.75p << EOF > $ps
# lon      lat      dep mrr   mtt   mff   mrt   mrf   mtf   exp   plon   plat   text
239.384  34.556   33  -.27  -2.13  2.40  -.07  -1.32  -.79  24   240.0  35   tensor
EOF

# Focal mechanisms are labeled with following font
gmt gmtset FONT_ANNOT_PRIMARY Times-Roman
# Here are several thrusting mechanisms in Aki and Richards convention
# New second argument -Sa is font size in points
# Old gmt psvelomeca format is used

gmt psmeca -Fo -R -J -Sa0.4i/16p -C0.25p,blueP5p -O -K -N << EOF >> $ps
# lon      lat      strike dip    rake    Mw      plon   plat   text
241.459  34.2088   112.3  42.2   89.8   6.6     238.  35.0  first
241.459  34.2088   120.0  60.0   86.0   5.2     238.  34.0  second
241.459  34.2088   290.0  55.0   90.0   5.9     238.  33.0  third
EOF

# Plot P and T axis and only the best double couple
# derived from moment tensor
gmt psmeca -R -J -Sd1c -O -K -N -Fa0.2c/id << EOF >> $ps
# lon      lat      dep   mrr   mtt   mff   mrt   mrf   mtf   exp   plon   plat
241     33      3    -.27  -2.13  2.40  -.07  -1.32  -.79  26      0      0
EOF

# hit the beach
gmt pscoast -O -R -J -W1p -Di >> $ps
rm gmt.*
```

psmeca

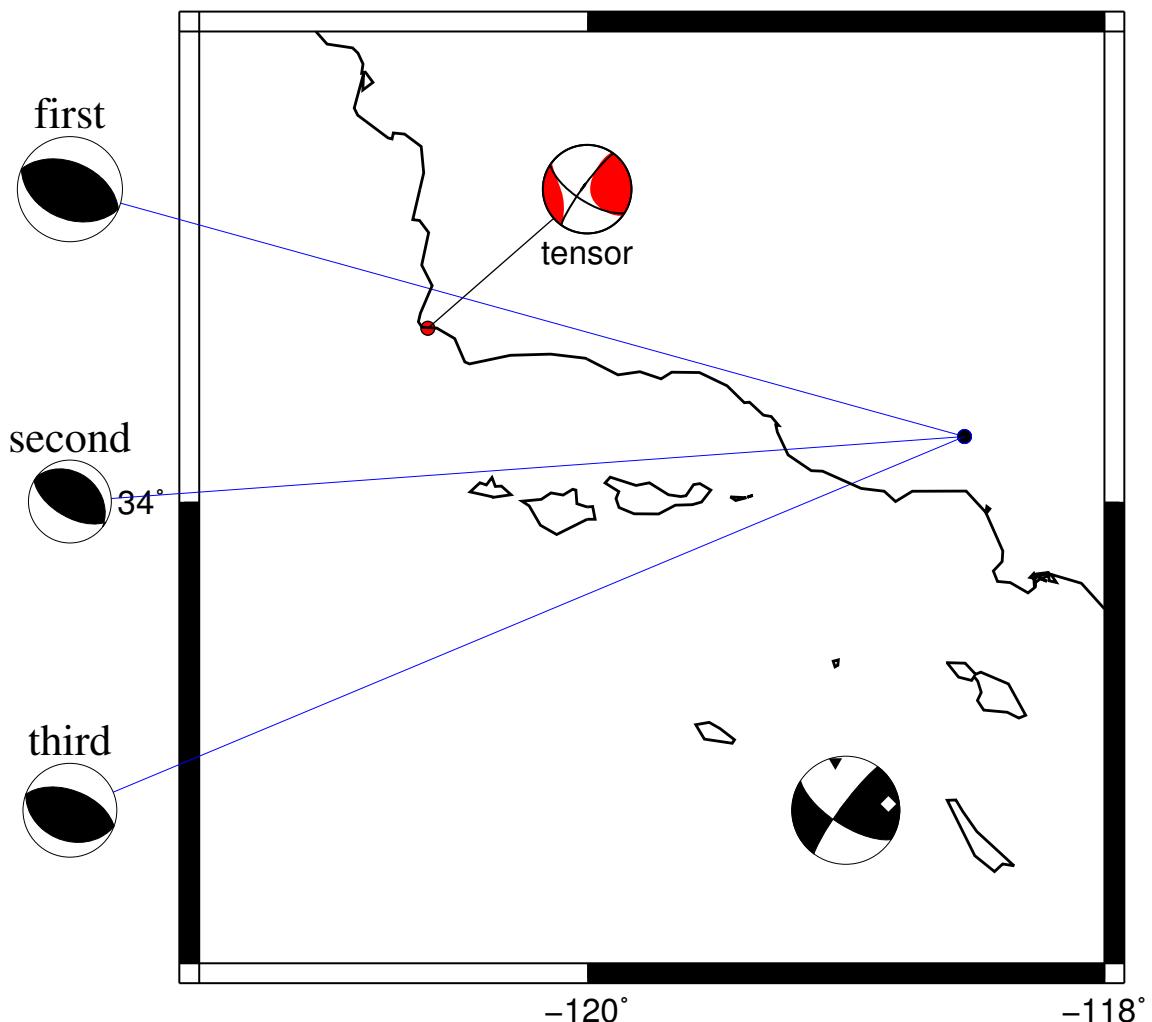


图 12.11: psmeca 示例图 3

12.98 pspolar

[官方文档 pspolar](#)

简介 将台站的极性信息画在震源球上

一般情况下, 需要使用 psmeca 绘制震源球, 再使用 pspolar 将每个台站的极性信息画在相应的震源球上。因而, 通常 psmeca 和 pspolar 需要连在一起使用。

12.98.1 选项

<infile> 输入数据的格式为:

```
station_code  azimuth  take-off_angle polarity
```

- `station_code` 通常是台站名

- **azimuth** 震相从源到台站的方位角
- **take-off_angle** 震相从源出发时的出射角
- **polarity**

- 压缩部分 (正极性): 可以取 c|C|u|U|+
- 拉伸部分 (负极性): 可以取 d|D|r|R|-
- 未定义: 其他字符

-C<new_lon>/<new_lat>[W<pen>][P<pointsize>] 将震源球放在新的位置，并将新位置与老位置之间连线。

-D<lon>/<lat> 震源球的位置, 需要与 psmecha 输入数据中震源球的位置相同

-E<color> 拉伸象限内台站的符号填充色

-F<color> 设置震源球的背景色, 默认不填充。

-G<color> 压缩象限内台站的符号填充色, 默认值为黑色

-M<size> 震源球的尺寸, 需要与 psmecha 中 -S 选项中震源球尺寸相同

-S<symbol_type><size> 类似于 psxy 的 -S 选项, 控制极性要以什么符号以及多大尺寸绘制到震源球上, 可选的符号类型包括 a|c|d|h|i|p|s|t|x

-N 不跳过地图边界外的符号

-Q<mode><args> 设置多个属性, 该选项可重复使用。

- **-Qe[<pen>]** 拉伸象限内符号的轮廓属性
- **-Qf[<pen>]** 震源球的轮廓属性
- **-Qg[<pen>]** 压缩象限内符号的轮廓属性
- **-Qh** 使用 HYPO71 输出的特殊格式
- **-Qs<half-size>/[V[<veccpar>]][G<fill>][L]** 绘制 S 波偏振方位角, 见官方文档
- **-Qt[<pen>]** station_code 的字体颜色

-T<angle>/<form>/<justify>/<fontsize> 将 station_code 写到图上, 默认值为 0.0/0/5/12 。<fontsize> 的单位为 p

-W<pen> 设置画笔属性

12.98.2 示例

```
#!/bin/bash
ps=pspolar_ex1.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i MAP_FRAME_WIDTH 0.1i \
MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i FONT_TITLE 18p

# this is Harvard CMT for tibethan earthquake (1997)
gmt psmecha -Fo -R85/89/25/50 -JX7i -P -M -Sm4i -N -L -K -G150 -T0 << EOF > $ps
# lon lat mrr mtt mff mrt mrf mtf ex nlon nlat
87 35 -0.26 -0.71 0.97 -0.20 -0.61 2.60 27 0 0
EOF

# and polarities observed
gmt pspolar -R -J -D87/35 -M4i -N -Sc0.3i -Qe -O \
-B0 -B+t"Tibet earthquake (1997) - polarities distribution" << EOF >> $ps
1 147.8 53 c
```

(下页继续)

(续上页)

```
2 318.6 53 c
3 311.9 53 c
4 122.5 45 c
5 87.1 44 c
6 259.9 44 c
7 358.0 43 d
8 32.3 40 d
9 144.5 40 c
10 206.2 40 d
11 30.0 36 d
12 88.3 31 c
13 326.5 31 c
14 298.4 29 c
15 298.3 29 c
16 316.2 28 c
17 301.5 27 c
18 300.7 27 c
19 303.0 27 d
20 302.7 26 c
21 301.7 26 c
22 302.3 26 c
23 302.2 26 c
24 314.1 26 c
25 296.2 26 c
26 302.3 26 c
27 146.8 26 c
28 145.7 26 d
29 145.7 26 c
30 307.0 26 c
31 311.9 26 c
32 136.4 25 c
33 297.6 25 c
34 306.1 25 c
35 306.8 25 c
36 307.6 25 c
37 346.5 25 c
39 306.5 24 c
40 317.3 24 c
41 305.2 24 c
42 305.9 24 c
43 311.9 24 c
44 307.5 24 c
45 138.7 24 d
46 322.4 24 c
47 305.3 24 c
48 304.9 24 c
49 309.3 24 c
50 307.6 24 c
51 315.5 24 d
52 310.3 24 c
53 308.5 24 c
54 307.4 24 c
55 307.5 24 c
56 307.4 24 c
```

(下页继续)

(续上页)

```
57 307.6 24 c  
58 307.1 24 c  
59 311.5 23 d  
61 243.5 23 d  
63 345.2 23 c  
64 117.0 21 d  
65 133.1 20 c  
66 116.0 20 c  
67 231.3 17 d  
68 139.9 16 c  
69 131.7 15 d  
70 114.1 15 c  
EOF  
rm gmt.*
```

Tibet earthquake (1997) – polarities distribution

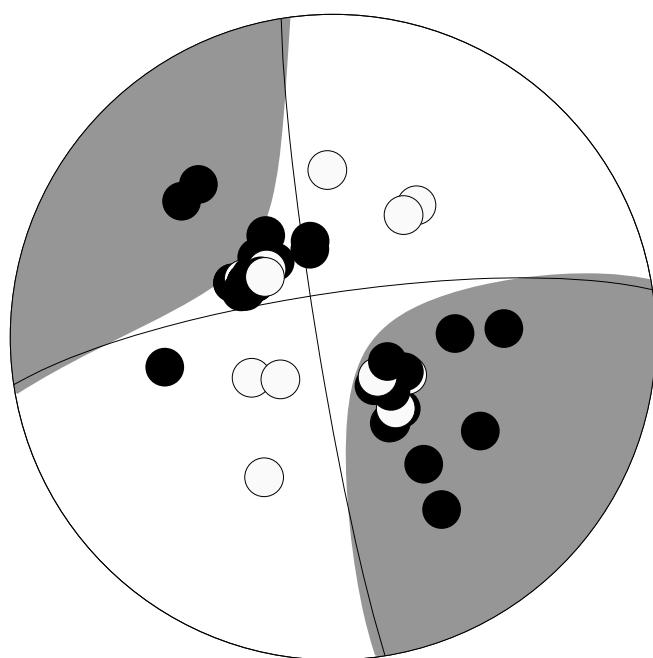


图 12.12: pspolar 示例

12.99 psrose

官方文档 [psrose](#)

简介 绘制极坐标下的直方图 (sector 图、rose 图或 windrose 图)

12.99.1 可选选项

<table> 输入文件, 数据格式为 `length azimuth`

若输入文件中只有 azimuth 一列数据, 则此时需要使用 `-i0` 选项, 此时所有的长度都默认为单位长度。

-A[r]<sector_width> 指定扇页宽度, 单位为度

1. 默认扇页宽度为 0, 即 windrose 图
2. 若扇页宽度不为 0, 则表示绘制 sector 图
3. 若扇页宽度不为 0 且使用了 `-Ar`, 则表示绘制 rose 图

-B 此模块中, X 表示径向距离, Y 表示方位角。Y 轴的标签是图片的标题, 比例尺长度由径向网格间隔决定。

-Cm[+w]<mode_file> 绘制矢量以显示 `<mode_file>` 中指定的主方向。

使用 `-Cm` 则计算并绘制平均方向。使用 `-Cm+w<mode_file>` 则将计算得到的平均方向及其他统计结果以如下格式保存到文件中:

```
mean_az, mean_r, mean_resultant, max_r, scaled_mean_r, length_sum, n, sign@alpha
```

其中最后一项可以取 0 或 1, 取决于平均结果是否 significant at the level of confidence set via `-Q`.

-D 对扇页对偏移, 使得其位于每个间隔的中间, 即第一个扇页的中心在 0 度处

-F 不绘制 scale length bar, 默认会在右下角绘制

-G<fill> 绘制扇页的填充色

-I 不绘制图形, 只计算 `-R` 选项所需要的参数。

以下统计信息会输出到标准输出:

```
n, mean az, mean r, mean resultant length, max bin sum, scaled mean, linear length
→sum
```

-L[<wlabel>,<elabel>,<slabel>,<nlabel>] 指定 0、90、180、270 度处的标签。

1. 对于 full-circle 图而言, 默认值为 WEST, EAST, SOUTH, NORTH
2. 对于 half-circle 图而言, 默认值为 90W, 90E, -, 0, 其中 - 表示不显示标签
3. 只使用 `-L` 但无其他参数表示不显示所有标签

-M<parameters> 与 `-C` 选项一起使用以修改矢量的属性。具体属性见[绘制矢量/箭头](#)一节

-Q[<alpha>] 设置置信水平, 用于决定平均结果是否显著, 默认值为 0.05。

注意: 临界值是近似结果且需要至少 10 个数据点; the critical resultants are accurate

to at least 3 significant digits. 对于更小的数据集，需要查询准确的统计表。

-R<r0>/<r1>/<az_0>/<az_1> 指定绘图的半径和方位角范围。

- $r0$ 取 0
- $r1$ 取最大长度
- 方位角范围取 $-90/90$ 或 $0/180$ 以绘制 half circle plot
- 方位角范围取 $0/360$ 以绘制 full circle

-S[n]<plot_radius>[u] 指定圆的半径。

-Sn 会将输入的半径归一化到 0 到 1。

-T 指定输入数据为 orientation 数据（即数据范围在 0-180 度范围内）而不是 0-360 度范围的 direction 数据。We compensate by counting each record twice: First as azimuth and second as azimuth + 180. Ignored if range is given as -90/90 or 0/180.

-W[v]<pen> 设置扇区边框的画笔属性。

-Wv<pen> 可用于设置绘制矢量时所需的画笔属性。

-Zu|<scale> 将数据的半径乘以 <scale>，比如 **-Z0.001** 会将数据的单位从 m 变成 km。

若不考虑半径，可以通过 **-Zu** 将所有的半径设置为单位长度。

-: 输入数据为 azimuth, radus 而不是 radius, azimuth

12.100 pssac

官方文档 [pssac](#)

简介 在地图上绘制 SAC 格式的地震波形数据

该命令会直接读取命令行中的 SAC 文件名，或者从输入文件中读入 SAC 文件名及控制参数信息，并将波形数据画在图上。

注解： pssac 模块修改自原 pssac 与 pssac2，其功能类似，但语法不同。

pssac 实现波形绘制的步骤是：

1. 读入 SACfiles 或 saclist
2. 根据 **-T** 选项确定参考时间
3. 根据 **-C** 选项和参考时间读入指定的波形数据段
4. 根据 **-F** 选项对数据进行预处理
5. 根据 **-M** 选项确定 Y 方向的缩放因子并对波形振幅进行缩放
6. 根据 **-Q** 决定是否交换 X 和 Y
7. 根据 **-E** 以及其他信息确定波形在地图上的位置
8. 根据 **-W** 选项绘制波形
9. 根据 **-G** 选项为波形涂色

12.100.1 必选选项

-J -R

SACfiles 要绘制在图上的一系列 SAC 文件名, 目前仅支持等间隔 SAC 数据
saclist SAC 文件列表, 每行包含一个 SAC 文件名及其对应的控制参数。

文件格式为:

```
filename [X Y [pen]]
```

- **filename** 是要绘制的 SAC 文件名
- X 和 Y 控制 SAC 波形的第一个数据点在地图上的位置。若省略 X 和 Y，则使用其默认值，否则此处指定的 X 和 Y 将具有最高优先级
 - 对于线性投影而言，X 默认是 SAC 文件的开始时间。使用 -T 选项会调整 X 的默认值，Y 值则由 -E 选项控制；
 - 对于地理投影而言，X 和 Y 默认是台站的经度和纬度。
- pen 控制当前 SAC 波形的线条属性

12.100.2 可选选项

-C[<t0>/<t1>] 只读取并绘制时间窗 <t0> 到 <t1> 范围内的波形。

<t0> 和 <t1> 均是相对于参考时间的秒数, 参考时间由 -T 选项决定。若未使用 -T 选项，则使用 SAC 头段中的参考时间 (kzdate 和 kztime)。

若只使用了 -C 但未指定时间窗，则 <t0>/<t1> 由 -R 选项的 <xmin>/<xmax> 决定。

-D<dx>[/<dy>] 使得波形偏离指定位置 <dx>/<dy>。若未指定 <dy> 则默认与 <dx> 相同。

-Ea|b|k|d|n[<n>]|u[<n>] 选择剖面类型 (即 Y 轴类型)

- a 方位角剖面
- b 反方位角剖面
- k 震中距剖面 (单位 km)
- d 震中距剖面 (单位 degree)
- n 波形编号剖面, 第一个波形的编号为 <n> (<n> 默认值为 0)
- u 用户自定义剖面, SAC 波形的 Y 位置由 SAC 头段变量 user<n> 决定, 默认使用 user0

-F[i][q][r] 绘图之前的数据预处理

- i 积分
- q 平方
- r 去均值

i|q|r 可以重复多次, 比如 -Fri i 会将加速度转化为位移。i|q|r 出现的顺序决定

了数据预处理的流程。

-G[p|n][+g<fill>][+z<zero>][+t<t0>/<t1>] 为波形的正/负部分涂色。

- 若 **-G** 后不接任何参数, 则默认将正值部分填充为黑色
- **p|n** 控制是要填充正值区域还是负值区域, 重复使用 **-G** 选项以分别为正/负部分设置不同的填充属性
- **+g<fill>** 填充色
- **+t<t0>/<t1>** 只填充 **<t0>/<t1>** 范围内的波形。参考时间由 **-T** 选项决定。
- **+z<zero>** 定义“零”参考线。从“零”到顶部为正值区域, 从“零”到底部为负值区域

-M<size>[<unit>]/<alpha> 控制 Y 方向的缩放。不使用的话, Y 方向即为振幅

- **<size>[<unit>]** 将所有波形在地图上的高度缩放到 **<size>[<unit>]**, 其中 **<unit>** 可以取 **i|c|p**, 默认单位由 **PROJ_LENGTH_UNIT** 控制。
- **<size>/<alpha>**
 - 若 **<alpha>** 小于 0, 则所有波形使用相同的比例因子。比例因子由第一个波形决定, 第一个波形将被缩放到 **<size>[<unit>]**
 - 若 **<alpha>** 等于 0, 则将所有波形乘以 **<size>**, 此时不允许有单位
 - 若 **<alpha>** 大于 0, 则将所有波形乘以 **<size>*r^<alpha>**, 其中 **r** 是以 km 为单位的距离

-Q 垂直绘制波形, 即 Y 轴是时间, X 轴是振幅

-S[i]<scale>[<unit>] 指定时间比例尺。

对于地理投影而言, 即表示图上一个单位距离所代表的波形描述。其中单位 **<unit>** 可以取 **c|i|p**。若未指定 **<unit>**, 则默认使用 **PROJ_LENGTH_UNIT** 所指定的单位。

在 **<scale>** 前加上 **i** 则时间比例尺的含义反过来, 即 **<scale>** 表示 1 秒长度的波形在图上的实际长度。

-T[+t<n>][+r<reduce_vel>][+s<shift>] 指定参考时间及偏移量

- **+t<tmark>** 指定参考时间 (即将所有波形沿着参考时间对齐), 其中 **<tmark>** 可以取 **-5(b), -4(e), -3(o), -2(a), 0-9(t0-t9)**
- **+r<reduce_vel>** 设置 reduce 速度, 单位 km/s
- **+s<shift>** 将所有波形偏移 **<shift>** 秒

-W<pen> 设置波形的画笔属性

12.100.3 示例

利用 SAC 的命令 **funcgen seismogram** 生成了波形, 想要绘制单个波形, 并分别为正负部分涂色:

```
gmt pssac seis.SAC -JX10c/5c -R9/20/-2/2 -Baf -Fr -Gp+gblack -Gn+gred > single.ps
```

利用 SAC 命令 **datagen sub tel *.z** 生成多个波形, 将其绘制在距离剖面上:

```
gmt pssac *.z -R200/1600/12/45 -JX15c/5c -Bx200+l'T(s)' -By5+lDegree -BWSen \
-Ed -M1.5c -W0.5p,red > distance_profile.ps
```

利用 SAC 命令 `datagen sub tel *.z` 生成多个波形，将其绘制在地图上：

```
gmt pssac *.z -JM15c -R-120/-40/35/65 -Baf -M1i -S300c -K > map.ps
saclst stlo stla f *.z | gmt psxy -J -R -St0.4c -Gblack -i1,2 -o >> map.ps
```

12.101 psscale

[官方文档 psscale](#)

简介 在地图上绘制灰色或彩色色条 (colorbar)

12.101.1 必选选项

-D[g|j|J|n|x]<refpoint>[+w<length>[/<width>]][+e[b|f][<length>]][+h|v][+]

指定色标的尺寸和位置。

1. [g|j|J|n|x]<refpoint> 指定底图上的参考点, 见[绘制修饰物](#)一节
2. +j<justify> 指定色标上的锚点, 默认锚点是 BL, 见[绘制修饰物](#)一节
3. +o<dx>[/<dy>] 指定参考点的额外偏移量, 见[绘制修饰物](#)
4. +w<length>[/<width>] 指定色标的长度和宽度。若未指定宽度, 则默认为长度的 4%; 若长度为负值则会反转色标
5. +h 绘制水平色标
6. +v 绘制垂直色标, 默认值
7. +e 选项为前景色和背景色加一个三角形, +ef 表示只加前景色三角形, +eb 表示只加背景色三角形, <length> 是三角的高度, 默认为色标宽度的一半
8. +m[a|c|l|u] 将标注、标签和单位放在色标的另一边, a 代表标注, l 代表标签, u 代表单位。c 表示将标签以单列字符垂直打印。
9. +n<text> 在色标开始处绘制一个矩形, 并用 NaN 的颜色填充

几种常用的放置色标的方式:

- 放在左边: -DjML+w2c/0.5c+o-1c/0c+m
- 放在右边: -DjMR+w2c/0.5c+o-1c/0c 可能存在问题
- 放在上方: -DjTC+w2c/0.5c+o0c/-1c+m
- 放在下方: -DjBC+w2c/0.5c+o0c/-1c+m 可能存在问题
- 放在左上角: -DjTL+w2c/0.5c+o-1c/0c+m
- 放在左下角: -DjBL+w2c/0.5c+o-1c/0c+m
- 放在右上角: -DjTR+w2c/0.5c+o-1c/0c
- 放在右下角: -DjBR+w2c/0.5c+o-1c/0c

12.101.2 选项

-B[p|s]<parameters> 设置 colorbar 的标注、刻度和网格线间隔。

在不使用 -B 选项, 或者不指定标注间隔时, 默认会根据 CPT 文件中每一行的内容

对 colorbar 进行标注, 具体见 [CPT 文件](#)。

默认情况下, 对于水平 colorbar 而言, X 轴的标签会放在 colorbar 的下边, Y 轴标签放在 colorbar 的右边; 对于垂直 colorbar 而言, X 轴的标签放在 colorbar 的右边, Y 轴标签放在 colorbar 的上边。除非在 -D 选项中使用了 +m 子选项。

-C 要绘制的 CPT 文件。

-F 为 colorbar 加上背景边框, 见 [绘制修饰物](#) 一节

-G<zlow>/<zhigh> 绘图前先对 CPT 文件做截断, 使得其只绘制 <zlow> 到 <zhigh> 之间的部分。若其中某个值等于 NaN, 则不对 CPT 的那一端做处理。

-I[<max_intens>|<low_i>/<high_i>] 为色标加上光照效果

1. **-I<max_intens>** 光照强度为 [-<max_intens>, +<max_intens>], 默认值为 1

2. **-I<low_i>/<high_i>** 指定非对称的光照强度范围

-L[i][<gap>] 生成等大小的颜色矩形。

默认情况下, 会根据 CPT 文件中 Z 值的范围决定颜色矩形的大小。若使用该选项, 则会忽略 -B 选项设置的间隔。若指定了 <gap> 且 CPT 文件是离散的, 则使用每个矩形的 Z 值下边界作为标注且将标注放在矩形的正中间。

If **i** is prepended we annotate the interval range instead. If **-I** is used then each rectangle will have its constant color modified by the specified intensity.

-M 使用 YIQ 变换将色标变成单调灰度色标

-N[p|<dpi>] 控制 color scale 如何用 PostScript 语言表示。

- **-Np** 用颜色矩形来表示 (比如离散的颜色)
- **-N<dpi>** 用图像来表示, <dpi> 默认值为 600

-Q 使用对数坐标, 刻度表示为 10 的次幂

CPT 文件中所有的 Z 值都会被转换成 $p = \log_{10}(z)$, 其中整数的 p 会以 10^p 的格式标注。

-S 去除不同色块之间的黑色网格线

-W<scale> 将 CPT 文件中所有的 Z 值乘以比例因子 <scale>

-Z<zfile> <zfile> 文件用于指定每个颜色块的宽度。

默认情况下, 颜色块的宽度由颜色的 Z 值范围决定, 比如 Z=0-100 对应的色块宽度是 Z=100-150 的色块宽度的两倍。

12.101.3 示例

```
gmt makecpt -T-200/1000/100 -Crainbow > t.cpt
gmt psscale -Ct.cpt -Dx8c/1c+w12c/0.5c+jTC+h -Bxaf+l"topography" -By+lk -l > map.ps
```

12.102 pssegy

[官方文档 pssegy](#)

12.103 pssegyz

[官方文档 pssegyz](#)

12.104 pssolar

[官方文档 pssolar](#)

简介 计算或/和绘制晨昏线以及民用、航海用以及天文用曙暮光区域

12.104.1 必选选项

-I 和 -T 中必须要有一个。

12.104.2 可选选项

-C 在一行内格式化打印 (以 Tab 键分隔) -I 选项输出的信息。输出内容包括:

1. 太阳的经度、纬度、方位角、高度角, 单位为度
2. 日出、日落、正午的时间, 单位为天
3. 日长, 单位为分钟
4. 考虑折射效应矫正后的太阳高度校正以及均时差, 单位为分钟

注解: 若没有通过 -I/lat 提供经纬度, 则太阳高度角之后的数据均以 (0,0) 作为参考点

示例:

```
$ gmt pssolar -I120/40+d2016-11-01T01:00:00+z8 -C
160.885755836 -14.5068940782 38.6719503593 -59.513608404 0.270214374769 0.
→706928713211 0.48857154399 628.868647356 -59.5102114599 16.4569766548
```

-G<fill>|c 根据晨昏线对黑夜区域填充颜色或图案, 见[填充](#)一节。也可以使用 -Gc 剪裁黑夜区域, 并通过使用 gmt psclip -C 停止区域剪裁, 见[psclip](#)。

-I[<lon>/<lat>][+d<date>][+z<TZ>] 输出太阳的当前位置、方位角和高度角。加上 <lon>/<lat> 则输出日出、日落、正午时间以及一天时间长度。用 +d<date> 指定 ISO 格式的日期时间 (比如 +d2000-04-25T10:00:00) 来计算特定时刻的太阳参数。如果有需要, 也可以通过 +z<TZ> 加上时区。

```
$ gmt pssolar -I120/40+d2016-11-01T01:00:00+z8
Sun current position: long = 160.885756 lat = -14.506894
                                         Azimuth = 38.6720 Elevation = -59.5136
Sunrise = 06:29
Sunset = 16:58
Noon = 11:44
Duration = 10:29
```

-M 将晨昏线数据以多段 ASCII 表格式写到标准输出 (或二进制格式, 见[-b 选项](#))。使用该选项, 则只输出数据不绘图。

-N 反转晨昏线“内”和“外”概念颠倒。仅可与 **-Gc** 一起使用以剪裁出白日区，不可与 **-B** 一同使用。

-T<dcna>[+d<date>][+z<TZ>] 绘制一个或多个不同定义的晨昏线。若需要导出晨昏线数据，见 **-M** 选项。

通过添加 **dcna** 来绘制一个或多个不同定义的晨昏线。其中，

- d 指晨昏线
- c 指民用曙暮光
- n 指航海曙暮光
- a 指天文曙暮光

+d<date> 为 ISO 格式的日期时间（例如 **+d2000-04-25T12:15:00**），以得到该时刻晨昏交替的位置。如果有需要，也可以通过 **+z<TZ>** 加上时区。

不同曙暮光区的定义如下图所示：

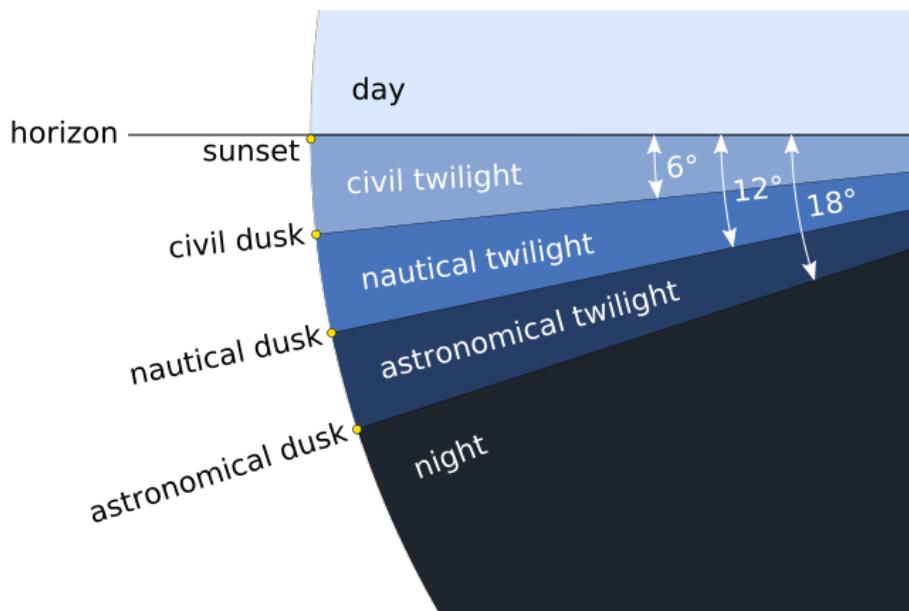


图 12.13: 曙暮光区的多种定义 (图片来自于 <https://en.wikipedia.org/wiki/Twilight>)

- 民用曙暮光分为晨间曙光区和晚间暮光区：
 - 晨间曙光区是指太阳的几何中心位于地平线以下 6° 至地平线以下 $0^{\circ}50'$ (或日出，即太阳上边缘接触地平线) 这段时间
 - 晚间暮光区是指太阳的几何中心位于地平线以下 $0^{\circ}50'$ (或日落，即太阳下边缘接触地平线) 至地平线以下 6° 这段时间
- 航海曙暮光指太阳中心位于地平线以下 $0^{\circ}50'$ 至 12° 这段时间
- 天文曙暮光指太阳中心位于地平线以下 $0^{\circ}50'$ 至 18° 这段时间

下面的命令绘制了晨昏线以及三条曙暮光线：

```
$ gmt pscoast -Rd -W0.1p -JQ0/14c -Ba -BWSen -Dl -A1000 -P -K > terminator.ps
$ gmt pssolar -R -J -W1p -Tdcna -O >> terminator.ps
```

-W[<pen>] 设置晨昏线的画笔属性, 见[画笔](#)。

12.104.3 示例

示例见 <http://gmt-china.org/example/ex009/>

12.104.4 BUGS

1. **-T+d<date>** 在取某些值时会段错误退出 (v5.3.1)

12.105 psternary

[官方文档](#) [psternary](#)

[简介](#) 绘制三角图解

psternary 从文件 (三列或者更多列数据) 或者标准输出中读取数据, 并在三角图中绘制符号。如果给定符号类型, 但是没有给出符号大小, **psternary** 会将第四列数据作为符号大小, 符号大小值小于 0 的将会被跳过。如果没指定符号类型, 就必须在数据的最后一列给出符号代码。

12.105.1 可选选项

-B[a|b|c]<args> 对于三角图, 三条边从下面这条边开始, 逆时针旋转, 分别称为 a、b、c。详细用法类似 **-Bx**, **-By**。

-C<cpt> 使用 cpt 文件, 或者直接使用 **-Ccolor1,color2[,color3,…]** 自动建立一个线性连续的 cpt。如果使用了 **-S** 选项, 符号填充颜色由第四列数值决定。其它字段向右移动一列, 如符号大小将会被放在第五列。

-JXwidth<unit> 线性投影, 仅用来指定三角图的宽度。

-La/b/c 设置三个顶点的标签, 标签距离顶点的距离为 **MAP_LABEL_OFFSET** 设定值的三倍。

-M 不绘图。将三角图数据 ($a,b,c[*z*]$) 转换为笛卡尔坐标 ($x,y,[*z*]$), x,y 为在三角图解基础上标准化后的值。 x,y 取值范围为 0-1, z 的取值范围为 0 到 $\frac{\sqrt{3}}{2}$

如果一个点在三角图中坐标为 (a, b, c), 则笛卡尔坐标 (x, y) 为:

$$x = \frac{(100 - a) + b}{2 \times 100}$$

$$y = \frac{\sqrt{3}}{2 \times 100} \times c$$

-Ramin/amax/bmin/bmax/cmin/cmax 分别给 a、b、c 三边指定最小值最大值。

-S 见 **psxy** 中的 **-S** 选项

12.106 pstext

[官方文档](#) [pstext](#)

简介 在图上写文本

该命令用于在图上添加文本，可以自由控制文本的大小、颜色、字体、角度。

除了可以添加键盘上可直接输入的字符外，还可以通过使用转义字符、特殊字体以及八进制码实现希腊字母、上下标等功能，详见[文字、转义序列、特殊字符](#)。

12.106.1 必选选项

<textfiles> 输入文件。最简单的情况下，输入数据包括三列，其格式为：

```
x  y  text
```

示例：

```
echo 2 2 SHOW TEXTS | gmt pstext -JX10c -R0/5/0/5 -Ba > test.ps
```

若使用了 **-F** 选项，则输入数据的格式为：

```
x  y  [font  angle  justify]  text
```

其中，括号内的三项是否存在以及三项的顺序由 **-F** 选项决定。

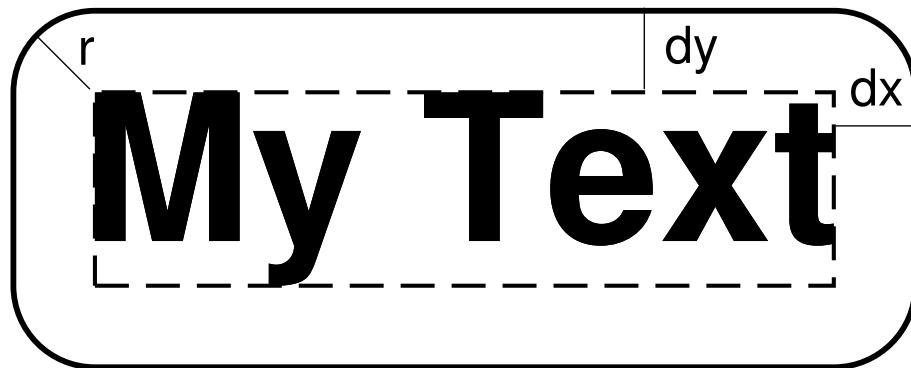
12.106.2 可选选项

-A 默认情况下，输入数据中 **angle** 是指沿水平方向逆时针旋转的角度。**-A** 选项表明 **angle** 是方位角，即相对于北向顺时针旋转的角度。

-C<dx>/<dy> 设置文本框与文本之间的空白区域，默认值为字体大小的 15%。该选项仅当指定了 **-W** 或 **-G** 选项时才有效。

<dx> 可以是具体的距离值也可以接 % 表示空白与当前字号的百分比。例如 **-C1c/1c** 或 **-C20%/30%**。

下图展示了 **-C** 选项的作用。



-D[j|J]<dx>/<dy>[+v[<pen>]] 文本在指定坐标的基础上偏移 **<dx>/<dy>**，默认值为 0/0 即不偏移。

使用 **pstext** 经常遇到的情况是在台站处标记台站名，此时传递给 **pstext** 的位置参数

通常是台站坐标, 因而 `pstext` 会将文本置于台站坐标处, 该选择用于将文本稍稍偏离台站坐标位置以避免文本挡住台站处的符号。

1. 若不指定 `<dy>`, 则默认 `<dy>=<dx>`
2. `-Dj<dx>/<dy>` 表示沿着 `<justify>` 所指定的方向偏移
3. `-DJ<dx>/<dy>` 表示 shorten diagonal offsets at corners by $\sqrt{2}$
4. 偏移量后加上 `+v` 表示绘制一条连接初始位置与偏移后位置的直线
5. `+v<pen>` 控制连线的画笔属性

-F[+a|+A[angle]][+c[justify]][+f[font]][+j[justify]][+h|+l|+r[<first>]]|+t[angle]

控制文本的角度、对齐方式和字体等属性。

1. `+f` 设置文本的字体, 见[文字](#)
2. `+a<angle>` 文本相对于水平方向逆时针旋转的角度
3. `+A<angle>` force text-baselines to convert into the -90/+90 range
4. `+j<justify>` 文本对齐方式, 见[锚点](#)

下面的命令中, 统一设置了所有文本的字号为 30p, 4 号字体, 红色, 文本旋转 45 度, 且以左上角对齐:

```
gmt pstext -R0/10/0/10 -JX10c/10c -B1g1 -F+f30p,4,red+a45+jTL > text.ps << EOF
3 4 Text1
6 8 Text2
EOF
```

若使用了 `+f` 子选项, 但是未给定 ``, 则意味着输入数据的每一行需要自定义本行的字体属性, 因为输入数据的格式要发生变化。例如 `-F+f` 选项要求的输入数据的格式为:

x	y	font	text
---	---	------	------

对于 `+a` 和 `+j` 同理。若 `+f`、`+a`、`+j` 中有两个以上未在命令行中指定参数, 则输入数据中要增加多列, 每列的顺序由这三个子选项的相对顺序决定。比如 `-F+f+a` 的输入数据格式是:

x	y	font	angle	text
---	---	------	-------	------

`-F+a+f` 的输入数据格式为:

x	y	angle	font	text
---	---	-------	------	------

`-F+f+j+a` 表示所有数据都需要单独指定字体、对齐方式和角度, 此时输入数据的格式为:

x	y	font	justification	angle	text
---	---	------	---------------	-------	------

`-F+a+j+f` 与前一个例子类似, 唯一的区别在于子选项的顺序不同, 而输入数据的格

式要与子选项的顺序相匹配，此时输入数据的格式为：

```
x y angle justification font text
```

`-F+f12p,Helvetica-Bold,red+j+a` 为所有行设置了统一的字体，但每一行需要单独指定对齐方式和角度，此时输入数据的格式为：

```
x y justification angle text
```

使用 `+c<justify>` 选项，则输入数据中不需要 XY 坐标，只需要文本即可，该选项直接从 `-R` 选项中提取出范围信息，并由对齐方式决定文本的坐标位置。比如 `-F+cTL` 表示将文本放在底图的左上角，在加上合适的偏移量即可放在任意位置。例如：

```
echo '(a)' | gmt pstext -R0/10/0/10 -JX10c/10c -B1 -F+cTL -Dj0.2c/0.2c > text.ps
```

通常来说，要绘制的文本都来自于输入数据的某一列。对于多段数据而言，还可以使用其他子选项来设置文本的来源：

- `+h` 会直接从多段数据的段头记录中提取文本：

```
gmt pstext -R0/10/0/10 -JX10c/10c -B1 -F+h > text.ps << EOF
> TEXT1
2 2
> TEXT2
5 5
EOF
```

- `+l` 会直接从多段数据的段头记录里的 `-L<label>` 中提取信息：

```
gmt pstext -R0/10/0/10 -JX10c/10c -B1 -F+h > text.ps << EOF
> -LTEXT1
2 2
> -LTEXT2
5 5
EOF
```

- `+r<first>` 会使用记录号作为文本（记录号从 `<first>` 起算）
- `+t<text>` 设置使用一个固定的字符串
- `+z<format>` 将 Z 值以特定的格式输出为字符串， 默认格式由 `FORMAT_FLOAT_MAP` 控制

-G 设置文本框的填充色。

除了设置填充色之外，`-G` 选项还有两个高级用法，即 `-Gc` 和 `-GC`。其中，`-Gc` 表示先绘制文本，然后将文本框裁剪出来，并打开裁剪选项，之后的绘图命令都不会覆盖文本所在区域，最后需要使用 `psclip` 的 `-C` 选项关闭裁剪。若不想要绘制文本只想要激活裁剪选项，可以使用 `-GC` 选项。

-L 用于列出 GMT 所支持的所有字体名及其对应的字号：

```
gmt pstext -L
```

-M 段落模式，用于输入大量文本。

输入文件必须是多段数据。数据段头记录的格式为：

```
> X Y [font angle justify] linespace parwidth parjust
```

1. 第一个字符是数据段开始标识符，默认为 >
2. 从第三列开始，包含了本段文本的设置信息
3. `font angle justify` 是可选的，由 `-F` 选项控制
4. `linespace` 行间距
5. `parwidth` 段落宽度
6. `parjust` 段落对齐方式，可以取为 `l` (左对齐)、`c` (居中对齐)、`r` (右对齐)、`j` (分散对齐)

段头记录后即为要显示在图上的文本，每段数据之间用空行分隔。

```
#!/bin/bash
gmt pstext -R0/3/0/5 -JX3i -F+f+a+j -B1 -M -N -Y4c << EOF > pstext_-M.ps
> 0 -0.5 12p,black 0 LT 13p 3i j
@%5%Figure 1.% This illustration shows nothing useful, but it still needs
a figure caption. Highlighted in @;255/0/0;red@;; you can see the locations
of cities where it is @_impossible@_ to get any good Thai food; these are to be@%
→avoided.
EOF

rm gmt.*
```

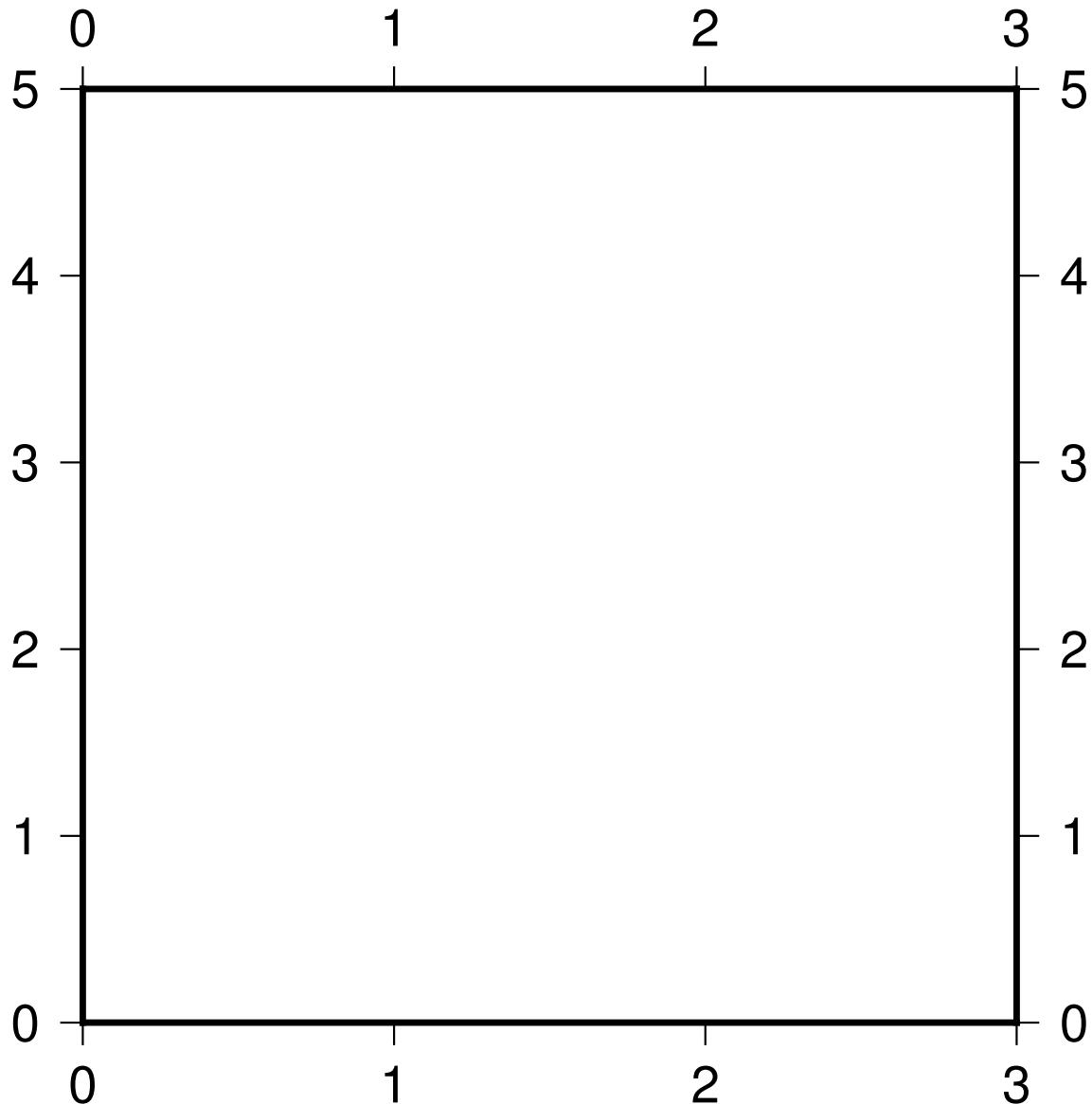


Figure 1. This illustration shows nothing useful, but it still needs a figure caption. Highlighted in red you can see the locations of cities where it is impossible to get any good Thai food; these are to be avoided.

图 12.14: 段落模式示意图

-N 位于地图边界外的文本也被绘制。

默认情况下,若文本超过了底图边框,则不显示该文本,即文本被裁剪掉了。使用 **-N** 选项,即便文本超出了底图边框的范围,也依然会显示。

-Ql|u 所有文本以小写(lower case)或大写(upper case)显示

-To|o|c|C 设置文本框的形状

1. -To : 直角矩形
2. -TO : 圆角矩形
3. -Tc : 凹矩形 (与 -M 选项一起使用)
4. -TC : 凸矩形 (与 -M 选项一起使用)

-W<pen> 设置文本框的边框属性, 默认值为 default, black, solid

-Z 3D 投影中, 需要在数据的第三列指定文本的 Z 位置, 数据格式为:

X	Y	Z	Text
---	---	---	------

此时强制使用 -N 选项。

12.106.3 示例

下面的例子中设置文本框的相关属性: 蓝色边框、淡蓝填充色、圆角矩形, 空白为 100%/100%

```
gmt pstext -R0/10/0/5 -JX10c/5c -B1 -Wblue -Glightblue -TO -C100%/100% > text.ps << EOF
3   1    Text1
6   3    Text2
EOF
```

12.107 psvelo

[官方文档 psvelo](#)

简介 在地图上绘制速度矢量、十字线、楔形图

12.107.1 -S 选项

-S 选项决定了要绘制的符号的类型, 以及输入数据的格式。

-Se<velscale>/<confidence>/<fontsize> 绘制速度椭圆。速度用矢量表示, 速度的不确定度用椭圆表示。输入数据中每列的含义为:

1. 台站经度
2. 台站纬度
3. 速度矢量的东向分量
4. 速度矢量的北向分量
5. 东向分量的不确定度 (1-sigma)
6. 北向分量的不确定度 (1-sigma)
7. 东向分量和北向分量的 correlation
8. 台站名 (可选)

说明:

1. <velscale> 控制了速度矢量的缩放比例, 其对应了速度值为 1 的箭头的长度, 默认单位由参数 PROJ_LENGTH_UNIT 决定, 也可以自己添加长度单位 i|c|p
2. <confidence> 设置了椭圆的 2D 置信区间的极限, 比如 0.95 代表 95% 置信区间

3. <fontsize> 是文本(台站名)的大小
4. 椭圆的填充色由 -G 选项控制
5. 椭圆的轮廓以及矢量的画笔属性由 -W 选项控

-Sr<velscale>/<confidence>/<fontsize> 绘制速度椭圆, 与 -Se 类似, 区别在于输入文件不同:

1. 经度
2. 纬度
3. 速度矢量的东向分量
4. 速度矢量的北向分量
5. 椭圆的半长轴
6. 椭圆的半短轴
7. 椭圆的长轴相对于水平方向逆时针旋转的角度
8. 台站名(可选)

-Sn<barscale> 绘制各向异性矢量, <barscale> 是各向异性矢量的缩放比例。输入数据中每列的含义为:

1. 经度
2. 纬度
3. 各向异性矢量的东向分量
4. 各向异性矢量的北向分量

-Sw<wedge_scale>/<wedge_mag> 绘制旋转饼图。输入数据中每列的含义为:

1. 经度
2. 纬度
3. 旋转的角度(以 radians 为单位)
4. 旋转的不确定度(以 radians 为单位)

<wedge_scale> 设置了饼图的大小。所有的数值在绘图之前会先乘以 <wedge_mag>。饼图的填充色由 -G 选项控制, 不确定度的颜色由 -E 控制。

-Sx<cross_scale> 绘制应变十字线。<cross_scale> 控制十字线的大小。输入文件中每列的含义为:

1. 经度
2. 纬度
3. eps1: 应变张量的扩张本征值, 扩张为正
4. eps2: 应变张量的压缩本征值, 扩张为正
5. eps2 的方位角(北方向顺时针旋转的角度)

12.107.2 选项

-A<params> 修改矢量的属性, 见[绘制矢量/箭头](#)一节

-D<sigma_scale> 控制 -Se 和 -Sr 中速度不确定度以及 -Sw 中旋转不确定度的缩放。该参数可以与 <confidence> 一起控制不确定度的绘制

-F<fill> 控制边框和标注的颜色

-E<fill> 控制 -Sw 中饼图不确定度的填充

-G<fill> 控制颜色？

-L 绘制线段。使用该选择，则椭圆和断层面会根据 **-W** 选项绘制轮廓

-N 不跳过区域范围外的符号

-W<pen> 设置速度箭头、椭圆轮廓以及断层边界的颜色

12.107.3 示例

```
#!/bin/bash
ps=psvelo_ex1.ps

# 左边三个符号
gmt psvelo << EOF -Xc -JM15c -R241/242/34/35 -B0.5 -BWeSn+tpsvelo \
-Sw3c/1.e7 -W1p -G60 -E210 -D2 -P -K > $ps
# lon      lat      spin(rad/yr) spin_sigma (rad/yr)
241.2 34.2  5.65E-08 1.17E-08
241.2 34.5 -4.85E-08 1.85E-08
241.2 34.7  4.46E-09 3.07E-08
EOF

# 右边三个符号
gmt psvelo -J -R -Sw3c/1.e7 -W1p -D2 -O -K << EOF >> $ps
# lon      lat      spin(rad/yr) spin_sigma (rad/yr)
241.7 34.2  5.65E-08 1.17E-08
241.7 34.5 -4.85E-08 1.85E-08
241.7 34.7  4.46E-09 3.07E-08
EOF
gmt pscoast -O -R -J -W1p -Di >> $ps
rm gmt.*
```

psvelo

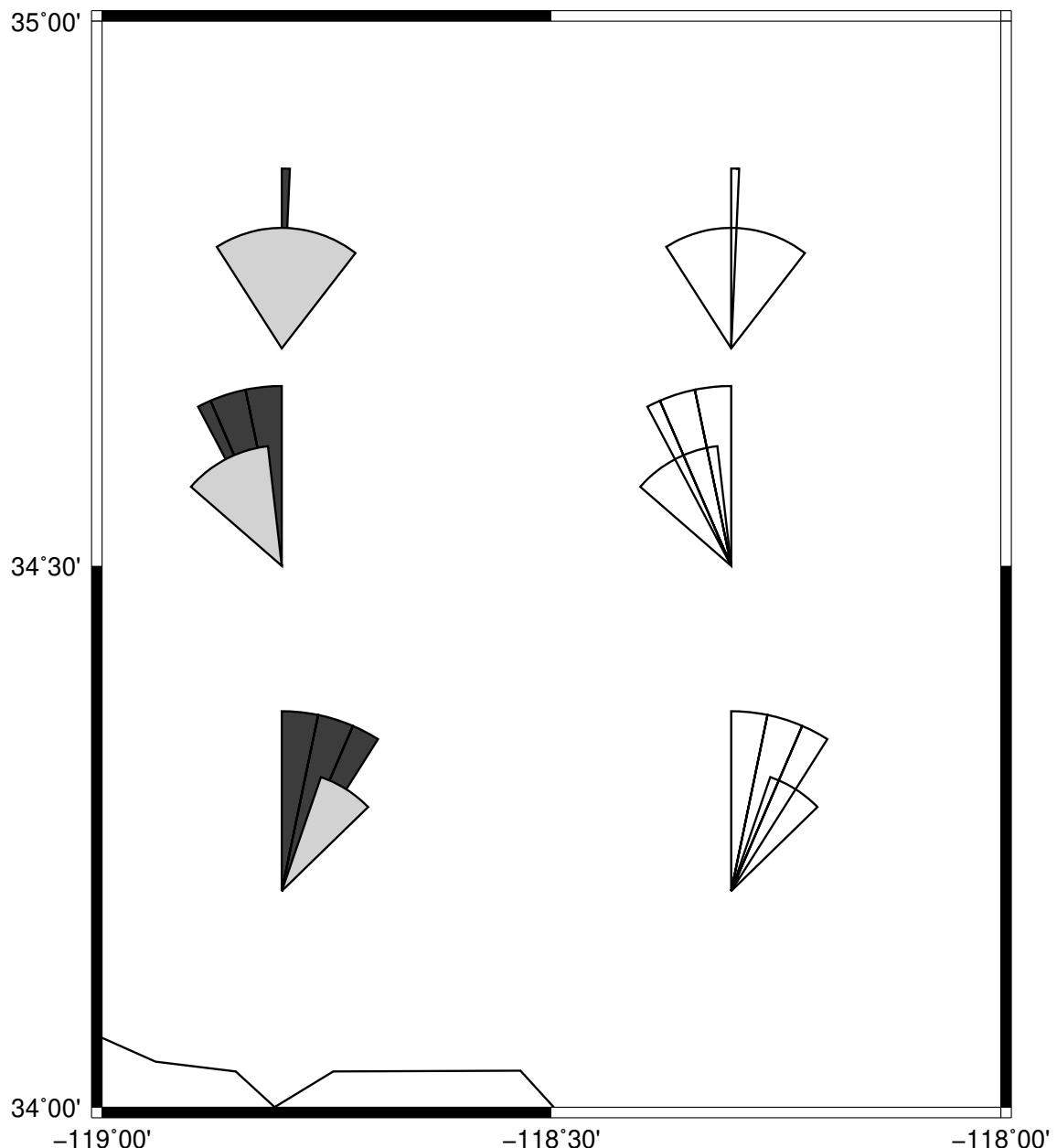


图 12.15: psvelo 示例图 1

```
#!/bin/bash
ps=psvelo_ex2.ps

gmt psvelo -R-10/10/-10/10 -Wthin,red \
           -Se0.2c/0.39/12 -B1g1 -BWeSn -Jx0.2i -Ggreen -Eblue -L -N \
           -A1c+p3p+e -K << EOF > $ps
# Long.   Lat.   Evel   Nvel   Esig   Nsig   CorEN SITE
# (deg)   (deg)   (mm/yr)   (mm/yr)
-8.      0.      5.0    0.0     4.0    6.0    0.500  4x6
-5.      0.      0.0    5.0     4.0    6.0    0.500  4x6
```

(下页继续)

(续上页)

```

-5.    5.    -5.0     0.0      4.0    6.0   0.500  4x6
-8.    5.     0.0    -5.0      0.0    0.0   0.500  4x6
-1.    5.     3.0     3.0      1.0    1.0   0.100  3x3
EOF

gmt psvelo -Se0.2c/0.39/14 -R -J -A0.25c+p0.25p+e -O << EOF >> $ps
# Long.  Lat.   Evel   Nvel   Esig   Nsig  CorEN SITE
# (deg)  (deg) (mm/yr)       (mm/yr)
  0.    -8.     0.0     0.0      4.0    6.0   0.100  4x6
-8.     5.     3.0     3.0      0.0    0.0   0.200  3x3
  0.     0.     4.0     6.0      4.0    6.0   0.300
-5.    -5.     6.0     4.0      6.0    4.0   0.400  6x4
  5.     0.    -6.0     4.0      6.0    4.0  -0.300 -6x4
  0.    -5.     6.0    -4.0      6.0    4.0  -0.500  6x-4
EOF
rm gmt.*

```

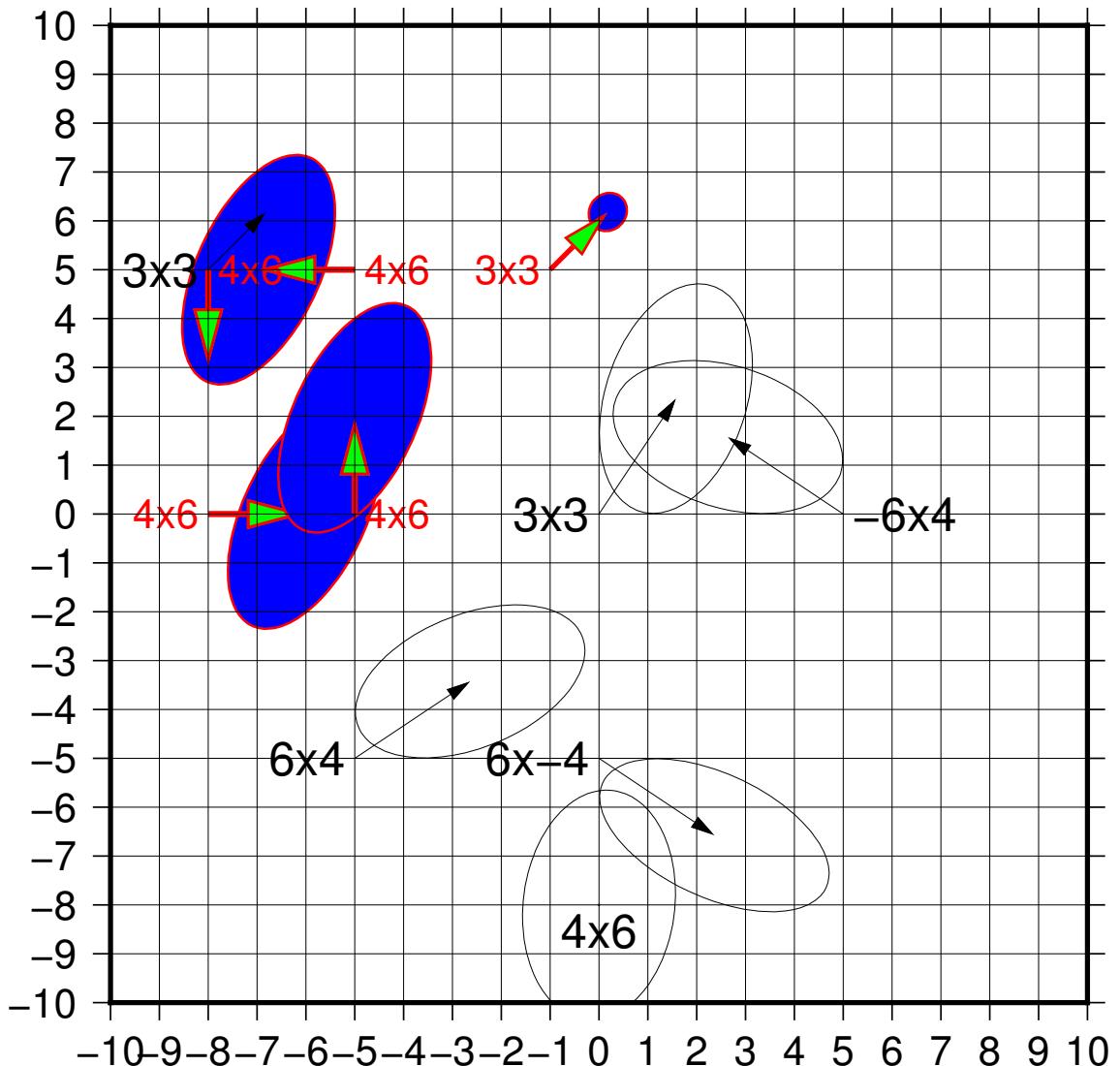


图 12.16: psvelo 示例图 2

下面的示例展示了如何绘制应变十字线:

```
gmt psvelo -JX10c/10c -R0/10/0/10 -Sx10 -W1p > test.ps << EOF
5 5 0.5 0.3 45
EOF
```

12.108 pswiggle

[官方文档 pswiggle](#)

12.109 psxy

[官方文档 psxy](#)

简介 在图上绘制线段、多边形和符号

该命令既可以用于画线段(多边形可以认为是闭合的线段)也可以用于画符号,唯一的区别在于是否使用了-S 选项。在不使用-S 选项的情况下,默认会将所有的数据点连成线,使用-S 选项则仅在数据点所在位置绘制符号。

12.109.1 可选选项

-A[m|p|x|y] 修改两点间的连接方式。

地理投影下,两点之间默认沿着大圆弧连接。

1. -A : 忽略当前的投影方式,直接用直线连接两点
2. -Am : 先沿着经线画,再沿着纬线画
3. -Ap : 先沿着纬线画,再沿着经线画

笛卡尔坐标下,两点之间默认用直线连接。

1. -Ax 先沿着 X 轴画,再沿着 Y 轴画
2. -Ay 先沿着 Y 轴画,再沿着 X 轴画

下图中,黑色曲线为默认情况;红线为使用-A 的效果;蓝线为使用-Ap 的效果;黄线为使用-Am 的效果:

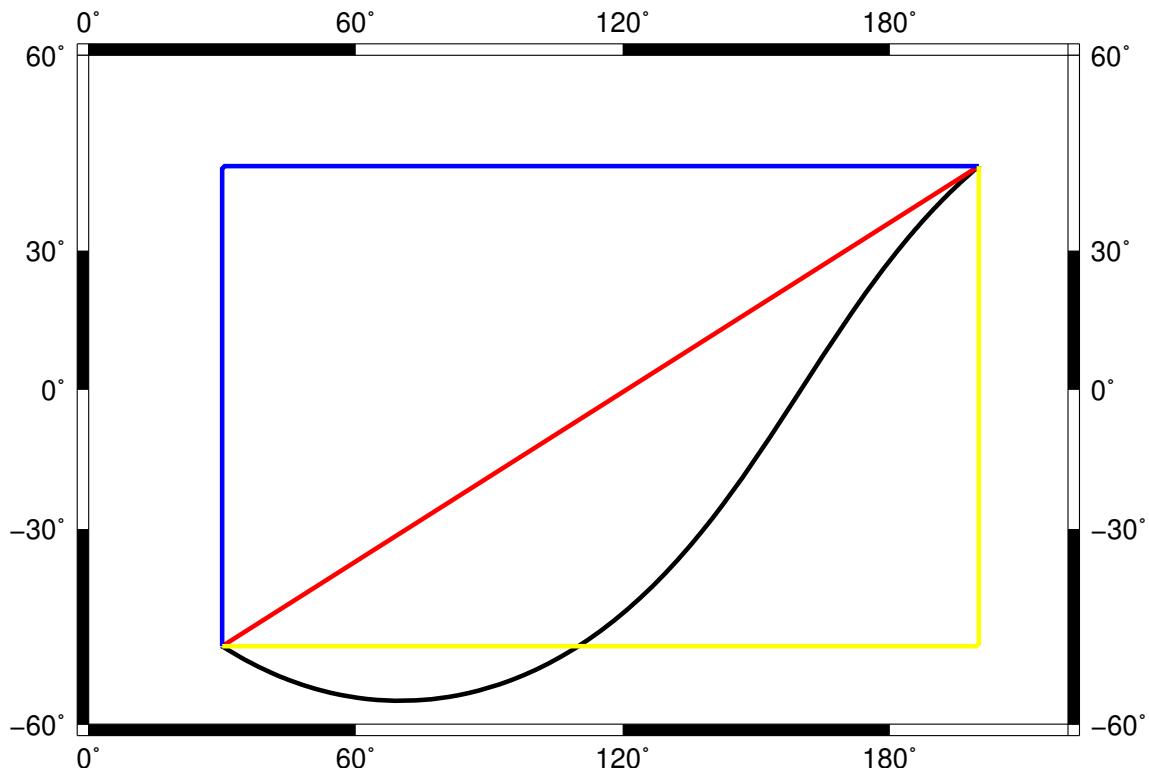


图 12.17: psxy -A 选项示意图

注: 由于这里投影比较特别, 所以沿着经线的线和沿着纬线的线, 看上去都是直线, 在其他投影方式下可能不会是这样。

-C<cpt> 指定 CPT 文件或颜色列表

该选项后跟一个 CPT 文件名, 也可以使用 -C<color1>, <color2>, ... 语法在命令行上临时构建一个颜色列表, 其中 <color1> 对应 Z 值为 0 的颜色, <color2> 对应 Z 值为 1 的颜色, 依次类推。

1. 若绘制符号 (即使用 -S 选项), 则符号的填充色由数据的第三列 Z 值决定, 其他数据列依次后移一列
2. 若绘制线段或多边形 (即未使用 -S 选项), 则需要在多段数据的头段中指定 -Z<val>, 然后从 cpt 文件中查找 <val> 所对应的颜色, 以控制线段或多边形的线条颜色

下面的例子展示了 -C<color1>, <color2>.. 用法:

```
gmt psxy -JX10c/10c -R0/10/0/10 -B1 -Cblue,red -W2p > test.ps << EOF
> -Z0
1 1
2 2
> -Z1
3 3
4 4
EOF
```

-D<dx>/<dy> 设置符号的偏移量。

该选项会将要绘制的符号或线段在给定坐标的基础上偏移 $\langle dx \rangle / \langle dy \rangle$ 距离。若未指定 $\langle dy \rangle$ ，则默认 $dy = dx$ 。

-E[x|y|X|Y][+a][+cl|f][+n][+w<cap>][+p<pen>] 绘制误差棒。

默认会绘制 X 和 Y 两个方向的误差棒。 $x|y$ 表示只绘制 X 方向和/或 Y 方向的误差棒，此时输入数据的格式为（具体格式由选项决定）：

```
X  Y  [size]  [X_error]  [Y_error]  [others]
```

例如，X 方向误差为 1：

```
echo 5 5 1 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Ex -W2p > test.ps
```

X 方向误差为 1, Y 方向误差为 0.5:

```
echo 5 5 1 0.5 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Exy -W2p > test.ps
```

使用 +a 表示 X 方向和/或 Y 方向为非对称误差棒，此时输入数据的格式为：

```
X  Y  [size]  [X_left_error X_right_error]  [Y_left_error Y_right_error]  [others]
```

例如：

```
echo 5 5 1 0.4 0.5 0.25 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Exy+a -W2p > test.ps
```

使用 X 和 Y 则绘制 box-and-whisker (即 stem-and-leaf) 符号。以 -EX 为例，此时数据数据格式为：

```
X 中位数  Y  0% 位数  25% 位数  75% 位数  100% 位数
```

25% 到 75% 之间的方框内可以用 -G 选项填充颜色：

```
echo 5 5 4 4.25 5.4 7 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -EX -Gred -W2p > test.ps
```

- 若使用 -EXY，则输入数据中至少需要 10 列；
 - 若在 X 或 Y 后加上了 +n，则需要在额外的第 5 列数据指定中位数的不确定性。
 - $+w<cap>$ 控制误差棒顶端帽子的长度，默认值为 7p
 - $+p<pen>$ 控制误差棒的画笔属性，默认值为 `defalut,black,solid`
 - 在使用 -C 选项时，可以从 CPT 文件中查找到符号所对应的颜色
 - $+c$ 表明将颜色应用于符号填充色和误差棒画笔属性
 - $+cf$ 表明仅将颜色用于填充符号
 - $+cl$ 表明仅将颜色用于设置误差棒画笔属性，并关闭符号填充色
- F[c|n|r][a|f|s|r|<refpoint>]** 修改数据点的分组和连接方式。

数据的分组方式有三种：

1. **a** 忽略所有数据段头记录，即将所有文件内的所有数据点作为一个单独的组，并将第一个文件的第一个数据点作为该组的参考点
2. **f** 将每个文件内的所有点分在一个组，并将每一组内的第一个点作为该组的参考点
3. **s** 每段数据内的点作为一组，并将每段数据的第一个点作为该组的参考点
4. **r** 每段数据内的点作为一组，并将每段数据的第一个点作为该组的参考点，每次连线后将前一个点作为新的参考点，该选项仅与 **-Fr** 连用（似乎与 **-Fcs** 等效？）
5. **<refpoint>** 指定某个点为所有组共同的参考点

在确定分组后，还可以额外定义组内各点的连接方式：

- **c** 将组内的点连接成连续的线段
- **r** 将组内的所有点与组内的参考点连线
- **n** 将每个组内的所有点互相连线

在不使用 **-F** 选项的情况下，默认值为 **-Fcs**。该选项的具体示例在后面给出。

-G<fill> 设置符号或多边形的填充色。多段数据中数据段头记录中的 **-G** 选项会覆盖命令行中的设置。

-I<intens> 模拟光照效果

<intens> 的取值范围为 -1 到 1，用于对填充色做微调以模拟光照效果。正值表示亮色，负值表示暗色，零表示原色。

-L[+b|d|D][+xl|r|x0][+yl|r|y0][+p<pen>] 构建闭合多边形。

默认情况下，psxy 只将数据点连起来，若首尾两个点不相同，则不会形成闭合多边形。使用 **-F**，则自动将数据的首尾两个点连起来，形成闭合多边形。

除了简单的首尾相连之外，还可以给线段加上包络线（类似于线段的误差）：

1. **+d** build symmetrical envelope around $y(x)$ using deviations $dy(x)$ given in extra column 3
2. **+D** build asymmetrical envelope around $y(x)$ using deviations $dy1(x)$ and $dy2(x)$ from extra columns 3-4.
3. **+b** build asymmetrical envelope around $y(x)$ using bounds $yl(x)$ and $yh(x)$ from extra columns 3-4.
4. **+xl|r|x0** connect first and last point to anchor points at either x_{min} , x_{max} , or x_0
5. **+yb|t|y0** connect first and last point to anchor points at either y_{min} , y_{max} , or y_0 .

Polygon may be painted (**-G**) and optionally outlined by adding **+pen** [no outline].

-N[r|c] 区域范围外的符号不会被裁剪，而会被正常绘制。

默认情况下，位于 $-R$ 范围外的符号不会被绘制的。使用该选项使得即便符号的坐标位于 $-R$ 指定的范围外，也会被绘制。需要注意的是，该选项对线段或多边形无效，线段和多边形总会被区域的范围裁剪。

对于存在周期性的地图而言，若符号出现在重复边界上，则会被重复绘制两次。比如：

```
gmt psxy -R0/360/-60/60 -JM10c -Bx60 -By15 -Sc2c > test.ps << EOF
360 0
EOF
```

会在地图的左右边界处分别两个半圆，该行为可以通过 $-N$ 选项修改：

1. $-N$ 关闭裁剪，符号仅绘制一次
2. $-Nr$ 关闭裁剪，但符号依然绘制两次
3. $-Nc$ 不关闭裁剪，但符号仅绘制一次

-T 忽略所有输入文件，包括标准输入流

该选项会忽略命令行中的输入文件以及标准输入流，在 Linux 下相当于将空文件 `/dev/null` 作为输入文件，因而该命令不会在 PS 文件中绘制任何图形。

该选项有如下几个用途：

1. `gmt psxy -J$J -R$R -T -K > $PS` 只写入文件头，见 $-K$ 和 $-O$ 选项的介绍
2. `gmt psxy -J$J -R$R -T -O >> $PS` 只写入文件尾，见 $-K$ 和 $-O$ 选项的介绍
3. `gmt psxy -J$J -R$R -T -X10c -Y10c >> $PS` 只移动坐标原点而不绘制任何图形

-W[<pen>][<attr>] 设置线段或符号轮廓的画笔属性。

1. $<\text{pen}>$ 见[画笔](#)一节
2. 若使用了 $+cl$ 则表示线条颜色由 CPT 文件控制
3. 若使用了 $+cf$ 则符号的填充色由 CPT 文件控制
4. 若使用了 $+c$ 则表示线条颜色和符号填充色同时由 CPT 文件控制
5. $-W$ 选项后还可以加上额外的选项，可以指定线条的额外属性，见[线条属性](#)一节

12.109.2 **-S** 选项

使用 $-S$ 选项，则表示要绘制符号。 $-S$ 选项的基本语法是：

```
-S[<symbol>][<size>[<u>]]
```

其中 $<\text{symbol}>$ 指定了符号类型， $<\text{size}>$ 为符号的大小， $<\text{u}>$ 为 $<\text{size}>$ 的单位。

不同的符号类型，需要的输入数据格式也不同，但可以统一写成（用 \dots 代表某符号特有的输入列）：

X	Y	...
---	---	-----

-S-|+|**a|c|d|g|h|i|n|s|t|x|y|p** 绘制一些简单的符号。

这几个符号比较简单，输入数据中不需要额外的列：

- **-S-**：短横线, **<size>** 是短横线的长度；
- **-S+**：加号, **<size>** 是加号的外接圆的直径；
- **-Sa**：五角星 (star), **<size>** 是外接圆直径；
- **-Sc**：圆 (circle), **<size>** 为圆的直径；
- **-Sd**：菱形 (diamond), **<size>** 为外接圆直径；
- **-Sg**：八边形 (octagon), **<size>** 为外接圆直径；
- **-Sh**：六边形 (hexagon), **<size>** 为外接圆直径；
- **-Si**：倒三角 (inverted triangle), **<size>** 为外接圆直径；
- **-Sn**：五边形 (pentagon), **<size>** 为外接圆直径；
- **-Sp**：点, 不需要指定 **<size>**, 点的大小始终为一个像素点；
- **-Ss**：正方形 (square), **<size>** 为外接圆直径；
- **-St**：三角形 (triangle), **<size>** 为外接圆直径；
- **-Sx**：叉号 (cross), **<size>** 为外接圆直径；
- **-Sy**：短竖线, **<size>** 为短竖线的长度；

对于小写符号 **acdghinst** , **<size>** 表示外接圆直径；对于大写符号 **ACDHINST** , **<size>** 表示符号的面积与直径为 **<size>** 的圆的面积相同。

下图给出了上面所给出的 symbol 所对应的符号：

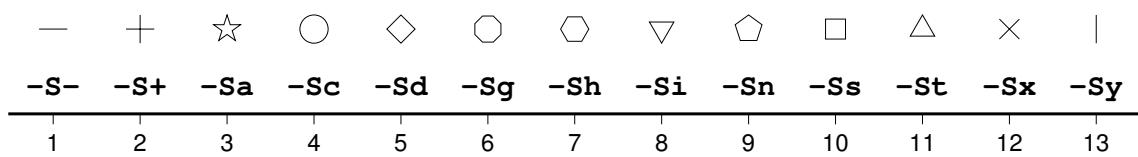


图 12.18: psxy -S 选项示意图

除了上述简单的符号之外，还有更多复杂的符号。

-Sb|B[<size>[<u>]] [b[<base>]] 绘制垂直 bar。

-Sb 用于在 X 坐标处绘制一个从 **<base>** 到 Y 位置的垂直 bar。

1. **<size>** 是 bar 宽度, 其单位可以是长度单位 **c|i|p** , 也可以用 **u** 表示 X 方向单位
2. 若不指定 **b<base>** , 其默认值为 **ymin**
3. 指定 **b<base>** , 为所有数据点指定 base 值
4. 加上 **b** 但未指定 **<base>** , 则需要额外的一列数据来指定 base 的值
5. **-SB** 与 **-Sb** 类似, 区别在于 **-SB** 绘制水平 bar

```
gmt psxy -R0/10/0/5 -JX15c/5c -B1 -Sb1cb > test.ps << EOF
2 3 1 0.5
4 2 1 1.5
8 4 1 2.5
EOF
```

-Se|E 绘制椭圆

-Se 用于绘制椭圆。对于椭圆而言，<size> 是不需要的。此时输入数据的格式为：

X	Y	方向	长轴长度	短轴长度
---	---	----	------	------

其中方向是相对于水平方向逆时针旋转的角度，两个轴的长度都使用长度单位，即 c|i|p

-SE 选项与 -Se 类似，区别在于：

- 第三列为方位角（相对于正北方向旋转的角度）。该角度会根据所选取的地图投影转换成角度
- 对于线性投影，长短轴的长度单位为数据单位，即与 -R 中数据范围的单位相同
- 对于地理投影，长轴和短轴的长度单位为千米，且不可更改

用长度单位指定一个椭圆：

```
echo 180 0 45 5c 3c | gmt psxy -R0/360/-90/90 -JN15c -B60 -Se > test.ps
```

线性投影下 -SE 的长短轴的单位为数据单位：

```
echo 180 0 45 300 100 | gmt psxy -R0/360/-90/90 -JX10c -B60 -SE > test.ps
```

地理投影下 -SE 的长短轴的单位是地理单位，默认长度单位为千米：

```
echo 80 0 45 22200 11100 | gmt psxy -R0/360/-90/90 -JN15c -B60 -SE > test.ps
echo 80 0 45 200d 100d | gmt psxy -R0/360/-90/90 -JN15c -B60 -SE > test2.ps
```

若长短轴长度相等，则椭圆退化成圆，可以用于绘制直径以千米为单位的圆，从而解决了 -Sc 只能用长度单位而不能用距离单位画圆的不足。这一特性可以用于绘制等震中距线。比如如下命令可以绘制 30 度等震中距线：

```
echo 80 0 0 60d 60d | gmt psxy -R0/360/-90/90 -JN15c -B60 -SE > test.ps
```

上面示例的输入数据中，方向和短轴长度都是多余的，所以 GMT 提供了 -SE-[<size>] 选项用于绘制直径为 <size> 的圆，若未指定 <size>，则需要在数据中指定圆的直径。比如 30 度和 60 度等震中距线可以用如下命令绘制：

```
gmt psxy -R0/360/-90/90 -JN15c -B60 -SE- > test.ps << EOF
180 0 60d
180 0 120d
```

(下页继续)

(续上页)

EOF

-Sf<gap>[/<size>][+l|+r][+b+c+f+s+t][+o<offset>][+p[<pen>]] 绘制 front, 即在线段上加上符号以表示断层等 front

1. <gap> 线段上符号之间的距离, 若 <gap> 为负值, 则解释为线段上符号的个数
2. <size> 为符号大小

1. 若省略了 <size>, 则默认为 <gap> 的 30%
2. 若 <gap> 为负值, 则 <size> 是必须的
3. +l 和 +r 分别表示将符号画在线段的左侧还是右侧, 默认是绘制在线段中间
4. +b 符号为 box
5. +c 符号为 circle
6. +t 符号为 triangle
7. +f 符号表示断层 (fault), 默认值。
8. +s 符号表示断层的滑动 (slip), 用于表示左旋或右旋断层。其可以接受一个可选的参数来控制绘制矢量时的角度。也可以用 +S 绘制一个弧形箭头
9. +o<offset> 将线段上的第一个符号相对于线段的起点偏离 <offset> 距离, 默认值为 0
10. 默认符号的颜色与线段颜色相同 (-W 选项), 可以使用 +p<pen> 为符号单独指定颜色, 也可以使用 +p, 即不绘制符号的轮廓。

下面的例子分别绘制了 +b、+c、+f、+s、+t 所对应的符号:

```
#!/bin/bash
R=150/200/20/50
J=M15c
PS=psxy_-Sf.ps
gmt psbasemap -R$R -J$J -B10 -K > $PS
gmt psxy -R$R -J$J -Sf2c/0.1i+l+b -Gblack -W -K -O >> $PS << EOF
155 30
160 40
EOF
gmt psxy -R$R -J$J -Sf2c/0.1i+l+c -Gblue -W -K -O >> $PS << EOF
165 30
170 40
EOF
gmt psxy -R$R -J$J -Sf2c/0.1i+l+f -Gred -W -K -O >> $PS << EOF
175 30
180 40
EOF
gmt psxy -R$R -J$J -Sf2c/0.3i+l+s+o1 -Gyellow -W -K -O >> $PS << EOF
185 30
190 40
EOF
gmt psxy -R$R -J$J -Sf1c/0.1i+l+t -Gwhite -W -B10 -K -O >> $PS << EOF
195 30
200 40
EOF
```

(下页继续)

(续上页)

```
gmt psxy -R$R -J$J -T -O >> $PS
rm gmt.*
```

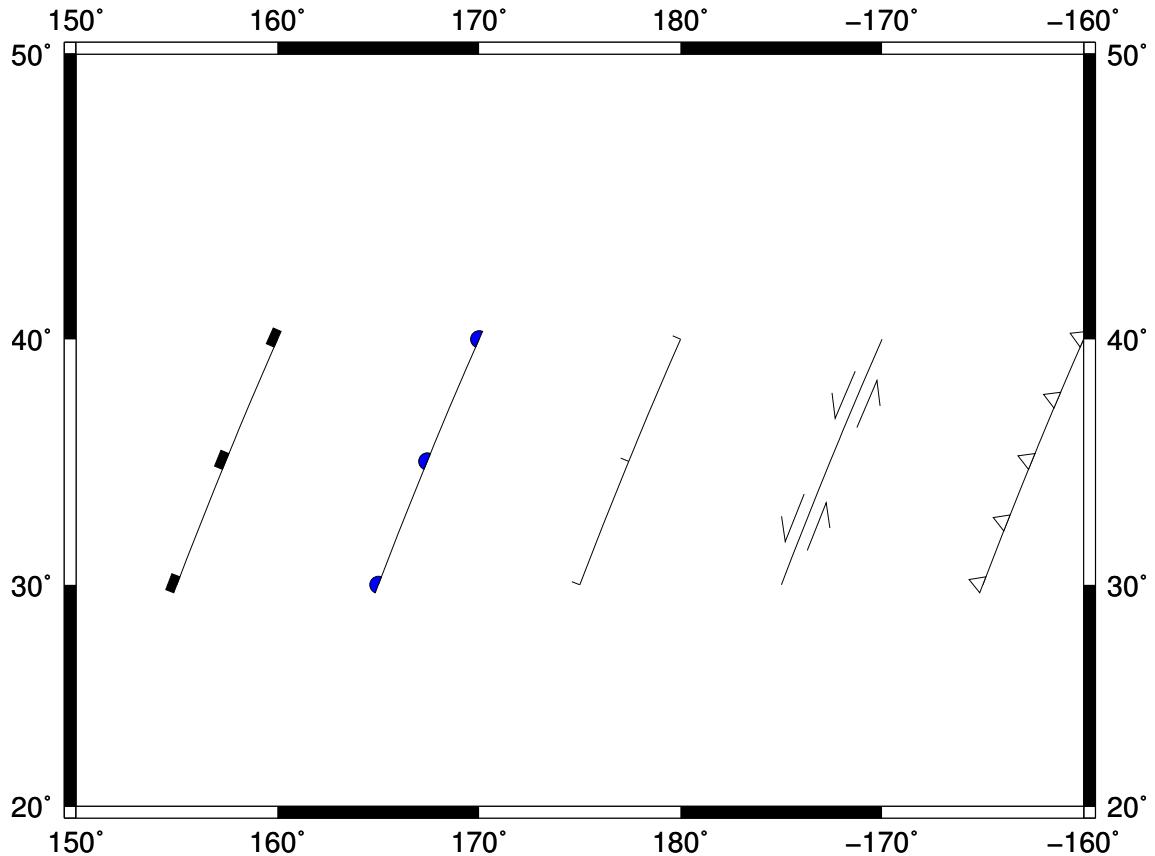


图 12.19: psxy -Sf 示意图

-Sj | J 绘制旋转矩形

其输入数据为:

X	Y	方向	X 轴长度	Y 轴长度
---	---	----	-------	-------

方向为相对于水平方向逆时针旋转的角度。

-SJ 与 **-Sj** 类似, 区别在于:

1. 输入的第三列是方位角
2. 对于地理投影, X 轴和 Y 轴长度的单位为地理单位, 默认为 km
3. 对于线性投影, X 轴和 Y 轴长度的单位与 **-R** 选项中数据范围的单位相同

若矩形的长宽相等, 则矩形退化成正方形, 此时可以使用 **-SJ-<size>**。**<size>** 是正方形的长度, 若未指定 **<size>** 则需要在输入数据的第三列指定长度。

-Sk<name>/<size> 绘制自定义的符号。

GMT 支持自定义符号, 该选项会依次在当前目录、`~/.gmt`、`$GMT_SHAREDIR/custom` 目录中寻找自定义符号的定义文件 `<name>.def`。定义文件中的符号默认

其大小为 1, 然后会根据 `<size>` 对其进行缩放。关于如何自定义符号, 见中文手册。

-Sl<size>+t<string>+j<justify> 绘制文本字符串

该选项的功能与 `pstext` 类似, 不知道为何要设计这个选项。

1. `<size>` 文本串的大小
2. `+t<string>` 指定文本串
3. `+j<justify>` 修改文本串的对齐方式, 默认为 CM

-Sm|M<size> 绘制数学圆弧

输入数据的格式为:

```
X Y radius_of_arc start_direction stop_direction
```

1. `<size>` 为矢量箭头的长度
2. 圆弧的线宽由 `-W` 选项设定
3. `-SM` 选项与 `-Sm` 完全相同, 只是当圆弧的夹角恰好是 90 度时, `-SM` 会用直角符号来表示
4. 圆弧的两端可加上额外的箭头, 见[绘制矢量/箭头](#)一节

```
#!/bin/bash
R=0/4/0.5/1.5
J=X6c/3c
PS=psxy_-Sm.ps

gmt psbasemap -R$R -J$J -Bxa1g1 -Bya0.5g0.5 -BWSen -K > $PS
gmt psxy -R$R -J$J -Sc0.15c -Gblack -K -O >> $PS << EOF
1 1
3 1
EOF
gmt psxy -R$R -J$J -Sm0.2c+b+e+g -Gblack -W0.5p,red -K -O >> $PS << EOF
1 1 10 60
EOF
gmt psxy -R$R -J$J -Sm0.2c+b+l -Gblack -W0.5p,blue -K -O >> $PS << EOF
3 1 1 10 150
EOF
gmt psxy -R$R -J$J -T -O >> $PS

rm gmt.*
```

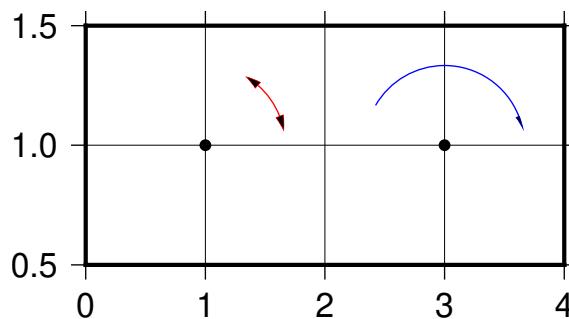


图 12.20: psxy -Sm 示意图

-Sq[<type>]<info>[:<labelinfo>] 绘制 quoted lines, 即带标注的线段, 比如等值线、带断层名的断层线等

<type> 有 6 种可选的方式:

1. d<dist>[<u>]/[<frac>] 指定标签之间的距离, 单位 <u> 为 c|i|p ;
<frac> 表示将第一个标签放在距离 quoted lines 起点 $\frac{1}{<frac>} * <dist>$ 处
2. D<dist>[<u>]/[<frac>] 指定标签之间的距离, 单位 <u> 可以取 e|f|k|M|n|u|d|m|s
3. f<ffile.d> 根据 ASCII 文件 <ffile.d> 的内容确定标签的位置。仅当 <ffile.d> 中指定的标签位置与 quoted lines 上数据点的位置完全匹配时才会被绘制
4. l<line1>[,<line2>, ...] 指定一个或多个以逗号分隔的线段, 标签会放在线段与 quoted line 相交的地方。<line> 的格式为 start_lon/start_lat/ stop_lon/stop_lat , 其中 start_lon/start_lat 以及 stop_lon/ stop_lat 可以用锚点中的两字符替换。
5. L<line1>[,<line2>, ...] 与 l 类似, 只是将线段解释为两点之间的大圆路径
6. n<n_label> 指定等间隔标签的数目, 见官方文档
7. N<n_label> 见官方文档
8. s<n_label> 见官方文档
9. S<n_label> 见官方文档
10. x<xfile.d> 见官方文档
11. X<xfile.d> 见官方文档

<labelinfo> 用于控制标签的格式, 其可以是下面子选项的任意组合, 详情见官方文档:

1. +a<angle>
2. +c<dx>/<dy>
3. +d
4. +e
5. +f
6. +g<color>
7. +j<just>
8. +l<label>
9. +L<label>
10. +n<dx>/<dy>
11. +o
12. +p<pen>
13. +r<min_rad>
14. +t[<file>]
15. +u<unit>
16. +v

17. +w

18. +x[<first>, <last>]

-Sr 绘制矩形

<size> 对该符号无效, 其输入格式为:

X	Y	X 轴长度	Y 轴长度
---	---	-------	-------

-SR 绘制圆角矩形

<size> 对该符号无用。其输入格式为:

X	Y	X 轴长度	Y 轴长度	圆角半径
---	---	-------	-------	------

-Sv|V|= 绘制矢量

-Sv 用于绘制矢量, 输入数据格式为:

X	Y	方向	长度
---	---	----	----

1. <size> 为矢量箭头的长度
2. 矢量宽度由 -W 控制
3. 更多箭头的属性见[绘制矢量/箭头](#)一节
4. -SV 与 -Sv 类似, 区别在于第三列是方位角而不是方向
5. -S= 与 -SV 类似, 区别在于第四列长度的单位是地理单位

```
echo 2 2 45 5c | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sv1c+e -W2p > test.ps
```

-Sw|W[+a]+r] 绘制楔形饼图 (pie wedge), 即饼图中的一个切片。加上 +a 表示只绘制弧线, +r 表示只绘制径向线。

楔形饼图所需要的输入数据格式为:

X	Y	start_direction	stop_direcrion
---	---	-----------------	----------------

1. <size> 是楔形饼图所对应的圆的直径
2. 对于 -Sw , 第三、四列是楔形的开始和结束方向, 其中方向定义为相对于 X 轴正方向 (即东向) 逆时针旋转的角度
3. 对于 -SW , 第三、四列是楔形的开始和结束方位角, 其中方位角定义为相对于北向顺时针旋转的角度。对于地理楔形而言, <size> 代表径向地理距离而不是

下面的示例分别用 -SW 和 -Sw 画了两个不同大小的楔形饼图:

```
#!/bin/bash
R=0/10/0/5
J=x1c
PS=psxy_-Sw.ps
gmt psbasemap -R$R -J$J -Ba1g1 -K > $PS
gmt psxy -R$R -J$J -Sw2c -Glightblue -K -O >> $PS << EOF
2 2 0 45
```

(下页继续)

(续上页)

```

EOF
gmt psxy -R$R -J$J -SW3c -Glightblue -K -O >> $PS << EOF
6 2 0 45
EOF
gmt psxy -R$R -J$J -T -O >> $PS

rm gmt.*

```

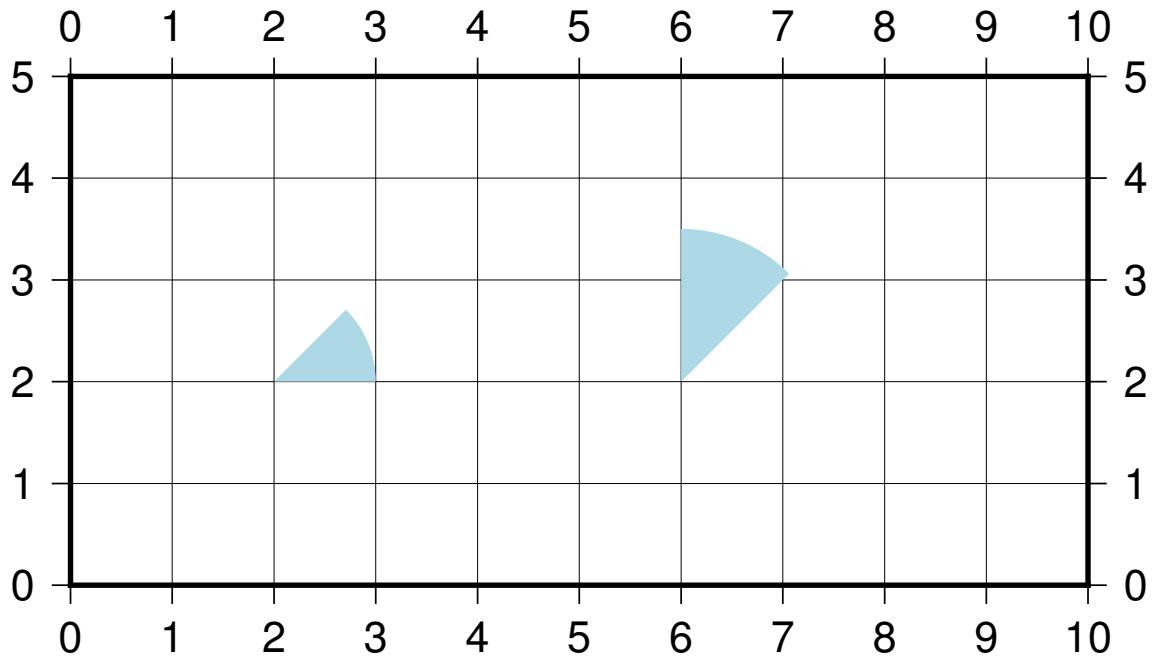


图 12.21: psxy -Sw 示意图。

左边-Sw, 右边-SW; 图中 1 格表示 1cm。

-S~[d|D|f|l|L|n|N|s|S|x|X]<info>[:<symbol>] 绘制 decorated line, 即带有符号的线段。详见官方文档。

12.109.3 输入数据格式

-S 选项相对复杂, 与不同的选项连用, 或者后面接不同的参数, 所需要的输入数据的格式也不同。不管是什么符号, 至少都需要给定符号的位置, 即 X 和 Y 是必须的:

X Y

不同的符号, 可能还需要额外的信息, 统一写成 (用 ... 代表某符号特有的输入列):

X Y ...

若 -S 指定了符号类型但未指定大小, 即 -S<symbol>, 若该符号类型需要指定大小, 则需要将符号大小放在输入数据的第三列, 其他输入数据的列号延后, 此时数据格式为:

```
x  y  size  ...
```

若 size ≤ 0 , 则跳过该记录行。

若 -S 选项后未指定符号代码, 则符号代码必须位于输入文件的最后—列

```
x  y  ...  symbol
```

若使用了 -C 和 -S 选项, 则符号的填充色由数据的第三列决定, 其他字段依次后移:

```
x  y  [z]  ...  symbol
```

因而总结一下输入数据的格式为:

```
x  y  [z]  [size]  ...  [symbol]
```

其中 ... 为某些符号所要求的特殊的数据列, symbol 是未指定符号时必须的输入列, size 是未指定大小时的输入列。

12.109.4 多段数据

对于多段数据而言, 每段数据的头段记录中都可以包含一些选项, 以使得不同段数据拥有不同的属性。头段记录中的选项会覆盖命令中选项的参数:

- -Gfill : 设置当前段数据的填充色
- -G- : 对当前数据段关闭填充
- -G : 恢复到默认填充色
- -W<pen> : 设置当前段数据的画笔属性
- -W : 恢复到默认画笔属性 MAP_DEFAULT_PEN
- -W- : 不绘制轮廓
- -Z<zval> : 从 cpt 文件中查找 Z 值 <zval> 所对应的颜色作为填充色
- -ZNaN : 从 cpt 文件中获取 NaN 颜色

12.109.5 示例

最简单的命令, 绘制线段或多边形, 此时数据输入需要两列, 即 X 和 Y:

```
gmt psxy -R0/10/0/10 -JX10c -B1 > test.ps << EOF
3 5
5 8
7 4
EOF
```

下面的脚本展示了 -F 选项的用法:

```
#!/bin/bash
ps=psxy_-F.ps
```

(下页继续)

(续上页)

```

# 此处定义了函数 plotpts, 用于绘制图中绿色和蓝色的圆圈, 并加上文字
function plotpts
{
    # Plots the two data tables and places given text
    gmt psxy -R -J -O -K -Sc0.2c -Ggreen -Wfaint t1.txt
    gmt psxy -R -J -O -K -Sc0.2c -Gblue -Wfaint t2.txt
    echo $* | gmt pstext -R -J -O -K -F+cTL+jTL+f12p -Dj0.05i
}

# 生成测试用的第一个数据文件
cat << EOF > t1.txt
10      10
48      15
28      20
>
40      40
30      5
5       15
EOF

# 生成测试用的第二个数据文件
cat << EOF > t2.txt
7       20
29     11
8       4
EOF
scl=0.06i
# -Fcs: 默认的连接方式
gmt psxy -R0/50/0/45 -Jx${scl} -P -Ba10 -BWSne -W1p t[12].txt -K > $ps
plotpts TWO DATA TABLES >> $ps

# -Fra
gmt psxy -R -J -Ba10 -BwSnE -W1p t[12].txt -Fra -O -K -X3.25i >> $ps
plotpts DATASET ORIGIN >> $ps

# -Frfr
gmt psxy -R -J -W1p -Ba10 -BWSne t[12].txt -Frfr -O -K -X-3.25i -Y3.15i >> $ps
plotpts TABLE ORIGIN >> $ps

# -Frss
gmt psxy -R -J -Ba10 -BwSnE -W1p t[12].txt -Frss -O -K -X3.25i >> $ps
plotpts SEGMENT ORIGIN >> $ps

# -Fr10/35
gmt psxy -R -J -Ba10 -BWSne -W1p t[12].txt -Fr10/35 -O -K -X-3.25i -Y3.15i >> $ps
plotpts FIXED ORIGIN >> $ps
echo 10 35 | gmt psxy -R -J -O -K -Sa0.4c -Gred -Wfaint >> $ps

# -Fnna
gmt psxy -R -J -Ba10 -BwSnE -W1p t[12].txt -Fnna -O -K -X3.25i >> $ps
plotpts NETWORK >> $ps

gmt psxy -R -J -O -T >> $ps
rm gmt.* t[12].txt

```

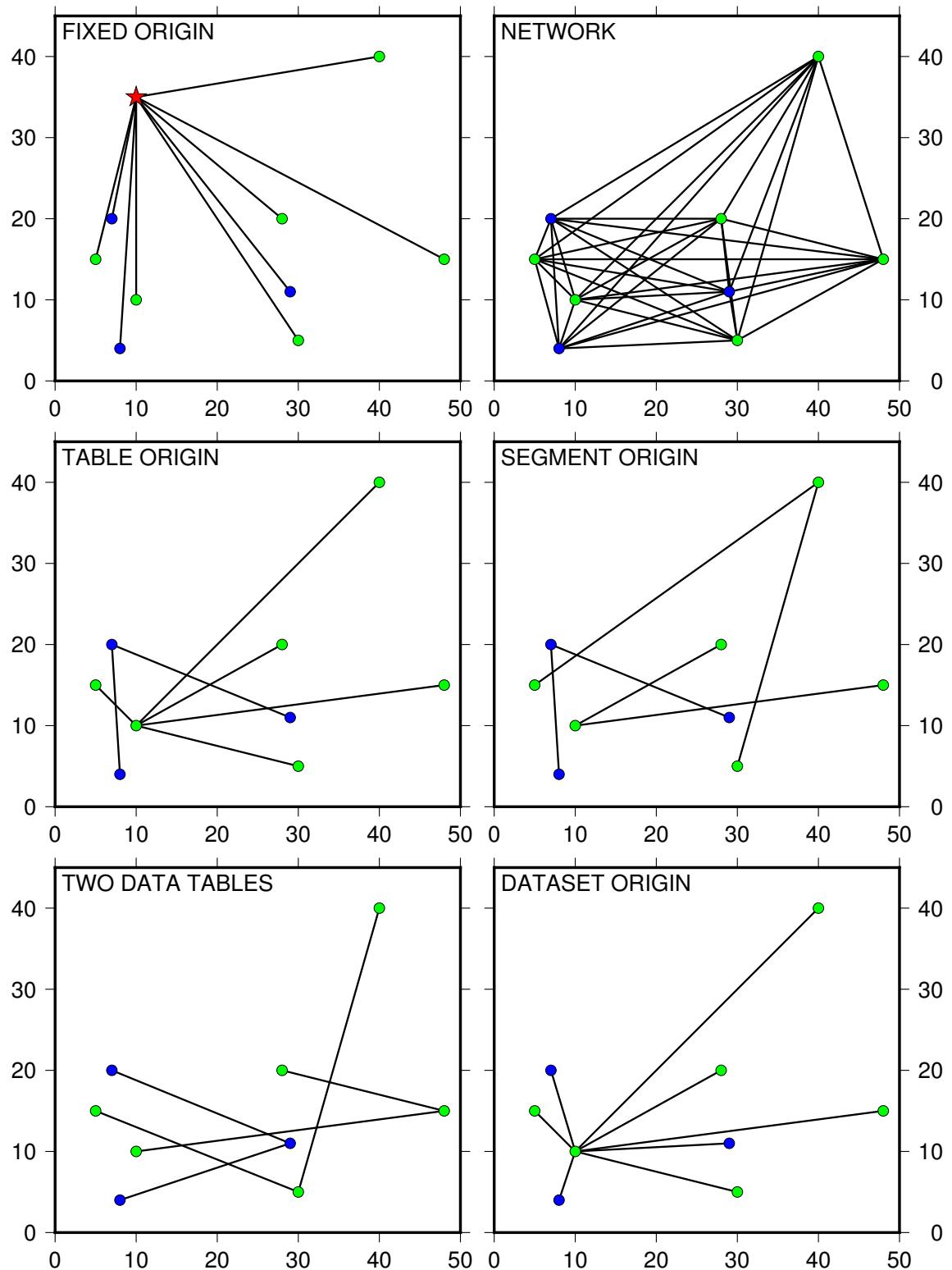


图 12.22: psxy -F 选项示意图

-L 选项的示例:

```
#!/bin/bash
ps=psxy_-L.ps
```

(下页继续)

(续上页)

```
cat << EOF > t.txt
1 1
2 3
3 2
4 4
EOF
gmt psxy -R0/5/0/5 -JX3i -K -B0 t.txt -Gred -W2p -L+xl > $ps
gmt psxy -R -J -O -K -B0 t.txt -Gred -W2p -L+xr -X3.25i >> $ps
gmt psxy -R -J -O -K -B0 t.txt -Gred -W2p -L+x4.5 -X3.25i >> $ps
gmt psxy -R -J -O -K -B0 t.txt -Gred -W2p -L+yt -X-6.5i -Y3.25i >> $ps
gmt psxy -R -J -O -K -B0 t.txt -Gred -W2p -L+yb -X3.25i >> $ps
gmt psxy -R -J -O -K -B0 t.txt -Gred -W2p -L+y4 -X3.25i >> $ps
gmt psxy -R -J -O -T >> $ps
rm gmt.* t.txt
```

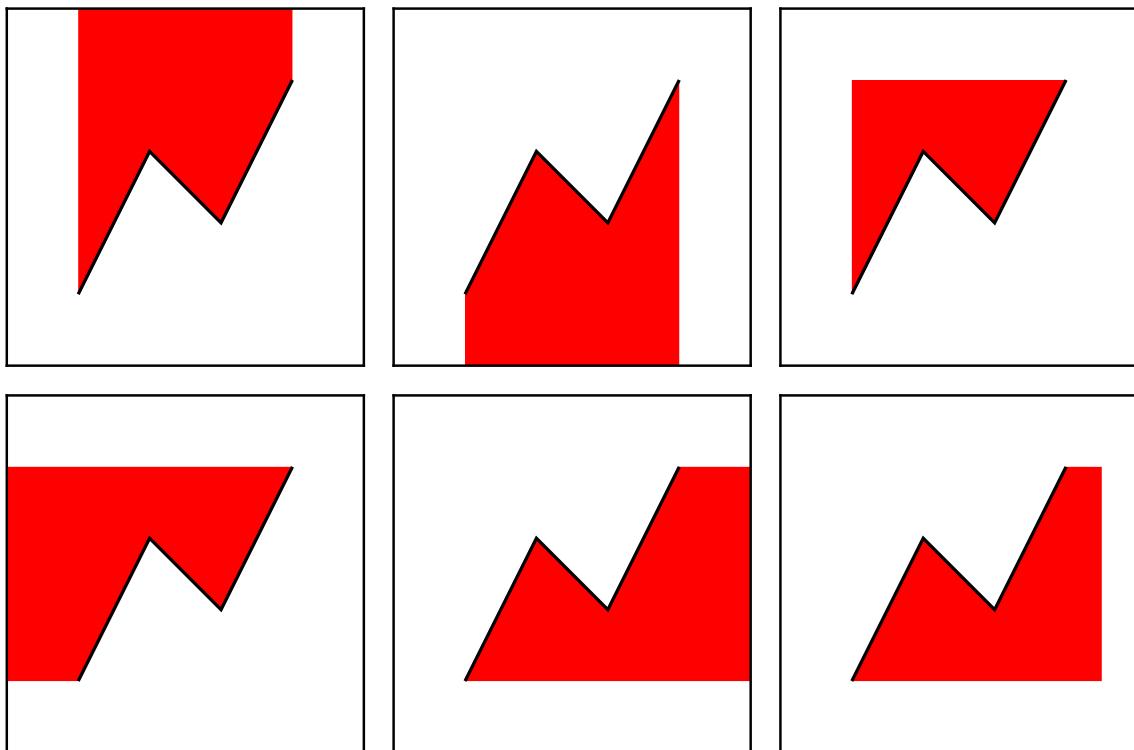


图 12.23: psxy -L 选项示意图

12.110 psxyz

[官方文档 psxyz](#)

12.111 rotconverter

[官方文档 rotconverter](#)

12.112 rotstsmoother

[官方文档 rotstsmoother](#)

12.113 sample1d

[官方文档 sample1d](#)

简介 对 1D 表数据进行重采样

该命令既可以对常规的一维数据（比如时间序列，自变量为时间）进行重采样，也可以对地理坐标下的测线进行重采样。

12.113.1 可选选项

<table> 多列表数据，其中某一列是自变量，其他列为因变量。自变量所在列必须递增或递减。本页面将自变量称为“时间”，因为该命令常用于处理时间序列，实际上自变量可以是任意物理量

-Af|p|m|r|R[+l] 指定插值方式。

- -Af 保留原始数据点，若有必要，则在原始数据点的中间加上额外的点
- -Am 对测线进行采样时，先沿着 Y 方向，再沿着 X 方向
- -Ap 对测线进行采样时，先沿着 X 方向，再沿着 Y 方向
- -Ar 等间距采样
- -AR 等间距采样，但会调整间距以适应自变量的原始长度
- +l if distances should be measured along rhumb lines (loxodromes)

-Fl|a|c|n[+1|+2] 插值方式

- l 线性插值
- a Akima 样条插值
- c natural cubic spline
- n 不插值，取最近的数据点作为插值后的值
- +1|+2 插值的同时计算 spline 的一阶或二阶插值

-I<inc>[<unit>] 默认的等间隔采样间隔是自变量第一个和第二个数据点的间隔，该选项可以自定义采样间隔 <inc>。

加上 <unit> 表明数据文件的前两列包含经纬度信息，重采样后的测线的采样间隔的单位是 <units>。若想要采样笛卡尔坐标下的 (x,y)，则需要指定单位为 c。

-N<knotfile> <knotfile> 中包含了一系列 X 坐标轴，使用该选项则会将原始数据插值到这些 X 坐标轴数据点上。

-S<start>[/<stop>] 对于等间隔采样而言，<start> 是第一个输出值的 X 位置，<stop> 是最后一个输出值的 X 位置。

-T<col> 指定输入数据中的哪列数据是自变量。

12.113.2 示例

输入数据的格式为：

```
time distance gravity magnetics bathymetry
```

使用 Akima spline 插值方式将其采样为 1 千米等间隔:

```
gmt sample1d profiles.tdgmb -I1 -Fa -T1 > profiles_equi_d.tdgmb
```

将 0 到 6 之间的数据用 cubic spline 方式重采样为 0.01 间隔, 不输出数据而是输出一阶偏导 (即斜率):

```
gmt sample1d points.txt -S0/6 -I0.01 -Fc+1 > slopes.txt
```

测线数据中包含经度、纬度和深度, 将其采样为每 2 海里一个点:

```
gmt sample1d track.txt -I2n -AR > new_track.dt
```

同上, 但确保包含了原始数据点:

```
gmt sample1d track.txt -I2n -Af > new_track.dt
```

To obtain a rhumb line (loxodrome) sampled every 5 km instead:

```
gmt sample1d track.txt -I5k -AR+l > new_track.dt
```

12.114 segy2grd

[官方文档](#) [segy2grd](#)

12.115 spectrum1d

[官方文档](#) [spectrum1d](#)

简介 计算一个时间序列的自功率谱, 或两个时间序列的互功率谱

spectrum1d 从标准输入流或数据文件中读取一列或两列数据。这些数据被当作是采样间隔为 **<dt>** 的等间隔的时间序列。**spectrum1d** 采用 Welch 方法, 即加窗多段平均周期图法, 计算输出自功率或互功率谱密度。其输出的功率谱的标准差, 是用 Bendat 和 Piersol 提供的算法。

spectrum1d 的输出文件有三列:

```
f|w   p   e
```

其中, **f** 代表频率, **w** 代表波长, **p** 代表计算的功率谱密度, **e** 代表一个标准差的值。

spectrum1d 的输出文件的文件名是使用统一的前缀 **name_stem**。如果使用了 **-C** 选项, 那么将会有 8 个文件输出, 否则只生成一个功率谱文件 (**.xpower**)。这些文件默认是以 ASCII 码格式, 除非用 **-bo** 选项指定为二进制格式输出。这 8 个文件介绍如下:

1. `name_stem.xpower`: $X(t)$ 的功率谱。单位是 $X \times X \times dt$ 。
2. `name_stem.ypower`: $Y(t)$ 的功率谱。单位是 $Y \times Y \times dt$ 。
3. `name_stem.cpower`: 一致性 (coherent) 的功率谱。单位和 `ypower` 一样。
4. `name_stem.npower`: 噪声的功率谱。单位和 `ypower` 一样。
5. `name_stem.gain`: 增益谱, 或传输函数的模。单位是 Y/X
6. `name_stem.phase`: 相位谱, 或传输函数的相位。单位是弧度。
7. `name_stem.admit`: 导纳 (Admittance) 谱, 或传输函数的实部。单位是 Y/X 。
8. `name_stem.coh`: (平方) 相干谱, 或者线性相关系数 (它是频率的函数)。 i 无单位, 取值范围为 $[0, 1]$ 。信噪比 $SNR = coh / (1 - coh)$ 。当 $coh=0.5$ 时, $SNR=1$ 。

除非使用 `-T` 选项, 否则以上文件会以单个文件单列的形式输出。

12.115.1 选项

-S<segment_size> <segment_size> 是一个 2 的指数组值, 用于控制 Welch 方法中分段平均时的窗口长度。它也决定了功率谱密度的最小频率分辨率和最大频率分辨率, 即 $1.0 / (\text{segment_size} \times dt)$ 和 $1.0 / (2 \times dt)$ (即 Nyquist 频率)。在功率谱密度中的一个标准误差大约为 $1.0 / (\text{n_data}/\text{segment_size})$, 比如 `segment_size=256`, 那么就需要 25600 个数据点去计算一个误差棒的 10%。互功率谱误差棒的计算则需要更多数据点, 而且是相干性的函数, 比较复杂。

table 输入文件名。它是 ASCII 类型的一列数据或两列数据。如果是一列数据文件, 就计算自功率谱; 如果是两列, 就计算互功率谱。若未指定文件名, `spectrum1d` 会从标准输入流中读取数据。

-C[xycnpage] 默认会输出全部 8 个文件。使用该选项可以指定输出 8 个文件中的某些文件。`x=xpower`、`y=ypower`、`c=cpower`、`n=npower`、`p=phase`、`a=admit`、`g=gain`、`o=coh`。

-D<dt> 设置读入的时间序列的时间采样间隔, 默认值是 1。

-L[m|h] 不去除信号中的线性趋势。默认情况下, 在对信号进行变换处理前会先去掉其中的线性趋势。`m` 表示去掉数据的均值, `h` 表示去掉数据的中值。

-N[name_stem] 输出文件名的前缀, 默认为 `spectrum`。若不使用此选项, 则输出的 8 个文件会合到一个文件里。

-T 不让单个分量的结果输出到标准输出流。

-W 输出文件中第一列是波长而不是频率。默认输出时第一列是频率。

12.115.2 示例

假设 `data.g` 是重力数据, 单位为 mGal, 空间采样间隔为 1.5 km。如下命令会输出数据的功率谱, 单位为 mGal^2/km 表示:

```
gmt spectrum1d data.g -S256 -D1.5 -Ndata
```

假设你除了有重力数据 `data.g` 之外, 还有在相同地点测得的地形数据 `data.t`, 单位为 m。计算二者之间的传输函数, 即 `data.t` 是输入, `data.g` 是输出:

```
paste data.t data.g | gmt spectrum1d -S256 -D1.5 -Ndata -C > results.txt
```

12.116 sph2grd

官方文档 [sph2grd](#)

12.117 sphdistance

官方文档 [sphdistance](#)

12.118 sphinterpolate

官方文档 [sphinterpolate](#)

12.119 sphtriangulate

官方文档 [sphtriangulate](#)

12.120 splitxyz

官方文档 [splitxyz](#)

12.121 surface

官方文档 [surface](#)

12.122 talwani2d

官方文档 [talwani2d](#)

12.123 talwani3d

官方文档 [talwani3d](#)

12.124 trend1d

官方文档 [trend1d](#)

12.125 trend2d

官方文档 [trend2d](#)

12.126 triangulate

官方文档 [triangulate](#)

12.127 x2sys_binlist

官方文档 [x2sys_binlist](#)

12.128 x2sys_cross

官方文档 [x2sys_cross](#)

12.129 x2sys_datalist

官方文档 [x2sys_datalist](#)

12.130 x2sys_get

官方文档 [x2sys_get](#)

12.131 x2sys_init

官方文档 [x2sys_init](#)

12.132 x2sys_list

官方文档 [x2sys_list](#)

12.133 x2sys_merge

官方文档 [x2sys_merge](#)

12.134 x2sys_put

官方文档 [x2sys_put](#)

12.135 x2sys_report

官方文档 [x2sys_report](#)

12.136 x2sys_solve

官方文档 [x2sys_solve](#)

12.137 xyz2grd

官方文档 [xyz2grd](#)

简介 将 XYZ 数据或 Z 数据转换成网格文件

该命令读取一个或多个 Z 数据或 XYZ 数据，并将其转换成二进制网格文件。

- 若某些节点没有数据值, 则这些节点会被赋值为 NaN
- 若某个节点由多个数据值, 则该节点的值为所有数据点的平均值

需要注意, 该模块只是将已存在的数据转换为网格数据, 并不具备网格化/插值功能。若需要网格化功能, 可以使用模块 `surface`、`greenspline`、`nearneighbor` 或 `triangulate`。

12.137.1 必选选项

<table> 输入表数据。可以只包含 Z 值, 也可以包含 (x,y,z) 值。可以是 ASCII 格式, 也可以是二进制格式。XYZ 数据不要求排序, Z 数据则必须排序, 且必须使用 -Z 选项。

-G<grdfile> 生成的网格文件名

-I<xinc>[<unit>][+e|+n]/<yinc>[<unit>][+e|+n] 指定 X 和 Y 方向的网格间隔

- `<xinc>` X 方向的网格间隔
- `<yinc>` Y 方向的网格间隔
- `<unit>` 网格间隔的单位。对于地理坐标系而言默认值为度, 可以设置单位为 `m|s|e|f|k|M|n|u`。
- `+e` 微调 X 和 Y 方向范围的最大值, 使得其是网格间隔的整数倍 (默认会微调网格间隔以适应给定的数据范围)
- `+n` 表明 `<xinc>` 和 `<yinc>` 不是网格间隔, 而是 X 和 Y 方向的节点数。此时会根据节点数、网格区域范围以及网格配准方式重新计算网格间隔。

注意:

1. 若 `<yinc>` 设置为 0, 则表示其与 `<xinc>` 相同
2. 若使用 `-R<grdfile>` 选项, 则网格网格已经根据网格文件自动初始化, 此时依然可以使用 `-I` 覆盖网格文件中的间隔值

12.137.2 可选选项

-A[d|f|l|m|n|r|s|S|u|z] 多个数据落在同一个网格节点内时的处理方式。

默认情况下, 若有多个数据落在同一个网格节点内, 会将这些数据的均值作为该节点的值。使用该选项可以控制这一行为:

- `-Af` 将第一个落在该节点内的数据值作为节点值
- `-As` 将最后一个落在该节点内的数据值作为节点值
- `-Al` 将落在该节点内的所有值的最小值作为节点值
- `-Au` 将落在该节点内的所有值的最大值作为节点值
- `-Ad` 将落在该节点内的所有值的最大值和最小值的差作为节点值
- `-Am` 将落在该节点内的所有值的均值作为节点值
- `-Ar` 将落在该节点内的所有值的 RMS 值作为节点值
- `-AS` 将落在该节点内的所有值的标准差作为节点值

- **-An** 将落在该节点内的数据数目作为该节点值, 该选项仅要求输入数据中有 XY 值
- **-Az** 将落在该节点内的所有值的和作为节点值

-D[+x<xname>][+y<yname>][+z<zname>][+s<scale>][+o<offset>][+n<invalid>][+t<title>][+r<remark>]

修改网格文件头段中的基本信息:

- **+x<xname>** X 变量名及其单位, 格式为 varname [unit], 比如 "distance [km]"
- **+s<scale>** 读入网格数据后要乘以的因子, 默认值为 1
- **+o<offset>** 读入数据后并乘以因子后要加入的常数, 默认值为 0
- **+n<invalid>** 指定特定的数用于表示该节点处无有效值, 默认为 NaN
- **+t<title>** 网格文件的标题
- **+r<remark>** 网格文件的注释信息

其他说明:

1. 未指定的项其值保持不变
2. 可以给一个空值以重置某一项, 比如使用 **+t** 而不指定标题则设置标题为空
3. 若文本中包含空格则需要用引号括起来
4. 对于地理数据(比如 **-fg**)而言, **<xname>** 和 **<yname>** 会自动设置

-S[<zfile>] 对输入的 Z 文件做字节序转换并输出到标准输出或保存到文件 **<zfile>** 中。该选项只做字节序转换, 不生成网格文件。该选项必须与 **-Z** 选项连用。

-Z[<flags>] 指定 Z 数据的格式。

对于 Z 数据而言, 由于没有 XY 坐标信息, 因而 Z 数据中必须提供所有节点处的值且按照一定的顺序排列。

在确定数据的排列顺序时, 需要指定数据的起点以及数据排序的方向。

1. 数据起点: 可以取网格的四个顶点中的任意一个作为起点, 四个顶点分别用 TL、TR、BL 和 BR 表示, 其中 T 代表 Top, B 代表 Bottom, L 代表 Left, R 代表 Right。
2. 排序方向: 选取数据顶点之后, 数据可以是行优先也可以是列优先, 所以有两种排序方式。若是行优先, 则 T|B 位于 L|R 的前面; 若列优先, 则 L|R 位于 T|B 的前面。

对于网格线配准的数据, 若数据在 X 方向是周期性的但输入数据中并不包含 $x=x_{\max}$ 处的冗余列, 则加上 **x**; 若数据在 Y 方向是周期性的但输入数据中并不包含 $y=y_{\max}$ 处的冗余行, 则加上 **y**; 加上 **s<n>** 用于跳过数据中最前面的 **<n>** 个字节; 若需要转换数据的字节序, 则加上 **w**。

除此之外, 还需要再加上一个字符表征数据类型:

- **A**: ASCII 表示, 每行一个或多个浮点数

- **a**: ASCII 表示, 每行一项, 不仅可以处理浮点数还可以处理日期时间数据和经纬度数据
- **c**: int8_t, 有符号单字节字符
- **u**: uint8_t, 无符号单字节字符
- **h**: int16_t, 有符号双字节整型
- **H**: uint16_t, 无符号双字节整型
- **i**: int32_t, 有符号四字节整型
- **I**: uint32_t, 无符号四字节整型
- **l**: int64_t, 长整型 (8 字节)
- **L**: uint64_t, 无符号长整型 (8 字节)
- **f**: 4 字节单精度浮点型
- **d**: 8 字节双精度浮点型

默认的输入数据格式为 **-ZTLa**。需要注意, **-Z** 选项仅对 Z 数据有效。

12.137.3 示例

将 ASCII 文件转换为网格数据:

```
gmt xyz2grd hawaii_grv.xyz -D+xdegree+ydegree+zGal+t "Hawaiian Gravity" +r "GRS-80"
→ Ellipsoid used" \
-Ghawaii_grv_new.nc -R198/208/18/25 -I5m -V
```

将二进制格式的 XYZ 文件 (单精度) 转换为网格文件:

```
gmt xyz2grd raw.b -D+xm+ym+zm -Graw.nc -R0/100/0/100 -I1 -V -Z -bi3f
```

将 USGS DEM 数据转换为网格数据:

```
gmt xyz2grd topo30.b -D+xm+ym+zm -Gustopo.nc -R234/294/24/50 -I30s -di-9999 -ZTLhw
```

第 13 章 绘图示例

13.1 绘制底图

13.1.1 线性坐标系

GMT 可以绘制最简单的线性 X-Y 图。

```
gmt psbasemap -R10/70/-3/8 -JX8c/5c -Bx10 -By3 -B+t"Linear X-Y Plot" > GMT_tutor1_1.ps
```

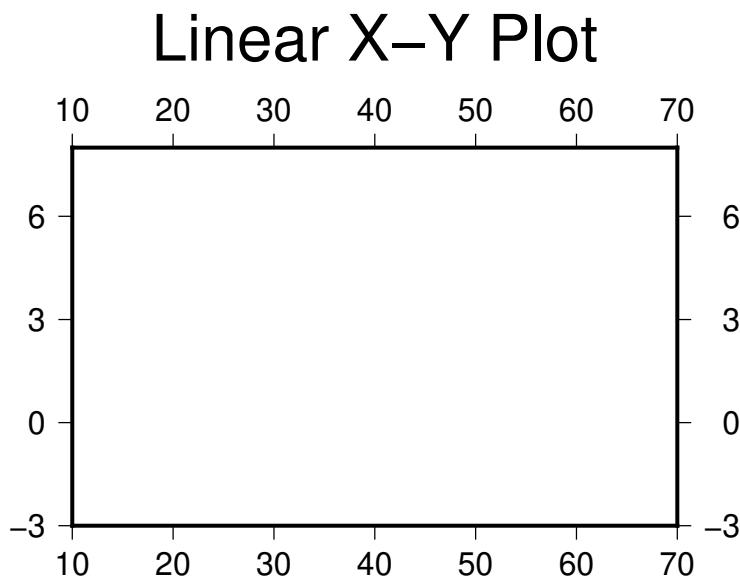


图 13.1: 线性 X-Y 图

在这个示例中:

- `-R10/70/-3/8` 设置了 X 轴范围是 10 到 70, Y 轴范围为 -3 到 8
- `-JX8c/5c` 指定了整张图为线性投影, 图的宽度 (X 轴长度) 为 8 厘米, 图的高度 (Y 轴长度) 为 5 厘米
- `-Bx10 -By3` 分别设置了 X、Y 轴标注以及刻度的间隔为 10 和 3
- `-B+t"Linear X-Y Plot"` 为整张图添加了标题
- `> GMT_tutor1_1.ps` GMT 绘图模块的输出是 PS 代码, 因而需要使用重定向符号 `>` 将 PS 代码输出到 PS 文件中

13.1.2 对数坐标系

下面展示如何用 GMT 绘制对数 X-Y 图。

```
gmt psbasemap -R1/10000/1e20/1e25 -JX15cl/10cl -Bxa2+l"Wavelength (m)" \
-Bya1pf3+l"Power (W)" -BWS > GMT_tutor1_2.ps
```

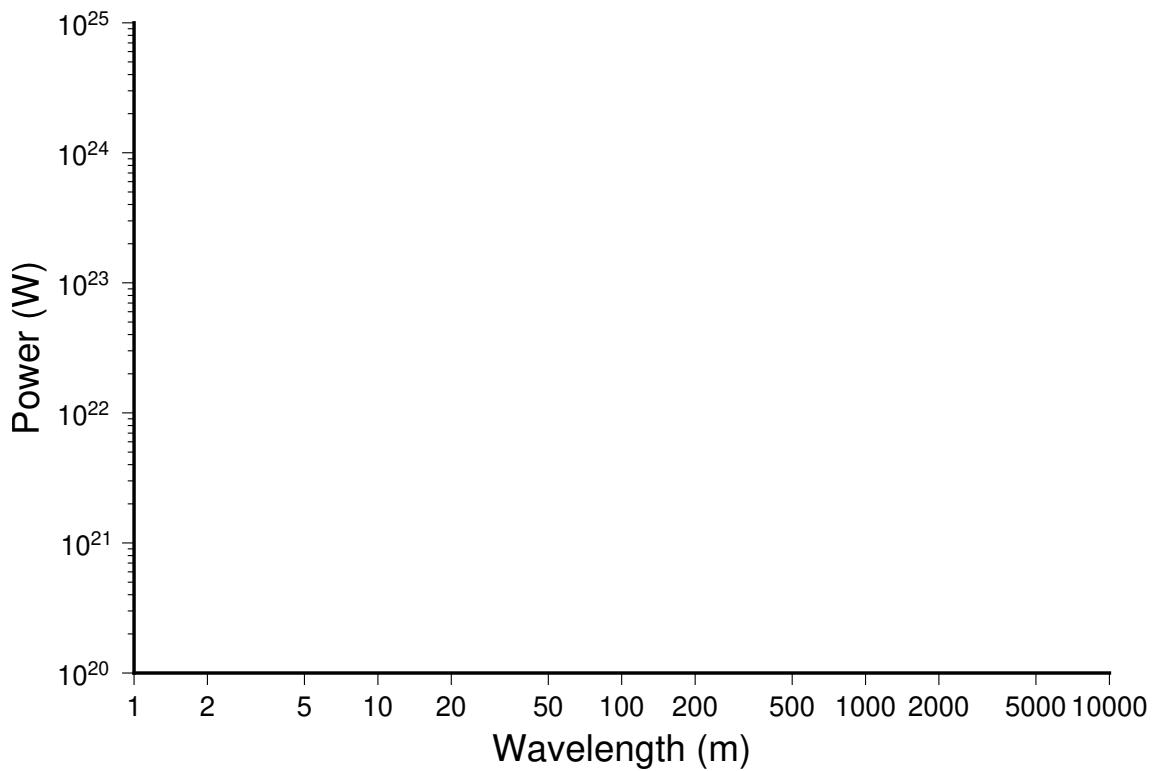


图 13.2: 对数 X-Y 图

此示例中:

- `-R1/10000/1e20/1e25` 设置了 X 和 Y 轴的范围
- `-JX15cl/10cl` 中 `l` 表明用对数轴表示
- `-B` 选项中 `+l` 用于指定每个轴的轴标签
- `-BWS` 表示只绘制图边框左 (W) 和下 (S) 边框

13.1.3 区域地图

GMT 自带了海岸线数据, 通过 `pscoast` 模块可以直接调用。

```
gmt pscoast -R-90/-70/0/20 -JM6i -P -Ba -Gchocolate > GMT_tutor1_3.ps
```

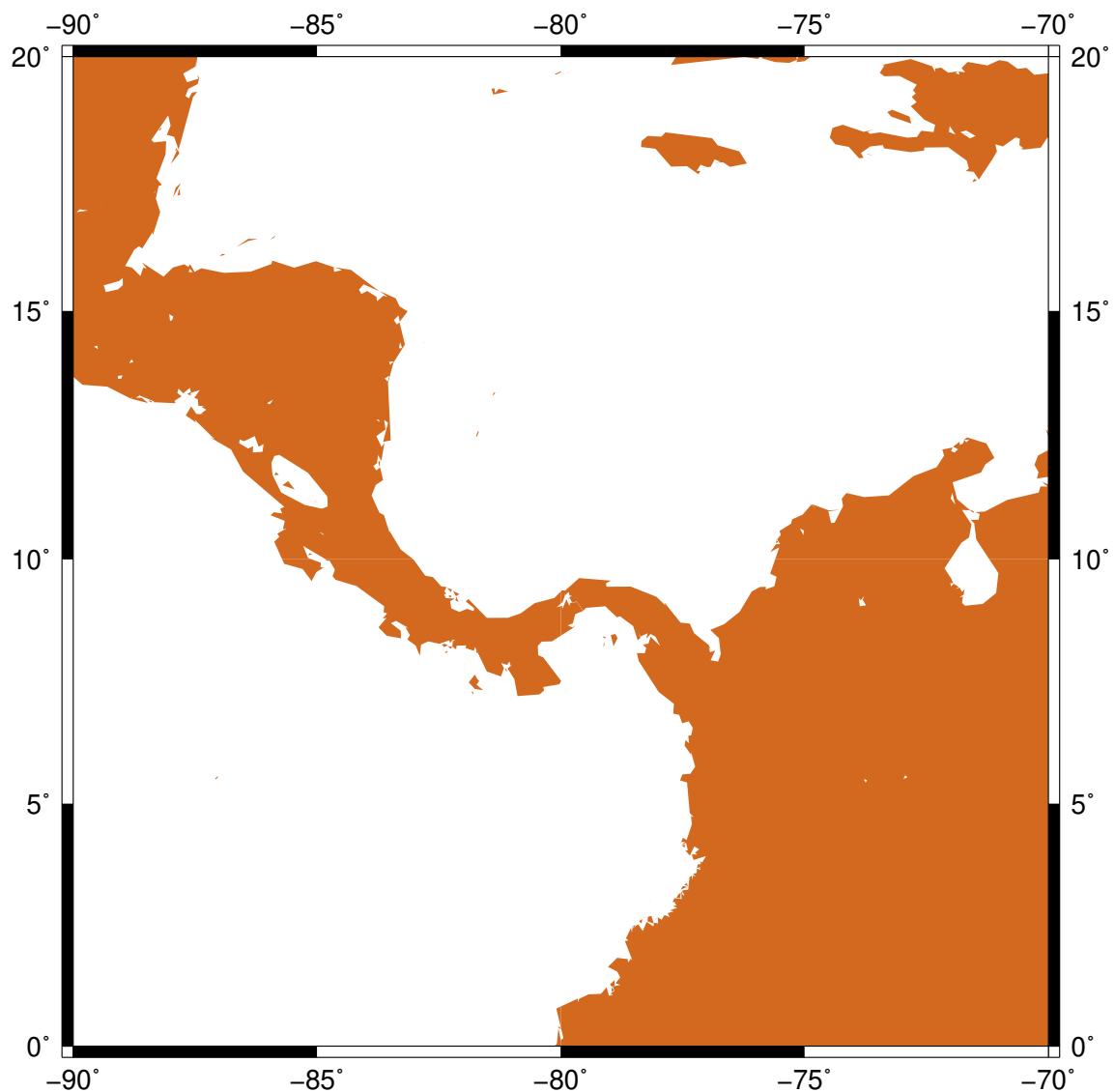


图 13.3: 区域地图

此示例中使用 `pscoast` 绘制了拉丁美洲区域的海岸线。

- `-R-90/-70/0/20` 指定了地理区域的范围
- `-JM6i` 表示使用墨卡托投影，地图的宽度为 6 英寸，高度由投影自动决定
- `-Ba` 会根据地理范围以及图片大小自动计算出适合的标注和刻度间隔
- `-Gchocolate` 将陆地区域填充颜色 chocolate

`pscoast` 还有很多常用的选项：

1. `-D` 选项海岸线数据的精度
2. `-G` 设置陆地区域的填充色
3. `-S` 设置海洋、湖泊区域的填充色
4. `-W` 绘制海岸线，并设置海岸线的画笔属性
5. `-N` 绘制政治边界
6. `-I` 绘制河流

7. -L 在图上绘制比例尺

13.1.4 全球地图

```
gmt pscoast -Rg -JK180/9i -Bag -Dc -A5000 -Gchocolate -SDarkTurquoise \
-Wthinnest > GMT_tutor1_4.ps
```

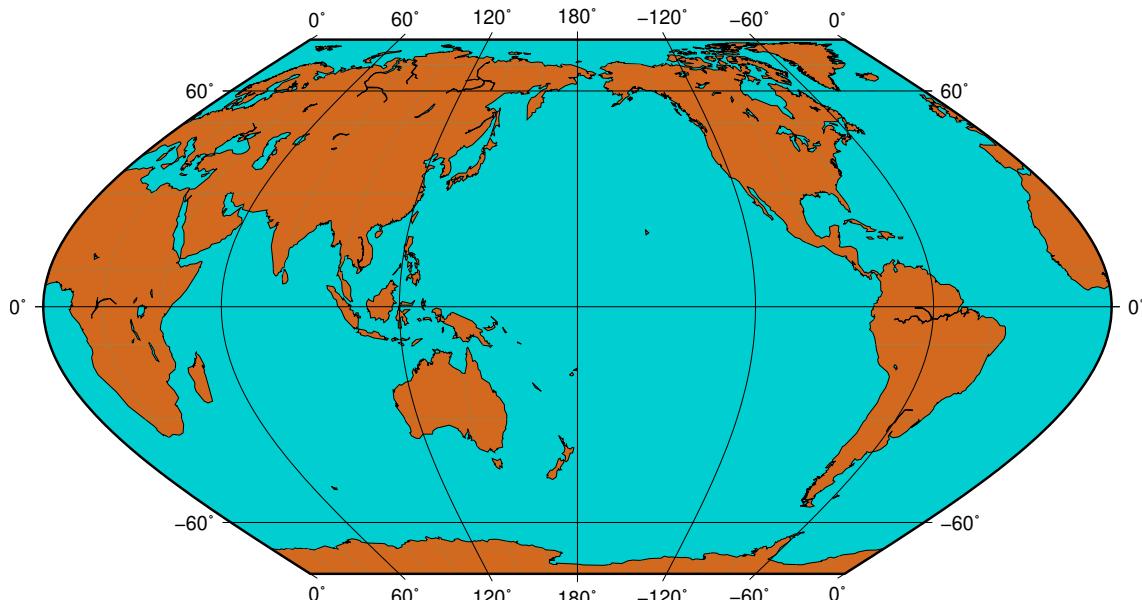


图 13.4: 全球地图

此示例中:

1. -JK180/9i 表明使用 Eckert 投影，地图中心位于经度 180 度，地图宽度为 9 英寸

13.2 绘制双 Y 图

待处理: 纵轴为速度, 下横轴为深度, 上横轴为压强

13.3 绘制间断 Y 轴

13.4 绘制线条

13.5 绘制符号

13.6 绘制海岸线

13.7 绘制震源球

13.8 绘制图例

13.9 绘制地形起伏图

本节描述如何绘制带有颜色和阴影的地形图。主要用到的模块有：

1. `grdcut`: 裁切地形数据
2. `grdgradient`: 制作阴影梯度数据
3. `grdimage`: 绘制地形
4. `pscoast`: 绘制政区边界、湿地(水体)、比例尺
5. `psscale`: 绘制色标图例

绘图过程中需要的 `Tibet.cpt`。

代码为：

```
#!/bin/bash
R=65/18/117/45+r
J=B90/0/40/30/6.5i
PS=GMT_tutor3.ps
D=earth_relief_03m.grd

gmt gmtset FONT_ANNOT_PRIMARY 10p

# 裁剪区域地形数据
gmt grdcut $D -R55/120/10/50 -GTibet.grd

# 计算区域地形梯度
gmt grdgradient Tibet.grd -A0 -Nt -Gint.grad

# 利用 psbasemap 确定格网标注等参数
gmt psbasemap -R$R -J$J -B10g10 -BwSEN -K > $PS

# 绘制地形
gmt grdimage -R$R -J$J -Bg10 Tibet.grd -Iint.grad -CTibet.cpt -K -O >> $PS

# 利用 pscoast 绘制水系、比例尺
gmt pscoast -R$R -J$J -N1/0.5p,white -Ia/0.15p,177/178/183 -I1/0.5p,61/99/172 -C81/174/
->254 -Lg86/18+c30+w1000+u+f -K -O >> $PS
```

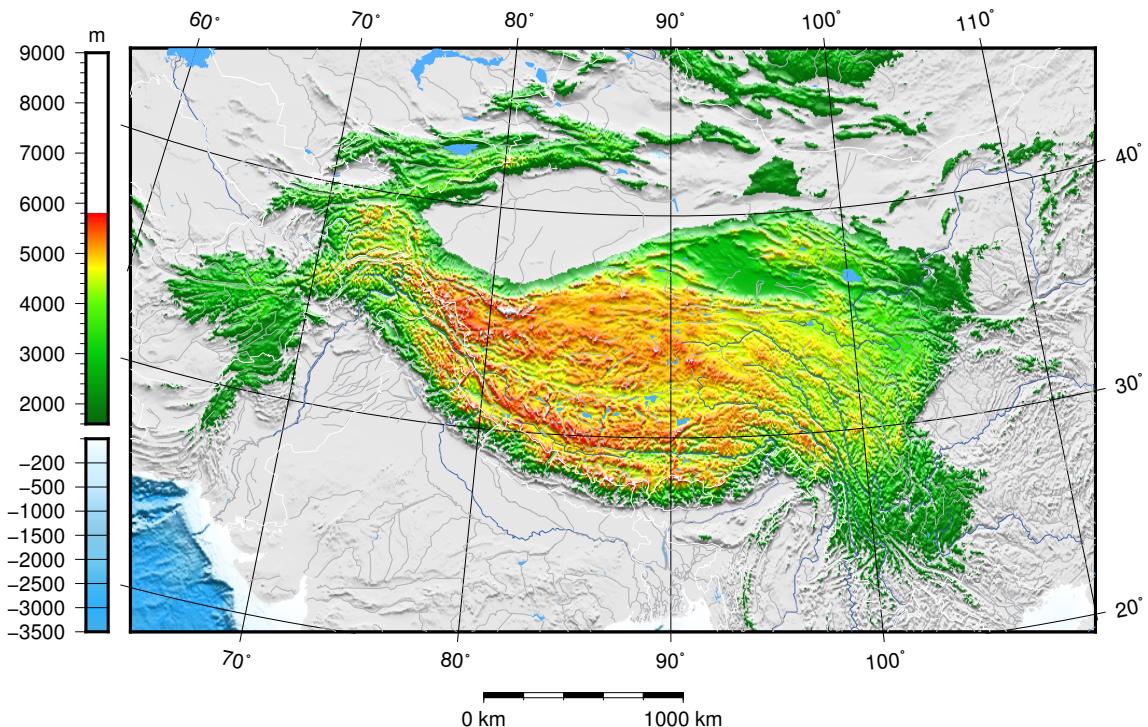
(下页继续)

(续上页)

```
# 绘制两个色标
gmt psscale -Dx-0.3i/1.4i+w2.5i/0.15i+ma -CTibet.cpt -G1600/9000 -By+lm -Bxa1000f200 -K
↪-0 >> $PS
gmt psscale -Dx-0.3i/0+w1.3i/0.15i+ma -CTibet.cpt -G-3500/0 -L -0 >> $PS

rm gmt.* Tibet.grd int.grad
```

结果:



主要参数解释:

1. **grdcut** 指从大区域地形中裁剪出自定义范围的数据（本例中为 55-120°E，纬度 10-50°N），该操作可降低后续梯度计算的复杂度，提高绘图效率。
2. **grdgradient** 模块中: **-A** 设置梯度计算的角度（以北为起始，顺时针计算），也可用 **-Aazim/azim2** 表示计算两个方向梯度并取最大值。**-N** 表示归一化算法，一般有 **-Nt**（累积 Cauchy 分布）和 **-Ne**（累积 Laplace 分布）两种。**-N** 后可接参数，具体参考帮助手册，一般使用默认即可。
3. **grdimage** 模块中: **-I** 接 **grdgradient** 计算得到的梯度文件，**-C** 接地形渲染颜色表文件。GMT5.4 版本中，**-I** 若未指定文件，则自动调用 **grdgradient** 计算梯度。
4. **pcoast** 模块绘制行政边界、水体等，为避免被地形覆盖，需在 **grdimage** 之后进行。**-N1/0.5p,white** 表示以 0.5p 宽的白色线条绘制国界。**-Ia/0.15p,177/178/183** 指绘制所有自然河流、人工运河，177/178/183 为线条颜色的 RGB 参数（浅灰）。**-I1** 表示绘制主要的恒流河，颜色为 61/99/172（深蓝）。**-C** 绘制湖泊，颜色为 81/174/254（天蓝色）。**-L** 绘制比例尺，其中 **g86/18** 指比例尺在图中

的位置是 86°E 、 18°N ；`+c30` 指所绘为 30°N ，中心经度之处的比例尺；`+w1000` 指比例尺长度为 1000km；`+u` 表示在文字标注后显示长度单位，即 km；`+f` 表示比例尺样式为黑白相间的 fancy 样式。

5. `psscale` 模块绘制图中左侧的两个色标。两个色标均绘制了颜色表 `Tibet.cpt` 的一部分，用 `-G` 表示截断范围。`-D` 选项表示色标在图中的位置：如第一句中 `-Dx-0.3i/1.4i` 表示第一个色标绘在左下角原点左移 0.3 inch，上移 1.4 inch 的位置，`+w2.5i/0.15i` 表示色标高 2.5 inch，宽 0.15 inch，`+ma` 表示色标的文字标注位于左侧。

13.10 绘制直方图

13.11 绘制图中图

13.12 绘制 SAC 格式地震波形图

13.13 绘制三角图解

13.14 绘制等值线图

13.15 多子图

绘制多子图与绘制一张图没有本质区别，只是需要通过 `-X` 和 `-Y` 选项 移动每张子图的原点位置即可。同时，在绘制多子图时，通常还需要在每张子图的左上角加上 (a) 等子图编号，这可以通过 `pstext` 来实现。

下面的命令展示了如何绘制四个子图，并给每个子图进行编号。需要注意，绘图时通常从左下角开始画起，因而这里绘制子图的顺序是 c => d => a => b。

```
#!/bin/bash
J=X5c/3c
R=0/5/0/3
PS=subplot.ps

gmt psxy -J$J -R$R -T -K > $PS

# 绘制子图 c
gmt psbasemap -J$J -R$R -Ba -BWSen -K -O >> $PS
# 此处可以放置一堆子图 A 内的绘制命令
# 给子图编号
echo "(c)" | gmt pstext -J$J -R$R -F+cTL+f15p,1 -Dj-1c/-0.75c -N -K -O >> $PS

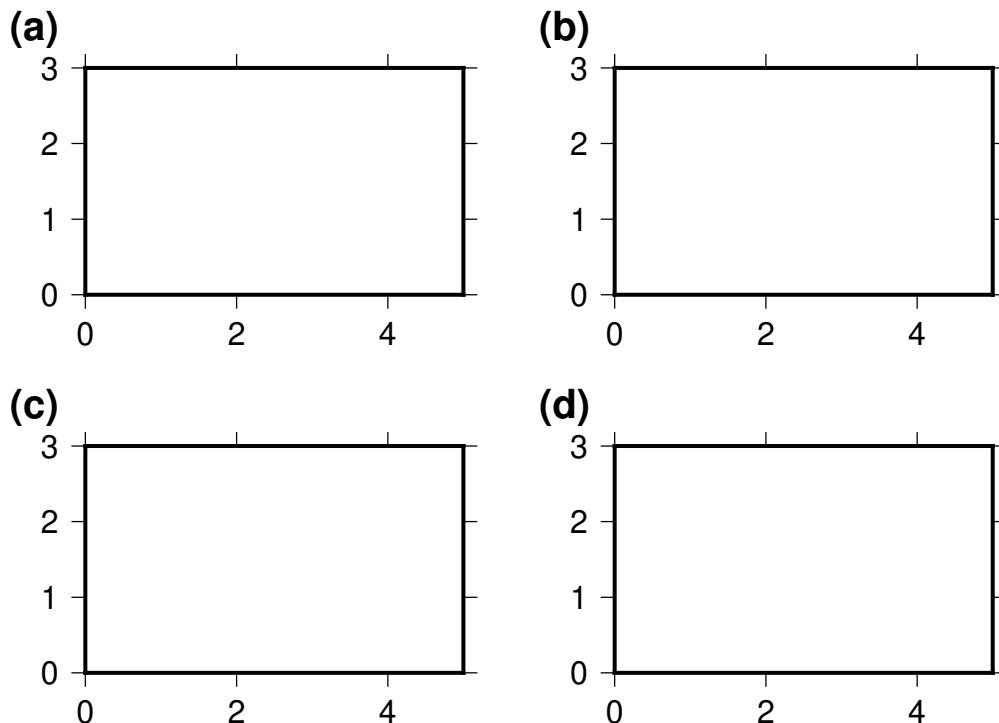
# 绘制子图 d 并编号
gmt psbasemap -J$J -R$R -Ba -BWSen -K -O -X7c >> $PS
echo "(d)" | gmt pstext -J$J -R$R -F+cTL+f15p,1 -Dj-1c/-0.75c -N -K -O >> $PS

# 绘制子图 a 并编号
gmt psbasemap -J$J -R$R -Ba -BWSen -K -O -X-7c -Y5c >> $PS
```

(下页继续)

(续上页)

```
echo "(a)" | gmt pstext -J$J -R$R -F+cTL+f15p,1 -Dj-1c/-0.75c -N -K -O >> $PS
# 绘制子图 b 并编号
gmt psbasemap -J$J -R$R -Ba -BWSen -K -O -X7c >> $PS
echo "(b)" | gmt pstext -J$J -R$R -F+cTL+f15p,1 -Dj-1c/-0.75c -N -K -O >> $PS
gmt psxy -J$J -R$R -T -O >> $PS
```



几点说明:

1. `+cTL` 的作用从 `-R` 选项中提取底图的左上角 (Top Left) 作为当前字符串的坐标;
2. 由于使用了 `+cTL`, 因而 `pstext` 不再需要 XY 坐标信息, 只需要给出字符串 (a)、(b) 即可;
3. `+f15p,1` 将编号字号设置为 5p, 1 号字体 (即粗体)
4. 将编号放在左上角还不够, 还需要使用 `-Dj-1c/-0.75c` 对编号的位置进行微调, 此处表示将编号向左边移动 1 cm 向上移动 0.75 cm
5. 由于编号位于子图的外部, 故而需要加上 `-N` 选项使得 `pstext` 绘制子图外部的字符串

第 14 章 API

GMT 目前提供了 C/Fortran、Matlab、Julia、Python 的程序接口。

14.1 GMT C API

GMT 为 C/Fortran 程序提供了两套 API, 包括 GMT API 和 postscriptlight 绘图库。C/Fortran 用户可以在自己的程序中直接调用这两套 API, 以实现在程序中调用 GMT 模块或底层绘图库的功能。

由于这部分内容涉及到底层 API, 所以通常只有高级用户才会涉及到, 因而不会翻译这部分内容, 高级用户若有需要, 请自行阅读相应文档。

- [GMT API](#)
- [postscriptlight](#)

14.2 GMT Matlab API

14.2.1 简介

GMT 的 Matlab 接口, 顾名思义, 提供了在 Matlab 中调用 GMT 命令的功能。通过该接口, GMT 的所有模块命令都可以在 Matlab 脚本中嵌入执行。GMT 命令生成的结果 (grid 格网数据、table 表格数据、CPT 颜色表、文本文件、图片等) 都可以作为 Matlab 变量进行运算; Matlab 中的矩阵变量也可以直接作为 GMT 的输入。

GMT/MATLAB 工具包用户请引用如下文章:

Wessel, P., and J. F. Luis The GMT/MATLAB Toolbox, *Geochem. Geophys. Geosyst.*, **18(2)**, 811-823, 2017. doi:[10.1002/2016GC006723](https://doi.org/10.1002/2016GC006723).

14.2.2 安装

Windows 平台

GMT5.3 以后的用户在 GMT 执行路径 (默认为 `C:\programs\gmt5\bin`) 下已经存在 `gmt.m` 和 `gmtmex.mexw64|32` 两个文件, 只要确保如下两点即可在 Windows 下使用该接口了。

- GMT 的执行路径已经加入了系统环境变量 path 中, 保证系统可调用 GMT 命令;
- GMT 的执行路径已经加入 Matlab 的搜索路径下, 保证 Matlab 可调用 GMT 命令, 如下图所示。

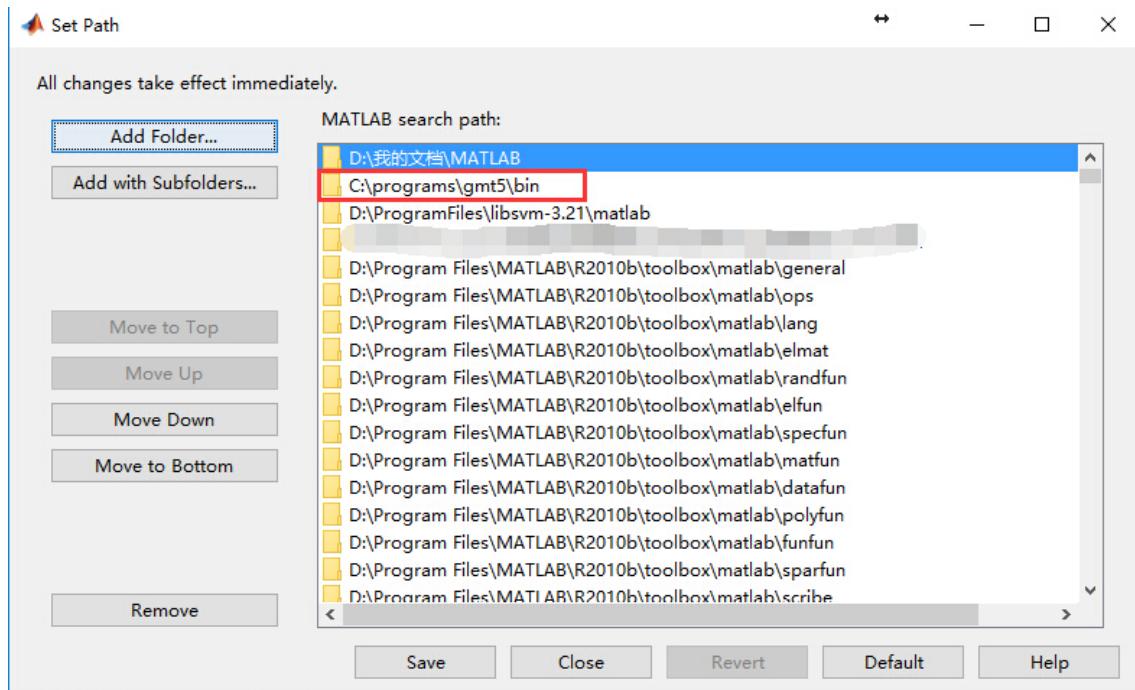


图 14.1: Matlab PATH 设置

测试安装是否正确: 在 Matlab 的命令行窗口直接敲入 `gmt`, 若出现 GMT 的版本及使用方法介绍, 则安装成功。

macOS 平台

在 macOS 上按照如下流程可以成功编译 GMT 的 Matlab 接口。但由于 Matlab 处理动态链接库的方式很特别, 因而该接口可能不太稳定。GMT 开发者正试图与 MathWorks 合作以解决这个问题, 将来以下编译方法可能会修改:

1. 安装 macOS 平台上最新版本的 GMT;
2. 运行安装目录下 `share/tools` 下的 `gmt_prepMex.sh` 文件。此操作会复制 GMT 的已安装文件到 `/opt/gmt` 目录下, 并且会重新检查所有的共享库;
3. 使用 `gmtswitch` 切换当前使用的 GMT 版本, 确保 `/opt/gmt` 下的 GMT 为当前激活版本;
4. 使用 `svn` 获取 `gmt-mex` 项目文件到本地:

```
svn checkout svn://gmtserver.soest.hawaii.edu/gmt-mex gmt-mex
```

5. 进入 `get-mex` 目录并编译生成 `gmtmex.mexmaci64`

```
cd gmt-mex/trunk/
autoconf
./configure --enable-matlab
make
```

6. 将 `gmt.m` 和 `gmtmex.mexmaci64` 所在目录添加到 MTATLAB 路径中
7. 确保 `gmt.conf` 文件中包含选项: `GMT_CUSTOM_LIBS=/opt/gmt/lib/gmt/`

```
plugins/supplements.so
```

经测试，该项目在 2015a、2015b 的 MATLAB 版本中可使用，对于更老版本的 MATLAB，还未进行测试。

Unix/Linux 平台

正在努力开发中，还望有志之士加入…

14.2.3 使用方法

GMT 接口完全模仿了传统的 matlab 命令，可以在命令行、m 文件或 IDE 中使用。形式是：

```
返回参数 = gmt('<module> <module-options>', 输入数据)
```

其中 **输入数据** 可以为 Matlab 的矩阵、结构体或数组等；**返回参数** 可直接在 Matlab 中参与后续的计算。调用 GMT 完毕后，清空缓存：

```
gmt('destroy')
```

入门级示例

在 matlab 环境中调用 **pscoast** 绘制地图：

```
gmt('pscoast -Rg -JA280/30/3.5i -Bg -Dc -A1000 -Gnavy -P > GMT_lambert_az_hemi.ps')
```

上例中，并不存在输入数据，也就是不存在与 Matlab 变量的交互，生成的 ps 文件在 Matlab 当前路径下。

进阶级示例

在 Matlab 环境中，绘制文字：

```
% 创建字符串数组
lines = {'5 6 Some label', '6 7 Another label'};
% 绘制
gmt('pstext -R0/10/0/10 -JX6i -Bafg -F+f18p -P > text.ps ', lines);
gmt('destroy');
```

上例中，字符串数组 **lines** 可以直接作为 **pstext** 的输入参数。

以上为单个输入参数，若需要多个输入参数，如何确定参数的先后顺序？

高手级示例

对一个矩阵数组进行格网化并绘图：

```
% 创建一个 100*3 矩阵, xyz 值均为 0~150 之间的随机数  
t= rand(100,3)*150  
% 利用 GMT 的 surface 命令对 t 进行格网化, 输出为结构体 G, 数组结构见附录  
G = gmt('surface -R0/150/0/150 -I1', t );  
% 利用 grd2cpt 创建颜色表文件, 输出为颜色表结构体 cpt  
cpt = gmt('grd2cpt -Cjet', G);  
% 利用 grdimage 绘制格网化结果  
gmt('grdimage -JX8c -Ba -P -C -G > crap_img.ps', G, cpt);  
gmt('destroy');
```

上例中, `grdimage` 命令需要两个输入参数: 颜色表 `cpt` 和格网数据 `G`, 两者先后顺序不可交换。强制性输入参数(本例中的 `G`)要在所有可选参数(本例中的 `cpt`)之前。若有多个选项参数, 强制性输入参数写在最前, 然后按顺序给出可选参数。

大神级示例

另一个多参数的例子:

```
x = linspace(-pi, pi)'; % 创建 x 值  
seno = sin(x); % 创建 y 值  
xyz = [x seno seno]; % 创建 xyz 三列数据, 其中 y=z  
cpt = gmt('makecpt -T-1/1/0.1'); % 创建 rainbow 颜色表  
% 绘制函数曲线, 以 z 值赋颜色。cpt 和 xyz 先后顺序不可交换。  
gmt('psxy -R-3.2/3.2/-1.1/1.1 -JX12c -Sc0.1c -C -P -Ba > seno.ps', xyz, cpt);  
gmt('destroy');
```

敲黑板, 上例 `psxy` 一句中, `-C` 为可选参数, 因此引号外 `cpt` 要在强制性输入数据 `xyz` 之后。

14.2.4 常见问题

- 使用完 GMT 接口后要记得 `gmt('destroy')` 释放内存, 不然有可能出现不可预知错误。
- `gmt` 括号内直接写 module 名, 看似 GMT4 语句, 实际只支持 GMT5 的语法。
- 绘制地理投影时, 经纬度标注可能会出现 %s 乱码(即使设置为不显示任何度分秒符号), 目前已知 Matlab2016 存在该问题, 其他版本还未有此类反馈。

14.2.5 附录

网格数据结构体

```
proj4          % Proj4 语法的投影方式 (可选)  
wkt           % WKT 语法的投影方式 (可选)  
range         % 1x6 向量, 表示数值范围: [x_min x_max y_min y_max z_min z_max]  
inc           % 1x2 向量, 表示采样间隔: [x_inc y_inc]  
registration   % 格网表达方式: 0 -> Grid registration; 1 -> Pixel registration  
nodata        % 空值对应的数值  
title         % 标题 (可选)  
comment        % Remark (可选)  
command        % 生成网格所使用的命令 (可选)
```

(下页继续)

(续上页)

datatype	% 数据格式 'float' 或 'double'
x	% [1 x n_columns] 向量, 表示 X 坐标值
y	% [1 x n_rows] 向量, 表示 Y 坐标值
z	% [n_rows x n_columns] 网格数组
x_units	% X 轴单位 (可选)
y_units	% Y 轴单位 (可选)
z_units	% Z 单位 (可选)
layout	% 一个三字符的字符串, 用于描述 image memory layout

图像结构体

proj4	% Proj4 语法的投影方式 (可选)
wkt	% WKT 语法的投影方式 (可选)
range	% 1x6 向量, 表示数值范围: [x_min x_max y_min y_max z_min z_max]
inc	% 1x2 向量, 表示采样间隔: [x_inc y_inc]
registration	% 格网表达方式: 0 -> Grid registration; 1 -> Pixel registration (默认值)
nodata	% 空值对应的数值
title	% 标题 (可选)
comment	% Remark (可选)
command	% 生成网格所使用的命令 (可选)
datatype	% 数据格式 'uint8' 或 'int8'
x	% [1 x n_columns] 向量, 表示 X 坐标值
y	% [1 x n_rows] 向量, 表示 Y 坐标值
image	% [n_rows x n_columns] 图像数组
x_units	% X 轴单位 (可选)
y_units	% Y 轴单位 (可选)
z_units	% Z 单位 (可选)
colormap	% CPT 结构体
alpha	% [n_rows x n_columns] alpha 数组
layout	% 一个四字符的字符串, 用于描述 image memory layout

CPT 结构体

colormap	% [ncolors x 3] 矩阵, 保存了 [0-1] 范围内的颜色值
alpha	% [ncolors x 1] 向量, 包含了 [0-1] 范围内的透明度值 (可选)
range	% [ncolors x 2] 矩阵, 对于每个颜色而言包含了 z_low 和 z_high 两个值
minmax	% [z_min z_max]
bnf	% [3 x 3] 矩阵, 包含了 [0-1] 范围内的前景色、背景色和 NaN 色
depth	% CPT 颜色深度, 取值 (1, 8, 24)
hinge	% hinge 值
cpt	
model	% RGB 或 CMYK
comment	% remark (可选)

PostScript 结构体

postscript	% 包含了全部 PostScript 代码的字符串
length	% 字符串的长度 (即字节数)
mode	% 1 表示仅包含文件头, 2 表示只包含文件尾, 3 表示完整为文件
comment	

14.3 GMT Julia API

14.3.1 简介

Julia 是一门为科学计算设计的编程语言，简单易学。其与 Matlab、Python 等编程语言都有相似之处。GMT 提供了 Julia 接口，使得 Julia 用户可以直接在 Julia 脚本中调用 GMT 的相关模块。

GMT 的 Julia 接口的官方地址为：<https://github.com/GenericMappingTools/GMT.jl>

注解： GMT 的 julia 接口还在开发中，因而用法可能出现改动，一切以 <https://github.com/GenericMappingTools/GMT.jl> 中的说明为准。

14.3.2 安装

关于 Julia 的安装请参考 [Julia 官方网站](#) 的相关说明。

启动 Julia 并按照如下方式即可安装 GMT 的 Julia 接口（即 Julia 下的 GMT 模块）：

```
$ julia
      _ _ _(_)_ _ _ | A fresh approach to technical computing
      ( ) | ( ) ( ) | Documentation: http://docs.julialang.org
      _ _ _|_|_ _ _ | Type "?help" for help.
      ||||| | / \^ | |
      |||_| | | | ( ) | |
      _/ | \_ \_ \_ | \_ \_ | | Version 0.5.0 (2016-09-19 18:14 UTC)
      |__| | | | | | | Official http://julialang.org/ release
      |__| | | | | | | x86_64-pc-linux-gnu

julia> Pkg.add("GMT")
INFO: Initializing package repository /home/seisman/.julia/v0.5
INFO: Cloning METADATA from https://github.com/JuliaLang/METADATA.jl
INFO: Cloning cache of Compat from https://github.com/JuliaLang/Compat.jl.git
INFO: Cloning cache of GMT from https://github.com/joa-quim/GMT.jl.git
INFO: Installing Compat v0.23.0
INFO: Installing GMT v0.0.3
INFO: Package database updated

julia> using GMT
INFO: Precompiling module Compat.
```

14.3.3 使用

在 Julia 中调用 GMT 的方式，与直接在命令行中调用 GMT 非常类似。通常来说，一个调用 GMT 的 Julia 脚本具有如下形式：

```
using GMT
```

```
返回值=gmt("<module> <options>", 输入参数)
gmt("destroy")
```

说明：

1. `using GMT` 的作用是在 Julia 中导入 GMT 模块，使得可以在 Julia 中通过 `gmt()` 函数调用 GMT 的所有模块
2. 安装完 GMT 即可后第一次使用 `using GMT` 时，Julia 会对 GMT 即可进行预编译，因而会消耗一段时间，但之后再调用时，速度就非常快了
3. 函数 `gmt()` 用于调用 GMT 模块，其第一个参数与 GMT 命令行版本的参数几乎一致，之后的参数是当前命令所需的输入数据
4. 最后，调用函数 `gmt("destroy")` 以清理不需要的内存

最简单的例子

最简单的情况是，只是简单的绘图，不需要输入数据，也不生成任何数据。

```
using GMT
```

```
gmt("pscoast -Rg -JA280/30/3.5i -Bg -Dc -A1000 -Gnavy -P > GMT_lambert_az_hemi.ps")
```

该 Julia 命令等效于命令行版本的：

```
gmt pscoast -Rg -JA280/30/3.5i -Bg -Dc -A1000 -Gnavy -P > GMT_lambert_az_hemi.ps
```

注解： Julia 接口能够正确使用的前提是 Julia 可以找到 GMT 的动态链接库文件。所以，若以上命令报错，则可以执行：

```
echo 'push!(Libdl.DL_LOAD_PATH, "/opt/GMT-5.4.5/lib64")' >> ~/.juliarc.jl
```

将 GMT 的动态库文件所在目录添加到 Julia 的搜索路径中。

向 GMT 传递数据

GMT 的 `surface` 命令会读入一个文本数据，对其进行插值以生成一个网格文件并绘图，例如：

```
gmt surface input.txt -Goutput.grd -R0/150/0/150 -I1
grdimage -Goutput.grd -JX8c -Ba -P -Cblue,red > crap_img.ps
```

下面看看在 Julia 中如何调用 `surface` 模块：

```
using GMT
t = rand(100,3) * 150;
```

(下页继续)

(续上页)

```
G = gmt("surface -R0/150/0/150 -I1", t);
gmt("grdimage -JX8c -Ba -P -Cblue,red > crap_img.ps", G)
```

本例生成了一个 100×3 的随机数矩阵 t , 并将其作为 `gmt()` 函数的第二个参数, 即将矩阵 t 作为 `surface` 命令的输入数据(即命令行中的 `input.txt`)。同时, 将 `surface` 命令的输出数据(即命令行中生成的网格数据 `-Goutput.grd`)保存到网格变量 G 中。

紧接着调用了 `grdimage` 模块绘制网格变量 G 。注意, 在命令中使用或不使用 `-G` 选项是完全等效的。即上面例子中的最后一个命令也可以写成:

```
gmt("grdimage -JX8c -Ba -P -Cblue,red -G > crap_img.ps", G)
```

向 GMT 传递多个数据

若需要向 GMT 命令传递多个数据, 则输入参数的顺序就变得很重要。

下面的例子在之前例子的基础上先生成了一个 CPT 文件, 再利用 `-C<cpt>` 选项绘图。

```
using GMT
t = rand(100,3) * 150;
G = gmt("surface -R0/150/0/150 -I1", t);
cpt = gmt("grd2cpt -Cblue,red", G);
gmt("grdimage -JX8c -Ba -P -C -G > crap_img.ps", G, cpt)
```

命令行版本中的命令应该是:

```
gmt grdimage -JX8c -Ba -P -Cbluered.cpt -Goutput.grd > crap_img.ps
```

在 Julia 中, CPT 文件和网格文件作为输入数据。参数列表中, 必须的输入(即 G)要放在可选输入(即 cpt)之前, 多个可选输入的参数位置由命令中选项的先后顺序决定。

其他示例

下面的例子展示了如何绘制一条彩色渐变的正弦函数曲线:

```
using GMT
x = linspace(-pi, pi);           # The *** var
seno = sin(x);                  # *yy*
xyz = [x seno seno];           # Duplicate *yy* so that it can be colored
cpt = gmt("makecpt -T-1/1/0.1"); # Create a CPT
gmt("psxy -R-3.2/3.2/-1.1/1.1 -JX12c -Sc0.1c -C -P -Ba > seno.ps", xyz, cpt)
```

注意, 此处输入参数的顺序必须是 xyz , cpt 而不能是 cpt , xyz 。

下面的例子展示了如何写字符串:

```
using GMT
lines = Any["5 6 Some label", "6 7 Another label"];
gmt("pstext -R0/10/0/10 -JM6i -Bafg -F+f18p -P > text.ps", lines)
```

在 Julia 脚本结束时，可以使用：

```
gmt("destroy")
```

以清理不必要的内存。

14.3.4 更多示例

GMT 官方将 GMT 自带的示例用 Julia 重写了一遍，Julia 用户可以阅读并运行这些 Julia 代码。

Julia 示例位于：`~/.julia/v0.5/GMT/test/gallery.jl` 中，你可以直接阅读 Julia 源码。

如果想要执行该 Julia 脚本，你需要将 `~/.julia/v0.5/GMT/test/gallery.jl` 的第 7 行修改为：

```
global g_root_dir = "/opt/GMT-5.4.5/share/"
global out_path = ""
```

并安装如下方式运行示例：

```
using GMT
include("/home/seisman/.julia/v0.5/GMT/test/gallery.jl")
gallery("ex01")
```

14.3.5 附录

`gmt()` 函数会返回多种类型的变量，比如上面例子中涉及到的 CPT 类型和网格类型的变量。因而需要在 Julia 中专门定义相关类型的变量。

Julia 中网格变量 `GMTJL_GRID` 的定义为：

```
type GMTJL_GRID
    proj4::String          # The type holding a local header and data of a GMT grid
    wkt::String             # Projection string in PROJ4 syntax (Optional)
    range::Array{Float64,1}  # 1x6 vector with [x_min x_max y_min y_max z_min z_max]
    inc::Array{Float64,1}   # 1x2 vector with [x_inc y_inc]
    registration::Int      # Registration type: 0 -> Grid registration; 1 -> Pixel
    registration           # The value of nodata
    title::String           # Title (Optional)
    comment::String          # Remark (Optional)
    command::String          # Command used to create the grid (Optional)
    datatype::String          # 'float' or 'double'
    x::Array{Float64,1}       # [1 x n_columns] vector with XX coordinates
```

(下页继续)

(续上页)

```

y::Array{Float64,1}      # [1 x n_rows]    vector with YY coordinates
z::Array{Float32,2}      # [n_rows x n_columns] grid array
x_units::String          # Units of XX axis (Optional)
y_units::String          # Units of YY axis (Optional)
z_units::String          # Units of ZZ axis (Optional)
layout::String            # A three character string describing the grid memory☒
↪layout
end

```

图片变量 GMTimage 的定义为:

```

type GMTimage           # The type holding a local header and data of a GMT image
proj4::String            # Projection string in PROJ4 syntax (Optional)
wkt::String               # Projection string in WKT syntax (Optional)
range::Array{Float64,1}   # 1x6 vector with [x_min x_max y_min y_max z_min z_max]
inc::Array{Float64,1}     # 1x2 vector with [x_inc y_inc]
registration::Int         # Registration type: 0 -> Grid registration; 1 -> Pixel☒
↪registration
nodata::Float64          # The value of nodata
title::String             # Title (Optional)
comment::String            # Remark (Optional)
command::String            # Command used to create the image (Optional)
datatype::String           # 'uint8' or 'int8' (needs checking)
x::Array{Float64,1}        # [1 x n_columns] vector with XX coordinates
y::Array{Float64,1}        # [1 x n_rows]    vector with YY coordinates
image::Array{UInt8,3}       # [n_rows x n_columns x n_bands] image array
x_units::String            # Units of XX axis (Optional)
y_units::String            # Units of YY axis (Optional)
z_units::String            # Units of ZZ axis (Optional) ==> MAKES NO SENSE
colormap::Array{Clong,1}    #
alpha::Array{UInt8,2}       # A [n_rows x n_columns] alpha array
layout::String              # A four character string describing the image memory☒
↪layout
end

```

DATASET 变量 GMTdataset 的定义为:

```

type GMTdataset
header::String
data::Array{Float64,2}
text::Array{Any,1}
comment::Array{Any,1}
proj4::String
wkt::String
end

```

CPT 变量 GMTcpt 的定义为:

```

type GMTcpt
colormap::Array{Float64,2}
alpha::Array{Float64,1}
range::Array{Float64,2}

```

(下页继续)

(续上页)

```
minmax::Array{Float64,1}
bfn::Array{Float64,2}
depth::Cint
hinge::Cdouble
cpt::Array{Float64,2}
model::String
comment::Array{Any,1}    # Cell array with any comments
end
```

PS 变量 GMTps 的定义为：

```
type GMTps
    postscript::String      # Actual PS plot (text string)
    length::Int             # Byte length of postscript
    mode::Int               # 1 = Has header, 2 = Has trailer, 3 = Has both
    comment::Array{Any,1}    # Cell array with any comments
end
```

保护环境，从阅读电子文档开始！

第 15 章 附录

15.1 风格指南

在用 GMT 绘图的时候，遵循一定的风格指南，可以让画图的过程更轻松，让脚本更易读、易改、更健壮、可移植性更高。

15.1.1 使用脚本来执行 GMT 命令

GMT 遵循了 UNIX 的设计思想，将不同的功能分别放在不同的命令中，因而在绘图时需要执行一系列命令。

若使用命令行来执行一系列命令，很容易弄混前一个命令是什么。将所有的绘图命令放在脚本中可以很方便地重复执行一系列命令，以对绘图的细节进行微调。

除非是一两个命令就可以解决的图，否则应一律使用脚本而非命令行。

Windows 下常用的脚本是 bat，Linux 常用的是 Bash、Perl 和 Python。使用什么脚本语言完全依赖于用户个人的需求与喜好，这里以 Bash 脚本为例，其他脚本同理。

```
#!/bin/bash

gmt psxy ...
gmt pscoast ...
gmt grdimage ...
gmt psxy ...
```

15.1.2 不要跨平台写脚本

不要在 Windows 下写 Bash 脚本然后复制到 Linux 下运行；也不要在 Linux 下写 Bat 脚本放在 Windows 下运行。这其中会遇到很多坑，包括但不限于：

- 默认编码不同，Windows 用 GBK，Linux 用 UTF8；
- 换行符不同，Windows 用 \r\n，Linux 用 \n；

如果你真的跨平台写了脚本并遇到各种奇怪的问题时，尝试着新建一个文件，然后把脚本重新手敲一遍。

15.1.3 使用变量

脚本不仅仅只是将一系列命令放在一个文件而已。绘图时有很多需要在多个命令中重复使用的东西，比如设置投影方式的 -J、设置绘图范围的 -R、文件名 xxx.ps。

关于如何使用变量，一般有两种定义方式，这两种方法各有利弊，尚待权衡：

1. 将参数作为变量的值

```
#!/bin/bash
J=M6i
R=0/360/-60/60
Bx=60
By=30
PS=map.ps

gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K > $PS
gmt psxy -J -R -Sa0.5c -Gred -O >> $PS << EOF
160 20
150 30
EOF
```

2. 将选项和参数作为变量的值

```
#!/bin/bash
J=-JM6i
R=-R0/360/-60/60
B=-Bx60 -By30
PS=map.ps

gmt pscoast $J $R $B -W1p -A1000 -K > $PS
gmt psxy -J -R -Sa0.5c -Gred -O >> $PS << EOF
160 20
150 30
EOF
```

15.1.4 不要省略参数

GMT 的一个特性是后面的命令可以继承前面命令的一些参数，比如前面的命令中指定了 `-JM10c -R0/360/-60/60`，后面的命令可以直接使用 `-J -R` 而不用重复给出更多的参数。这样的设计减少了用户的键入。

省略参数虽然带来了一点点方便，但也可能会造成一些麻烦：

1. 写 GMT 脚本时由于需要经常修改、增添命令或调整各个命令之间的顺序。在省略了部分参数的情况下，调整各个命令之间的顺序就可能导致 `-J -R` 出现在第一个，有时会造成意想不到的错误。
2. 参数可以省略本质上是因为之前的命令将标准选项的参数写到了 GMT 历史文件 `gmt.history` 中，因而当在同一个目录里同时运行两个相同或不同的脚本时，两个脚本就会读写同一个 `gmt.history` 文件，进而可能导致一个脚本读到的内容是另外一个脚本写的。

因而，尽量不要省略参数。相同的参数在多个命令里要写很多遍，这样很麻烦，但是因为前面已经把这些参数定义成变量了，所以只是多敲了几个字符而已，因此带来的好处可不少。

```
#!/bin/bash
J=M6i
R=0/360/-60/60
Bx=x60
By=y30
PS=map.ps

gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K > $PS
gmt psxy -J$J -R$R -Sa0.5c -Gred -O >> $PS << EOF
160 20
150 30
EOF
```

15.1.5 开始与结束

多个绘图命令会将 PS 代码依次写入到一个 PS 文件中。绘图命令的顺序有时会影响到成图的效果，最常见的例子就是，如果先 `pscoast` 再 `grdimage`，则 `grdimage` 的效果就会覆盖 `pscoast` 的效果。因而在绘制一张稍复杂的图时，经常需要在原有的代码中增添、删除或修改已有命令的顺序，这个时候尤其需要注意 `-K`、`-O` 以及重定向符号的使用。

下面的代码解决了这个问题：

```
#!/bin/bash
J=M6i
R=0/360/-60/60
Bx=x60
By=y30
PS=map.ps

# 写入 PS 文件头
gmt psxy -J$J -R$R -T -K > $PS

# 一系列绘图命令
gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K -O >> $PS

# 写入 PS 文件尾
gmt psxy -J$J -R$R -T -O >> $PS
```

此处使用了专门的两个命令用于开始和结束一个 PS 绘图。这样做的好处在于：中间的所有绘图命令都使用 `-K -O >>`，不必再考虑这个命令是第一个还是最后一个了，也可以随意删除或修改任何一个命令而不必担心造成其它效果。

因而，实际写绘图脚本时，先把开始和结束这两个命令写对，然后在两个命令的中间写入真正的绘图命令。每新增一个绘图命令，都可以执行一下脚本，以检查绘图效果，若效果正确，则继续添加下一个绘图命令。

15.1.6 命令中选项的顺序

GMT 命令对各个选项的顺序是没有规定的, 所以理论上选项之间怎么排序都可以。但对于大多数命令而言, 选项遵循一定的顺序可以减少错误的发生。

推荐的选项顺序是:

```
gmt 模块名 输入文件 -J -R -B ... -X -Y -K -O >> PSfile
```

总结其规则如下:

1. 如果当前命令需要一个输入文件, 则将输入文件紧跟在模块名的后面
2. `-J` 和 `-R` 选项紧跟在输入文件的后面
3. `-K -O` 位于重定向符号 `>` 或 `>>` 之前
4. `-X` 和 `-Y` 选项放在 `-K -O` 之前
5. 其余选项则放在 `...` 所在位置

15.1.7 使用 SI 单位制

GMT 支持 SI 单位制和 US 单位制, 默认是 SI 单位制。由于 GMT 的开发者是美国人, 官方的文档使用的是 US 单位制, 因而国内的 GMT 用户在学习的过程中也就习惯性地使用了 US 单位制。

实际上, 国内用户对于 US 单位制没有太多的概念, `-X1i` 远远没有 `-X2.5c` 直观。SI 单位制是国际标准单位, 也是中国人熟悉的单位, 使用 SI 单位制会使得微调更简单。

15.1.8 不要依赖于 GMT 的系统设置

你所写的每一个脚本, 将来都可能传给后来人使用, 可能在任一台机器上使用。要保证脚本每次运行的结果完全一致, 并不是一个简单的事情。

不要省略单位

当使用 `-JM10` 时, GMT 会默认使用当前的系统默认单位 (一般来说是 `c`, 也就是厘米), 当脚本在另一台系统默认单位为 `i` 的机器上运行时, 绘图的结果会完全不同。

conf 文件的使用

不要手动修改 `gmt.conf` 文件!

GMT 中提供了 `gmtset` 模块可以用于修改缺省参数, 比如标题的字体、大小等等。该命令会在当前工作目录下生成一个 `gmt.conf` 文件, 进而影响到接下来绘图命令的执行效果。

合理的使用方式如下:

```
#!/bin/bash
```

(下页继续)

(续上页)

```
# 用 gmtset 修改默认参数
gmt gmtset MAP_FRAME_TYPE plain

# 绘图
gmt psxy ...
gmt pscoast ...
gmt psxy ...

# 删除参数文件
rm gmt.*
```

在脚本的最后 `rm gmt.*` 删除了两个临时文件, 一个是 `gmt.history`, 其记录了标准选项的命历史, 另一个是 `gmt.conf`, 记录了当前的参数。

删除这些文件的原因在于:

- 临时文件, 应该删除
- 脚本已经执行完毕, 不应该遗留下无用的文件
- 保留 `gmt.conf` 文件, 可能会导致下次执行脚本时产生不同的效果

有这样一种可怕的情况: 假如你在 `$HOME` 下执行了 `gmtset` 命令, 然后画了一个简单的图, 但是却忘记删除 `$HOME` 下生成的 `gmt.conf` 文件, 这会影响到其它目录中几乎所有 GMT 脚本的执行效果, 而且这个问题很难排查。要避免这种情况的发生需要遵循几个原则:

1. 尽量不要在 `$HOME` 下执行 GMT 命令 (可能会产生临时文件, 难以清理)
2. 尽量不要使用命令行执行 GMT 命令 (因为你很可能会忘记你刚刚执行过哪些命令)
3. 使用 `gmtset` 的脚本, 最后一定要记得删除 `gmt.conf`

15.1.9 -P 选项的使用

只有第一个绘图命令中的 `-P` 选项是起作用的, 所以不需要在每个绘图命令里都使用 `-P` 选项, 当然若是每个绘图命令都使用了 `-P` 选项也没有问题, 只是不够简洁而已。

两种推荐的使用方式:

1. 在开始 PS 文件时使用该选项:

```
#!/bin/bash
J=M20c
R=0/360/-60/60
Bx=x60
By=y30
PS=map.ps

gmt psxy -J$J -R$R -T -K -P > $PS
gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K -O >> $PS
gmt psxy -J$J -R$R -T -O >> $PS
rm gmt.*
```

2. 修改 PS_PAGE_ORIENTATION , 不使用 -P 选项

```
#!/bin/bash
J=M20c
R=0/360/-60/60
Bx=x60
By=y30
PS=map.ps

gmt set PS_PAGE_ORIENTATION portrait
gmt psxy -J$J -R$R -T -K > $PS
gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K -O >> $PS
gmt psxy -J$J -R$R -T -O >> $PS
rm gmt.*
```

15.1.10 不要滥用-B 选项

-B 选项用于绘制边框并控制边框的绘制效果, 即每个使用 -B 选项的命令都会绘制一次边框, 在没有使用 -X 和 -Y 的情况下, 多个命令重复使用 -B 选项会绘制多次边框, 但由于边框是重合的, 所以会看不出来区别。

对于 -B 选项, 合理的用法是仅在第一个命令中使用。

15.1.11 verbose 模式

GMT 命令的输出信息常用于在写脚本时判断命令执行是否正确, 而在真正执行时过多的输出信息反而会扰乱用户的屏幕输出。合理的使用 verbose 模式的方式有三种:

1. 写脚本时每个命令都加上 -V 选项, 待确认脚本正确无误之后删除所有 -V
2. 定义 Verbose 变量

```
#!/bin/bash

J=M20c
R=0/360/-60/60
Bx=x60
By=y30
PS=map.ps

#V=-V      # 调试时用这个
#V=        # 调试完成用这个

gmt psxy -J$J -R$R -T -K -P $V > $PS
gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K -O $V >> $PS
gmt psxy -J$J -R$R -T -O $V >> $PS
rm gmt.*
```

3. 修改缺省参数

```
#!/bin/bash
J=M20c
R=0/360/-60/60
Bx=x60
By=y30
```

(下页继续)

(续上页)

```
PS=map.ps

gmt gmtset GMT_VERBOSE TRUE
gmt psxy -J$J -R$R -T -K > $PS
gmt pscoast -J$J -R$R -B$Bx -B$By -W1p -A1000 -K -O >> $PS
gmt psxy -J$J -R$R -T -O >> $PS
rm gmt.*
```

从使用上的简洁来看，最简单的是第三种方法。

15.1.12 慎用-X 和-Y

使用这两个选项会导致坐标原点的移动。因而使用的时候需要相当慎重。

1. 除极个别的情况外，-X 和 -Y 选项应该仅在绘制组合图（即一张图多个子图）时使用
2. 对于非组合图，也可以在第一个绘图命令中使用 -Xc -Yc 使得整个绘图框架位于纸张的中央
3. 不要仅仅为了将某个符号或文字移动到某个位置就使用这两个选项，如果真的有这种需求的话，应该使用绝对坐标 -Xa1c -Ya1c，其仅影响当前命令的绘图位置

15.1.13 网格文件后缀

GMT 主要使用 netCDF 格式作为网格数据的格式，其标准后缀名为 .nc。

需要注意以下两个事实：

1. GMT 不会对后缀进行检测，所以后缀是什么都不重要
2. GMT 之前的版本中曾经自定义了一种网格数据格式，并使用后缀 .grd，因而很多脚本中都使用了 .grd 作为后缀。

15.2 命令行及脚本基础

15.2.1 命令行基础

本节会简单介绍一下在使用 GMT 的过程中需要掌握的一些与 GMT 无关的基础知识。如无特殊提示，则表明该基础知识既适用于 Linux、macOS 也适用于 Windows。

文件后缀

经常接触的 .doc、.ppt，以及前面提到的 .sh、.bat 等，都是文件后缀。文件后缀的主要作用是表明该文件是什么格式或什么类型的文件。

Windows 下，每一个文件后缀都与特定的应用程序相关联，比如后缀为 .doc 的文件默认都是用 MS Word 打开，当然也可以选择用其他应用程序打开。后缀为 .bat 的文件，在双击的时候会直接被执行，想要编辑的话就需要右键编辑。

Linux/macOS 下，文件后缀就没那么重要了。一个 bash 脚本可以用 .sh 结尾，也可以用 .gmt 结尾，甚至没有后缀。不管后缀是什么，如果用 sh 来执行该脚本，这个脚本就

会被当成 bash 脚本；如果用 `perl` 来执行该脚本，这个脚本就会被当成 Perl 脚本（bash 脚本被当成 Perl 脚本来解释，会直接报错）。因而在 Linux/macOS 下，对文件的后缀并没有严格的要求，后缀的作用仅仅是让用户一眼就可以看出来文件是什么格式而已。

标准输出流和标准错误流

标准输出流 (STDOUT) 用于显示输出数据，标准错误流 (STDERR) 用于显示错误消息。一般来说，标准输出流和标准错误流都是屏幕。

GMT 中，模块的正常输出都发送到标准输出流，模块的语法、警告、错误以及进程报告都发送到标准错误流，由于这两者默认情况下都是屏幕，所以标准输出流和标准错误流会混在一起，因而需要对输出进行重定向。常见的做法可以是下面的一种或多种的组合：

1. 将标准输出流重定向到数据文件中
2. 将标准错误流重定向到日志文件中
3. 将标准输出流通过管道传递给下一个命令

重定向

这里只介绍 GMT 中经常使用的最简单的重定向功能。

对于标准输出流：

- `>`：将标准输出流重定向到新文件中。若该文件已存在，则覆盖文件中原内容；若该文件不存在，则创建该文件
- `>>`：将标准输出流追加到文件中。若文件已存在，则将标准输出流追加到已有文件后面；若文件不存在，则创建该文件

对于标准错误流，重定向符号是 `2>` 和 `2>>`。这里的 `2` 表示标准错误流，大于号的含义与标准输出流相同。

管道

除了上面提到的重定向符号之外，还有一种常用的类似重定向的操作，即管道，用符号 `|` 表示。

管道的作用是，将上一个命令的标准输出作为下一个命令的标准输入。

举例如下，假设文件 `input.dat` 中包含了一系列地震的经度、纬度和震级共三列数据，想要在图上画很多圆表示地震的位置，圆的大小表示震级的大小。可以用类似如下命令：

```
gmt psxy input.dat -Rxxx ... > test.ps
```

此时 `plot` 模块会读取 `input.dat` 文件的内容作为其输入。

也可以使用管道：

```
cat input.dat | gmt psxy ...
```

`cat` 命令会读取 `input.dat` 的内容并将其发送到标准输出流, 由于使用了管道, 标准输出流中的内容被 `gmt plot` 接收作为自己的标准输入流。

当然还可以使用 `gawk`

```
gawk '{print $1, $2, $3/10}' input.dat | gmt psxy ...
```

`gawk` 会读取 `input.dat` 的内容, 并对数据做简单处理并输出。

标准输入流

GMT 的某些模块需要数据才可以画图, 这些数据可以来自于文件, 或来自于标准输入流。

比如要绘制地震的分布, 可以把地震的经纬度信息放在文件 `event.loc` 中, 其内容如下:

```
100.0 40.0
110.0 45.0
```

将这些数据传给 GMT 有如下几种方法。

1. 直接在命令行指定文件名, 命令会自动读取该文件的内容:

```
gmt psxy event.loc -R70/140/20/60 -JM6i -B5 -Sc0.2c -Gred > map.ps
```

2. 直接从键盘输入

标准输入流的默认设备是键盘。下面的例子中直接从键盘输入 GMT 所需的数据。首先执行 `gmt plot` 命令, 然后键盘键入两行数据, 再按下 `Ctrl+C` 中断输入, GMT 会给出中断警告, 然后按回车键即可:

```
$ gmt psxy -R70/140/20/60 -JM6i -B5 -Sc0.2c -Gred > map.ps
100.0 40.0
110.0 45.0
Interrupt at /lib64/libc.so.6(__read+0x10)[0x7f8383e8d980]
Tuser: 0.004s Tsys: 0.004s VmRSS: 8340kB VmSize: 114268kB
Press return to continue, ctrl-c to quit.
$
```

3. 标准输入流重定向 <

< 的作用是读取 < 后的文件的内容并将其作为标准输入流, 与直接在命令行指定文件名类似:

```
gmt psxy -R70/140/20/60 -JM6i -B5 -Sc0.2c -Gred > map.ps < event.loc
```

4. 通过管道输入

管道可以将前一个命令的标准输出作为后一个命令的标准输入:

```
cat event.loc | gmt psxy -R70/140/20/60 -JM6i -B5 -Sc0.2c -Gred > map.ps
```

5. Here Documents

示例如下, 两个 EOF 之间的所有数据都会被传递给 GMT:

```
gmt psxy -R70/140/20/60 -JM6i -B5 -Sc0.2c -Gred > map.ps << EOF  
100.0 40.0  
110.0 45.0  
EOF
```

说明:

1. 上面列出的 5 种方式中, 常用的是第 1、4、5 种;
2. Here Documents 方法中, EOF 可以被替换成其他任意字符(比如 END), 只要保证开始和结束的符号一致即可
3. Here Documents 方法仅适用于 bash, 不适用于 bat

倒引号

倒引号, 也称为反引号, 英文为 backtick 或 backquote。键盘上按键 1 左边的键, 那个像顿号的就是。倒引号的作用是将一个命令的标准输出插在另一个命令的任意位置。

例如, 想要用 plot 绘制某数据时, 需要提供数据的范围 -R, 而 gmtinfo 模块可以用于计算并输出数据的范围, 即需要将 gmtinfo 的输出作为 plot 的一个选项。

比如:

```
$ gmt info in.dat -I1/1  
-R0/10/0/10  
$ gmt psxy in.dat -JX10c -R0/10/0/10 > map.ps
```

上面的做法需要人工干预, 不适合脚本自动化, 可以利用倒引号将 gmtinfo 的输出保存到变量中:

```
#!/bin/bash  
  
R=`gmt info in.dat -I1/1`  
gmt psxy in.dat -JX10c $R > map.ps
```

上面的例子还可以进一步简化。此处变量 \$R 只需要用一次, 因而没有必要把 gmtinfo 的输出信息保存到变量中, 可以直接在 psxy 命令中使用倒引号:

```
$ gmt psxy in.dat -JX10c `gmt info in.dat -I1/1` > map.ps
```

此处, bash 首先会执行倒引号内的命令, 然后用 gmtinfo 的输出替换整个倒引号部分, 再执行替换后的命令。这样的写法更易于自动化。

通配符

UNIX 下提供了通配符功能，使得可以基于文件名的模式选择一组文件。

UNIX 下的通配符包括：

表 15.1：通配符

通配符	含义
*	匹配任意数目的任意字符
?	匹配任意单个字符
[ABC]	匹配中括号内的任意单个字符
[A-Z]	匹配给定范围内的任意单个字符

示例：

1. `data_*.d` 会匹配所有以 `data_` 开头，并以 `.d` 结尾的文件
2. `line_?.d` 会匹配所有以 `line_` 开头，后接任意一个字符，并以 `.d` 结尾的文件
3. `section_1[0-9]0.part` 会匹配 `section_1x0.part` 中 `x` 为 0 到 9 的文件
4. `section_[12].part` 会匹配 `section_1.part` 和 `section_2.par` 两个文件

15.2.2 bash 简明教程

bash 是什么

bash 是类 UNIX 系统下最常见的 shell。bash 是一个程序也是一种编程语言。

bash 的创建

新建一个文本文件，后缀名任意，通常用 `.sh`。

bash 脚本可以直接用 `bash script.sh` 的形式执行，也可以通过如下命令给脚本执行权限，然后使用 `./script.sh` 的方式执行：

```
chmod +x ./script.sh
# 赋予脚本文件可执行权限
./script.sh
```

常用命令

echo 命令

`echo`，顾名思义，就是回显的意思，在屏幕上输出 `hello world`：

```
echo hello world
```

重定向符号

常见的重定向符号有 > 和 >>。

默认情况下, 命令执行结果是输出到终端窗口。使用重定向后, 将改成输出到另一个位置(文件)。> 和 >> 的区别在于 > 会先将文件内容清空然后再将命令执行结果输入, 而 >> 会将命令执行结果直接追加到文件末尾:

```
echo 第一行 > newfile.txt
echo 第二行 >> newfile.txt
```

管道符号

管道符号 | 会将其左侧命令的输出结果作为输入传递给其右侧语句:

```
echo 5 5 | gmt psxy -R0/10/0/10 -JX10c -Ba -Sa0.5c -Gred > map.ps
```

删除文件

rm 命令用来删除文件。使用 GMT 命令绘图后通常会在文件夹下产生一些临时文件, 这时可以使用 rm 命令进行删除:

```
echo 1 2 > tmp1.txt
echo 3 4 >> tmp1.txt
echo 5 6 > tmp2.txt
gmt psxy tmp1.txt -R0/10/0/10 -JX10c -Ba -Sa1c -Gred -K > map.ps
gmt plot tmp2.txt -R -J -Sc0.5c -Gyellow -O >> map.ps
rm tmp* gmt.*
```

定义变量

bash 定义变量不需要申明变量的类型:

```
VAR="some"
# = 号的前后不能加空格
echo $VAR
# 在使用变量时, 必须加上美元符号
```

添加注释

在行首添加 #, 即可注释该行。

循环

多种循环结构:

```

while cmd1; do
    cmd2;
done

until cmd1; do
    cmd2;
done

for name in words; do
    cmd;
done

for ((exp1; exp2; exp3)); do
    cmd;
done

case word in
    pat1) cmd1;;
    pat2) cmd2;;
    pat3) cmd3;;
esac

```

一个实际的例子:

```

a="1 2 3 4 5 a b c"
for i in $a; do
    echo $i
done
a=(1 2 3 4 5 a b c)
for ((i=0;i<10;++i)); do
    echo $i
    echo ${a[i]}
done

```

15.2.3 bat 简明教程

Bat 是什么

Bat 是 Windows/DOS 系统下的批处理文件格式，由一系列 DOS 命令语句组成。Windows 下 GMT 语句要写成 bat 格式执行。

Bat 的创建

新建一个文本文件，将后缀改为 .bat 即可。可以由一般的文本编辑器（如记事本）编辑，然后双击直接运行。

常用命令

echo 命令

echo 其实是一个开关命令，执行 echo off 将关闭回显，它后面所有命令都不显示命令本身，只显示执行后的结果。如果在某一行执行 echo on 命令将重新打开回显。

echo 常见的用法是在屏幕上显示信息，例如 echo hello world 命令将在屏幕上显示 hello world。

重定向符号

常见的重定向符号有 > 和 >>。默认情况下，命令执行结果是输出到 DOS 窗口，使用重定向后，将改成输出到另一个位置（文件）。> 和 >> 的区别在于 > 会先将文件内容清空然后再将命令执行结果输入，而 >> 会将命令执行结果直接追加到文件末尾：

```
echo 第一行 > newfile.txt  
echo 第二行 >> newfile.txt
```

管道符号

管道符号 | 会将其左侧命令的输出结果作为输入传递给其右侧语句：

```
echo 5 5 | gmt psxy -R0/10/0/10 -JX10c -Ba -Sa0.5c -Gred > map.ps
```

定义变量

可以使用 set 命令定义变量，然后用 % 变量名称% 来引用变量：

```
set Year=2018  
set Month=12  
set PS=%Year%-%Month%.ps  
echo 文件名为 %PS%
```

这段代码运行的结果为：文件名为 2018-12.ps。

暂停执行

pause 用来暂停命令的执行，以便查看终端输出的过程。

一般情况下双击执行 bat 脚本 cmd 窗口会一闪而过，为了查看中间是否出错，在 bat 文件最后一行（或需要暂停的地方）写一个 pause。

删除文件

del 命令用来删除文件。使用 GMT 命令绘图后通常会在文件夹下产生一些临时文件，这时可以使用 del 命令进行删除：

```

echo 1 2 > tmp1.txt
echo 3 4 >> tmp1.txt
echo 5 6 > tmp2.txt
gmt psxy tmp1.txt -R0/10/0/10 -JX10c -Ba -Sa1c -Gred -K > map.ps
gmt psxy tmp2.txt -R -J -Sc0.5c -Gyellow -O >> map.ps
del tmp* gmt.*

```

注释语句

rem 命令用来添加注释:

```

rem 绘制地图
echo 2 2 | gmt psxy -R0/5/0/5 -JX5c -B1 -Sc0.5c -Gblack -K > map.ps
echo 3 3 | gmt psxy -R -J -Sc0.5c -Gred -O >> map.ps

```

循环语句

bat 文件中 for 命令用法较为复杂, 这里只介绍如何利用 for 命令批量绘图。

命令格式: `for %%i in (command1) do (command2)`

表示将 comand1 中的每一个结果赋值给 i, 代入到 command2 中执行命令, 在 command2 中依然用 %%i 表示这个值:

```

rem 开启变量延迟
setlocal enabledelayedexpansion
for /f %%i in ('dir /s/b "*.DAT"') do (
set file=%%i
rem 去除文件后缀名, 即去除文件最后 4 个字符
set file=!file:~0,-4!
gmt surface %%i -R73/135/17/54 -I5m -Gtmp.grd
gmt grdimage tmp.grd -R73/135/17/54 -JM12c -Cmycpt.cpt -Ba !file!.pdf
del gmt.*
)

```

上段表示将当前目录下所有 DAT 文件都进行插值并绘图, 生成不同的 pdf 文件。由于在循环中, 又将 i 赋值给新的变量 file, 为了感知每次循环中 i 值的变化, 需要在 bat 脚本开头设置 `setlocal enabledelayedexpansion`, 并且 file 变量的引用用 `!file!` 表示。

15.2.4 Unix 常用工具

awk

<http://www.ruanyifeng.com/blog/2018/11/awk.html>

15.3 GMT 颜色表

GMT 支持通过颜色名指定 663 种颜色。本表列出了 GMT 支持的所有颜色名及其对应的 RGB 值。

下载: A4 PNG 格式 | A4 PDF 格式 | USLetter PNG 格式 | USLetter PDF 格式

15.4 GMT 预定义填充图案

下图列出了 GMT 中预定义的 90 种位图图案 (右键查看大图并放大才能看到差异)，所有图案都是使用默认的黑白色在 `<dpi>` 取 300 的环境下生成的。每张小图中左半部分使用 `-Gp` 绘制，右半部分使用 `-GP` 绘制。

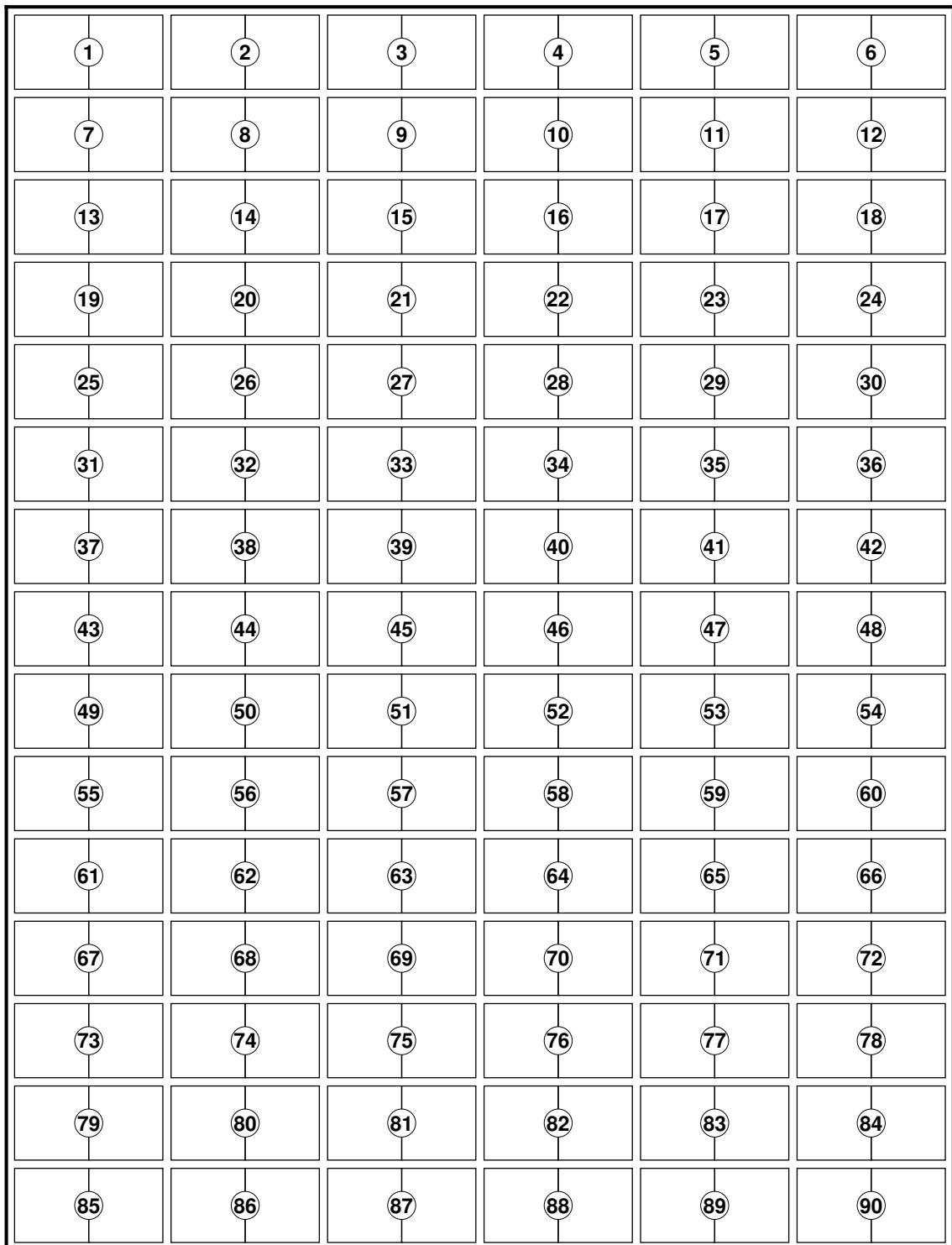


图 15.1: GMT 内置位图图案

下载: PDF 格式 | PNG 格式

15.5 GMT 底层原理

15.5.1 命令行历史

在使用多个命令绘制一张图时, 这些命令可能使用了完全相同的标准选项参数, 如果对于每个命令都需要把标准选项的参数写一遍, 则太麻烦了, 在这种情况下, GMT 允许你省略标准选项的参数。例如:

```
gmt basemap -JX10c/10c -R0/10/0/10 -B1 -K > text.ps  
gmt plot station.dat -J -R -O >> text.ps
```

GMT 每个命令在执行的时候, 会将当前命令所使用的基本选项的参数保存到命令行历史文件 `gmt.history` 中, 比如上面第一个命令指定了 `-JX10c/10c -R0/10/0/10`, 会生成如下内容:

```
# GMT 5 Session common arguments shelf  
BEGIN GMT 5.1.2  
B 1  
J X  
JX X10c/10c  
R 0/10/0/10  
L 1  
END
```

在执行第二个命令时使用了 `-J -R`, 省略了具体的参数值, 此时命令会读取 `gmt.history` 文件中需要的参数值。

抛开具体的细节不看, 可以认为后面的命令继承了前面命令的基本选项参数, 如果之后的命令显式指定了基本选项的参数, 则后面命令中的参数会覆盖 `gmt.history` 中保存的值。

需要注意, 如果在一个命令中通过管道或其他手段调用了多个 GMT 模块, GMT 无法保证 `gmt.history` 文件是按照从左到右的顺序处理的。

重要: 命令历史这一特性看起来非常方便, 但不建议依赖这一特性! 建议将常用的基本选项的参数定义成变量使用。

15.6 目录结构

安装好 GMT 之后, 可以看看 GMT 安装路径下的目录结构, 看看 GMT 究竟安装了哪些东西。

本文用 `$GMTHOME` 表示 GMT 的安装路径, 在 Linux 下一般是 `/usr/local/GMT5` 或 `/opt/GMT5`, 在 Windows 下一般是 `C:\programs\gmt5`。

`$GMTHOME` 中包含了 GMT 的全部文件。下面列出了 `$GMTHOME` 的目录树，并对其中重要的文件及目录做简单解释。以下内容以 Linux 系统为准，Windows 系统有微小差异：

```
.
|-- bin
|-- include
|-- lib64
`-- share
```

15.6.1 bin 目录

`bin` 目录下包含如下文件：

```
bin
|-- gmt                      # GMT 主程序
|-- isogmt                    # GMT 主程序(隔离模式)
|-- gmt-config                # GMT 安装信息获取工具
|-- gmt_shell_functions.sh    # bash 辅助脚本(仅使用于 Linux/macOS)
|-- gmtswitch                 # GMT 版本切换工具
|-- gmt.m                     # GMT/MATLAB 工具包(仅限于 Windows)
`-- *.dll                     # 动态库文件(仅限于 Windows)
```

几点注意事项：

1. `bin` 目录下除了上面提到的文件之外，可能还有很多类似 `plot`、`grdimage` 这样的文件，其本质上是指向二进制文件 `gmt` 的软链接。这些软链接的目的是为了部分兼容 GMT4 的语法，所以这些软链接都可以手动删除；
2. `bin` 目录下需要使用的只有 `gmt`，即 `gmt` 可以认为是 GMT5 中唯一的一个命令。

15.6.2 include 目录

`include` 目录下有 `gmt` 子目录，`gmt` 子目录中包含了 GMT 的头文件：

```
include
`-- gmt
```

仅当自己写程序调用 GMT 函数库时才需要使用这些头文件，因而一般用户不需要关注。

15.6.3 lib 目录

32 位系统下目录名为 `lib`，64 位系统下目录名为 `lib64` 或 `lib`，以下统称为 `lib` 目录。`lib` 目录下包含了 GMT 的动态库文件：

```
lib
|-- gmt
|   '-- plugins
|       '-- supplements.so      # GMT 扩展函数库
|-- libgmt.so                 # GMT 主函数库
`-- libpostscriptlight.so     # PSL 绘图函数库
```

仅当自己写代码调用 GMT 函数库时才需要使用, 因而一般用户不需要关注。

15.6.4 share 目录

share 目录中包含了 GMT 运行所需的众多辅助文件:

```
share
|-- VERSION                      # GMT 版本号
|-- coast                         # 含海岸线数据
|-- cpt                           # 含全局 CPT 文件
|-- custom                         # 包含 GMT 自定义符号的 def 文件
|   '-- gmt_custom_symbols.conf    # 用户自定义符号的配置文件
|-- dbase                          # 含 grdraster 命令所使用的数据库(默认无数据)
|   '-- grdraster.info            # grdraster 命令的配置文件
|-- dcw                            # 含国界、省界数据
|-- doc                            # 官方文档
|   '-- examples                  # 官方示例
|   '-- html                      # 官方文档网页版
|   '-- pdf                       # 官方文档 PDF 版
|   '-- supplements               # 补充包的说明文档
|   '-- tutorial                  # GMT 教程中所使用的数据
|-- localization                   # 不同语言的时间定义
|-- man                            # man 格式文档
|-- postscriptlight                # PS 所需要的文件
|   '-- PSL_custom_media.conf     # 用户自定义的纸张大小
|   '-- PSL_custom_fonts.txt      # 用户自定义的 PS 字体
|-- spotter                        # 板块运动相关数据
|-- tools                          # GMT 辅助工具
|-- mgd77                          # mgd77 辅助文件
|-- mgg                            # mgg 辅助文件
`-- x2sys                          # x2sys 辅助文件
```

15.7 三种距离计算方式

在计算地球或其它星体上任意两点间的距离时, GMT 提供了三种不同的计算方式。这三种方式在精度和效率上各有权衡, 用户可以根据自己的需求选择适合的距离计算方式。

15.7.1 Flat Earth 距离

地球上任意两点 A 和 B 的 Flat Earth 距离计算公式:

$$d_f = R \sqrt{(\theta_A - \theta_B)^2 + \cos\left[\frac{\theta_A + \theta_B}{2}\right] \Delta\lambda^2}$$

其中 R 是地球平均半径(由参数 `PROJ_MEAN_RADIUS` 控制), θ 是纬度, $\Delta\lambda = \lambda_A - \lambda_B$ 是经度差。式中地理坐标的单位均是弧度, 且需要考虑到跨越经度的周期性问题。

该方法的特点是计算速度快但精度不高, 适用于纬度相差不大且对计算效率要求比较高的情况。

可以通过在距离量前加上前缀 `-` 指定使用该方法计算两点间的距离。对于某些只需要指定距离单位而不需要指定具体距离值的情况下, 则可以在距离单位前加上前缀 `-` 表示该

距离用 Flat Earth 方法计算。比如, `-S-50M` 表示设定搜索半径为 50 海里, 其中距离用 Flat Earth 方法计算。

15.7.2 大圆路径距离

该方法将地球近似为一个半径为 R 的球, 地球上任意两点 A 和 B 的大圆路径距离可以用 Haversine 公式 计算:

$$d_g = 2R \sin^{-1} \sqrt{\sin^2 \frac{\theta_A - \theta_B}{2} + \cos \theta_A \cos \theta_B \sin^2 \frac{\lambda_A - \lambda_B}{2}}$$

该方法是 GMT 默认使用的距离计算方法, 适用于大多数情况。比如, `-S5000f` 表示距离为 5000 英尺。

有两个 GMT 参数可以控制大圆路径距离的计算细节, 分别是:

- `PROJ_MEAN_RADIUS` 地球平均半径
- `PROJ_AUX_LATITUDE` 指定将大地纬度转换为多个适合球状近似的辅助纬度中的其中一个

需要注意, 这两个选项仅当 `PROJ_ELLIPSOID` 不为 `sphere` 时才有效。

15.7.3 测地距离

地球上两点间的精确距离可以用 Vincenty (1975) 的完全椭球公式计算。该方法计算得到的距离精度最高精确到 0.5 毫米, 同时也是计算速度的最慢的方式。

可以通过在距离或距离单位前加上前缀 `+` 来指定用该方法计算距离。比如, `-S+20k` 表示用该方法计算的 20 千米的距离。

除了 Vincenty 完全椭球公式外, 还可以将参数 `PROJ_GEOIDESIC` 设置成 Rudoe (GMT4 所使用的计算公式) 或 Andoyer (近似公式, 精确到 10 米量级) 以使用不同的计算公式。

15.7.4 总结

GMT 在计算距离时有三种算法: FLAT Earth 距离、大圆路径距离和测地距离。三种方法的计算精度由低到高, 计算速度由高到低。

对于一个距离量, 比如 `40k`, 默认使用大圆弧距离, 可以使用 `-40k` 表示 FLAT Earth 距离, `+40k` 表示测地距离。

15.8 环境变量

15.8.1 \$GMT_SHAREDIR

指定 GMT 的 share 目录的位置。若该变量为空, 则 GMT 会使用默认值 `$(GMTHOME)/share`。

15.8.2 \$GMT_DATADIR

用于指定用户自己的数据文件的放置目录。当在命令中给定文件名时, GMT 首先会在当前目录寻找该文件。若找不到, 则会到环境变量 \$GMT_DATADIR 所指定的目录中寻找。

对于一些常用的数据文件, 可以放在特定的目录中, 并将任意一个环境变量指向该目录, 则在命令中使用该文件时只需要指定文件名而不必给出完整路径。

Linux 下多个目录之间用冒号分隔; Windows 下用分号分隔。

Linux 下以 / 结尾的目录会被递归搜索。(此句存疑)。

15.8.3 \$GMT_USERDIR

用于指定用户自定义的配置文件的放置目录。比如 gmt.conf 、自定义的符号、CPT、数学宏、网格文件自定义后缀 gmt.io 等。

若 \$GMT_USERDIR 未定义, 则使用默认值 \${HOME}/.gmt 。

15.8.4 \$GMT_TMPDIR

GMT 写临时文件 gmt.history 和 gmt.conf 的路径。若未指定, 则默认写到当前目录。

15.8.5 \$GMT_CACHEDIR

用户缓存 GMT 从其服务器上下载的数据。若未指定, 则默认下载到 \${HOME}/.gmt/cache 下。可以使用 gmt clear cache 清空缓存目录。

15.9 自定义字体

GMT 默认支持了 35 种字体, 如果想要使用额外的字体, 则需要自己配置。可以修改 \$GMTHOME/share/postscriptlight/PSL_custom_fonts.txt 文件或在当前目录下新建配置文件 PSL_custom_fonts.txt 。

文件中每行包含三列数据, 分别是: PS 字体名、字体的高度(单位为 p)以及布尔值 0 或 1 告诉 GMT 是否要对文字重新编码。例如:

```
LinBiolinum0    0.700    0
LinLibertineOB   0.700    0
```

使用 gmt pstext -L 可查看所有字体对应的编号, 在 GMT 中即可直接使用这些编号指定字体。

实际使用的时候有几个问题需要注意:

1. 第一列是 PostScript Name, 与字体文件的文件名不同, 但尚不清楚如何提取某个字体文件的 PS 文件名
2. 是否要对文字进行编码? 取 0 还是 1, 这个尚不确定

3. 调用该字体生成 PS 文件后，直接用 `gs` 打开会看不到效果，需要给 `gs` 加上 `-sFONTPATH=dirname` 选项指定字体文件名，或指定系统环境变量 `$GS_FONTPATH`
4. 在用 `psconvert` 将 PS 文件转换成其他格式时，需要使用 `-C-sFONTPATH=dirname` 将自定义字体的路径传递给 `gs`，使得字体可以被嵌入到新格式中

15.10 获取 GMT 开发版

从 GMT 官方网站下载页面下载得到的 GMT 源码是发布版源码，即面向用户的源码。而 GMT 维护者在开发时所使用的源码是开发版源码。

GMT 开发版源码托管在 GitHub 上，其网址为：<https://github.com/GenericMappingTools/gmt>

下载源码：

```
git clone https://github.com/GenericMappingTools/gmt.git
```

更新源码：

```
cd gmt
git pull origin master
```

编译源码的方式与正常安装 GMT 的方式相同。

15.11 用 GMT 制作动画

GMT 只能生成 PS 格式的图片，所以 GMT 自身并不具有制作动画的功能。但动画本质上就是将一系列独立的图片按照顺序快速播放，所以可以利用 GMT 绘制多张独立的图片，并将这些图片作为动画的每一帧。

在制作一个动画时，通常需要按照顺序考虑如下问题：

1. 一个动画需要多少帧？太少的帧数会导致动画存在不连续感，太多的帧数则可能需要绘制/播放很长时间。建议定义相关变量，使得可以根据实际情况进行微调
2. 通常一张动图中有很多元素是不变的。比如海岸线底图、一些图例信息等等。这些不变的元素，没有必要为每一帧都重复绘制。可以直接将这些不变的元素绘制完，然后将未关闭的文件复制到新文件中，并在新文件中绘制余下的部分
3. 帧计数器应初始化为 0，并随着循环不断递增
4. 根据帧计数器中的值构建文件名，文件名必须有规律且唯一。例如 `plot_0001.ps` 就是不错的命名方式（0001 中的前置 0 不可省略，因为在将一系列图片转换成动画时需要涉及到各帧之间的先后顺序问题）
5. 将 PS 文件转换成图片，建议使用 GMT 自带的 `psconvert` 模块
6. 使用 ImageMagick 的 `convert` 命令将图片转换成 gif 或者 avi 等格式

7. 删除临时文件

以下是制作动画的整个流程的示意代码：

```

#!/bin/bash
PS=plot.ps

# 绘制图中的静态元素
gmt cmd1 ... -K > $PS
gmt cmd2 ... -K -O >> $PS

# 开始循环, i 为帧计数器, MAX 为最大帧数
for ((i=0; i<MAX; i++))
do
    # 构建新文件名 plot_xxxx.ps
    file = `echo $i | awk '{printf "plot_%04d.ps", $1}'`'
    # 将已经绘制好静态元素的文件复制到新文件
    # 此时 file 中已经包含了静态元素
    cp $PS $file
    # 向 file 中添加动态元素
    gmt cmd3 ... -K -O >> $file
    gmt cmd4 ... -K -O >> $file
    gmt cmd5 ... -O >> $file
    # 将 PS 文件转换为某种图片格式
    gmt psconvert -A -P $file
done

# 利用 convert 命令将图片转换为 GIF 动图
convert *.jpg plot.gif

# 清理临时文件
rm *.ps *.jpg

```

15.12 目录参数

有一些环境变量以及 GMT 配置参数，可以用于指定某些特殊用途的目录。这些环境变量和配置参数包括：

- `$GMT_SHAREDIR` GMT 的 share 目录所在位置，通常不用设置，GMT 会自动猜测其所在位置
- `$GMT_DATADIR` 或 `DIR_DATA` 可以指向一个或多个目录，用于放置用户常用的数据文件。目录之间用冒号分隔（Windows 下用分号分隔）。任何以斜杠 / 结尾的目录会被递归搜索（Windows 下不会）。若二者同时有值，以 `DIR_DATA` 的值优先
- `$GMT_CACHEDIR` 或 `DIR_CACHE` 用于放置 GMT 模块从 GMT 服务器上下载的临时数据
- `$GMT_USERDIR` 用户放置自定义配置文件的地方，比如用户自定义的 `gmt.conf` 文件、自定义符号、CPT 文件、数学宏、网格文件后缀文件等。若该变量未定义，则默认值为 `$HOME/.gmt`。
- `$GMT_TMPDIR` 临时文件（比如 `gmt.history` 和 `gmt.conf`）放置的目录。若未设置，则默认为当前目录

- *DIR_DCW* DCW 数据放置的目录
- *DIR_GSHHG* 海岸线数据放置的目录

当命令行中有文件需要读入时, GMT 不仅仅会在当前目录下寻找文件, 还会到这些特殊变量中寻找。GMT 会依次到下列目录中寻找文件:

1. 当前目录
2. GMT 参数 *DIR_DATA* 所定义的目录
3. GMT 参数 *DIR_CACHE* 所定义的目录
4. 环境变量 *\$GMT_USERDIR* 所定义的目录
5. 环境变量 *\$GMT_CACHEDIR* 所定义的目录
6. 环境变量 *\$GMT_DATADIR* 所定义的目录

15.13 等值线标注和“线条标注”

GMT 中可以使用采用 `grdcontour` 和 `pscontour` 模块来绘制等值线, 每条等值线都可以附加一个标注。GMT 中 `psxy` 和 `psxyz` 模块也可以使用 `-Sq` 选项绘制带有标注的线段。

在需要为等值线/线段附加标注的时候, 如何优化标注的位置是一个很困难的主题。GMT 提供了不同的算法确定标注的位置, 并且可以自由地指定标注的属性。本章总结了标注的属性和位置确定方法, 并给出了一些应用实例。

15.13.1 标注的位置

GMT 中提供了 5 种算法来自动确定标注的位置。对于 `grdcontour` 和 `pscontour` 模块, 可以通过 `-G` 选项指定使用哪种标注定位算法, 对于 `psxy` 和 `psxyz` 模块, 则可以通过 `-Sq` 选项指定使用哪种标注定位算法。

不管是 `-G` 还是 `-Sq`, 其所需要的信息是完全相同的, 采用 `<code><info>` 的格式来指定算法和相应参数, 其中 `<code>` 取不同的值代表不同的算法, `<info>` 则是各个算法所对应的参数。

下述内容给出了不同 `<code>` 对应的算法和相应的参数:

d: 完整的语法为 `d<dist>[c|i|p] [/frac]`

根据地图上的投影距离确定标注的位置, 可以指定长度单位或采用默认值 `[PROJ_LENGTH_UNIT]`。从等值线的起始位置开始, 以 `<dist>` 为步长, 沿着等值线布置标注。为了保证可以标注总长度小于 `<dist>` 的封闭曲线, 可以指定 `<frac>` 参数, 将第一个标注放置在距封闭曲线起点 $d = <dist> * <frac>$ 的位置上, `frac` 的默认值为 0.25。

D: 完整的语法为 `D<dist>[d|m|s|e|f|k|M|n] [/<frac>]`

与 `d` 相似, 但是其输入数据必须是地理坐标 (同时必须选择地图投影), 距离为沿等值线的地表真实距离。可以附加距离的单位, 其中 `d|m|s|e|f|k|M|n` 各个单位的

含义见[单位](#)一节。其它参数的意义与 d 相同。

f: 完整的语法为 `f<fix.txt>[/<slop>[c|i|p]]`

其中，ASCII 文件 `<fix.txt>` 中每条记录的前两列（坐标）指定了标注的位置。当文件中的坐标与等值线的距离小于 `<slop>`（附加单位或使用默认值 `PROJ_LENGTH_UNIT`）时，才会显示标注。`<slope>` 的默认值为 0，即文件中的坐标必须与线段上的坐标完全匹配。

l: 完整的语法为 `l<line1>[,<line2>[,...]]`

指定一个或多个以逗号分隔的直线段，在这些直线段与等值线的交点位置放置标注。通过起点 `<start>` 和终点 `<stop>` 的坐标来定义每个直线段 `<line>`。起点和终点的坐标可以是常规坐标，如斜杠分隔的经纬度，或与地图区域相关的 2 个字母组合成的子选项。这些字母的取值与 `pstext` 中对齐方式的取值相同，即 `[L|C|R][B|M|T]`

第一个字母代表横坐标 `<x>`，第二个字母代表纵坐标 `<y>`，如 `LB` 代表地图的左下角。在模块`grdcontour` 中还可以使用子选项 `Z+`、`Z-` 代表网格数据中全局最大值点或最小值点的坐标。例如，直线段 `line LT/RB` 代表地图左上角到右下角的对角线，`Z-/135W/15S` 代表网格数据中最小值点与 $(135^{\circ}\text{W}, 15^{\circ}\text{S})$ 之间的直线段。

L: 除起点与终点之间的线段为大圆弧外，其余内容与子选项 `l` 相同。

n: 完整的语法为 `n<number>[/<minlength>[c|i|p]]`

沿等值线放置 `<number>` 个标注，即将等值线分割为 `<number>` 段，标注位于每段的中心位置。还可以通过指定最小距离 `<minlength>` 来保证相邻标注之间的距离不小于 `<minlength>`。

N: 完整的语法为 `N<number>[/<minlength>[c|i|p]]`

除标注位于每段终点位置外 (`<number> >= 2`)，与子选项 `n` 类似。当 `number=-1` 时，标注位于等值线的起点；当 `number=+1` 时，标注位于等值线的终点。

x: 完整的语法为 `x<cross.d>`

ASCII 文件 `<cross.d>` 内给出了多段数据，这些线段与等值线的交点即是标注的位置。

X: 除了 ASCII 文件中定义的线段为大圆弧外，与子选项 `x` 的类似。

每调用一次等值线绘制模块，只能指定一种确定标注位置的算法。

15.13.2 标注的属性

确定标注的位置之后，还需要指定标注的属性。对于等值线绘制模块，在 `-A` 选项后以 `+<code>[<参数>]` 的格式定义不同的属性；对于线条绘制模块中，则是在 `-Sq` 选项后用冒号 `:` 来分隔标注的属性和标注的位置。

部分属性只能用于线条绘制模块，因此，首先列出了两个模块通用的属性。这些属性包括：

+a: 控制标注的角度和线条的角度间的相互关系：

1. 后面加上 **n** 表示二者相互垂直；
2. 后面加上 **p** 表示二者之间相互平行，调用 `grdcontour` 模块时，还可以附加 **u** 或 **d** 表示标注的上边缘指向更高或更低的等值线；
3. 给定角度 **<angle>** 表示自水平方向开始逆时针方向旋转的角度

+c: 每个标注周围存在一个假想的文本框，等值线在这个区域内是不可见的。默认的文本框精确的围限了标注，可以指定水平向和竖直向的间隙（相对于标注的基线）。若水平向和竖直向的间隙值不同，需要以斜杠分隔，可以在间隙值后附加长度单位 (**c|i|m|p**)，也可以指定间隙与标注所采用字体的百分比，默认值为 15%。

+d: Debug 模式。标注所在位置也会绘制等值线，用来测试等值线的位置。

+d: 延迟模式，延迟标注文字的绘制。

+f: 指定标注文字的字体、大小和颜色等，可参考 `pstext`。字体的默认值参见 *FONT_ANNOT_PRIMARY*。

+g: 指定文本框的填充效果，颜色的默认值与 *PS_PAGE_COLOR* 相同。

+j: 指定标注内容与标注位置之间的对齐方式，默认值为 **CM**，指定值可以覆盖默认值，参数值由 2 个字母组成，取值范围分别为 **[L|C|R][B|M|T]**。对于弯曲的标注文字 (**+v**)，只有竖直向对齐方式起作用。

+o: 指定文本框的形状为圆角矩形，只有对文本框进行填充或显示轮廓时才起作用。对于弯曲的标注文字 (**+v**) 不起作用。

+p: 指定文本框轮廓线的线条属性，默认值为 **[0.25p,black]**。

+r: 当曲率半径低于给定值时，不放置标注，可以指定曲率半径的单位，默认值为 0。

+u: 在标注后加单位 **<unit>**。通常在单位和标注之间有一个空格，若想去掉这个间隔，需要在单位前加连字符 (-)。调用 `grdcontour` 模块时，若给出这个属性，却未指定单位时，则使用网格头段中 *z* 值的单位。

+v: 根据线条摆动情况放置弯曲的标注，当标注长度较大时，该属性尤其有用。默认值为给定角度的不可见的直线段。

+w: 标注所在位置处等值线的角度，是对附近的 *width* 个点，进行最小二乘拟合计算的，*width* 的默认值为 10。

+=: 与 **+u** 非常相似，用于指定 *prefix* 的单位。

对于等值线绘制模块，标注的内容为等值线的数值（可以通过 **+u** 或 **+=** 属性来修改）。对于线条绘制模块来说，还可以指定下述属性：

+l: 在标注位置放置相同的内容，如果标注内容包含空格，则需要用引号括起来。

+L: 通过附加子选项指定标注的内容，可用的子选项包括：

+Lh: 采用多段数据的头记录作为标注内容（假设输入为多段数据，如果不是多段数据，则采用文件头记录）。首先扫描 **-L<子选项>** 属性，若没有指定该选项，则采用数据段头记录首字符（默认为 >）后的第一个单词。

+Ld: 采用笛卡尔坐标系内的距离作为标注内容的距离单位，可以指定单位，如 **c|i|p**，默认值为 **[PROJ_LENGTH_UNIT]**。标注内容的格式参见:ref:*FORMAT_FLOAT_OUT* <*FORMAT_FLOAT_OUT*>。

+LD: 采用真实地表距离计算标注内容，可以指定单位，如

d|e|f|k|m|M|n|s, 默认值为弧度 **d**。

+Lf: 采用 ASCII 文件 *fix.txt* 中第 2 列数据之后的所有文字作为标注的内容, 显然, 该属性需要在指定标注位置算法 (**f**) 的前提下, 才能起作用。

+Ln: 采用多段数据中当前数据段的顺序号作为标注内容。

+LN: 采用斜杠分隔的文件号-当前数据段顺序号作为标注内容。

+Lx: 与属性 **h** 类似, 多段数据头记录的来源为 *cross.d* 文件。显然, 该属性需要在指定标注位置算法 (**x|X**) 的前提下, 才能起作用。

15.13.3 等值线标注位置实例

本节通过一些简单的实例说明等值线标注位置选项的作用。首先, 在实例 1 中, 采用部分全球大地水准面数据 (geoid), 绘制了等值线。所选择的区域包含了大地水准面的两级, Indian Low 和 New Guinea High。

等距离放置标注

第 1 个实例使用标注位置算法的默认值, 沿等值线每 1.5 英寸放置一个标注:

```
#!/bin/bash
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno1.ps
gmt grdcontour geoid.nc -J -O -B20f10 -BWSne -C10 -A20+f8p -Gd1.5i -S10 -T+llH >> GMT_
contour-anno1.ps
rm gmt.*
```

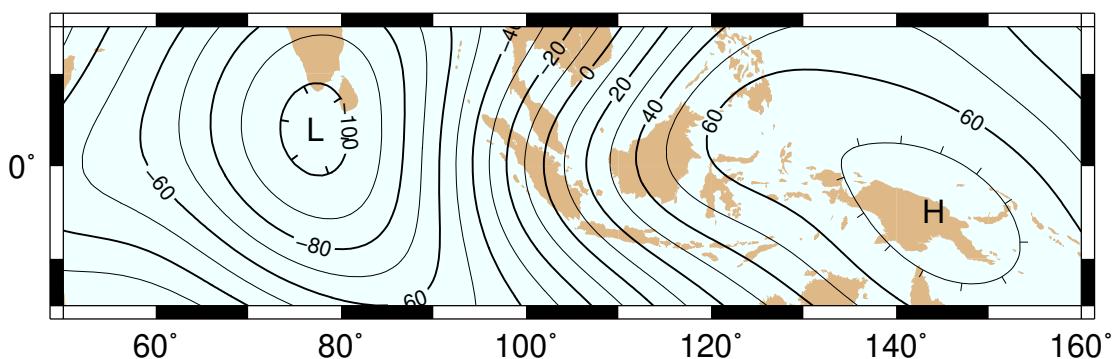


图 15.2: 通过指定 **-Gd** 选项的参数, 确定了标注的位置 (等值线上相距 1.5 英寸的点)

给定标注个数

现在指定每条等值线上标注的个数。每条等值线上只放置 1 个标注, 并且要求等值线的长度不小于 1 英寸,

```
#!/bin/bash
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno2.ps
gmt grdcontour geoid.nc -J -O -B20f10 -BWSne -C10 -A20+f8p -Gn1/1i -S10 -T+llH >> GMT_
contour-anno2.ps
rm gmt.*
```

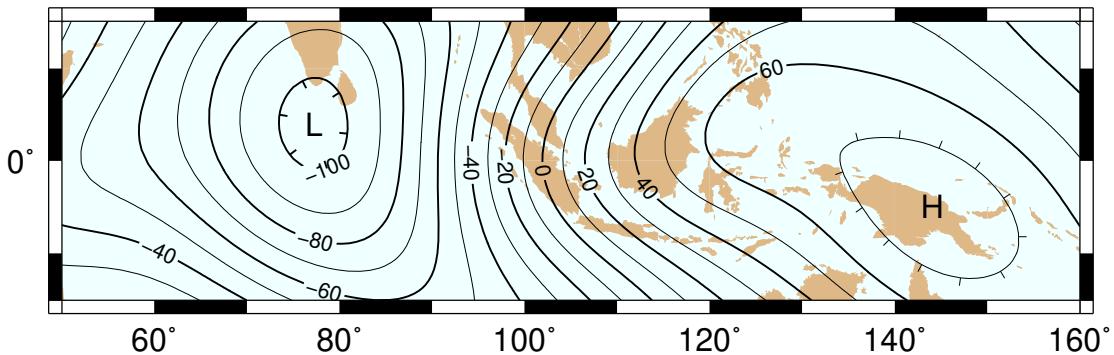


图 15.3: 通过指定 **-Gn** 选项的参数, 确定了标注的位置 (每条长度超过 1 英寸的等值线的中心位置)

给定标注位置

给定标注所在位置的坐标, 由于坐标不是严格位于等值线上, 指定了非 0 距离值, 即标注位置与等值线距离的上限。

根据等值线的几何形状, 自动计算标注的角度。为了帮助理解, 通过指定选项 **-A** 中的 **+d*** 属性, 采用了 debug 模式, 即在每个给定位置上绘制了一个小圆圈。

```
#!/bin/bash
cat << EOF > fix.txt
80      -8.5
55      -7.5
102     0
130     10.5
EOF
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno3.ps
gmt grdcontour geoid.nc -J -O -B20f10 -BWSe -C10 -A20+d+f8p -Gffix.txt/0.1i -S10 -T+llH >> GMT_contour-anno3.ps
rm gmt.*
```

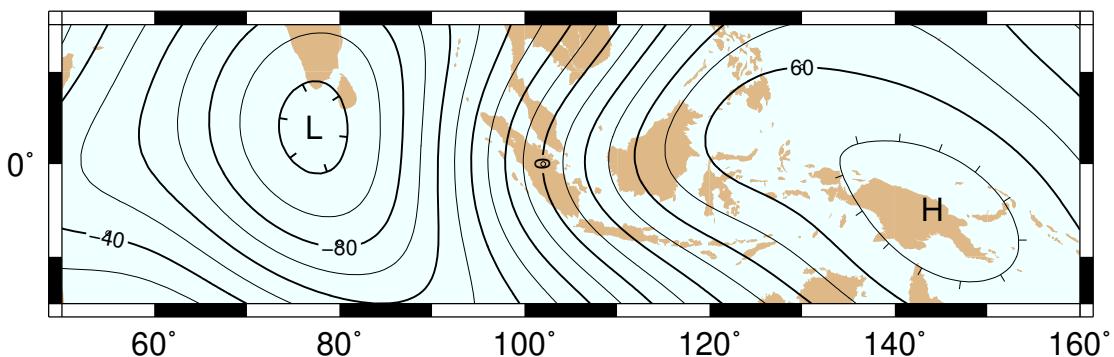


图 15.4: 通过指定 **-Gf** 选项的参数, 确定了标注的位置 (等值线上与给定点距离最小的点)

线段与等值线交点处放置标注

通过指定 **-Gl** 或 **-GL** 选项的参数来定义线段, 将标注放置在线段与等值线的交点。

```
#!/bin/bash
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno4.ps
gmt grdcontour geoid.nc -J -O -B20f10 -BWSne -C10 -A20+d+f8p -GLZ-/Z+ -S10 -T+llH >> GMT_contour-anno4.ps
rm gmt.*
```

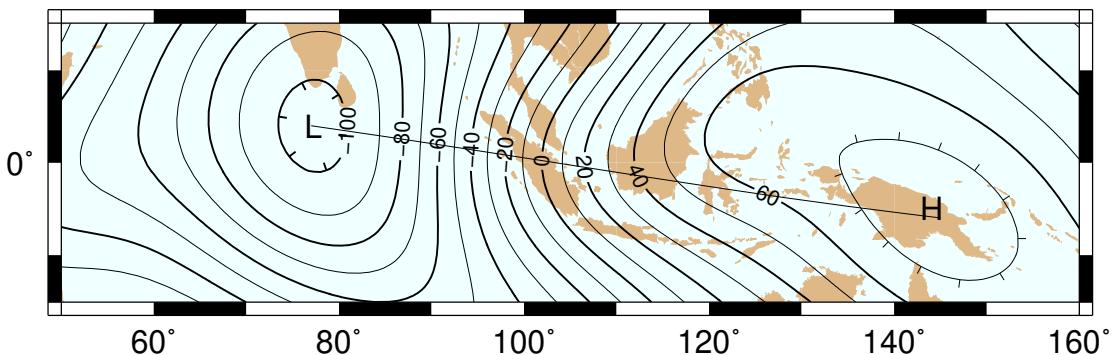


图 15.5: 通过指定 **-GL** 选项的参数确定了标注的位置 (大圆弧与等值线的交点)

图中的标注位于数据极值点连线 (**Z-/Z+**) 与等值线的交点。图中极值点连线为两点之间的大圆弧，在其与等值线交点位置处放置了标注。同一幅地图中，可以分别指定多条线段。

广义的线段与等值线相交算法

如果需要指定的与等值线相交的线段比较多，或线段数据来自其他数据集，可以使用广义的相交算法确定标注的位置。多段数据文件 *cross.txt* 中定义了三条曲线，在这三条曲线与等值线交点位置处放置了标注，

```
#!/bin/bash
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno5.ps
gmt grdcontour geoid.nc -J -O -B20f10 -BWSne -C10 -A20+d+f8p -GXcross.txt -S10 -T+llH >> GMT_contour-anno5.ps
rm gmt.*
```

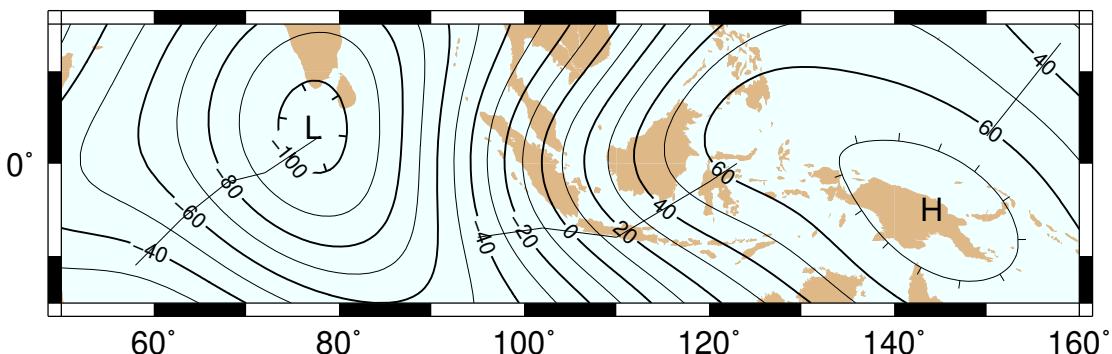


图 15.6: 通过指定 **-GX** 选项的参数 (多段数据文件 *cross.txt*), 确定了标注的位置

15.13.4 标注属性实例

本节通过实例说明标注属性的作用，采用 `psxy` 绘制了大地水准面极值点之间的大圆弧，并且沿着该大圆弧从 ETOPO5 数据集中提取了高程数据。高程数据文件 (`transect.txt`) 中包括了 经度、纬度、距离、大地水准面、高程数据。

按照沿大圆弧距离放置标注, 1

在本实例中将标注的走向从沿大圆弧改变为跨大圆弧，并指定了不透明的文本框和轮廓线，增加了标注的可读性。沿大圆弧每 1000km 放置一个标注，使用距离值作为标注的内容。标注的方向与大圆弧垂直：

```
#!/bin/bash
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno6.ps
gmt grdcontour geoid.nc -J -O -K -B20f10 -BWSne -C10 -A20+d+f8p -Gl50/10S/160/10S -S10 -T+l >> GMT_contour-anno6.ps
gmt psxy -R -J -O -SqD1000k:+g+LD+an+p -Wthick transect.txt >> GMT_contour-anno6.ps
rm gmt.*
```

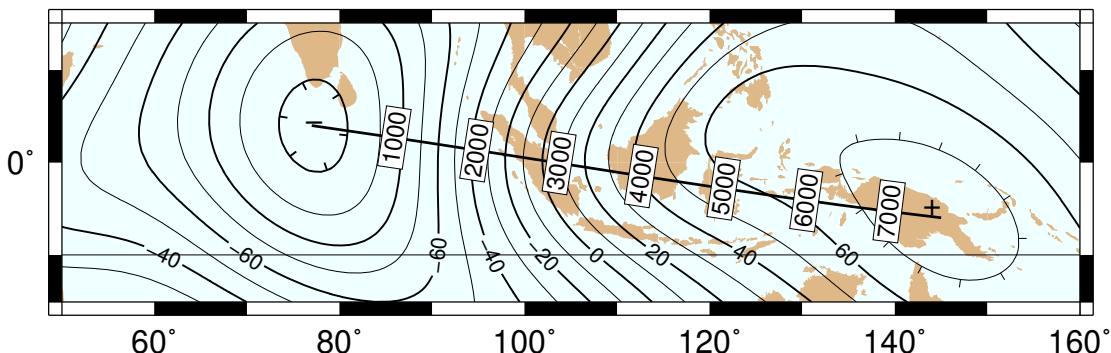


图 15.7：通过指定 `-Sq` 选项的参数控制标注属性。

图中显示了上述命令的综合效果。值得注意的是，大圆弧的起点和终点没有与表示极值点的“-”和“+”符号完全重合。造成这个现象的原因是，极值点符号“-”和“+”的坐标是等值线的平均值，而不是全局或局部极值的位置。

按照沿大圆弧距离放置标注, 2

与上一个实例不同的是，本实例中标注与大圆弧平行，以弧度指定标注位置，并添加弧度单位。文本框的形状为圆角矩形，且标注内容与文本框的底色呈反色显示。

```
#!/bin/bash
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno7.ps
gmt grdcontour geoid.nc -J -O -K -B20f10 -BWSne -C10 -A20+d+u" m"+f8p -Gl50/10S/160/10S -S10 -T+l >> GMT_contour-anno7.ps
gmt psxy -R -J -O -SqD15d:+gblack+fwhite+LD+o+u"\260" -Wthick transect.txt >> GMT_contour-anno7.ps
rm gmt.*
```

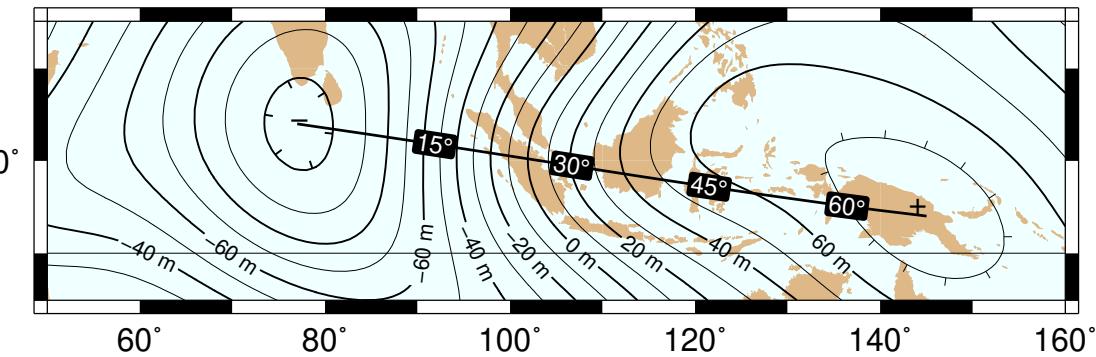


图 15.8: 另一个标注属性实例

使用不同数据集定义标注的内容和位置

本实例中采用沿大圆弧的海底地形数据作为标注的内容，按照沿大圆弧的距离，每 1500km 放置一个标注。因此需要使用 `awk` 程序从 `transect.txt` 文件中抽取距离为 1500km 倍数的记录，并创建一个新文件，指定标注的位置和内容：

```
#!/bin/bash
gmt convert -i0,1,4 -Em150 transect.txt | gawk '{print $1,$2,int($3)}' > fix2.txt
gmt pscoast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500 -K -P > GMT_contour-anno8.ps
gmt grdcontour geoid.nc -J -O -K -B20f10 -BWSne -C10 -A20+d+u" m"+f8p -GL50/10S/160/10S
gmt -S10 -T+l >> GMT_contour-anno8.ps
gmt psxy -R -J -O -Sqffix2.txt:+g+an+p+Lf+u" m"+f8p -Wthick transect.txt >> GMT_contour-anno8.ps
rm gmt.*
```

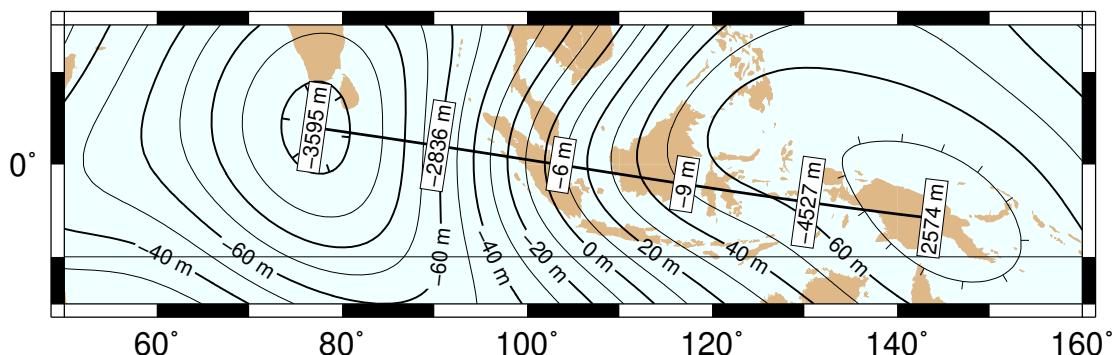


图 15.9: 标注的位置和内容来自不同的数据集

15.13.5 综合实例

最后，采用之前章节中论述的多个标注位置确定方法和属性设置，绘制了一幅比较复杂的综合性图件。假设在 Canary Islands 发生了灾难性滑坡，图件显示了所引发的海啸的走时（以小时为单位）。根据海啸走时和海底地形绘制了彩图，对等值线和线条进行了标注。完整的脚本如下：

```

#!/bin/bash
R=-R-85/5/10/55
gmt grdgradient topo5.nc -Nt1 -A45 -Gtopo5_int.nc
gmt set FORMAT_GEO_MAP ddd:mm:ssF FONT_ANNOT_PRIMARY +9p FONT_TITLE 22p
gmt project -E-74/41 -C-17/28 -G10 -Q > great_NY_Canaries.txt
gmt project -E-74/41 -C2.33/48.87 -G100 -Q > great_NY_Paris.txt
km=`echo -17 28 | gmt mapproject -G-74/41/k -fg --FORMAT_FLOAT_OUT=%.0f -o2` 
cat << EOF > ttt.cpt
0      lightred      3      lightred
3      lightyellow    6      lightyellow
6      lightgreen     100     lightgreen
EOF
gmt grdimage ttt_atl.nc -Itopo5_int.nc -Cttt.cpt $R -JM5.3i -P -K -nc+t1 > GMT_contour-
↪anno9.ps
gmt grdcontour ttt_atl.nc -R -J -O -K -C0.5 -A1+u" hour"+v+f8p,Bookman-Demi -GL80W/31N/
↪17W/26N,17W/28N/17W/50N -S2 >> GMT_contour-anno9.ps
gmt psxy -R -J -Wfatter,white great_NY_Canaries.txt -O -K >> GMT_contour-anno9.ps
gmt pscoast -R -J -B20f5 -BWSne+Tsunami travel times from the Canaries" -N1/thick -O -
↪K -Glightgray -Wfaint -A500 >> GMT_contour-anno9.ps
gmt convert great_NY_*.txt -E | gmt psxy -R -J -O -K -Sa0.15i -Gred -Wthin >> GMT_
↪contour-anno9.ps
gmt psxy -R -J -Wthick great_NY_Canaries.txt -O -K -Sqn1:+f8p,Times-Italic+l"Distance"
↪Canaries to New York = $km km"+ap+v >> GMT_contour-anno9.ps

gmt psxy -R -J -Wthinner great_NY_Paris.txt -SqD1000k:+an+o+gblue+LDk+f7p,Helvetica-
↪Bold,white -O -K >> GMT_contour-anno9.ps
cat << EOF | gmt ptext -R -J -O -K -Gwhite -Wthin -Dj0.1i/0.1i -F+f8p,Bookman-Demi+j >>
↪ GMT_contour-anno9.ps
74W      41N      RT      New York
2.33E     48.87N    CT      Paris
17W      28N      CT      Canaries
EOF
gmt psxy -R -J -O -T >> GMT_contour-anno9.ps
rm gmt.* topo5_int.nc

```

Tsunami travel times from the Canaries

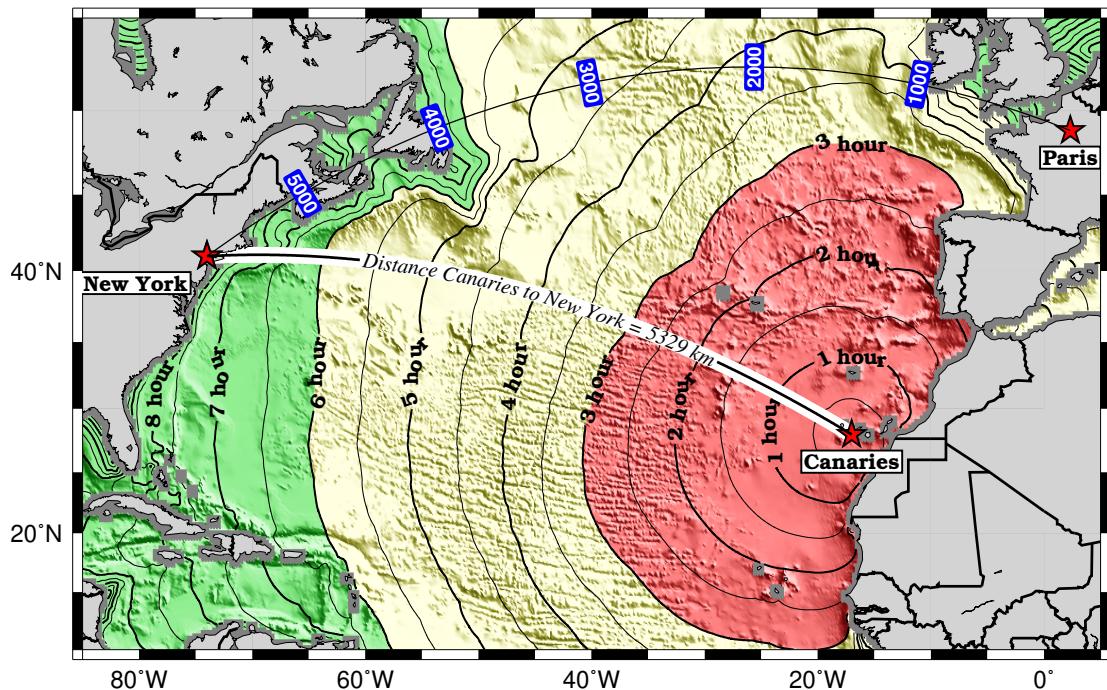


图 15.10: Canary Islands 到大西洋沿岸的海啸走时图, 特别是纽约。当发生灾难性滑坡时, 纽约将在 8 小时后遭遇大海啸。

15.14 通过 GDAL 读写网格文件

待处理: 尚未完成, 欢迎翻译。

官 方 文 档 见 http://gmt.soest.hawaii.edu/doc/latest/GMT_Docs.html#modifiers-to-read-and-write-grids-and-images-via-gdal

索引

Non-alphabetical

纸张原点 (PAPER_ORIGIN), [84](#)
网格线 (GRID), [84](#)

B

backtracker, [226](#)
blockmean, [226](#)
blockmedian, [226](#)
blockmode, [227](#)

C

colors, [61](#)

D

dimfilter, [227](#)

F

fill, [65](#)
filter1d, [227](#)
fitcircle, [228](#)
fonts, [67](#)

G

gmt, [231](#)
gmt2kml, [233](#)
gmt5syntax, [233](#)
gmt.conf, [183](#)
gmt-config, [232](#)
gmtconnect, [233](#)
gmtconvert, [234](#)
gmtdefaults, [234](#)
gmtflexure, [235](#)
gmtget, [235](#)
gmtgravmag3d, [236](#)
gmtinfo, [236](#)
gmtlogo, [237](#)
gmtmath, [238](#)
gmtpmodeler, [238](#)

gmtregress, [238](#)
gmtselect, [239](#)
gmtset, [242](#)
gmtsimplify, [243](#)
gmtspatial, [244](#)
gmtswitch, [244](#)
gmtvector, [244](#)
gmtwhich, [244](#)
gpsgriddler, [245](#)
gravfft, [245](#)
grd2cpt, [245](#)
grd2rgb, [245](#)
grd2xyz, [246](#)
grdblend, [248](#)
grdclip, [249](#)
grdcontour, [250](#)
grdconvert, [264](#)
grdcut, [250](#)
grdedit, [252](#)
grdffft, [254](#)
grdfilter, [254](#)
grdflexure, [254](#)
grdgradient, [254](#)
grdgravmag3d, [254](#)
grdhisteq, [254](#)
grdimage, [254](#)
grdinfo, [255](#)
grdlandmask, [258](#)
grdmask, [260](#)
grdmath, [262](#)
grdpaste, [262](#)
grdpmodeler, [262](#)
grdproject, [262](#)
grdraster, [263](#)
grdredpol, [264](#)
grdrotater, [265](#)

grdsample, 265
grdseamount, 267
grdspotter, 267
grdtrack, 267
grdtrend, 267
grdvector, 268
grdview, 269
grdvolumne, 269
greenspline, 271
gshhg, 271

H

hotspotter, 271

I

img2google, 271
img2grd, 271
isogmt, 271

K

kml2gmt, 272

M

makecpt, 273
mapproject, 273
mgd77convert, 274
mgd77header, 274
mgd77info, 274
mgd77list, 274
mgd77magref, 274
mgd77manage, 275
mgd77path, 275
mgd77sniffer, 275
mgd77track, 275

N

nearneighbor, 275

O

originator, 275

P

pen, 63
postscript, 95
project, 275
psbasemap, 277
psclip, 283
pscoast, 284
pscontour, 290
psconvert, 290
pscoupe, 294
pshistogram, 300
psimage, 302
pslegend, 303
psmask, 310
psmeca, 310
pspolar, 317
psrose, 320
pssac, 322
psscale, 325
pssegy, 326
pssegyz, 326
pssolar, 327
psternary, 329
pstext, 329
psvelo, 335
pswiggle, 340
psxy, 340
psxyz, 356

R

rotconverter, 356
rotsmoother, 356

S

sample1d, 357
segy2grd, 358
spectrum1d, 358
sph2grd, 360
sphdistance, 360
sphinterpolate, 360

sphtriangulate, [360](#)

splitxyz, [360](#)

surface, [360](#)

T

talwani2d, [360](#)

talwani3d, [360](#)

transparency, [146](#)

trend1d, [360](#)

trend2d, [360](#)

triangulate, [360](#)

V

刻度 (TICK), [84](#)

坐标轴 (AXIS), [84](#)

子图, [84](#)

底图原点 (BASEMAP_ORIGIN), [84](#)

底图 (BASEMAP), [84](#)

W

标注 (ANNOT), [84](#)

标签 (LABEL), [84](#)

标题 (TITLE), [84](#)

X

x2sys_binlist, [360](#)

x2sys_cross, [361](#)

x2sys_datalist, [361](#)

x2sys_get, [361](#)

x2sys_init, [361](#)

x2sys_list, [361](#)

x2sys_merge, [361](#)

x2sys_put, [361](#)

x2sys_report, [361](#)

x2sys_solve, [361](#)

xyz2grd, [361](#)