GMT 中文手册 v6.0

GMT 中文社区

2019 年 11 月 09 日

保护环境, 从阅读电子文档开始!

目　录

第 1 章　简介 **23**

1.1 GMT 简介 23

1.1.1 GMT 是什么 23

1.1.2 GMT 的历史 23

1.1.3 GMT 开发者 23

1.1.4 GMT 的特点 24

1.1.5 其他制图软件 25

1.2 GMT 版本 25

1.2.1 版本号 25

1.2.2 GMT 主流版本 . . . 25

1.3 引用 26

1.4 学习资源 27

第 2 章　安装 **29**

2.1 Linux 下安装 GMT 29

2.1.1 Fedora 29

2.1.2 RHEL/CentOS . . . 30

2.1.3 Ubuntu/Debian 用户 30

2.2 Windows 下安装 GMT . . . 30

2.2.1 安装 GMT 31

2.2.2 安装其它可选软件 . 32

2.3 macOS 下安装 GMT 33

2.3.1 使用 Homebrew 安装 33

2.3.2 使用 GMT 安装包 . 34

2.3.3 使用 Macports 安装 34

2.4 跨平台 GMT 安装方案 . . . 35

2.4.1 通过 conda 安装 . . 35

2.5 Linux/macOS 下编译 GMT 源码 35

2.5.1 编译及运行依赖 . . . 35

2.5.2 安装依赖软件 36

2.5.3 下载源码及数据 . . . 37

2.5.4 安装 GMT 37

2.5.5 修改环境变量 39

2.5.6 测试是否安装成功 . 40

2.5.7 升级/卸载 GMT . . 40

第 3 章　入门教程 **41**

3.1 GMT 初探 41

3.1.1 GMT 初探: Linux 篇 41

3.1.2 GMT 初探: macOS 篇 43

3.1.3 GMT 初探: Windows 篇 46

3.2 命令初探 48

3.2.1 GMT 绘图脚本模板 48

3.2.2 绘制第一张图 49

3.2.3 GMT 命令格式 . . . 50

3.3 绘制底图 51

3.3.1 全球地图 51

3.3.2 区域地图 53

3.3.3 线性坐标轴 55

3.3.4 三维线性坐标轴 . . . 56

3.4 绘制海岸线 57

3.4.1 绘制海岸线和湖岸线 57

3.4.2 设置要绘制的对象的最小面积 60

3.4.3 填充陆地与水体 . . . 60

3.4.4 绘制国界/州界 . . . 61

3.4.5 添加比例尺 62

3.5 绘制线段和多边形 63

3.5.1 绘制一条线段 63

3.5.2 绘制一个多边形 . . . 65

3.5.3 绘制多条线段 67

3.5.4 绘制多个多边形 . . . 68

3.5.5 大圆弧路径 69

3.6 绘制符号 70

3.6.1 符号一览 70

3.6.2 绘制简单符号 70
3.6.3 大小变化的符号 . . . 72
3.6.4 颜色变化的符号 . . . 72
3.6.5 绘制不同的符号 . . . 73
3.6.6 大小、颜色和符号类型都变化的符号 . . . 74
3.7 添加文字 75
3.7.1 最简单的示例 75
3.7.2 文字属性 76
3.7.3 文本框 77
3.7.4 对齐方式 77
3.7.5 文本偏移量 78
3.7.6 变化的文字属性 . . . 79
3.8 添加图例 79
3.8.1 自动图例 79
3.8.2 手动设置图例 80
3.9 绘制地形起伏 81
3.9.1 全球地形起伏数据 . 82
3.9.2 绘制全球地形起伏图 82
3.9.3 绘图区域地形起伏图 83
3.9.4 增加光照效果 83
3.9.5 添加色标 84
3.9.6 制作 CPT 文件 . . . 87
3.10 理解图层 87
3.10.1 图件分析与拆解 . . . 88
3.10.2 绘制底图 88
3.10.3 绘制震中和台站位置 89
3.10.4 绘制射线路径 90
3.10.5 添加台站名 91
3.10.6 图层的先后顺序 . . . 93
3.11 GMT 单行模式 95
3.12 GMT 图中图模式 96
3.12.1 用矩形框标记研究区域 96
3.12.2 用颜色标记研究区域 97
3.13 GMT 子图模式 98
3.13.1 子图布局 98
3.13.2 第一张子图 99
3.13.3 共用 X/Y 轴 100
3.13.4 复杂布局 101
3.14 GMT 多图模式 102

第 4 章 GMT 基础 105
4.1 画布 105
4.1.1 图片格式 105
4.1.2 画布颜色 105
4.1.3 画布大小 106
4.2 单位 106
4.2.1 长度单位 106
4.2.2 距离单位 107
4.2.3 小练习 107
4.3 颜色 107
4.3.1 颜色名 107
4.3.2 RGB 108
4.3.3 HSV 109
4.3.4 CMYK 109
4.3.5 灰色 109
4.3.6 透明色 110
4.3.7 颜色小结 110
4.4 画笔 110
4.4.1 画笔线宽 110
4.4.2 画笔颜色 111
4.4.3 画笔线型 111
4.4.4 小练习 112
4.5 填充 113
4.5.1 填充颜色 113
4.5.2 填充图案 113
4.5.3 小练习 114
4.5.4 附录: GMT 预定义填充图案 114
4.6 文字 115
4.6.1 文字大小 116
4.6.2 字体 116
4.6.3 填充色 117
4.6.4 描边 117
4.6.5 小练习 117
4.7 特殊字符 118
4.7.1 八进制码指定特殊字符 118

4.7.2 使用 12 号或 34 号字体输入特殊字符 . 120
4.7.3 特殊字符小结 120
4.7.4 小练习 121
4.8 转义序列 121
4.8.1 转义字符 121
4.8.2 小练习 122
4.8.3 注意事项 123
4.9 绘制矢量/箭头 123
4.9.1 矢量头位置及形状 . 123
4.9.2 矢量头线条颜色和填充色 124
4.9.3 其它属性 124
4.9.4 矢量类型及输入数据格式 125
4.10 线条属性 126
4.10.1 端点偏移量 126
4.10.2 线条平滑 127
4.10.3 端点箭头 127
4.10.4 其它属性 128
4.11 锚点 128
4.12 绘制修饰物 129
4.12.1 定位 130
4.12.2 背景面板 132
4.13 数据类型 132
4.13.1 地理坐标 133
4.13.2 绝对时间坐标 133
4.13.3 相对时间坐标 134
4.13.4 一般坐标值 134

第 5 章 表数据 135
5.1 ASCII 表 135
5.1.1 ASCII 表简介 135
5.1.2 文件头记录 136
5.1.3 多段数据 136
5.1.4 数据段头记录中的额外属性 137
5.2 二进制表 138
5.3 netCDF 表 138
5.4 兼容 OGR 的 GMT 矢量数据格式 138
5.4.1 简介 138
5.4.2 OGR/GMT 格式 . . 139
5.4.3 OGR/GMT 元数据 . 139
5.4.4 OGR/GMT 数据 . . 141
5.4.5 示例 141

第 6 章 网格文件 145
6.1 网格文件格式 145
6.2 读 netCDF 文件 146
6.2.1 读二维单变量 netCDF 文件 147
6.2.2 读取二维多变量 netCDF 文件 147
6.2.3 读取三维单/多变量 netCDF 文件 148
6.2.4 读取四维单/多变量 netCDF 文件 149
6.2.5 读取一维单/多变量 netCDF 文件 149
6.2.6 修改坐标单位 150
6.3 写 netCDF 文件 150
6.3.1 文件名格式 150
6.3.2 分块与压缩 151
6.4 网格配准 151
6.4.1 网格线配准 152
6.4.2 像素配准 152
6.5 边界条件 152
6.5.1 默认边界条件 152
6.5.2 周期边界条件 152
6.5.3 地理边界条件 153
6.6 查看 netCDF 文件 153

第 7 章 CPT 文件 155
7.1 CPT 类型 155
7.1.1 分类 CPT 155
7.1.2 常规 CPT 156
7.2 内置 CPT 157
7.3 制作 CPT 160

7.3.1 周期性 CPT 文件 . 162
7.3.2 动态 CPT 162
7.4 使用 CPT 163
7.5 CPT 资源 164

第 8 章 标准选项 165
8.1 -R 选项 166
8.1.1 五种方式 166
8.1.2 六种坐标 168
8.2 -J 选项 169
8.2.1 GMT 投影代码 . . . 169
8.2.2 Proj4 投影代码 . . . 171
8.2.3 GMT+PROJ4 . . . 172
8.3 -B 选项 173
8.3.1 边框设置 173
8.3.2 轴设置 175
8.3.3 地理底图 177
8.3.4 笛卡尔线性轴 178
8.3.5 笛卡尔 $\log_{10}$ 轴 . . . 178
8.3.6 笛卡尔指数轴 179
8.3.7 时间轴 179
8.3.8 弧度轴 π 的标注 . . 182
8.3.9 自定义轴 183
8.4 -V 选项 184
8.5 -U 选项 184
8.6 -X 和 -Y 选项 185
8.7 -a 选项 187
8.8 -b 选项 188
8.9 -c 选项 189
8.10 -d 选项 190
8.11 -e 选项 190
8.12 -f 选项 192
8.13 -g 选项 192
8.14 -h 选项 193
8.15 -i 和 -o 选项 194
8.16 -j 选项 194
8.16.1 三种距离计算方式 . 195
8.17 -n 选项 196
8.18 -r 选项 196
8.19 -p 选项 196
8.20 -s 选项 198
8.21 -t 选项 199
8.22 -x 选项 199
8.23 -: 选项 200

第 9 章 投影方式 201
9.1 -Jx: 笛卡尔变换 201
9.1.1 笛卡尔线性坐标 . . . 201
9.1.2 笛卡尔对数投影 . . . 204
9.1.3 笛卡尔指数投影 . . . 204
9.2 -Jp: 极坐标线性投影 205
9.3 -Ja: Lambert 方位等面积投影 206
9.3.1 矩形地图 207
9.3.2 半球地图 207
9.3.3 震源辐射花样 208
9.4 -Jb: Albers 圆锥等面积投影 208
9.5 -Jc: Cassini 圆柱投影 . . . 209
9.6 -Jcyl_stere: 圆柱立体投影 . 210
9.7 -Jd: 等距圆锥投影 211
9.8 -Je: 方位等距投影 212
9.9 -Jf: 球心方位投影 213
9.10 -Jg: 正交投影 214
9.11 -Jh: 等面积 Hammer 投影 . 216
9.12 -Ji: 正弦曲线投影 217
9.13 -Jj: Miller 圆柱投影 218
9.14 -Jk: Eckert 投影 219
9.15 -Jl: Lambert 圆锥保角投影 220
9.16 -Jm: Mercator 投影 221
9.17 -Jn: Robinson 投影 222
9.18 -Jo: 倾斜 Mercator 投影 . . 223
9.19 -Jpoly: 多圆锥投影 224
9.20 -Jq: 圆柱等距投影 225
9.21 -Jr: Winkel Tripel 投影 . . 226
9.22 -Js: 立体等角投影 227
9.22.1 极区立体地图 227
9.22.2 矩形立体地图 228
9.22.3 一般立体地图 228
9.23 -Jt: 横向 Mercator 投影 . . 229
9.24 -Ju: 通用横向 Mercator(UTM) 投影 231

9.25 -Jv: Van der Grinten 投影 . 232
9.26 -Jw: Mollweide 投影 233
9.27 -Jy: 圆柱等面积投影 234

第 10 章 配置参数 237
10.1 配置参数简介 237
10.1.1 查看配置参数的值 . 237
10.1.2 修改配置参数的值 . 237
10.2 配置参数示例 238
10.3 FONT 参数 239
10.4 MAP 参数 240
10.4.1 边框相关参数 240
10.4.2 标注相关参数 240
10.4.3 标签相关参数 241
10.4.4 刻度相关参数 241
10.4.5 网格线相关参数 . . . 242
10.4.6 标题相关参数 243
10.4.7 其它参数 243
10.5 COLOR 参数 244
10.5.1 CPT 相关参数 . . . 244
10.5.2 光照相关参数 244
10.6 DIR 参数 244
10.7 FORMAT 参数 245
10.7.1 日期的输入/输出/绘图格式 245
10.7.2 时间的输入/输出/绘图格式 246
10.7.3 地理坐标的输出/绘图格式 246
10.7.4 浮点数的输出/绘图格式 247
10.7.5 其他数据的绘图格式 247
10.8 IO 参数 248
10.8.1 表数据相关参数 . . . 248
10.8.2 网格文件相关参数 . 249
10.8.3 其他 IO 参数 250
10.9 PROJ 参数 250
10.10 PS 参数 251
10.11 TIME 参数 254
10.12 其他参数 256
10.12.1 数据下载相关参数 . 256
10.12.2 算法选择相关参数 . 257
10.12.3 其他参数 257

第 11 章 地学数据集 259
11.1 地学数据集 259
11.1.1 数据库原理 259
11.1.2 建立 GMT 数据库 . 259
11.2 GSHHG: 全球高精度海岸线数据 260
11.2.1 数据精度 260
11.2.2 数据内容 261
11.2.3 使用示例 262
11.2.4 备注 264
11.3 DCW: 世界数字图表 264
11.3.1 区域代码 264
11.3.2 使用说明 266
11.3.3 使用示例 267
11.3.4 备注 269
11.4 earth_relief: 全球地形起伏数据 269
11.4.1 地形起伏数据简介 . 269
11.4.2 使用方法 271
11.4.3 缓存空间问题 271
11.4.4 技术细节 271
11.4.5 数据来源及引用 . . . 272
11.5 GADM: 全球行政区划数据库 272
11.5.1 数据下载 272
11.5.2 数据格式及转换 . . . 272
11.5.3 数据分级 273
11.5.4 使用示例 274
11.5.5 许可协议 276
11.5.6 备注 276

第 12 章 中文支持 277
12.1 Linux 下的 GMT 中文支持 277
12.1.1 ghostscript 的中文支持 277
12.1.2 GMT 的中文支持 . . 278
12.1.3 GMT 中文测试 . . . 278

12.2 macOS 下的 GMT 中文支持 279
12.2.1 ghostscript 的中文支持 279
12.2.2 GMT 的中文支持 . . 280
12.2.3 GMT 中文测试 . . . 280
12.3 Windows 下的 GMT 中文支持 281
12.3.1 ghostscript 的中文支持 281
12.3.2 配置 Ghostscript 环境变量 282
12.3.3 gsview 的中文支持 . 282
12.3.4 GMT 的中文支持 . . 282
12.3.5 GMT 中文测试 . . . 283
12.4 自定义字体 284
12.4.1 基本原理 284
12.4.2 ghostscript 中文配置 285
12.4.3 GMT 中文支持 . . . 286
12.4.4 GMT 中文测试 . . . 287
12.4.5 对其他发行版的若干说明 288
12.4.6 参考资料 289

第 13 章 模块手册 291

13.1 basemap 295
13.1.1 必选选项 295
13.1.2 可选选项 298
13.1.3 示例 298
13.1.4 BUGS 299
13.2 begin 300
13.2.1 语法 300
13.2.2 可选选项 300
13.2.3 示例 301
13.2.4 PS 文件注意事项 . . 302
13.2.5 UNIX shell 注意事项 302
13.2.6 相关模块 302
13.3 clear 302
13.3.1 语法 303
13.3.2 可选选项 303
13.3.3 示例 303
13.3.4 相关模块 303
13.4 clip 303
13.4.1 必选选项 304
13.4.2 可选选项 304
13.4.3 示例 304
13.5 coast 304
13.5.1 必选选项 305
13.5.2 可选选项 305
13.5.3 示例 309
13.5.4 FAQ 310
13.6 colorbar 310
13.6.1 必选选项 310
13.6.2 选项 311
13.6.3 示例 312
13.7 coupe 312
13.7.1 剖面类型 312
13.7.2 选项 313
13.7.3 示例 314
13.8 docs 318
13.8.1 语法 319
13.8.2 必须参数 319
13.8.3 可选选项 319
13.8.4 示例 319
13.8.5 相关模块 320
13.9 end 320
13.9.1 语法 320
13.9.2 可选参数 320
13.9.3 示例 321
13.9.4 禁用自动显示 321
13.9.5 相关模块 321
13.10 figure 321
13.10.1 语法 321
13.10.2 必须选项 321
13.10.3 可选选项 322
13.10.4 PS 文件注意事项 . 323
13.10.5 示例 323
13.10.6 技术细节 324
13.10.7 相关模块 324
13.11 filter1d 324
13.11.1 必选选项 324

13.11.2 选项 325
13.11.3 示例 325
13.12 fitcircle 325
13.12.1 必选选项 326
13.12.2 选项 326
13.12.3 示例 327
13.13 gmt 328
13.13.1 示例 329
13.14 gmt-config 329
13.15 gmt5syntax 330
13.15.1 用法 330
13.16 gmtconnect 330
13.16.1 选项 330
13.16.2 示例 331
13.17 gmtdefaults 331
13.17.1 必选选项 331
13.17.2 可选选项 331
13.17.3 示例 331
13.18 gmtget 332
13.18.1 语法 332
13.18.2 必选选项 332
13.18.3 可选选项 332
13.18.4 示例 332
13.18.5 相关模块 333
13.19 gmtinfo 333
13.19.1 最小示例 333
13.19.2 可选选项 333
13.20 gmtlogo 334
13.20.1 必选选项 335
13.20.2 可选选项 335
13.20.3 示例 335
13.21 gmtselect 336
13.21.1 七个空间准则 . . . 336
13.21.2 其他选项 338
13.21.3 示例 338
13.22 gmtset 339
13.22.1 必选选项 339
13.22.2 可选选项 339
13.22.3 FAQ 340
13.23 gmtsimplify 340
13.23.1 选项 340
13.23.2 示例 340
13.23.3 参考文献 341
13.24 gmtwhich 341
13.24.1 必选选项 341
13.24.2 可选选项 341
13.25 grd2xyz 341
13.25.1 必选选项 341
13.25.2 可选选项 342
13.25.3 示例 343
13.26 grdblend 343
13.26.1 必选选项 343
13.26.2 可选选项 343
13.26.3 示例 344
13.27 grdclip 344
13.27.1 必选选项 344
13.27.2 可选选项 345
13.27.3 示例 345
13.28 grdconvert 345
13.28.1 必选选项 345
13.28.2 示例 345
13.29 grdcut 346
13.29.1 必选选项 346
13.29.2 可选选项 346
13.29.3 示例 347
13.29.4 相关 347
13.30 grdedit 347
13.30.1 必选选项 348
13.30.2 可选选项 348
13.30.3 示例 349
13.31 grdimage 349
13.31.1 必选选项 349
13.31.2 可选选项 349
13.31.3 示例 350
13.32 grdinfo 350
13.32.1 必选选项 350
13.32.2 可选选项 351
13.33 grdlandmask 353
13.33.1 必选选项 353
13.33.2 可选选项 354

13.33.3 注意事项 354
13.33.4 示例 355
13.34 grdmask 355
13.34.1 必选选项 355
13.34.2 可选选项 355
13.34.3 注意事项 356
13.34.4 示例 356
13.35 grdpaste 357
13.35.1 必选选项 357
13.35.2 示例 357
13.36 grdproject 357
13.36.1 必选选项 357
13.36.2 可选选项 357
13.36.3 示例 358
13.37 grdsample 358
13.37.1 必选选项 358
13.37.2 可选选项 358
13.37.3 示例 359
13.38 grdtrack 359
13.38.1 选项 359
13.38.2 相关 359
13.39 grdtrend 359
13.39.1 必选选项 360
13.39.2 可选选项 360
13.39.3 示例 360
13.40 grdvector 360
13.40.1 必选选项 360
13.40.2 可选选项 361
13.40.3 示例 361
13.41 grdvolume 361
13.41.1 必选选项 362
13.41.2 可选选项 362
13.41.3 示例 362
13.41.4 引用 363
13.42 histogram 363
13.42.1 必选选项 363
13.42.2 可选选项 363
13.42.3 示例 364
13.43 image 364
13.43.1 必选选项 365
13.43.2 可选选项 365
13.43.3 示例 366
13.44 inset 366
13.44.1 inset begin 语法 . . 366
13.44.2 必须选项 367
13.44.3 可选选项 367
13.44.4 示例 368
13.45 kml2gmt 368
13.45.1 必选选项 368
13.45.2 可选选项 368
13.45.3 示例 368
13.46 legend 369
13.46.1 选项 369
13.46.2 图例文件格式 . . . 369
13.46.3 示例 371
13.47 makecpt 375
13.47.1 选项 375
13.48 mapproject 375
13.48.1 最小示例 376
13.48.2 -I 选项 376
13.48.3 -Q 选项 376
13.49 meca 376
13.49.1 -S 选项 376
13.49.2 选项 379
13.49.3 示例 380
13.50 plot 380
13.50.1 可选选项 380
13.50.2 -S 选项 384
13.50.3 输入数据格式 . . . 392
13.50.4 多段数据 393
13.50.5 示例 393
13.51 polar 396
13.51.1 选项 397
13.51.2 示例 397
13.52 project 400
13.52.1 必选选项 402
13.52.2 可选选项 402
13.52.3 示例 402
13.53 psconvert 403
13.53.1 必选选项 403

13.53.2 可选选项 403
13.53.3 其他 406
13.53.4 FAQ 406
13.53.5 BUGS 406
13.54 rose 407
13.54.1 可选选项 407
13.55 sac 408
13.55.1 必选选项 409
13.55.2 可选选项 409
13.55.3 示例 410
13.56 sample1d 411
13.56.1 可选选项 411
13.56.2 示例 412
13.57 solar 412
13.57.1 必选选项 412
13.57.2 可选选项 412
13.57.3 示例 414
13.57.4 BUGS 414
13.58 spectrum1d 415
13.58.1 选项 415
13.58.2 示例 416
13.59 subplot 416
13.59.1 subplot begin 语法 416
13.59.2 必须选项 417
13.59.3 可选选项 418
13.59.4 subplot set 420
13.59.5 subplot set 语法 . . 420
13.59.6 可选选项 420
13.59.7 subplot end 421
13.59.8 subplot end 语法 . 421
13.59.9 示例 421
13.60 ternary 424
13.60.1 可选选项 424
13.61 text 425
13.61.1 必选选项 425
13.61.2 可选选项 426
13.61.3 示例 430
13.62 velo 430
13.62.1 -S 选项 430
13.62.2 选项 431
13.62.3 示例 432
13.63 xyz2grd 435
13.63.1 必选选项 435
13.63.2 可选选项 435
13.63.3 示例 437

第 14 章 绘图示例 439
14.1 绘制震源球 439
14.1.1 震源球大小随震级变化 439
14.1.2 固定大小的震源球 . 440
14.1.3 震源球大小随震级变化，颜色随深度变化 441
14.2 绘制直方图 442
14.2.1 垂直直方图 442
14.2.2 水平直方图 443
14.2.3 累积直方图 444
14.2.4 百分比直方图 445
14.3 制作动画 446
14.3.1 绘制正弦函数动画 . 447
14.3.2 绘制旋转的地球 . . . 447
14.4 中文图 449
14.4.1 中文文字 449
14.4.2 中文月份 450
14.4.3 中文星期 451

第 15 章 API 453
15.1 GMT C API 453
15.2 GMT Matlab API 453
15.2.1 简介 453
15.2.2 安装 453
15.2.3 使用方法 455
15.2.4 常见问题 456
15.3 GMT/Julia 456
15.4 GMT/Python 457

第 16 章 附录 459
16.1 命令行及脚本基础 459
16.1.1 命令行基础 459
16.1.2 bash 简明教程 . . . 462
16.1.3 bat 简明教程 465

16.1.4 Unix 常用工具 . . . 467
16.2 现代模式 vs 经典模式 . . . 467
16.2.1 经典模式的问题 . . . 468
16.2.2 现代模式的优点 . . . 468
16.2.3 从经典到现代 469
16.3 GMT 底层原理 469
16.3.1 命令行历史 469
16.4 环境变量 470
16.4.1 $GMT_SHAREDIR . . . 470
16.4.2 $GMT_DATADIR . . . 470
16.4.3 $GMT_USERDIR . . . 470
16.4.4 $GMT_TMPDIR 471
16.4.5 $GMT_CACHEDIR . . . 471
16.5 目录参数 471
16.6 等值线标注和“线条标注” . 471
16.6.1 标注的位置 472
16.6.2 标注的属性 473
16.6.3 等值线标注位置实例 474
16.6.4 标注属性实例 477
16.6.5 综合实例 479

图目录

1 GMT 核心开发者及指导委员会部分成员 24

1 使用 GMT 绘制的第一张图 50
2 全球地图（无边框版） 51
3 全球地图（有边框版） 52
4 使用国家代码指定绘图区域 53
5 使用经纬度范围指定绘图区域 54
6 为区域地图添加标题 55
7 线性坐标轴 56
8 三维线性坐标轴 57
9 美国海岸线 58
10 绘制海岸线 59
11 绘制海岸线和湖岸线 59
12 设置要绘制的对象的最小面积 60
13 填充颜色 61
14 绘制国界与州界 62
15 添加比例尺 63
16 全球地形起伏图 82
17 台湾区域地形图 83
18 带光照效果的台湾区域地形图 84

1 GMT 内置位图图案 115
2 ISOLation1+ 和 Standard+ 编码下的八进制码 119
3 Symbol 和 Pifont 字体八进制码 120
4 三种矢量：（左）笛卡尔矢量；（中）地理矢量；（右）弧形矢量 125
5 线段起点偏移示意图 127
6 线条自动样条插值示意图 127
7 线条端点箭头示意图 128
8 GMT 修饰物背景面板 132

1 GMT 网格配准方式 151

1 abyss 158
2 bathy 158
3 batlow 158

4 berlin . . . 158
5 bilbao . . . 158
6 broc . . . 158
7 categorical . . . 158
8 cool . . . 158
9 copper . . . 158
10 cork . . . 158
11 cubhelix . . . 158
12 cyclic . . . 158
13 davos . . . 158
14 dem1 . . . 158
15 dem2 . . . 158
16 dem3 . . . 158
17 dem4 . . . 158
18 drywet . . . 158
19 earth . . . 158
20 elevation . . . 158
21 etopo1 . . . 158
22 gebco . . . 158
23 geo . . . 159
24 globe . . . 159
25 gray . . . 159
26 grayC . . . 159
27 hawaii . . . 159
28 haxby . . . 159
29 hot . . . 159
30 ibcso . . . 159
31 inferno . . . 159
32 jet . . . 159
33 lajolla . . . 159
34 lapaz . . . 159
35 lisbon . . . 159
36 magma . . . 159
37 nighttime . . . 159
38 no_green . . . 159
39 ocean . . . 159
40 oleron . . . 159
41 oslo . . . 159
42 paired . . . 159
43 panoply . . . 159

44 plasma . . . 159
45 polar . . . 159
46 rainbow . . . 159
47 red2green . . . 159
48 relief . . . 159
49 roma . . . 160
50 seafloor . . . 160
51 sealand . . . 160
52 seis . . . 160
53 split . . . 160
54 srtm . . . 160
55 terra . . . 160
56 tofino . . . 160
57 tokyo . . . 160
58 topo . . . 160
59 turku . . . 160
60 vik . . . 160
61 viridis . . . 160
62 world . . . 160
63 wysiwyg . . . 160

1 -R 选项指定数据范围 . . . 168
2 地理底图示例 1 . . . 177
3 地理底图示例 2 . . . 178
4 笛卡尔线性轴 . . . 178
5 对数坐标轴 . . . 179
6 指数投影坐标轴 . . . 179
7 时间轴示例 1 . . . 180
8 时间轴示例 2 . . . 181
9 时间示例 3 . . . 181
10 时间轴示例 4 . . . 181
11 时间轴示例 5 . . . 182
12 时间轴示例 6 . . . 182
13 时间轴示例 7 . . . 182
14 自定义坐标轴 . . . 184
15 **-U** 选项加时间戳 . . . 185

1 笛卡尔坐标的线性变换 . . . 202
2 地理坐标的线性变换 . . . 203
3 日期时间坐标的线性变换 . . . 204

4 对数投影 204
5 指数变换 205
6 极坐标用法示例 206
7 使用 Lambert 方位等面积投影绘制矩形地图 207
8 使用 Lambert 方位等面积投影绘制半球地图 208
9 Albers 圆锥等面积投影 209
10 Cassini 投影绘制 Sardinia 岛 210
11 使用 Gall 立体投影绘制世界地图 211
12 等距圆锥地图投影 212
13 使用等距方位投影绘制全球图 213
14 球心方位投影 214
15 使用正交投影绘制半球 215
16 透视投影 216
17 使用 Hammer 投影绘制全球地图 217
18 使用正弦曲线投影绘制世界地图 218
19 使用间断正弦曲线投影绘制世界地图 218
20 使用 Miller 圆柱投影绘制世界地图 219
21 Eckert IV 投影绘制全球图 220
22 Eckert VI 投影绘制全球图 220
23 Lambert 保角圆锥投影 221
24 Mercator 投影 222
25 使用 Robinson 投影绘制全球地图 223
26 使用 `-Joc` 倾斜 Mercator 投影 224
27 使用 `-JOa` 倾斜 Mercator 投影 224
28 多圆锥投影 225
29 使用 Plate Carrée 投影绘制全球地图 226
30 使用 Winkel Tripel 投影绘制全球地图 227
31 极区立体保角投影 228
32 矩形边界下的极区立体保角投影 228
33 一般立体投影 229
34 矩形横向 Mercator 地图 230
35 全球横向 Mercator 地图 231
36 使用 Van der Grinten 投影绘制全球图 233
37 使用 Mollweide 投影绘制全球地图 234
38 使用 Behrman 圆柱等面积投影绘制地图 235

1 方向玫瑰图 297
2 磁场玫瑰图 299
3 coupe 示例图 1 316
4 coupe 示例图 2 318

5 fitcircle 示例 328
6 legend 示例图 1 372
7 legend 示例图 2 375
8 plot -A 选项示意图 381
9 plot -S 选项示意图 385
10 plot -Sf 示意图 388
11 plot -Sm 示意图 389
12 plot -Sw 示意图。 392
13 plot -F 选项示意图 395
14 plot -L 选项示意图 396
15 polar 示例 400
16 project 示意图 401
17 曙暮光区的多种定义（图片来自于 https://en.wikipedia.org/wiki/Twilight） 414
18 段落模式示意图 429
19 velo 示例图 1 433
20 velo 示例图 2 434

1 垂直直方图 443
2 水平直方图 444
3 累积直方图 445
4 百分比直方图 446

1 Matlab PATH 设置 454

1 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) 475
2 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) 475
3 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) 476
4 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) . . . 476
5 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 . . 477
6 通过指定 **-Sq** 选项的参数控制标注属性. 478
7 另一个标注属性实例 478
8 标注的位置和内容来自不同的数据集 479
9 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。 480

保护环境, 从阅读电子文档开始!

表目录

1 地球科学常用绘图软件比较 25

1 GMT 预定义画笔宽度名 111
2 Word 字号与 GMT 中字号 (p) 对应关系 116
3 GMT 转义字符 122
4 欧洲特殊字母 122

1 GMT 支持的网格文件格式 145

1 GMT 标准选项列表 165
2 GMT 投影代码 170
3 Proj4 投影代码 171
4 GMT 时间单位 180

1 GMT 预定义纸张大小 253

1 通配符 462

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。手册详尽地介绍了 GMT 的用法并提供了大量的实用示例。其既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。

文档源码在 GitHub 上开源，欢迎更多的志愿者加入到文档的维护工作中来。

快速链接：

- 模块列表
- 配置参数
- 通用选项
- 投影方式
- 颜色名列表
- 字体列表
- 内置 *CPT*
- 特殊字符一览
- 地学数据
- 图库
- 引用 *GMT*
- 学习资源

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之一。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: 除 30 多种地图投影外, 还有笛卡尔线性坐标轴、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 年 1 月 4 日;
- 2019 年 11 月 1 日, GMT 6.x 的第一个正式版发布; 目前最新版本 GMT 6.0.0 发布于 2019 年 11 月 1 日。

想了解更多关于 GMT 的历史故事, 可以观看/收听下面的视频/音频:

- Don’t Panic Geocast 对 Paul Wessel 和 Leonardo Uieda 的采访 http://www.dontpanicgeocast.com/?p=638
- Don’t Panic Geocast 对 Walter Smith 的采访 https://www.dontpanicgeocast.com/?p=742
- Paul Wessel 在 GMT 20 周年的演讲 https://av.tib.eu/media/19869

1.1.3 GMT 开发者

GMT 的核心开发者有 7 位, 分别是 Paul Wessel、Walter H. F. Smith、Remko Scharroo、Joaquim F. Luis、Leonardo Uieda、Florian Wobbe 和 Dongdong Tian。GMT 的开发在 GitHub 上进行, 任何用户均可通过多种方式向 GMT 做贡献。

图 1: GMT 核心开发者及指导委员会部分成员

从左至右依次为 Dongdong Tian、David Sandwell（指导委员会主席）、Walter H.F. Smith、Paul Wessel、Joaquim Luis、Leonardo Uieda 和 Dave Caress（指导委员会成员）。照片拍摄于 2019 年 7 月 29 日至 8 月 2 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 是免费的开源软件，其源码遵循 GNU 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 源码由高度可移植的 ANSI C 语言写成，其完全兼容于 POSIX 标准，可以运行在 Linux、macOS 等类 UNIX 系统及 Windows 上。GMT 不仅公开了软件源代码，还提供了 Windows 和 macOS 下的二进制安装包，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，进而绘制复杂图件

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

和 EPS，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用于制图。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。

表 1: 地球科学常用绘图软件比较

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 版本号遵循 语义化版本号规范。GMT 版本号格式为 *major.minor.patch*，例如 **4.5.18**。其中 *major* 为主版本号，*minor* 为次版本号，*patch* 为补丁版本号。

根据语义化版本规范的要求：

- 当有极大更新，例如重写底层代码时，会增加主版本号 *major*。因而 *major* 不同的两个版本的 API 接口，以及语法、功能上可能有差异
- 当有较大更新，比如新增模块或者新增功能时，会增加次版本号 *minor*
- 若只是修复代码 BUG 或修复文档描述，则增加补丁版本号 *patch*

因而，GMT 6.x.x 与 5.x.x 在底层存在很大差异，两个版本的语法不完全兼容。GMT 5.4.x 相对于 5.3.x 增加了更多的功能，而 GMT 5.4.5 相对于 5.4.4 则主要是修复了一些 BUG。

1.2.2 GMT 主流版本

GMT 目前主流版本有 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

[3] 需要额外安装 M_Map

[4] 需要额外安装 ggmap

GMT4 GMT4 的最终版本为 4.5.18，发布于 2018 年 7 月。开发者不再对 GMT4 进行任何维护或更新，所有 BUG 将不会得到修复。

GMT5 GMT5 相对于 GMT4 有诸多改进，其命令语法更统一，选项设计更合理，还增加了很多新功能。其中，有用且常用的功能包括：

- **-Bafg** 自动确定坐标轴的标注、刻度和网格间隔
- 支持透明色，且支持透明图层
- **-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
- 使用 **-p** 可以绘制任意 3D 视角图

目前 GMT5 已进入维护期，不再增加新功能，仅修复已知 BUG。

GMT6 GMT6 是 GMT 目前的最新版本，也是开发者在着重维护和更新的版本。

GMT6 特点在于：

- 新增现代模式，极大简化了绘图脚本，并且避免了使用中的常见错误
- 兼容 GMT5 和 GMT4
- 新增模块
  - **movie** 模块用于方便地制作动画
  - **docs** 模块用于直接打开模块的网页文档
  - **subplot** 模块可以方便地绘制多子图
  - **inset** 模块则可以绘制小图

由于 GMT6 语法的简洁性，且其尽最大可能兼容 GMT4 和 GMT5 命令，不论是 GMT 新用户还是老用户，都建议学习并使用 GMT6 的现代模式。

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，可以考虑在文中引用 GMT 的如下文章来回报 GMT 开发者。

GMT6 用户: Wessel, P., Luis, J., Uieda, L., Scharroo, R., Wobbe, F., Smith, W. H. F., & Tian, D. (2019). The Generic Mapping Tools Version 6. *Geochemistry, Geophysics, Geosystems*, 20. https://doi.org/10.1029/2019GC008515

GMT5 用户: Wessel, P., Smith, W. H. F., Scharroo, R., Luis, J., & Wobbe, F. (2013). Generic Mapping Tools: Improved Version Released. *Eos, Transactions American Geophysical Union*, 94(45), 409–410. https://doi.org/10.1002/2013EO450001

GMT4/GMT3 用户: Wessel, P., & Smith, W. H. F. (1998). New, improved version of Generic Mapping Tools released. *Eos, Transactions American Geophysical Union*, 79(47), 579. http://dx.doi.org/10.1029/98EO00426

GMT3.0 用户: Wessel, P., & Smith, W. H. F. (1995). New version of the Generic Mapping Tools released. *Eos, Transactions American Geophysical Union*, 76(33), 329. http://dx.doi.org/10.1029/95EO00198

GMT2 用户: Wessel, P., & Smith, W. H. F. (1991). Free software helps map and display data, *Eos, Transactions American Geophysical Union*, 72(41), 445–446. http://dx.doi.org/10.1029/90EO00319

GMT/MATLAB 工具箱用户: Wessel, P., & Luis J. (2017). The GMT/MATLAB Toolbox. *Geochemistry, Geophysics, Geosystems*, 18, 811–823. http://dx.doi.org/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., & Wessel, P. (2008) Directional median filtering for regional-residual separation of bathymetry. *Geochemistry, Geophysics, Geosystems*, 9, Q03005. http://dx.doi.org/10.1029/2007GC001850

grdredpol 模块 Luis, J. F. & Miranda, J. M. (2008). Reevaluation of magnetic chrons in the North Atlantic between 35°N and 47°N: Implications for the formation of the Azores Triple Junction and associated plateau. *Journal of Geophysical Research*, 113, B10105. http://dx.doi.org/10.1029/2007JB005573

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

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

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

Wessel, P. & Becker J. M. (2008). Interpolation using a generalized Green' s function for a spherical surface spline in tension. *Geophysical Journal International*, 174, 21–28. http://dx.doi.org/10.1111/j.1365-246X.2008.03829.x

1.4 学习资源

这一节列出网络上可免费获取的 GMT 学习资源。各个学习资源的质量以及维护程度不同，**建议用户以英文官方文档以及中文社区参考手册为主，其余学习资源仅供参考**。

入门教程

1. *《GMT 中文手册》中的入门教程*
2. GMT 官方入门教程
3. 《地学 GMT 绘图》系列视频教程（作者: @liuzhumei）
   - 《GMT5 入门教程》
   - 《GMT6 新特性》
4. GMT Tutorial 作者: 鄭懷F (适用于 GMT5)

参考手册

1. GMT 中文手册
2. GMT 官方参考手册
3. GMT 官方模块手册

图库与示例

1. *《GMT 中文手册》中的示例*
2. GMT 官方示例
3. GMT 中文社区示例
4. GMT 中文社区博客

论坛/讨论组

1. 地学 GMT 中文社区 QQ 群：1 群（218905582，已满）；2 群（791856541）
2. GMT 官方论坛
3. GMT Gitter 聊天室

第 2 章　安装

2.1 Linux 下安装 GMT

尽管大多数 Linux 发行版都提供了 GMT 二进制包，可以直接通过软件包管理器 `apt-get` 或 `yum` 安装。但发行版提供的 GMT 版本都很老，不建议使用。

针对 Fedora/RHEL/CentOS 用户，GMT 的官方 RPM 仓库提供了最新版本的 GMT。

2.1.1 Fedora

Fedora 用户可以启用 GMT 官方 RPM 仓库 以安装 GMT 最新版本:

```
# 启用 GMT 官方 RPM 仓库
dnf copr enable genericmappingtools/gmt

# 安装最新版 GMT
dnf install gmt

# 当有新版本发布时可直接更新
dnf update gmt
```

除此之外，还可以安装如下可选包以使用 GMT 的更多功能:

```
# 地理数据格式转换工具
dnf install gdal

# 制作 GIF 格式动画需要 GraphicsMagick
dnf install GraphicsMagick

# 制作 MP4、WebM 格式动画需要 ffmpeg
dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).
↪noarch.rpm
dnf install ffmpeg
```

注解: 如果你已经安装了 Fedora 系统仓库提供的 GMT 软件包，你必须在使用 GMT 官方仓库前卸载旧的 GMT 安装包。使用如下命令:

```
dnf uninstall GMT dcw-gmt gshhg-gmt-nc4 gshhg-gmt-nc4-full gshhg-gmt-nc4-high
```

2.1.2 RHEL/CentOS

RHEL/CentOS 用户可以启用 GMT 官方 RPM 仓库 以安装 GMT 最新版本:

```
# 安装 epel-release
yum install epel-release

# 启用 GMT 官方仓库 (仅限于 RHEL/CentOS 7 用户)
yum install yum-plugin-copr
yum copr enable genericmappingtools/gmt

# 启用 GMT 官方仓库 (仅限于 RHEL/CentOS 6 用户)
wget https://copr.fedorainfracloud.org/coprs/genericmappingtools/gmt/repo/epel-6/
↪genericmappingtools-gmt-epel-6.repo -O /etc/yum.repos.d/genericmappingtools-gmt-epel-6.repo

# 安装最新版 GMT
yum install gmt

# 当有新版本发布时可直接更新
yum update gmt
```

除此之外, 还可以安装如下可选包以使用 GMT 的更多功能:

```
# 地理数据格式转换工具
yum install gdal

# 制作 GIF 格式动画需要 GraphicsMagick
yum install GraphicsMagick

# 制作 MP4、WebM 格式动画需要 ffmpeg
yum localinstall --nogpgcheck https://download1.rpmfusion.org/free/el/rpmfusion-free-release-$(rpm -
↪E %rhel).noarch.rpm
yum install ffmpeg
```

注解: 如果你已经安装了 EPEL 仓库提供的 GMT 软件包, 你必须在使用 GMT 官方仓库前卸载旧的 GMT 安装包。使用如下命令:

```
yum uninstall GMT dcw-gmt gshhg-gmt-nc4 gshhg-gmt-nc4-full gshhg-gmt-nc4-high
```

2.1.3 Ubuntu/Debian 用户

Ubuntu/Debian 用户目前只能通过编译源码的方式安装 GMT 最新版。具体编译方法见*Linux/macOS 下编译 GMT 源码*。

2.2 Windows 下安装 GMT

GMT 提供了 Windows 下的安装包, 可以直接安装使用。

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

GMT 6.0.0 安装包中不仅包含了 GMT，还包含了运行 GMT 所需的如下软件：

- GDAL：用于多种地学数据格式的转换
- FFmpeg：用于生成 MP4 格式的动画
- Ghostscript：用于生成 PDF、JPG 等图片格式

2.2.1 安装 GMT

1. 下载 GMT 6.0.0 安装包

   - gmt-6.0.0-win64.exe (64 位)
   - gmt-6.0.0-win32.exe (32 位)

2. 安装 GMT

   双击安装包即可安装。在 "Choose components" 页面，建议将所有选项都勾选上。

   注解： GMT 安装包中内置了 Ghostscript 最新版，但其不支持中文。若想要 GMT 支持中文，则需要在安装 GMT 时不选择 Ghostscript 组件，待 GMT 安装完成后再自行安装 Ghostscript。

   注解： 安装过程中可能会出现如下警告:

   ```
   Warning! Failed to add GMT to PATH. Please add the GMT bin path to PATH manually.
   ```

   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手动修改系统环境变量 **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 bin 目录的路径，默认值为 `C:\programs\gmt6\bin`。需要注意 "path" 变量值中多个路径之间用英文分号分隔

3. 测试安装是否正确

   安装完成后，点击 "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
6.0.0
```

即表示安装成功。

4. 卸载 GMT

   若想要卸载 GMT，可以进入 GMT 安装目录，找到并双击执行 `Uninstall.exe` 即可完成卸载。偶尔会遇到卸载不干净的情况，可以等卸载程序执行完成后再手动删除 GMT 安装目录即可。

5. 升级 GMT

   GMT 目前不具备自动更新功能。如果想要升级新版本，通常需要先卸载旧版本。卸载完成后，再下载并安装新版安装包以完成升级。

2.2.2 安装其它可选软件

为了更好地使用 GMT，你还可以根据自己的需求安装如下软件。

1. 安装 Git for Windows (**推荐**)

   Git for Windows 为 Windows 用户提供了 Bash 以及 Linux 下常用的多个命令。如果想要在 Windows 下运行 Bash 脚本，推荐安装 Git for Windows。

   下载地址: https://git-scm.com/download/win

2. 安装 GraphicsMagick (**根据需求选择是否安装**)

   GMT 的 **movie** 模块在制作 GIF 格式的动画时需要使用 GraphicsMagick。

   如有制作 GIF 动画的需求，可以下载安装这个软件，并将其 bin 目录加入到系统环境变量 **PATH** 中，以保证 GMT 可以找到其提供的 **gm** 命令。

3. 安装 Ghostscript 9.50 (**根据是否需要中文支持决定是否安装**)

   GMT 需要使用 Ghostscript 生成 PDF、JPG 等格式的图片，因而 Ghostscript 是必须的。GMT 安装包中自带了 Ghostscript，但是其并不支持在 GMT 图片中添加中文。

   如果有在 GMT 图片中添加中文的需求，则需要在安装 GMT 时不安装 Ghostscript 组件，然后自己再自行安装 Ghostscript。安装过程中记得勾选 `Generate cidfmap for Windows CJK TrueType fonts`。

   安装包下载地址:

   - gs950w64.exe (64 位)
   - gs950w32.exe (32 位)

> **警告：** ghostscript 9.27 存在严重 bug，会导致生成的图片中有用信息被裁剪。请确保自己安装的不是 9.27 版本。

4. 安装 GSview 5.0 (**不推荐**)

   GSview 是一个 PostScript 阅读器。GMT6 默认生成 PDF 格式的图片，因而无需安装 GSview。如果坚持想要生成并查看 PS 格式的图片，则可以安装 GSview。

   - gsv50w64.exe (64 位)
   - gsv50w32.exe (32 位)

5. 安装 UnixTools (**不推荐**)

   如果想要在 Windows 下运行 Batch 脚本，但同时想要使用各种 Linux 下的常用命令，则可以使用 GMT 中文社区整理的 Unix 小工具合集包 UnixTools。

   直接下载并解压，将解压得到的 exe 文件移动到 GMT 的 bin 目录即可。

   下载地址: UnixTools.zip

2.3 macOS 下安装 GMT

macOS 下可以直接使用 GMT 提供的安装包，也可以使用 Homebrew 或 Macports 软件管理工具进行安装。**推荐使用 Homebrew**。

2.3.1 使用 Homebrew 安装

Homebrew 是 macOS 下的第三方软件包管理工具。未安装 Homebrew 的用户，可以访问其 官网 以了解如何安装与使用。

1. 安装 GMT:

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

2. 安装 GMT 依赖的其它软件:

```
# 必须软件包
$ brew install ghostscript

# 安装生成动画所需要的软件包(可选)
$ brew install graphicsmagick ffmpeg
```

3. 重新打开一个终端，检测安装是否成功:

```
$ gmt --version
6.0.0
```

4. 升级 GMT。当有新版本发布时，可以执行如下命令升级 GMT:

```
brew upgrade gmt
```

5. 卸载 GMT。执行如下命令即可卸载 GMT:

```
brew uninstall gmt
```

2.3.2 使用 GMT 安装包

GMT 为 macOS 用户提供了 dmg 安装包，其不仅包含了 GMT，还包含了运行 GMT 所需的 Ghostscript、GDAL、GraphicsMagick 和 FFmpeg，可以直接双击安装使用。

1. 下载：gmt-6.0.0-darwin-x86_64.dmg
2. 双击 dmg 包，在弹出的 Finder 窗口中，将 **GMT-6.0.0.app** 拖动到 Applications 目录
3. 双击应用中的 GMT 图标以启动 GMT。GMT 会启动一个终端并显示欢迎信息。根据欢迎信息中的提示将如下语句添加到 `~/.bash_profile` 中以修改 PATH 环境变量:

```
export GMTHOME=/Applications/GMT-6.0.0.app/Contents/Resources
export PATH=${GMTHOME}/bin:${PATH}
export PROJ_LIB=$GMTHOME/share/proj6
export MAGICK_CONFIGURE_PATH=$GMTHOME/lib/GraphicsMagick-1.3.33/config
```

注解: 以上内容仅供参考，请务必根据 GMT 欢迎信息中的提示修改环境变量。

4. 打开一个终端，输入如下命令，检测安装是否成功:

```
$ gmt --version
6.0.0
```

5. 卸载 GMT

   若想要卸载 GMT，可直接到 `/Application` 目录下找到 GMT，直接删除即可。
6. 升级 GMT

   GMT 包不支持自动升级，因而要先删除旧 GMT 包，再下载新版安装包并按照上面的步骤重新安装，即实现升级 GMT。

2.3.3 使用 Macports 安装

Macports 是 macOS 下的第三方软件包管理工具。

1. 安装 GMT:

```
$ sudo port install gdal +hdf5 +netcdf +openjpeg
$ sudo port install gmt6
```

2. 安装 GMT 依赖的其他软件:

```
$ sudo port install graphicsmagick ffmpeg
```

3. 重新打开一个终端，检测安装是否成功:

```
$ gmt --version
6.0.0
```

4. 升级 GMT。当有新版本发布时，可以执行如下命令升级 GMT:

```
$ sudo port selfupdate
$ sudo port upgrade gmt6
```

5. 卸载 GMT。执行如下命令即可卸载 GMT:

```
$ sudo port uninstall gmt6
```

2.4 跨平台 GMT 安装方案

跨平台安装方案是指，以下安装方式同时适用于 Linux、macOS 和 Windows。

2.4.1 通过 conda 安装

conda 是由 Anaconda 提供的一个跨平台软件包管理器。conda 的 conda-forge 通道提供了最新的 GMT。

如果你是 Anaconda 用户，则可以直接通过如下命令安装。

安装 GMT 6.0.0:

```
conda install gmt -c conda-forge
```

注意：由于 conda-forge 没有提供 Windows 下的 GraphicsMagick，因而 Windows 用户还需自行下载并安装 GraphicsmMgick 才能制作 GIF 格式的动画。

安装完成后，在终端执行如下命令以验证:

```
$ gmt --version
6.0.0
```

2.5 Linux/macOS 下编译 GMT 源码

这一节介绍如何在 Linux 或者 macOS 下编译 GMT 源代码。Windows 用户如果想编译 GMT 源码请参考 GMT 官方 编译指南。

2.5.1 编译及运行依赖

GMT 的编译及运行需要如下软件：

- CMake: >=2.8.5
- Ghostscript

- netCDF (>4.0 且支持 netCDF-4/HDF5)
- curl

除此之外, 还可以安装如下软件库以增强 GMT 的更多功能:

- GDAL: 读写其他地学常用的网格和图片格式
- PCRE: 正则表达式支持
- FFTW: 快速傅里叶变换库 (>=3.3, macOS 下不需要)
- GLib: GTHREAD 多线程支持
- LAPACK: 快速矩阵反演库 (macOS 下不需要)
- BLAS: 快速矩阵运算库 (macOS 下不需要)
- GraphicsMagick: 生成 GIF 格式的动画
- FFmpeg: 生成 MP4 格式的动画

2.5.2 安装依赖软件

对于 Ubuntu/Debian:

```
# 安装编译所需软件包
$ sudo apt-get install build-essential cmake libcurl4-gnutls-dev libnetcdf-dev ghostscript
# 安装可选软件包
$ sudo apt install gdal-bin libgdal-dev liblapack3 libglib2.0-dev libpcre3-dev libfftw3-single3␣
↪liblapack-dev
# 安装制作动画所需的软件包
$ sudo apt install graphicsmagick ffmpeg
```

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 cmake make glibc ghostscript netcdf-devel libcurl-devel
# 安装可选软件包
$ sudo yum install gdal gdal-devel lapack-devel openblas-devel glib2-devel pcre-devel fftw-devel
# 安装其他可选包
$ sudo yum localinstall --nogpgcheck https://download1.rpmfusion.org/free/el/rpmfusion-free-release-
↪$(rpm -E %rhel).noarch.rpm
$ sudo yum install GraphicsMagick ffmpeg
```

对于 macOS 用户, 建议使用 Homebrew 安装依赖:

```
# 安装必须依赖
brew install cmake curl netcdf ghostscript
# 安装可选依赖
brew install gdal pcre2 glib fftw graphicsmagick ffmpeg
```

警告: GMT 需要使用 ghostscript 生成 PDF、JPG 等格式的图片。但 ghostscript 9.27 存在严重 bug, 会导致生成的图片中有用信息被裁剪。请使用 `gs --version` 确

认安装的 ghostscript 不是 9.27 版本。

2.5.3 下载源码及数据

编译 GMT 需要下载如下三个文件：

1. GMT 6.0.0 源码：gmt-6.0.0-src.tar.gz
2. 全球海岸线数据 GSHHG：gshhg-gmt-2.3.7.tar.gz
3. 全球数字图表 DCW：dcw-gmt-1.1.4.tar.gz

2.5.4 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-6.0.0.tar.gz
$ tar -xvf gshhg-gmt-2.3.7.tar.gz
$ tar -xvf dcw-gmt-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-6.0.0/share/gshhg
$ mv dcw-gmt-1.1.4 gmt-6.0.0/share/dcw-gmt

# 切换到 gmt 源码目录下
$ cd gmt-6.0.0

# 用文本编辑器新建并打开 CMake 用户配置文件
# Linux 用户
$ gedit cmake/ConfigUser.cmake
# macOS 用户
$ open -a TextEdit cmake/ConfigUser.cmake
```

向 `cmake/ConfigUser.cmake` 文件中加入如下语句:

```
set (CMAKE_INSTALL_PREFIX "/opt/GMT-6.0.0")
set (COPY_GSHHG TRUE)
set (COPY_DCW TRUE)
set (GMT_INSTALL_MODULE_LINKS FALSE)
```

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-6.0.0` 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-6.0.0` 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` 建议设置为 `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。用户可以将 GMT 提供的配置模板 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-6.0.0 目录下，不是 gmt-6.0.0/cmake 目录下
$ mkdir build
$ cd build/
$ cmake ..
```

cmake .. 会检查系统软件是否满足 GMT 的依赖关系，过程中会输出大量信息，并在最后汇总输出检查结果。我们只需要关注检查结果是否正确即可。正常情况下结果结果如下，若存在一些差异也没有问题。只要过程中不出现报错，即可。如果出现报错，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:

```
*
*  GMT Version:                : 6.0.0
*
*  Options:
*  Found GSHHG database        : /home/user/GMT/gmt-6.0.0/share/gshhg (2.3.7)
*  Found DCW-GMT database      : /home/user/GMT/gmt-6.0.0/share/dcw-gmt
*  Found GMT data server       : https://oceania.generic-mapping-tools.org
*  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
*  LAPACK library              : yes
*  BLAS library                : yes
*  License restriction         : no
*  Triangulation method        : Shewchuk
*  OpenMP support              : enabled
*  GLIB GTHREAD 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
*  Found Ghostscript (gs)      : yes (9.50)
*  Found GraphicsMagick (gm)   : yes (1.3.33)
*  Found ffmpeg                : yes (4.2.1)
```

(下页继续)

(续上页)

```
*  Found open                  : yes
*  Found ogr2ogr               : yes (2.4.2)
*  Found gdal_translate        : yes (2.4.2)
*
*  Locations:
*  Installing GMT in           : /opt/GMT-6.0.0
*  GMT_DATADIR                 : /opt/GMT-6.0.0/share
*  GMT_DOCDIR                  : /opt/GMT-6.0.0/share/doc
*  GMT_MANDIR                  : /opt/GMT-6.0.0/share/man
-- Configuring done
-- Generating done
```

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。但据用户报告，某些 Ubuntu 发行版下使用 `-j` 选项会导致编译过程卡死。若出现此种情况，建议去除 `-j` 选项。

2.5.5 修改环境变量

打开终端，使用如下命令用文件编辑器打开 Bash 配置文件 `~/.bashrc`:

```
# Linux 用户
gedit ~/.bashrc

# macOS 用户
open ~/.bash_profile
```

然后向文件末尾加入如下语句以修改环境变量。修改完成后保存文件并退出，然后重启终端使其生效:

```
export GMT6HOME=/opt/GMT-6.0.0
export PATH=${GMT6HOME}/bin:$PATH
```

(下页继续)

(续上页)

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${GMT6HOME}/lib64
```

说明:

- 第一个命令添加了环境变量 GMT6HOME
- 第二个命令修改 GMT6 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 GMT6 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

2.5.6 测试是否安装成功

重新打开一个终端, 键入如下命令, 若正确显示 GMT 版本号, 则表示安装成功:

```
$ gmt --version
6.0.0
```

2.5.7 升级/卸载 GMT

按照上面的配置, GMT 会被安装到 /opt/GMT-6.0.0 目录下。若想要卸载 GMT, 可以直接删除整个 /opt/GMT-6.0.0 即可。

GMT 不支持自动更新, 因而若想要升级 GMT, 通常建议先卸载 GMT, 然后再下载新版源码并按照上面的步骤重新编译安装。

当然, 高级用户也可以同时安装多个版本的 GMT, 但需要注意环境变量 PATH 的设置。

第 3 章　入门教程

3.1 GMT 初探

这一节将介绍如何开始使用 GMT。本节分为三个小节，分别针对 Linux、macOS 和 Windows 系统。用户应根据自己当前的操作系统阅读相应的小节。

3.1.1 GMT 初探: Linux 篇

启动终端

GMT 是一个纯命令行软件，没有任何的图形界面。所有的绘图操作都需要通过在终端和脚本中执行命令来完成。终端是 Linux 系统的标配，通常你可以在系统的 “应用程序” 中找到并启动它。

运行 GMT

启动终端后，敲入 `gmt` 以执行 GMT 命令。你将看到 GMT 的欢迎界面信息，类似于:

```
    GMT - The Generic Mapping Tools, Version 6.0.0 [64-bit] [8 cores]
    (c) 1991-2019 The GMT Team (https://www.generic-mapping-tools.org/team.html).

Supported in part by the US National Science Foundation (http://www.nsf.gov/)
and volunteers from around the world.

GMT is distributed under the GNU LGP License (http://www.gnu.org/licenses/lgpl.html).

usage: gmt [options]
    gmt <module name> [<module-options>]

options:
--help            List descriptions of available GMT modules.
--new-script      Write GMT modern mode script template to stdout.
--show-bindir     Show directory with GMT executables.
--show-citation   Show the most recent citation for GMT.
--show-cores      Show number of available cores.
--show-datadir    Show directory/ies with user data.
--show-dataserver Show URL of the remote GMT data server.
--show-doi        Show the DOI for the current release.
--show-modules    Show all module names.
--show-library    Show path of the shared GMT library.
--show-plugindir  Show directory for plug-ins.
--show-sharedir   Show directory for shared GMT resources.
--version         Print GMT version number.

if <module-options> is '=' we call exit (0) if module exist and non-zero otherwise.
```

生成脚本模板

继续在终端中敲入:

```
gmt --new-script > myplot.sh
```

该命令会生成一个 GMT 模板脚本，并保存到 Bash 脚本文件 `myplot.sh` 中。

注解: 本教程所有示例均使用 Bash 脚本，要求读者对 Bash 脚本及 Unix 命令行有最基本的了解。不了解的用户请阅读网络上 Bash 相关教程，或本手册中附录部分命令行及脚本基础。

查看并编辑脚本文件

Bash 脚本文件是一个纯文本文件，可以直接用文本编辑器打开。比如，可以使用大多数 Linux 都自带了的文本编辑器 **gedit** 打开该脚本文件:

```
gedit myplot.sh
```

打开脚本文件后会看到如下内容:

```
#!/usr/bin/env bash
# GMT modern mode bash template
# Date:    2019-09-10T00:44:39
# User:    seisman
# Purpose: Purpose of this script
export GMT_SESSION_NAME=$$  # Set a unique session name
gmt begin figurename
    # Place modern session commands here
gmt end show
```

其中，以 # 开头的行尾注释行，**export GMT_SESSION_NAME=$$** 这一行属于高级用法，可以忽略。因而核心内容只有两行，即 **gmt begin** 和 **gmt end** 这两行。

编辑脚本，在 **gmt begin** 和 **gmt end** 中间添加 GMT 命令，将脚本修改如下:

```
#!/usr/bin/env -S bash -e
# GMT modern mode bash template
# Date:    2019-09-10T00:44:39
# User:    seisman
# Purpose: Purpose of this script
export GMT_SESSION_NAME=$$  # Set a unique session name
gmt begin figurename
    gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot"
gmt end show
```

编辑完成后记得保存文件。

执行脚本以绘图

回到终端，运行 Bash 脚本:

```
bash myplot.sh
```

待脚本执行完成后，会自动用阅读器（通常是 evince）打开生成的 PDF 格式的图片文件。你将看到如下图所示的图片。

Source Code

这基本上就是运行 GMT 脚本的基本流程，即:

- 生成脚本模板
- 编辑脚本，添加 GMT 绘图命令
- 运行脚本并查看绘图效果

3.1.2 GMT 初探: macOS 篇

启动终端

GMT 是一个纯命令行软件，没有任何的图形界面。所有的绘图操作都需要通过在终端和脚本中执行命令来完成。

macOS 下可以使用 `Command+` 空格键启动 Splitlight，在弹出的搜索框中输入 "terminal" 再按回车即可打开终端。

运行 GMT

启动终端后，敲入 `gmt` 以执行 GMT 命令。你将看到 GMT 的欢迎界面信息，类似于:

```
    GMT - The Generic Mapping Tools, Version 6.0.0 [64-bit] [8 cores]
    (c) 1991-2019 The GMT Team (https://www.generic-mapping-tools.org/team.html).

Supported in part by the US National Science Foundation (http://www.nsf.gov/)
```

（下页继续）

(续上页)

```
and volunteers from around the world.

GMT is distributed under the GNU LGP License (http://www.gnu.org/licenses/lgpl.html).

usage: gmt [options]
    gmt <module name> [<module-options>]

options:
--help            List descriptions of available GMT modules.
--new-script      Write GMT modern mode script template to stdout.
--show-bindir     Show directory with GMT executables.
--show-citation   Show the most recent citation for GMT.
--show-cores      Show number of available cores.
--show-datadir    Show directory/ies with user data.
--show-dataserver Show URL of the remote GMT data server.
--show-doi        Show the DOI for the current release.
--show-modules    Show all module names.
--show-library    Show path of the shared GMT library.
--show-plugindir  Show directory for plug-ins.
--show-sharedir   Show directory for shared GMT resources.
--version         Print GMT version number.

if <module-options> is '=' we call exit (0) if module exist and non-zero otherwise.
```

生成脚本模板

继续在终端中敲入:

```
gmt --new-script > myplot.sh
```

该命令会生成一个 GMT 模板脚本，并保存到 Bash 脚本文件 `myplot.sh` 中。

注解: 本教程所有示例均使用 Bash 脚本，要求读者对 Bash 脚本及 Unix 命令行有最基本的了解。不了解的用户请阅读网络上 Bash 相关教程，或本手册中附录部分命令行及脚本基础。

查看并编辑脚本文件

Bash 脚本文件是一个纯文本文件，可以直接用文本编辑器打开。比如，可以使用 macOS 自动的文本编辑器打开该脚本文件:

```
open myplot.sh
```

打开脚本文件后会看到如下内容:

```
#!/usr/bin/env bash
# GMT modern mode bash template
```

(下页继续)

(续上页)

```
# Date:    2019-09-10T00:44:39
# User:    seisman
# Purpose: Purpose of this script
export GMT_SESSION_NAME=$$  # Set a unique session name
gmt begin figurename
    # Place modern session commands here
gmt end show
```

其中，以 # 开头的行尾注释行，**export GMT_SESSION_NAME=$$** 这一行属于高级用法，可以忽略。核心内容只有两行，即 **gmt begin** 和 **gmt end** 这两行。

编辑脚本，在 **gmt begin** 和 **gmt end** 中间添加 GMT 命令，将脚本修改如下:

```
#!/usr/bin/env -S bash -e
# GMT modern mode bash template
# Date:    2019-09-10T00:44:39
# User:    seisman
# Purpose: Purpose of this script
export GMT_SESSION_NAME=$$  # Set a unique session name
gmt begin figurename
    gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot"
gmt end show
```

编辑完成后记得保存文件。

执行脚本以绘图

回到终端，运行 Bash 脚本:

```
bash myplot.sh
```

待脚本执行完成后，会自动用阅读器（通常是 evince）打开生成的 PDF 格式的图片文件。你将看到如下图所示的图片。

Source Code

这基本上就是运行 GMT 脚本的基本流程，即：

- 生成脚本模板
- 编辑脚本，添加 GMT 绘图命令
- 运行脚本并查看绘图效果

3.1.3 GMT 初探：Windows 篇

启动 CMD 命令行

GMT 是一个纯命令行软件，没有任何的图形界面。所有的绘图操作都需要通过在终端和脚本中执行命令来完成。Windows 下的默认终端是“命令提示符”，也就是 CMD 命令行。

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。

运行 GMT

启动 CMD 后，敲入 `gmt` 以执行 GMT 命令。你将看到 GMT 的欢迎界面信息，类似于：

```
    GMT - The Generic Mapping Tools, Version 6.0.0 [64-bit] [8 cores]
    (c) 1991-2019 The GMT Team (https://www.generic-mapping-tools.org/team.html).

Supported in part by the US National Science Foundation (http://www.nsf.gov/)
and volunteers from around the world.

GMT is distributed under the GNU LGP License (http://www.gnu.org/licenses/lgpl.html).

usage: gmt [options]
    gmt <module name> [<module-options>]

options:
--help            List descriptions of available GMT modules.
--new-script      Write GMT modern mode script template to stdout.
--show-bindir     Show directory with GMT executables.
--show-citation   Show the most recent citation for GMT.
--show-cores      Show number of available cores.
--show-datadir    Show directory/ies with user data.
--show-dataserver Show URL of the remote GMT data server.
--show-doi        Show the DOI for the current release.
--show-modules    Show all module names.
--show-library    Show path of the shared GMT library.
--show-plugindir  Show directory for plug-ins.
--show-sharedir   Show directory for shared GMT resources.
--version         Print GMT version number.

if <module-options> is '=' we call exit (0) if module exist and non-zero otherwise.
```

生成脚本模板

继续在终端中敲入：

```
gmt --new-script > myplot.bat
```

该命令会生成一个 GMT 模板脚本，并保存到 Batch 脚本文件 `myplot.bat` 中。

注解: Batch 是 Windows 自带的脚本语言，但本教程中所有示例均使用 Unix 下常用的 Bash 脚本。因而 Windows 用户有两种选择：

1. 安装 Git for Windows 并使用其提供的 Bash，本手册中的所有命令都将可以直接使用。要求读者对 Bash 脚本及 Unix 命令行有最基本的了解。不了解的用户请阅读网络上 Bash 相关教程，或本手册中附录部分命令行及脚本基础。
2. 继续使用 Windows 的 Batch 脚本。要求读者对 Batch 脚本和 Bash 脚本均有所了解，并知道二者用法的差异，以便于将手册中的 Bash 脚本转换为 Batch 脚本。不了解的用户请阅读网络上 Bash 和 Batch 相关教程，或本手册中附录部分命令行及脚本基础。

查看并编辑脚本文件

Batch 脚本文件是一个纯文本文件，可以直接用文本编辑器打开。例如，Windows 下自带的记事本即可打开该脚本文件。

打开脚本文件后会看到如下内容:

```
REM GMT modern mode batch template
REM Date:    2019-09-02T23:34:25
REM User:    seisman
REM Purpose: Purpose of this script
REM Set a unique session name:
@echo off
set GMT_SESSION_NAME=7492
gmt begin figurename
    REM Place modern session commands here
gmt end show
```

其中，以 **REM** 开头的行尾注释行，**set GMT_SESSION_NAME=7492** 这一行属于高级用法，可以忽略。核心内容只有两行，即 **gmt begin** 和 **gmt end** 这两行。

编辑脚本，在 **gmt begin** 和 **gmt end** 中间添加 GMT 命令，将脚本修改如下:

```
REM GMT modern mode batch template
REM Date:    2019-09-02T23:34:25
REM User:    seisman
REM Purpose: Purpose of this script
REM Set a unique session name:
@echo off
set GMT_SESSION_NAME=7492
gmt begin figurename
```

(下页继续)

(续上页)

```
    gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot"
gmt end show
```

编辑完成后记得保存文件。

执行脚本以绘图

回到 CMD，直接输入 Batch 脚本名以运行该脚本:

```
myplot.bat
```

待脚本执行完成后，会自动用 PDF 阅读器打开生成的 PDF 格式的图片文件。你将看到如下图所示的图片。

Source Code

这基本上就是运行 GMT 脚本的基本流程，即:

- 生成脚本模板
- 编辑脚本，添加 GMT 绘图命令
- 运行脚本并查看绘图效果

3.2 命令初探

上一节我们使用 GMT 绘制了第一张图，但是没有对其中的细节做任何介绍。这一节我们将简单介绍一下 GMT 的命令。

3.2.1 GMT 绘图脚本模板

GMT 在绘图时，总是以*begin* 开始，并以*end* 结束。所有的**绘图命令**都放在 **begin** 与 **end** 之间，而非绘图命令可以放在任何地方。

注解: GMT 老用户可能会对此比较陌生。

GMT 自 6.0.0 版本开始，引入了一种全新的绘图命令执行模式，称之为现代模式。GMT5 及之前的命令风格称之为经典模式。GMT6 既支持经典模式也支持现代模式。现代模式与经典模式可以完成相同的绘图功能，但现代模式大大简化了绘图代码，并极大避免了用户最容易出错的地方，因而推荐用户使用现代模式而非经典模式。

本手册将只使用现代模式，而不介绍经典模式。GMT 新用户只需要按照本手册直接学习现代模式即可；对于 GMT 老用户，建议阅读现代模式与经典模式的差异 或者观看视频教程 GMT6 新特性，并开始使用现代模式进行绘图。

在前一节中我们使用 `gmt --new-script` 命令生成一个绘图模板。忽略掉绘图模板中的注释语句，一个最最基本的 GMT 绘图脚本的模板是:

```
gmt begin
#
# 其它命令，包括 GMT 绘图命令、数据处理命令以及其它 UNIX 命令
#
gmt end
```

你可以直接执行这个脚本，但是因为这里什么命令也没用，所以这个脚本不会生成任何图片。

绘图时我们通常需要指定图片文件名和图片格式，因而更实用的绘图脚本模板为:

```
gmt begin FigureName pdf
#
# 其它命令，包括 GMT 绘图命令、数据处理命令以及其它 UNIX 命令
#
gmt end show
```

- *FigureName* 指定了要生成的图片文件名，你可以指定任意文件名，但最好避免在文件名中使用特殊符号和空格。若不给定文件名，则默认文件名为 `gmtsession`
- 紧跟在图片文件名后的 **pdf** 指定了要生成的图片格式（若不指定格式，则默认图片格式为 PDF）。GMT 支持多种图片格式，pdf、ps、eps、jpg、png、bmp 等等。若想要一次性生成多种格式的图片，则可以使用逗号将多种格式连接起来，如 **pdf,png** 会同时生成 PDF 和 PNG 格式的图片
- **gmt end** 后面加上 **show**，则 GMT 会在绘图完成后，使用系统自带的阅读器软件自动打开生成的图片文件，供用户预览绘图效果

3.2.2 绘制第一张图

在 GMT 绘图脚本模板的基础上，向 **begin** 和 **end** 语句中间加入正确的 GMT 绘图命令，即可实现用 GMT 绘图。

下面的脚本使用*coast* 绘制了一张全球地图。执行该脚本，会生成文件名为 `GlobalMap`、格式为 PNG 和 PDF 的图片文件，并且 GMT 会在绘图结束后自动打开生成的图片文件。

```
gmt begin GlobalMap png,pdf
    gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot"
gmt end show
```

图 1: 使用 GMT 绘制的第一张图

小技巧: 不理解 **gmt coast** 这一行命令的含义？没关系，稍后会详细介绍。

小技巧: 想了解 **gmt coast** 的具体用法？打开终端，敲入 `gmt docs coast`，GMT 会自动帮你打开 **coast** 模块的说明文档。

3.2.3 GMT 命令格式

一个 GMT 命令通常由 **gmt** + **模块名** + **选项** + **参数**构成。比如上面的例子中:

```
gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot"
```

- 所有的 GMT 命令都需要以 **gmt** 开头
- **coast** 是模块名，这个模块可以用于绘制海岸线
- **-R**、**-J**、**-G**、**-B** 等以 **-** 开头的是模块的选项
- `-Baf` 中 `af` 是 `-B` 选项的参数
- `-B+t"My First Plot"` 中 **+t** 为 **-B** 的子选项，`"My First Plot"` 则是子选项 **+t** 的参数

关于 GMT 命令的几点说明:

- 若模块名以 **gmt** 开头，则模块名中的 **gmt** 可省略。比如 `gmt gmtset xxx xxx` 可简写为 `gmt set xxx xxx`
- 模块名、选项等均区分大小写
- 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
- 子选项以 + 开头，后接**单个字符**以及子选项的参数
- 不以 - 开头的参数，通常都会被当做文件，GMT 会尝试去读取
- 各选项间以空格分隔，选项内部不能有空格。选项内部的字符串中若存在空格，在 Bash 下可以使用单引号或双引号括起来，而在 Windows 的 Batch 下则只能使用双引号。

3.3 绘制底图

GMT 可以绘制多种不同类型的底图，包括全球地图、区域地图，以及线性坐标轴、对数轴、指数轴、极坐标轴，甚至 3D 坐标轴。

这一节我们将通过使用*coast* 和*basemap* 模块绘制不同类型的底图，并了解 GMT 中 **-J**、**-R** 和 **-B** 选项的基本用法。

3.3.1 全球地图

要绘制地图，就需要将地球的三维球面投影到一个二维面上，投影的过程需要指定投影方式。GMT 中使用 **-J** 选项指定地图投影参数以及地图的尺寸。同时，我们还需要使用 **-R** 选项指定要绘制的区域范围（即经纬度范围）。

下面用最简单的命令绘制了一张全球地图。

```
gmt begin GlobalMap png,pdf
    gmt coast -JH180/12c -R0/360/-90/90 -W0.5p -A10000
gmt end show
```

图 2: 全球地图（无边框版）

先忽略后面的 **-W0.5p -A10000**（后面章节会介绍到），这里只关注 **-J** 和 **-R** 选项。

此示例中：

1. **-JH180/12c** 指定地图投影参数，**H** 表示使用 Hammer 投影，地图中心位于经度 180° 处，地图宽度为 12 厘米（**12c**，**c** 表示单位厘米）；
2. **-R0/360/-90/90** 指定要绘制的区域范围，即经度 0° 到 360°，纬度-90° 到 90°，四个数字之间用斜杠 / 分隔。

上面使用了非常简单的 GMT 命令即绘制出了一张全球地图的底图，但其跟我们常见的全球地图比还缺了个底图边框。GMT 中可以使用 **-B** 选项为地理底图加上边框并绘制经纬线。

```
gmt begin GlobalMap png,pdf
    gmt coast -JH180/12c -Rg -Bg -W0.5p -A10000
gmt end show
```

图 3: 全球地图（有边框版）

跟上一个脚本相比，此处我们做了两点修改：

1. 使用 **-Rg** 代替了 **-R0/360/-90/90**。这二者是完全等效的。由于绘制全球地图是很常见的需求，因而 GMT 为其设计了一个更简单的写法。**-Rg** 中 **g** 代表 **g**lobal。
2. 增加了 **-Bg** 选项。**-B** 用于给底图加上边框，其中的 **g** 则用于向底图中加入网格线（**g** 代表 **g**rid）。

小技巧: GMT 支持 30 多种不同的地图投影方式，详情见投影方式 一章。将上面的示例脚本中 **-JH180/12c** 修改为下面列出的一些值以了解不同投影方式之间的差异：

- **-JA280/30/12c**
- **-JI180/12c**
- **-JK180/12c**
- **-JN180/12c**
- **-JR180/12c**

3.3.2 区域地图

绘制区域地图与绘制全球地图没有本质区别，我们依然需要使用 **-J** 指定投影方式，使用 **-R** 指定绘图区域范围，使用 **-B** 指定底图边框。

下面的 GMT 命令绘制了一张新西兰地图。

```
gmt begin NewZealandMap png,pdf
    gmt coast -JM12c -RNZ -Ba -W0.5p -A10000
gmt end show
```

图 4: 使用国家代码指定绘图区域

这里我们使用了：

- **-JM12c** 指定了使用墨卡托投影（绘制区域地图最常用的投影方式），地图宽度为 12 厘米；
- **-RNZ** 指定了绘图区域范围，此处 **NZ** 是新西兰的国家代码，GMT 会自动根据国家代码提取该国家的区域范围并用于绘图；
- **-Ba** 用于给底图添加边框。对于某些地图投影，GMT 默认使用图中所示的黑白相间的底图边框。**a** 用于添加标注（**a**nnotation，即图中的经纬度数字）。

如果对于根据国家代码自动确定的区域范围不太满意，我们也可以使用 **-R** 选项最原始的方式来指定区域范围。即依次给定绘图区域的最小经度、最大经度、最小纬度和最大纬度。记起来也很容易，可以记为 *xmin/xmax/ymin/ymax*，或者直接记“西东南北”。

通过上面的图，我们大概知道了新西兰的经纬度范围。我们可以直接使用 **-R165/180/-48/-32** 指定绘图区域。

```
gmt begin NewZealandMap png,pdf
    gmt coast -JM12c -R165/180/-48/-32 -Bafg -W0.5p -A10000
gmt end show
```

图 5: 使用经纬度范围指定绘图区域

上面的例子中我们还调整了 **-B** 选项，**-Bafg** 表示绘制底图边框的标注（**a**nnotation）、刻度线（**f**rame，即图中黑白线段的间隔）和网格线（**g**rid）。标注、刻度线以及网格线的间隔由 GMT 算法自动决定。

对 GMT 自动算法确定的间隔不满意？你还可以任意指定自己想要的间隔。下面例子中，我们对底图进行了进一步的自定义，为 X 轴和 Y 轴分别指定了间隔，并给底图添加了标题：

- **-Bxa4g2** 表示设置 X 轴（**x**）的标注间隔为 4 度、网格线间隔为 2 度（**a4g2**）
- **-Bya3g3** 表示设置 X 轴（**y**）的标注间隔为 3 度、网格间隔为 3 度（**a3g3**）
- **-BWSen** 中 WSEN 分别是西南东北四个方向的英文单词首字母，用于控制四条边的属性。大写的 **WS** 表示给西边和南边添加标注，而小写的 **en** 则表示对于东边和北边只绘制边框但不显示标注
- **-BWSen+t" New Zealand"** 中 **+t** 用于给整张图添加标题（**t**itle）。由于标题 New Zealand 中含有空格，我们需要使用引号将其括起来

```
gmt begin NewZealandMap png,pdf
    gmt coast -JM12c -R165/180/-48/-32 -Bxa4g2 -Bya3g3 -BWSen+t"New Zealand" -W0.5p -A10000
gmt end show
```

图 6: 为区域地图添加标题

3.3.3 线性坐标轴

GMT 最擅长绘制地图，同时也适合绘制最简单的线性坐标轴。同样的，要想绘制线性坐标轴，我们依然需要使用 **-R** 指定坐标轴范围，使用 **-B** 绘制边框并指定边框和坐标轴属性，同时，我们还需要使用 **-JX** 指定线性坐标轴的宽度和高度。

下面的示例中，我们使用了：

- **-JX8c/5c** 表明线性坐标轴的宽度为 8 厘米，高度为 5 厘米
- **-R10/70/-4/8** 表明 X 轴范围为 10 到 70，Y 轴范围为-4 到 8
- **-Bxa10f5g10+l" X Label"** 设置了 X 轴的属性，**a10f5g10** 分别设置了标注（annotation）间隔为 10，刻度（frame）间隔为 5，网格线（grid ）间隔为 10；**+l" X Label"** 则为 X 轴添加了标签；对于 Y 轴同理；

```
gmt begin linearXY png,pdf
    gmt basemap -R10/70/-4/8 -JX8c/5c -Bxa10f5g10+l"X Label" -Bya4f2g2+l"Y Label" -BWSen+t"Linear X-
↪Y Plot"
gmt end show
```

图 7: 线性坐标轴

GMT 同样也可以绘制对数轴、指数轴以及极坐标轴，在这一节中不再介绍。

3.3.4 三维线性坐标轴

三维坐标轴相对于二维坐标轴多了一个 Z 轴。与二维线性坐标轴类似，我们需要针对 Z 轴指定 Z 轴的范围、Z 轴的高度以及三维视角。

下面的命令绘制了一个三维线性坐标轴，相比于传统的二维线性坐标轴，其不同之处在于：

- **-R** 选项中有 6 个数字，后面两个数字表示 Z 轴最小值和最大值
- **-JZ5c** 指定了 Z 轴的高度为 5 厘米
- **-Bzaf** 指定了 Z 轴的标注和刻度属性
- **-BSEwnZ+b** 中 **Z+b** 表示绘制一条 Z 轴，并绘制出整个长方体的所有边
- **-p130/30** 指定了看这个长方形的视角。130 和 30 分别为三维视角的方位角和高度角。这个解释起来稍复杂，读者可以自己试试修改这两个参数并查看效果。方位角的取值范围为 0 到 360 度，高度角的取值范围为 0 到 90 度。

```
gmt begin 3DMap png,pdf
    gmt basemap -R10/70/-4/8/-10/10 -JX8c/5c -JZ5c -Bxa10+lX -Bya4+lY -Bzaf+lDepth -BSEwnZ+b+t"3D␣
↪Plot" -p130/30
gmt end show
```

图 8: 三维线性坐标轴

3.4 绘制海岸线

GMT 中使用*coast* 模块绘制海岸线。**coast** 模块，顾名思义是海岸线、湖岸线的意思。该模块不仅可以绘制这些岸线，还可以绘制比例尺和行政边界等。

这一节我们将通过绘制一张美国地图来介绍 **coast** 模块的基本用法。

3.4.1 绘制海岸线和湖岸线

下面的脚本用最简单的命令绘制了一张美国地图。其中 **-R** 选项设置了绘图区域，**-JM15c** 设置投影方式和图片尺寸，**-Baf** 设置底图边框属性，这些在前一节都已经做了介绍。

coast 模块的选项 **-W0.5p,black** 表示绘制岸线，并设置岸线为 0.5p 宽的黑色线条。

```
gmt begin coastline png,pdf
    gmt coast -R-130/-50/20/60 -JM15c -Baf -W0.5p,black
gmt end show
```

图 9: 美国海岸线

上图看上去有些复杂。这是因为默认情况下 **-W** 选项会绘制所有水体与陆地的界限，包括海岸线、湖岸线、湖中岛的边界等。根据水体或陆地的不同，**coast** 模块将岸线分为从 1 到 4 的四个级别，依次指海岸线、湖岸线、湖中岛，以及湖中岛内的湖边界。

因而，我们可以在使用 **-W** 选项时指定要绘制哪一个等级的岸线。下面的脚本中，我们使用 **-W1/0.5p,black** 表示用 0.5p 宽的黑色线条绘制 1 级海岸线。

```
gmt begin coastline png,pdf
    gmt coast -R-130/-50/20/60 -JM15c -Baf -W1/0.5p,black
gmt end show
```

图 10: 绘制海岸线

可以在一个命令中多次使用 **-W** 选项，分别用不同的线条绘制不同等级的岸线。这里我们使用 **-W1/0.5p,black** 绘制 0.5p 黑色的海岸线，同时使用 **-W2/1p,lightred** 绘制 1p 宽的浅红色湖岸线。

```
gmt begin coastline png,pdf
    gmt coast -R-130/-50/20/60 -JM15c -Baf -W1/0.5p,black -W2/1p,lightred
gmt end show
```

图 11: 绘制海岸线和湖岸线

3.4.2 设置要绘制的对象的最小面积

上图看上去依然有些乱。主要是因为美国有大大小小的很多岛屿和湖泊，GMT 默认会把这些岛屿和湖泊都画出来。如果能够只绘制比较大的岛屿或湖泊就好了，我们可以通过 **-A** 选项设置要绘制的对象的最小面积（单位为平方千米）。

```
gmt begin coastline png,pdf
    gmt coast -R-130/-50/20/60 -JM15c -Baf -W1/0.5p,black -W2/1p,lightred -A5000
gmt end show
```

图 12: 设置要绘制的对象的最小面积

这里我们使用 **-A5000** 表示只绘制面积大于 5000 平方千米的湖泊或岛屿。这样子得到的图看上去更加干净简洁了。

3.4.3 填充陆地与水体

上面介绍了如何使用 **-W** 绘制海岸线和湖岸线。我们还可以不绘制岸线，而是为陆地和水体设置不同的填充色。

```
gmt begin coastline png,pdf
    gmt coast -R-130/-50/20/60 -JM15c -Baf -A5000 -Gred -Slightblue -Clightred
gmt end show
```

图 13: 填充颜色

其中，**-G** 设置了陆地区域的填充色，**-S** 设置水体的颜色，**-C** 则设置湖泊的颜色（若不指定 **-C**，则湖泊颜色由 **-S** 控制）。

当然，你也可以同时使用 **-W** 选项和 **-G** 选项，即绘制岸线并填充颜色。

3.4.4 绘制国界/州界

使用 **-N** 选项可以绘制国界/州界等行政边界。**-N1** 表示绘制国界线，**-N2** 表示绘制州界/省界线（目前只有美洲各国以及澳大利亚的国界的数据）。

```
gmt begin coastline png,pdf
    gmt coast -R-130/-50/20/60 -JM15c -Baf -A5000 -Gred -Slightblue -Clightred -N1/1p -N2/0.25p
gmt end show
```

图 14: 绘制国界与州界

除了 **-N** 选项之外，**-E** 选项也可以用于绘制州界、国界和省界，在*DCW: 世界数字图表* 一节会详细介绍。本入门教程中暂时略过。

3.4.5 添加比例尺

最后，我们还需要为地图添加比例尺。为了绘制比例尺，我们需要提供如下参数：

- 要绘制哪个纬度的比例尺
- 比例尺在图中的位置
- 比例尺的长度

在下面的例子中，我们使用了 **-Lg-60/25+c25+w1000k+f+u** 增加比例尺，其中：

- **+w1000k** 表示比例尺长度为 1000 千米
- **+c25** 表示绘制纬度为北纬 25° 处的比例尺
- **g-60/25** 则表示将比例尺画在北纬 25° 西经 60° 处
- **+f** 表示比例尺的风格为图中所示黑白相间的铁轨形式
- **+u** 表示显示比例尺对应的单位

```
gmt begin coastline png,pdf
gmt coast -R-130/-50/20/60 -JM15c -Baf -A5000 -Gred -Slightblue -Clightred -Lg-60/25+c25+w1000k+f+u
gmt end show
```

图 15: 添加比例尺

3.5 绘制线段和多边形

绘制线条和多边形是日常绘图最常见的需求之一，也是所有绘图软件必备的功能。这一节我们将学习如何使用 GMT 的*plot* 模块绘制线段和多边形。

3.5.1 绘制一条线段

要绘制一条线段，就必须提供线段上数据点的信息，即数据的 X 坐标和 Y 坐标。*plot* 会自动将输入数据中相邻的两点连接起来。

以下面的数据为例，这个数据中包含了三个坐标点 (2,2)、(8,2) 和 (5,7):

```
2   2
8   2
5   7
```

下面的例子中，我们首先使用 UNIX 下的 `cat` 命令将数据写入到文件 `points.dat` 中，然后使用 **basemap** 模块绘制了一张底图，并使用 **plot** 模块绘制输入文件 `points.dat` 中的数据。

警告: Windows Batch 用户请注意！Batch 不支持 EOF 语法！

```
cat > points.dat << EOF
2   2
8   2
5   7
EOF
```

> 这一段代码的作用是将两个 EOF 中间的三行数据保存到文件 points.dat 中。由于 Batch 不支持这一语法，Batch 用户需要在当前目录手动创建文件 points.dat 并将数据复制到文件中并保存。
>
> 其它示例也有相同的问题，Batch 用户自行修改，不再专门解释。

图中，**plot** 模块在绘图时自动将三个点连接起来，绘制出了一条线段。

```
cat > points.dat << EOF
2   2
8   2
5   7
EOF

gmt begin SimpleLine png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot points.dat
gmt end show
```

想要修改线段的粗细或颜色？很简单，可以使用 **plot** 模块的 **-W** 选项设置画笔属性。画笔属性包括三个部分：线宽、颜色以及线型，三者之间用逗号隔开。

下面的脚本中，我们给 **plot** 模块添加了 **-W2p,red,-** 选项，即设置了画笔属性为 **2p** 宽的红色虚线。**p** 是 GMT 中的一个长度单位。

```
cat > points.dat << EOF
2   2
8   2
5   7
EOF
```

(下页继续)

(续上页)

```
gmt begin SimpleLine png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot points.dat -W2p,red,-
gmt end show
```

你可以尝试修改线宽、颜色和线型，并查看绘图效果。几种常见的线型包括 -、.、.-和 -.。

3.5.2 绘制一个多边形

plot 在绘制线段时默认是不将线段首尾连接起来的，可以使用 **-L** 选项将线段的首尾连接起来，构成了一个闭合多边形。

```
cat > points.dat << EOF
2   2
8   2
5   7
EOF

gmt begin polygon png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot points.dat -W4p,lightblue -L
gmt end show
```

我们还可以使用 **-G** 选项为闭合多边形填充颜色。

```
cat > points.dat << EOF
2   2
8   2
5   7
EOF

gmt begin polygon png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot points.dat -W4p,lightblue -Glightred -L
gmt end show
```

这样我们就得到了一个内部为浅红色、轮廓为浅蓝色的多边形了。如果只想要填充颜色而不绘制轮廓，只需要使用 **-G** 而不使用 **-W** 即可。

```
cat > points.dat << EOF
2   2
8   2
5   7
EOF

gmt begin polygon png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot points.dat -Glightred -L
gmt end show
```

3.5.3 绘制多条线段

学会了如何绘制一条线段，我们就可以通过多次调用 **plot** 模块来绘制多条线段了。但是，如果画几十条线段就需要写几十行 **plot** 命令并准备几十个输入文件了，这样太麻烦也太不切实际了。

实际上，我们可以将所有线段的数据点都保存到一个输入文件中，例如:

```
>
1   2
4   2
4   8
>
9   2
6   2
6   8
```

每个线段都包含了若干个数据点，在第一个数据点之前有一个 > 用于标记新的一段数据的开始。这种数据称之为**多段数据**。

与绘制一条线段的命令完全相同，由于输入数据中有两段数据，**plot** 模块为我们绘制出了两条线段。同样的，两条线段均为线宽为 **1p** 的红色实线。

```
cat > lines.dat << EOF
>
1   2
4   2
4   8
>
9   2
6   2
6   8
EOF

gmt begin MultiLines png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot lines.dat -W1p,red
gmt end show
```

3.5.4 绘制多个多边形

使用相同的输入数据，通过加上 **-L** 可以构成闭合多边形，加上 **-G** 为多边形设置填充色。

```
cat > lines.dat << EOF
>
1   2
4   2
4   8
>
9   2
6   2
6   8
EOF

gmt begin MultiPolygons png,pdf
    gmt basemap -JX10c -R0/10/0/10 -Baf
    gmt plot lines.dat -W1p,red -L -Glightred
```

(下页继续)

(续上页)

```
gmt end show
```

3.5.5 大圆弧路径

在笛卡尔坐标系下，绘制线段时，任意两点之间会以直线方式连接；而在地理投影下，任意两点之间则使用大圆弧路径方法会连接。如果想要在地理投影下也是要直线连接两点，则需要使用 **-A** 选项设置画笔属性。

下面的命令中，我们首先使用 **coast** 绘制了一张全球地图，接着使用 **plot** 模块绘制了地球上两点之间的连线（红色，以大圆弧路径方式连接），然后，我们加上了 **-A** 选项再次绘制了这两点之间的连线（蓝色，以直线方式连接）。

```
cat > twopoints.dat << EOF
115         30
250         30
EOF

gmt begin map png,pdf
    gmt coast -JH180/12c -Rg -B0 -W0.5p -A10000
    gmt plot twopoints.dat -W2p,red
    gmt plot twopoints.dat -W2p,blue -A
gmt end show
```

3.6 绘制符号

绘制各种类型的符号也是常见的绘图需求之一。GMT 支持绘制十几种常见的符号类型，还支持自定义复杂的符号类型，足以满足日常的科研绘图需求。这一节将介绍如何使用*plot* 模块绘制多样的符号。

3.6.1 符号一览

plot 模块绘制符号需要使用 **-S** 选项。**-S** 选项后面紧跟着符号类型代码，以及符号大小。GMT 中常见的十几种简单符号及其对应的符号类型代码如下图所示。比如 **c** 代表圆（**c**ircle），**t** 代表三角形（**t**riangle）。

Source Code

3.6.2 绘制简单符号

以绘制圆圈为例，通过查询文档或者看上图可知，圆圈对应的符号类型代码为 **c**。**-Sc0.5c** 则表示绘制直径为 0.5 厘米的圆圈。为了绘制圆圈，我们需要给定圆圈的位置，因而输入数据中需要提供圆圈的 X 和 Y 坐标。

下面的示例中在 (2,3)、(5,6)、(8,2) 三个点绘制了三个直径为 0.5 厘米的圆圈。

```
gmt begin symbols png,pdf
gmt plot -R0/10/0/10 -JX10c/10c -Baf -Sc0.5c << EOF
2 3
5 6
8 2
EOF
gmt end show
```

与绘制线段和多边形类似，我们可以使用 **-W** 选项控制符号轮廓的画笔属性，使用 **-G** 选项为符号设置填充色。下面的命令就绘制了三个黑边红色圆圈。

```
gmt begin symbols png,pdf
gmt plot -R0/10/0/10 -JX10c/10c -Baf -Sc0.5c -W1p,black -Gred << EOF
2 3
5 6
8 2
EOF
gmt end show
```

若想要绘制其它符号，则只需要使用对应的符号类型代码即可。例如，将 **-Sc0.5c** 改成 **-St0.5c** 则会绘制三角形。

当然，某些符号类型可能需要额外的输入数据。比如 **-Sr** 表示绘制长方形，此时输入数据中除了需要长方形的 X 和 Y 坐标之外，还需要长方形的宽和高，因而输入数据需要四列。对于这些输入数据稍复杂的符号，可以阅读文档以了解其输入数据的格式。

3.6.3 大小变化的符号

若想要绘制大小变化的符号，则需要在输入数据中额外加上一列以控制每个符号的大小，同时，在 **-S** 选项中则不再需要指定符号大小。

下面的示例中，**-Sc** 中没有指定圆圈大小，此时输入数据的第三列控制圆圈大小。由此，我们即得到了大小变化的符号。

```
gmt begin symbols png,pdf
gmt plot -R0/10/0/10 -JX10c/10c -Baf -Sc -W1p,black -Gred << EOF
2 3 0.3
5 6 0.8
8 2 0.5
EOF
gmt end show
```

3.6.4 颜色变化的符号

前面提到，使用 **-G** 选项可以为符号填充颜色，但其只能同时为所有符号指定单一的颜色。如果想要让符号的颜色根据某个数值的不同而使用不同的颜色，则需要使用 **-C** 选项。**-C** 选项表示符号的填充色由某个数值以及 CPT 颜色表所控制。CPT 颜色表给出了数值与颜色之间的对应关系。因而对于任意一个符号，我们都可以给其一个数值，GMT 会根

据该数值从 CPT 颜色表中找到对应的颜色作为该符号的填充色。因而，在输入数据中，我们需要在 X 和 Y 坐标的基础上额外加一列 Z 值，用于控制符号的填充色。

下面的示例中，我们首先使用 **makecpt** 模块，以 GMT 内置 CPT 颜色表 **hot** 为基础，生成了一个新的 CPT 颜色表。关于 CPT 颜色表的具体细节在后面会介绍到。此处，读者只需要知道，我们制作了一个 CPT 文件供后面的命令使用。该 CPT 颜色表为 0 到 3 之内的每个数值都对应了一个颜色。

同时，对于输入数据，我们额外增加一列（通常称这一列为 Z 值），该列的值决定了符号的填充色。

```
gmt begin symbols png,pdf
gmt makecpt -Chot -T0/3/1
gmt plot -R0/10/0/10 -JX10c/10c -Baf -Sc0.5c -W1p,black -C << EOF
2   3   0
5   6   1
8   2   2
EOF
gmt end show
```

3.6.5 绘制不同的符号

前面说到，**-S** 选项中指定不同的符号类型代码则代表绘制不同的符号，但每次只能指定一种符号类型代码。如果想要不同的数据使用不同的符号绘制，则可以只指定符号大小而不指定符号类型代码，并在输入数据的最后一列指定符号类型代码。

下面的示例中，我们使用 **-S0.5c** 指定了符号的大小，但是没有指定符号类型。输入数据的最后一列中 **c**、**t** 和 **i** 则分别为三个数据指定了各自的符号类型。

```
gmt begin symbols png,pdf
gmt basemap -R0/10/0/10 -JX10c/10c -Baf
gmt plot -S0.5c -W1p,black -Gred << EOF
2 3 c
5 6 t
8 2 i
EOF
gmt end show
```

3.6.6 大小、颜色和符号类型都变化的符号

前面介绍了如何绘制大小变化或颜色变化或符号类型变化的符号。那么如何实现颜色、大小以及符号三者都变化的符号呢? 可以使用 **-S** 但不指定符号类型以及符号大小, 同时使用 **-C** 选项。

关键问题在于, 此时的数据数据格式是怎样的。根据 GMT 的规定, 此时输入数据的格式为:

```
X   Y   Z   size    symbol
```

即第三列控制符号填充色, 第四列为符号大小, 最后一列为符号类型代码。

```
gmt begin symbols png,pdf
gmt makecpt -Chot -T0/3/1
gmt plot -R0/10/0/10 -JX10c/10c -Baf -S -W1p,black -C << EOF
2   3   0   0.3 c
5   6   1   0.8 t
8   2   2   0.5 i
EOF
gmt end show
```

3.7 添加文字

GMT 的*text* 模块可以用于添加文字。

3.7.1 最简单的示例

若需要添加文字，则输入数据中必须给出文字的 X 和 Y 坐标以及具体的文字。因而，输入数据有三列:

```
X   Y   text
```

下面的示例首先用 **basemap** 模块绘制了一张底图，然后使用 **text** 模块在底图的不同位置添加了文字。

```
gmt begin map png,pdf
gmt basemap -R0/10/0/6 -JX10c/6c -Bafg1 -BWSen
gmt text << EOF
5 1 GMT TEXT1
5 3 GMT TEXT2
5 5 GMT TEXT3
EOF
gmt end show
```

3.7.2 文字属性

当然，我们可以为文字设置更丰富的属性，比如文字大小、字体、文字颜色以及文字旋转角度等。这可以通过 **-F** 选项来实现。

-F+f*font* 可以设置文字的属性，包括文字大小、字体和颜色，三者之间用逗号分隔。**-F+a** 则可以设置文字的旋转角度。

注解: GMT 默认支持 35 种字体，可以使用 `gmt text -L` 查看 GMT 支持的字体名及其对应的字体编号。

下面的示例中，**-F+f16p,1,red+a30** 即表示文字大小为 16p，字体为 1 号字体，颜色为红色，文字旋转角度为 30 度。

```
gmt begin map png,pdf
gmt basemap -R0/10/0/6 -JX10c/6c -Bafg1 -BWSen
gmt text -F+f16p,1,red+a30 << EOF
5 1 GMT TEXT1
5 3 GMT TEXT2
5 5 GMT TEXT3
EOF
gmt end show
```

3.7.3 文本框

对于每个文本字符串，我们还可以为其加上文本框。

- **-W** 选项控制文本框的画笔属性
- **-G** 选项控制文本框的填充色
- **-C** 选项控制文字与文本框之间的空白

```
gmt begin map png,pdf
gmt basemap -R0/10/0/6 -JX10c/6c -Bafg1 -BWSen
gmt text -F+f16p,1,red+a30 -W1p -Glightblue -C25%/25% << EOF
5 1 GMT TEXT1
5 3 GMT TEXT2
5 5 GMT TEXT3
EOF
gmt end show
```

3.7.4 对齐方式

对于任意一个文本，我们还可以设置其对齐方式与偏移量。GMT 中文本的默认对齐方式为居中对齐，即将整个文本的中心放在指定的 X 和 Y 坐标处。当然，用户也可以自行指定文本的对齐方式。

文本的对齐方式由水平对齐方式和垂直对齐方式共同决定。水平对齐方式有三种：左对齐（**L**eft）、居中对齐（**C**enter）和右对齐（**R**ight）。垂直对齐方式有三种：顶部对齐（**T**op）、居中对齐（**M**iddle）和底部对齐（**B**ottom）。三种水平对齐方式和三种垂直对齐方式，构成了文本的 9 种对齐方式。

-F+j 用于指定文本对齐方式。下面的示例中，**-F+jTL** 表示文本对齐方式为 **TL**（Top + Left），即表示以左上角方式对齐。从下图中可以看到，三个文本框的左上角被放在了 (5,1)、(5,3) 和 (5,5) 处。

```
gmt begin map png,pdf
gmt basemap -R0/10/0/6 -JX10c/6c -Bafg1 -BWSen
gmt text -F+f16p,1,red+jTL -W1p << EOF
```

（下页继续）

(续上页)

```
5 1 GMT TEXT1
5 3 GMT TEXT2
5 5 GMT TEXT3
EOF
gmt end show
```

3.7.5 文本偏移量

使用 **-D** 选项还可以对文本设置额外的偏移量。下面的示例中，**-D0.5c/0.5c** 分别设置了文本在 X 方向和 Y 方向的偏移量。

```
gmt begin map png,pdf
gmt basemap -R0/10/0/6 -JX10c/6c -Bafg1 -BWSen
gmt text -F+f16p,1,red+jTL -D0.5c/0.5c -W1p << EOF
5 1 GMT TEXT1
5 3 GMT TEXT2
5 5 GMT TEXT3
EOF
gmt end show
```

3.7.6 变化的文字属性

如果想要不同的文字有不同的文本属性，可以多次调用 **text** 模块，每次设置不同的文本属性。当然，还有更加灵活的办法，可以一个命令中实现变化的文字属性。

下面的例子中，使用了 **-F+f+a+j** 选项。上面已经介绍到，**+f** 设置文本属性，**+a** 设置文本旋转角度，**+j** 设置文本对齐方式。但我们并没有指定具体的属性值，因而需要在输入数据中增加额外的数据列。输入数据的格式由 **+f**、**+a** 和 **+j** 的顺序决定，因而此时输入数据的格式为:

```
X   Y   font    angle   justification   text
```

下面的示例中，第三列为字体属性，第四列为文本旋转角度，第五列为文本对齐方式。

```
gmt begin map png,pdf
gmt basemap -R0/10/0/6 -JX10c/6c -Bafg1 -BWSen
gmt text -F+f+a+j -W1p -Glightblue << EOF
5 1 12p,0,red       0   TL GMT TEXT1
5 3 15p,1,blue      30  MC GMT TEXT2
5 5 18p,2,yellow    180 TL GMT TEXT3
EOF
gmt end show
```

3.8 添加图例

绘制了线条与符号后，通常还需要添加图例，以解释不同数据的含义。GMT 中使用*legend* 模块添加图例。

3.8.1 自动图例

在使用*plot* 模块绘制线条或符号时，可以额外加上 **-l***label* 选项以指定当前线段或符号的图例标签。

下面的示例中，我们利用前面学到的知识绘制了线段和两种符号，同时使用了 **-l** 选项为线段和符号均添加了标签。在绘图结束时，GMT 会自动根据命令中提供的信息在右上

角绘制了图例。

```
gmt begin auto-legend png,pdf
gmt basemap -R0/10/0/10 -JX10c -Baf
gmt plot -W1p,blue -l"Profile" << EOF
3 3
6 8
EOF
gmt plot -Gred -Sa0.3c -W0.5p -l"Event" << EOF
5  5
EOF
gmt plot -Gblue -St0.3c -W0.5p -l"Station" << EOF
2 3
4 6
8 5
EOF
gmt end show
```

-l 选项还可以接受更多子选项以实现更多图例效果，在本教程中不再赘述。

3.8.2 手动设置图例

如果对于自动生成的图例不满意，还可以使用 **legend** 模块绘制更复杂的图例，其输入文件有自己的一套规则，详情见 **legend** 模块的说明文档。这节只介绍最简单也最常用的图例，即符号和线条的图例。针对绘制符号和线条，**legend** 的输入格式为：

S *dx1 symbol size fill pen dx2 text*

- **S** 表明这一行用于绘制符号或线段。
- *dx1* 是符号或线段与图例左边框的距离
- *symbol* 是要绘制的符号类型代码；若想要绘制线段，则设置 *symbol* 为 **-**
- *size* 符号尺寸或线段长度
- *fill* 符号填充色；若不需要填充色，则可设置为 **-**
- *pen* 符号轮廓的画笔颜色；若不需要绘制符号轮廓，则可设置为 **-**

- *text* 符号对应的文字说明
- *dx2* 文字与左边框之间的距离

下面的示例中，我们绘制了四种符号，以及线段、矢量线和断层线。

```
gmt begin map png,pdf
gmt basemap -R0/10/0/8 -JX10c/8c -Baf -BWSen
gmt legend -DjBR+w2.8c+o0.1c/0.1c -F+p1p+glightblue << EOF
# symbols
S 0.25c c 0.3c -      0.25p,blue 0.8c circle
S 0.25c t 0.3c cyan   0.25p      0.8c triangle
S 0.25c i 0.3c blue   0.25p,red  0.8c triangle2
S 0.25c e 0.3c yellow 0.25p      0.8c ellipse
# lines
S 0.25c - 0.5c - 0.25p 0.8c line
S 0.25c - 0.5c - 0.25p,- 0.8c dashline
S 0.25c v0.1i+a40+e 0.25i magenta 0.25p 0.8c vector
S 0.25c f0.1i+l+t 0.25i blue 0.25p 0.8c fault
EOF
gmt end show
```

我们可以通过修改 **-D** 和 **-F** 进一步修改图例的位置和属性。上面的例子中，

- **jBR** 表示将图例放在底图的右下角（BottomRight）
- **+o0.1c/0.1c** 表示将图例在右下角的基础上再加上额外的偏移量
- **+w2.8c** 设置了图例框的宽度
- **-F+p1p+glightblue** 则设置了图例框的轮廓和填充色

3.9 绘制地形起伏

绘制地图时另一个常见的需求是绘制全球或者区域的地形起伏作为底图。

GMT 中*grdimage* 模块可以绘制二维网格，其原理是建立 Z 值与颜色之间的映射关系，每个坐标点不同的 Z 值用不同的颜色表征。地形起伏，即不同经纬度处的高程或海深数据，也是一种二维数据，因而我们可以使用 **grdimage** 模块绘制地形起伏。

3.9.1 全球地形起伏数据

要绘制全球或区域地形起伏图，首先需要拥有地形起伏数据，即不同经纬度处的高程或海深数据。GMT 收集整理并提供了一套全球地形起伏数据，供用户直接使用。这套地形起伏数据分为包含了从 60 弧分到 1 弧秒的多种不同精度的全球地形起伏数据，以满足不同的绘图区域大小的需求。

GMT 用户可以通过给定文件名 `@earth_relief_xxx` 的方式来指定要使用某个精度的地形。*xxx* 用于指定数据精度，例如 **15m**、**01m** 和 **15s** 分别表示数据分辨率为 15 弧分、1 弧分和 15 弧秒。

文件名前的 @ 表示该数据是由 GMT 官方提供并维护的数据，当第一次使用该文件时，GMT 会自动从服务器下载该数据并保存到本地的 GMT 数据目录中，当以后再使用该文件时，则直接读取本地文件，而无需重新下载该数据。

关于全球地形起伏数据的详细介绍见*earth_relief: 全球地形起伏数据*。

3.9.2 绘制全球地形起伏图

下面我们将使用 **grdimage** 模块绘制 5 弧分精度的全球地形起伏数据（`@earth_relief_05m`）。不同的高程用不同的颜色表示，从中我们很容易看出不同地区的地形变化。

```
gmt begin global_relief png,pdf
gmt grdimage @earth_relief_05m -JH180/10c
gmt end show
```

图 16: 全球地形起伏图

3.9.3 绘图区域地形起伏图

想要绘制区域地形起伏？只需要在 **grdimage** 模块中使用 **-R** 选项指定区域经纬度范围即可。这里我们设置了绘图区域为 118°E 至 125°E、20°N 至 26°N。由于绘图区域比较小，所以我们选用了更高精度的 30 弧秒的地形起伏数据。

注解： 30 弧秒的地形起伏数据大小约为 800Mb。第一次使用该数据时 GMT 会自动从服务器下载该文件，因而会很耗时，请耐心等待。

```
gmt begin taiwan_relief png,pdf
gmt grdimage @earth_relief_30s -JM15c -R118/125/20/26 -Baf -BWSen
gmt end show
```

图 17: 台湾区域地形图

3.9.4 增加光照效果

为了让地形起伏图更加立体，我们可以为图幅加上光照效果。我们可以指定光照的方向、强度等参数，也可以直接使用 **-I+d** 以使用默认的光照效果。

下面的示例中，我们加上了 **-I+d** 以增加光照效果。跟上图比一比，是否图片更加立体也更加美观了呢？

```
gmt begin taiwan_relief png,pdf
gmt grdimage @earth_relief_30s -JM15c -R118/125/20/26 -Baf -BWSen -I+d
gmt end show
```

图 18: 带光照效果的台湾区域地形图

增加光照效果本质上是计算了每个点沿着某个方位角的方向梯度，然后根据每个点的方向梯度的正负以及振幅调节该点颜色的亮度值。对于向阳处，其方向梯度为正值，则增加该点颜色的亮度；对于背阴处，其方向梯度为负值，则降低该点颜色的亮度。由此达到增加光照效果、增强立体感的目的。

3.9.5 添加色标

前面提到，**grdimage** 绘制地形起伏数据本质上就是将高程的数值与颜色之间对应起来。二者之间的对应关系由色标文件（即 CPT 文件）决定。那么，上图使用的是怎么样的 CPT 呢？高程数值与颜色之间的对应关系又是怎样的呢？不同的颜色代表的具体数值又是多少呢？这就需要用*colorbar* 向图中添加色标。

```
gmt begin taiwan_relief png,pdf
gmt grdimage @earth_relief_30s -JM15c -R118/125/20/26 -Baf -BWSen -I+d
gmt colorbar
gmt end show
```

上面的脚本中 **colorbar** 命令在地形图的下方添加了一个色标，但是色标下面的有一团很乱的标注，这显示不是我们想要的。我们可以使用 **-B** 选项设置色标的标注间隔，并为色标添加一个标签。

```
gmt begin taiwan_relief png,pdf
gmt grdimage @earth_relief_30s -JM15c -R118/125/20/26 -Baf -BWSen -I+d
gmt colorbar -Bxaf+l"Elevation (m)"
gmt end show
```

当然，我们还可以更进一步调整色标的位置、长度等属性。下面的脚本中，我们使用了 **-D** 选项将色标放在了地形起伏图的右侧中间（**JMR**）向右偏移 1.5 厘米，色标长度为 10 厘米，并将标签放在了色标左侧（**+ml**）。

```
gmt begin taiwan_relief png,pdf
gmt grdimage @earth_relief_30s -JM15c -R118/125/20/26 -Baf -BWSen -I+d
gmt colorbar -DJMR+w10c+o1.5c/0c+ml -Bxa1000f+l"Elevation (m)"
gmt end show
```

3.9.6 制作 CPT 文件

上面的示例中使用的是 GMT 的默认 CPT 文件。用户也可以使用*makecpt* 或 grd2cpt 制作 CPT 文件。

下面的示例中，我们使用*makecpt* 模块在 GMT 内置 CPT **globe** 的基础上生成了一个-8000 到 8000 范围内的新 CPT 文件。生成的 CPT 文件将作为当前 CPT 文件，供接下来的 **grdimage** 和 **colorbar** 命令使用。

```
gmt begin taiwan_relief png,pdf
gmt basemap -JM15c -R118/125/20/26 -Baf -BWSen
gmt makecpt -Cglobe -T-8000/8000
gmt grdimage @earth_relief_30s -I+d
gmt colorbar -Bxa2000+l"Elevation/m"
gmt end show
```

3.10 理解图层

在前面几节中，我们已经学会了如何使用 GMT 绘制底图、海岸线、线段、符号、文字、地形起伏等等。这一节我们将把前面学到的内容综合起来，试着去绘制下面这张地震学经常见到的大圆弧路径图。在这一过程中，我们将试着去理解 GMT 中图层的概念。

Source Code

3.10.1 图件分析与拆解

上面这张图看上去有些复杂，实际上是由很多部分构成的。将整张图拆解一下可知，上图由如下几个部分构成：

1. 地形起伏作为底图
2. 震中位置（五角星）
3. 台站位置（三角形）
4. 射线路径（图中大圆弧）
5. 台站名（文字）

这几个部分，在教程的前几节都已经做过介绍，因而只需要将前几节的内容综合起来即可。

3.10.2 绘制底图

我们首先使用*grdimage* 模块绘制底图，并使用*colorbar* 模块添加色标：

```
#!/bin/bash
gmt begin layers png,pdf

# 绘制地形起伏底图
gmt basemap -JH180/15c -Rg -B0
gmt grdimage @earth_relief_10m -Cetopo1 -I+d
gmt colorbar -Bxa2000f+l"Elevation (m)"
gmt end show
```

3.10.3 绘制震中和台站位置

接下来，使用*plot* 模块绘制五角星和三角形。

```
#!/bin/bash
gmt begin layers png,pdf

# 绘制地形起伏底图
gmt basemap -JH180/15c -Rg -B0
gmt grdimage @earth_relief_10m -Cetopo1 -I+d
gmt colorbar -Bxa2000f+l"Elevation (m)"

# 绘制震中位置
gmt plot -Sa0.5c -W0.5p,black,solid -Gyellow << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt plot -St0.2c -W0.5p,black,solid -Gblack << 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 end show
```

3.10.4 绘制射线路径

再使用*plot* 模块绘制线段。默认情况下，**plot** 会自动用大圆路径连接地球上的两个位置，因而我们只需要用 > 分隔多个线段，每个线段给定两个坐标点（即地震位置和台站位置）即可。

```
#!/bin/bash
gmt begin layers png,pdf

# 绘制地形起伏底图
gmt basemap -JH180/15c -Rg -B0
gmt grdimage @earth_relief_10m -Cetopo1 -I+d
gmt colorbar -Bxa2000f+l"Elevation (m)"

# 绘制震中位置
gmt plot -Sa0.5c -W0.5p,black,solid -Gyellow << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt plot -St0.2c -W0.5p,black,solid -Gblack << 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 plot -W1p,red << 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

gmt end show
```

3.10.5 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*text* 模块。这里我们使用了 **-F+f9p,1,black+j** 选项，因而输入数据是 4 列:

```
X   Y   对齐方式    TEXT
```

-Dj0.1c/0.1c 则是将文本在对齐方式的基础上做进一步的偏移以避免文字覆盖线段或符号。

```
#!/bin/bash
gmt begin layers png,pdf
gmt basemap -JH180/15c -Rg -B0
gmt grdimage @earth_relief_10m -Cetopo1 -I+d
gmt colorbar -Bxa2000f+l"Elevation (m)"

# 绘制震中位置
gmt plot -Sa0.5c -W0.5p,black,solid -Gyellow << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt plot -St0.2c -W0.5p,black,solid -Gblack << 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 plot -W1p,red << 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

# 添加文本
gmt text -F+f9p,1,black+j -Dj0.1c/0.1c << EOF
-77.15  38.89 ML Washington
76.22  -69.22 ML Zhongshanzhan
104.39  29.90 MR Zigong
13.14   52.50 TC Berlin
19.99  -34.52 ML Bredasdorp
150.36 -33.79 ML Sydney
```

(下页继续)

(续上页)

```
-52.47 -31.62 ML Pelotas
EOF

gmt end show
```

3.10.6 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。

对脚本中命令的先后顺序进行微调，如下所示：

```
#!/bin/bash
gmt begin layers png,pdf
gmt basemap -JH180/15c -Rg -B0
gmt grdimage @earth_relief_10m -Cetopo1 -I+d
gmt colorbar -Bxa2000f+l"Elevation (m)"

# 绘制大圆路径
gmt plot -W1p,red << EOF
>
130.72 32.78
104.39 29.90
>
130.72 32.78
13.14 52.50
>
130.72 32.78
```

(下页继续)

(续上页)

```
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 plot -Sa0.5c -W0.5p,black,solid -Gyellow << EOF
130.72 32.78
EOF

# 绘制台站位置
gmt plot -St0.2c -W0.5p,black,solid -Gblack << 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 text -F+f9p,1,black+j -Dj0.1c/0.1c << EOF
-77.15  38.89 ML Washington
76.22  -69.22 ML Zhongshanzhan
104.39  29.90 MR Zigong
13.14   52.50 TC Berlin
19.99  -34.52 ML Bredasdorp
150.36 -33.79 ML Sydney
-52.47 -31.62 ML Pelotas
EOF

gmt end show
```

这样我们就通过组合一系列简单的 GMT 命令，得到了一个复杂的 GMT 图件。

3.11 GMT 单行模式

在之前的教程中，有很多示例都只需要一个 GMT 命令即可完成绘图。比如:

```
gmt begin GlobalMap png,pdf
    gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot"
gmt end show
```

在这种简单的情况下，每次都需要写 **gmt begin** 和 **gmt end** 未免有些麻烦。

针对这种简单情况，GMT 提供了“单行模式”。即当绘图只需要一个 GMT 命令时，可省略 **gmt begin** 和 **gmt end**，只需要在绘图命令后加上 **-***format figname* 即可。

例如，上面的三行命令可以用单行模式写成一行命令:

```
gmt coast -Rg -JH15c -Gpurple -Baf -B+t"My First Plot" -pdf,png GlobalMap
```

命令一下子就简单了很多，也可以直接将命令复制粘贴到终端执行。

显然，单行模式只适用于单个命令成图的情况，而大部分实际绘图都需要多个命令才能实现。

注解: GMT 官方文档以及本手册中，很多示例都只需使用单个 GMT 命令来展示某个模块的用法或某个选项的绘图效果。

出于简化代码的考虑，本手册中的很多演示示例都采用了 GMT 单行模式。读者应理解其中的差异，并了解如何修改单行模式的命令并应用到自己的实际绘图脚本中。

3.12 GMT 图中图模式

在绘制区域地图时，为了更清晰地显示研究区域的地理位置，通常会额外绘制一个更大区域范围的地图，并在大区域地图中标记出研究区域的位置。这种图称之为 inset map。中文没有统一的翻译，大家一般称之为图中图、小图或者插页图。

GMT 中*inset* 模块用于管理图中图模式。图中图模式以 **inset begin** 开始，并以 **inset end** 结束。

使用 **inset begin** 进入图中图模式，其作用是在纸张上规划出一个矩形区域，并将绘图原点移动到该矩形区域的左下角。接下来的所有绘图命令均只在该区域内进行操作。当使用 **inset end** 结束图中图模式时，绘图原点会自动恢复到图中图模式之前的位置，且所有设置参数都将回到之前的状态。

3.12.1 用矩形框标记研究区域

下面的示例中研究区域为日本东京周围的一个小区域。为了展示其地理位置，我们在大图的左下角的小图中绘制了日本全图，同时在小图中用矩形框出了大图中的研究区域范围。

inset begin 定义了小图的位置位于大图左下角（**-DjBL**），小图区域的宽度为 3 厘米，高度为 3.6 厘米（**+w3c/3.6c**），并且相对大图左下角偏移 0.1 厘米（**+o0.1c**）。同时还设置了小图区域的背景色为白色（**+gwhite**），并绘制了小图区域的边框（**+p1p**）。

在小图区域内，我们使用 **coast** 模块绘制了日本全图，小图投影参数为 **-JM?**，其中 **?** 表示根据 **inset begin** 中 **-D** 选项指定的宽度自动决定小图宽度。同时我们使用 **plot** 模块的 **-Sr+s** 选项在小图中绘制了一个对应于大图区域的矩形框，该命令需要输入矩形区域的两个对角顶点的经纬度。

```
#!/usr/bin/env bash
gmt begin inset-map png,pdf
    gmt coast -R139.2/140.5/34.8/36 -JM12c -Baf -BWSne -W2p -A1000 -Glightbrown -Sazure1 --FORMAT_
↪GEO_MAP=dddF
    gmt inset begin -DjBL+w3c/3.6c+o0.1c -F+gwhite+p1p
        gmt coast -R129/146/30/46 -JM? -EJP+glightbrown+p0.2p -A10000
        # 使用 -Sr+s 绘制矩形区域
        echo 139.2 34.8 140.5 36 | gmt plot -Sr+s -W1p,blue
    gmt inset end
gmt end show
```

3.12.2 用颜色标记研究区域

下面的示例中研究区域为澳大利亚。为了展示其地理位置，我们在大图的右上角区域绘制了全球地图，并用特殊的颜色将澳大利亚区域标记出来。

inset begin 定义了小图的位置位于大图右上角（**-DjTR**），小图区域宽度为 1.5 英寸（**+w1.5i**）。同样的，在小图内部我们在需要指定地图宽度的地方使用了 **?** 让 GMT 自动帮我们决定小图的宽度。

```
#!/usr/bin/env bash
gmt begin inset-example png,pdf
    gmt coast -R110E/170E/44S/9S -JM6i -B -BWSne -Wfaint -N2/1p -Gbrown -EAU+gbisque
    gmt inset begin -DjTR+w1.5i+o0.15i/0.1i -F+gwhite+p1p+c0.1c
        gmt coast -JG120/30S/? -Rg -Bg -Wfaint -Gbrown -EAU+gbisque -A5000
    gmt inset end
gmt end show
```

3.13 GMT 子图模式

有些时候，尤其是发表文章时，需要将多张独立的图放在一张图中，并编号 abcd，一般称这些独立的图为子图。

GMT 中有两种方式可以绘制多子图：

- 常规方式：在绘图时使用*-X 和 -Y 选项* 手动移动每个子图的原点
- 现代方式：使用*subplot* 模块提供的子图模式来布局和管理多子图

现代方式更加简洁易用，建议使用现代方式。仅当图片非常复杂或不规则时，才推荐使用常规方式。

3.13.1 子图布局

subplot 模块提供的子图模式可以非常方便地绘制多子图。

subplot begin 用于设计子图的布局、尺寸以及其它属性。其将整张画布划分为 N 行 M 列的规则网格区域，每个网格区域内都可以包含一张独立的子图。例如:

```
gmt subplot begin 2x3 -Fs5c/3c
```

定义了一个 2 行 3 列的子图布局。**-Fs5c/3c** 则指定了每个子图区域的宽度为 5 厘米，高度为 3 厘米。相邻子图之间的间隔则可以用 **-M** 选项控制。最终得到的子图布局如下图所示：

Source Code

subplot set 用于激活指定的子图，接下来的所有绘图命令都将在该子图内进行绘制。为了指定某个子图，则需要知道每个子图的编号。GMT 中可以通过 **行号，列号**或者 **索引号**（即第几个子图）的方式来指定子图。

注解: 行号、列号和索引号，均从 0 开始起算。因而对于一个 N 行 M 列的子图布局而言，行号为 0 到 N-1，列号为 0 到 M-1，索引号为 0 到 N*M-1。

上图中同样给出了每个子图的编号，图中红色数字为子图的行列号，而括号中的蓝色数字则是子图的索引号。因而，你可以使用如下命令中的任意一个来激活第三个子图，接下来的所有绘图命令均只在第三个子图内进行:

```
gmt subplot set 1,0

gmt subplot set 3
```

最后记得使用 **subplot end** 退出子图模式:

```
gmt subplot end
```

3.13.2 第一张子图

下面就利用上面学到的知识绘制一张 2 行 2 列的子图。

```
gmt begin map png,pdf
  gmt set FONT_TAG 15p FONT_HEADING 20p MAP_HEADING_OFFSET 0p
  gmt subplot begin 2x2 -Fs5c/3c -A -M0.2c/0.1c -T"My Subplot Heading"
    gmt subplot set 0
    gmt basemap -R0/10/0/10 -JX? -Baf -BWSen

    gmt subplot set 1
    gmt basemap -R0/20/0/10 -JX? -Baf -BWSen

    gmt subplot set 2
    gmt basemap -R0/10/0/20 -JX? -Baf -BWSen

    gmt subplot set 3
    gmt basemap -R0/20/0/20 -JX? -Baf -BWSen
  gmt subplot end
gmt end show
```

在这个例子中，我们用 **subplot begin** 定义了一个 2 行 2 列 (**2x2**) 的子图布局，每个子图区域宽 5 厘米高 3 厘米 (**-Fs5c/3c**)。除此之外，我们还使用了一些可选选项对图的

细节进行微调:

- **-A**: 对每个子图进行自动编号 abcd
- **-M0.2c/0.1c**: 调整相邻子图之间的空白距离, X 方向间隔为 0.2 厘米, Y 方向间隔为 0.1 厘米
- **-T"My Subplot Heading"**: 为整张图加上一个总标题
- 调整子图编号的大小 (*FONT_TAG*)、总标题文字大小 (*FONT_HEADING*) 以及总标题相对于底图的偏移量 (*MAP_HEADING_OFFSET*)

在子图模式内, 我们使用 **subplot set 0** 的方式依次激活每个子图。在每个子图内绘图时, 我们使用了线性投影方式 **-JX?**。通常我们需要指定图片的宽度或高度, 这里我们使用了 **?** 让 GMT 根据子图区域的大小自动帮我们选择最合适的子图宽度。

小技巧: 本示例中使用了如下命令来依次激活四个子图:

```
gmt subplot set 0
gmt subplot set 1
gmt subplot set 2
gmt subplot set 3
```

实际上, 我们可以直接使用 **subplot set** 而不指定子图编号, GMT 会自动为我们激活 "下一个" 子图。

3.13.3 共用 X/Y 轴

上面示例中的四张子图, 每行的两张子图有相同的 Y 轴范围, 每列的两张子图有相同的 X 轴范围。此时可以使用 **-S** 选项设置各子图之间共用 X 或 Y 轴。

```
gmt begin map png,pdf
  gmt set FONT_TAG 15p FONT_HEADING 20p MAP_HEADING_OFFSET 10p
  gmt subplot begin 2x2 -Fs5c/3c -A -M0.2c/0.2c -T"My Subplot Heading" -SRl -SCb -BWSrt
    gmt basemap -R0/10/0/10 -JX? -c
    gmt basemap -R0/20/0/10 -JX? -c
    gmt basemap -R0/10/0/20 -JX? -c
    gmt basemap -R0/20/0/20 -JX? -c
  gmt subplot end
gmt end show
```

-SRl 表示一行内（**R**ow）的子图共用 Y 轴，且只在左边（**l**）轴显示标注，**-SCb** 表示一列内（**C**olumn）的子图共用 X 轴，且只在底部（**b**）轴显示标注。

当然你也可以不使用 **-S** 选项，而是在每个子图中使用不同的 **-B** 选项分别为每个子图设置不同的轴属性。

3.13.4 复杂布局

subplot 目前尚不支持嵌套。如果想要使用更复杂的子图布局，则需要更多的人工调整。

下面的例子在一个 2 行 2 列的子图布局中绘制了三张子图，其中第一张子图占据了第一行。

```
gmt begin complex-subplot png,pdf
  gmt subplot begin 2x2 -Fs5c/3c -A
    gmt subplot set 0 -A'(a)'
    gmt basemap -R0/10/0/10 -JX11.75c/3c -Baf -BWSen
    echo 5 5 TEXT | gmt text -JX11.75c/3c
    gmt subplot set 2 -A'(b)'
    gmt basemap -R0/5/0/5 -JX? -Baf -BWSen
    gmt subplot set 3 -A'(c)'
    gmt basemap -R0/5/0/5 -JX? -Baf -BWSen
  gmt subplot end
gmt end show
```

在绘制三个底图时，后两个底图均使用了 **-JX?**，因而 GMT 会自动根据子图区域的大小确定子图的尺寸；而为了使得第一张子图占据两个子图区域的空间，我们使用了 **-JX11.75c/3c** 来人工指定其子图宽度，其中子图宽度 11.75 厘米是需要人工调整的。

注解: 以上示例存在尚未修复的 BUG。

子图模式下每张子图的大小是由 **subplot begin** 的 **-F** 选项控制。用户可以使用 **-J** 指定某个子图的尺寸，但该尺寸不会被记住，接下来的命令若不指定 **-J** 则会使用子图模式默认的尺寸，继而出错。

目前的解决办法是，在该子图内的所有命令中均使用相同的 **-J** 选项。例如，子图 a 强行指定了子图大小为 **-JX11.75c/3c**，则该子图内的其余命令（如 **text** 命令）必须也使用 **-JX11.75c/3c**。若省去，则图会出错。

由于我们跳过了第二个子图区域，自动标签功能会将三个子图依次编号为 a、c、d，这显然不是我们想要的，因而我们使用了 **-A' (a)'** 选项手动设置子图编号。需要注意的是，由于小括号在 Bash 中有特殊含义，所以这里 **(a)** 两边加了单引号以避免 Bash 对小括号进行解释。

注解: 对于 Windows Batch 用户，不可使用单引号。可以使用双引号，或者不使用引号。

3.14 GMT 多图模式

如果你想要在一个脚本中同时绘制多张图，并且想要在多张图之间来回切换，则需要使用*figure* 模块。

figure 的用法与*begin* 相同，唯一的区别在于，你可以在 **gmt begin** 和 **gmt end** 中多次使用 **gmt figure** 来创建新图，或激活已有的图。

下面的示例中，我们使用 **figure** 模块在一个脚本中指定了两张图，并不断在两张图之间来回切换:

```
gmt begin
    # 创建 Fig1 并在 Fig1 中绘图
    gmt figure Fig1 png
    gmt basemap -R0/10/0/10 -JX10c -Baf

    # 创建 Fig2 并在 Fig2 中绘图
    gmt figure Fig2 png
    gmt basemap -R0/5/0/5 -JX10c -Baf

    # 切换回 Fig1，并绘制圆圈
    gmt figure Fig1
    echo 5 5 | gmt plot -Sc1c -W2p

    # 切换回 Fig2，并绘制三角形
    gmt figure Fig2
    echo 1 2 | gmt plot -St1c -W1p
gmt end show
```

最终会生成两张图，第一张图中在底图中绘制了圆圈，第二张图中则在底图中绘制了三角形。

保护环境, 从阅读电子文档开始!

第 4 章　GMT 基础

这一章介绍 GMT 中所有模块通用的基础知识。

4.1 画布

要画一张图，首先需要准备一个画布。GMT 默认的画布是一个 PDF 文件。

4.1.1 图片格式

GMT 支持多种矢量图片格式和位图图片格式。

矢量图片格式包括 PDF（推荐使用）、PS 和 EPS 格式；位图图片格式包括 PNG（推荐）、JPG（推荐）、BMP、PPM 和 TIFF 格式。

在开始 GMT 绘图时，即可指定要生成的图片格式。例如，下面的示例指明同时生成 PDF 和 PNG 格式的图片：

```
gmt begin figname pdf,png
gmt ...
gmt end show
```

小技巧： 如无特殊需求，不建议使用 PS 图片格式。PS 图片格式具有如下缺点（在后面会提到）：

1. 不支持透明色
2. 默认画布大小为 A4，而其它格式默认画布大小均为 11.55 米 x11.55 米（允许的最大尺寸）
3. PS 阅读器只有一两款，不如 PDF 或其它图片格式选择性多

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：

```
gmt begin map pdf,png
gmt set PS_PAGE_COLOR lightred
gmt ...
gmt end show
```

4.1.3 画布大小

GMT 中默认画布宽 11.55 米高 11.55 米，在生成图片文件时会自动将未使用的白色区域裁减掉，因而用户无需担心。

唯一的例外是 PS 图片格式，若选择 PS 作为图片格式，则默认画布大小为 A4 纸大小。当用户需要更大的画布时，需要修改参数*PS_MEDIA* 来修改画布尺寸：

```
gmt begin map ps
gmt set PS_MEDIA A1
gmt ...
gmt end
```

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的距离。

4.2.1 长度单位

GMT 中的长度量可以使用厘米（**c**m）、英寸（**i**nch）或点（**p**oint）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（**d**egree of arc）
- **m**：弧分（**m**inute of arc）
- **s**：弧秒（**s**econd of arc）
- **k**：千米（**k**ilometer）
- **e**：米（m**e**ter）**默认单位**

还有几个不常用的单位：

- **f**：英尺（**f**oot）
- **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。

4.2.3 小练习

1. 执行命令 `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 plot -JX10c/10c -R0/10/0/10 -Baf -Sc3c -W1p -pdf test
```

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 支持通过颜色名指定 663 种颜色。下图中列出了 GMT 支持的所有颜色名及其对应的 RGB 值。

GMT 颜色表下载： A4 PNG 格式 | A4 PDF 格式 | USLetter PNG 格式 | USLetter PDF 格式

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（**R**ed）、绿（**G**reen）、蓝（**B**lue）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（**H**ue）、饱和度（**S**aturation）和明度（**V**alue），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：

- **C**yan：青色，又称为天蓝色或是湛蓝
- **M**agenta：品红色，又称为洋红色
- **Y**ellow：黄色
- blac**K**：定位套版色（黑色）

GMT 中可以通过 $c/m/y/k$ 的方式指定 CMYK 格式的颜色。c、m、y、k 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 自定义了多个颜色名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

每一种颜色, 都可以额外指定其透明度。

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 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**	10p
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

在颜色 中已经介绍了如何指定颜色，所有的颜色均可用于指定画笔的颜色。

4.4.3 画笔线型

画笔线型属性 *style* 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 **.**；
   - **dashdot**：划点线，等效于 **-.**；
   - **dotdash**：点划线，等效于 **.-**。

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. 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 (默认值为 0)。

   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**。

4.4.4 小练习

下面的命令用*plot* 模块绘制了一条线段。读者可以将命令中 **-W***pen* 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 plot -R0/10/0/5 -JX10c/5c -Wpen -pdf pens << EOF
0 2
10 2
EOF
```

Source Code

发挥你的想象力, 想想如何绘制如下图所示的黑白相间的铁轨线? 点击 “Source Code” 可以查看下图对应的绘图脚本。

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。

最常见的情况是使用 **plot** 模块绘制一个多边形区域或一个符号，然后使用 **-G***fill* 选项为其填充颜色。为了方便描述，这一节以 **-G***fill* 为例。

4.5.1 填充颜色

给多边形或符号填充颜色很简单，直接用 **-G***color* 即可。颜色在颜色 一节已经介绍过了。比如 **-Gred** 表示填充色为红色，**-G230/200/0** 表示填充色为 **230/200/0**。

4.5.2 填充图案

除了可以填充颜色之外，还可以使用填充图案（pattern）。比如地质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：**-GP**|**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*GMT 预定义填充图案*
- 取文件名，表示使用一个 1、8 或 24 位 Sun 光栅文件作为位图图案

+**r***dpi* 设置了用于填充的位图图案的分辨率。*dpi* 越大，则填充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. 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*image* 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作为填充图案
3. 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*plot* 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 plot -JX10c/10c -R0/10/0/10 -Baf -Sc4c -Gfill -pdf test
```

请尝试将命令中的 **-G***fill* 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**

4.5.4 附录: 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 1: GMT 内置位图图案

下载: PDF 格式 | PNG 格式

4.6 文字

文字，也称文本或字符串。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*。

在模块的语法介绍中，通常使用 *font* 表示需要用户指定文字属性，用户应自行将其脑补为 *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。

4.6.1 文字大小

文字大小, 即常说的字号, 可以用数字加单位表示。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 Microsoft Word 中字号大小的对应关系。如下表所示:

表 2: Word 字号与 GMT 中字号 (p) 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-DemiItalic***
2	*Helvetica-Oblique*	19	Bookman-Light
3	***Helvetica-BoldOblique***	*20*	*Bookman-LightItalic*
4	Times-Roman	21	Helvetica-Narrow
5	**Times-Bold**	**22**	**Helvetica-Narrow-Bold**
6	*Times-Italic*	*23*	*Helvetica-Narrow-Oblique*
7	***Times-BoldItalic***	***24***	***Helvetica-Narrow-BoldOblique***
8	Courier	25	NewCenturySchlbk-Roman
9	**Courier-Bold**	*26*	*NewCenturySchlbk-Italic*
10	*Courier-Oblique*	**27**	**NewCenturySchlbk-Bold**
11	***Courier-BoldOblique***	***28***	***NewCenturySchlbk-BoldItalic***
12	Σψμβολ (Symbol)	29	Palatino-Roman
13	AvantGarde-Book	*30*	*Palatino-Italic*
14	*AvantGarde-BookOblique*	**31**	**Palatino-Bold**
15	**AvantGarde-Demi**	***32***	***Palatino-BoldItalic***
16	***AvantGarde-DemiOblique***	*33*	*ZapfChancery-MediumItalic*
		34	✺❁❒❄✤❇■✳❂❁▼▲ (ZapfDingbats)

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 text -L` 可以在命令行查看 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

可以为文字指定填充色或填充图案，也就是常说的文字颜色。

在颜色 和填充 一节中我们已经介绍了如何设置填充色和填充图案，其中介绍的内容均可用于设置文本颜色。

4.6.4 描边

在给文字指定填充色的同时，还可以在填充色 *fill* 后加上 =*pen* 以指定文本轮廓（即描边）的画笔属性。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

text 模块可以用于添加文字。下面的命令在图上写了文字 **TEXT**

```
echo 2.5 0.5 TEXT | gmt text -R0/5/0/1 -JX15c/2c -F+ffont -pdf text
```

将 **-F+f***font* 中的 *font* 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 *font* 取不同值的效果：

Source Code

TEXT	30p,5,-=1p,black,solid
TEXT	30p,5,blue=1p,black,solid
TEXT	30p,Courier-Bold,blue
TEXT	30p,8,red
TEXT	30p,8
TEXT	30p

4.7 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。

这些特殊字符无法通过键盘直接输入，GMT 中可以通过输入八进制码的方式指定特殊字符。

4.7.1 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+

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

Standard+

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

图 2: 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` 命令查到你所安装的 GMT 的默认字符编码方式。

这张表如何查询呢？假如你的 GMT 的默认字符编码方式为 **ISOLation1+** 编码。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`。而如果你的 GMT 默认字符编码为 **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。

4.7.2 使用 12 号或 34 号字体输入特殊字符

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

octal	0	1	2	3	4	5	6	7
\04x		!	∀	#	∃	%	&	∋
\05x	(	)	*	+	,	−	.	/
\06x	0	1	2	3	4	5	6	7
\07x	8	9	:	;	<	=	>	?
\10x	≅	Α	Β	Χ	Δ	Ε	Φ	Γ
\11x	Η	Ι	ϑ	Κ	Λ	Μ	Ν	Ο
\12x	Π	Θ	Ρ	Σ	Τ	Υ	ς	Ω
\13x	Ξ	Ψ	Ζ	[	∴	]	⊥	_
\14x	‾	α	β	χ	δ	ε	φ	γ
\15x	η	ι	φ	κ	λ	μ	ν	ο
\16x	π	θ	ρ	σ	τ	υ	ϖ	ω
\17x	ξ	ψ	ζ	{	|	}	~	

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

ZapfDingbats

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

\24x		❡	❢	❣	❤	❥	❦	❧
\25x	♣	♦	♥	♠	①	②	③	④
\26x	⑤	⑥	⑦	⑧	⑨	⑩	❶	❷
\27x	❸	❹	❺	❻	❼	❽	❾	❿
\30x	①	②	③	④	⑤	⑥	⑦	⑧
\31x	⑨	⑩	❶	❷	❸	❹	❺	❻
\32x	❼	❽	❾	❿	➔	→	↔	↕
\33x	➘	➙	➚	➛	➜	→	→	➡
\34x	➡	➡	➢	➢	➢	➡	➡	➡
\35x	➡	⇨	⇨	⇨	⇨	⇨	⇨	⇨
\36x		⇨	⇨	➡	➡	➡	➡	➡
\37x	➡	➡	➡	➡	➡	➡	⇛	

图 3: 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，例如字符 λ 对应的八进制码为 `\154`。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

4.7.3 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 **ISOLatin1+** 表中找到自己需要的字符，然后使用其对应的八进制吗

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

4.7.4 小练习

使用 **ISOLatin1+** 表中的八进制码输入字符 ± 234:

```
echo 5 2 \261 234 | gmt text -Jx1c -R0/10/0/4 -Baf -pdf test
```

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. 使用两个反斜杠来代替一个反斜杠
echo 5 2 \\261 234 | gmt text -Jx1c -R0/10/0/4 -Baf -pdf test

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

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

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

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 text -JX10c/10c -R0/10/0/10 -F+f15p,12 -pdf test
```

如果需要在一堆正常字符中插入一个 Symbol 表中的特殊字符，可以使用转义字符（下一节会详细介绍）**@~** 或 **@%12%** 临时将字体 Symbol 字体。例如:

```
gmt basemap -R0/10/0/10 -JX10c/10c -BWSne -Bx2+l"T@~\161@~t" -pdf test
```

4.8 转义序列

前面介绍了如何设置文字的文字大小、字体和颜色以及如何输入特殊字符。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。

4.8.1 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 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
@@	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: 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	Ç	**@c**	ç
@N	Ñ	**@n**	ñ
@U	Ü	**@u**	ü
@s	ß	**@i**	í

4.8.2 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	abcδεϕghi
2@~p@~r@+2@+h@-0@-	$2\pi r^2 h_0$
S@#mall@# C@#aps@#	SMALL CAPS
Thi is @_underline@_	This is underline
@%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 @@	ℭ @
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 text -R0/10/0/3 -Jx1c -Bxaf+l"Label" -BWSen -F+f20p -pdf test
```

4.8.3 注意事项

1. 上标/下标不支持嵌套，即只支持一层上标/下标
2. *text* 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
3. 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
4. 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%**。

4.9 绘制矢量/箭头

矢量是一个有大小和方向的量，通常用箭头 表征矢量。一个矢量由两个独立的部分组成：矢量线和矢量头。矢量线与一般的线没有区别，通常由画笔属性（*画笔*）控制。这一节则主要介绍矢量头的属性及控制方式。

GMT 中能够绘制矢量的模块有*plot*、plot3d、*grdvector*、*velo* 、*rose*、*ternary* 和*polar*。其中最常见的绘制矢量的模块是*plot*。

下面的命令使用*plot* 的 **-Sv** 选项绘制了一个最简单的矢量。

```
echo 1 1 0 3 | gmt plot -R0/5/0/2 -JX5c/2c -Sv0.5c+e -W1.5p -Gred -png vector
```

接下来将介绍如何通过在 **-Sv0.5c** 后增加更多的子选项来进一步修改矢量头的属性。

4.9.1 矢量头位置及形状

在绘制矢量时，默认只绘制矢量线而不绘制矢量头。下面的几个子选项用于指定矢量头的位置以及形状：

- **+b**[**t**|**c**|**a**|**A**|**i**|**I**][**l**|**r**]：在矢量线的起点加上矢量头
- **+e**[**t**|**c**|**a**|**A**|**i**|**I**][**l**|**r**]：在矢量线的终点加上矢量头
- **+m**[**f**|**r**][**t**|**c**|**a**][**l**|**r**]：在矢量线的中间加上矢量头

  t|**c**|**a**|**A**|**i**|**I** 用于控制矢量头的形状。取不同值所对应的矢量头形状如下图所示：

  Source Code

  - **l**|**r** 表示只绘制矢量头的左半边或右半边（默认两边都绘制）。其中左定义为从矢量起点看向终点时的左侧
  - **f**|**r** 在 **+m** 中用于指定矢量头的方向沿着正方向或逆方向（默认为正方向，即从起点指向终点）

  注解： **+m** 不能与 **+b** 或 **+e** 一起使用

- **+l** 只绘制左半个矢量头
- **+r** 只绘制右半个矢量头
- **+a***angle* 用于控制矢量箭头的顶端的夹角，默认值为 30 度。若矢量头形状为 **t** 或 **c** 则表示端点线或端点圆圈的大小
- **+h***shape* 进一步控制矢量头 **a** 的形状，其中 *shape* 可以取-2 到 2 之间的值。设置该子选项的值等效于修改参数*MAP_VECTOR_SHAPE*（默认值为 0）

  下面展示了 **+h** 后取不同值的矢量箭头的形状：

  Source Code

4.9.2 矢量头线条颜色和填充色

默认情况下，**-W** 选项同时控制矢量线和矢量头的画笔属性，**-G** 选项则控制矢量头的填充色。下面两个子选项可以单独控制矢量头的画笔属性和填充色。

- **+p***pen* 设置矢量头的画笔属性，*pen* 为 **-** 表示不绘制矢量头的轮廓
- **+g***fill* 设置矢量头的填充色，*fill* 为 **-** 表示不填充

4.9.3 其它属性

还可以使用如下子选项进一步控制矢量头的属性：

- **+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，这可能会出现矢量线很短而矢量头过大，或者矢量线很长而矢量头过小的情况。该子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* 用于增长或缩短矢量线首端或尾端的长度。其中 **b** 表示首端，**e** 表示尾端。*trim* 为正值表示缩短矢量线，为负值表示增长矢量线。也可以直接使用 **+t***trim1*/*trim2* 分别为首端和尾端指定增长/缩短量。

4.9.4 矢量类型及输入数据格式

GMT 提供了三类矢量：

- 笛卡尔矢量：矢量起点到终点的矢量线以直线方式连接
- 地理矢量：矢量起点到终点之间的矢量线以大圆弧路径连接
- 弧形矢量：矢量线是以某一点为中心的一段圆弧

plot 模块中：

- **-Sv** 或 **-SV** 用于绘制笛卡尔矢量
- **-S=** 用于绘制地理矢量
- **-Sm** 用于绘制弧形矢量

下面的例子分别绘制了三种矢量：

```
gmt begin arrows pdf,png
# 笛卡尔矢量
echo 0.5 1.5 4.5 1.5 | gmt plot -R0/5/0/5 -JX1.75i -Sv0.2i+s+b+e -W1.5p -Gred
# 地理矢量
echo 10 -35 90 8000 | gmt plot -R0/90/-41.17/41.17 -JM1.75i -S=0.2i+b+e -W1.5p -Gred -X2i
# 弧形矢量
echo 0.5 0.5 0.9i 0 90 | gmt plot -R0/5/0/5 -JX1.75i -Sm0.2i+b+e -W1.5p -Gred -X2i
gmt end
```

图 4: 三种矢量：(左) 笛卡尔矢量；(中) 地理矢量；(右) 弧形矢量

笛卡尔矢量和地理矢量

对于笛卡尔矢量和地理矢量，输入数据的格式为：

```
X    Y    angle  length
```

其中 X 和 Y 是矢量起点坐标，*angle* 为矢量方向（相对于水平方向逆时针旋转的角度或相对于北向顺时针的方位角），*length* 为矢量线长度（长度单位为 **c**|**i**|**p** 或 km）。

使用如下子选项则可以指定其它输入数据格式：

- **+s** 表示将输入数据的中的 *angle* 和 *length* 解释为矢量的终点坐标

- **+j**[**b**|**c**|**e**] 表示将输入数据中的 X 和 Y 坐标解释为矢量的起点坐标、中点坐标或终点坐标

对于地理矢量，还可以使用如下选项：

- **+q** 表明输入数据中的 *angle* 和 *length* 表示相对于某个指定极点的大圆路径的起始和结束的角度。该极点默认为北极，可以使用 **+o** 修改该点的位置
- **+o***plon*/*plat* 用于指定 **+q** 中大圆路径中的倾斜极点

对于笛卡尔矢量而言，还可以使用：

- **+z***scale*[*unit*] 表示输入数据中的 *angle* 和 *length* 被解释为矢量的 X 分量 dx 和 Y 分量 dy，然后计算出对应的极坐标下的方向和长度，并将矢量长度乘以 *scale*

圆弧矢量

对于圆弧矢量，输入数据中需要给定圆弧对应的圆心坐标、半径，以及圆弧起始和结束的角度。

4.10 线条属性

对于线条而言，其基本的画笔属性由三个：线宽、颜色和线型，在画笔 中已经做过介绍。除此之外，某些模块还可以为线条设置额外的属性，这些额外的属性可以通过在画笔属性后加上子选项来是实现。

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** 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 **c**|**i**|**p** 或距离单位

下面的示例展示了 **+o** 子选项的用法和使用效果。图中，细线和粗线使用了相同的输入数据，其中细线没有使用 **+o**，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

图 5: 线段起点偏移示意图

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

图 6: 线条自动样条插值示意图

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段绘制端点箭头。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
- *vspecs* 用于指定端点箭头的属性，详见绘制矢量/箭头

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+pfaint+bc+ve0.3i+gblue**

Source Code

图 7: 线条端点箭头示意图

4.10.4 其它属性

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

锚　是船舶停泊时固定船只使之不能漂走的工具。GMT 中的锚点也具有类似的作用，用于将某个元素固定在图中的某个位置。这一节将介绍 GMT 中锚点的概念，具体的使用场景及使用方法将在下一节介绍。

对于任意一个矩形元素，GMT 为其定义了 9 个锚点。每个锚点的位置用一个水平位置代码和一个垂直位置代码组合定义得到。水平位置代码可以取 `L`、`C` 或 `R`，分别表示左中右；垂直位置代码可以取 `T`、`M` 或 `B`，分别表示上中下。3 个水平位置代码与 3 个垂直位置代码自由组合，得到 9 个锚点，每个锚点均对应矩形元素的某个特定位置，如下图中红点和红字所示。例如，锚点 `BL` 位于矩形元素的左下角，而锚点 `MC` 则位于矩形元素的中心。

Source Code

此处的矩形元素并不一定是一个真正的矩形，GMT 中很多绘图元素都可以抽象为一个矩形元素。例如常规的矩形底图、非矩形的地理底图、比例尺、色标、指南针、文本字符串等，都可以抽象为一个矩形元素。

例如，对于一个非矩形的地理底图来说，其 9 个锚点的位置如下图所示：

Source Code

指南针、比例尺、图例、色标、文本字符串等也可以抽象为一个矩形，也有自己的锚点。下图展示了色标的 9 个锚点的位置：

Source Code

4.12 绘制修饰物

GMT 除了可以绘制线段、符号、矢量外，还可以绘制如下修饰物：

1. 地图比例尺
2. 色标
3. 图例
4. 方向玫瑰图
5. 磁场玫瑰图
6. 叠加图片
7. GMT logo
8. 小图 (map inset)

这 8 个修饰物可以用不同的模块绘制，且有各自的语法。所有这些修饰物都可以抽象为一个矩形元素。这一节介绍一种通用的机制，以指定这些修饰物在图上的位置以及修饰物的背景面板。

4.12.1 定位

绘制修饰物的模块大多有一个选项用于指定修饰物在底图上的位置，其基本语法为:

```
[g|j|J|n|x]<refpoint>[+j<justify>][+o<dx>[/<dy>]]
```

为了精确地将修饰物放在底图上的任意一点，GMT 需要做四件事:

1. 通过 `[g|j|J|n|x]<refpoint>` 指定底图上的一个**参考点**
2. 通过 `+j<justify>` 指定修饰物上的一个**锚点**
3. 通过 `+o<dx>[/<dy>]` 对修饰物的锚点做偏移
4. 将修饰物的偏移后的锚点与底图上的参考点重合以实现修饰物的放置

下图展示了 GMT 如何实现修饰物的精确放置。其中大矩形代表底图，小矩形代表修饰物。本示例中首先指定了底图左上角为参考点，并指定了修饰物的左上角为锚点，通过 `+o<dx>/<dy>` 对修饰物锚点进行微调。

Source Code

底图参考点

GMT 提供了 5 种方法用于指定底图的参考点，使得用户可以灵活地指定底图内/外的任意一点作为参考点。五种方法分别以 `x|g|n|j|J` 开头。若未指定使用何种方式，则默认使用 `x` 方式指定参考点。

通过坐标指定参考点

数据坐标 用 `g<lon>/<lat>` 指定参考点的数据坐标。例如 `g135/20` 表示参考点的坐标为 `135/20`

绘图坐标 用 `x<X>/<Y>` 指定参考点的绘图坐标，即给出参考点相对于绘图原点的偏移距，例如 `x4c/2c` 表示参考点位于底图原点的右侧 4 厘米，上方 2 厘米处

归一化坐标 用 n<X>/<Y> 指定参考点的归一化坐标。归一化坐标是指，底图所对应的矩形的 X 轴范围为 0 到 1，Y 轴同理。使用负值或者大于 1 的值则可以指定底图外的一点为参考点。比如 n0.2/0.1

通过锚点指定参考点

可以用 j<anchor> 或 J<anchor> 的方式指定底图的某个锚点作为底图参考点。其中锚点 <anchor> 有 9 个可能的取值，在锚点 一节中已经详细介绍过。例如 jTL 指定了底图左上角为参考点。

使用 j<anchor> 或 J<anchor> 方式指定底图参考点的同时，也设置了修饰物上默认使用的锚点位置。使用 j<anchor> 方式修饰物的默认锚点与参考点锚点相同。例如 jTL 指定底图左上角为参考点，同时也指定了修饰物左上角为其锚点，即底图左上角与修饰物左上角重合，则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。

使用 J<anchor> 方式，修饰物的默认锚点是与底图参考点锚点呈镜像相反。例如 JTL 指定底图的左上角作为参考点，同时指定了修饰物的默认锚点是 BR 即右下角（镜像对称）。因而底图的左上角与修饰物的右下角重合，则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。

修饰物锚点

在指定底图参考点后，可以进一步用 +j<anchor> 指定修饰物上的锚点。

若未使用 +j<anchor> 指定锚点，则修饰物的锚点按如下规则取默认值：

1. 若使用 j<anchor> 方式指定底图参考点，则修饰物锚点与底图锚点相同
2. 若使用 J<anchor> 方式指定底图参考点，则修饰物锚点使用与底图参考点镜像相反的锚点
3. 若使用其它三种方式指定参考点，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

指定底图参考点和修饰物锚点后，即可将二者重合起来，实现修饰物在底图上的定位。在此基础上，可以额外使用 +o<dx>/<dy> 指定修饰物锚点的偏移量。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，因而就需要为修饰物锚点指定额外的偏移量以增加定位的灵活性。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。

4.12.2 背景面板

对于任意一个修饰物，GMT 都可以为其绘制一个背景面板。背景面板的位置和大小由修饰物直接决定。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>]]
```

- +g<fill> 指定面板填充色，默认不填充
- +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
- +r<radius> 绘制圆角边框，<radius> 为圆角的半径
- +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,dahsed，默认使用*MAP_DEFAULT_PEN*
- +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
- +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 8: 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+glightblue+c0.1i (不包含最内侧虚线框)

4.13 数据类型

GMT 可以绘制笛卡尔坐标轴、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，需要的数据类型也不同。GMT 所支持的数据类型主要分为四大类:

- 地理坐标
- 绝对时间坐标
- 相对时间坐标
- 一般浮点数

4.13.1 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。

4.13.2 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`

4.13.3 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`

4.13.4 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`

第 5 章　表数据

表数据，也称为列数据或多列数据，英文为 table data，常用于表示点和线。

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，其具有直观、可直接编辑的特点；二进制表和 netCDF 表相对不直观，但通常文件大小更小。

5.1 ASCII 表

5.1.1 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。ASCII 表常用于表示点或线。表数据中有 N 行 M 列，每行称为一个记录，每列称为一个字段。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，

5.1.2 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。可以使用*-h 选项* 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
```

5.1.3 多段数据

以绘制断层为例，在绘制断层的时候，可以将每条断层线的经纬度信息分别放在单独的文件中。但当断层数量很多时，这样做会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。

数据段头记录总是以某个特定的字符作为开头，GMT 默认的段头记录的标识符为 >，即所有以 > 开头的行都会被认为是一个段头记录，其标志着新一段数据的开始。

下面是一个包含两个数据段的多段数据，每段数据分别构成一个多边形:

```
>
10  20
15  30
5   25
>
20  20
35  30
```

(下页继续)

(续上页)

```
40  40
35  40
```

可以使用如下命令绘制上面的多段文件:

```
gmt plot input.dat -JX10c -R0/50/0/50 -Baf -W1p -Gred -pdf lines
```

用户可以通过设置*IO_SEGMENT_MARKER* 将段头记录的标识符修改为任意字符，例如 @、% 等。

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

5.1.4 数据段头记录中的额外属性

数据段头记录不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。比如在绘制线段时，可以在段头记录中加上如下一系列选项，以分别控制每段线段的属性:

- -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
- …

下面的多段数据，分别设置两段数据拥有不同的画笔颜色和填充色:

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

可以使用如下命令绘制上面的多段数据:

```
gmt plot input.dat -JX10c -R0/50/0/50 -Baf -W1p -Gred -pdf lines
```

查看绘图结果可以发现，数据段头记录中的选项参数会覆盖命令行中相应选项的值。

5.2 二进制表

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。二进制表的文件大小通常比对应的 ASCII 表小很多，且读写速度更快。对于有大量输入输出需求的任意而言，可以将表数据以二进制表的形式保存，以提高数据文件的读写效率。

GMT 在读取二进制表数据时，是无法猜测数据的具体格式的。因而需要使用*-b 选项*告诉 GMT 要读入的数据中有多少个字段，每个字段的数据类型等。

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。二进制表中通常用一个值为 NaN 的记录作为数据段头记录来标记每段数据的开始。

5.3 netCDF 表

表数据也可以用 netCDF 格式保存。该格式的好处在于通用性。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。

5.4 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，已经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）
- MULTIPOINT: 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: 多个多边形数据（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:

```
# @Je4326 @JgX @Jp"+proj=longlat +ellps=WGS84+datum=WGS84 +no_defs"
# @Jw"GEOGCS[\"WGS84\",DATUM[\"WGS_1984\",SPHEROID\"WGS84\",6378137,\
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、interger、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=longlat +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
```

(下页继续)

(续上页)

```
178.2 -45.4
>
# @P
...
```

保护环境, 从阅读电子文档开始!

第 6 章　网格文件

6.1 网格文件格式

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 支持的网格数据格式主要分为四大类:

1. netCDF4 标准网格文件格式
2. netCDf3 “旧” 网格文件格式
3. GMT 自定义的二进制文件格式
4. 其他软件自定义的网格文件格式

其中最常见也最推荐使用的网格数据格式是 netCDF4 标准网格文件格式。GMT 默认使用该格式, 一般以 `.nc` 或 `.grd` 作为文件后缀。本章的介绍也将只针对 netCDF 格式。

四大类网格文件格式进一步可细分为更多的数据格式。GMT 所支持的所有网格文件格式在下表列出, 每种网格文件格式均对应一个两字符 ID。

表 1: GMT 支持的网格文件格式

ID	说明
	GMT netCDF4 标准格式
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)
	GMT netCDF3 "旧"格式
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)
	GMT 自定义二进制格式
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)
	其他网格文件格式
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 GMT 源码中的 `gmt_customio.c`。

6.2 读 netCDF 文件

netCDF 格式的设计相当灵活，单个 netCDF 文件中可以包含**多个多维变量**。而 GMT 只能直接处理包含单个二维变量的 netCDF 文件。因而，对于单变量二维 netCDF 文件，直接给出文件名即可；而对于复杂的多变量多维 netCDF 文件，则需要用户在指定文件名

时给出额外的信息。

6.2.1 读二维单变量 netCDF 文件

对于二维单变量 netCDF 文件，用户只需要直接给出网格文件的文件名，GMT 可以自动检测 netCDF 文件的格式并读入。具体来说，可以按照如下格式指定网格文件文件名:

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

其中

- `<name>` 是网格文件名，必须指定，其它均是可选项
- `<ID>` 显式告诉 GMT 当前文件的格式 ID
- `<scale>` 将数据乘以比例因子 `<scale>`，默认值为 1
- `<offset>` 将数据加上一个常数 `<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。在读入网格文件时，会先乘以比例因子再加上偏移量。

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

6.2.2 读取二维多变量 netCDF 文件

对于包含多个二维变量的 netCDF 网格文件，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。用户可以通过在网格文件名后加上后缀 `?<varname>` 的方式指定要读取某个特定的二维变量，即:

```
<filename>?<varname>
```

其中 `<varname>` 是 netCDF 文件中包含的变量名，其可以通过 netCDF 提供的命令 `ncdump -c file.nc` 得到。

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"
```

注解: 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。

6.2.3 读取三维单/多变量 netCDF 文件

最常见的三维单/多变量 netCDF 文件是地震成像得到的三维地球参考模型。三个维度分别是经度、纬度和深度，变量通常是 P 波速度、S 波速度等。

在遇到三维单/多变量 netCDF 文件时，GMT 默认只读取第一个变量的第一层数据（通常是深度值最小的那一层）。此时，可以将其当做一个变量数组，并通过如下两种方式指定读取特定层的数据。

1. 变量名后加上 `[<index>]` 以指定某一层的索引值。第一层的索引值为 0，第二层的索引值为 1，依次类推。
2. 变量名后加上 `(<level>)` 以指定获取深度为 `<level>` 处值。若网格文件中在 `<level>` 指定的深度处并不存在数据，则 GMT 会找到离 `<level>` 最近的有数据的那一深度的值，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（下面只列出与深度有关的部分）：

```
# 前面省略部分内容
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。则可以使用如下命令：

```
# 读取第二层(即深度 100km)处的 P 波速度
gmt grdinfo "file.nc?vp[1]"

# 读取深度 200 千米处的 P 波速度
gmt grdinfo "file.nc?vp(200)"
```

注解: Linux 下问号、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

6.2.4 读取四维单/多变量 netCDF 文件

对于四维单/多变量 netCDF 文件，处理方法类似。假设有一个四维单变量 netCDF 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` 可以查看四个纬度的取值范围:

```
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
```

此时可以将变量 `<pressure>` 当成一个二维数组。

为了得到 depth=10，time=24 处的变量值，可以用:

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

或者:

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

在本例中，时间维度在前，深度维度在后。

6.2.5 读取一维单/多变量 netCDF 文件

一维单/多变量 netCDF 文件，即前面所说的以 netCDF 格式保存的表数据。即表数据中的每一列分别保存为 netCDF 文件中的一个变量。GMT 自带的 GSHHG 数据和 DCW 数据就是一维多变量 netCDF 文件。

同样的，可以使用 `ncdump -c file.nc` 来查看 netCDF 文件所包含的变量名。然后即可通过在文件名后加上一系列用斜杠分隔的变量名来使用这些一维变量。例如:

```
# 将文件中的 lon 变量和 lat 变量作为输入数据的第 1 和 2 列
gmt plot "file.nc?lon/lat" ...
gmt plot "file.nc?lon/lat" ...

# 将文件中的变量 time、lat 和 lon 分别作为输入数据的三列
gmt convert "file.nc?time/lat/lon" ...
```

如果要使用的变量是一个二维变量，并且其优先维度与其他被选变量相同，则该变量整体会被输出。例如，一个 netCDF 文件中包含 6 个时间步，其记录了 4 个点的温度。则变量 `temp` 是一个 6x4 的数组，因而使用如下命令会输出如下信息:

```
$ gmt convert "file.nc?time/temp
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
2012-06-27T12:00:00 27.2 27.2 27.5 27.5
```

如果只需要某个点的温度，例如第二列数据，则可以使用:

```
$ gmt convert "file.nc?time/temp[1]
```

6.2.6 修改坐标单位

某些 GMT 模块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。

6.3 写 netCDF 文件

6.3.1 文件名格式

GMT 生成网格数据时默认使用 `nf` 格式（默认网格文件格式由 GMT 参数*IO_GRIDFILE_FORMAT* 控制）。可以通过在网格文件名后加上额外的信息以指定网格文件的格式。网格文件的文件名格式为:

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

- `<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
- `<offset>` 将数据加上一个常数 `<offset>`，默认值为 0
- `<scale>` 将数据乘以比例因子 `<scale>`，默认值为 1
- `<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。在写网格文件时，会先加上偏移量，再乘以比例因子。

6.3.2 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，会启用分块功能。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，无需读取整个数据。

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。

6.4 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 $x = x_{min}, x_{min} + x_{inc}, x_{min} + 2 \cdot x_{inc}, \ldots, x_{max}$ 和 $y = y_{min}, y_{min} + y_{inc}, y_{min} + 2 \cdot y_{inc}, \ldots, y_{max}$ 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 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。

6.4.1 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。

6.4.2 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。

6.5 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：

6.5.1 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。

6.5.2 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。

6.5.3 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

6.6 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。

保护环境, 从阅读电子文档开始!

第 7 章　CPT 文件

CPT 全称是 Color Palette Table，也称为调色板或色标文件。

CPT 文件定义了数值与颜色之间的映射关系，对于每一个数值，都可以从 CPT 文件中查找到该数值对应的颜色。因而，CPT 文件常用于绘制 2D 网格文件、随着数值变化颜色的线条或随着数值变化颜色的符号等。

7.1 CPT 类型

CPT 文件可以分成两类，一种适用于分类数据，一种适用于常规数据。下面会逐一介绍两种类型的 CPT 文件的格式，以及其如何实现数值与颜色之间的映射。

7.1.1 分类 CPT

分类 CPT 文件适用于分类数据。所谓分类数据，是指数据只能取有限个离散的值。比如将陆地分为不同的类型：沙漠、森林、冰川等等。定义 1 代表沙漠，2 代表森林，3 代表冰川。显然值取 1.5 是没有意义的。

分类 CPT 文件的格式为:

```
<key1>      <fill1>     [;<label1>]
<key2>      <fill2>     [;<label2>]
...
<keyn>      <filln>     [;<labeln>]
B   <fill_background>
F   <fill_foreground>
N   <fill_nan>
```

其中：

- `<key>` 是分类数据的可能取值。`<key>` 必须单调递增但可不连续
- `<fill>` 是相应数据的填充色。可以是颜色，也可以是图案，见填充
- `<label>` 则是数据的标签（可选）。通常设置为类型名，其主要有两个用途：让用户知道每一个可能取值代表的具体类型；在使用*colorbar* 绘制色标时可以将标签显示到色标旁边。

`<key>` 还可以取以下特殊值：

- `B`：定义小于 `<key1>` 的值的填充色，即背景色，默认值由*COLOR_BACKGROUND* 控制
- `F`：定义大于 `<keyn>` 的值的填充色，即前景色，默认值由*COLOR_FOREGROUND* 控制

- N: 定义值为 NaN 时使用的填充色，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
```

从中可以看出 CPT 文件是如何完成数据与颜色之间的映射:

- 当数据值为 0 时，其填充色为黄色，代表沙漠
- 当数据值为 1 时，其填充色为绿色，代表森林
- 当数据值为 2 时，其填充色为红色，代表冰川

7.1.2 常规 CPT

常规 CPT 文件适用于连续变化的数据。对于连续变化的数据，无法为每个可能的数据值都指定颜色，通常的做法是为某些特定值指定颜色，其它值的颜色则通过插值得到。

常规 CPT 文件的格式为:

```
<z0>     <color_min_1>  <z1>   <color_max_1>    [L|U|B]    [;<label>]
<z1>     <color_min_2>  <z2>   <color_max_2>    [L|U|B]    [;<label>]
...
<zn-1>   <color_min_n>  <zn>   <color_max_n>    [L|U|B]    [;<label>]
B   <fill_background>
F   <fill_foreground>
N   <fill_nan>
```

常规 CPT 文件中的每一行均定义了一个 Z 值范围所对应的颜色范围。以第一行为例，其定义了数据值为 `<z0>` 时的颜色为 `<color_min_1>`，数据值为 `<z1>` 时的颜色为 `<color_max_1>`，而数据值位于 `<z0>` 到 `<z1>` 之间时颜色则通过对 `<color_min_1>` 和 `<color_max_1>` 线性或非线性插值得到。由此即定义了任意数值与颜色之间的映射关系。

在每一行后面，还可以加上两个可选项:

- `;<label>` 是 Z 值范围所对应的标签，使用 *colorbar* 模块绘制色标时加上 `-L` 选项则会将 `<label>` 作为色标的标注
- `L|U|B` 用于控制绘制色标时标注的放置位置。`L`、`U`、`B` 分别标注每个 Z 值范围的下限、上限或者同时标注 Z 值上下限

Z 值还可以取以下特殊值:

- B: 定义小于 `<z0>` 的值的填充色，即背景色，默认值由 *COLOR_BACKGROUND* 控制
- F: 定义大于 `<zn>` 的值的填充色，即前景色，默认值由 *COLOR_FOREGROUND* 控制
- N: 定义值为 NaN 时使用的填充色，默认值由 *COLOR_NAN* 控制

GMT 对于常规 CPT 文件的格式由如下要求：

1. Z 值必须单调递增
2. Z 值不能存在间断，即每行的最大 Z 值必须与下一行的最小 Z 值相同
3. B|F|N 语句要放在 CPT 文件的开头或结尾
4. 可以指定颜色也可以指定填充图案，若 <color_min_1> 指定了填充图案，则 <color_max_1> 必须设置为 -

常规 CPT 文件的格式是相对比较灵活的，其又可以进一步细分为连续型 CPT 和间断性 CPT：

连续型 CPT 当一行的最大 Z 值与下一行的最小 Z 值对应同一个颜色

例如，下面的示例 CPT 文件中在 Z 取 0.5 处颜色是连续的：

```
0       black   0.5     red
0.5     red         1   yellow
```

间断型 CPT 即某一行的最大 Z 值与下一行的最小 Z 值对应不同的颜色。

例如，下面的示例 CPT 文件中，第一行表明 Z 取 0.5 时为红色，第二行表明 Z 取 0.5 时为蓝色，即在 Z 取 0.5 处颜色存在间断：

```
0       black   0.5     red
0.5     blue    1       yellow
```

进一步，当某一行内最小 Z 值和最大 Z 值对应相同的颜色时则构成了单色间断型 CPT。例如，下面的示例 CPT 文件中，第一行指定了 Z 取 0 和 0.5 均定义为黑色，意味着 Z 取 0 到 0.5 之间的任意数值均为黑色。第二行则指定了 Z 取 0.5 到 1 之间的任意数值均为红色：

```
0       black   0.5     black
0.5     red     1       red
```

7.2 内置 CPT

GMT 内置了一个分类型 CPT 和几十个常规型 CPT 文件。这一节仅展示 GMT 内置 CPT 的配色，具体如何使用这些内置 CPT 文件会在后面做进一步介绍。

下面列出了 GMT 内置的若干个 CPT 文件的具体配色以及 CPT 文件名（位于图下方）。每张图中有两个色标，上面的色标是原始的 CPT 文件，可以使用如下命令绘制得到：

```
gmt colorbar -D0c/0c+w10c/1c+h -B0 -C<cpt> -pdf CPT
```

下面的色标是利用*makecpt* 对原始 CPT 文件做离散处理得到的单色型 CPT 文件，可以用如下命令绘制得到：

```
gmt begin CPT png,pdf
gmt makecpt -C<cpt> -T-1/1/0.25
gmt colorbar -C -D0c/0c+w10c/1c+h -B0
gmt end
```

图 1: abyss

图 3: batlow

图 5: bilbao

图 7: categorical

图 9: copper

图 11: cubhelix

图 13: davos

图 15: dem2

图 17: dem4

图 19: earth

图 21: etopo1

表 1 – 续上页

图 23: geo

图 25: gray

图 27: hawaii

图 29: hot

图 31: inferno

图 33: lajolla

图 35: lisbon

图 37: nighttime

图 39: ocean

图 41: oslo

图 43: panoply

图 45: polar

图 47: red2green

表 1 – 续上页

 图 49: roma	
 图 51: sealand	
 图 53: split	
 图 55: terra	
 图 57: tokyo	
 图 59: turku	
 图 61: viridis	
 图 63: wysiwyg	

GMT 内置 CPT 文件中，某些是针对特定的需求设计的，因而常用于特定的图件中：

1. 海深：abyss、bathy、gebco、ibcso
2. 陆地高程：elevation
3. 全球地形起伏：dem1、dem2、dem3、dem4、earth、etopo1、geo、globe、oleron relief、terra、topo、world
4. 地震成像：seis、polar
5. 分类型数据：categorical

7.3 制作 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 colorbar 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
SIZE=w15c/0.25c

gmt begin cpt_4 pdf,png
gmt makecpt -Chaxby -T-10/10/1 -H > Icpt.cpt
gmt makecpt -Chaxby -T-10/10/1 -H -Iz > Icpt_z.cpt
gmt makecpt -Chaxby -T-10/10/1 -H -Ic > Icpt_c.cpt
gmt makecpt -Chaxby -T-10/10/1 -H -Icz > Icpt_cz.cpt

gmt plot -R1/10/1/10 -JX20c -T

gmt colorbar -CIcpt.cpt   -Dn0.5/0.25+jCM+$SIZE+h+e+n -B+l"Master CPT"
gmt colorbar -CIcpt_c.cpt -Dn0.5/0.4+jCM+$SIZE+h+e+n  -B+l"-Ic"
gmt colorbar -CIcpt_z.cpt -Dn0.5/0.55+jCM+$SIZE+h+e+n -B+l"-Iz"
gmt colorbar -CIcpt_cz.cpt -Dn0.5/0.7+jCM+$SIZE+h+e+n -B+l"-Icz"

rm Icpt*.cpt
gmt end
```

注：这些 CPT 的调整选项顺序也是有优先级的。比如 -Iz 使用后，所有其他的更改都

是基于 z 值反向的前提进行的。

CPT 文件可以在*grdimage*、*plot*、plot3d 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。

7.3.1 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，现在有一个周期性 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 当作时周期性 CPT 文件。

Source Code

7.3.2 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
gmt begin cpt_3 pdf,png
gmt makecpt -Cglobe -T-8000/3000
gmt colorbar -C -Baf -Dx0/0+w4.5i/0.1i+h
gmt colorbar -Cglobe -Baf -Dx0/0+w4.5i/0.1i+h -Y0.5i
gmt end
```

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 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 值从以米为单位变换成 `<unit>`

7.5 CPT 资源

更多的 CPT 可以访问：

1. http://soliton.vm.bytemark.co.uk/pub/cpt-city/
2. http://www.fabiocrameri.ch/colourmaps.php

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 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
-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
-c	选择特定的子图
-d	将输入或输出中的 *nodata* 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
-j	确定球面距离的计算方式
-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 五种方式

可以用五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<code1>,<code2>,...[+r|R[<incs>]]

   通过直接指定区域代码 <code1>、<code2> 等间接指定研究区域范围。

   区域代码可以表示某个大洲、某个国家、某个州或省，详情见*DCW: 世界数字图表* 一节。

   +r|+R<incs> 用于对通过区域代码得到的研究区域范围进一步扩张或缩减。其中 +r 表示调整区域范围使得范围是 <incs> 的整数倍，+R 表示 <incs> 被解释为区域范围向外扩展的增量。其中 <incs> 有三种形式：

   1. <inc>：四个边具有相同的增量
   2. <xinc>/<yinc>：X 和 Y 方向具有不同的增量
   3. <winc>/<einc>/<sinc>/<ninc>：上下左右四个边具有不同的增量

   例如 -RFR+r1 表示取法国国界对应的区域范围并将其调整到最近的整数度数。
5. -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 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 六种坐标

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类型的数据来指定数据的范围。-R 选项可以使用五种不同的方式来指定数据范围：

1. 一般浮点坐标
2. 地理坐标
3. 绝对时间坐标
4. 相对时间坐标
5. 投影后坐标
6. 弧度坐标

其中地理坐标、浮点坐标、绝对时间坐标和相对时间坐标在数据类型 中已经介绍过，此处不再赘述。

地理坐标

对于常用的地图范围，GMT 提供了两个简化写法：

- -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 400km x 600km 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）。

弧度坐标

对于以弧度为单位的数据而言，通常会需要指定数据范围是 π 的倍数或分数，此时可以使用 [+|-][s]pi[f] 形式，其中 s 和 f 可以是任意整数或浮点数，s 表示 scale 即倍数，f 表示 fraction 即分数。

例如 -2pi/2pi3/0/1 表示 X 轴的数据范围是 -2π 到 $2/3\pi$，即相当于-360 度到 120 度。

当使用这种形式时，程序只在 π 相关的值处进行标注，即 n 倍的 π，1.5π 以及 3/4，2/3，1/2，1/3，1/4 倍的 π。

8.2 -J 选项

-J 选项用于指定坐标变换方式或地图投影方式，即将数据投影到画布上所采用的函数。

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|+|-]
```

其中，<δ> 和 <Δ> 用于指定投影代码，前者为小写字母，后者为大写字母。<parameters> 是零个或多个由斜杠分隔的投影参数，参数数目由投影方式决定。

投影代码使用小写字母时，-J 的最后一个参数 <scale> 表示底图比例尺，即图上距离与真实地球距离之间的换算关系。<scale> 可以有两种格式:

- 单个数字加单位，例如 2c，表示真实地球距离的 1 度投影到画布上为 2 厘米

- 1:xxxx 格式，例如 1:10000000 表示画布上的 1 厘米对应真实地球距离的 10000000 厘米

投影代码为大写字母时，-J 的最后一个参数 <width> 表示底图宽度。h|+|- 可进一步修改最后一个参数 <width> 的含义，其中：

- h 表示 <width> 为底图的高度
- + 表示 <width> 为底图维度中最大的那个维度的长度
- - 表示 <width> 为底图维度中最小的那个维度的长度

注解: 几乎所有地图投影方式，都只能指定底图宽度或高度中的任一个，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。

画图时通常建议使用大写投影代码以直接指定图片宽度，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 2: GMT 投影代码

-J 代码	说明
-JA$\mathrm{lon}_0/\mathrm{lat}_0[/\mathit{horizon}]/\mathit{width}$	Lambert azimuthal equal area
-JB$\mathrm{lon}_0/\mathrm{lat}_0/\mathrm{lat}_1/\mathrm{lat}_2\mathit{width}$	Albers conic equal area
-JC$\mathrm{lon}_0/\mathrm{lat}_0\mathit{width}$	Cassini cylindrical
-JCyl_stere/$[\mathrm{lon}_0[/\mathrm{lat}_0/]]\mathit{width}$	Cylindrical stereographic
-JD$\mathrm{lon}_0/\mathrm{lat}_0/\mathrm{lat}_1/\mathrm{lat}_2\mathit{width}$	Equidistant conic
-JE$\mathrm{lon}_0/\mathrm{lat}_0[/\mathit{horizon}]/\mathit{width}$	Azimuthal equidistant
-JF$\mathrm{lon}_0/\mathrm{lat}_0[/\mathit{horizon}]/\mathit{width}$	Azimuthal gnomonic
-JG$\mathrm{lon}_0/\mathrm{lat}_0[/\mathit{horizon}]/\mathit{width}$	Azimuthal orthographic
-JG$\mathrm{lon}_0/\mathrm{lat}_0\mathit{alt}/\mathit{azim}/\mathit{tilt}/\mathit{twist}/W/H/\mathit{width}$	General perspective
-JH$[\mathrm{lon}_0/]\mathit{width}$	Hammer equal area
-JI$[\mathrm{lon}_0/]\mathit{width}$	Sinusoidal equal area
-JJ$[\mathrm{lon}_0/]\mathit{width}$	Miller cylindrical

下页继续

表 2 – 续上页

-J 代码	说明
-JKf[lon_0/]*width*	Eckert IV equal area
-JKs[lon_0/]*width*	Eckert VI equal area
-JLlon_0/lat_0/lat_1/lat_2*width*	Lambert conic conformal
-JM[lon_0[/lat_0/]]*width*	Mercator cylindrical
-JN[lon_0/]*width*	Robinson
-JOalon_0/lat_0*azim*/*width*	Oblique Mercator, 1: origin and azim
-JOblon_0/lat_0/lon_1/lat_1*width*	Oblique Mercator, 2: two points
-JOclon_0/lat_0/lon_p/lat_p*width*	Oblique Mercator, 3: origin and pole
-JP[**a**]*width*[/*origin*]	Polar [azimuthal] (θ, r) (or cylindrical)
-JPoly[lon_0[/lat_0/]]*width*	(American) polyconic
-JQ[lon_0[/lat_0/]]*width*	Equidistant cylindrical
-JR[lon_0/]*width*	Winkel Tripel
-JSlon_0/lat_0[/*horizon*]/*width*	General stereographic
-JT[lon_0[/lat_0/]]*width*	Transverse Mercator
-JU*zone*/*width*	Universal Transverse Mercator (UTM)
-JV[lon_0/]*width*	Van der Grinten
-JW[lon_0/]*width*	Mollweide
-JX*width*[**l**|**p***exp*|**T**|**t**][/*height*[**l**|**p***exp*|**T**|**t**]][**d**]	Linear, $\log_{10}$, $x^a - y^b$, and time
-JYlon_0/lat_0*width*	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。`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<code>[<parameters>/]<scale>
-J<Code>[<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 3: Proj4 投影代码

代码	说明
-Jaea/lon_0/lat_0/lat_1/lat_2/*scale*	Albers conic equal area

下页继续

表 3 – 续上页

代码	说明
-Jaeqd/lon_0/lat_0[/*horizon*]/*scale*	Azimuthal equidistant
-Jcass/lon_0/lat_0/*scale*	Cassini cylindrical
-Jcea/lon_0/lat_0/*scale*	Cylindrical equal area
-Jcyl_stere/[lon_0[/lat_0/]]*scale*	Cylindrical stereographic
-Jeqc/[lon_0[/lat_0/]]*scale*	Equidistant cylindrical
-Jeqdc/lon_0/lat_0/lat_1/lat_2/*scale*	Equidistant conic
-Jgnom/lon_0/lat_0[/*horizon*]/*scale*	Azimuthal gnomonic
-Jhammer/lon_0/*scale*	Hammer equal area
-Jeck4/lon_0/*scale*	Eckert IV equal area
-Jeck6/lon_0/*scale*	Eckert VI equal area
-Jlaea/lon_0/lat_0[/*horizon*]/*scale*	Lambert azimuthal equal area
-Jlcc/lon_0/lat_0/lat_1/lat_2/*scale*	Lambert conic conformal
-Jmerc/[lon_0[/lat_0/]]*scale*	Mercator cylindrical
-Jmill/lon_0/*scale*	Miller cylindrical
-Jmoll/[lon_0/]*scale*	Mollweide
-Jnsper/lon_0/lat_0*alt/azim/tilt/twist/W/H/scale*	General perspective
-Jomerc/lon_0/lat_0*azim/scale*	Oblique Mercator, 1: origin and azimuth
-Jomerc/lon_0/lat_0/lon_1/lat_1/*scale*	Oblique Mercator, 2: two points
-Jomercp/:lon_0/lat_0/lon_p/lat_p/*scale*	Oblique Mercator, 3: origin and pole
-Jortho/lon_0/lat_0[/*horizon*]/*scale*	Azimuthal orthographic
-Jpolar/[**a**]*scale*[/*origin*]	Polar [azimuthal] (θ, r) (or cylindrical)
-Jpoly/[lon_0[/lat_0/]]*scale*	(American) polyconic
-Jrobin/[lon_0/]*scale*	Robinson
-Jsinu/lat_0/*scale*	Sinusoidal equal area
-Jstere/lon_0/lat_0[/*horizon*]/*scale*	General stereographic
-Jtmerc/[lon_0[/lat_0/]]*scale*	Transverse Mercator
-Jutm/*zone/scale*	Universal Transverse Mercator (UTM)
-Jvandg/[lon_0/]*scale*	Van der Grinten
-Jwintri/[lon_0/]*scale*	Winkel Tripel
-Jxy*xscale*[**l**|**p***exp*|**T**|**t**][/*yscale*[**l**|**p***exp*|**T**|**t**]][**d**]	Linear, $\log_{10}$, $x^a - y^b$, and time

8.2.3 GMT+PROJ4

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绘制。

-B 选项有两套语法，分别用于设置底图边框以及每条边的属性，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，具体写法在下面会进一步介绍
- +t<title> 指定当前底图的标题。该标题位于底图上方的中间位置。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。标题的文本属性由*FONT_TITLE* 控制。标题与上边框之间的距离由*MAP_TITLE_OFFSET* 控制。
- +g<fill> 为底图内部填充背景色，见填充
- +n 只绘制边框，而不绘制刻度线和标注
- +b 仅适用于 3D 底图，使用该子选项则会根据绘制 3D 底图的 12 条边
- +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

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]lrbtu
```

Source Code

二维底图（上图左图）有四条边，分别用东西南北（WSEN 或 wsen）或左右上下（lrtb）的单词首字母表示。每条边都有四种状态:

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 wsen => 绘制某条边，该边有刻度、无标注
4. 出现小写字母 lrtb => 绘制某条边，该边无刻度、无标注

下图给出了不同的 -B 选项绘制出来的效果图。读者可以修改如下命令中的 -B 选项来尝试不同搭配的效果:

```
gmt basemap -R0/4/0/4 -JX10c -BWS -pdf axes
```

Source Code

3D 底图有 12 条边（上图右图）。对于 3D 底图而言，Zzu 用于控制 Z 轴的绘制。

- Z 表示有刻度和标注
- z 表示有刻度无标注
- u 表示无刻度无标注

默认只绘制一条 Z 轴，可以额外加上 1234 的任意组合来表示要绘制哪些 Z 轴。其中 1 始终表示位于左下角的 Z 轴，其他 Z 轴按逆时针顺序编号。加上 +b 子选项则绘制全部 12 条边。

下图展示了 3D 绘图中 -B 选项的不同用法。读者可以修改如下命令中的 -B 选项来实现不同搭配的效果:

```
gmt basemap -R0/10/0/10/0/10 -JX5c -JZ5c -Bxaf -Byaf -Bzaf -BwesnZ+t'-BwesnZ' -p130/30 -pdf map
```

Source Code

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>[+a<angle>|n|p][+l|L<label>][+s<label>][+p<prefix>][+u<unit>]
```

以上语法也可以被拆分为两部分:

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

其中，

- p|s 表示一级属性 (primary) 和二级属性 (secondary)
- x|y|z 表示设置哪一条轴的属性
- <interval> 设置刻度、网格线、标注的间隔
- +a<angle>|n|p 用于设置标注的倾斜角度，其中 <angle> 是相对于水平方向的旋转角度，取值范围为-90 到 90。+an 等效于 +a90 即垂直标注，+ap 等效于 +a0 即平行标注。对于 Y 轴标注而言，不支持任意角度的标注，仅支持 +ap 和 +an。
- +l<label> 用于给指定的轴加标签。默认情况下，X 轴标签文字方向平行于 X 轴，Y 轴标签文字方向平行于 Y 轴。对于 Y 轴，可以使用 +L<label> 使得 Y 轴标签文字方向平行于 X 轴
- +s<label> 与 +l<label> 类似，也用于给指定的轴添加标签。当同时使用 +l<label> 和 +s<label> 时，前者用于指定左轴或下轴的标签，而后者用于指定右轴和上轴的标签。
- +p<prefix> 在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

x|y|z

x|y|z 用于指明要设置哪条边的属性，默认值为 xy，即同时设置 X 轴和 Y 轴的属性。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。例如:

```
-Bxaf -Byaf
-Bxyzaf
```

<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> 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），表明要设置轴的哪个属性的间隔
- <stride> 用于设置间隔，stride 为 0，表示不绘制
- ±<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，通常只在绘制时间轴时才使用

-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 basemap -JX10c/10c -R0/10/0/10 -Ba2f1g1 -pdf test
```

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

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。

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 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 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 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 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 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使用于全部轴。

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 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 begin GMT_-B_time1 pdf,png
gmt set FORMAT_DATE_MAP=-o FONT_ANNOT_PRIMARY +9p
gmt basemap -R2000-4-1T/2000-5-25T/0/1 -JX5i/0.2i -Bpa7Rf1d -Bsa1O -BS
gmt end
```

图 7: 时间轴示例 1

需要注意，-Bsa1O 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 begin GMT_-B_time2 pdf,png
gmt set FORMAT_DATE_MAP "o dd" FORMAT_CLOCK_MAP hh:mm FONT_ANNOT_PRIMARY +9p
gmt basemap -R1969-7-21T/1969-7-23T/0/1 -JX5i/0.2i -Bpa6Hf1h -Bsa1K -BS
gmt basemap -Bpa6Hf1h -Bsa1D -BS -Y0.65i
gmt end
```

图 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，月标注位于对应月的中间而不是三个月间隔的中间。

```
gmt begin GMT_-B_time3 pdf,png
gmt set FORMAT_DATE_MAP o FORMAT_TIME_PRIMARY_MAP Character FONT_ANNOT_PRIMARY +9p
gmt basemap -R1997T/1999T/0/1 -JX5i/0.2i -Bpa3Of1o -Bsa1Y -BS
gmt end
```

图 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 begin GMT_-B_time4 pdf,png
gmt set FORMAT_CLOCK_MAP=-hham FONT_ANNOT_PRIMARY +9p TIME_UNIT d
gmt basemap -R0.2t/0.35t/0/1 -JX-5i/0.2i -Bpa15mf5m -Bsa1H -BS
gmt end
```

图 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 begin GMT_-B_time5 png,pdf
gmt set FORMAT_DATE_MAP u FORMAT_TIME_PRIMARY_MAP Character \
       FORMAT_TIME_SECONDARY_MAP full FONT_ANNOT_PRIMARY +9p
gmt basemap -R1969-7-21T/1969-8-9T/0/1 -JX5i/0.2i -Bpa1K -Bsa1U -BS
gmt set FORMAT_DATE_MAP o TIME_WEEK_START Sunday FORMAT_TIME_SECONDARY_MAP Chararacter
gmt basemap -Bpa3Kf1k -Bsa1r -BS -Y0.65i
gmt end
```

图 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 begin GMT_-B_time6 pdf,png
gmt set FORMAT_DATE_MAP "o yy" FORMAT_TIME_PRIMARY_MAP Abbreviated
gmt basemap -R1996T/1996-6T/0/1 -JX5i/0.2i -Ba1Of1d -BS
gmt end
```

图 12: 时间轴示例 6

第七个例子：

```
gmt begin GMT_-B_time7 pdf,png
gmt set FORMAT_DATE_MAP jjj TIME_INTERVAL_FRACTION 0.05 FONT_ANNOT_PRIMARY +9p
gmt basemap -R2000-12-15T/2001-1-15T/0/1 -JX5i/0.2i -Bpa5Df1d -Bsa1Y -BS
gmt end
```

图 13: 时间轴示例 7

8.3.8 弧度轴π 的标注

如果坐标轴以弧度为单位，用户可以直接指定 π 的整数倍或分数倍作为标注间隔，其格式为 `[+|-][s]pi[f]` ，其中 s 表示标注间隔是 π 的 s 倍，而 f 表示标注间隔为 π 的 f 分之一。

示例:

```
gmt basemap -JX10c/5c -R-12pi/12pi/-1/1 -Bxa3pi -pdf test1
gmt basemap -JX10c/5c -R-pi/pi/-1/1 -Bxapi4 -pdf test2
```

8.3.9 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 begin GMT_-B_custom pdf,png
gmt basemap -R416/542/0/6.2831852 -JX-5i/2.5i -Bpx25f5g25+u" Ma" -Bpycyannots.txt -BWS+glightblue
gmt basemap -R416/542/0/6.2831852 -JX-5i/2.5i -Bsxcxannots.txt -BWS \
              --MAP_ANNOT_OFFSET_SECONDARY=10p --MAP_GRID_PEN_SECONDARY=2p
gmt end
rm -f [xy]annots.txt
```

图 14: 自定义坐标轴

8.4 -V 选项

-V 选项可以控制命令的 verbose 等级以输出不同等级的命令执行信息到标准错误流，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**：**q**uiet 模式，不输出任何错误和警告信息
- **-Vn**：**n**omral 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**：**c**ompatibility 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**：**v**erbose，即输出错误、警告以及数据处理的进程信息
- **-Vl**：**l**ong 模式，会额外输出详细的进程报告
- **-Vd**：**d**ebug 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。

8.5 -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[*label*][**+c**][**+j***just*][**+o***dx*/*dy*]

- **-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***label* 会在时间戳后打印字符串 *label*，比如 `-U"This is string"`
- **-U+c** 会在时间戳后打印当前命令
- **-U+j***just***+o***dx*/*dy* 用于控制将时间戳的哪个锚点 与当前底图的左下角对齐，以及其偏移量。例如，`-U+jBL+o0/0` 表示将时间戳的左下角与底图左下角对齐

GMT 参数中有一些可以控制时间戳：

- *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* 时间戳中文本字体

```
gmt basemap -R0/10/0/5 -JX10c/3c -Bx1 -By1 -U"This is a GMT logo" -png GMT_-U
```

图 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，因而一张图中只需要在一个绘图命令中使用 **-U** 即可。

注解： 修改 UNIX 环境变量 **TZ** 可以控制时间戳中时间所使用的时区。

8.6 -X 和 -Y 选项

`-X` 和 `-Y` 选项用于控制绘图的原点。

当需要在一张图上绘制多个子图时，通常可以使用 GMT 提供的子图模式。若子图的布局不规则，或对于子图原点有更复杂的需求，则需要使用 `-X` 和 `-Y` 选项移动子图的底图原点的位置。

`-X` 和 `-Y` 的用法类似。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> 是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
- -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
- -X 不接任何额外参数, 则继承前一个 GMT 命令使用该选项时的参数值

-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 begin test pdf
gmt basemap -JX5c/2c -R0/5/0/2 -B1
gmt basemap -B1 -X7c
gmt basemap -B1 -X-7c -Y4c
gmt basemap -B1 -X7c
gmt end
```

上图用四个 basemap 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, 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

实际绘图时会发现一些不方便的地方。比如 -X 和 -Y 的偏移量与前一张底图的大小息息相关。若修改了前一张底图的大小，则下一张底图的偏移量也需要相应修改。为解决这一问题，GMT6 为该选项引入了一种新的语法:

```
-X[+|-]w[[+|-|/]<xshift>[u]]
-Y[+|-]h[[+|-|/]<yshift>[u]]
```

其中 w 和 h 分别表示前一底图的宽度和高度。

看上去语法很复杂，举几个例子:

- -Yh+2c: 沿着 Y 轴上移，上移距离为**前一底图高度 +2 厘米**
- -Xw-2c: 沿着 X 轴右移，右移距离为**前一底图宽度-2 厘米**
- -Xw/2: 沿着 X 轴右移，右移距离为**前一底图宽度/2**
- -Y-h-2c: 沿着 Y 轴下移，下移距离**前一底图高度 +2 厘米**
- -X-w+2c: 沿着 X 轴左移，左移距离**前一底图宽度-2 厘米**

因而，上面的示例可以改写为更灵活的版本:

```
gmt begin test pdf
gmt basemap -JX5c/2c -R0/5/0/2 -B1
gmt basemap -B1 -Xw+2c
gmt basemap -B1 -X-w-2c -Yh+2c
gmt basemap -B1 -Xw+2c
gmt end
```

8.7 -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。

8.8 -b 选项

-b 选项用于控制二进制文件的输入/输出格式。

在表数据 一章中已经中已经介绍过，表数据可以是 ASCII 格式、二进制格式或 netCDF 格式。ASCII 格式很直观，可以直接看到有多少列数据。而二进制格式的数据，GMT 在读入数据时无法数据有多少列，也无法知道每一列的格式。因而需要使用 -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。<ncols><type> 表示有 <ncols> 列个 <type> 类型的数据。多个 <ncols><type> 可以用逗号分隔。每个 <ncols><type> 都可以 w 表示对这些列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。

其中 GMT 支持的数据类型 <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

若未指定 <ncols>，则假定 <ncols> 为程序所期望的数据列数，所有列均为 <type> 所指定的数据类型。

还可以使用-h 选项 跳过二进制文件开头的若干个字节。

举几个例子:

1. -bi2f,1i 表示要读入的二进制数据中有 3 列，前两列是单精度浮点型，最后一列为四字节有符号整型

8.9 -c 选项

该选项用于控制绘图命令在哪一个子图中进行绘制，该选项仅可在子图模式 下才能使用。所有绘图模块均可以使用该选项。该选项使得当前及之后的一系列绘图命令均在指定的子图中绘制，直到某个绘图命令再次使用该选项为止。

其有两种用法:

```
-c[<row>,<col>]
-c[<index>]
```

前者表示当前及接下来的绘图命令在第 <row> 行、第 <col> 列子图中绘制，后者则表示当前及接下来的绘图命名在第 <index> 个子图内进行。

<row>、<col>、<index> 均从 0 开始起算。

若只使用 -c 而不指定子图的行和列，则 GMT 会自动激活 “下一个” 子图面板。

以下示例展示了该选项的基本用法，其等效于在绘图命令前使用 subplot set 命令。

```
#!/bin/bash
R=0/5/0/3

gmt begin subplot pdf,png
gmt set FONT_TAG 15p,1

gmt subplot begin 2x2 -Fs5c/3c -A'(a)'+JTL+o0.2c/0.4c -M0.5c/0.2c
gmt basemap -R$R -Ba -BWSen -c

gmt basemap -R$R -Ba -BWSen -c

gmt basemap -R$R -Ba -BWSen -c
# 下面的命令未使用 -c 选项，但依然在 2 行 1 列子图中绘制
echo 2 2 | gmt plot -Sc0.5c -Gred -W1p

gmt basemap -R$R -Ba -BWSen -c

gmt subplot end
gmt end
```

8.10 -d 选项

-d 选项用于将某些特定值当作 NaN。

GMT 中用 NaN 表示数据缺失。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，因而需要在将数据传递给 GMT 时告诉 GMT 某个特定值表示数据缺失。

-d 选项的语法是:

```
-d[i|o]<nodata>
```

- -d<nodata> 将输入数据中的 <nodata> 替换成 NaN；同时，将输出数据中的 NaN 替换成 <nodata>；
- -di<nodata> 对于输入数据，将数据中的 <nodata> 替换成 NaN；
- -do<nodata> 对于输出数据，将数据中的 NaN 在输出时替换成 <nodata>。

8.11 -e 选项

GMT 命令在读入数据时默认会处理读入的所有数据记录。-e 选项的作用是筛选或排除匹配某个字符串或者正则表达式的数据记录。

-e 选项的用法有两种：

1. 匹配某个字符串：-e[~]"<pattern>"
2. 匹配某个正则表达式：-e[~]/<regexp>/[i]

在匹配字符串或正则表达式前加上 ~ 表示取反，即排除匹配字符串或正则表达式的数

据记录。如果数据记录中本身就包括字符 ~，则需要使用 \~ 对其进行转义。对于匹配正则表达式而言，还可以加上 i 表示忽略大小写。

以一个常见的应用场景举例。假如文件 input.dat 中有一堆数据点，每个数据都对应某个分类，文件内容如下:

```
1 1 type1
2 2 type1
3 3 type1
4 4 type2
5 5 type2
6 6 type2
7 7 type3
8 8 type10
9 9 type10
10 10 null
```

可以使用 -e 选项筛选出自己需要的数据记录。

筛选出所有匹配 type 的记录:

```
$ gmt select input.dat -e'type'
1   1       type1
2   2       type1
3   3       type1
4   4       type2
5   5       type2
6   6       type2
7   7       type3
8   8       type10
9   9       type10
```

排除所有匹配 null 的记录:

```
$ gmt select input.dat -e~'null'
1   1       type1
2   2       type1
3   3       type1
4   4       type2
5   5       type2
6   6       type2
7   7       type3
8   8       type10
9   9       type10
```

筛选所有匹配 type2 的记录:

```
$ gmt select input.dat -e'type2'
4   4       type2
5   5       type2
6   6       type2
```

筛选所有匹配 type1 的记录:

```
# 错误写法，因为 type1 也包含在字符串 type10 中
$ gmt select input.dat -e'type1'
1   1       type1
2   2       type1
3   3       type1
8   8       type10
9   9       type10

# 正确写法
# 此处使用了正则表达式，\ ``$`` 表示行末匹配
$ gmt select input.dat -e/type1$/
1   1       type1
2   2       type1
3   3       type1
```

正则表达式的具体用法不在本手册的范围之内，用户请自行搜索“正则表示式”。

8.12 -f 选项

-f 选项用于显式指定当前输入或输出数据中每一列的具体含义。其基本语法为:

```
-f[i|o]<colinfo>
```

默认情况下，该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。

<colinfo> 是一系列用逗号分隔的字符串。每个字符串包含两个部分：列号和数据类型，其指定了输入/输出数据中每一列的含义。

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 0:2:5 表示第 1、3、5 列数据；1:5 表示第 2 至 6 列数据。

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。

例如，-fi0y,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。

8.13 -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. 使用数据段头记录来标记一段数据的开始，详见*ASCII 表*
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][+n|+p]
```

- 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 <gap> 的单位
- 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> 表示两个数据的绝对值超过 <gap> 则分段
- <gap>+n 表示前一数据减去当前数据超过 <gap> 则分段
- <gap>+p 表示当前数据减去前一数据超过 <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。

8.14 -h 选项

-h 选项用于在读/写数据时跳过文件开头的若干个记录。其语法为:

```
-h[i|o][<n>][+c][+d][+r<remark>][+t<title>]
```

- <n> 表示要跳过的记录数
- i 和 o 分别表示仅对读数据或写数据时有效

对于输出数据，默认会将输入文件中的头段信息原样输出。使用如下子选项可以修改这一行为:

- +c 将列名写到输出的头段记录中
- +d 删除之前的头段信息
- +r<remark> 加一个 remarke 注释语句到输出文件的开头
- +t<title> 加一个 title 语句到输出文件的开头

几点注意事项:

1. 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。

2. 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。

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

8.15 -i 和 -o 选项

-i 和 -o 选项分别用于对输入和输出的数据进行列选择以及简单的代数运算。

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；或者某个命令的默认输出有很多列，却只想要输出其中的某几列。

-i 选项可以从输入数据中选择任意列，并对其进行四则运算以及取对数操作。其语法为：

```
-i<col>[+l][+s<scale>][+o<offset>][,...]
```

-o 选项用于输出指定的列。其基本语法为：

```
-o<col>[,...]
```

-i 和 -o 选项后接以逗号分隔的列号（列号从 0 开始）或列号范围，以指定输入/输出数据中需要保留的列及其顺序。列号范围的格式为 <start>[:<inc>]:[<stop>]，若省略 <inc> 则默认其值为 1。

对于 -i 选项而言，每个列号后还可以加上子选项以对每列数据进行简单的代数运算：

- +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>

举几个例子：

- -i3,6,2 表示读入数据中的第 4、7、3 列
- -o3,1,3 表示输出数据中的第 4、2、4 列，即第四列会被输出两次
- -i1-3,5 表示读入数据中的 2-4 列和第 6 列
- -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。

8.16 -j 选项

-j 选项用于控制球面上两点间距离的计算方式。其语法为：

```
-je|f|g
```

- `-jg` 将地球当做球体，用大圆路径公式计算球面距离，GMT 默认使用此方式
- `-jf` 使用 Flat Earth 公式计算球面距离
- `-je` 使用测地公式计算球面距离，计算距离时考虑了地球椭率

8.16.1 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：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。用户可以根据自己的需求选择适合的距离计算方式。

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。

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。

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。

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。

除了 Vincenty 完全椭球公式外，还可以将参数*PROJ_GEODESIC* 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。

8.17 -n 选项

-n 选项用于控制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-sphine 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), 默认打开抗混淆选项
- +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, 可以进一步加上 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, 以保证插值后的网格数据的范围不超过输入网格数据的范围
- +t<threshold> 用于控制 NaN 网格点的插值行为, 默认值为 0.5。具体解释见官方文档。

8.18 -r 选项

GMT 中的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 选项则可以显式指定输入/输出的网格文件的配准方式。其语法为:

```
-r[g|p]
```

其中, -rg 表示网格线配准, -rp 表示像素配准。

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

-p 选项用于控制 2D 底图或 3D 底图的透视视角。

对于一个 2D 底图 (比如二维平面内的矩形) 或者一个 3D 底图 (比如三维空间内的立方体), 从不同的方向看时会看到不同的形态。该选项用于控制从怎样的透视视角去看一张图。其语法为:

```
-p[x|y|z]<azim>[/<elev>[/<zlevel>]][+w<lon0>/<lat0>[/<z0>]][+v<x0>/<y0>]
```

<azim> 指定视角的方位角，即相对于北方向顺时针旋转的角度，默认值为 180。

<elev> 指定视角的海拔，即视角相对于纸张平面向上旋转的角度，其取值范围为 0 到 90（不含 0），其中 0 表示视角与纸张平面平行，90 表示视角垂直于纸张平面，默认值为 90。

下图给出了 2D 底图下透视视角取不同值时所看到的底图。

Source Code

对于 3D 底图而言（使用了 -JZ 选项），默认会在 Z 轴底部绘制 XY 平面的边框。

- 设置 <zlevel> 则可以指定在 Z=<zlevel> 平面内绘制边框。
- -px|y|z 则分别用于指定绘制 YZ 平面、XZ 平面或 XY 平面。默认为 -pz 即绘制 XY 平面

下图给出了 3D 底图上使用不同的 -p 选项绘制出来的效果图。绘制底图使用的命令为:

```
gmt basemap -R0/10/0/10/0/20 -JX3c -JZ3c -Baf -Bzaf -BwsENZ1+b -p135/40 -pdf 3D-map
```

Source Code

在绘制动画时，如果不同帧使用的 -p 透视参数不同，则由于投影的问题可能会导致底图在纸上的位置出现不规则运动。可以有三种方式解决：

1. 加上 + 使得数据范围的中心点固定在纸张的中心
2. 加上 +w<lon0>/<lat0>/<z> 将地图上的某个点固定在纸张的中心
3. 加上 +v<x0>/<y0> 使得数据范围的中心点或地图上的某个点，在投影后的位置始终固定在纸张上的 <x0>/<y0> 处

除此之外，若 -p 选项后不接任何参数，则其会继承前一 GMT 绘图命令的 -p 参数。

8.20 -s 选项

-s 选项用于控制是否输出含有 NaN 的记录。

默认情况下，命令会输出所有记录，包括那些某列值为 NaN 的记录。使用该选项可以控制含 NaN 的记录的输出。其语法为:

```
-s[<cols>][+a|+r]
```

- -s 若 Z 值（即第三列）为 NaN 则不输出
- -s+a 表示任意一列有 NaN 则不输出
- -s+r 表示反操作，即只输出某列有 NaN 的记录
- <cols> 用于指定要检查的列，即只有指定的所有列都为 NaN 时，才输出或不输出该记录。<cols> 是一系列用逗号分隔的列号或者列号范围。列号范围的格式为 <start>[:<inc>]:<stop>，若省略 <inc> 则默认其值为 1。比如 2,5,7 和 1,2:4，表示检测这些列数据是否全是 NaN。

举几个例子。输入数据 input.dat 的内容为:

```
1 1 1   0
2 2 NaN 0
3 3 3   NaN
```

不使用 -s 选项则会输出所有记录:

```
$ gmt select input.dat
1   1       1       0
2   2       NaN     0
3   3       3       NaN
```

使用 -s 选项则会抑制第三列为 NaN 的记录的输出:

```
$ gmt select input.dat -s
1   1       1       0
3   3       3       NaN
```

使用 -sa 选项则只有任意一列有 NaN 则不输出该记录:

```
$ gmt select input.dat -s+a
1   1       1       0
```

使用 -s2 选项则检查第三列 (列号从 0 开始) 是否为 NaN:

```
$ gmt select input.dat -s2
1   1       1       0
3   3       3       NaN
```

使用 -s2,3 则抑制第 3 和 4 列均为 NaN 的记录的输出:

```
$ gmt select input.dat -s2,3
1   1       1       0
2   2       NaN     0
3   3       3       NaN
```

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 *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。

GMT 支持的多种图片格式中, PS 和 EPS 格式均不支持透明效果。若想要看到透明效果, 需要指定生成 PDF、PNG 或 JPG 格式。

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、gpsgridder、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
- -:o 表明该选项仅交换输出数据的前两列

需要注意，该选项仅用于交换输入/输出的前两列数据，-R 选项指定绘图区域时始终是经度在前，网格文件中经度始终是第一维度。

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 math -T0/100/1  T SQRT = sqrt.txt
gmt math -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 begin GMT_linear pdf,png
gmt plot -R0/100/0/10 -JX3i/1.5i -Bag -BWSne+gsnow -Wthick,blue,- sqrt.txt
gmt plot -St0.1i -N -Gred -Wfaint sqrt10.txt
gmt end
```

图 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 begin GMT_linear_d pdf,png
gmt set MAP_GRID_CROSS_SIZE_PRIMARY 0.1i MAP_FRAME_TYPE FANCY FORMAT_GEO_MAP ddd:mm:ssF
gmt coast -Rg-55/305/-90/90 -Jx0.014i -Bagf -BWSen -Dc -A1000 -Glightbrown -Wthinnest -Slightblue
gmt end
```

图 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 begin GMT_linear_cal pdf,png
gmt set FORMAT_DATE_MAP o TIME_WEEK_START Sunday FORMAT_CLOCK_MAP=-hham \
        FORMAT_TIME_PRIMARY_MAP full
gmt basemap -R2001-9-24T/2001-9-29T/T07:0/T15:0 -JX4i/-2i -Bxa1Kf1kg1d \
              -Bya1Hg1h -BWsNe+glightyellow
gmt end
```

图 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 begin GMT_log pdf,png
gmt plot -R1/100/0/10 -Jx1.5il/0.15i -Bx2g3 -Bya2f1g2 -BWSne+gbisque \
         -Wthick,blue,- -h sqrt.txt
gmt plot -Ss0.1i -N -Gred -W -h sqrt10.txt
gmt end
```

图 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 begin GMT_pow pdf,png
gmt plot -R0/100/0/10 -Jx0.3ip0.5/0.15i -Bxa1p -Bya2f1 -BWSne+givory \
         -Wthick sqrt.txt
gmt plot -Sc0.075i -Ggreen -W sqrt10.txt
gmt end
```

图 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 begin Jp png,pdf
gmt set FORMAT_GEO_MAP +D FONT_TITLE 14p,1,red
gmt basemap -JP5c  -R0/360/0/1 -Bxa45f -B+t"-JP5c -R0/360/0/1" -Y10c
gmt basemap -JPa5c -R0/360/0/1 -Bxa45f -B+t"-JPa5c -R0/360/0/1" -X8c
gmt basemap -JPa5c -R0/90/0/1 -Bxa45f -Ba0.2 -BWNe+t"-JPa5c -R0/90/0/1" -X8c
gmt basemap -JPa5c/45 -R0/90/0/1 -Bxa45f -Ba0.2 -BWNse+t"-JPa5c/45 -R0/90/0/1" -X-16c -Y-6.5c
gmt basemap -JPa5c/45 -R0/90/3480/6371 -Bxa45f -Ba -BWNse+t"-JPa5c/45 -R0/90/3480/6371" -X8c -Y1c
```

（下页继续）

(续上页)

```
gmt basemap -JPa5c/45z -R0/90/3480/6371 -Bxa45f -Ba -BWNse+t"-JPa5c/45z -R0/90/3480/6371" -X8c
gmt end
```

图 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（0ºE/40ºS）和右上角（60ºE/10ºS）的坐标来指定区域范围。

```
gmt begin GMT_lambert_az_rect pdf,png
gmt set FORMAT_GEO_MAP ddd:mm:ssF MAP_GRID_CROSS_SIZE_PRIMARY 0
gmt coast -R0/-40/60/-10r -JA30/-30/4.5i -Bag -Dl -A500 -Gp300/10 -Wthinnest
gmt end
```

图 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 coast -Rg -JA280/30/3.5i -Bg -Dc -A1000 -Gnavy -png GMT_lambert_az_hemi
```

图 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 网使用的就是 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 125ºE/20ºN，25 度和 45 度纬线是两条标准纬线。

```
gmt begin GMT_albers pdf,png
gmt set MAP_GRID_CROSS_SIZE_PRIMARY 0
gmt coast -R110/140/20/35 -JB125/20/25/45/5i -Bag -Dl -Ggreen -Wthinnest -A250
gmt end
```

图 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 coast -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 --FONT_LABEL=12p -png GMT_cassini
```

图 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	66.159467º
Kamenetskiy' s First	55º
Gall' s stereographic	45º
Bolshoi Sovietskii Atlas Mira or Kamenetskiy' s Second	30º
Braun' s cylindrical	0º

```
gmt begin GMT_gall_stereo pdf,png
gmt set FORMAT_GEO_MAP dddA
gmt coast -R-180/180/-60/80 -JCyl_stere/0/45/4.5i -Bxa60f30g30 -Bya30g30 -Dc -A5000 \
            -Wblack -Gseashell4 -Santiquewhite1
gmt end
```

图 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 begin GMT_equidistant_conic pdf,png
gmt set FORMAT_GEO_MAP ddd:mm:ssF MAP_GRID_CROSS_SIZE_PRIMARY 0.05i
gmt coast -R-88/-70/18/24 -JD-79/21/19/23/4.5i -Bag -Di -N1/thick,red -Glightgreen -Wthinnest
gmt end
```

图 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 100ºW/40ºN，离投影中心 180 度的点在图中的最外边界处。

```
gmt coast -Rg -JE-100/40/4.5i -Bg -Dc -A10000 -Glightgray -Wthinnest -png GMT_az_equidistant
```

图 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 coast -Rg -JF-120/35/60/4.5i -B30g15 -Dc -A10000 -Gtan -Scyan -Wthinnest -png GMT_gnomonic
```

图 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得畸变。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 coast -Rg -JG-75/41/4.5i -Bg -Dc -A5000 -Gpink -Sthistle -png GMT_orthographic
```

图 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 180º 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 60º 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 coast -Rg -JG4/52/230/90/60/180/60/60/5i -Bx2g2 -By1g1 -Ia -Di -Glightbrown \
            -Wthinnest -Slightblue --MAP_ANNOT_MIN_SPACING=0.25i -png GMT_perspective
```

图 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，赤道和中心经线是执行，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 coast -Rg -JH4.5i -Bg -Dc -A10000 -Gblack -Scornsilk -png GMT_hammer
```

图 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 coast -Rd -JI4.5i -Bxg30 -Byg15 -Dc -A10000 -Ggray -png GMT_sinusoidal
```

图 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 160ºW、20ºW 和 60ºE 处。为了生成间断地图，必须调用 *coast* 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 begin GMT_sinus_int pdf,png
gmt coast -R200/340/-90/90 -Ji0.014i -Bxg30 -Byg15 -A10000 -Dc -Gblack
gmt coast -R-20/60/-90/90 -Ji0.014i -Bxg30 -Byg15 -Dc -A10000 -Gblack -X1.96i
gmt coast -R60/200/-90/90 -Ji0.014i -Bxg30 -Byg15 -Dc -A10000 -Gblack -X1.12i
gmt end
```

图 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 coast -R-90/270/-80/90 -Jj1:400000000 -Bx45g45 -By30g30 -Dc -A10000 \
            -Gkhaki -Wthinnest -Sazure -png GMT_miller
```

图 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-Greiffendorff 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 40º30’ (IV) 和 49º16’ (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>
```

<lon> 为中心经线，默认值为地图区域的中心。

Eckert IV 示例：

```
gmt coast -Rg -JKf4.5i -Bg -Dc -A10000 -Wthinnest -Givory -Sbisque3 -png GMT_eckert4
```

图 21: Eckert IV 投影绘制全球图

Eckert VI 示例:

```
gmt coast -Rg -JKs4.5i -Bg -Dc -A10000 -Wthinnest -Givory -Sbisque3 -png GMT_eckert4
```

图 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 33ºN 和 45ºN。

```
gmt begin GMT_lambert_conic pdf,png
gmt set MAP_FRAME_TYPE FANCY FORMAT_GEO_MAP ddd:mm:ssF MAP_GRID_CROSS_SIZE_PRIMARY 0.05i
gmt coast -R-130/-70/24/52 -Jl-100/35/33/45/1:50000000 -Bag -Dl -N1/thick,red \
            -N2/thinner -A500 -Gtan -Wthinnest,white -Sblue
gmt end
```

图 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 begin GMT_mercator pdf,png
gmt set MAP_FRAME_TYPE fancy
gmt coast -R0/360/-70/70 -Jm1.2e-2i -Bxa60f15 -Bya30f15 -Dc -A5000 -Gred
gmt end
```

图 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 coast -Rd -JN4.5i -Bg -Dc -A10000 -Ggoldenrod -Ssnow2 -png GMT_robinson
```

图 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:

```
-JO[a|A]<lon>/<lat>/<azi>/<width>
-Jo[a|A]<lon>/<lat>/<azi>/<scale>

-JO[b|B]<lon>/<lat>/<lon2>/<lat2>/<width>
-Jo[b|B]<lon>/<lat>/<lon2>/<lat2>/<scale>

-JO[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 coast -R270/20/305/25r -JOc280/25.5/22/69/4.8i -Bag -Di -A250 -Gburlywood \
            -Wthinnest -TdjTR+w0.4i+f2+l+o0.15i -Sazure --FONT_TITLE=8p \
            --MAP_TITLE_OFFSET=0.05i -png GMT_obl_merc
```

图 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。

```
Source Code
```

图 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 coast -R-180/-20/0/90 -JPoly/4i -Bx30g10 -By10g10 -Dc -A1000 -Glightgray \
            -Wthinnest -png GMT_polyconic
```

图 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，默认为地图区域的中心
- `<lat>` 是标准纬线，默认为赤道，若指定了标准纬线，则必须指定中心经线

```
gmt coast -Rg -JQ4.5i -B60f30g30 -Dc -A5000 -Gtan4 -Slightcyan -png GMT_equi_cyl
```

图 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	61.7º
Ronald Miller Equirectangular	50.5º
Ronald Miller, minimum continental distortion	43.5º
Grafarend and Niermann	42º
Ronald Miller, minimum overall distortion	37.5º
Plate Carrée, Simple Cylindrical, Plain/Plane	0º

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 coast -Rd -JR4.5i -Bg -Dc -A10000 -Gburlywood4 -Swheat1 -png GMT_winkel
```

图 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 coast -R-30/30/60/72 -Js0/90/4.5i/60 -B10g -Dl -A250 -Groyalblue \
            -Sseashell -png GMT_stereographic_polar
```

图 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 begin GMT_stereographic_rect pdf,png
gmt set MAP_ANNOT_OBLIQUE 30
gmt coast -R-25/59/70/72r -JS10/90/11c -B20g -Dl -A250 -Gdarkbrown -Wthinnest -Slightgray
gmt end
```

图 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 begin GMT_stereographic_general pdf,png
gmt set MAP_ANNOT_OBLIQUE 0
```

(下页继续)

(续上页)

```
gmt coast -R100/-42/160/-8r -JS130/-30/4i -Bag -Dl -A500 -Ggreen -Slightblue -Wthinnest
gmt end
```

图 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 coast -R20/30/50/45r -Jt35/0.18i -Bag -Dl -A250 -Glightbrown -Wthinnest \
            -Sseashell -png GMT_transverse_merc
```

图 34: 矩形横向 Mercator 地图

```
gmt coast -R0/360/-80/80 -JT330/-45/3.5i -Ba30g -BWSne -Dc -A2000 \
            -Slightblue -G0 -png GMT_TM
```

图 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 coast -Rg -JV4i -Bxg30 -Byg15 -Dc -Glightgray -A10000 -Wthinnest -png GMT_grinten
```

图 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 coast -Rd -JW4.5i -Bg -Dc -A10000 -Gtomato1 -Sskyblue -png GMT_mollweide
```

图 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:

Balthasart	50º
Gall	45º
Hobo-Dyer	37º30' (= 37.5º)
Trystan Edwards	37º24' (= 37.4º)
Caster	37º04' (= 37.0666º)
Behrman	30º
Lambert	0º

```
gmt coast -R-145/215/-90/90 -JY35/30/4.5i -B45g45 -Dc -A10000 -Sdodgerblue \
            -Wthinnest -png GMT_general_cyl
```

图 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　配置参数

10.1 配置参数简介

除了丰富的命令行选项之外，GMT 提供了 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）等。

10.1.1 查看配置参数的值

每个配置参数都有一个系统默认值。使用:

```
gmt defaults -D
```

即可查看所有 GMT 配置参数及其默认值。

使用:

```
gmt get FORMAT_GEO_MAP
```

可以查看单个配置参数 `FORMAT_GEO_MAP` 的当前值。

10.1.2 修改配置参数的值

GMT 提供了多种方法来控制或修改配置参数的值。

设置全局参数 用*gmtset* 模块可以为 GMT 设置全局参数，此类参数会影响到接下来所有 GMT 命令的执行，直到绘图结束或者被*gmtset* 再次修改为其他值为止。例如:

```
gmt begin map png
# 设置全局参数 FONT_ANNOT_PRIMARY 的值为 12p,Times-Bold,red
gmt set FONT_ANNOT_PRIMARY 12p,Times-Bold,red
gmt basemap ...
gmt end
```

设置临时参数 在单个命令上加上 `--KEY=value` 可以临时设置配置参数的值。此类参数仅对当前命令有效，而不影响接下来其他命令的执行效果。例如:

```
gmt begin map png
# 使用默认参数绘制底图
gmt basemap ...
# 该底图的 FONT_ANNOT_PRIMARY 为 12p,Times-Bold,red
gmt basemap ... --FONT_ANNOT_PRIMARY=12p,Times-Bold,red
# 使用默认参数绘制底图
```

（下页继续）

(续上页)

```
gmt basemap ...
gmt end
```

使用配置文件设置全局参数 可以将需要配置的一系列参数值写到 GMT 配置文件 `gmt.conf` 中。当 GMT 在执行时，会在当前目录->~/.gmt/以及家目录下寻找 GMT 配置文件 `gmt.conf`。若找到该配置文件，则会读取该配置文件中参数的值作为全局参数。

此种方式通常用于制作某个特定风格的图件（比如黑底白线）或者某个符合某个期刊特定要求的图件。可以使用:

```
gmt defaults -D > gmt.conf
```

生成一个包含所有参数的配置文件，然后手动修改。

10.2 配置参数示例

下面的图展示了常用的用于控制绘图效果的配置参数。

Source Code

Source Code

Source Code

10.3 FONT 参数

这一节列出所有字体相关的参数，参数的默认值在中括号内列出。

FONT 同时设置所有 FONT 类参数（FONT_LOGO 除外）的字体

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 +，则其它字体、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）标注的字体 [14p,Helvetica,black]

FONT_LABEL 轴标签的字体 [16p,Helvetica,black]

FONT_TITLE 图上方标题的字体 [24p,Helvetica,black]

FONT_HEADING 子图模式下总标题的字体 [32p,Helvetica,black]

FONT_TAG 子图模式下每个子图编号（如 a)、ii）等）的字体 [20p,Helvetica,black]

FONT_LOGO GMT 时间戳中字符串的字体 [8p,Helvetica,black]

该参数中仅字体 ID 有效，字号及颜色均无效。

10.4 MAP 参数

10.4.1 边框相关参数

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_FRAME_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_FRAME_AXES 要绘制/标注的轴 [`WSENZ`]

默认值为 `WSENZ`，即 2D 底图下绘制并标注四条边，可以通过 `-B` 选项控制实际绘制的边。详情见*-B 选项* 选项。

10.4.2 标注相关参数

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_DEGREE_SYMBOL 在地图上绘制 “度” 时所使用的符号 [degree]

可以取 `ring|degree|colon|none`。下图给出了取不同值时的绘图效果：

Source Code

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。下图给出了取不同值时的绘图效果：

Source Code

MAP_ANNOT_OBLIQUE 控制倾斜投影下标注和刻度线的显示

可以将如下任意几个数字求和的结果作为该参数的值：

- `1`：当网格线穿过底图边界时添加标注，否则仅在上下边界处标注经度，在左右边界处标注纬度
- `2`：经度标注水平绘制
- `4`：纬度标注水平绘制
- `8`：倾斜的刻度线会扩展使得其长度等于指定刻度线长度
- `16`：忽略网格线与边框的夹角，刻度线总是垂直于底图边框
- `32`：纬度标注平行于底图边框

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 到 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]

10.4.3 标签相关参数

MAP_LABEL_OFFSET 轴标注底部与轴标签顶部间的距离 [8p]

10.4.4 刻度相关参数

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%

10.4.5 网格线相关参数

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 一级网格十字线的大小，0 表示绘制连续的网格线 [0p]

MAP_GRID_CROSS_SIZE_SECONDARY 二级网格十字线的大小，0 表示绘制连续的网格线 [0p]

MAP_POLAR_CAP 控制网格线在两极附近的显示 [85/90]

若取值为 `none`，则表示不对极点附近的网格线做特殊处理。否则可以按 `<lat>/<dlon>` 格式取值，表示在 `-<lat>` 到 `+<lat>` 纬度范围内正常绘制网格线；在大于 `+<lat>` 和小于 `-<lat>` 纬度区域内，则按照 `<dlon>` 指定的经线间隔绘制网格线。GMT 会在 `<lat>` 纬度处绘制一个圈圈以分隔这两个纬度区间。

下图展示了该参数取不同值时的绘图效果：

Source Code

10.4.6 标题相关参数

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]

10.4.7 其它参数

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_ORIGIN_X 新绘图在纸张上的原点的 X 坐标（仅适用于 GMT 经典模式）[72p]

MAP_ORIGIN_Y 设置新绘图在纸张上的原点的 Y 坐标（仅适用于 GMT 经典模式）[72p]

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_SCALE_HEIGHT 地图比例尺的高度 [5p]

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_VECTOR_SHAPE 矢量箭头的形状 [0]

可以取-2 到 2 之间的任意实数。下面展示了取-2、-1、0、1 和 2 时的矢量箭头的形状：

Source Code

10.5 COLOR 参数

这一节列出所有与颜色相关的配置参数，参数的默认值在中括号内列出。

10.5.1 CPT 相关参数

COLOR_BACKGROUND 数据 Z 值小于 CPT 文件中最小值时使用的背景色 [black]

COLOR_FOREGROUND 数据 Z 值大于 CPT 文件中最大值时使用的前景色 [white]

COLOR_NAN 数值 Z 值为 NaN 时使用的颜色 [127.5]

COLOR_MODEL 对 CPT 文件做插值生成新 CPT 时所使用的色彩模型 [none]

可以取如下值：

- none：使用 CPT 文件中指定的 COLOR_MODEL
- rgb：在 RGB 色彩空间中插值
- hsv：在 HSV 色彩空间中插值
- cmyk：假定颜色是 CMYK 色彩空间，但在 RGB 空间内插值

10.5.2 光照相关参数

某些绘图模块（如*grdimage*、*colorbar*）可以利用强度文件模拟光照效果。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，以避免模拟的光照过亮或过暗。

COLOR_HSV_MIN_S 负强度最小值对应的 S 值，取值范围 0 到 1 [1.0]

COLOR_HSV_MAX_S 正强度最大值对应的 S 值，取值范围 0 到 1 [0.1]

COLOR_HSV_MIN_V 负强度最小值对应的 V 值，取值范围 0 到 1 [0.3]

COLOR_HSV_MAX_V 正强度最大值对应的 V 值，取值范围 0 到 1 [1.0]

10.6 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_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，首先在当前目录下寻找该文件，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_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

DIR_CACHE GMT 用于存储从 GMT 服务器下载得到的临时文件的目录

10.7 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。

10.7.1 日期的输入/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 ddmmyyy、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。

10.7.2 时间的输入/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]

10.7.3 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。

10.7.4 浮点数的输出/绘图格式

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 显示则会出错
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。

10.7.5 其他数据的绘图格式

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 GMT 安装目录下 `share/localization` 目录中的语言定义文件。

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。

10.8 IO 参数

10.8.1 表数据相关参数

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`，则等效于使用*-h 选项*

IO_HEADER_MARKER 输入 ASCII 表文件中文件头记录的标识符 [`#`]

若希望输入和输出数据中使用不同的文件头标识符，则可以使用逗号分隔输入和输出数据的文件头标识符，比如 `#,:`。

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*ASCII 表*。

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` [maybe]

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*ASCII 表* 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

若想要将字符 `B` 或 `N` 作为段数据标识符，而不是使用上面提到的特殊含义，则必须

使用 \B 或 \N。

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。

10.8.2 网格文件相关参数

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_GRIDFILE_SHORTHAND 是否支持自动识别网格文件后缀的功能 [false]

GMT 中也可以将网格文件的后缀与网格文件格式关联起来这样 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。

IO_GRIDFILE_FORMAT GMT 默认使用的网格文件格式 [nf]

见网格文件格式 一节。

10.8.3 其他 IO 参数

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* 功能相同。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

10.9 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* 一节的介绍。

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 (此处不一一列举, 详见官方文档)。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. `Rudoe` given for legacy purpose
3. `Andoyer` 精度为 10 米量级，比 `Vincenty` 快 5 倍

10.10 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+`]

GMT 使用的字符集编码方式。可选值包括：

- `Standard`
- `Standard+`
- `ISOLatin1`
- `ISOLatin1+`
- `ISO-8859-x`（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_COLOR_MODEL 生成 PS 代码时颜色所使用的色彩模型 [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_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

下图展示了该参数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，但因参数设置不同而导致线段看上去长度不同。

Source Code

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]

注解: 仅在 GMT 经典模式下有效。GMT 现代模式下直接使用无限大的纸张。

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 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
```

（下页继续）

(续上页)

```
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`]

注解: 仅 GMT 经典模式下有效，现代模式下纸张始终是 protrait 模式。

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_CONVERT 现在模式下在执行 `gmt end` 命令时 GMT 会自动调用*psconvert*生成用户指定格式的图片。该选项用于控制调用 `psconvert` 是的默认参数，多个参数之间以逗号分隔 [`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` 。

10.11 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 [Monday]

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_REPORT 控制 GMT 运行进度报告中是否显示时间戳 [`none`]

可以取三个值：

- `none` 不显示时间戳
- `clock` 显示绝对时间
- `elapsed` 显示自会话开始所经历的时间

TIME_IS_INTERVAL 控制输入的日期时间数据截断和微调 [`off`]

其可以取如下三类值：

- `off`：即不对输入数据做任何截断和调整

- +<n><u>: <n> 为某个整数，<u> 为某个时间单位。其表示将输入的日期时间数据截断为 <n><u> 的整数倍，并将其放在紧接着的时间间隔的中间
- -<n><u>: 同样，但将该输入数据放在前一个时间间隔的中间

时间间隔单位 <u> 可以取如下值:

- y 年
- o 月
- u 周
- h 小时
- m 分钟
- s 秒

下面的示例在时间坐标系下绘制了三个红点，每个红点的输入数据都是:

```
1997-03-05 1
```

图中展示了 TIME_IS_INTERVAL 取不同值时的效果:

- off: 1997-03-05 解释为 1997-03-05T00:00:00.00.0
- +1o: 1997-03-05 解释为 1997-03-15T12:00:00.0
- -1o: 1997-03-05 解释为 1997-02-15T12:00:00.0

Source Code

TIME_INTERVAL_FRACTION 确定时间轴开头和结尾的部分时间间隔是否需要标注 [0.5]

对于时间轴而言，若开头/结尾部分的时间间隔大于指定的时间间隔的某个比例，则绘制开头/结尾部分的标注并将标注置于时间间隔的中间。

10.12 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。

10.12.1 数据下载相关参数

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_SERVER GMT 数据服务器地址，默认使用 SOEST 官方镜像 [http://www.soest.hawaii.edu/gmt/data]

GMT_DATA_SERVER_LIMIT 从 GMT 服务器上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。可以给定文件大小上限的字节数，也可以加上 `k`、`m` 或 `g` 表示 KB、MB 或 GB。

10.12.2 算法选择相关参数

GMT_TRIANGULATE 设置 triangulate 模块中算法代码的来源 [Watson]

triangulate 模块的核心源码有两个版本，`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_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

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_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`

10.12.3 其他参数

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_CUSTOM_LIBS 要链接的自定义 GMT 库文件，默认值为空

GMT 支持自定义模块。用户可以写一个 GMT 模块，并将其编译成动态函数库。通过设置该参数告知 GMT 该函数库的位置，即可通过 `gmt xxx` 的语法调用自定义模块，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 /，则表示在 GMT 的 bin 目录下的 gmt_plugins 子目录下寻找库文件。

GMT_LANGUAGE 设置 GMT 绘图时使用的语言 [us]

不同的语言中，月份、星期几、东西南北的表达方法是不同的。该参数用于设置 GMT 绘图时所使用的语言。GMT 支持多种语言，各语言的定义文件位于 GMT 安装目录中 share/localization 目录下的文件。

此处仅列举几个常见语言如下：

- cn1 简体中文
- cn2 繁体中文
- uk 英式英语
- us 美式英语
- jp 日语
- kr 韩语
- …

实际使用时，除了需要修改该参数外，可能还需要修改相应的字符编码和字体。

若设置语言为 cn1 即简体中文并正确设置中文字体，则 GMT 在绘制时可以显式“一月”、“星期一”、“周一”等中文。相关示例见中文图。

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_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_GRAPHICS_FORMAT 现代模式下默认的图片文件格式 [pdf]

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

建立 GMT 数据库的方法很简单，只要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。多个目录之间用逗号 ，分隔。例如:

```
export GMT_DATADIR=~/GMTDB/data1,~/GMTDB/data2
```

Linux 和 macOS 用户，可以进一步简化为:

```
export GMT_DATADIR=~/GMTDB/
```

GMT_DATADIR 中的目录以 / 结尾，GMT 则会在 ~/GMTDB 及其子目录下递归地寻找文件。需要注意，Windows 似乎不支持这一功能。

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*coast* 模块可以直接绘制 GSHHG 中的数据，也可以使用*coast* 的 **-M** 选项将数据导出为纯文本文件供其它程序使用。这一节将利用*coast* 模块介绍 GSHHG 数据，关于*coast* 模块的详细用法，见*coast* 模块的说明文档。

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
```

coast 模块的 **-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绘图速度慢且文件太大；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

coast 模块中有如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.

coast 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

coast 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 coast -R-130/-70/24/52 -JL-100/35/33/45/15c -Ba -Dh -A1000 -W1/0.5p -png map
```

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 coast -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 -
↪png map
```

绘制 1-3 级海岸线，并为陆地、还有、湖泊填充不同的颜色：

```
gmt coast -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 -png map
```

绘制海岸线、国界和美国州界：

```
gmt coast -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 -png map
```

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*coast* 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*coast* 的 **-M** 选项将边界数据导出为纯文本文件供其它程序使用。

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

每个国家都有一个国家代码，国家代码可以直接从 ISO Country Codes 中查找。也可以从 DCW 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
```

也可以直接在中查看。

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

GMT 中至少有两处会使用 DCW 数据：

1. *-R 选项* 中可以直接使用区域代码以间接指定绘图范围
2. *coast* 模块 **-E***code1*,*code2*,…选项调用 DCW 数据绘制或导出国界/省界

洲代码、国家代码和省代码都由两个字符构成，为了避免可能的冲突，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 coast -R15/180/-20/70 -JM12c -Baf -E=AS+p0.25p,red -png dataset_dcw_01
```

绘制国界

绘制中国大陆边界（不含台湾、香港、澳门），其中 **-R** 选项可省略：

```
gmt coast -JM15c -Baf -ECN+p0.25p,red -R70/140/10/60 -png dataset_dcw_02
```

绘制省界

绘制内蒙古，并设置边界颜色和填充颜色，其中 **-R** 选项可省略:

```
gmt coast -JM12c -Baf -ECN.15+p1p,blue+gred -R90/130/35/55 -png dataset_dcw_03
```

导出省界数据

导出内蒙古的边界数据:

```
gmt coast -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 1 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
earth_relief_01d	1 弧度	112 KB
earth_relief_30m	30 弧分	377 KB
earth_relief_20m	20 弧分	783 KB
earth_relief_15m	15 弧分	1.4 MB
earth_relief_10m	10 弧分	2.9 MB
earth_relief_06m	6 弧分	7.5 MB
earth_relief_05m	5 弧分	11 MB
earth_relief_04m	4 弧分	16 MB
earth_relief_03m	3 弧分	28 MB
earth_relief_02m	2 弧分	58 MB
earth_relief_01m	1 弧分	214 MB
earth_relief_30s	30 弧秒	778 MB
earth_relief_15s	15 弧秒	2.6 GB
earth_relief_03s	3 弧秒	6.8 GB
earth_relief_01s	1 弧秒	41 GB
srtm_relief_03s	3 弧秒	6.8 GB
srtm_relief_01s	1 弧秒	41 GB

这些地形起伏数据保存在 GMT 的服务器上。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/server` 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。

小技巧: 由于第一次使用地形数据时，GMT 需要从服务器下载数据，通常会需要很长时间。建议用户可以在闲置时提前将精度为 15 弧秒到 1 户分的地形数据下载到自己的计算机上。

可以复制并执行如下命令以下载不同精度的地形数据:

```
gmt which -Gu @earth_relief_01d
gmt which -Gu @earth_relief_30m
gmt which -Gu @earth_relief_20m
gmt which -Gu @earth_relief_15m
gmt which -Gu @earth_relief_10m
gmt which -Gu @earth_relief_06m
gmt which -Gu @earth_relief_05m
gmt which -Gu @earth_relief_04m
gmt which -Gu @earth_relief_03m
gmt which -Gu @earth_relief_02m
gmt which -Gu @earth_relief_01m
gmt which -Gu @earth_relief_30s
```

（下页继续）

(续上页)

```
gmt which -Gu @earth_relief_15s
```

不建议提前下载 1 弧秒和 3 弧秒的地形数据，主要原因在于，这两套数据占据硬盘空间太大。基于同样的理由，GMT 服务器上这两套数据不是以单个文件的形式存放，而是被分成了多个小块，当用户需要绘制某个区域的高精度地形时，GMT 会自动下载该区域的所有区块的地形数据，然后合并成单个网格数据供用户使用。

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形起伏数据。

使用示例:

```
gmt grdinfo @earth_relief_60m

gmt grdimage -JH15c @earth_relief_15m -pdf map
```

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_SERVER_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 data** 命令清空整个数据缓存目录

11.4.4 技术细节

- 15 弧秒的数据来源于 SRTM15+V2 [http://dx.doi.org/10.1029/2019EA000658]
- 30 弧秒及更低分辨率的全球地形数据均是 SRTM15+V2 的衍生产品。GMT 利用笛卡尔高斯滤波对其进行重采样以防止混叠现象。可以使用*grdinfo* 查看生成网格文件所使用的滤波命令。
- 3 弧秒和 1 弧秒的数据来自于 NASA 提供的 SRTM 数据。数据被划为为 1 度 x1 度的区块。在使用时，GMT 会根据 **-R** 选项指定的区域范围只下载区域内的地形数据。
- 原始的 SRTM3 和 SRTM1 数据只在北纬 60 度到南纬 60 度的陆地上有数据。当使用 **@earth_relief_01s** 或 **@earth_relief_03s** 时，GMT 会自动对 **@earth_relief_15s** 数据对增采样以填充缺失的海洋部分。
- 如果想使用最原始的只包含陆地的 SRTM 地形数据，则可以使用 **@srtm_relief_03s** 或 **srtm_relief_01s**。
- 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。对于 3 弧秒和 1 弧秒的数据，是以 JPEG2000 图片格式保存在 GMT 服务器上的，一旦数据下载到本地目录中，则

会被转换为压缩的 netCDF4 格式，这一步通过 GDAL 来实现，且要求 GDAL 支持 openjpeg。

11.4.5 数据来源及引用

1. SRTM15+V2 [http://dx.doi.org/10.1029/2019EA000658]
2. SRTMGL3 数据：https://lpdaac.usgs.gov/dataset_discovery/measures/measures_products_table/srtmgl3_v003
3. SRTMGL1 数据：https://lpdaac.usgs.gov/dataset_discovery/measures/measures_products_table/srtmgl1_v003

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 2.4.2，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不可省略）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
gmt begin gadm_level0 pdf,png
gmt plot -JM15c -R72/136/15/54 gadm36_CHN_0.gmt
gmt plot gadm36_HKG_0.gmt
gmt plot gadm36_MAC_0.gmt
gmt plot gadm36_TWN_0.gmt
gmt end
```

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
gmt begin gadm_level1 pdf,png
gmt plot -JM15c -R72/136/15/54 gadm36_CHN_1.gmt
gmt plot gadm36_HKG_0.gmt
gmt plot gadm36_MAC_0.gmt
gmt plot gadm36_TWN_0.gmt
gmt end
```

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。

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 plot -JM10c -R114/120/29/35 gadm36_CHN_Anhui_2.gmt -png gadm_level2
```

绘图效果如下：

11.5.5 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。

11.5.6 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　中文支持

12.1 Linux 下的 GMT 中文支持

本文介绍如何让 GMT 在 Linux 下支持中文。

注解: 据用户反映，本文所介绍的内容可能存在严重问题！

Linux 用户请阅读自定义字体 一文。

12.1.1 ghostscript 的中文支持

Linux 的中文字体较少，这里使用 Windows 下提供的四个基本字体：宋体、仿宋、黑体和楷体。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。

可以使用 cjk-gs-support 项目提供的脚本 cjk-gs-integrate.pl 来实现 ghostscript 的中文支持。

1. 从 Windows 下获取四种基本字体的字体文件（文件名类似于 `simsun.ttc`）并复制到 `/usr/share/fonts/winfonts/` 目录下
2. 下载脚本 cjk-gs-integrate.pl
3. `cjk-gs-integrate.pl` 脚本的执行依赖于命令 `kpsewhich`，该命令由 TeXLive 提供。执行 `kpsewhich --version` 检查 `kpsewhich` 这个命令是否存在。若不存在，则需要单独安装。

   对于 Ubuntu/Debian 用户，执行:

   ```
   sudo apt-get install texlive-binaries
   ```

   对于 CentOS/RHEL/Fedora 用户，执行:

   ```
   sudo yum install texlive-kpathsea-bin
   ```

4. 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。

12.1.2 GMT 的中文支持

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
$ gedit ~/.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
STSong-Light--UniGB-UTF8-V  0.700    1
STFangsong-Light--UniGB-UTF8-V  0.700    1
STHeiti-Regular--UniGB-UTF8-V   0.700   1
STKaiti-Regular--UniGB-UTF8-V   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的横排和竖排两种方式。

用 gmt text -L 命令查看 GMT 当前的字体配置:

```
$ gmt text -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
43 STSong-Light--UniGB-UTF8-V
44 STFangsong-Light--UniGB-UTF8-V
45 STHeiti-Regular--UniGB-UTF8-V
46 STKaiti-Regular--UniGB-UTF8-V
```

其中 39-46 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。

12.1.3 GMT 中文测试

注解: 请自行确认你的中文字体编号。如果编号不是 39 到 46，请自行修改以下测试脚本。

```
#!/bin/bash
gmt begin GMT_Chinese pdf,png
gmt set FONT_TITLE 25p,41,black
gmt set FONT_LABEL 15p,39,black
```

(下页继续)

(续上页)

```
gmt text -R0/8/0/4 -JX12c/4c -Bxaf+l"X 轴" -Byaf+l"Y 轴" -BWSen+t" 中文标题" -F+f << EOF
2 3.5 25p,39,black 中文宋体
2 2.5 25p,40,blue  中文仿宋
2 1.5 25p,41,red   中文黑体
2 0.5 25p,42,green 中文楷体
4 3.5 25p,43,black 中文宋体
5 3.5 25p,44,blue  中文仿宋
6 3.5 25p,45,red   中文黑体
7 3.5 25p,46,green 中文楷体
EOF
gmt end
```

成图效果如下:

12.2 macOS 下的 GMT 中文支持

本文介绍如何让 GMT 在 macOS 下支持中文。

12.2.1 ghostscript 的中文支持

首先需要使 ghostscript 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. `cjk-gs-integrate.pl` 脚本的执行依赖于命令 `kpsewhich`，该命令由 TeXLive 提供。执行 `kpsewhich --version` 检查 `kpsewhich` 这个命令是否存在。若不存在，则需要单独安装。使用 homebrew 安装 basictex 或 mactex-no-gui:

```
# 以下二选一即可，第一个更小，第二个更完整
brew cask install basictex
brew cask install mactex-no-gui
```

3. 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。

12.2.2 GMT 的中文支持

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
STSong-Light--UniGB-UTF8-V  0.700    1
STFangsong-Light--UniGB-UTF8-V  0.700    1
STHeiti-Regular--UniGB-UTF8-V   0.700   1
STKaiti-Regular--UniGB-UTF8-V   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的横排和竖排两种方式。

用 `gmt text -L` 命令查看 GMT 当前的字体配置:

```
$ gmt text -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
43 STSong-Light--UniGB-UTF8-V
44 STFangsong-Light--UniGB-UTF8-V
45 STHeiti-Regular--UniGB-UTF8-V
46 STKaiti-Regular--UniGB-UTF8-V
```

其中 39-46 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。

12.2.3 GMT 中文测试

注解: 请自行确认你的中文字体编号。如果编号不是 39 到 46，请自行修改以下测试脚本。

```
#!/bin/bash
gmt begin GMT_Chinese pdf,png
gmt set FONT_TITLE 25p,41,black
gmt set FONT_LABEL 15p,39,black

gmt text -R0/8/0/4 -JX12c/4c -Bxaf+l"X 轴" -Byaf+l"Y 轴" -BWSen+t" 中文标题" -F+f << EOF
2 3.5 25p,39,black 中文宋体
2 2.5 25p,40,blue  中文仿宋
2 1.5 25p,41,red   中文黑体
2 0.5 25p,42,green 中文楷体
4 3.5 25p,43,black 中文宋体
5 3.5 25p,44,blue  中文仿宋
6 3.5 25p,45,red   中文黑体
7 3.5 25p,46,green 中文楷体
EOF
gmt end
```

成图效果如下：

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。

12.3 Windows 下的 GMT 中文支持

12.3.1 ghostscript 的中文支持

通常，在 `C:\Program Files\gs\gs9.26\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\gs9.26\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。

12.3.2 配置 Ghostscript 环境变量

为了能够在将 PS 文件转换为其他图片格式时也支持中文，需要设置环境变量 *GS_FONTPATH*。具体步骤如下：

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，新建变量 *GS_FONTPATH* 并设置其值为 *C:Windowsfonts*

12.3.3 gsview 的中文支持

注解: 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。

12.3.4 GMT 的中文支持

新建 GMT 自定义字体配置文件 `C:\Users\用户名\.gmt\PSL_custom_fonts.txt`（若不存在 `C:\Users\用户名\.gmt` 目录则需新建该目录。Windows 的文件管理器无法新建以 `.` 开头的文件夹，因而需要打开 CMD，然后执行命令 `mkdir .gmt` 以创建该文件夹）。

向 GMT 自定义字体配置文件 C:\Users\用户名\.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
STSong-Light--GB-EUC-V  0.700    1
STFangsong-Light--GB-EUC-V  0.700    1
STHeiti-Regular--GB-EUC-V   0.700   1
STKaiti-Regular--GB-EUC-V   0.700   1
```

用 gmt text -L 查看 GMT 字体:

```
$ gmt text -L
Font #  Font Name
------------------------------------
0   Helvetica
1   Helvetica-Bold
...    ......
39 STSong-Light--GB-EUC-H
40 STFangsong-Light--GB-EUC-H
41 STHeiti-Regular--GB-EUC-H
42 STKaiti-Regular--GB-EUC-H
43 STSong-Light--GB-EUC-V
44 STFangsong-Light--GB-EUC-V
45 STHeiti-Regular--GB-EUC-V
46 STKaiti-Regular--GB-EUC-V
```

可以看到，新添加的四种中文字体对应的字体编号为 39 到 46。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，V 表示竖排文字。强烈建议在执行测试脚本前确认自己的中文字体编号。

12.3.5 GMT 中文测试

注解: 请自行确认你的中文字体编号。如果编号不是 39 到 46，请自行修改以下测试脚本。

```
gmt begin map pdf,png
gmt set FONT_TITLE 25p,41,black
gmt set FONT_LABEL 15p,39,black

echo 2 3.5 25p,39,black 中文宋体  > tmp
echo 2 2.5 25p,40,blue  中文仿宋 >> tmp
echo 2 1.5 25p,41,red   中文黑体 >> tmp
echo 2 0.5 25p,42,green 中文楷体 >> tmp
echo 4 3.5 25p,43,black 中文宋体 >> tmp
echo 5 3.5 25p,44,blue  中文仿宋 >> tmp
echo 6 3.5 25p,45,red   中文黑体 >> tmp
echo 7 3.5 25p,46,green 中文楷体 >> tmp
```

(下页继续)

(续上页)

```
gmt text tmp -R0/8/0/4 -JX12c/4c -Bxaf+l"X 轴" -Byaf+l"Y 轴" -BWSen+t" 中文标题" -F+f
del tmp
gmt end
```

成图效果如下：

注解: 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。

12.4 自定义字体

GMT 默认支持 35 种 PS 标准字体。如果想要使用额外的字体（比如其他西文字体或中日韩字体），则需要用户自行配置。在安装 一章中已经简要介绍了如何利用第三方提供的配置脚本在 Linux、macOS 下修改 ghostscript 配置文件以使得 GMT 支持中文字体。这一节则更详细地介绍修改 ghostscript 配置文件和 GMT 字体配置文件的基本原理。

本文依然以四个基本的中文字体为例。

12.4.1 基本原理

GMT 本质上是生成 PS 文件，并利用 ghostscript 将其转换为其他图片格式。因而，为 GMT 自定义字体本质上分为两步：

1. 修改 ghostscript 配置文件
2. 修改 GMT 配置文件

12.4.2 ghostscript 中文配置

中文配置文件

不同系统下 ghostscript 的的中文配置文件的位置不同。此处以 CentOS 7 为例。

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
```

配置文件的内容

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 ;
/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。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。

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] >> ;
```

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。

12.4.3 GMT 中文支持

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。

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 0-38 为 GMT/gs 默认支持的字体，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。

12.4.4 GMT 中文测试

测试脚本:

```
#!/bin/bash
gmt begin GMT_Chinese png,pdf
gmt set FONT_TITLE 30p,39,black
gmt set FONT_LABEL 15p,39,black

gmt text -R0/10/0/4 -JX15c/5c -Bxafg+l"X 轴" -Byafg+l"Y 轴" \
         -BWSen+t" 中文标题" -F+f << 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 end
```

成图效果如下:

12.4.5 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

1. 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。
2. 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

1. 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
```

2. ghostscript 中文配置文件路径为: `/etc/ghostscript/cidfmap.d/90gs-cjk-resource-gb1.conf`
3. 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-arphic-uming fonts-arphic-ukai
```

4. gs 中文配置文件的默认内容为:

```
/BousungEG-Light-GB <<FileType /TrueType /Path (/usr/share/fonts/truetype/arphic/uming.ttc) /
↪SubfontId 0 /CSI [(GB1) 4] >> ;
```

（下页继续）

(续上页)

```
/GBZenKai-Medium    <</FileType /TrueType /Path (/usr/share/fonts/truetype/arphic/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/arphic/uming.ttc) /
↪SubfontId 0 /CSI [(GB1) 4] >> ;
/GBZenKai-Medium    <</FileType /TrueType /Path (/usr/share/fonts/truetype/arphic/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
/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

1. ghostscript 中文配置文件需要用如下命令安装:

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

2. 其他部分未做测试，估计跟 Ubuntu 15.05 差不多。

12.4.6 参考资料

1. GMT 软件显示汉字的技术原理与实现，赵桂儒，《测绘通报》

2. ghostscript 中文打印经验
3. GMT 中文支持
4. 维基词条: PostScript
5. Debian Wiki

第 13 章　模块手册

主程序/脚本	
gmt	GMT 主程序
gmt-config	返回 GMT 动态函数库的基本信息
现代模式会话管理	
begin	初始化一个新的 GMT 现代模式会话
end	结束现代模式会话，生成并显示图片
figure	设置当前图片的属性
subplot	管理和设置子图模式
inset	管理和设置图中图模式
docs	打开指定模块的 HTML 文档
clear	删除缓存目录、数据目录或会话目录，以及当前配置文件
绘图相关模块	
basemap	绘制底图
coast	在地图上绘制海岸线、河流、国界线
plot	在图上绘制线段、多边形和符号
text	在图上写文本
colorbar	在图上绘制色标
legend	绘制图例
histogram	统计并绘制直方图
rose	绘制极坐标下的直方图（sector 图、rose 图或 windrose 图）
events	绘制特定时刻的事件符号和标签信息
plot3d	在 3D 图上绘制线段、多边形和符号
image	将 EPS 或光栅图片放在图上
solar	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
clip	打开或关闭多边形裁剪路径
sac	在地图上绘制 SAC 格式的地震波形数据
meca	在地图上绘制震源机制解
polar	在震源球上绘制台站极性
velo	在地图上绘制速度矢量、十字线、楔形图
coupe	绘制震源机制解的剖面图
ternary	绘制三角图解

下页继续

表 1 – 续上页

主程序/脚本	
mask	将没有数据覆盖的区域裁剪或覆盖住
contour	使用直接三角化法对数据进行等值线绘制
wiggle	沿着测线绘制 z = f(x,y) 数据
segy	在图上绘制 SEGY 文件
segyz	在 3D 图上绘制 SEGYZ 文件
gmtlogo	在图上绘制 GMT 图形 logo
grdvector	根据两个网格文件绘制矢量场
grdimage	在图上绘制网格数据
grdcontour	根据网格文件绘制等值线
grdview	利用网格文件绘制 3D 视角图或表面网格图
movie	制作动画
1D 数据处理	
makecpt	生成 CPT 文件
gmtselect	根据多个准则筛选数据
project	将数据点投影到线或大圆路径上，生成测线，坐标转换
gmtconvert	表数据格式转换、列提取、列粘贴
trend1d	一维数据的多项式拟合
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
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
gmtmath	对表数据进行数学计算操作
mapproject	地图变化的正变化和逆变换
gmtspatial	线段和多边形的地理空间操作
gmtvector	2D 和 3D 下笛卡尔矢量操作
gmtregress	1D 数据的线性回归
2D 数据处理	
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
grdpaste	将两个网格沿着其共同边界拼接成一个文件
grdblend	将多个部分重叠的网格文件合并成一个网格文件
grdtrack	获取指定地理位置处的网格值
grdgradient	计算网格的方向梯度
grdclip	对网格文件的 Z 值做裁剪
grd2xyz	将网格文件转换成表数据

下页继续

表 1 – 续上页

主程序/脚本	
grdconvert	在不同的网格格式之间互相转换
grdedit	修改网格文件的头段或内容
grdinfo	从网格文件中提取基本信息
grdsample	对网格文件做重采样
grdlandmask	根据海岸线数据创建陆地-海洋的 mask 网格文件
grdvolume	计算网格数据中某个等值线所包围的表面积和体积
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
grdmath	对网格文件做数学计算操作
grdfilter	对网格文件做空间域或时间域滤波
grdfft	对网格文件在波数域或频率域做操作
grdfill	对网格文件中的无值区域进行插值
grdhisteq	对网格做直方图均衡
grd2cpt	根据网格文件的值生成 CPT 文件
blockmean	使用 L2 范式对 (x,y,z) 数据做区块平均
blockmedian	使用 L1 范式对 (x,y,z) 数据做区块平均
blockmode	使用模估计对 (x,y,z) 数据做区块平均
surface	使用可调节张量连续曲率样条插值法对数据进行网格化
splitxyz	将表数据拆分为单独的数据段
triangulate	对表数据做三角剖分和网格化
nearneighbor	使用 “Nearest neighbor” 算法对数据进行网格化
trend2d	二维数据的多项式拟合
greenspline	使用格林函数样条进行插值
sph2grd	根据球谐系数计算网格
sphdistance	计算球面上的 Voronoi 距离、节点或 natural nearest-neighbor grid
sphinterpolate	球面上张量数据的球面网格化
sphtriangulate	球面数据的 Delaunay 或 Voronoi 构建
dimfilter	在空间域对网格数做方向性滤波
参数设置	
gmtdefaults	列出所有 GMT 参数的当前值
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
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

下页继续

表 1 – 续上页

主程序/脚本	
gmtwhich	返回指定文件的完整路径
格式转换	
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
gmt2kml	将 GMT 表数据转换为 Google Earth 的 KML 文件
grdconvert	在不同的网格格式之间互相转换
psconvert	将 GMT 生成的 PS 文件转换为其他图片格式
mgd77 相关模块	
mgd77convert	Convert MGD77 data to other file formats
mgd77header	Create MGD77 headers from A77 files
mgd77info	Extract information about MGD77 files
mgd77list	Extract data from MGD77 files
mgd77magref	Evaluate the IGRF or CM4 magnetic field models
mgd77manage	Manage the content of MGD77+ files
mgd77path	Return paths to MGD77 cruises and directories
mgd77sniffer	Along-track quality control of MGD77 cruises
mgd77track	Plot track-line map of MGD77 cruises
x2sys 相关模块	
x2sys_binlist	Create bin index listing from track data files
x2sys_cross	Calculate crossovers between track data files
x2sys_datalist	Extract content of track data files
x2sys_get	Get track listing from track index database
x2sys_init	Initialize a new x2sys track database
x2sys_list	Extract subset from crossover data base
x2sys_merge	Merge an updated COEs table (smaller) into the main table (bigger)
x2sys_put	Update track index database from track bin file
x2sys_report	Report statistics from crossover data base
x2sys_solve	Determine least-squares systematic correction from crossovers
SPOTTER 相关模块	
backtracker	Generate forward and backward flowlines and hotspot tracks
gmtpmodeler	Evaluate a plate motion model at given locations
grdpmodeler	Evaluate a plate motion model on a geographic grid
grdrotater	Finite rotation reconstruction of geographic grid
originater	Associate seamounts with nearest hotspot point sources
rotconverter	Manipulate total reconstruction and stage rotations

下页继续

表 1 – 续上页

主程序/脚本	
rotsmoother	Get mean rotations and covarience from set of finate rotations
grdspotter	Create CVA image from a gravity or topography grid
hotspotter	Create CVA image from seamount locations
POTENTIAL 相关模块	
gmtflexure	Compute flexural deformation of 2-D loads, forces and bending moments
grdflexure	Compute flexural deformation of 3-D surfaces for various rheologies
gmtgravmag3d	Compute the gravity/magnetic anomaly of a 3-D body
grdgravmag3d	Computes the gravity effect of one (or two) grids
gravfft	Compute gravitational attraction of 3-D surfaces and a little more
grdredpol	Compute the Continuous Reduction To the Pole, AKA differential RTP
grdseamount	Compute synthetic seamount bathymetry
talwani2d	Compute geopotential anomalies over 2-D bodies
talwani3d	Compute geopotential anomalies over 3-D bodies
其他模块	
segy2grd	Converting SEGY data to a GMT grid
gshhg	Extract data tables from binary GSHHS or WDBII data files
img2google	Create Google Earth KML tiles from bathymetry Mercator img grid
img2grd	Extract a subset from an img file in Mercator or Geographic format
gpsgridder	Interpolate GPS velocity vectors using Green' s functions
其他脚本	
gmt5syntax	将旧的 GMT4 脚本修改成新的 `gmt <module>` 语法
gmtswitch	GMT 多版本之间切换

13.1 basemap

官方文档 basemap
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

13.1.1 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 basemap -R-180/180/-70/70 -JM10c -Bx60 -By30 -pdf test
```

-L[g|j|J|n|x]<refpoint>+c[<slon>/]<slat>+w<length>[e|f|k|M|n|u][+a<align>][+f][+j<justif

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 basemap -R90/110/30/40 -JM10c -Bx5 -By5 -Lg95/35+c35+w800k+lscale+u+f -png test

gmt basemap -R90/180/-50/50 -Jm0.025i -Bafg -B+tMercator -Lx1i/1i+c0+w5000k -png␣
↪mercator
```

-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"。标签的文字属性由*FONT_TITLE* 控制

下图展示了方向玫瑰图的效果图：

Source Code

图 1: 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>][+l<w>,
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

13.1.2 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 basemap -R0/10/0/10 -JX10c/5c -Aoutline.txt
```

-F[d|l|t][+c<clearances>][+g<fill>][+i[[<gap>/]<pen>]][+p[<pen>]][+r[<radius>]][+s[[<dx>
控制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

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.1.3 示例

下面的脚本绘制了磁场玫瑰图：

```
#!/bin/bash
# Magnetic rose with a specified declination
gmt begin basemap_ex4 pdf,png
gmt basemap -R-10/-2/12.8812380332/0.661018975345r -JOc0/0/50/60/7i -Baf -BWSne -X1.25i \
```

（下页继续）

(续上页)

```
        --MAP_ANNOT_OBLIQUE=34 --FONT_ANNOT_PRIMARY=12p
gmt basemap -Tmg-2/0.5+w2.5i+d-14.5+t45/10/5+i0.25p,blue+p0.25p,red+l+jCM \
        --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
gmt basemap -DjTR+w2.9i/3.9i+o0.05i -F+p+ggray95
echo "5.5 3.8 GMT DEFAULTS" | gmt text -R0/7/0/5 -Jx1i -F+f14p,Helvetica-Bold+jCM
gmt text -F+f12p+jLM << EOF
4.1 3.50 FONT_TITLE
4.1 3.25 MAP_TITLE_OFFSET
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 end
```

图 2: 磁场玫瑰图

13.1.4 BUGS

1. 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)

13.2 begin

官方文档 begin
简介 初始化一个新的 GMT 现代模式会话

在 GMT 现代模式下，一个 GMT 绘图总是以 **gmt begin** 开始，以 **gmt end** 结束。

begin 模块告诉 GMT 要开始一个新的现代模式会话。如果你的脚本只绘制一张图，那么你可以直接指定要生成的图片的文件名和文件格式。如果你的脚本绘制多张图，则你需要使用*figure* 来分别为每张图指定文件名和文件格式。现代会话模式下，每个会话均独立于其他进程，每个会话负责管理各自的配置参数、命令历史等，因而可以同时执行多个 GMT 会话而不会互相干扰。

除了可以指定图片文件名和文件格式之外，还可以通过 *options* 指定生成图片过程中所使用的*psconvert* 选项。

13.2.1 语法

gmt begin [*prefix*] [*formats*] [*options*] [**-V**[*level*]]

13.2.2 可选选项

prefix 图片文件名前缀，默认值为 **gmtsession**。图片文件名后缀由 *formats* 自动决定。

如果一个 GMT 会话只用于进行计算而不绘图，或者需要绘制多张图，则不需要指定该参数。

注解: 文件名中应尽量避免出现空格。若存在空格，则文件名必须用单引号括起来。

---

formats 图片文件格式。多个格式之间可以用逗号分开。默认图片格式为 **pdf**，由参数*GMT_GRAPHICS_FORMAT* 控制。

GMT 支持输出如下矢量图片格式：

- `pdf`：Portable Document Format [默认格式]
- `ps`：Plain PostScript
- `eps`：Encapsulated PostScript

GMT 支持输出如下位图图片格式：

- `bmp`：Microsoft Bit Map
- `jpg`：Joint Photographic Experts Group Format
- `png`：Portable Network Graphics（不透明背景）
- `PNG`：Portable Network Graphics（透明背景）
- `ppm`：Portable Pixel Map

- `tif`：Tagged Image Format File

options GMT 现代模式本质上是先生成 PS 文件，再通过调用*psconvert* 自动转换成用户指定的图片格式。此处可以设置要传递给模块*psconvert* 的选项，多个选项之间用逗号分隔。

默认值为 **A**，表示将 **-A** 选项传给*psconvert*。

其他可选的选项包括：

- **A**[*args*]: 裁剪图片
- **C***args*: 额外传递给 Ghostscript 的选项
- **D***dir*: 指定图片的输出目录
- **E***dpi*: 设置图片分辨率
- **H***factor*: 对图片做平滑以避免混叠
- **M***args*: 为当前图片叠加前景图片或背景图片
- **Q***args*: 设置图像和文本的抗锯齿选项
- **S**：把 Ghostscript 命令输出到标准错误输出，且不删除所有中间文件

详细解释见*psconvert* 的说明文档。

-V[***level***] ***(more …)*** 设置 verbose 等级 [c]

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）

-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明

-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明

13.2.3 示例

开始一个会话，并使用默认值。此时会生成名为 gmtsession.pdf 的图片文件:

```
gmt begin
gmt ...
gmt end
```

开始一个 GMT 会话，并指定图片名为 *Figure_2*，图片格式为 PDF 和 PNG 格式:

```
gmt begin Figure_2 pdf,png
gmt ...
gmt end show
```

设置额外的参数以控制生成图片时的额外空白:

```
gmt begin map pdf,png A+m1c
gmt ...
gmt end show
```

13.2.4 PS 文件注意事项

如果用户想要输出 PS 格式的图片，则应额外留意画布尺寸。对于其他图片格式而言，GMT 默认使用无穷大（10 米 x10 米）的画布。而对于 PS 格式而言，GMT 则默认使用 A4 大小的画布。若用户绘制的图片超过 A4 纸张的大小，则可能会造成显示不完全。针对这种情况，建议用户修改参数 PS_MEDIA 以显式指定纸张大小。例如:

```
gmt begin map ps
gmt set PS_MEDIA A3
gmt ...
gmt end
```

13.2.5 UNIX shell 注意事项

现代模式的工作原理是，在使用 **gmt begin** 时利用父进程 ID 创建唯一的会话目录，并将很多信息保存到该会话目录中。脚本中接下来的命令拥有共同的父进程 ID，因而接下来的命令可以向唯一会话目录中写入信息或读取信息，以实现多个命令之间的互相通信。然而，UNIX 下某些 shell 的实现不完全统一，脚本执行过程中父进程 ID 可能出现变化，后面执行的命令无法正确获取前面命令的父进程 ID，因而导致命令之间的信息交流出现错误。最常见的情况是在使用 UNIX 管道时，可能会生成子 shell 进而导致父进程 ID 出现变化。

如果你在 GMT 现代模式脚本中使用了管道，执行过程中出现了类似无法找到目录 `gmt6.#####` 这样的错误，这极有可能是你所使用的 UNIX shell 存在此类问题。解决办法是，在脚本开始的地方设置环境变量 **GMT_SESSION_NAME** 为进程 ID。在 Bash shell 应该是:

```
export GMT_SESSION_NAME=$$
gmt begin
gmt ...
gmt end show
```

在 C shell 中应该是:

```
setenv GMT_SESSION_NAME $$
gmt begin
gmt ..
gmt end show
```

13.2.6 相关模块

clear, *docs*, *end*, *figure*, *inset*, *subplot*, *gmt*

13.3 clear

官方文档 clear
简介 删除缓存目录、数据目录或会话目录，以及当前配置文件

13.3.1 语法

gmt clear all | **cache** | **defaults** | **data** | **sessions** [**-V**[*level*]]

13.3.2 可选选项

all 删除所有项目，包括缓存目录（~/**.gmt/cache**）、数据目录（~/**.gmt/server**）、会话目录（~/**.gmt/sessions**）以及当前配置文件
defaults 删除现代模式下当前会话的 GMT 配置文件
cache 删除 GMT 缓存目录（默认为 ~/**.gmt/cache**）及其内容
data 删除 GMT 数据目录（默认为 ~/**.gmt/server**）及其内容
sessions 删除 GMT 会话目录（默认为 ~/**.gmt/sessions**）及其内容

-V[***level***] *(more …)* 设置 verbose 等级 [c]

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）
-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明
-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明

13.3.3 示例

清空 GMT 缓存目录:

```
gmt clear cache
```

删除现代模式会话下的当前配置文件:

```
gmt clear defaults
```

13.3.4 相关模块

begin, *docs*, *end*, *figure*, *inset*, *subplot*, *gmt*

13.4 clip

官方文档 clip
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，clip 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 `clip -C` 以关闭裁剪区域。

13.4.1 必选选项

-J -R

13.4.2 可选选项

-A[m|p|x|y] 修改两点间的连接方式，详情见*plot* 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。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

13.4.3 示例

```
# 打开裁剪路径
gmt begin
gmt clip my_region.xy -R0/40/0/40 -Jm0.3i
# 其他绘图命令
# ...

# 关闭裁剪路径
gmt clip -C
gmt end show
```

13.5 coast

官方文档 coast
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。

13.5.1 必选选项

-J -R

13.5.2 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*plot* 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<precent>：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>][+f][+j<j`
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 coast ，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"。标签的文字属性由 *FONT_TITLE* 控制

-Tm[g|j|J|n|x]<refpoint>+w<width>[+d<dec>[/<dlabel>]][+i<pen>][+j<justify>][+l<w>,<e>,<
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，岛里又有个湖。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

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.5.3 示例

```
gmt coast -R-30/30/-40/40 -Jm0.1i -B5 -I1/1p,blue -N1/0.25p,- \
        -I2/0.25p,blue -W0.25p,white -Ggreen -Sblue -png africa
```

```
gmt coast -R-30/-10/60/65 -Jm1c -B5 -Gp100/28 -png iceland
```

将非洲区域裁剪出来，并在其中的陆地部分绘制地形:

```
gmt begin map png
gmt coast -R-30/30/-40/40 -Jm0.1i -B5 -Gc
gmt grdimage etopo5.nc -Ccolors.cpt
gmt coast -Q
gmt end
```

绘制部分国家的国界线（似乎有 BUG）:

```
gmt coast -JM6i -Baf -EGB,IT,FR+gblue+p0.25p,red+r -EES,PT,GR+gyellow -png map
```

提取冰岛的海岸线数据:

```
gmt coast -R-26/-12/62/68 -Dh -W -M > iceland.txt
```

13.5.4 FAQ

1. 错误消息:

   ```
   coast: 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 `--disbale-netcdf4` 选项

13.6 colorbar

官方文档 colorbar
简介 在地图上绘制灰色或彩色色条（colorbar）

13.6.1 必选选项

`-D[g|j|J|n|x]<refpoint>[+w<length>[/<width>]][+e[b|f][<length>]][+h|v][+j<justify>][+m[a`

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

13.6.2 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 = log10(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。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.6.3 示例

```
gmt begin map
gmt makecpt -T-200/1000/100 -Crainbow
gmt colorbar -C -Dx8c/1c+w12c/0.5c+jTC+h -Bxaf+l"topography" -By+lkm
gmt end
```

13.7 coupe

官方文档 coupe
简介 绘制震源机制解的剖面图

meca 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*coupe* 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*meca* 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

13.7.1 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

13.7.2 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*plot* 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*meca* 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 只是为了使得该命令可以使用与*meca* 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。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.7.3 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=pscoupe_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

# 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.

# 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 \
```

（下页继续）

(续上页)

```
        -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.*
```

图 3: coupe 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=pscoupe_ex2.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.*
```

图 4: coupe 示例图 2

13.8 docs

官方文档 docs

简介 打开指定模块的 GMT 官方的 HTML 文档

docs 用系统默认浏览器打开指定模块的 HTML 文档。若本地存在 HTML 文档，则优先使用本地文档；否则则打开 GMT 文档网站。除了指定模块外，还可以打开 *gmt*、*cookbook*、*gallery*、*settings*、*api* 和 *tutorial* 页面。

13.8.1 语法

gmt docs [**-Q**] [**-S**] [**-V**[*level*]] *module-name* [*-option*]

13.8.2 必须参数

module-name 要查看文档的模块名。

除了模块名之外，还支持几个特殊的名字:

- **gmt**: 打开 GMT 命令的帮助文档
- **cookbook**: 打开 GMT 参考手册
- **tutorial**: 打开 GMT 入门教程
- **api**: 打开 GMT API 手册
- **settings**: 打开 GMT 配置参数文档
- **gallery**: 打开 GMT 图库

13.8.3 可选选项

-Q 仅显示文档的网页链接而不打开文档，适用于没有安装图形界面的服务器。若使用该选项，则其必须是 **docs** 的第一个选项。

-S **docs** 默认优先打开本地文档，若本地文档不存在，则打开 GMT 文档网站。该选项强制 **docs** 使用浏览器打开 GMT 文档网站上的网页。

-*option* 指定选项（例如 **-R**），则 **docs** 会打开模块文档并定位到模块文档的该选项处。

注意，该功能对本地文档无效，因而当使用该功能时 GMT 会默认添加 **-S** 选项以打开远程文档。

-V[***level***] *(more …)* 设置 verbose 等级 [c]

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）

-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明

-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明

13.8.4 示例

查看*grdimage* 的文档:

```
gmt docs grdimage
```

查看*grdimage* 的文档链接:

```
gmt docs -Q grdimage
```

查看*grdimage* 在 GMT 文档网站上的链接:

```
gmt docs -Q -S grdimage
```

查看*coast* 的 **-B** 选项:

```
gmt docs coast -B
```

查看 GMT 配置参数列表:

```
gmt docs settings
```

查看图库:

```
gmt docs gallery
```

13.8.5 相关模块

begin, *clear*, *end*, *figure*, *inset*, *subplot*, *gmt*

13.9 end

官方文档 end
简介 结束现代模式会话，生成并显示图片

end 模块用于结束由 **begin** 模块创建的 GMT 当前会话，并在当前目录生成指定格式的图片文件，还支持使用系统默认图片阅读器自动打开生成的图片。

13.9.1 语法

gmt end [**show**] [**-V**[*level*]]

13.9.2 可选参数

show 用系统默认图片阅读器自动打开所有当前会话生成的图片文件。

注解: 若环境变量 **GMT_END_SHOW** 值为 **off**，则 **gmt end show** 不会打开图片文件。

-V[***level***] *(more …)* 设置 verbose 等级 [c]

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）
-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明
-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明

13.9.3 示例

用 **begin** 创建一个会话，绘图，并用 **end** 结束当前会话:

```
gmt begin
gmt basemap -R0/10/0/10 -JX10c -Baf
gmt end show
```

13.9.4 禁用自动显示

对于单行模式的命令以及以 **gmt end show** 结尾的脚本，GMT 会自动显示生成的图片。如果想不修改脚本但并自动显示图片，则可以设置为环境变量 **GMT_END_SHOW** 为 `off`。

13.9.5 相关模块

begin, *clear*, *docs*, *figure*, *inset*, *subplot*, *gmt*

13.10 figure

官方文档 figure
简介 设置当前图片的属性

一个 GMT 现代模式会话下可以绘制一张或多张图件。对于一个会话中绘制多张图件的情况，**figure** 模块可以指定当前图件的文件名和格式。

该模块必须在你开始向当前图件绘图之前执行，且每次调用 **figure** 模块都会开启一张新的图件。你也可以再次使用 **figure** 模块在多个图件之间来回切换，但此时不能再指定 *formats* 或 *options* 等选项。一个会话中的每张图都有各自的历史记录和配置参数，因而不同图之间的 **-R** 和 **-J** 选项等信息不会出现错乱。

13.10.1 语法

gmt figure *prefix* [*formats*] [*options*] [**-V**[*level*]]

13.10.2 必须选项

prefix 图片文件名前缀，默认值为 **gmtsession**。图片文件名后缀由 *formats* 自动决定。

注解: 文件名中应尽量避免出现空格。若存在空格，则文件名必须用单引号括起来。

13.10.3 可选选项

formats 图片文件格式。多个格式之间可以用逗号分开。默认图片格式为 **pdf**，由参数*GMT_GRAPHICS_FORMAT* 控制。

GMT 支持输出如下矢量图片格式：

- `pdf`：Portable Document Format [默认格式]
- `ps`：Plain PostScript
- `eps`：Encapsulated PostScript

GMT 支持输出如下位图图片格式：

- `bmp`：Microsoft Bit Map
- `jpg`：Joint Photographic Experts Group Format
- `png`：Portable Network Graphics（不透明背景）
- `PNG`：Portable Network Graphics（透明背景）
- `ppm`：Portable Pixel Map
- `tif`：Tagged Image Format File

options GMT 现代模式本质上是先生成 PS 文件，再通过调用*psconvert* 自动转换成用户指定的图片格式。此处可以设置要传递给模块*psconvert* 的选项，多个选项之间用逗号分隔。

默认值为 **A**，表示将 **-A** 选项传给*psconvert*。

其他可选的选项包括：

- **A**[*args*]: 裁剪图片
- **C***args*: 额外传递给 Ghostscript 的选项
- **D***dir*: 指定图片的输出目录
- **E***dpi*: 设置图片分辨率
- **H***factor*: 对图片做平滑以避免混叠
- **M***args*: 为当前图片叠加前景图片或背景图片
- **Q***args*: 设置图像和文本的抗锯齿选项
- **S**：把 Ghostscript 命令输出到标准错误输出，且不删除所有中间文件

详细解释见*psconvert* 的说明文档。

-V[***level***] ***(more …)*** 设置 verbose 等级 [c]

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）

-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明

-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明

13.10.4 PS 文件注意事项

如果用户想要输出 PS 格式的图片，则应额外留意画布尺寸。对于其他图片格式而言，GMT 默认使用无穷大（10 米 x10 米）的画布。而对于 PS 格式而言，GMT 则默认使用 A4 大小的画布。若用户绘制的图片超过 A4 纸张的大小，则可能会造成显示不完全。针对这种情况，建议用户修改参数 PS_MEDIA 以显式指定纸张大小。例如:

```
gmt begin
gmt figure map ps
gmt set PS_MEDIA A3
gmt ...
gmt end
```

13.10.5 示例

在当前的现代模式下开启一个新绘图，图片名为 Regional，格式为 PDF 和 EPS 格式:

```
gmt begin
gmt figure Regional pdf,eps
gmt ...
gmt end show
```

开启一个新绘图，名为 GlobalMap，图片格式为 JPEG，并对图片进行裁剪，四边均保留 1 厘米的空白:

```
gmt begin
gmt figure GlobalMap jpg A+m1c
gmt ...
gmt end show
```

在一个会话中绘制两张图，并在两张图之间来回切换:

```
gmt begin

# 在 Fig1 中绘制
gmt figure Fig1 png
gmt ...

# 在 Fig2 中绘制
gmt figure Fig2 png
gmt ...

# 切换回 Fig1
gmt figure Fig1
gmt ...

# 切换回 Fig2
gmt figure Fig2
gmt ...

gmt end show
```

13.10.6 技术细节

如果你在为 GMT 设计外部接口并且你不希望在会话结束时自动转换并生成图片，你可以指定图片文件名或图片格式为 -。

13.10.7 相关模块

begin, *clear*, *docs*, *end*, *inset*, *subplot*, *gmt*

13.11 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。

13.11.1 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 or 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。

13.11.2 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 ((abs(n_left - n_right)) / (n_left + n_right)) > 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。

13.11.3 示例

数据 cruise.gmtd 中包含了四列数据，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.gmtd -T0/1.0e5/2000 -FG10000 -N3 -V > filtered_cruise.gmtd
```

```
gmt filter1d v3312.dt -FM50 -T0/100000/25 -L10 -S0.3 > v3312_filt.dt
```

13.12 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。

13.12.1 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.

13.12.2 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].

13.12.3 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 5: fitcircle 示例

13.13 gmt

官方文档 gmt
说明 GMT 主程序

GMT 的数据处理和绘图功能均通过使用 **gmt** 调用相应模块来实现，其基本语法为：

gmt *module module-options*

其中 *module* 是 GMT 模块名，*module-options* 是模块所支持的选项。

除了调用模块外，GMT 还有自己的一些选项可以使用：

- **gmt** **–help**：列出 GMT 提供的所有模块名及其功能
- **gmt** **–new-script**[=*L*]：生成现代模式模板脚本。默认会根据当前 Shell 环境确定脚

本语言，用户也可以自行指定脚本语言 *L*（可以取 **bash**、**csh** 或 **batch**）

- **gmt –show-bindir**：显示 GMT 的 bin 目录
- **gmt –show-citation**：显示 GMT 的参考文献引用信息
- **gmt –show-cores**：显示当前计算机可以使用的核数
- **gmt –show-datadir**：显示 GMT 的数据目录，默认为空
- **gmt –show-dataserver**：显示 GMT 远程数据服务器的网址
- **gmt –show-doi**：显示当前版本的 DOI
- **gmt –show-library**：显示 GMT 共享库文件的路径
- **gmt –show-modules**：列出 GMT 的所有模块名
- **gmt –show-plugindir**：显示 GMT 的插件目录
- **gmt –show-sharedir**：显示 GMT 的 share 目录
- **gmt –version**：显示 GMT 版本
- **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

13.13.1 示例

使用 **gmt –new-script** 会根据用户当前使用的 SHELL 环境生成对应的模板脚本。例如，在 Bash 环境下可以使用如下命令生成模板脚本:

```
gmt --new-script > plot.sh
```

在 Windows DOS 下默认会生成 batch 脚本。可以直接指定要生成某种脚本语言的模板:

```
gmt --new-script=bash > plot.sh
```

13.14 gmt-config

gmt-config 是 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，以供其它程序使用。

gmt-config 提供了如下选项:

```
--help        显示此帮助信息并退出
--all         显示全部选项的值
--bits        库文件是 32 位还是 64 位
--cc          编译过程中使用的 C 编译器
--cflags      C 预处理器和编译器的 CFLAGS: -I/opt/GMT/include/gmt
--datadir     GMT 数据目录，默认为空
--dataserver  GMT 数据服务器网址
--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
--libdir      library 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
```

13.15 gmt5syntax

官方文档 gmt5syntax
简介 将旧的 GMT4 脚本修改成 GMT5 的 **gmt** *module* 语法

GMT4 中每个模块都是一个单独的命令，这可能会造成命令冲突。从 GMT5 开始，GMT 只提供一个命令，即 **gmt**。其他所有命令都变成了 **gmt** 的模块。

因而，GMT4 中的 **psbasemap** 命令在 GMT5 中应写作 **gmt psbasemap**。

gmt5syntax 是一个 Perl 脚本，作用是在命令前加上 **gmt** 将 GMT4 的语法改成 GMT5 的新语法。该脚本位于 `${GMTHOME}/share/tools` 目录下。

13.15.1 用法

```
gmt5syntax old_script > new_script
```

13.16 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。

13.16.1 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` ）i 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 end-points 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.

13.16.2 示例

```
gmt connect segment_*.txt -Tf0.1 > new_segments.txt
```

```
gmt connect my_lines.txt -T150e -DMap_segment_%04d.dat
```

13.17 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

13.17.1 必选选项

无

13.17.2 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

13.17.3 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 -D | grep LABEL
```

13.18 gmtget

官方文档 gmtget
简介 列出单个或多个 GMT 参数的当前值

13.18.1 语法

gmt get [**-G***defaultsfile*] [**-L**] *PARAMETER1* [*PARAMETER2 PARAMETER3* …]

13.18.2 必选选项

PARAMETER 要查看的 GMT 参数名。GMT 中参数列表见配置参数

13.18.3 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。

13.18.4 示例

列出一个参数的当前值:

```
$ gmt get MAP_FRAME_TYPE
fancy
```

列出多个参数的当前值:

```
$ gmt get MAP_GRID_CROSS_SIZE_PRIMARY MAP_GRID_CROSS_SIZE_SECONDARY
24p,Helvetica,black 16p,Helvetica,black
```

使用 **-L** 选项一行显示一个参数值:

```
$ gmt get FONT_TITLE FONT_LABEL MAP_FRAME_TYPE -L
24p,Helvetica,black
16p,Helvetica,black
fancy
```

13.18.5 相关模块

gmt.conf, *gmtdefaults*, *gmtset*

13.19 gmtinfo

官方文档 gmtinfo
简介 从表数据中提取极值信息

13.19.1 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。

13.19.2 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 plot -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。

13.20 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。

13.20.1 必选选项

无

13.20.2 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[<dx>/<d
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 ）

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.20.3 示例

单独绘制一个 2 英寸宽的 GMT logo:

```
gmt logo -Dx0/0+w2i -pdf map
```

将 GMT logo 作为一个图层放在当前底图的左上角:

```
gmt begin logo
gmt ...
gmt logo -DjTR+o0.1i/0.1i+w3i -F+glightblue+s
gmt end
```

13.21 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）

13.21.1 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*coast* 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*plot* 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<precent>：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。

13.21.2 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[cfglrsz] 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 。

13.21.3 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
```

13.22 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。

13.22.1 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
```

13.22.2 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>

-[BJRXxYycp]<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 。

13.22.3 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。

13.23 gmtsimplify

官方文档 gmtsimplify
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。

13.23.1 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。

13.23.2 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 xylines.d -T0.45 > new_xylines.d
```

13.23.3 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.

13.24 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。

13.24.1 必选选项

`<files>` 任意数据文件名

13.24.2 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。

13.25 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。

13.25.1 必选选项

`<grid>` 要转换的 2D 网格文件

13.25.2 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 。

13.25.3 示例

```
gmt grd2xyz hawaii_grv.nc > hawaii_grv.xyz
```

```
gmt grd2xyz raw_data.nc -ZTLf > hawaii_grv.b
```

13.26 grdblend

官方文档 grdblend
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。

13.26.1 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

13.26.2 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
-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

13.26.3 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 grdblend 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 grdblend MB_*.nc -Gblend.nc -R0/360/-90/90 -I1m/1m -V
```

13.27 grdclip

官方文档 grdclip
简介 对网格文件的 Z 值范围做裁剪

13.27.1 必选选项

<ingrid> 输入网格文件名
-G<outgrid> 输出网格文件名

13.27.2 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

13.27.3 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
```

13.28 grdconvert

官方文档 grdconvert
简介 在不同的网格格式之间互相转换

13.28.1 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，则需要使用 -N 选项，以保证不将 GMT 头段结构写到文件中。

13.28.2 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
```

To creat a dumb file saved as a 32 bits float GeoTiff using GDAL:

```
gmt grdmath -Rd -I10 X Y MUL = lixo.tiff=gd:GTiff
```

13.29 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

13.29.1 必选选项

<ingrid> 输入网格文件名
-G<outgrid> 输出网格文件名

13.29.2 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

13.29.3 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
```

13.29.4 相关

- *grdpaste*
- *grdsample*

13.30 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

13.30.1 必选选项

<grid> 要修改的 2D 网格文件

13.30.2 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<title>][+r<re

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。

13.30.3 示例

13.31 grdimage

官方文档 grdimage
简介 在地图上绘制网格数据

13.31.1 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。

13.31.2 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

-U[*label*][+**c**][+**j***just*][+**o***dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级, 可取值范围为 0 (不透明) 到 100 (全透明)

13.31.3 示例

使用默认的光照效果:

```
gmt grdimage stuff.nc -JX6i+ -I+d -pdf map
```

13.32 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

13.32.1 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: ETOPO5 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: 0.0833333333333 name: longitude [degrees_east] nx: 601
test.nc: y_min: -30 y_max: 30 y_inc: 0.0833333333333 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;

13.32.2 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    0.0833333333333 0.0833333333333 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
-I0.0833333333333/0.0833333333333

$ 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为网格信息。

-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
```

13.33 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。

13.33.1 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

13.33.2 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*plot* 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<precent>：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*coast* 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。

13.33.3 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。

13.33.4 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
```

13.34 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

13.34.1 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

13.34.2 可选选项

`-A[m|p|x|y]` 控制两点之间的连接方式，见*plot* 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。

13.34.3 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。

13.34.4 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
```

13.35 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。

13.35.1 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

13.35.2 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
```

13.36 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。

13.36.1 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

13.36.2 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，将矩形区域变换成地理区域。
-Mc|i|p 指定投影后的测量单位，默认值由参数 PROJ_LENGTH_UNIT 决定。

13.36.3 示例

将地理网格数据变换成 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
```

13.37 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 grdfft 或 grdfilter 对网格文件做滤波。

13.37.1 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

13.37.2 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。

13.37.3 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
```

13.38 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。

13.38.1 选项

-G<grdfile> 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。

13.38.2 相关

- *project*

13.39 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：

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.

13.39.1 必选选项

`<gridfile>` 2D 网格文件名
`-N<n_model>[+r]` 指定要拟合的模型。

`<n_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。

13.39.2 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。

13.39.3 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
```

13.40 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。

13.40.1 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

13.40.2 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。

13.40.3 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 -pdf gradient
```

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 -pdf globe
```

13.41 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。

13.41.1 必选选项

<grdfile> 输入的 2D 网格文件名

13.41.2 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 <unit>^2 ，而输出的体积单位则是 <unit>^2 * z_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> 。

13.41.3 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=250m 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
```

13.41.4 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.

13.42 histogram

官方文档 histogram
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。

13.42.1 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

13.42.2 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<font>][+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<font> 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。

-U[*label*][+**c**][+**j***just*][+**o***dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.42.3 示例

访问绘制直方图 以查看更多示例。

13.43 image

官方文档 image
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

13.43.1 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： **image** 模块并不支持 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
```

13.43.2 可选选项

`-D[g|j|J|n|x]<refpoint>+r<dpi>+w[-]<width>[/<height>][+j<justify>][+n<nx>[/<ny>]][+`
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>` 以及原图的横纵比进行缩放；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>]][+s[<dx>/`
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<radisu>` 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

-U[*label*][+**c**][+**j***just*][+**o***dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.43.3 示例

```
gmt image logo.jpg -Dx0/0+w1i -F+pthin,blue -png image
```

```
gmt image tiger.eps -Dx2i/1i+jTR+w3i -png image
```

```
gmt image 1_bit.ras -Gbbrown -Gfred -Dx0/0+w1c+n5 -png image
```

13.44 inset

官方文档 inset
简介 管理和设置图中图模式

inset 模块用于管理图中图模式。即在纸张上规划出一小片区域，并限制接下来的绘制操作均只在该小区域内操作。

inset 模块包含两个命令：

- **inset begin** 用于设置图中图模式，其定义了图中图区域的位置和大小
- **inset end** 用于结束图中图模式，所有的操作都会回到原大图中。

在图中图中，用户可以使用任意的投影方式和投影区域。若投影方式中底图宽度或比例用 ? 表示，则会根据 **-D** 选项设置的小图大小自动确定小图的最佳尺寸。

13.44.1 inset begin 语法

gmt inset begin -D*inset-box* [**-F***box*] [**-M***margins*] [**-N**] [**-V**[*level*]] [**-PAR**=*value*]

13.44.2 必须选项

-D*xmin/xmax/ymin/ymax*[+r][+u*unit*] 类似于-*R* 选项，通过指定大图中的一个矩形区域作为小图的绘图区域。

- +**r**: 与-*R* 选项类似，表明坐标为矩形区域的左下角和右上角坐标。
- +**u***unit*: 与-*R* 选项类似，表明此时的坐标为投影后坐标

-D[g|j|J|n|x]***refpoint***+w***width***[/***height***][+j***justify***][+o***dx***[/***dy***]] 指定小图区域的位置和大小。

- **g**|**j**|**J**|**n**|**x**]*refpoint*: 指定小图区域的参考点，通常用 **j** 比较方便。例如 `jTL` 表示小图区域位于左上角，**jBR** 表示小图区域位于右下角。详情见绘制修饰物 一节
- +**w***width*[/*height*]: 设置小图区域的宽度和高度
- +**j***justify*: 控制小图区域的锚点，见锚点 一节
- +**o***dx*[/*dy*]: 控制小图区域相对于参考点的额外偏移量，见绘制修饰物 一节

13.44.3 可选选项

-F[+c***clearances***][+g***fill***][+i[[***gap***/]***pen***]][+p[***pen***]][+r[***radius***]][+s[[***dx***/***dy***/][***shade***]]] 设置小图区域的背景面板属性。

- +**g***fill*: 设置背景面板填充色，默认不填充
- +**p**[*pen*]]: 设置背景面板的边框，默认使用参数*MAP_FRAME_PEN* 的值
- +**r**[*radius*]: 设置背景面板边框为圆角矩形，*radius* 为圆角半径
- +**i**[[*gap*/]*pen*]: 在背景面板边框的内部绘制一个内边框，*gap* 为内外边框的距离，*pen* 为内边框线条属性
- **+c*clearances*: 面板边框内部的额外空白**
  - +**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
- +**s**[[*dx*/*dy*/][*shade*]]: 为背景面板添加阴影效果。*dx*/*dy* 为阴影相对面板的偏移量，*shade* 为阴影颜色。

详情见绘制修饰物 一节。

-M*margins* 小图区域内部的额外空白区域 [默认值没有空白]。其可以取三种形式：

1. 一个值，表示四个边的空白相同
2. 两个用斜杠分隔的值，分别设置水平和垂直方向的空白
3. 四个用斜杠分隔的值，分别设置左右下上四条边的空白

-N 不裁剪超过小图区域边界的部分。

-V[***level***] *(more …)* 设置 verbose 等级 [c]

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）

-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明
-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明
–PAR=*value* 临时修改 GMT 参数的值，可重复多次使用。参数列表见 gmt.conf。

13.44.4 示例

示例见*GMT 图中图模式*。

13.45 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。

13.45.1 必选选项

`<kmlfiles>` 要转换的 KML 文件

13.45.2 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

13.45.3 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
```

13.46 legend

官方文档 legend
简介 在图上添加图例

13.46.1 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>[/<dy>]`
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。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.46.2 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，其后可以加上子选项 +e[b|f][<length>]][+h][+ma|c|l|u][+n[<txt>]
3. 还可以添加其他一些参数：-B、-I、-L、-M、-N、-S、-Z、-p 等，详情见*colorbar* 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*basemap* 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*text* 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*plot* 命令的 -S 选项。若 symbol 设置为 - 则表示绘制线段
2. dx1 是符号中心与左边框的距离
3. dx2 是 text 与左边框的距离
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

13.46.3 示例

```
#!/bin/bash
gmt begin legend pdf,png
gmt legend -R-10/10/-10/10 -JM6i -F+gazure1 -Dx0.5i/0.5i+w5i/3.3i+jBL+l1.2 -C0.1i/0.1i -B5f1 << EOF
# 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
```

（下页继续）

(续上页)

```
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
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
gmt end
```

图 6: legend 示例图 1

```
#!/bin/bash
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
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
```

(下页继续)

(续上页)

```
L - - R Houston
L - - R USA
L - - R 2,714,856
L - - R subtropical
L - - C N
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 begin legend pdf,png
gmt gmtset FONT_ANNOT_PRIMARY 12p  FONT_LABEL 12p
gmt legend -Dx0/0+w5.6i+jBL+l1.1 -C0.05i -F+p+gsnow1 -B0 table.txt
gmt end
rm -f table.txt t.cpt
```

Eight Largest Cities in North America						
	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 7: legend 示例图 2

13.47 makecpt

官方文档 makecpt
简介 制作 GMT CPT 文件

13.47.1 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

13.48 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

13.48.1 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）。

13.48.2 -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
```

13.48.3 -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

13.49 meca

官方文档 meca
简介 在地图上绘制震源机制解

13.49.1 -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 meca -Rg -JN15c -Sa2c -pdf meca << 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 meca -Rg -JN15c -Sc2c -pdf meca << 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 meca -Rg -JN15c -Sm2c -pdf meca << 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

13.49.2 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<pointsize>] 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*plot* 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。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.49.3 示例

访问绘制震源球 以查看示例。

13.50 plot

官方文档 plot
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。

13.50.1 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:

Source Code

图 8: plot -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 plot -JX10c/10c -R0/10/0/10 -B1 -Cblue,red -W2p -png test << 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 <dx>/<dy> 距离。若未指定 <dy> ，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 plot -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Ex -W2p -png test
```

X 方向误差为 1，Y 方向误差为 0.5:

```
echo 5 5 1 0.5 | gmt plot -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Exy -W2p -png test
```

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 plot -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Exy+a -W2p -png test
```

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 plot -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -EX -Gred -W2p -png test
```

- 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 表面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，plot 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 xmin, xmax, or x0
5. +yb|t|<y0> connect first and last point to anchor points at either ymin, ymax, or y0.

Polygon may be painted (**-G**) and optionally outlined by adding **+p***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 plot -R0/360/-60/60 -JM10c -Bx60 -By15 -Sc2c -png test << 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

-W[<pen>][<attr>] 设置线段或符号轮廓的画笔属性。

1. <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

-U[*label*][+**c**][+**j***just*][+**o***dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.50.2 -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 <symbol> 指定了符号类型，<size> 为符号的大小，<u> 为 <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 ... 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 ACDGHINST，<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：

Source Code

图 9: plot -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 plot -R0/10/0/5 -JX15c/5c -B1 -Sb1cb -png test << 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变换成角度
- 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 plot -R0/360/-90/90 -JN15c -B60 -Se -png test
```

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 plot -R0/360/-90/90 -JX10c -B60 -SE -png test
```

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 plot -R0/360/-90/90 -JN15c -B60 -SE -png test
echo 80 0 45 200d  100d  | gmt plot -R0/360/-90/90 -JN15c -B60 -SE -png test2
```

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 plot -R0/360/-90/90 -JN15c -B60 -SE -png test
```

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 plot -R0/360/-90/90 -JN15c -B60 -SE- -png test << EOF
180 0 60d
180 0 120d
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
gmt begin plot_-Sf pdf,png
gmt basemap -R150/200/20/50 -JM15c -B10
gmt plot -Sf2c/0.1i+l+b -Gblack -W << EOF
155 30
160 40
EOF
gmt plot -Sf2c/0.1i+l+c -Gblue -W << EOF
165 30
170 40
EOF
gmt plot -Sf2c/0.1i+l+f -Gred -W << EOF
175 30
180 40
EOF
gmt plot -Sf2c/0.3i+l+s+o1 -Gyellow -W << EOF
185 30
190 40
EOF
gmt plot -Sf1c/0.1i+l+t -Gwhite -W -B10 << EOF
195 30
200 40
EOF
gmt end
```

图 10: plot -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*text* 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 度是，-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
gmt begin plot_-Sm pdf,png
gmt basemap -R0/4/0.5/1.5 -JX6c/3c -Bxa1g1 -Bya0.5g0.5 -BWSen
gmt plot -Sc0.15c -Gblack << EOF
1 1
3 1
EOF
gmt plot -Sm0.2c+b+e+g -Gblack -W0.5p,red << EOF
1 1 1 10 60
EOF
gmt plot -Sm0.2c+b+l -Gblack -W0.5p,blue << EOF
3 1 1 10 150
EOF
gmt end
```

图 11: plot -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>*<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<font>
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 plot -R0/10/0/10 -JX10c/10c -B1 -Sv1c+e -W2p -png test
```

-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
gmt begin plot_-Sw pdf,png
gmt basemap -R0/10/0/5 -Jx1c -Ba1g1
gmt plot -Sw2c -Glightblue << EOF
2 2 0 45
EOF
gmt plot -SW3c -Glightblue << EOF
6 2 0 45
EOF
gmt end
```

图 12: plot -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>[:<symbolinfo>]` 绘制 decorated line，即带有符号的线段。详见官方文档。

13.50.3 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**最后一列**

```
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。

13.50.4 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

13.50.5 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 plot -R0/10/0/10 -JX10c -B1 -png test << EOF
3 5
5 8
7 4
EOF
```

下面的脚本展示了 -F 选项的用法:

```
#!/bin/bash

# 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 plot -Sc0.2c -Ggreen -Wfaint t1.txt
    gmt plot -Sc0.2c -Gblue -Wfaint t2.txt
    echo $* | gmt text -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
```

(下页继续)

(续上页)

```
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
# -Fcs: 默认的连接方式
gmt begin plot_-F pdf,png
gmt basemap -R0/50/0/45 -Jx0.06i -Ba10 -BWSne
gmt plot -W1p t[12].txt
plotpts TWO DATA TABLES

# -Fra
gmt plot -W1p t[12].txt -Fra -X3.25i
plotpts DATASET ORIGIN

# -Frf
gmt plot -W1p t[12].txt -Frf -X-3.25i -Y3.15i
plotpts TABLE ORIGIN

# -Frs
gmt plot -W1p t[12].txt -Frs -X3.25i
plotpts SEGMENT ORIGIN

# -Fr10/35
gmt plot -W1p t[12].txt -Fr10/35 -X-3.25i -Y3.15i
plotpts FIXED ORIGIN
echo 10 35 | gmt plot -Sa0.4c -Gred -Wfaint

# -Fna
gmt plot -W1p t[12].txt -Fna -X3.25i
plotpts NETWORK

gmt end
rm t[12].txt
```

图 13: plot -F 选项示意图

-L 选项的示例：

```
#!/bin/bash
cat << EOF > t.txt
1 1
2 3
3 2
4 4
EOF
gmt begin plot_-L pdf,png
gmt basemap -R0/5/0/5 -JX3i -B0
gmt plot t.txt -Gred -W2p -L+xl
gmt plot t.txt -Gred -W2p -L+xr -X3.25i
gmt plot t.txt -Gred -W2p -L+x4.5 -X3.25i
gmt plot t.txt -Gred -W2p -L+yt -X-6.5i -Y3.25i
gmt plot t.txt -Gred -W2p -L+yb -X3.25i
gmt plot t.txt -Gred -W2p -L+y4 -X3.25i
gmt end
rm t.txt
```

图 14: plot -L 选项示意图

13.51 polar

官方文档 polar
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 meca 绘制震源球，再使用 polar 将每个台站的极性信息画在相应的震源球上。因而，通常 meca 和 polar 需要连在一起使用。

13.51.1 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 meca 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 meca 中 -S 选项中震源球尺寸相同

-S<symbol_type><size> 类似于 plot 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[<vecpar>]][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

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.51.2 示例

```
#!/bin/bash
```

（下页继续）

(续上页)

```
gmt begin polar_ex1 png,pdf
gmt set 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 meca -Fo -R85/89/25/50 -JX7i -M -Sm4i -N  -L -G150 -T0 << EOF
# 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 polar -D87/35 -M4i -N -Sc0.3i -Qe -B0 -B+t"Tibet earthquake (1997) - polarities distribution" <
↪< EOF
1 147.8 53 c
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
```

(下页继续)

(续上页)

```
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
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
gmt end
```

图 15: polar 示例

13.52 project

官方文档 project
简介 生成测线、将数据点投影到测线上

该模块具有三个主要功能：

1. 生成测线
2. 指定测线和数据点，得到该点在测线坐标系下的坐标，即下文中提及的 `p` 和 `q`
3. 指定测线和数据点，得到该点在测线上的投影点的坐标，即下文中提及的 `r` 和 `s`

以上三个功能均要求用户首先定义测线，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，如果要生成测线，输出测线上的点的坐标，可以使用 -G 选项。在指定测线后，用户再给出一个点 (x,y) ，可以得到该点在测线的投影点以及该点在测线坐标系下的坐标:

```
x  y   z   p   q   r   s
```

其中:

- x 和 y 是数据在原坐标系下的坐标
- z 是输入数据中的其余所有列
- p 和 q 是数据点 (x,y) 在测线坐标系下的坐标
- r 和 s 是数据点 (x,y) 在测线上的投影点在原坐标系下的坐标

可以使用 -F 选项设置要输出哪些变量。

下面详细解释一下这些变量的物理意义。

Source Code

图 16: project 示意图

图中的红点就是给出的点 (x,y)。绿色粗线为测线。测线上有 3 个点，蓝色和绿色两个点为测线的起点和终点。以测线起点为原点，以测线为 P 轴，在测线起点按右手螺旋法则做垂直于 P 轴的 Q 轴，构成测线坐标系。点 (x,y) 在测线坐标系的坐标即为 (p,q)。紫色点为点 (x,y) 在测线上的投影点，坐标为 (r,s)。

13.52.1 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

13.52.2 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。

13.52.3 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.xyp
```

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.xyp
```

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 -Frs -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
```

13.53 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。

13.53.1 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
```

13.53.2 可选选项

-A[u][<margins>][-][+p<paint>][+p[<pen>]][+r][+s[m]|S<width>[<u>]/<height>[<u>]]
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:

1. 一个数，四条边的额外边距相同，如 -A0.5c
2. 两个数字，分别指定 X 和 Y 方向的额外边距，如 -A0.5c/1c
3. 四个数字，分别指定左右下上四条边的边距，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 **g**raphics or **t**ext. 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（不透明背景）；
- `G`：PNG（透明背景）；
- `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。

13.53.3 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 的图件而言，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；

13.53.4 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。

13.53.5 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）

13.54 rose

官方文档 rose
简介 绘制极坐标下的直方图（sector 图、rose 图或 windrose 图）

13.54.1 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

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.55 sac

官方文档 sac
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: sac 模块修改自原 pssac 与 pssac2，其功能类似，但语法不同。

---

sac 模块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

13.55.1 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

13.55.2 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 -Frii 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

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.55.3 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 sac seis.SAC -JX10c/5c -R9/20/-2/2 -Baf -Fr -Gp+gblack -Gn+gred -png single
```

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 sac *.z -R200/1600/12/45 -JX15c/5c -Bx200+l'T(s)' -By5+lDegree -BWSen \
    -Ed -M1.5c -W0.5p,red -png distance_profile
```

利用 SAC 命令 `datagen sub tel *.z` 生成多个波形，将其绘制在地图上:

```
gmt begin map pdf
gmt sac *.z -JM15c -R-120/-40/35/65 -Baf -M1i -S300c
saclst stlo stla f *.z | gmt plot -St0.4c -Gblack -i1,2
gmt end
```

13.56 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。

13.56.1 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。

13.56.2 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
```

13.57 solar

官方文档 solar
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

13.57.1 必选选项

-I 和 -T 中必须要有一个。

13.57.2 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较正以及均时差, 单位为分钟

注解: 若没有通过 -Ilon/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 solar -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 clip -C 停止区域剪裁，见*clip* 。

-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<data> 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 solar -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 17: 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°50'（或日出，即太阳上边缘接触地平线）这段时间
  - 晚间曙光区是指太阳的几何中心位于地平线以下 0°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°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°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 begin terminator png
gmt coast -Rd -W0.1p -JQ0/14c -Ba -BWSen -Dl -A1000
gmt solar -W1p -Tdcna
gmt end
```

`-W[<pen>]` 设置晨昏线的画笔属性，见画笔 。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.57.3 示例

示例见 http://gmt-china.org/example/ex009/

13.57.4 BUGS

1. `-T+d<date>` 在取某些值时会段错误退出（v5.3.1）

13.58 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*X*dt 。
2. name_stem.ypower: Y(t) 的功率谱。单位是 Y*Y*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。

13.58.1 选项

-S<segment_size> <segment_size> 是一个 2 的指数数值，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/(segment_size*dt) 和 1.0/(2*dt) （即 Nyquist 频率）。在功率谱密度中的一个标准误差大约为 1.0/(n_data/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[xycnpago] 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。

13.58.2 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
```

13.59 subplot

官方文档 subplot
简介 管理和设置子图模式

subplot 模块可以将当前纸张分隔成若干个网格区域，每个区域内都可以包含一张单独的子图。

subplot 模块提供了三条指令:

- **subplot begin** 进入子图模式，并设置子图的布局
- **subplot set** 用于指定接下来的绘图操作在哪一个子图中进行
- **subplot end** 用于结束子图模式

在子图模式中，需要注意如下几点:

- **-X** 和 **-Y** 选项无法在子图模式中使用，可以使用 **-C** 选项作为替代
- 在使用 **-J** 选项时，可以使用 **?** 来指定地图宽度或比例尺，此时，GMT 会根据子图的大小自动确定最合适的地图尺寸
- 对于笛卡尔投影，若想要 X 和 Y 轴共用相同的比例尺，则可以使用 **-Jx?**

13.59.1 subplot begin 语法

gmt subplot begin *nrows***x***ncols* **-F**[**f**|**s**]*width*/*height*[+**f***wfracs*/*hfracs*][+**c***dx*/*dy*][+**g***fill*][+**p***pen*][+**w***pen*] [**-A**[*autolabel*][+**j**|**J***refpoint*][+**c***dx*/*dy*][+**g***fill*][+**p***pen*][+**o***dx*/*dy*][+**r**][+**R**][+**v**]] [**-C**[*side*]/*clearance*[**u**]] [**-B***parameters*] [**-J***projection*] [**-M***margins*] [**-R***region*]

[**-S***layout*] [**-T***title*] [**-V**[*level*]] [**-X**[**a**|**c**|**f**|**r**][*xshift*[**u**]]] [**-Y**[**a**|**c**|**f**|**r**][*xshift*[**u**]]] [**–PAR**=*value*]

13.59.2 必须选项

nrowsxncols 指定子图的行数和列数。

每一行和每一列均有相同的子图数目。注意：你无需在每个子图内都绘图。

-F[**f**|**s**]***width/height***[**+f*wfracs/hfracs***][**+c*dx/dy***][**+g*fill***][**+p*pen***][**+w*pen***] 指定图片的尺寸。有两种方式：

- **-Ff**: 直接指定整张图片的尺寸
- **-Fs**: 指定单个子图的尺寸

除此之外，还可以为整张图加上背景色和边框：

- **+p***pen*: 为整张图加上背景矩形边框
- **+w***pen*: 绘制子图区域之间的水平和垂直分割线
- **+g***fill*: 为整张图的背景矩形填充颜色
- **+c***dx/dy*: 设置背景矩形与整张图之间的额外空白

-F[**f**]***width/height***[**+f*wfracs/hfracs***] 指定整张图片的宽度 *width* 和高度 *height*。

这种情况下，GMT 会根据整张图片的尺寸以及子图的数目自动计算每张子图的尺寸。在计算子图尺寸时会考虑每个子图的刻度线、标注、标签所占据的空间，以及不同子图之间的间隔。整张图的最外圈的刻度线、标记和标签不算在整张图片尺寸之内。

默认所有行和列的尺寸都是相同的。若想要子图的每一行具有不同的高度，或者子图的每一列具有不同的宽度，则可以使用 **+f**，后面紧跟着一系列逗号分隔的宽度比例和以逗号分隔的高度比例。单个数则表示所有行或列有相同的宽度或高度。

例如，对于一个 2x2 的子图，使用**-Ff**12c/12c+f3,1/1,2 则表示

- 整张图的宽度和高度均为 12 厘米
- 第一列占 3 个宽度，第二列占 1 个宽度
- 第一行占 1 个高度，第二行占 2 个高度

-Fs*widths/heights* 通过指定每个子图的宽度和高度间接指定图片尺寸。

在这种情况下，整张图片的尺寸由每个子图的尺寸以及刻度线、标注、标签占据的空间和子图之间的间隙共同决定，但最外圈的刻度线、标记、标签所占据的空间不算在整张图的尺寸之内。

默认所有行和列的尺寸都是相同的。若想要给每行或每列指定不同的子图尺寸，可以加上一系列以逗号分隔的宽度，然后再加上一个斜杠，并加上一系列以逗号

分隔的高度。单个数则表示所有行或列有相同的宽度或高度。

例如，对于一个 2x2 的子图，使用 **-Fs**4c,8c/4c 则表示第一列为 4 厘米宽，第二列为 8 厘米宽，所有列的高度均为 4 厘米高。注意，宽度值或高度值的数目必须是一个或者与行数/列数相匹配。

对于地理地图而言，每张子图的高度由地图区域 **-R** 以及投影方式 **-J** 决定。有两个选择：(1) 指定子图高度为 0，并同时指定 **-R** 和 **-J**，利用其计算每张子图的高度，但要求所有子图必须共享相同的研究区域和投影方式；(2) 不断尝试并修改子图的高度以得到最佳的绘图布局。

13.59.3 可选选项

-A[*autolabel*][+**j**|**J***refpoint*][+**c***dx*/*dy*][+**g***fill*][+**p***pen*][+**o***dx*/*dy*][+**r**][+**R**][+**v**]
为子图自动添加编号。

autolabel 可以是单个数字或字母，也可以在数字或字母的一侧或两侧加上括号。其设置了左上角第一张子图的编号，而其余子图则按照递增的顺序依次编号。默认值为 **a**)。

注解： 括号在 Unix Shell 中有特殊含义，可以将其用单引号括起来，即 `'(a)'`。

加上子选项可以指定编号的更多属性：

- +**j**|**J***refpoint*: 指定编号在子图中的位置，默认值为 `TL`，即编号位于子图的左上角。+**j** 和 +**J** 分别适用于子图编号位于子图内部和外部的情况
- +**o***dx*[/*dy*] 设置子图编号相对于 +**j**|**J** 指定的参考位置间的额外偏移量，默认值为 FONT_TAG 的 20%
- +**p***pen* 为子图编号加上文本框
- +**g***fill* 为子图编号的文本框填充颜色
- +**c***dx*[/*dy*]: 设置子图编号与文本框轮廓间的距离，默认值为 FONT_TAG 的 15%
- +**r** 表示用小写罗马数字编号
- +**R** 表示用大小罗马数字编号
- +**v** 表示沿着垂直列方向依次增加编号，默认沿着水平行方向依次增加

-B*parameters* *(more …)* 设置底图边框和轴属性。该选项会应用到所有子图。

-C[*side*]/*clearance*[**u**] 设置子图区域内某个边的额外空白量。这些额外的空白量可以用于绘制比例尺、添加额外的文字等。

side 可以取 **e w s n** 分别代表东西南北四条边；也可以取 **x** 或 **y**，分别表示设置东西或南北方向两条边的空白；若不指定 *side*，则表示同时设置四条边上的空白量。

该选项可以重复多次，对不同边分别设置不同的间距。

subplot begin 中该选项对所有子图均有效，而 **subplot set** 中使用该选项则仅对当前子图有效。

注解： 在子图模式下不能使用 **-X** 和 **-Y**，可以使用 **-C** 作为替代。

-J*projection* *(more …)* 设置地图投影方式。

-M*margins* 相邻子图之间的额外空白

margins 可以有三种情况：

1. 取一个值，表示子图四个方向的空白 [默认值为*FONT_ANNOT_PRIMARY* 规定的字体大小]
2. 取两个值，表示子图的水平和垂直方向的空白，两个值之间用斜杠分隔
3. 取四个值，表示子图的左右下上四个方向的空白，四个值之间用斜杠分隔

-R*xmin/xmax/ymin/ymax*[+r][+u*unit*] *(more …)* 指定绘图区域。该选项会应用于所有子图

-SC[b|t][+l*label*][+t] 设置一列中的所有子图共用 X 轴

当一列所有子图共用 X 轴时，默认只有第一行子图的顶部（**t**）X 轴和最后一个子图的底部（**b**）X 轴有标注。

- **-SCb** 一列中只有最后一行子图的底部 X 轴有标注
- **-SCt** 一列中只有第一行子图的顶部 X 轴有标注
- **+l***label* 为共用的 X 轴添加标注
- **+t** 为每个子图的标题预留空间
- **+tc** 为第一行的所有子图的标题预留空间

-SR[l|r][+l*label*][+p][+t] 设置一行中所有子图共用 Y 轴

当一行所有子图共用 X 轴时，默认只有第一列子图的左边 Y 轴和最后一列子图的右边 Y 轴有标注。

- **-SRl** 一行中只有第一列子图的左边 Y 轴有标注
- **-SRr** 一行中只有最后一列子图的右边 Y 轴有标注
- **+l***label* 为共用的 Y 轴添加标注
- **+p** 设置所有标注与 Y 轴平行

-T*heading* 设置整张图的总标题，标题文字的属性由 FONT_HEADING 控制。

每张子图各自的标题可以用 **-B** 或 **-S** 选项控制。

-V[*level*] *(more …)* 设置 verbose 等级 [c]

-X[a|c|f|r][*xshift*[u]] *(more …)* 沿 X 方向移动绘图原点。

-Y[**a**|**c**|**f**|**r**][***yshift***[**u**]] *(more …)* 沿 Y 方向移动绘图原点。

-^ 或 - 显示简短的帮助信息，包括模块简介和基本语法信息（Windows 下只能使用 **-**）
-+ 或 + 显示帮助信息，包括模块简介、基本语法以及模块特有选项的说明
-? 或无参数 显示完整的帮助信息，包括模块简介、基本语法以及所有选项的说明
–**PAR=*value*** 临时修改 GMT 参数的值，可重复多次使用。参数列表见 gmt.conf。

13.59.4 subplot set

subplot set 通过指定子图的行列号或索引号以激活某个特定的子图，接下来的所有绘图命令将只在该子图内进行绘制。其与*-c 选项* 功能基本一致。

子图的行号、列号、索引号均从 0 开始算起。因而对于 MxN 的子图而言，行号的取值为 0 到 M-1，列号的取值为 0 到 N-1，索引号的取值则为 0 到 (M*N-1)。

若使用 **subplot set** 但未指定子图的行列号或索引号，则 GMT 会自动激活“下一个”子图。例如，对于一个 2 行 2 列的图而言，每次使用 **subplot set** 而不指定子图行和列，则按照行优先顺序依次激活子图 `0,0` → `0,1` → `1,0` → `1,1`。若 **subplot begin** 中使用了 **-A+v** 选项，则按照列优先顺序依次激活子图 `0,0` → `1,0` → `0,1` → `1,1`。

13.59.5 subplot set 语法

gmt subplot set [*row,col*|*index*] [**-A***fixedlabel*] [**-C***side*/*clearance*[**u**]] [**-V**[*level*]]

13.59.6 可选选项

row,col* | *index 指定要激活的子图的行列号或索引号。行列号、索引号均从 0 开始算起。

若不指定子图行列号或索引号，则自动激活“下一个”子图。
-A*fixedlabel* 设置当前子图的编号，忽略 **subplot begin** 中 **-A** 选项设置的自动编号。

这一选项可以用于临时修改单个子图的编号，但该选项只能修改编号字符串，其余属性（如位置、文本框）等均只能继承自 **subplot begin** 的 **-A** 选项。
-C*side*/*clearance*[*u*] 设置子图的某个边的额外空白量。这些额外的空白量可以用于绘制比例尺、添加额外的文字等。

side 可以取 **e w s n** 分别代表东西南北四条边。该选项可以重复多次，对不同边分别设置不同的间距。**subplot begin** 该选项对所有子图均有效，而 **subplot set** 中使用该选项则仅对当前子图有效。

注解: 在子图模式下不能使用 **-X** 和 **-Y**，可以使用 **-C** 作为替代。

13.59.7 subplot end

该命令用于结束当前的子图模式。

在结束子图模式时，其会进行如下操作：

- 对所有子图进行编号
- 将绘图原点重置回之前的原点位置
- 更新历史信息，设置为线性投影，并给出整张图的大小，使得用户可以使用 **-DJ** 方式放置 colorbar 等。

13.59.8 subplot end 语法

gmt subplot end [**-V**[*level*]]

13.59.9 示例

下面的示例展示了如何设置一张 2x2 布局的图，并使用不同的方式指定要激活的子图。不同的设置方式的效果是相同的，用户在使用时可根据需求选择最直观最简便的方式。

方法 1：使用 subplot set 指定子图行列号

注意行列号均从 0 开始。

```
gmt begin map png,pdf
    gmt subplot begin 2x2 -Fs5c/2.5c -A
    gmt subplot set 0,0
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set 0,1
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set 1,0
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set 1,1
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot end
gmt end
```

方法 2: 使用 subplot set 指定子图索引号

注意子图索引号从 0 开始:

```
gmt begin map png,pdf
    gmt subplot begin 2x2 -Fs5c/2.5c -A
    gmt subplot set 0
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set 1
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set 2
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set 3
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot end
gmt end
```

方法 3: 使用 subplot set 但不指定子图号

每次使用 **subplot set** 但不指定子图行列号或索引号, 则会自动激活 "下一个" 子图:

```
gmt begin map png,pdf
    gmt subplot begin 2x2 -Fs5c/2.5c -A
    gmt subplot set
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot set
    gmt basemap -R0/10/0/10 -Baf
    gmt subplot end
gmt end
```

方法 4: 使用 -c 选项

-c 选项 的功能与 **subplot set** 类似，可以用于激活指定的子图。其后可以接子图行列号或索引号，也可以只使用 **-c** 自动激活下一个子图:

```
gmt begin map png,pdf
    gmt subplot begin 2x2 -Fs5c/2.5c -A
    gmt basemap -R0/10/0/10 -Baf -c
    gmt basemap -R0/10/0/10 -Baf -c
    gmt basemap -R0/10/0/10 -Baf -c
    gmt basemap -R0/10/0/10 -Baf -c
    gmt subplot end
gmt end
```

下面展示了如何设置一个 2x2 的图，并设置共用 X 和 Y 轴:

```
gmt begin panels png,pdf
  gmt subplot begin 2x2 -Fs10c/5cc -M5p -A -SCb -SRl -Bwstr
    gmt subplot set
    gmt basemap -R0/80/0/10
    gmt subplot set
    gmt basemap
    gmt subplot set
    gmt basemap
    gmt subplot set
    gmt basemap
  gmt subplot end
gmt end
```

下面的示例展示了如何绘制一个不完全规则的子图。这个示例中，实际上只使用了子图 0、2、3，而第一张图同时占据了子图 0 和 1 的空间。在这种情况下，GMT 的自动编号功能无法正确编号，因而需要在 **subplot set** 中为每个子图单独指定编号。

```
gmt begin map pdf,png
    gmt subplot begin 2x2 -Fs5c/3c -A -M0

    gmt subplot set 0
    gmt basemap -R0/20/0/10 -JX11.2c/3c -Baf -BWSen

    gmt subplot set 2 -A'(b)'
    gmt basemap -R0/20/0/10 -JX? -Baf -BWSen

    gmt subplot set 3 -A'(c)'
    gmt basemap -R0/20/0/10 -JX? -Baf -BWSen

    gmt subplot end
gmt end
```

13.60 ternary

官方文档 ternary
简介 绘制三角图解

ternary 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，ternary 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。

13.60.1 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*plot* 中的 -S 选项

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.61 text

官方文档 text
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。

13.61.1 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 text -JX10c -R0/5/0/5 -Ba -pdf test
```

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。

13.61.2 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。

Source Code

-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 text 经常遇到的情况是在台站处标记台站名，此时传递给 text 的位置参数通常是台站坐标，因而 text 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<text>|+z<forma
控制文本的角度、对齐方式和字体等属性。

1. +f<font> 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 text -R0/10/0/10 -JX10c/10c -B1g1 -F+f30p,4,red+a45+jTL -pdf text << EOF
3 4 Text1
6 8 Text2
EOF
```

若使用了 +f 子选项，但是未给定 <font> ，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 text -R0/10/0/10 -JX10c/10c -B1 -F+cTL -Dj0.2c/0.2c -pdf text
```

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 text -R0/10/0/10 -JX10c/10c -B1 -F+h -pdf text << 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 text -R0/10/0/10 -JX10c/10c -B1 -F+l -pdf text << 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 *clip* 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 text -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 begin text_-M png,pdf
gmt text -R0/3/0/5 -JX3i -F+f+a+j -B1 -M -N << EOF
```

（下页继续）

(续上页)

```
> 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
gmt end
```

图 18: 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。

-U[*label*][+c][+j*just*][+o*dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.61.3 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 text -R0/10/0/5 -JX10c/5c -B1 -Wblue -Glightblue -TO -C100%/100% -pdf text << EOF
3   1   Text1
6   3   Text2
EOF
```

13.62 velo

官方文档 velo
简介 在地图上绘制三种符号：矢量（向量）、十字线和旋转饼图

13.62.1 -S 选项

-S 选项决定要绘制哪一种符号，及其输入数据的格式。

-Se<velscale>/<confidence>/<fontsize> 绘制矢量，并可选择用椭圆表示不确定度。输入数据中每列的含义为：

1. 起点的经度
2. 起点的纬度
3. 矢量的东向分量
4. 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. 名称（可选）

说明：

1. <velscale> 控制了矢量的缩放比例，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> 是名称的文本字号

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> 与 -Se 类似，区别在于输入文件不同：

1. 起点的经度
2. 起点的纬度
3. 矢量的东向分量
4. 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. 名称（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. 旋转饼图的旋转角度（弧度制）
4. 旋转饼图的不确定度（弧度制）

旋转饼图由两个同心扇形组成。一个扇形是旋转饼图本身（下称饼图扇形），另一个扇形则表示旋转饼图的不确定度（下称不确定度扇形）。两个扇形的半径由 <wedge_scale> 给出。两个扇形的角度分别为旋转饼图的旋转角度和不确定度乘以 <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。饼图扇形从正北开始，顺时针旋转。不确定度扇形从旋转饼图的结束开始，向顺时针和逆时针两个方向旋转，两个方向旋转的角度均为不确定度的 2 倍。

-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）

13.62.2 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

-U[*label*][+**c**][+**j***just*][+**o***dx*/*dy*] *(more …)* 在图上绘制 GMT 时间戳 logo

-t[*transp*] *(more …)* 设置图层的透明等级，可取值范围为 0（不透明）到 100（全透明）

13.62.3 示例

```
#!/bin/bash

gmt begin velo_ex1 png,pdf
# 左边三个符号
gmt velo -JM15c -R241/242/34/35 -B0.5 -BWeSn+tvelo \
        -Sw3c/1.e7 -W1p -G60 -E210 -D2 << EOF
# 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 velo -Sw3c/1.e7 -W1p -D2 << EOF
# 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 coast -W1p -Di
gmt end
```

图 19: velo 示例图 1

```
#!/bin/bash

gmt begin velo_ex2 png,pdf
gmt velo -R-10/10/-10/10 -Wthin,red -Se0.2c/0.39/12 -B1g1 \
        -BWeSn -Jx0.2i -Ggreen -Eblue -L -N -A1c+p3p+e << EOF
# 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
   -5.     5.    -5.0    0.0     4.0    6.0  0.500  4x6
```

(下页继续)

(续上页)

```
   -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 velo -Se0.2c/0.39/14 -A0.25c+p0.25p+e << EOF
# 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
gmt end
```

图 20: velo 示例图 2

下面的示例展示了如何绘制应变十字线:

```
gmt begin test pdf
gmt velo -JX10c/10c -R0/10/0/10 -Sx10 -W1p << EOF
5 5 0.5 0.3 45
EOF
gmt end
```

13.63 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` 。

13.63.1 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

13.63.2 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<re

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=xmax 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=ymax 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。

13.63.3 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
```

保护环境, 从阅读电子文档开始!

第 14 章　绘图示例

14.1 绘制震源球

meca 模块可以用于绘制震源球。

绘制震源球时需要向 GMT 提供震源的相关信息，包括：

1. 震源位置及深度
2. 震源机制解
3. 震级
4. 震源球在地图上的位置

其中，震源机制解可以用多种不同的方式描述。*meca* 模块用 `-S` 选项用于指定震源机制解的描述方式。支持的描述方式有五种：

1. `-Sa` Aki-Richards 描述
2. `-Sc` Harvard CMT
3. `-Sm|d|z` GCMT 零迹矩张量
4. `-Sp` 两个断层平面的部分参数
5. `-Sx|y|t` T、N、P 轴

具体的格式见*meca* 的说明文档。

14.1.1 震源球大小随震级变化

```
#!/bin/bash
gmt meca -JQ104/15c -R102.5/105.5/30.5/32.5 -Ba -Sa1c -png beachball_1 << EOF
# 经度 纬度 深度 (km) strike dip rake 震级 newX newY ID
104.33 31.91 39.8 32 64 85 7 0 0 A
104.11 31.52 27.1 22 53 57 6 0 0 B
EOF
```

本示例在震源位置处绘制了两个震源球。震源球的大小随震级变化。`-Sa1c` 的 `1c` 指定了 5 级地震的震源球大小为 1 厘米。其它地震震源球的大小根据震级按照如下公式进行缩放:

```
size = M / 5 * <scale>
```

例如, 本例中 7 级地震的大小为 1.4 厘米。

14.1.2 固定大小的震源球

使用选项 `-M` 使所有震源球的大小均为定值, 不随震级变化。

```
#!/bin/bash
gmt meca -JQ104/15c -R102.5/105.5/30.5/32.5 -Ba -Sa1c -M -png beachball_2 << EOF
# 经度 纬度 深度 (km) strike dip rake 震级 newX newY ID
104.33 31.91 39.8 32 64 85 7 0 0 A
104.11 31.52 27.1 22 53 57 6 0 0 B
EOF
```

14.1.3 震源球大小随震级变化，颜色随深度变化

使用 -G 和 -E 选项可以分别为震源球的压缩象限和拉伸象限指定不同的颜色。使用 -Z 选项则可以更进一步，根据某个具体的值动态指定压缩象限的颜色。

本例中展示如何用压缩象限的颜色表示震源的深度。需要先生成一个 CPT 文件，为每个深度段设置不同的颜色。然后，使用*meca* 模块的 -Z 选项即可。

```
#!/bin/bash
CPT=meca.cpt

# 生成 CPT 文件，为每个深度段设置不同的颜色
cat << EOF > $CPT
 0   0-1-1   20   0-1-1
20  60-1-1   40  60-1-1
40 120-1-1   60 120-1-1
60 240-1-1  100 240-1-1
EOF

gmt begin beachball_3 png,pdf
gmt basemap -JQ104/15c -R102.5/105.5/30.5/32.5 -Ba -BWSEN
gmt meca -C+s5p -Sa1.3c -Z$CPT << EOF
# 经度 纬度 深度 (km) strike dip rake 震级 newX newY ID
104.33 31.91 39.8  32 64   85 7.0      0     0 A
104.11 31.52 27.1  22 53   57 6.0      0     0 B
103.67 31.13  6.4  86 32  -65 8.0      0     0 C
103.90 31.34 43.6 194 84  179 4.9 104.18 30.84 D
103.72 31.44 67.3  73 84 -162 4.9 103.12 31.64 E
104.12 31.78 12.7 186 68  107 4.7 103.83 32.26 F
104.23 31.61 62.0  86 63  -51 4.7 104.96 31.69 G
EOF
```

(下页继续)

(续上页)

```
gmt colorbar -C$CPT -DjBL+w5c/0.5c+ml+o0.8c/0.4c -Bx+lDepth -By+lkm -L -S
gmt end
```

14.2 绘制直方图

histogram 模块可以用于统计并绘制直方图。

该模块可以绘制不同类型的直方图，具体选项为：

1. `-Z -Z0` 为默认情况下的缺省值，绘制统计直方图，`-Z1` 绘制百分比直方图
2. `-Q` 绘制累积直方图
3. `-A` 绘制水平直方图，默认情况绘制垂直直方图

以下示例使用的数据为 `eq.dat`。数据格式为三列，分别为：地震纬度、地震经度、震源深度。

14.2.1 垂直直方图

```
gmt histogram eq.dat -R0/35/0/600 -JX8c -Bxa5+l"Depth/km" -Bya100+l"Counts" -BWSne -D+f10p,4+o8p -
↪W5+b -L1p -i2 -png histVert
```

图 1: 垂直直方图

此示例中：

- `-R0/35/0/600` 设置了 X 轴范围是 0 到 35，Y 轴范围为 0 到 600
- `-JX8c` 指定了整张图为线性投影，图的宽度（X 轴长度）和图的高度（Y 轴长度）均为 8 厘米
- `-Bxa5+l"Depth/km" -Bya100+l"Counts"` 分别设置了 X、Y 轴标注、刻度的间隔、标签
- `-D+f10p,Times-Roman+o8p` 为每个 bar 设置标注，标注字号为 10p、字体为 4 号 Times-Roman，标注与 bar 之间距离为 8p
- `-W5+b` 设置直方图 bin 宽度为 5，`+b` 表示将落在范围外的数据包含在第一个或最后一个 bin 中
- `-L1p` 设置 bar 边框为 1p
- `-i2` 从输入文件 eq.dat 中读取第 3 列，`-i0` 表示输入文件第 1 列

14.2.2 水平直方图

`-A` 用于控制绘制水平直方图，上一个例子中默认绘制垂直直方图。

```
gmt histogram eq.dat -R0/35/0/600 -JX8c -Bxa5+l"Depth/km" -Bya100+l"Counts" -BWSne -D+f10p,4+o8p -A␣
↪-W5+b -L1p -i2 -png histHori
```

图 2: 水平直方图

14.2.3 累积直方图

-Q 表示绘制累积直方图。

```
gmt histogram eq.dat -R0/35/0/1000 -JX8c -Bxa5+l"Depth/km" -Bya200+l"Counts" -BWSne -W5+b -L1p -i2 -
↪Q -png histCumu
```

图 3: 累积直方图

14.2.4 百分比直方图

-Z1 设置纵轴为百分比，-N 绘制等效的正态分布曲线。

```
gmt histogram eq.dat -R0/35/0/100 -JX8c/8c -Bxa5+l"Depth/km" -Bya20+l"Frequency" -BWSne -W5+b -L1p -
↪i2 -N -Z1 -png histFreq
```

图 4: 百分比直方图

14.3 制作动画

GMT 提供了 movie 模块用于简化制作动图的流程。

动画本质上将一系列静态的图片（每张静态图片称为动画的一帧）按照顺序快速切换。movie 模块的作用就是生成一系列静态图片，并利用外部工具将所有图片合并转换成单个动图。

一张动图中有很多不变的元素，比如底图、海岸线、图例等。通常来说，一张动图可以分为三个图层：

1. 静态的背景图层（可选）
2. 动态元素图层（必须）
3. 静态的前景图层（可选）

用户需要准备一至三个 GMT 脚本，分别用于绘制三个图层。其中，绘制动态元素图层所需的脚本中可以使用一系列 GMT 预定义的变量。

GMT 生成动画的基本流程是：

1. 调用第一个脚本生成静态背景图层
2. 调用第三个脚本生成静态前景图层
3. 调用第二个脚本生成中间的动态元素所在的图层
4. 将静态背景图层、中间动态图层和静态前景图层按照顺序叠在一起，得到一帧静态图片

5. 循环执行第 3 和第 4 步，得到动图所需的所有帧静态图片
6. 调用外部工具将所有静态图片合并转换为动图图片

14.3.1 绘制正弦函数动画

```
#!/usr/bin/env bash

# 绘制静态背景图层的脚本
cat << EOF > pre.sh
gmt math -T0/360/20 T SIND = sin_point.txt
gmt math -T0/360/2 T SIND = sin_curve.txt
gmt begin
    gmt basemap -R0/360/-1.2/1.6 -JX3.5i/1.65i -X0.35i -Y0.25i \
    -BWSne+glightgreen -Bxa90g90f30+u'\232' -Bya0.5f0.1g1 --FONT_ANNOT_PRIMARY=9p
gmt end
EOF

# 绘制动态元素图层的脚本
cat << EOF > main.sh
gmt begin
#   Plot smooth blue curve and dark red dots at all angle steps so far
    last=\`gmt math -Q \${MOVIE_FRAME} 10 MUL =\`
    gmt convert sin_curve.txt -Z0:\${last} | gmt plot -W1p,blue -R0/360/-1.2/1.6 -JX3.5i/1.65i -X0.
↪35i -Y0.25i
    gmt convert sin_point.txt -Z0:\${MOVIE_FRAME} | gmt plot -Sc0.1i -Gdarkred
#   Plot bright red dot at current angle and annotate
    gmt plot -Sc0.1i -Gred <<< "\${MOVIE_COL0} \${MOVIE_COL1}"
    printf "0 1.6 a = %3.3d" \${MOVIE_COL0} | gmt text -F+f14p,Helvetica-Bold+jTL -N -Dj0.1i/0.05i
gmt end
EOF

# movie 模块调用了 pre.sh 和 main.sh 脚本以生成动画
# 生成动画
gmt movie main.sh -Sbpre.sh -C4ix2ix125 -Tsin_point.txt -Z -Nanim01 -D5 -A+l
# 将动画中的一帧保存为静态图片
gmt movie main.sh -Sbpre.sh -C4ix2ix125 -Tsin_point.txt -Z -Nanim01 -D5 -M10,png -Fnone
```

14.3.2 绘制旋转的地球

本例中绘制了一个旋转的地球动画，图上展示了地壳的的年龄。

```
#!/usr/bin/env bash

# 绘制静态背景图层的脚本并生成绘图过程中需要使用的数据
cat << EOF > pre.sh
gmt begin
        gmt math -T0/360/5 -I T 5 SUB = longitudes.txt
        gmt makecpt -Cdem2 -T0/6000 -H > t.cpt
        gmt grdgradient @earth_relief_20m -Nt1.25 -A45 -Gintens.nc
gmt end
EOF

# 生成绘图主脚本
cat << EOF > main.sh
gmt begin
        # Let HSV minimum value go to zero
        gmt set COLOR_HSV_MIN_V 0
        # Fake simulation of sun illumination from east combined with relief slopes
        gmt grdmath intens.nc X \${MOVIE_COL1} SUB DUP -180 LE 360 MUL ADD 90 DIV ERF ADD 0.25 SUB␣
↪= s.nc
        # Plot age grid first using age cpt
        gmt grdimage @age.3.20.nc -Is.nc -C@crustal_age.cpt -JG\${MOVIE_COL0}/0/6i -X0 -Y0
        # Clip to expose land areas only
        gmt coast -Gc
        # Overlay relief over land only using dem cpt
        gmt grdimage @earth_relief_20m -Is.nc -Ct.cpt
        # Undo clipping and overlay gridlines
        gmt coast -Q -B30g30
gmt end
EOF

# 制作动画
gmt movie main.sh -Sbpre.sh -C6ix6ix100 -Tlongitudes.txt -Nanim02 -H2 -Z -A+l
gmt movie main.sh -Sbpre.sh -C6ix6ix100 -Tlongitudes.txt -Nanim02 -H2 -Z -M10,png -Fnone

rm -rf main.sh pre.sh
```

14.4 中文图

注解: GMT 本身不支持中文，若需要在 GMT 上显示中文，需要进行额外的配置。具体方法见中文支持 一章。本文假定用户已经做好了 GMT 中文支持的配置工作。

注解: GMT 中文语言文件是 GMT 安装目录下的文件 `share/localization/gmt_cn1.locale`。该中文语言文件默认为 GB2312 编码方式。对于 Linux 和 macOS 用户，需要人工将其修改为 UTF8 编码才能正常显示中文的月份和星期。Windows 用户则不需要对其进行处理。

修改文件编码方式的方式有很多，请自行查找。我使用的是 enca的如下命令修改编码:

```
enca -L zh_CN -x UTF-8 gmt_cn1.locale
```

14.4.1 中文文字

本例中展示了如何在绘图时使用中文标签和中文标题，以及如何打印横排和竖排的中文。左图中 Y 轴标签与纵轴平行，中图中 Y 轴标签与 X 轴平行，右图中 Y 轴标签单独绘制并使用了竖排中文字体。右图中也可以使用*text* 模块手动添加 Y 轴标签。

```
gmt begin chinese-texts png
# 左图
gmt set FONT_TITLE 24p,41,black FONT_LABEL 16p,39,black
```

(下页继续)

(续上页)

```
gmt basemap -R0/10/0/5 -JX6c/4c -Bxaf+l'横轴' -Byaf+l'纵轴' -BWSen+t'中文标题'
gmt text -F+f << EOF
3 2.5 20p,39,black 横排文字
8 4.0 20p,43,red   竖排文字
EOF

# 中图
gmt set FONT_TITLE 24p,41,black FONT_LABEL 16p,39,black
gmt basemap -R0/10/0/5 -JX6c/4c -Bxaf+l'横轴' -Byaf+L'纵轴' -BWSen+t'中文标题' -X9c
gmt text -F+f << EOF
3 2.5 20p,39,black 横排文字
8 4.0 20p,43,red   竖排文字
EOF

# 右图
gmt set FONT_TITLE 24p,41,black FONT_LABEL 16p,39,black
gmt basemap -R0/10/0/5 -JX6c/4c -Bxaf+l'横轴' -BWSen+t'中文标题' -X9c
gmt basemap -Byaf+L'纵轴' -BW --FONT_LABEL=16p,43,black --MAP_LABEL_OFFSET=20p
gmt text -F+f << EOF
3 2.5 20p,39,black 横排文字
8 4.0 20p,43,red   竖排文字
EOF

gmt end
```

14.4.2 中文月份

GMT 支持中文的月份。要想使用中文表示月份，需要设置*GMT_LANGUAGE* 为中文，即 cn1，并设置标注的字体为中文。

```
gmt begin chinese-months png
# 设置 GMT 语言为中文
gmt set GMT_LANGUAGE cn1
# 设置标注的字体为中文字体
gmt set FONT_ANNOT_PRIMARY 16p,39,black

# FORMAT_TIME_PRIMARY_MAP 为 full 表示显示“一月”
gmt set FORMAT_DATE_MAP o FORMAT_TIME_PRIMARY_MAP full
gmt basemap -R1997T/1998T/0/10 -JX25c/4c -Bpxa1O -Byaf -BWSen

# FORMAT_TIME_PRIMARY_MAP 为 abbreviated 表示显示“一”
gmt set FORMAT_DATE_MAP o FORMAT_TIME_PRIMARY_MAP abbreviated
gmt basemap -R1997T/1998T/0/10 -JX25c/4c -Bpxa1O -Byaf -BWSen -Y-6c
gmt end
```

14.4.3 中文星期

GMT 支持中文的星期。要想使用中文表示星期几，需要设置*GMT_LANGUAGE* 为中文，即 cn1，并设置标注的字体为中文。

```
gmt begin chinese-weeks png
# 设置 GMT 语言为中文
gmt set GMT_LANGUAGE cn1
# 设置标注的字体为中文字体
gmt set FONT_ANNOT_PRIMARY 12p,39

# FORMAT_TIME_PRIMARY_MAP 为 full 表示显示“星期一”
gmt set FORMAT_DATE_MAP u FORMAT_TIME_PRIMARY_MAP full
gmt basemap -R1969-7-21T/1969-7-28T/0/4 -JX15c/3c -Bpxa1K -Byaf -BWSen

# FORMAT_TIME_PRIMARY_MAP 为 abbreviated 表示显示“周一”
gmt set FORMAT_DATE_MAP u FORMAT_TIME_PRIMARY_MAP abbreviated
gmt basemap -R1969-7-21T/1969-7-28T/0/4 -JX15c/3c -Bpxa1K -Byaf -BWSen -Y-5c
gmt end
```

保护环境, 从阅读电子文档开始!

第 15 章　API

GMT 提供了 API，使得用户可以直接在 C/Fortran、Matlab、Julia、Python 程序中直接调用 GMT。

15.1 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

15.2 GMT Matlab API

15.2.1 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.

15.2.2 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 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 平台下最新版本的 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…

15.2.3 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。

15.2.4 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。

15.3 GMT/Julia

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。

- 源码地址：https://github.com/GenericMappingTools/GMT.jl
- 官方文档：https://www.generic-mapping-tools.org/GMT.jl

注解: 目前 GMT.jl 正在开发中。

15.4 GMT/Python

从 GMT 6.0 开始，GMT 提供了 Python 接口，用户可以在 Python 中直接调用 GMT 绘图。

- 项目地址: https://github.com/GenericMappingTools/pygmt
- 官方文档: https://www.pygmt.org

注解: PyGMT 目前正在开发中。

保护环境, 从阅读电子文档开始!

第 16 章　附录

16.1 命令行及脚本基础

16.1.1 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，然后按下回车键即可：

```
$ 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 input -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 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

16.1.2 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.txo
```

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 plot -R0/10/0/10 -JX10c -Ba -Sa0.5c -Gred -pdf map
```

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 begin map pdf
gmt plot tmp1.txt -R0/10/0/10 -JX10c -Ba -Sa1c -Gred
gmt plot tmp2.txt -R -J -Sc0.5c -Gyellow
gmt end
rm tmp*
```

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
```

(下页继续)

(续上页)

```
    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
```

16.1.3 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 plot -R0/10/0/10 -JX10c -Ba -Sa0.5c -Gred -pdf map
```

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 begin map pdf,png
gmt psxy tmp1.txt -R0/10/0/10 -JX10c -Ba -Sa1c -Gred
gmt psxy tmp2.txt -R -J -Sc0.5c -Gyellow
gmt end
del tmp*
```

注释语句

rem 命令用来添加注释:

```
rem 绘制地图
gmt begin map pdf
echo 2 2 | gmt plot -R0/5/0/5 -JX5c -B1 -Sc0.5c -Gblack
```

（下页继续）

(续上页)

```
echo 3 3 | gmt plot -R -J -Sc0.5c -Gred
gmt end
```

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。

16.1.4 Unix 常用工具

awk

http://www.ruanyifeng.com/blog/2018/11/awk.html

16.2 现代模式 vs 经典模式

GMT 自 6.0.0 版本开始，引入了一种全新的绘图命令执行模式，称之为现代模式。而 GMT5 以及之前的命令风格则称之为经典模式。GMT6 既支持经典模式也支持现代模式，因而 GMT6 可以完全兼容 GMT5 的代码。本手册只介绍现代模式，而不介绍经典模式。对于 GMT 新用户而言，只需要了解现代模式；而对于 GMT 老用户，应了解现代模式与经典模式的区别，并尽快开始使用现代模式进行绘图。

这一节将主要介绍经典模式存在的问题，现代模式的优点，以及如何将经典模式的代码转换为现代模式。

16.2.1 经典模式的问题

一个 GMT 经典模式下的绘图脚本如下所示:

```
#!/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
gmt grdcut $D -R55/120/10/50 -GTibet.grd
gmt grdgradient Tibet.grd -A0 -Nt -Gint.grad

gmt psbasemap -R$R -J$J -B10g10 -BwSEN -K -P > $PS
gmt grdimage -R$R -J$J -Bg10 Tibet.grd -Iint.grad -CTibet.cpt -K -O >> $PS
gmt pscoast -R$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
gmt psscale -Dx-0.3i/1.4i+w2.5i/0.15i+ma -CTibet.cpt -G1600/9000 -By+lm -Bxa1000f200 -K -O >> $PS
gmt psscale -Dx-0.3i/0+w1.3i/0.15i+ma -CTibet.cpt -G-3500/0 -L -O >> $PS
gmt psconvert -A -P -Tf $PS
rm gmt.* Tibet.grd int.grad
```

从上面的示例脚本中以及实际使用经验中会发现 GMT 经典模式存在的一些明显的缺点或问题:

1. 大部分命令需要使用 -K 和 -O 选项, 而且极其容易出错
2. 需要使用 -P 选项决定画布的摆放方式
3. 每个绘图命令都需要使用重定向符号 > 或 >>
4. 每个绘图命令中都需要给定 PS 文件的名字
5. 大部分命令都需要重复使用 -J 和 -R 选项, 尽管具体数值可以省略
6. 需要调用单独的命令将生成的 PS 文件转换为 PDF、JPG、PNG 等格式

16.2.2 现代模式的优点

上面的示例在现代模式下可以改写为:

```
#!/bin/bash
D=@earth_relief_03m.grd

gmt grdcut $D -R55/120/10/50 -GTibet.grd
gmt grdgradient Tibet.grd -A0 -Nt -Gint.grad

gmt begin tutor3 pdf,png
gmt set FONT_ANNOT_PRIMARY 10p
gmt basemap -JB90/0/40/30/6.5i -R65/18/117/45+r -B10g10 -BwSEN
gmt grdimage -Bg10 Tibet.grd -Iint.grad -CTibet.cpt
gmt coast -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
gmt colorbar -Dx-0.3i/1.4i+w2.5i/0.15i+ma -CTibet.cpt -G1600/9000 -By+lm -Bxa1000f200
gmt colorbar -Dx-0.3i/0+w1.3i/0.15i+ma -CTibet.cpt -G-3500/0 -L
rm Tibet.grd int.grad
gmt end
```

现代模式下的 GMT 绘图命令更加简洁，具有如下优点：

1. 不再需要考虑 -K、-O 和 -P 选项
2. 不再需要使用 > 或 >> 重定向，也无需为每个命令都指定 PS 文件名
3. 用户完全不需要意识到 PS 文件的存在
4. 不再需要为每个命令都是用 -R 和 -J 选项
5. 最终可生成 PDF、PNG 等多种格式的图片文件，无需再转换
6. 自动确定纸张大小并进行裁剪
7. 整个脚本在独立的临时文件夹下执行，因而可以同时执行多个脚本而不互相干扰
8. 不再需要手动清理 gmt.conf 和 gmt.history 等临时文件

除此之外，现代模式还提供了更方便的子图模式和图中图模式：

- *subplot*: 子图模式
- *inset*: 图中图模式

16.2.3 从经典到现代

将一个经典模式的脚本改成现代模式下的脚本也很简单，基本遵循如下几点：

1. 绘图脚本以 gmt begin figname pdf 开头，以 gmt end 结束
2. 去掉 -K、-O、-P、重定向符号以及 PS 文件名
3. 去掉多余的 -R 和 -J 选项
4. 某些模块重新命名了。经典模式下以 ps 开头的模块省略了 ps，比如 pscoast => coast、psbasemap => basemap。几个例外包括 psxy => plot，psxyz => plot3d，psscale => colorbar。

16.3 GMT 底层原理

16.3.1 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标准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
```

(下页继续)

(续上页)

```
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标准选项参数，如果之后的命令显式指定了标准选项的参数，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！建议将常用的标准选项的参数定义成变量使用。

16.4 环境变量

16.4.1 $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 。

16.4.2 $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。多个目录之间用逗号分隔。

Linux 和 macOS 下以 / 结尾的目录会被递归搜索，而 Windows 暂不支持这一功能。

16.4.3 $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。

16.4.4 $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。

16.4.5 $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。

16.5 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。目录之间用逗号分隔。任何以斜杠 / 结尾的目录会被递归搜索（Windows 不支持此功能）。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

16.6 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 plot 和 plot3d 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。

16.6.1 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 plot 和 plot3d 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 (135ºW, 15ºS) 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。

16.6.2 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，却不指定单位时，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。

16.6.3 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 begin contour_annot1 pdf,png
```

(下页继续)

(续上页)

```
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+f8p -Gd1.5i -S10 -T+lLH
gmt end
```

图 1: 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 begin contour_annot2 pdf,png
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+f8p -Gn1/1i -S10 -T+lLH
gmt end
```

图 2: 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 begin contour_annot3 pdf,png
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+d+f8p -Gffix.txt/0.1i -S10 -T+lLH
gmt end
```

图 3: 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，将标注放置在直线段与等值线的交点。

```
#!/bin/bash
gmt begin contour_annot4 pdf,png
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+d+f8p -GLZ-/Z+ -S10 -T+lLH
gmt end
```

图 4: 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 begin contour_annot5 pdf,png
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+d+f8p -GXcross.txt -S10 -T+lLH
gmt end
```

图 5: 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

16.6.4 标注属性实例

本节通过实例说明标注属性的作用，采用 `plot` 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 begin contour_annot6 pdf,png
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+d+f8p -Gl50/10S/160/10S -S10 -T+l
gmt plot -SqD1000k:+g+LD+an+p -Wthick transect.txt
gmt end
```

图 6: 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 begin contour_annot7 pdf,png
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+d+u" m"+f8p -Gl50/10S/160/10S -S10 -T+l
gmt plot -SqD15d:+gblack+fwhite+LD+o+u\\260 -Wthick transect.txt
gmt end
```

图 7: 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 begin contour_annot8 pdf,png
gmt convert -i0,1,4 -Em150 transect.txt | gawk  '{print $1,$2,int($3)}' > fix2.txt
gmt coast -R50/160/-15/15 -JM5.3i -Gburlywood -Sazure -A500
gmt grdcontour geoid.nc -B20f10 -BWSne -C10 -A20+d+u" m"+f8p -Gl50/10S/160/10S -S10 -T+l
gmt plot -Sqffix2.txt:+g+an+p+Lf+u" m"+f8p -Wthick transect.txt
gmt end
```

图 8: 标注的位置和内容来自不同的数据集

16.6.5 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
gmt begin contour_annot9 pdf,png
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 grdgradient @earth_relief_05m -R-85/5/10/55 -Nt1 -A45 -Gtopo5_int.nc
gmt grdimage ttt_atl.nc -Itopo5_int.nc -Cttt.cpt -R-85/5/10/55 -JM5.3i -nc+t1
gmt grdcontour ttt_atl.nc -C0.5 -A1+u" hour"+v+f8p,Bookman-Demi -GL80W/31N/17W/26N,17W/28N/17W/50N -
↪S2
gmt plot -Wfatter,white great_NY_Canaries.txt
gmt coast -B20f5 -BWSne+t"Tsunami travel times from the Canaries" -N1/thick -Glightgray -Wfaint -
↪A500
gmt convert great_NY_*.txt -E | gmt plot -Sa0.15i -Gred -Wthin
gmt plot -Wthick great_NY_Canaries.txt -Sqn1:+f8p,Times-Italic+l"Distance Canaries to New York =
↪$km km"+ap+v

gmt plot -Wthinner great_NY_Paris.txt -SqD1000k:+an+o+gblue+LDk+f7p,Helvetica-Bold,white
```

(下页继续)

(续上页)

```
cat << EOF | gmt text -Gwhite -Wthin -Dj0.1i/0.1i -F+f8p,Bookman-Demi+j
74W        41N        RT        New York
2.33E        48.87N        CT        Paris
17W        28N        CT        Canaries
EOF
rm topo5_int.nc
gmt end
```

图 9: 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。

索引

B

basemap, 295
begin, 299

C

clear, 302
clip, 303
coast, 304
colorbar, 310
colors, 107
coupe, 312

D

docs, 318

E

end, 320

F

figure, 321
fill, 112
filter1d, 324
fitcircle, 325
fonts, 115

G

gmt, 328
gmt5syntax, 330
gmt.conf, 235
gmt-config, 329
gmtconnect, 330
gmtdefaults, 331
gmtget, 332
gmtinfo, 333
gmtlogo, 334
gmtselect, 335
gmtset, 339
gmtsimplify, 340
gmtwhich, 341
grd2xyz, 341
grdblend, 343
grdclip, 344
grdconvert, 345
grdcut, 346
grdedit, 347
grdimage, 349
grdinfo, 350
grdlandmask, 353
grdmask, 355
grdpaste, 357
grdproject, 357
grdsample, 358
grdtrack, 359
grdtrend, 359
grdvector, 360
grdvolume, 361

H

histogram, 363

I

image, 364
inset, 366

K

kml2gmt, 368

L

legend, 369

M

makecpt, 375
mapproject, 375
meca, 376

P

pen, 110

plot, 380
polar, 396
project, 400
psconvert, 403

R

rose, 406

S

sac, 408
sample1d, 411
solar, 412
spectrum1d, 414
subplot, 416

T

ternary, 424
text, 425
transparency, 199

V

velo, 430

X

xyz2grd, 434