

江苏高校优势学科建设工程资助项目 资助
江苏高校品牌专业建设工程资助项目

NCL
SHUJU CHULI YU HUITU
SHIXI JIAOCHENG

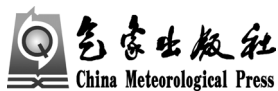
NCL数据处理与绘图 实习教程

-----© 施宁 于恩涛 汪君 孙晓娟 李忠贤 编著

江苏高校优势学科建设工程 资助
江苏高校品牌专业建设工程

NCL 数据处理与绘图 实习教程

施 宁 于恩涛 汪 君 孙晓娟 李忠贤 编著



内 容 简 介

本书首先介绍了 NCL 下载及安装方法,其后通过 9 个简单的绘图实例讲解了 NCL 的基本绘图方法,随后通过 19 个应用实例进一步综合讲解了 NCL 对于各种常见数据格式的输入输出、复杂图形的绘制方法以及常见计算函数的使用方法。此外,附录中还提供了 124 个常用计算函数及程序、53 个常用绘图函数及程序、25 个处理 WRF 输出数据的计算和绘图函数以及 13 类常用绘图参数的使用方法及说明,并提供了常用函数的索引,这些内容可方便读者日后参考查阅。

本书适合大气科学专业及其他地学专业本科及研究生实习使用,也可供相关专业研究生、科研业务人员参考使用。

图书在版编目(CIP)数据

NCL 数据处理与绘图实习教程 / 施宁等编著. -- 北京 : 气象出版社, 2017.6
ISBN 978-7-5029-6567-9

I. ①N… II. ①施… III. ①计算机制图-高等学校-教材 IV. ①TP391.72

中国版本图书馆 CIP 数据核字(2017)第 127487 号

NCL Shuju Chuli Yu Huitu Shixi Jiaocheng

NCL 数据处理与绘图实习教程

出版发行: 气象出版社

地 址: 北京市海淀区中关村南大街 46 号

邮政编码: 100081

电 话: 010-68407112(总编室) 010-68408042(发行部)

网 址: <http://www.qxcbs.com>

E-mail: qxcbs@cma.gov.cn

责任编辑: 黄红丽

终 审: 吴晓鹏

责任校对: 王丽梅

责任技编: 赵相宁

封面设计: 博雅思企划

印 刷: 北京中新伟业印刷有限公司

开 本: 720 mm×960 mm 1/16

印 张: 15.25

字 数: 305 千字

彩 插: 4

版 次: 2017 年 6 月第 1 版

印 次: 2017 年 6 月第 1 次印刷

定 价: 40.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等,请与本社发行部联系调换。

前 言

NCAR Command Language(NCL)是美国国家大气研究中心(NCAR)针对大气科学研究与气象业务需求而推出的免费的数据处理与绘图软件。该软件提供了大量的计算函数及子程序,充分满足了大气科学中各种数据格式输入输出及数值计算的需求,极大地降低了用户编写代码的时间成本;此外,NCL 还拥有强大的绘图能力,不仅可绘制大气科学中各种常见图形类型,且其图形效果美观。目前,NCL 已被大气科学界公认为一款功能强大的计算与绘图工具,正受到越来越多的高校研究生及相关从业人员的重视与喜爱。

本书主要依据 NCL 官网内容,同时结合作者近年来在气候学中的科研工作以及在南京信息工程大学中的 NCL 实践教学情况,针对大气科学科研与业务中几个具有代表性问题进行系统分析与讲解,使读者能够对 NCL 的使用方法有基本的理解与掌握。具体章节安排如下:第 1 章介绍了 NCL 的安装方法;第 2 章列出了各种图形的基本绘制方法;第 3 章至第 5 章通过几个具体问题进一步综合介绍了折线图、直方图、等值线图、矢量图等图形的绘制方法,同时涵盖了 ASCII、binary、NetCDF 数据的输入输出以及 EOF 分析、合成、回归、检验等计算方法;第 6 章介绍了其他几种较常见的图形绘制方法;第 7 章讲解了针对 WRF 模式输出结果的绘制方法;第 8 章介绍了结合 bash 编程(如调用 Fortran 程序)的使用方法。由于 NCL 版本不断升级完善,部分名称更加简洁、功能更加全面的新的函数或程序已实现部分旧的函数或程序的功能。对此,本书仅介绍新函数或程序。比如 ColorShadeLeGeContour、Shade-GeLeContour、ShadeGtContour、ShadeLtContour、ShadeLtGtContour 这五个旧的绘图函数所实现的功能可由一个新的绘图函数 gsn_contour_shade 实现,本书仅介绍新的绘图函数 gsn_contour_shade。

针对 NCL 的绘图步骤较为烦琐、初学者不易掌握的特点,本书侧重强调了绘制图形的基本方法。各章节示例脚本中的关键语句已用粗体标识。同时,本书还提供了删除这些关键语句的不完整脚本(可通过气象出版社 <http://www.qxcbs.com/ebook/nclsx/mdata.html> 下载)。读者可通过对照本书中的示例脚本,补充完善这些经删减的脚本,从而加深理解这些关键语句的含义与作用,进而较好地掌握相关图形的绘制方法。附录中列出了一些常用计算函数和绘图函数的使用方法、常用绘图参数(resources)的修改方法等,可供读者日后进一步拓展学习。此外,本书最后还提供了常用函数的索引,以方便读者学习及查阅。因此,本书不仅能满足科研院校学生

实习的需要,同时也可作为相关从业人员的 NCL 使用参考材料。

本书在编写出版过程中,得到了南京信息工程大学及其教务处、大气科学学院领导的大力支持,在此向他们表示诚挚的谢意。同时感谢江苏高校优势学科建设工程、江苏高校品牌专业建设工程(PPZY2015A016)、江苏省高等教育教改研究立项课题(2015JSJG032)和南京信息工程大学大气科学与环境气象实习教材建设项目(SXJC2016B101)为本书的撰写提供了经费支持。本书第 1 章由南京信息工程大学施宁老师与中国科学院大气物理研究所于恩涛老师共同撰写。此外,施宁老师还撰写了第 5、6、8 章及附录,于恩涛老师还撰写了第 7 章。中国科学院大气物理研究所汪君高级工程师撰写了第 2 章,南京信息工程大学李忠贤老师与孙晓娟老师分别撰写了第 3 章和第 4 章。南京信息工程大学大气科学学院王晓琼硕士研究生完成本书附录 D 中常用计算函数及附录 F 中 WRF 相关的计算函数及绘图函数的整理工作,余淘硕士研究生完成附录 E 中常用绘图函数的整理工作。特别感谢南京信息工程大学大气科学学院史湘军、潘玉洁两位老师对本书提出了许多宝贵修改意见。由于作者学识有限,时间仓促,谬误在所难免,敬请读者批评指正。

作 者

2017 年 3 月

目 录

前 言

第 1 章 NCL 安装与运行	(1)
1.1 实习目的	(1)
1.2 实习内容及要求	(1)
1.3 操作步骤示例及讲解	(1)
1.3.1 安装 Cygwin/X(Windows 系统)	(1)
1.3.2 下载 NCL 软件	(4)
1.3.3 解压解包 NCL	(5)
1.3.4 设置 NCARG_ROOT 及 DISPLAY 环境变量	(5)
第 2 章 绘图的基本方法	(7)
2.1 实习目的	(7)
2.2 实习内容及要求	(7)
2.3 脚本讲解及图形示例	(8)
2.3.1 标题及坐标轴标签	(8)
2.3.2 色板的使用	(9)
2.3.3 一根折线	(11)
2.3.4 多根折线及图例	(12)
2.3.5 等值线及坐标轴刻度	(13)
2.3.6 等值线及地图	(15)
2.3.7 矢量箭头	(17)
2.3.8 标识、多边形、任意折线及文本	(18)
2.3.9 组图	(21)
2.4 思考题	(23)
第 3 章 台站降水量随时间的演变	
——折线图	(24)
3.1 实习目的	(24)
3.2 实习内容及要求	(24)
3.3 脚本讲解及图形示例	(25)

3.3.1	台站降水资料的读取、处理及输出	(25)
3.3.2	四个特定台站的夏季降水(折线)	(26)
3.3.3	北京站逐年夏季降水量(直方图)	(29)
3.3.4	北京、上海两站在三年中的夏季降水量(直方图)	(31)
3.4	思考题	(34)
第 4 章 厄尔尼诺年的环流合成		
	——NetCDF 数据读取、计算和存储,等值线、矢量箭头、 图层叠加	(35)
4.1	实习目的	(35)
4.2	实习内容及要求	(35)
4.3	脚本讲解及图形示例	(36)
4.3.1	厄尔尼诺年各变量的合成图	(36)
4.4	思考题	(44)
第 5 章 北极涛动的统计特征分析		
	——EOF 分析、回归分析、相关分析、显著性检验	(45)
5.1	实习目的	(45)
5.2	实习内容及要求	(45)
5.3	脚本讲解及图形示例	(46)
5.3.1	海平面气压场对 EOF1 时间序列的回归	(46)
5.3.2	绘制 PC1 及其 9 年滑动平均	(51)
5.3.3	各层纬向平均气温场与 PC1 的相关系数	(52)
5.3.4	AO 指数与极涡指数的关系	(55)
5.4	思考题	(58)
第 6 章 特殊图形的绘制		
	——省界内绘图、卫星投影、兰勃特等角投影和小波图	(59)
6.1	实习目的	(59)
6.2	实习内容及要求	(59)
6.3	脚本讲解及图形示例	(60)
6.3.1	台站降水资料的读取并在江苏省内绘制	(60)
6.3.2	卫星投影图	(63)
6.3.3	兰勃特等角投影图	(65)
6.3.4	小波图	(70)
6.4	思考题	(74)

第 7 章 天气研究和预报模式输出数据制图	(75)
7.1 实习目的	(75)
7.2 实习内容及要求	(75)
7.3 脚本讲解	(75)
7.3.1 多层嵌套区域显示	(76)
7.3.2 地表数据制图	(79)
7.3.3 剖面图	(81)
7.3.4 风玫瑰图	(84)
7.3.5 组图	(85)
7.3.6 利用 gsn 系统函数制作 WRF 图	(89)
7.4 思考题	(92)
第 8 章 bash 脚本编程	(93)
8.1 实习目的	(93)
8.2 实习内容及要求	(93)
8.3 脚本讲解	(93)
8.3.1 利用 bash 脚本制作动画	(94)
8.3.2 调用 Fortran 90 脚本	(95)
8.4 思考题	(96)
参考文献	(97)
附录	(98)
附录 A 官网网址	(98)
附录 B 实习资料下载地址	(98)
附录 C 加载库函数文件	(98)
附录 D 常用计算函数	(99)
1. 函数 acos(value)	(99)
2. 函数 addfile(file_path, status)	(100)
3. 函数 addfiles(file_path, status)	(100)
4. 函数 all(logical_array)	(102)
5. 函数 any(logical_array)	(102)
6. 函数 asciiread(filepath, dimensions, datatype)	(102)
7. 函数 asciiwrite (filepath, var)	(103)
8. 函数 asin(value)	(103)
9. 函数 atan(value)	(103)

10. 函数 atan2(y, x)	(104)
11. 函数 avg(x)	(104)
12. 函数 cd_calendar(time, option)	(104)
13. 函数 ceil(value)	(105)
14. 函数 center_finite_diff_n(q, r, rCyclic, opt, dim)	(106)
15. 函数 conform(x, r, ndim)	(107)
16. 程序 copy_VarMeta(var_from, var_to)	(107)
17. 函数 cos(value)	(107)
18. 函数 day_of_year(year, month, day)	(108)
19. 函数 dim_avg_n_Wrap(x, dims)	(108)
20. 函数 dim_cumsum_n_Wrap(x, opt, dims)	(109)
21. 函数 dim_max_n(x, dims)	(109)
22. 函数 dim_min_n(x, dims)	(110)
23. 函数 dim_rmvmean_n_Wrap(x, dims)	(110)
24. 函数 dim_standardize_n_Wrap(x, opt, dims)	(110)
25. 函数 dim_stddev_n_Wrap(x, dims)	(110)
26. 函数 dim_sum_n_Wrap(x, dims)	(110)
27. 函数 dim_sum_wgt_n_Wrap(x, w, opt, dims)	(110)
28. 函数 dim_variance_n_Wrap(x, dims)	(110)
29. 函数 dimsizes(data)	(110)
30. 函数 dtrend_msg_n(x, y, remove_mean, return_info, dim) ...	(111)
31. 函数 dtrend_n(y, return_info, dim)	(112)
32. 函数 dv2uvF_Wrap(dv)	(112)
33. 函数 dv2uvG_Wrap(dv)	(112)
34. 函数 eofunc_n_Wrap(data, neval, optEOF, dim)	(112)
35. 函数 eofunc_north(eval, N, prinfo)	(113)
36. 函数 eofunc_ts_n_Wrap(data, evec, optETS, dim)	(114)
37. 函数 epflux(u, v, t, plvl, lat, opt)	(114)
38. 函数 escrc(x, y, mxlag)	(116)
39. 函数 escorc(x, y)	(116)
40. 函数 ezfftb(cf, xbar)	(117)
41. 函数 ezfftf(x)	(117)
42. 函数 f2fsh_Wrap(grid, outdims)	(118)
43. 函数 f2gsh_Wrap(grid, outdims, twave)	(118)

44. 函数 fbindirread(path, rec_num, rec_dims, rec_type)	(119)
45. 程序 fbindirwrite (path, var)	(119)
46. 程序 fbinnumrec(path)	(120)
47. 程序 fbinrecread(path, rec_num, rec_dims, rec_type)	(120)
48. 程序 fbinrecwrite(path, rec_num, var)	(121)
49. 函数 filwgts_lanczos(nwt, ihp, fca, fcb, nsigma)	(122)
50. 函数 floor(value)	(123)
51. 函数 fspan(start, finish, npts)	(123)
52. 函数 g2fsh_Wrap(grid, outdims)	(123)
53. 函数 g2gsh_Wrap(grid, outdims, twave)	(123)
54. 函数 gc_latlon (lat1, lon1, lat2, lon2, npts, iu)	(124)
55. 函数 get1Dindex (x, wanted_value)	(125)
56. 函数 get1Dindex_Exclude(x, exclude_value)	(125)
57. 程序 gradsf(z, gzx, gzy)	(126)
58. 程序 gradsg(z, gzx, gzy)	(126)
59. 函数 ind(larray)	(126)
60. 函数 ind_resolve(indices, dsizes)	(127)
61. 函数 int2p_n_Wrap(pin, xin, pout, linlog, pdim)	(127)
62. 函数 ismissing(data)	(128)
63. 函数 ispan(start, finish, stride)	(128)
64. 函数 linint2_Wrap(xi, yi, fi, fiCyclicx, xo, yo, foOption) ...	(129)
65. 函数 linmsg_n(x, opt, dim)	(130)
66. 函数 local_max(x, cyclic, delta)	(130)
67. 函数 local_max_1d(x, cyclic, delta, iopt)	(131)
68. 函数 local_min(x, cyclic, delta)	(131)
69. 函数 local_min_1d(x, cyclic, delta)	(131)
70. 函数 log(value)	(131)
71. 函数 log10(value)	(132)
72. 函数 max(value)	(132)
73. 函数 maxind(arg)	(132)
74. 函数 min(value)	(133)
75. 函数 minind(value)	(133)
76. 函数 mod(n, m)	(133)
77. 函数 month_to_season(xMon, season)	(133)

78. 函数 monthday(year, day)	(134)
79. 函数 ndtooned(val)	(134)
80. 函数 new(dimension_sizes, vartype, parameter)	(135)
81. 函数 num(val)	(135)
82. 函数 onedtond(val, dims)	(135)
83. 函数 pot_vort_isobaric (p, u, v, t, lat, gridType, opt)	(136)
84. 程序 print(data)	(137)
85. 程序 printVarSummary(data)	(137)
86. 函数 product(x)	(137)
87. 函数 reshape(val, dims)	(138)
88. 函数 readAsciiTable(filename, ncol, data_type, opt)	(138)
89. 函数 reg_multlin(y, x, option)	(139)
90. 函数 regCoef_n(x, y, dims_x, dims_y)	(139)
91. 函数 regline(x, y)	(140)
92. 函数 rtest(r, nr, opt)	(141)
93. 函数 runave_n_Wrap(x, nave, opt, dim)	(141)
94. 函数 short2flt(x)	(143)
95. 函数 sin(value)	(143)
96. 函数 smth9_Wrap(x, p, q, wrap)	(143)
97. 函数 sqrt(value)	(144)
98. 函数 stddev(x)	(144)
99. 函数 student_t(t, df)	(144)
100. 函数 sum(x)	(144)
101. 函数 svdcov(x, y, nsvd, homlft, hetlft, homrgt, hetrgt) ...	(145)
102. 函数 svdcov_sv(x, y, nsvd, svLeft, svRight)	(146)
103. 函数 svdstd(x, y, nsvd, homlft, hetlft, homrgt, hetrgt) ...	(147)
104. 函数 svdstd_sv(x, y, nsvd, svLeft, svRight)	(147)
105. 程序 system(command)	(147)
106. 函数 systemfunc(command)	(147)
107. 函数 tan(value)	(148)
108. 函数 ttest(ave1, var1, s1, ave2, var2, s2, iflag, tval_opt)	(148)
109. 函数 uv2dv_cfd(u, v, lat, lon, boundOpt)	(149)
110. 函数 uv2vrdivF (u, v)	(150)
111. 函数 uv2vrdivG (u, v)	(150)

112. 函数 uv2vr_cfd (u, v)	(150)
113. 函数 uv2sfvpF(u, v)	(150)
114. 函数 uv2sfvpG(u, v)	(151)
115. 函数 vibeta(p, x, linlog, psfc, pbot, ptop)	(151)
116. 函数 vint2p(datai, hbcofa, hbcofb, plevo, psfc, intyp, p0, ii, kxtrp)	(152)
117. 函数 vr2uvF_Wrap(u, v)	(153)
118. 函数 vr2uvG_Wrap(u, v)	(153)
119. 函数 wavelet(y, mother, dt, param, s0, dj, jt看, npad, noise, isigtest, sigigtest, nadof)	(153)
120. 函数 wgt_areaave(q, wgt, wgt, opt)	(154)
121. 函数 wgt_runave_n_Wrap(x, wgt, opt, dim)	(156)
122. 函数 wgt_vert_avg_beta(p, datai, psfc, punits, opt)	(157)
123. 函数 where(condtnl_expr, true_value, false_value)	(158)
124. 程序 write_matrix(data, fmtf, option)	(159)
附录 E 常用绘图函数	(160)
1. 函数 ColorNegDashZeroPosContour (plot, ncolor, zcolor, pcolor)	(160)
2. 函数 gsn_add_polygon(wks, plot , x, y, res)	(160)
3. 函数 gsn_add_polyline(wks, plot , x, y, res)	(161)
4. 函数 gsn_add_polymarker(wks, plot , x, y, res)	(161)
5. 函数 gsn_add_shapefile_polygons(wks, plot , shp_name, res)	(162)
6. 函数 gsn_add_shapefile_polylines(wks, plot , shp_name, res)	(162)
7. 函数 gsn_add_shapefile_polymarkers(wks, plot , shp_name, res)	(162)
8. 函数 gsn_add_text(wks, plot, text, x, y, res)	(162)
9. 函数 gsn_attach_plots(base_plot, plots, res_base, res_plots)	(162)
10. 函数 gsn_contour_shade(plot, lowval, highval, opt)	(163)
11. 函数 gsn_csm_attach_zonal_means(wks, map, data, res)	(163)
12. 函数 gsn_csm_blank_plot(wks, res)	(163)
13. 函数 gsn_csm_contour(wks, data, res)	(164)
14. 函数 gsn_csm_contour_map(wks, data, res)	(164)
15. 函数 gsn_csm_contour_map_ce(wks, data, res)	(164)
16. 函数 gsn_csm_contour_map_overlay(wks, data1, data2, res1, res2)	(165)
17. 函数 gsn_csm_contour_map_polar(wks, data, res)	(165)
18. 函数 gsn_csm_hov(wks, data, res)	(165)

19. 函数 gsn_csm_lat_time(wks, data, res)	(166)
20. 函数 gsn_csm_map(wks, res)	(166)
21. 函数 gsn_csm_map_ce(wks, res)	(166)
22. 函数 gsn_csm_map_polar(wks, res)	(166)
23. 函数 gsn_csm_pres_hgt(wks, data, res)	(167)
24. 函数 gsn_csm_pres_hgt_streamline(wks, data, xcomp, zcomp, res)	(167)
25. 函数 gsn_csm_pres_hgt_vector(wks, data, xcomp, zcomp, res)	(167)
26. 函数 gsn_csm_streamline(wks, u, v, res)	(167)
27. 函数 gsn_csm_streamline_contour_map(wks, u, v, data, res)	(168)
28. 函数 gsn_csm_streamline_contour_map_ce(wks, u, v, data, res) ...	(168)
29. 函数 gsn_csm_streamline_contour_map_polar(wks, u, v, data, res)	(168)
30. 函数 gsn_csm_streamline_map(wks, u, v, res)	(169)
31. 函数 gsn_csm_streamline_map_ce(wks, u, v, res)	(169)
32. 函数 gsn_csm_vector_scalar_map_polar(wks, u, v, data, res)	(169)
33. 函数 gsn_csm_x2y(wks, x1, x2, y, res1, res2)	(169)
34. 函数 gsn_csm_x2y2(wks, x1, x2, y1, y2, res1, res2)	(170)
35. 函数 gsn_csm_xy(wks, x, y, res)	(170)
36. 函数 gsn_csm_xy2(wks, x, y1, y2, res1, res2)	(170)
37. 函数 gsn_csm_xy3(wks, x, yL, yR, yR2, resL, resR, resR2) ...	(171)
38. 函数 gsn_csm_y(wks, y, res)	(171)
39. 程序 gsn_define_colormap(wks, color_map)	(171)
40. 程序 gsn_draw_colormap(wks)	(172)
41. 程序 gsn_labelbar_ndc(wks, nboxes, labels, x, y, res)	(172)
42. 程序 gsn_legend_ndc(wks, nitems, labels, x, y, res)	(172)
43. 程序 gsn_merge_colormaps(wks, color_map1, color_map2) ...	(173)
44. 函数 gsn_open_wks(type, name)	(173)
45. 程序 gsn_panel(wks, plot, dims, res)	(173)
46. 程序 gsn_polygon_ndc(wks, x, y, res)	(173)
47. 程序 gsn_polyline_ndc(wks, x, y, res)	(174)
48. 程序 gsn_polymarker_ndc(wks, x, y, res)	(174)
49. 程序 gsn_text_ndc(wks, text, x, y, res)	(174)
50. 程序 overlay(base_id, transform_id)	(174)

51. 函数 read_colormap_file(filename)	(174)
52. 函数 ShadeCOI(wks, plot, w, time, res)	(175)
53. 函数 WindRoseColor(wks, wspd, wdir, numPetals, circFr, spdBounds, colorBounds, res)	(175)
附录 F WRF 相关的计算函数、绘图函数及绘图参数	(176)
1. 函数 wrf_contour(nc_file, wks, data, res)	(176)
2. 函数 wrf_interp_1d(v_in, z_in, z_out)	(177)
3. 函数 wrf_interp_2d_xy(v3d, xy)	(178)
4. 函数 wrf_interp_3d_z(v3d, vert, loc)	(179)
5. 函数 wrf_ll_to_ij(lon, lat, opt)	(180)
6. 函数 wrf_map(nc_file, wks, res)	(181)
7. 函数 wrf_map_overlays(nc_file, wks, (/graphics/), pltres, mpres)	(181)
8. 函数 wrf_overlays(nc_file, wks, (/graphics/), pltres, mpres)	(182)
9. 程序 wrf_smooth_2d(fld, iter)	(182)
10. 函数 wrf_time_c(Times, opt)	(182)
11. 函数 wrf_user_getvar(nc_file, fld, it)	(183)
12. 函数 wrf_user_intrp2d(var2d, loc_param, angle, res)	(185)
13. 函数 wrf_user_intrp3d (var3d, H, plot_type, loc_param, angle, res)	(186)
14. 函数 wrf_user_list_times(file_handle)	(186)
15. 函数 wrf_user_ll_to_ij(nc_file, lons, lats, res)	(187)
16. 函数 wrf_user_ij_to_ll (nc_file, i, j, res)	(187)
17. 函数 wrf_user_vert_interp(file_handle, field, vert_coordinate, interp_levels, opts)	(188)
18. 函数 wrf_vector(nc_file, wks, data_u, data_v, res)	(190)
19. 函数 wrf_wps_close_int(istatus)	(190)
20. 函数 wrf_wps_dom (wks, mpres, lnres, txres)	(190)
21. 函数 wrf_wps_open_int(filename)	(191)
22. 函数 wrf_wps_rddata_int(istatus, nx, ny)	(191)
23. 程序 wrf_wps_rdhead_int(istatus, rhead, field, date, units, map_source, description)	(191)
24. 函数 wrf_wps_read_int (filename)	(191)

25. 程序 wrf_wps_write_int(filename, field, units, description, data, opt)	(192)
附录 G 常用绘图参数	(192)
1. 等值线 contour(cn)	(193)
2. 标识、多边形、任意折线 (gs)	(195)
3. gsn 高级接口 (gsn)	(196)
4. 地图 map(mp)	(199)
5. 色标 labelbar(lb)	(202)
6. 图例 legend(lg)	(202)
7. 图题 title(ti)	(203)
8. 坐标(tm 及 tr)	(205)
9. 字符 text(tx)	(206)
10. 箭头 vector(vc)	(207)
11. 视图 viewport(vp)	(209)
12. 折线(xy)	(209)
13. 流线 streamline (st)	(210)
附录 H 常用绘图参数图示	(211)
附录 I 常用色板	(213)
附录 J 常见错误提醒	(217)
函数索引	(222)
地球科学	(222)
应用数学	(223)
绘图	(224)
输入输出	(226)
统计分析	(226)
数组操作及查询	(227)
其他	(228)

实验实习资料下载说明

本书所用实习资料可以从气象出版社网站下载。网址
为: <http://www.qxcbs.com/ebook/nclsx/mdata.html>。

第 1 章 NCL 安装与运行

1.1 实习目的

学会在 Windows、Linux 和 Mac OSX 三个操作系统中任选一个操作系统中安装 NCL。

1.2 实习内容及要求

(1)对于 Windows 操作系统用户

①下载并安装 Cygwin/X。请安装至“d:/”下。

②下载对应版本 NCL 压缩包。将该压缩包放至“d:/cygwin/app/ncl”，解压解包。

③设置 NCARG_ROOT 及 DISPLAY 环境变量。

(2)对于 Linux 操作系统用户

①下载对应版本 NCL 压缩包。将该压缩包放至“/app/ncl”，解压解包。

②设置 NCARG_ROOT 及 DISPLAY 环境变量。

(3)对于 Mac OSX 操作系统用户

要求与 Linux 操作系统用户一致,但将 NCL 压缩包放至“/opt/ncl”,解压解包。

注意,无论在何种操作系统中,若上述路径不存在,可自行创建。本书将以上述路径为例进行讲解。另外,用户也可任意指定路径,只需修改相应参数,本书不再赘述。

1.3 操作步骤示例及讲解

1.3.1 安装 Cygwin/X(Windows 系统)

在 Windows 操作系统下,首先需下载安装 Linux 模拟器 Cygwin/X。下载地址为 <http://x.cygwin.com>。NCL 官网提供了详细的 Cygwin/X 下载及安装方法(<http://www.ncl.ucar.edu/Download/cygwin.shtml>)。注意,若操作系统为 Linux 或 Mac OSX,则不需下载 Cygwin/X,读者可直接跳至 1.3.2 节。

除 NCL 官网提供的方法外,本书将提供一种更为简便的 Cygwin/X 下载与安装

方法,只需下载一个压缩文件“cygwin. rar”,存放地址见附录 B“实习资料下载地址”^①。经测试,该方法在 Windows XP、Win7、Win8 及 Win10 系统下均适用。下载后放至“d:\”下,通过 WinRAR 或其他压缩软件解压至当前路径。双击快捷方式文件“Cygwin Terminal”(图 1.1 中实线框)即可启动 Cygwin。启动后,系统会在“d:\cygwin\home”路径下自动生成一个与 Windows 操作系统登录名同名的文件夹(图 1.2 中实线框)。本例中,系统生成了一个名为“X1”的文件夹。



图 1.1 Cygwin 文件夹及其路径



图 1.2 自动生成的文件夹及其文件

若要启动 X 窗口,双击“d:\cygwin\bin”下的“Xwin. exe”即可(图 1.3 中实线框)。注意,快捷方式文件“Cygwin Terminal”与“Xwin. exe”位于不同路径下,已在图 1.1 及图 1.3 中用虚线框标示。

^① 本书所用实习资料可以从气象出版社网站下载。网址为: <http://www.qxchs.com/ebook/nclsx/mdata.html>。

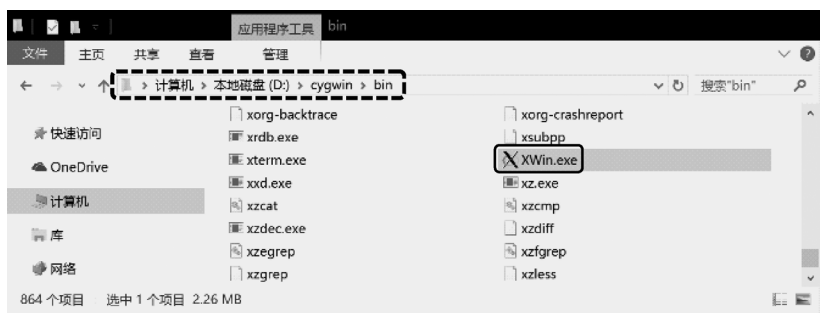


图 1.3 X 窗口及其路径

若用户需将 Cygwin 安装在其他路径, 比如“c:\program”, 则首先将“cygwin.rar”解压在该路径下, 再鼠标右击“Cygwin Terminal”(图 1.1 中实线框), 将其中“快捷方式”下“目标(T):”中的 mintty.exe 路径改为用户所解压缩的路径“C:\program\cygwin\bin\mintty.exe”(图 1.4 中实线框)。保存后再双击“Cygwin Terminal”以启动 cygwin。注意, 不要删除路径后的参数设置。

在有些 Windows 操作系统上可能出现双击“Cygwin Terminal”后会弹出 3 个窗口。这一问题的解决方法为, 鼠标右击“Cygwin Terminal”文件, 在“兼容性”下勾选“以兼容模式运行这个程序”, 在下拉菜单中勾选“Windows 7”(图 1.5 实线框)。



图 1.4 Cygwin Terminal 的路径设置

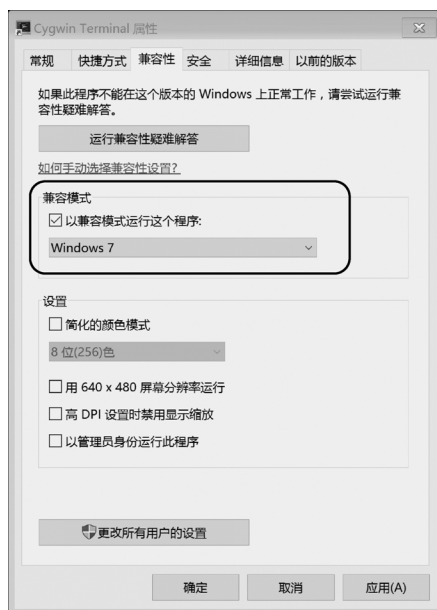


图 1.5 Cygwin Terminal 属性修改

1.3.2 下载 NCL 软件

下载地址为 <https://www.earthsystemgrid.org/dataset/ncl.html>。首先选择 NCL 版本号最高的版本。同一版本号 NCL 又细分为三类,分别为软件的源代码、支持 OPeNDAP(可访问远程数据)预编译的可执行文件(precompiled binaries)和不支持 OPeNDAP 的预编译可执行文件。可见,后两类均为预编译可执行文件,区别在于是否支持 OPeNDAP 功能。注意,对于 Cygwin/X(Windows)系统,该网址只提供支持 OPeNDAP 功能的 NCL 版本。对于源代码类,用户可自行编译 NCL 可执行文件,但用户需具备编译器、相关软件库及库函数编译等的相关知识。因此,建议初级用户下载预编译可执行文件。选好后,点击“Download options”可发现,该网址依据编译 NCL 时所使用的操作系统及编译器提供了更多的细分版本(图 1.6)。

NCL Version 6.4.0 precompiled binaries (OPeNDAP-enabled)

Download Options For Selection

11 files

<input type="checkbox"/>	File	Size	Format	Location
<input type="checkbox"/>	ncl_ncarg-6.4.0-CentOS6.8_64bit_gnu447.tar.gz	100.47 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-CentOS7.3_64bit_gnu485.tar.gz	99.33 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-CYGWIN-NT-10.0-WOW_i686.tar.gz	92.89 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-Debian7.11_32bit_gnu472.tar.gz	98.8 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-Debian7.11_64bit_gnu472.tar.gz	101.08 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-Debian8.6_64bit_gnu492.tar.gz	100.62 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-MacOS_10.11_64bit_gnu610.tar.gz	84.57 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-MacOS_10.12_64bit_gnu530.tar.gz	83.65 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-RHEL6.4_64bit_gnu447.tar.gz	85.23 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-RHEL6.4_64bit_intel1215.tar.gz	105.28 MB		DISK
<input type="checkbox"/>	ncl_ncarg-6.4.0-SUSE12.1_64bit_intel1603.tar.gz	103.38 MB		DISK

图 1.6 可供下载的不同版本的 NCL

若在 Cygwin/X(Windows)系统下安装使用 NCL,请下载形如“ncl * CYGWIN *.tar.gz”文件(图 1.6 实线框)。

若在 Linux 系统下,则需首先查看系统类型、位数及 gcc 版本。请在终端中输入如下两行命令进行查询:

```
cat /etc/issue
gcc --version
```

如返回值为“Debian”和“4.7.2”,同时系统为64位,则选择形如“ncl_ncarg-6.4.0-Debian8.6_64bit_gnu492.tar.gz”(图1.6虚线框)文件。注意,如果可供下载NCL的gcc版本与Linux系统所用gcc版本并不完全对应,则下载最为接近gcc版本。此外,如果用户的Linux系统是“CentOS”或者“Fedora”,则选择下载“Red-Hat”版本,若是“Ubuntu”系统则选择“Debian”版本。

若在Mac OSX系统下,下载与本机OSX系统对应的版本,OSX版本可以从系统菜单“关于本机”查看,gcc版本查看方式与Linux相同。如果Mac OSX是10.12版本,gcc为5.3.0,则选择形如“ncl_ncarg-6.4.0-MacOS_10.12_64bit_gnu530.tar.gz”(图1.6点线框)文件。若本机gcc版本过低,则建议升级本机gcc版本。

1.3.3 解压解包 NCL

将NCL打包压缩文件“ncl_ncarg-6.4.0-CYGWIN_NT-10.0-WOW_i686.tar.gz”拷贝至要安装的路径,即Windows下的“d:\cygwin\app\ncl”,Linux下的“/app/ncl”以及Mac OSX下“/opt/ncl”。随后在Cygwin(Linux, Mac OSX)终端中进入对应的路径,并输入“tar”命令进行解压缩(图1.7)。解压后在当前目录下会生成bin、lib、include三个文件夹,其中bin文件夹中包含了所有可执行文件,lib文件夹包含了库函数,include文件夹中包含了头文件。注意,Cygwin终端中使用tar命令解压缩与1.3.1节中介绍的利用软件WinRAR解压“cygwin.rar”的方式不同,不要混淆。尤其是不要使用软件WinRAR解压NCL软件压缩包,否则可能导致NCL不能正常运行。

```
X1@X1 /app/ncl
$ tar -zxvf ncl_ncarg-6.4.0-CYGWIN_NT-10.0-WOW_i686.tar.gz
```

图 1.7 终端中解压解包 NCL 打包压缩文件

1.3.4 设置 NCARG_ROOT 及 DISPLAY 环境变量

环境变量的设置在不同的Shell(bash,csh,ksh)中略有差异,需要根据本机使用的Shell进行具体设置,本书主要以bash为例进行说明。

用vi或其他文本编辑器打开\$HOME路径下的“.bashrc”文件(图1.2虚线框)。在对应系统的终端中输入命令“vi ~/.bashrc”。随后在该文件中添加如下三行后保存:

```
export NCARG_ROOT=/app/ncl
export PATH=$NCARG_ROOT/bin:$PATH
export DISPLAY=:0.0 (Linux 和 Mac OSX 系统中可忽略本行)
```


在设置环境变量时需注意以下两点,一是在设置 NCARG_ROOT 环境变量时,其值需对应 NCL 的安装路径。假如用户未按本书说明,将 NCL 解压解包文件存放在“/home/user/bin/ncl6”下,则需在“.bashrc”文件中设置“export NCARG_ROOT=/home/user/bin/ncl6”。二是环境变量 NCARG_ROOT 中间的“_”不可缺少,同时该环境变量及 DISPLAY 环境变量全部大写,不可小写。

为使上述修改生效,需在终端输入“source ~/.bashrc”。对于 Linux 和 Mac OSX 系统,此时安装完毕。在终端中输入“ncl”后回车,即可启动 NCL。对于 Windows 系统,在 cygwin 终端中运行命令“source ~/.bashrc”后,屏幕上可能会出现“-bash: \$'\r': command not found”的错误提醒。这需要继续在 cygwin 终端输入命令“dos2unix ~/.bashrc”以对文件 ~/.bashrc 进行格式转换,完毕后再重新输入命令“source ~/.bashrc”即可。其命令输入过程见图 1.8。

```
X1@X1 ~
$ dos2unix ~/.bashrc
dos2unix: 正在转换文件 /home/X1/.bashrc 为Unix格式...

X1@X1 ~
$ source ~/.bashrc

X1@X1 ~
$ ncl
Copyright (C) 1995-2017 - All Rights Reserved
University Corporation for Atmospheric Research
NCAR Command Language Version 6.4.0
The use of this software is governed by a License Agreement.
See http://www.ncl.ucar.edu/ for more details.
ncl 0> █
```

图 1.8 启动 NCL

第 2 章 绘图的基本方法

2.1 实习目的

- (1)掌握添加标题(title)及坐标轴名称的基本方法。
- (2)掌握使用色板的基本方法。
- (3)掌握绘制折线(XY line)的基本方法。
- (4)掌握绘制等值线(contour)的基本方法。
- (5)掌握绘制矢量箭头(vector)的基本方法。
- (6)掌握在地图上添加标识(marker)、任意折线(polyline)、多边形(polygon)及文本(text)的基本方法。
- (7)掌握绘制组图(panel)的基本方法。

2.2 实习内容及要求

首先下载附录 B“实习资料下载地址”中所有以“ex”为开始的 ncl 脚本,再根据以下实习内容与要求,补充完善相应脚本:

(1)添加修改标题及坐标轴标签。修改 ex_title.ncl,添加主图题“This is a main title”及 X 和 Y 轴的名称“X axis”及“Y axis”,并修改其颜色和大小;添加左、中、右标题。

(2)色板的使用。将 ex_color.ncl 中的色板改为“hotcold_18lev”;图中大于 5 的正值用红色表示,小于-5 的负值用蓝色表示,(-5,5)范围内数值用白色表示。

(3)绘制折线。修改 ex_xy.ncl,线条颜色为“black”,4.5 倍粗。用点线的形式表示线条;标识选用实心圆点,其颜色改为红色。

(4)绘制折线及图例(legend)。修改 ex_xy_legend.ncl,添加一个 X 轴数组,它由 500 个数值组成,其数值范围为 $[-100,100]$ 。将绘图函数 gsn_csm_y 改为 gsn_csm_xy。依据 y 数组中的数值绘制四根折线,每根折线 500 个数值。将四根折线分别设置为不同的线型、粗细及颜色。添加线条图例,并改变图例上默认的标签说明(label),改变图例的大小及位置,使其位于图形右上方。

(5)绘制等值线及坐标轴刻度。修改 ex_contour_tm.ncl,将等值线从-50 开始

画起,直至 60,间隔为 10;2 倍粗,填色;色标(labelbar)以“vertical”方式排放;正、负等值线分别用黑色实线、虚线表示,略去 0 线。Y 轴主坐标的间隔为 5,对于 X 轴上数值为 5、15、25、35 的位置用坐标轴标签“Jan”、“Feb”、“Mar”、“Apr”表示。

(6)绘制修改等值线及地图。修改 `ex_cnmap.ncl`,在 $[30\sim 130^{\circ}\text{E}, 10\sim 80^{\circ}\text{N}]$ 的地理范围内绘制大于 1500 m 的地表高度,等值线间隔为 1000,填色,其中小于 1500 m 的等值线用透明色。经纬度线用虚线表示,0.5 粗细。

(7)绘制矢量箭头。修改 `ex_vector.ncl`,将箭头的类型设定为“FillArrow”,箭头头部大小一致,设置箭头的边缘颜色为白色,内部填充颜色为黑色,略去小于 5 的矢量箭头,箭头之间的最小距离为 0.02,将参考箭头移至地图右上角。

(8)添加标识、多边形、任意折线和文本。修改 `poly_ex.ncl`,标记南京位置,并添加字符“Nanjing”;绘制台风的移动路径,标记每天的位置;用填色表示一个关键区域。

(9)绘制组图。修改 `ex_panel.ncl`,利用四组随机数绘制四个图形,使其按 2×2 排列;将 4 副图中等值线数值范围设为一致;关闭每幅图各自的色标,采用公用色标;添加一个总标题;每幅图上分别添加 ABCD。

2.3 脚本讲解及图形示例

此节给出 2.2 节中所有的完整脚本,其中粗体为关键语句。网上提供的原脚本中已删除这些关键语句,读者可自行参考对照。

2.3.1 标题及坐标轴标签

```
ex_title.ncl(图 2.1):
begin
  npts = 500
  y = 500. + .9 * ispan(0,npts-1,1) * sin(0.031415926535898 * ispan(0,npts-1,1)) ;创建一个数组 y

  wks = gsn_open_wks("eps","title_ex")

  res = True
  res@tiMainString = "This is a main title" ; 添加主图题
  res@tiMainFontHeightF = 0.05 ; 主图题字体大小
  res@tiMainFontColor = "black" ; 主图题字体颜色
```

```

res@gsnCenterString = "Center String" ; 添加中间标题
res@gsnLeftString   = "Left String"   ; 添加左标题
res@gsnRightString  = "Right String"  ; 添加右标题

res@tiXAxisString = "X axis"          ; 添加 X 轴名称
res@tiYAxisString = "Y axis"          ; 添加 Y 轴名称
res@tiXAxisFontHeightF = 0.03         ; X/Y 轴名称的字体大小
res@tiXAxisFontColor = "black"        ; X/Y 轴名称的字体颜色

plot = gsn_csm_y(wks, y, res)
end

```

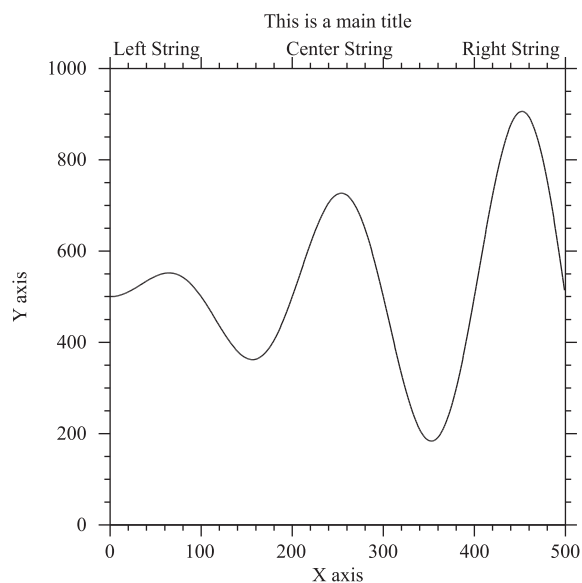


图 2.1 标题及坐标轴标签

2.3.2 色板的使用

```

ex_color.ncl(图 2.2):
begin
  f = addfile("./data/air.mon.mean.nc", "r") ; NCEP/NCAR 再分析资料(Kal-
  nay et al., 1996)
  var = short2flt(f->air(0, :, :, :)) ; 将短(short)型数据转换为浮点型,数组

```

大小为[time | 753] x [level | 17] x [lat | 73] x [lon | 144]

```
wks = gsn_open_wks("eps","color")
gsn_define_colormap(wks,"hotcold_18lev") ;将默认色板更换为"hotcold_18lev"
```

```
res = True
res@gsnAddCyclic = True
res@cnFillOn = True
res@pmTickMarkDisplayMode = "Always" ;坐标标签上添加度符号
```

```
res@gsnLeftString = "" ;不绘制左标题
res@gsnRightString = "" ;不绘制右标题
```

;;;绘制指定的等值线及填充的颜色

```
res@cnLevelSelectionMode = "ExplicitLevels"
res@cnLevels = (/ -35, -25, -15, -5, 5/)
res@cnFillColors = (/2,4,6,8,11,13/)
```

```
plot = gsn_csm_contour_map_ce(wks,var({700},:,:),res)
end
```

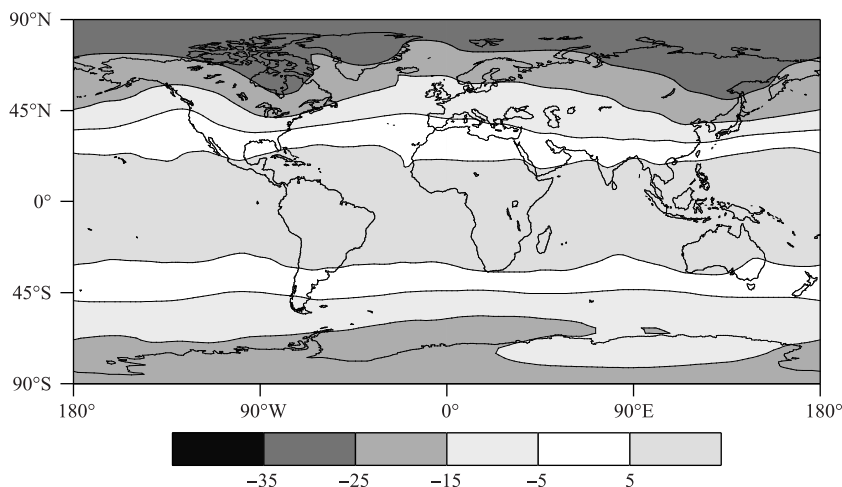


图 2.2 气温图(单位:℃, 附彩图 2.2)

2.3.3 一根折线

```

ex_xy.ncl(图 2.3):
begin
  y = sin(0.0628 * ispan(0,100,1))

  wks = gsn_open_wks("eps", "xy_ex")

  res = True
  res@xyLineColor      = 1      ; 通常而言,1 表示前景色为黑色,0 表示背景色
                             为白色
  res@xyLineThicknessF = 4.5    ; 4.5 倍粗

  res@xyMarkLineMode   = "MarkLines" ; 用点线绘制折线
  res@xyMarker          = 16          ; 见附录图 H.2
  res@xyMarkerColor     = "black"     ; 点的颜色

  plot = gsn_csm_y(wks,y,res)
end

```

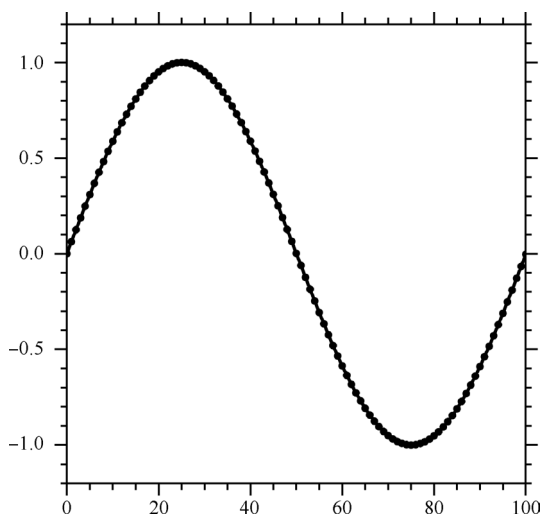


图 2.3 折线

2.3.4 多根折线及图例

```

ex_xy_legend.ncl(图 2.4):
begin
  npts = 500
  x = fspan(-100,100,npts) ; 创建一个从-100 至 100 的 500 个数值的等
差数列

  y = new((/4,500/),"float") ; 创建一个 2D 数据,用以表示 4 根折线
  y(0,:) = sin(0.0628 * x)
  y(1,:) = cos(0.0628 * x)
  y(2,:) = cos(0.0628 * x) * sin(0.0628 * x)
  y(3,:) = cos(0.0628 * x) + sin(0.0628 * x)

  wks = gsn_open_wks("eps","xy_legend_ex")

  res = True

  ; 坐标轴的取值范围
  res@trXMinF = min(x)
  res@trXMaxF = max(x)
  res@trYMinF = min(y)-0.2 ; 留出适当的空白,绘制图例
  res@trYMaxF = max(y)+0.6 ; 留出适当的空白,绘制图例

  res@xyLineThicknesses = (/1,2,3,4/) ; 不同的粗细
  res@xyLineColors = (/ "black", "black", "black", "black" /) ; 也可用不同的
颜色,比如 (/ "NavyBlue", "HotPink", "OrangeRed", "ForestGreen" /)
  res@xyDashPatterns = (/0,2,4,6/) ; 不同的线型,见附录图 H.6

  res@pmLegendDisplayMode = "Always"; 添加图例
  res@xyExplicitLegendLabels = (/ "one", "two", "three", "four" /) ; 每条折线的
名称
  res@pmLegendWidthF = 0.2 ; 图例的宽度
  res@pmLegendHeightF = 0.1 ; 图例的高度

```

res@pmLegendOrthogonalPosF = -1.05 ; 图例垂直移动的量
res@pmLegendParallelPosF = 0.82 ; 图例水平移动的量

```
plot = gsn_csm_xy(wks, x, y, res)
end
```

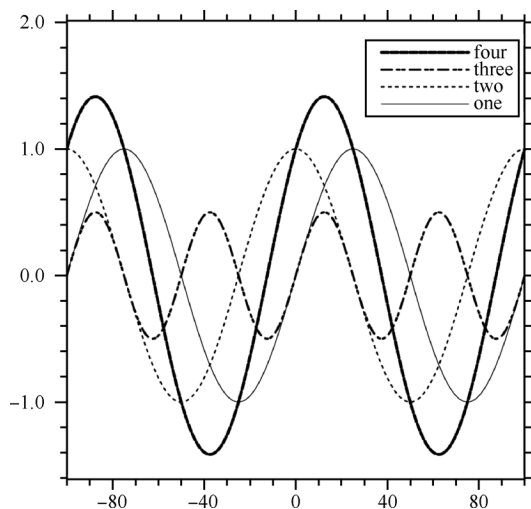


图 2.4 折线及图例

2.3.5 等值线及坐标轴刻度

`ex_contour_tm.ncl`(图 2.5):

```
begin
```

`z = generate_2d_array(15,15,-100.,110.,0,(/40,40/))` ; 创建一个(/40,40/)的二维随机数组,该数组中最小值为-100,最大值为 110

```
wks = gsn_open_wks("eps","contour_tm_ex")
```

```
res = True
```

```
res@gsnDraw = False
```

```
res@gsnFrame = False
```

res@gsnContourZeroLineThicknessF = 0. ; 略去 0 线

```

res@cnLevelSelectionMode = "ManualLevels"    ; 等间隔地指定等值线范围
res@cnMinLevelValF       = -50
res@cnMaxLevelValF       = 60
res@cnLevelSpacingF      = 10

res@cnFillOn              = False

res@cnLinesOn             = True
res@cnLineThicknessF      = 2

res@lbOrientation = "Vertical"

res@tmXBMode = "Explicit"          ; 指定 X 轴坐标标签
res@tmXBValues = (/5,15,25,35/)    ; 在数值为 5/15/25/35 的 X 轴上分别用
tmXBLabels 中的数值表示
res@tmXBLabels = (/ "Jan", "Feb", "Mar", "Apr" /)

res@tmXTOn = False

res@tmYLMode = "Manual"            ; 默认“Automatic”
res@tmYLTickStartF = 0              ; 最小刻度值
res@tmYLTickEndF   = 40            ; 最大刻度值
res@tmYLTickSpacingF = 5           ; 刻度间隔

plot = gsn_csm_contour(wks,z,res)
plot = ColorNegDashZeroPosContour(plot,"black","black","black") ; 正值用黑
色实线,负值用黑色虚线. 注意,虽然此处对 0 值等值线设置了黑色,但由于前面 res
@gsnContourZeroLineThicknessF = 0 已将其设为 0 倍粗细,所以图中并未显示 0 值
等值线

draw(plot)
frame(wks)
end

```

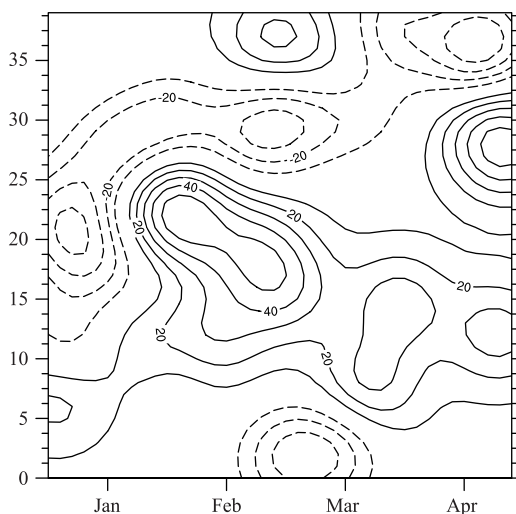


图 2.5 等值线及坐标轴

2.3.6 等值线及地图

`ex_cnmap.ncl`(图 2.6):

`begin`

`f=addfile("./data/hgt.sfc.nc","r")` ; NCEP/NCAR 再分析资料(Kalnay et al., 1996)

`var = short2flt(f->hgt(0, :, :))` ;将短型数据转换为浮点型

`wks = gsn_open_wks("eps","contourmap")`

`gsn_define_colormap(wks,"GMT_gray ")`

`res` = `True`

`res@gsnLeftString` = "" ; 不绘制左标题

`res@gsnRightString` = "" ; 不绘制右标题

`res@pmTickMarkDisplayMode` = "Always" ; 坐标标签上添加度符号

;;绘制的地图区域

`res@mpMinLatF` = 10

`res@mpMaxLatF` = 80

`res@mpMinLonF` = 30

```
res@mpMaxLonF    = 130
```

```
res@mpFillOn      = False
```

```
res@mpGridAndLimbOn = True    ; 设置经纬度线
```

```
res@mpGridLineThicknessF = 0.5
```

```
res@mpGridLineDashPattern = 2    ; 见附录图 H.6
```

```
res@mpGridSpacingF    = 10.
```

```
res@cnFillOn        = True
```

```
res@cnLinesOn       = False
```

```
res@cnLevelSelectionMode = "ExplicitLevels"    ; 绘制特定等值线
```

res@cnLevels = (/1500,2500,3500,4500/) ; 由于是等差数列,因此也可将上行的“cnLevelSelectionMode”设为“ManualLevels”,再设定等值线数值范围及其间隔

res@cnFillColors = (/ -1,10,8,6,4 /) ; 使用色板“GMT_gray”中对应位置的颜色

```
plot = gsn_csm_contour_map(wks, var, res)
end
```

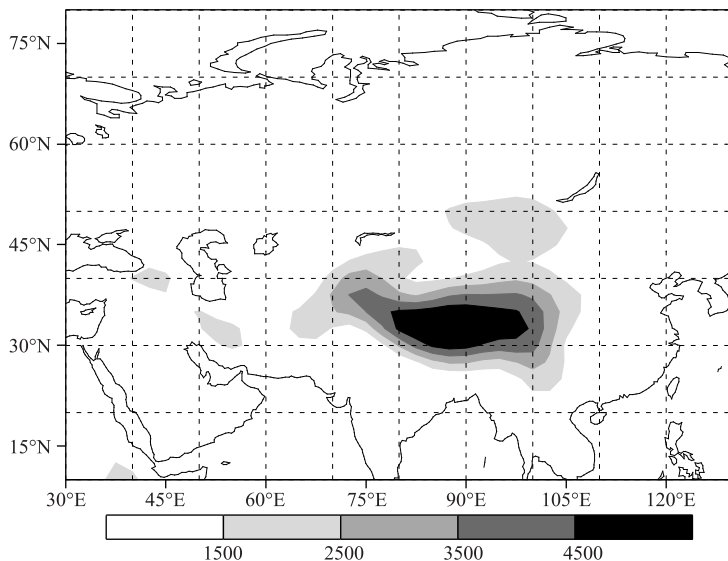


图 2.6 地图及填色等值线(单位:m)

2.3.7 矢量箭头

```

ex_vector.ncl(图 2.7):
begin
  fu = addfile("./data/u850-197901-201412.nc","r") ; NCEP/NCAR 再分析
资料(Kalnay, et al. 1996.)
  fv = addfile("./data/v850-197901-201412.nc","r") ;

  u = short2flt(fu->uwnd(0,{850},:,:)) ;仅挑取第 0 个时次
  v = short2flt(fv->vwnd(0,{850},:,:))

  wks = gsn_open_wks("eps","vector_ex")

  res                                = True
  res@gsnLeftString                  = ""
  res@gsnRightString                 = ""

  res@pmTickMarkDisplayMode = "Always" ; 坐标标签上添加度符号

  res@vcGlyphStyle = "FillArrow" ; 见附录图 H.1

  res@vcFillArrowEdgeColor = "white" ; 箭头边缘颜色
  res@vcFillArrowFillColor = "black" ; 箭头内部的填充颜色
  res@vcMinMagnitudeF      = 5.0      ; 小于 5 则不绘制
  res@vcMinDistanceF       = 0.02     ; 设定箭头之间的最小距离(在单位坐
标系中),防止某些地区(比如极区附近)箭头过密(或过疏)

  ;; 箭头头部大小一致。若要绘制头部大小不一致箭头可参考第 4.3.1 节示例
  res@vcFillArrowMinFracWidthF = 1.0
  res@vcFillArrowHeadMinFracXF = 1.0
  res@vcFillArrowHeadMinFracYF = 1.0

  ;; 参考箭头

```

```

res@vcRefAnnoOn           = True
res@vcRefMagnitudeF       = 20      ;单位长度箭头所表示的实际物
理量的大小
res@vcRefLengthF          = 0.04   ;单位长度箭头在图形中的长度
res@vcRefAnnoBackgroundColor = "white" ;背景颜色
res@vcRefAnnoPerimOn      = False   ;关闭边框
res@vcRefAnnoFontHeightF  = 0.015  ;参考箭头标签的字体大小
res@vcRefAnnoString1On    = False   ;设定参考箭头上方的字符
res@vcRefAnnoString2On    = True    ;不设定参考箭头下方的字符
res@vcRefAnnoString2      = "20 m/s"
res@vcRefAnnoOrthogonalPosF = -1.25 ;垂直移动其位置

```

```

plot = gsn_csm_vector_map(wks,u,v,res)
end

```

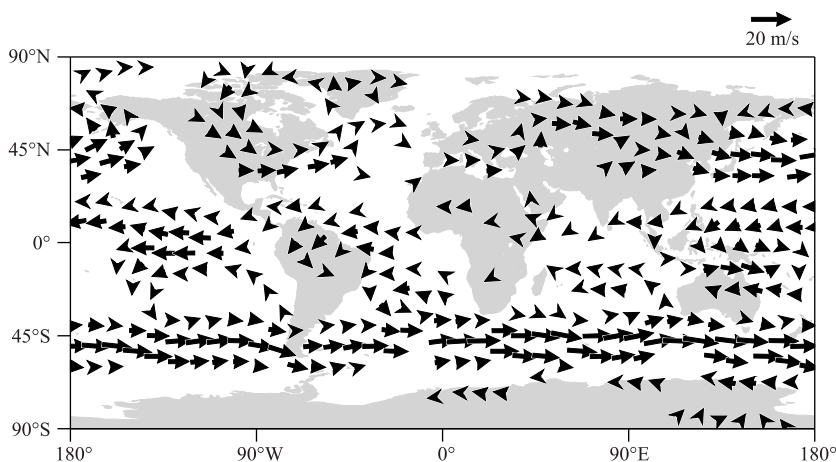


图 2.7 850 hPa 风场图

2.3.8 标识、多边形、任意折线及文本

```

ex_poly.ncl(图 2.8):
begin
;南京的经纬度
lat_nj = 32.03

```

```
lon_nj = 118.46
```

;假定一低压中心在某 5 天中的经纬度位置

```
lat_line = (/20,21,22,24,27,29/)
```

```
lon_line = (/124,122.5,120,119,120,121/)
```

;假定该低压系统在后 4 天中的主要影响范围

```
center_lat = 25
```

```
center_lon = 120
```

```
maxlat_region = center_lat+5
```

```
minlat_region = center_lat-5
```

```
maxlon_region = center_lon+3
```

```
minlon_region = center_lon-3
```

```
wks = gsn_open_wks("eps","poly_ex")
```

```
res = True
```

```
res@gsnFrame = False
```

```
res@gsnDraw = False
```

```
res@mpMinLatF = 15
```

```
res@mpMaxLatF = 40
```

```
res@mpMinLonF = 110
```

```
res@mpMaxLonF = 130
```

```
res@pmTickMarkDisplayMode = "Always" ; 坐标标签上添加度符号
```

```
map = gsn_csm_map(wks,res)
```

;添加多边形(即低压影响的范围)。注意,为使图形闭合,最后一个点的位置需与第一个点重合

```
boxlat = (/ maxlat_region, maxlat_region, minlat_region, minlat_region, maxlat_region/)
```

```
boxlon = (/minlon_region, maxlon_region, maxlon_region, minlon_region,
minlon_region/)
```

```
gonres                = True
gonres@gsFillColor    = "black"
gonres@gsFillOpacityF = 0.5      ; 50% 透明
dum = gsn_add_polygon(wks, map, boxlon, boxlat, gonres)
```

;; 添加任意折线(即低压中心的移动路径)

```
lnres                = True
lnres@gsLineColor    = "black"
lnres@gsLineThicknessF = 3.0
lnres@gsLineDashPattern = 16      ; 见附录图 H.6
dum2 = gsn_add_polyline(wks, map, lon_line, lat_line, lnres)
```

;; 添加 marker(即低压中心位置)

```
mkres                = True
mkres@gsMarkerColor  = "black"
mkres@gsMarkerSizeF  = 0.01      ; 标识大小。在 0~1 范围内设置
mkres@gsMarkerIndex  = 16        ; 见附录图 H.2
dum22 = gsn_add_polymarker(wks, map, lon_line, lat_line, mkres)
```

;; 添加 marker(南京位置)

```
mkres@gsMarkerColor  = "black"
mkres@gsMarkerSizeF  = 0.02
mkres@gsMarkerIndex  = 3
dum3 = gsn_add_polymarker(wks, map, lon_nj, lat_nj, mkres)
```

;; 添加文字

```
txres                = True
txres@txFontHeightF  = 0.015
txres@txFontColor    = "black"
dum4 = gsn_add_text(wks, map, "Nanjing", lon_nj, lat_nj+1.5, txres)
```

```

draw(map)
frame(wks)
end

```

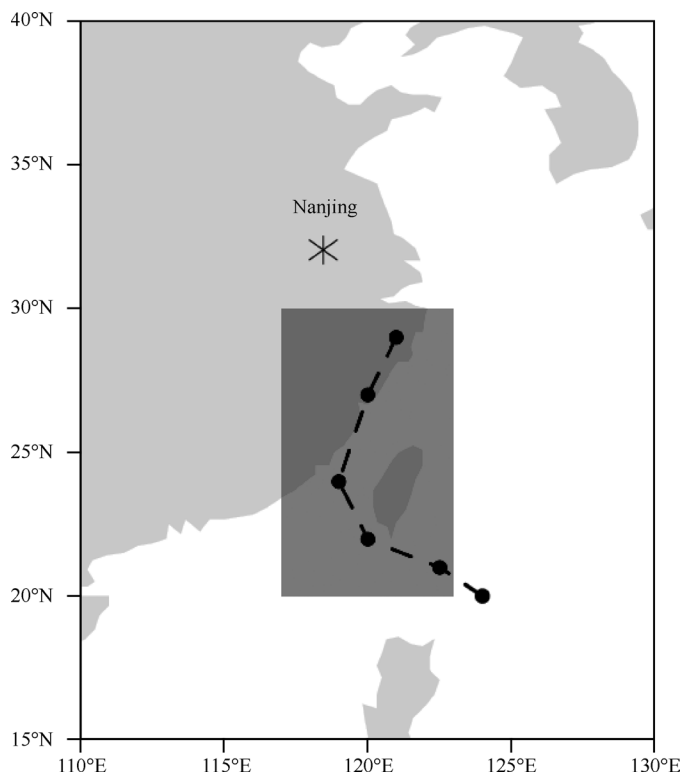


图 2.8 标识、多边形、任意折线及文本

2.3.9 组图

ex_panel.ncl(图 2.9):

```
begin
```

;生成四组(/129,129/)的二维随机数组

data1 = generate_2d_array(10, 12, 0., 27., 0, (/129,129/));最小(大)值为 0(27)

data2 = generate_2d_array(13, 13, 5., 25., 1, (/129,129/));最小(大)值为 5(25)

data3 = generate_2d_array(10, 10, 2., 29., 2, (/129,129/));最小(大)值为 2(29)

data4 = generate_2d_array(9, 10, 3., 24., 3, (/129,129/));最小(大)值为 3(24)


```
plot = new(4, graphic)
wks = gsn_open_wks("eps", "panel_ex")
gsn_define_colormap(wks, "gsltod") ; 使用灰色色板 gsltod, 见附录图I.3(a)
```

```
res = True
res@gsnFrame = False
res@gsnDraw = False
```

```
res@cnFillOn = True
res@cnLinesOn = False
res@cnLineLabelsOn = False
res@cnInfoLabelOn = False
```

; 为公用一个色标, 需为每幅图设定相同的等值线

```
res@cnLevelSelectionMode = "ExplicitLevels"
```

```
res@cnLevels = (/2, 5, 7, 10, 13, 16, 19, 22, 25, 28/)
```

```
res@lbLabelBarOn = False ; 关闭每个图形各自的色标, 以公用一个色标
```

```
plot(0) = gsn_csm_contour(wks, data1, res)
plot(1) = gsn_csm_contour(wks, data2, res)
plot(2) = gsn_csm_contour(wks, data3, res)
plot(3) = gsn_csm_contour(wks, data4, res)
```

```
pres = True
```

```
pres@txString = "Paneling"
```

```
pres@gsnPanelLabelBar = True ; 公用一个色标
```

```
pres@gsnPanelFigureStrings = (/ "A", "B", "C", "D" /)
```

```
pres@amJust = "TopLeft" ; 调整标签的位置
```

```
gsn_panel(wks, plot, (/2, 2/), pres)
```

```
end
```

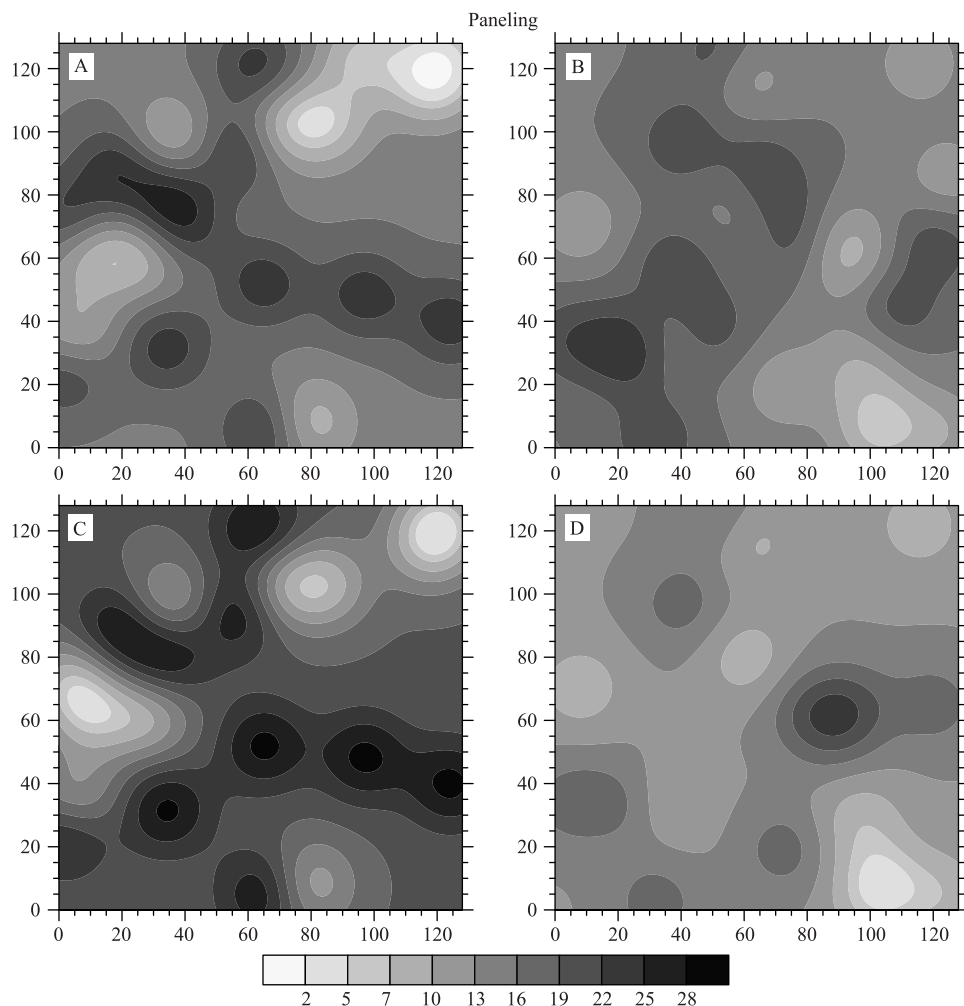


图 2.9 2X2 组图

2.4 思考题

(1) 修改 `ex_color.ncl`, 改用“BlueDarkRed18”色板, 小于 -4 用蓝色, 大于 4 用红色, 其余用背景色白色。

(2) 结合 `ex_xy_legend.ncl` 与 `ex_poly.ncl`, 在 `xy` 折线图中 `[-40, 1]` 处添加红色星号, 在 `[-30, 1]` 与 `[-20, -1]` 两点处绘制任意蓝色折线(可参考第 3 章示例及附录 G)。

(3) 修改 `ex_panel.ncl`, 4 幅图形按如下方式排列: 第一行 2 幅, 第二行 1 幅, 第三行 1 幅(可参考第 4 章示例)。

第 3 章 台站降水量随时间的演变

——折线图

3.1 实习目的

- (1)掌握读取及输出 ASC II 文件、binary 文件的基本方法。
- (2)进一步掌握折线及图例的绘制方法。
- (3)掌握绘制直方图的方法。注意,NCL 官网提供的柱状图(Histogram)绘图方法中所调用的 gsn_histogram 绘图函数是对变量的频率或数值范围区间进行统计,这与此例所调用的绘图函数 gsn_csm_xy 并不相同。
- (4)掌握在非地图图形上添加文本及直方图图例的基本方法。

3.2 实习内容及要求

首先下载附录 B 中“实习资料下载地址”中所有以“txt”为结尾的降水资料,以及如下四个 NCL 脚本:data-station.ncl,plot-xy-4stat.ncl,plot-bar-bj.ncl 和 plot-bar-2stat.ncl。再根据以下实习内容与要求,补充完善相应脚本。

(1)输入输出 ASC II、binary 文件,添加元数据(meta data)。补充 data-station.ncl 脚本,读取国家气候中心提供的 1951 年至 2011 年 160 站 6、7、8 月降水资料(ASC II 文件, <http://ncc.cma.gov.cn/Website/index.php?ChannelID=43&WCHID=5>),计算各站的夏季(JJA)降水量,将 1979 年至 2008 年这一时间段的数据输出为 binary 及 ASC II 文件。

(2)绘制修改折线及图例,调整视图(viewport)的长宽比。补充 plot-xy-4stat.ncl 脚本,绘制北京、上海、丽江、乌鲁木齐四个台站的逐年夏季降水量。用四根不同线型、颜色、粗细的折线表示四个台站的夏季降水量。添加降水量数值为 150 的阈值线,标出 2000 年对应的直线。将图例移至图中。

(3)绘制直方图。补充 plot-bar-bj.ncl 脚本,计算北京站自 1979 年后各年的夏季异常降水量,用直方图表示。用黑色直方图表示正异常,灰色直方图表示负异常。

(4)添加文本及直方图图例。补充 plot-bar-2stat.ncl 脚本,用直方图绘制北

京及上海两站在 1980、1990、2000 年这三年中的夏季降水量。在一幅图中用不同图型的直方图表示两个台站,添加图例。

3.3 脚本讲解及图形示例

此节给出 3.2 节中所有的完整脚本,其中粗体为关键语句。网上提供的原脚本中已删除这些关键语句,读者可自行参考对照。

3.3.1 台站降水资料的读取、处理及输出

```
data-station.ncl:
begin
  nstat = 160 ; 160 站
  year_s = 1951
  year_e = 2011
  nyear = year_e - year_s + 1

  ; 读入 6 月降水量
  path6 = "./data/r1606.TXT"
  m6 = asciiread(path6, (/nyear, nstat/), "float")

  ; 添加元数据
  m6!0 = "year"
  m6!1 = "stations"
  m6&.year = ispan(year_s, year_e, 1)
  m6&.stations = ispan(1, nstat, 1)

  m7 = m6 ; 创建新的变量 m7
  m8 = m6 ; 创建新的变量 m8
  ; 读入 7 月、8 月两个月的降水量
  path7 = "./data/r1607.TXT"
  path8 = "./data/r1608.TXT"

  m7 = asciiread(path7, (/nyear, nstat/), "float")
  m8 = asciiread(path8, (/nyear, nstat/), "float")
```

;仅计算 1979 年至 2008 年逐年的 JJA 平均

mm = m6({1979:2008}, :) ;创建新的变量

mm = m6({1979:2008}, :) + m7({1979:2008}, :) + m8({1979:2008}, :)

;输出为无格式二进制文件

fbindirwrite("./data/precip-160-JJA-30yr.grd", mm) ; 注意, 如果该路径下已有同名文件, 则输出的 **mm** 将被追加至该文件的最后, 而不会替换该文件

;输出为 ASCII 文件

opt = True

opt@fout = "./data/precipitation-JJA.txt"

write_matrix(mm, "160f7.2" , opt) ; 每行 160 个值

end

3.3.2 四个特定台站的夏季降水(折线)

plot-xy-4stat.ncl(图 3.1):

begin

name = (/ "Beijing", "Shanghai", "Lijiang", "Wulumuqi" /)

irec = (/33,56,115,160/) - 1 ; 对应北京、上海、丽江及乌鲁木齐四个站在 160 站中的位置, 具体可查看 station-name.txt

ntime = 30 ; 共 30 年时间

nstat = 160 ; 共 160 站

;读取全部数据, 注意数据存放的路径, 它在该例中存放在 ./data 下

precip = fbindirread("./data/precip-160-JJA-30yr.grd", 0, (/ntime, nstat/), "float")

;添加元数据

precip!0 = "year"

precip!1 = "station"

precip&.year = ispan(1979, 2008, 1)

precip&.station = ispan(1, 160, 1)

```

; 挑出这四个站的数据
preci_st = preci(:,irec)

; *****
; create plot
; *****

wks = gsn_open_wks("eps", "plot-precip-4stations") ;生成 eps 图

res = True

res@vpHeightF = 0.35 ; 改变视图的大小
res@vpWidthF = 0.6 ;

res@trXMinF = 1979 ; 设定 X 轴坐标范围
res@trXMaxF = 2008
res@trYMaxF = 1400 ; 设定 Y 轴坐标范围
res@trYMinF = 0

res@xyMonoLineColor = False ; 折线使用多种颜色
res@xyLineThicknesses = (/3., 1., 1., 1./) ; 设定不同的粗细, 3 为 3 倍粗, 4
个值对应 4 个站
res@xyDashPattern = 0 ; 设定全部折线为实线
res@xyMarkLineModes = (/ "MarkLines", "MarkLines", "MarkLines", "Mark-
Lines"/) ; 设定折线的类型, 如果都是“Markers”, 则用标识表示折线
res@xyMarkers = (/9, 7, 5, 3/) ; 设定折线上标识的类型。见附录图 H.2
res@xyMarkerSizeF = 0.01 ; 设定折线上标识的大小
res@xyLineColors = (/ "black", "black", "black", "black"/) ; 设定折线的颜色

res@tiXAxisString = "year" ; 添加 X 轴名称
res@tiYAxisString = "precipitation (units: mm)" ; 添加 Y 轴名称

res@pmLegendDisplayMode = "Always" ; 打开图例
res@pmLegendSide = "top" ; 改变图例的位置, 默认为“bottom”
res@pmLegendParallelPosF = 0.18 ; 向右移动

```

```

res@pmLegendOrthogonalPosF = -0.32 ; 向下移动
res@pmLegendWidthF         = 0.2   ; 改变图例的宽度
res@pmLegendHeightF        = 0.1   ; 改变图例的高度

res@lgLabelFontHeightF     = .015   ; 改变图例标签字体的大小
res@lgPerimOn              = True   ; 图例上不绘制边框

res@xyExplicitLegendLabels = name    ; 指定图例标签的文字内容

res@gsnXRefLine            = 2000   ; 绘制垂直于 X 轴的参考线
res@gsnXRefLineDashPattern = 16     ; 设定其线型
res@gsnXRefLineThicknessF = 0.5     ; 设定其粗细

res@gsnYRefLine            = 150    ; 绘制垂直于 Y 轴的参考线
res@gsnYRefLineDashPattern = 16     ; 设定其线型
res@gsnYRefLineThicknessF = 0.5     ; 设定其粗细

top_plot = gsn_csm_xy(wks, preci_st & year, preci_st(station|:, year|:), res)

```

end

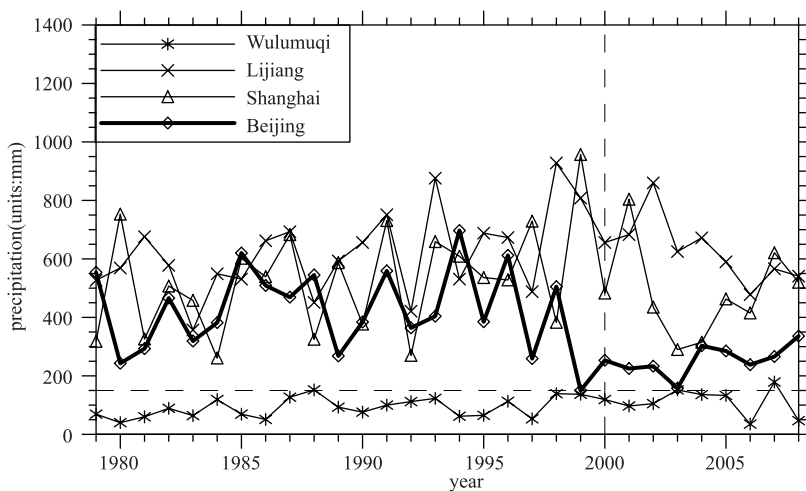


图 3.1 四个台站的夏季降水量

3.3.3 北京站逐年夏季降水量(直方图)

```

plot-bar-bj.ncl(图 3.2):
begin
  irec = 33 - 1 ; 对应北京站在 160 站中位置,具体可查看 station-name.txt

  ntime = 30 ; 共 30 年时间
  nstat = 160 ; 共 160 站

;读取全部数据,注意数据存放的路径,该例中存放在./data 下
preci = fbindirread("./data/precip-160-JJA-30yr.grd",0,(/ntime,nstat/),"float")

;添加元数据
preci!0="year"
preci!1="station"
preci&.year = ispan(1979,2008,1)
preci&.station = ispan(1,160,1)

;挑出北京站的数据,并计算各年的异常,保留元数据
preci_st = dim_rmvmed_n_Wrap(preci(:,irec),0)

wks = gsn_open_wks("eps","plot-bar-1station") ;生成 eps 图

res = True
res@gsnDraw = False ; 暂不画
res@gsnFrame = False ; 暂不翻页

res@vpHeightF = 0.35 ; 改变视图的大小
res@vpWidthF = 0.6 ;

res@trXMinF = 1979 ; 设定 X 轴坐标范围
res@trXMaxF = 2008 ;
res@trYMaxF = 450 ; 设定 Y 轴坐标范围
res@trYMinF = -300 ;

```



```
res@gsnXYBarChart          = True
res@gsnXYBarChartBarWidth  = 0.5          ; 设定直方柱(bar)的宽度
```

; 设定 bar 图的颜色

```
res@gsnYRefLine            = 0.          ; 设定参考值
res@gsnAboveYRefLineColor  = "black"     ; 大于该参考值用黑色
res@gsnBelowYRefLineColor  = "gray"      ; 小于该参考值用灰色
```

; res@gsnXYBarChartColors = (/ "black" /) ; 如果不设定参考值, 则可通过该绘图参数来设定颜色

```
res@tiXAxisString= "year"          ; 添加 X 轴名称
res@tiYAxisString= "precipitation (units: mm)" ; 添加 Y 轴名称
```

```
top_plot = gsn_csm_xy(wks, preci_st&.year, preci_st, res)
```

```
draw(top_plot) ; 绘图
```

```
frame(wks)      ; 翻页
```

```
end
```

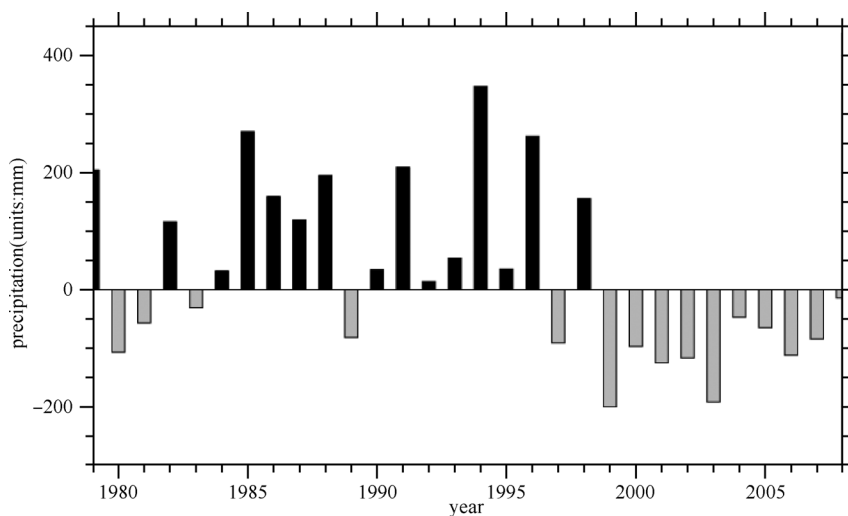


图 3.2 北京站逐年夏季降水量

3.3.4 北京、上海两站在三年中的夏季降水量(直方图)

```

plot-bar-2stat.ncl(图 3.3):
begin
  name = (/ "Beijing", "Shanghai" /)
  irec = (/33,56/) -1      ; 指定要分析的台站
  fp   = (/6,17/)         ; 设定直方图类型用以表示两个台站降水量见附录图
H.3
  co   = (/ "black", "gray" /) ; 设定直方图填充的颜色

  syear = ispan(1980,2000,10) ; 指定要分析的年份,也可直接赋值 syear=(/1980,
1990,2000/)
  nyear = dimsizes(syear)

  ntime = 30   ; 共 30 年时间
  nstat = 160  ; 共 160 站

; 读取全部数据,注意数据存放的路径,该例中存放在 ./data 下
prec = fbindirread("./data/prec-160-JJA-30yr.grd",0,(/ntime,nstat/),"
float")

; 添加元数据
prec!0="year"
prec!1="station"
prec&.year = ispan(1979,2008,1)
prec&.station = ispan(1,160,1)

; 挑出两站在特定年份的数据,这里利用了命名的行列(named dimension)对行列
顺序进行了转置
prec_st = prec(station|irec,{year|syear})
ndim = dimsizes(prec_st)

; *****
wks = gsn_open_wks("eps", "plot-bar-2stat")

```

```
res = True
res@gsnDraw = False
res@gsnFrame = False
```

```
res@vpWidthF = 0.7
res@vpHeightF = 0.5
```

```
res@trXMinF = 0.5
res@trXMaxF = 3.5
res@trYMinF = 0
res@trYMaxF = 900
```

```
res@tmXBMode = "Explicit" ; 指定坐标标签
res@tmXBValues = ispan(1, nyear, 1)
res@tmXBLLabels = ispan(1980, 2000, 10)
```

```
res@tiXAxisString = "year" ; 添加 X 轴名称
res@tiYAxisString = "precipitation (units: mm)" ; 添加 Y 轴名称
```

```
bw = 0.2
res@gsnXYBarChart = True
res@gsnXYBarChartBarWidth = bw ; 设定直方柱的宽度
res@gsnXYBarChartPatterns = fp(0)
res@gsnXYBarChartColors = co(0)
res@gsnXYBarChartOutlineThicknessF = 1
plot1 = gsn_csm_xy(wks, ispan(1, nyear, 1) - bw/2, preci_st(0, :), res); 每一个
台站都要单独画一次
```

```
res@gsnXYBarChartPatterns = fp(1) ; 用不同类型
res@gsnXYBarChartColors = co(1) ; 用不同颜色
plot2 = gsn_csm_xy(wks, ispan(1, nyear, 1) + bw/2, preci_st(1, :), res)
```

;在直方图上添加数字,表明降水量数值

```
txres = True
```

```
txres@txFontHeightF = 0.015 ; 设定字体大小
```

```
label_value = ndtooned(preci_st)
```

```
label = floattointeger(label_value) ; 虽有“warning”出现,但不影响出图
```

;; 如果用户需画多个台站,则需自行对如下数组及参数进行修改

```
xloc = preci_st ; 新建一个与 preci_st 一样大小的数组
```

```
xloc(0,0:ndim(1)-1) = ispan(1,nyear,1) - bw/2
```

```
xloc(1,0:ndim(1)-1) = ispan(1,nyear,1) + bw/2
```

```
xloc1d = ndtooned(xloc)
```

```
yloc1d = label_value+10 ; 向上挪动
```

```
text = gsn_add_text(wks,plot2,label,xloc1d,yloc1d ,txres)
```

; 在空白处手动添加图例

```
lbres = True
```

```
lbres@vpWidthF = 0.3 ; 视图的宽度
```

```
lbres@vpHeightF = 0.1 ; 视图的高度
```

```
lbres@lbBoxMajorExtentF = 0.36 ; 缩小图例,使其中间有些空白
```

```
lbres@lbMonoFillPattern = True ; 是否采用同种填充图案
```

```
lbres@lbLabelFontHeightF = 0.015 ; 标签字体大小
```

```
lbres@lbPerimOn = False ; 周围是否加上方框
```

```
; lbres@lgPerimColor = "black" ; 方框的颜色
```

```
lbres@lbFillPattern = fp(1)
```

```
lbres@lbFillColors = co(1)
```

gsn_labelbar_ndc(**wks**,1,**name**(1),0.6,0.7,**lbres**) ; 绘制上海站直方图的图例,
第2个参数表示只画1个图例框

```
lbres@lbFillPattern = fp(0)
```

```
lbres@lbFillColors = co(0)
```

gsn_labelbar_ndc(**wks**,1,**name**(0),0.587,0.76,**lbres**) ; 绘制北京站直方图的
图例

```
draw(plot1)
draw(plot2)
frame(wks)
end
```

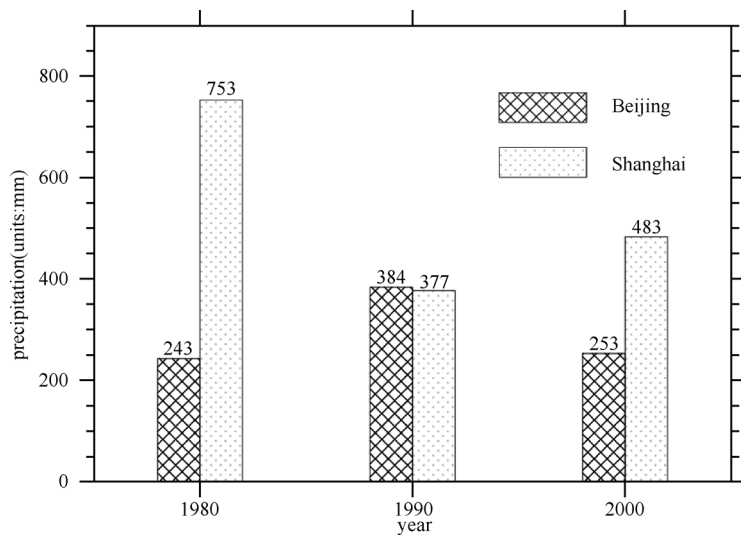


图 3.3 北京、上海站在 1980 年、1990 年、2000 年中的夏季降水量

3.4 思考题

(1) 修改 `plot-bar-bj.ncl`, 改为折线图, 大于 0 的用红色填充, 小于 0 的用蓝色 (可参考第 5 章示例)。

(2) 修改 `plot-bar-2stat.ncl`, 用直方图表示北京、上海和丽江三个站在 2001 年至 2005 年中的异常降水量。

第4章 厄尔尼诺年的环流合成

——NetCDF数据读取、计算和存储, 等值线、矢量箭头、图层叠加

4.1 实习目的

- (1)掌握面积权重平均的计算方法。
- (2)掌握 NetCDF 数据的读取方法。
- (3)掌握使用多种色板的基本方法。
- (4)进一步掌握矢量箭头的绘图方法。
- (5)掌握图层的叠加方法。
- (6)进一步掌握组图的绘制方法。

4.2 实习内容及要求

首先下载附录B中“实习资料下载地址”中所有以“.nc”为结尾的数据资料以及 plot-comp-enso.ncl 脚本。其中海表面温度(sea surface temperature, SST)为 NOAA Extended Reconstructed Sea Surface Temperature (ERSST) Version 4 资料,名为 sst-197901-201501.nc;其余四种资料均为 NCEP/DOE AMIP-II 再分析资料(Kanamitsu et al, 2012),名称分别为 h300-197901-201412.nc、u850-197901-201412.nc、v850-197901-201412.nc。再根据以下实习内容与要求,补充完善 plot-comp-enso.ncl 脚本。

(1)nc 数据读取及变量的截取。读取 1979 年 1 月至 2014 年 11 月的 sst、h300、u850、v850、air2m。注意,各变量的分辨率、时间长度不同。选取 1979 年 12 月至 2014 年 11 月这一共同时段进行处理分析。

(2)数值计算及 t 检验。计算各变量的冬季(DJF)平均及每年的冬季异常。计算 nino3.4 指数(区域平均),并挑选出大于 0.8 个标准差的所有年份并进行合成和检验(t 检验)。

(3)绘制修改等值线、矢量箭头,图层叠加。绘制合成的厄尔尼诺年(El nino, Bjerknes, 1966; Bjerknes, 1969)的 sst、h300、V850 及 T_{2m} 。绘制图形时,①经纬度线间

隔 15 度,细虚线表示;②用两种灰色分别表示通过 95% 及 99% 显著性检验的区域;③在绘制 sst 合成场时,仅绘制 -0.75, -0.25, 0.25, 0.75 和 1.25 这 5 根等值线;④绘制合成的 h300 及 850 hPa 的风场时,将 850 hPa 的风场叠加在 h300 场上;⑤h300 的等值线间隔为 15;⑥绘制 T_{2m} 时,仅绘制从 -2 至 2,间隔为 0.5 的等值线;⑦在海温合成图上用一定程度的透明填色表示 nino 3.4 区,边框用黑色;⑧三幅子图分两行摆放,第 1 行 1 幅图,第 2 行 2 幅图,且每幅图上添加标号及各变量名称;⑨整组图的图题上方写上“El nino”。

4.3 脚本讲解及图形示例

此节给出 4.2 节中所有的完整脚本,其中粗体为关键语句。网上提供的原脚本中已删除这些关键语句,读者可自行参考对照。

4.3.1 厄尔尼诺年各变量的合成图

```

plot-comp-enso.ncl(图 4.1):
begin
year=ispan(1979,2013,1) ; 1979 年至 2013 年共 35 年的冬季

it_s=197912 ;起始年月
it_e=201411 ;结束年月

refmag = 3 ;标准长度的参考箭头所表示的风速大小

;;;read data ;;;;;;;;;;;;;;
;; sst
f_sst = addfile("./data/sst-197901-201501.nc", "r")

time = f_sst->time ; 读取日期
YYYYMM = cd_calendar(time, -1) ; 转换成公历日期

rec_s = ind(it_s.eq.YYYYYMM) ;开始日期的记录号,对应着 1979 年 12 月
rec_e = ind(it_e.eq.YYYYYMM) ;结束日期的记录号,对应着 2014 年 11 月
sst = f_sst->sst(rec_s:rec_e, :, :) ;截取 1979 年 12 月至 2014 年 11 月这一
时段资料,即 1979 年 12 月是第一时次

```

```
;; h300
f_h300 = addfile("./data/h300-197901-201412.nc", "r")
h300 = short2flt(f_h300->hgt(rec_s:rec_e,0,{-90:90},:))

;; u850
f_u850 = addfile("./data/u850-197901-201412.nc", "r")
u850 = short2flt(f_u850->uwnd(rec_s:rec_e,0,{-90:90},:)) ; 850 hPa

;; v850
f_v850 = addfile("./data/v850-197901-201412.nc", "r")
v850 = short2flt(f_v850->vwnd(rec_s:rec_e,0,{-90:90},:)) ; 850 hPa

;; air2m
f_air2m = addfile("./data/air2m-197901-201412.nc", "r")
air2m = short2flt(f_air2m->air(rec_s:rec_e,0,{-90:90},:)) ; T at 2m

;;;DJF 平均 & 异常 ;;;;;;;;;;;;;;
sst_DJF = month_to_season(sst, "JFM") ;JFM 季节平均,详见附录 D 中关于
month_to_season 的说明。这里实际是 12/1/2 月三个月平均,因为本脚本已将数据
处理成 1979 年 12 月为第一个时次
copy_VarMeta(sst(0, :, :), sst_DJF(0, :, :))
sst_DJF!0 = "year"
sst_DJF&.year=year

sst_ano = dim_rmvmean_n_Wrap(sst_DJF,0)

;; h300
h300_DJF = month_to_season(h300, "JFM")
copy_VarMeta(h300(0, :, :), h300_DJF(0, :, :))
h300_DJF!0 = "year"
h300_DJF&.year=year

h300_ano = dim_rmvmean_n_Wrap(h300_DJF,0)
```

```

;; u850 与 h300 维数大小相同
u850_DJF = month_to_season(u850, "JFM")
copy_VarMeta(h300_DJF, u850_DJF)

u850_ano = dim_rmvmean_n_Wrap(u850_DJF, 0)

;; v850 与 h300 维数大小相同
v850_DJF = month_to_season(v850, "JFM")
copy_VarMeta(h300_DJF, v850_DJF)

v850_ano = dim_rmvmean_n_Wrap(v850_DJF, 0)

;; air2m
air2m_DJF = month_to_season(air2m, "JFM")
copy_VarMeta(air2m(0, :, :), air2m_DJF(0, :, :))
air2m_DJF!0 = "year"
air2m_DJF&.year = year

air2m_ano = dim_rmvmean_n_Wrap(air2m_DJF, 0)

;;; (3) enso index (5N-5S, 170-120W) ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ensoi = wgt_areaave_Wrap(sst_DJF(:, {-5:5}, {190:240}), 1., 1., 0) ; 0 表示
仅用非缺省的数值进行计算
ensoi = dim_standardize_Wrap(ensoi, 1) ; 1 表示标准化时除以 [N] ; 0 表示除以
[N-1]

;;; (4) composite ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
irec_positive = ind(ensoi.gt.0.8)
nnumb = dimsizes(irec_positive)

sst_comp = dim_avg_n_Wrap(sst_ano(irec_positive, :, :), 0)
h300_comp = dim_avg_n_Wrap(h300_ano(irec_positive, :, :), 0)
u850_comp = dim_avg_n_Wrap(u850_ano(irec_positive, :, :), 0)
v850_comp = dim_avg_n_Wrap(v850_ano(irec_positive, :, :), 0)

```

```

air2m_comp = dim_avg_n_Wrap(air2m_ano(irec_positive, :, :), 0)

;;; (5) t-test ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; sst
sst_std = dim_variance_n_Wrap(sst_ano(irec_positive, :, :), 0)
sst_std = sqrt(sst_std/numb)
sst_std = where(sst_std.eq.0, sst_std@_FillValue, sst_std)
t_sst = sst_comp/sst_std
confi_sst = sst_comp
confi_sst = student_t(t_sst, numb-1)

;; h300
h300_std = dim_variance_n_Wrap(h300_ano(irec_positive, :, :), 0)
h300_std = sqrt(h300_std/numb)
t_h300 = h300_comp/h300_std
confi_h300 = h300_comp
confi_h300 = student_t(t_h300, numb-1)

;; air2m
air2m_std = dim_variance_n_Wrap(air2m_ano(irec_positive, :, :), 0)
air2m_std = sqrt(air2m_std/numb)
t_air2m = air2m_comp/air2m_std
confi_air2m = air2m_comp
confi_air2m = student_t(t_air2m, numb-1)

;;; (5) plot
wks = gsn_open_wks("eps", "plot-comp-enso")
gsn_define_colormap(wks, "rainbow+gray") ; 调用 rainbow+gray 色板

base = new(3, "graphic")
plot = new(3, "graphic")

res = True ; 调整地图及显著性等值线, 每个子图均需
该 res

```

res@gsnAddCyclic = True ; 添加循环点,否则会在 0 度经线左侧出现一根白条

res@gsnDraw = False

res@gsnFrame = False

res@gsnLeftString = ""

res@gsnRightString = ""

resc = res ;通过复制创建新的变量

resv = res ;

rest = res ;

res@mpFillOn = False ;地图不填色

res@mpCenterLonF = 180 ; 地图的中心经度

res@mpGeophysicalLineThicknessF = 0.5 ; 地图边界的粗细

res@pmTickMarkDisplayMode = "Always" ; 坐标标签上添加度符号

res@mpGridAndLimbOn = True ; 绘制经纬度线

res@mpGridLatSpacingF = 15 ;纬线间隔

res@mpGridLonSpacingF = 15 ;经线间隔

res@mpGridLineDashPattern = 2 ;见附录图 H.6

res@mpGridLineThicknessF = 0.5 ;经纬线粗细

res@cnFillOn = True ; 填色等值线

res@cnLinesOn = True ; 绘制等值线

res@cnLineColor = "white" ; 颜色

res@cnLineThicknessF = 0.3 ; 粗细

res@cnLineLabelsOn = False ; 关闭标签

res@cnLevelSelectionMode = "ExplicitLevels"

res@cnLevels = (/0.01,0.05/)

res@cnFillPalette = "GMT_gray" ; 用 GMT_gray 进行填色。即同时使用第 2 种色板

res@cnFillColors = (/5,7,-1/) ; -1 为透明

res@cnInfoLabelOn = False ; 关闭图右下方的等值线信息标签

res@lbLabelBarOn = False ; 关闭色标

```
resc@cnLevelSelectionMode = "ExplicitLevels" ; 指定每根需绘制的等值线
resc@cnLevels              = (/ -0.75, -0.25, 0.25, 0.75, 1.25/) ;
resc@cnFillOn              = False           ; 关闭等值线填色
resc@cnLineThicknessF      = 2.             ; 等值线粗细
resc@gsnContourZeroLineThicknessF = 0. ; 不画 0 值线
resc@cnLineLabelsOn        = False          ; 关闭等值线数值标签
resc@cnLineDashPattern     = 16             ; 见附录图 H.6
resc@cnInfoLabelOn         = True           ; 打开图右下方的等值线信息标签
resc@cnInfoLabelOrthogonalPosF = 0.05      ; 移动等值线信息标签的位置

resv@vcPositionMode = "ArrowTail" ; 格点的位置对应着箭头尾部
resv@vcGlyphStyle   = "Fillarrow" ; 其余三种选项为" LineArrow", "Wind-
Barb", "CurlyVector"

resv@vcFillArrowEdgeThicknessF = 2 ; 箭头边界粗细
resv@vcFillArrowEdgeColor      = "white" ; 箭头边界颜色
resv@vcFillArrowFillColor     = "black" ; 箭头内部填充颜色
resv@vcFillArrowWidthF        = 0.1 ; 见附录图 H.1
resv@vcFillArrowHeadXF        = 0.6 ; 同上
resv@vcFillArrowHeadYF        = 0.2 ; 同上
resv@vcFillArrowHeadInteriorXF = 0.25 ; 同上

resv@vcMinDistanceF          = 0.03 ; 箭头之间的最小距离(在单位
坐标系中)
resv@vcMinMagnitudeF         = 1.0 ; 不绘制小于该数值的箭头

;**** 设定参考箭头 ****
resv@vcRefAnnoOn             = True
resv@vcRefMagnitudeF          = refmag ;单位长度的箭头所表示的实际物理量
的大小
resv@vcRefLengthF            = 0.045 ;单位长度的箭头在图形中的长度
resv@vcRefAnnoBackgroundColor = "white" ;背景颜色
resv@vcRefAnnoPerimOn        = False ;关闭边框

resv@vcRefAnnoFontHeightF     = 0.015 ;参考箭头标签字体大小
```

```
resv@vcRefAnnoString1On      = False      ;不绘制参考箭头上方的字符
resv@vcRefAnnoString2On      = True        ;绘制参考箭头下方的字符
resv@vcRefAnnoString2        = refmag+" m/s"
```

```
resv@vcRefAnnoSide            = "Top"       ;参考箭头放至图形上方
resv@vcRefAnnoOrthogonalPosF  = -0.12      ;调整其垂直位置
resv@vcRefAnnoParallelPosF    = 0.95       ;调整其水平位置
```

```
res@gsnCenterString           = "sst"       ;子图的主标题
res@gsnCenterStringFontHeightF = 0.03      ;标题字体的大小。由于后面没有
修改该值,所以每幅子图的主标题字体均是此大小
```

```
base(0) = gsn_csm_contour_map(wks,confi_sst,res) ;只有底图才可有地图(map)
plot(0) = gsn_csm_contour(wks,sst_comp, resc)    ;叠加的图形中不可有地图。
注意,所调用的绘图函数中并无“map”
```

```
plot(0) = ColorNegDashZeroPosContour(plot(0),"blue","white","red");负值
用蓝色虚线表示,0 线用白色实线,正值用红色实线
```

```
overlay(base(0),plot(0))      ;绘制地图的图形必须放在最下图层
```

```
; 绘制多边形及折线以标明 nino 3.4 区
```

```
plres                                = True
plres@gsLineColor                    = "black"
plres@gsLineThicknessF               = 1.0
```

```
gres                                = True
gres@gsFillColor                     = "yellow"
gres@gsFillOpacityF                  = 0.5
gres@gsLineColor                     = "black"
```

```
latx = (/ -5, 5, 5, -5, -5/) ; nino3.4 区的经纬度坐标位置
lonx = (/ 190, 190, 240, 240, 190/) ;
```

```
dum1 = gsn_add_polyline(wks, base(0), lonx, latx, plres)
dum2 = gsn_add_polygon(wks, base(0), lonx, latx, gres)
```

```

res@gsnCenterString      = "h300&V850"
resc@cnLevelSelectionMode = "AutomaticLevels"
resc@cnLevelSpacingF      = 15.
base(1) = gsn_csm_contour_map(wks, confi_h300, res)
plot(1) = gsn_csm_contour(wks, h300_comp, resc)
plot(1) = ColorNegDashZeroPosContour(plot(1), "blue", "white", "red")
overlay(base(1), plot(1))

plotv   = gsn_csm_vector(wks, u850_comp, v850_comp, resv)
overlay(base(1), plotv) ; 也可用 gsn_csm_vector_map(wks, h300_comp, u850,
v850, res_new)

res@gsnCenterString      = "air2m"
resc@cnLevelSelectionMode = "ManualLevels"
resc@cnMaxLevelValF      = 2
resc@cnMinLevelValF      = -2
resc@cnLevelSpacingF      = 0.5
base(2) = gsn_csm_contour_map(wks, confi_air2m, res)
plot(2) = gsn_csm_contour(wks, air2m_comp, resc)
plot(2) = ColorNegDashZeroPosContour(plot(2), "blue", "black", "red")
overlay(base(2), plot(2))

resP = True ; 绘制组图
resP@txString      = "El nino" ; 添加主标题
resP@txFontHeightF = 0.03 ; 修改其大小

resP@gsnPanelFigureStrings = ("/(a)", "(b)", "(c)"/) ; 各个子图的标号
resP@gsnPanelFigureStringsFontHeightF = 0.015 ; 字体的大小
resP@amJust = "TopLeft" ; 摆放的位置, 默认是
“BottomRight”

resP@gsnPanelRowSpec = True ; 指定每行绘制的子图的个数
gsn_panel(wks, base, (/1, 2/), resP) ; 第1行绘制1幅, 第2行绘制2幅
end

```

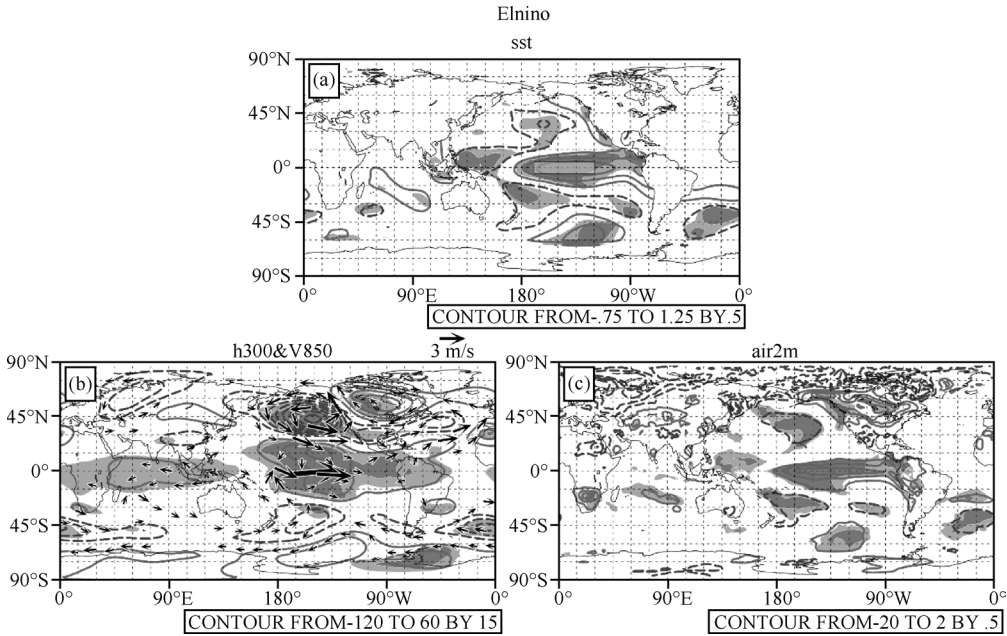


图 4.1 厄尔尼诺年环流合成图(附彩图 4.1)

(a) sst, (b) h300 (等值线)与 V850(箭头), (c) air2m;图中深(浅)阴影表示通过置信度为 0.01(0.05)显著性检验的区域;红色实线(蓝色虚线)表示正(负)异常

4.4 思考题

(1)修改 `plot-comp-enso.ncl` 脚本,绘制冬季(DJF)厄尔尼诺典型年份中各变量在随后的春季(MAM)的合成图。

(2)修改 `plot-comp-enso.ncl` 脚本,绘制 La Nina 年的合成图。将三幅子图按 3 行 1 列排列。

第5章 北极涛动的统计特征分析

——EOF分析、回归分析、相关分析、 显著性检验

5.1 实习目的

- (1)掌握 Empirical Orthogonal Function(EOF)分解的基本方法。
- (2)掌握生成 NetCDF 文件的基本方法。
- (3)掌握回归分析及其检验方法。
- (4)掌握相关分析及其检验方法。
- (5)掌握垂直剖面图的绘制方法。
- (6)掌握散点图的绘制方法。

5.2 实习内容及要求

首先下载附录B中“实习资料下载地址”中所有以“.nc”为结尾的数据资料,以及 plot-reg-AO.nc, plot-xy-PC1.nc, plot-corr-AO.nc 和 plot-scatter-AO.nc 脚本。再根据以下实习内容与要求,补充完善相应脚本:

(1)EOF分析及回归分析。补充 plot-reg-AO.nc 脚本,对1979年至2012年每年1月20°N以北的海平面气压(SLP)场(mslp.mon.mean.r2.nc)进行EOF分解,第1模态即为北极涛动(Arctic Oscillation, AO, Thompson and Wallace, 1998)。对该第一模态进行North检验(North et al., 1982),如果它能显著地与其余模态相分离,则在图的正上方写上“EOF1 is significant”。EOF分解时,需以各格点的纬度余弦值为权重。对EOF1的时间序列进行标准化,并输出为NetCDF数据。绘制SLP场对EOF1时间序列的回归图,并进行t检验。北半球极射赤道投影,地图最下点经度为180,纬度为10°N。

(2)滑动平均。补充 plot-xy-PC1.nc 脚本,绘制 EOF1 时间序列及其 9 年滑动平均值。大于 0 值用灰色填色。

(3)相关分析及显著性检验。补充 plot-corr-AO.nc 脚本,计算 EOF1 时间序列与各层纬向平均气温(air.mon.mean.nc)的相关关系。对 EOF1 时间序列与气

温场相关关系进行显著性检验。通过 95% 显著性检验的地区用灰色阴影叠加。

(4) 面积平均及散点图。补充 `plot-scatter-AO.ncl` 脚本, 对 300 hPa 上 65°N 以北地区以格点面积为权重的区域平均位势高度场 (`h300-197901-201412.nc`) 定义极涡强度指数。以 EOF1 时间序列即 AO 指数为 X 轴, 以极涡强度指数为 Y 轴, 用散点图表示 1979 年至 2010 年每年 1 月情况。添加两者的线性拟合直线, 标题上显示出两者的相关系数。

5.3 脚本讲解及图形示例

此节给出 5.2 节中所有的完整脚本, 其中粗体为关键语句。网上提供的原脚本中已删除这些关键语句, 读者可自行参考对照。

5.3.1 海平面气压场对 EOF1 时间序列的回归

```
plot-reg-AO.ncl(图 5.1):
begin
    ;;;; read SLP data
    f = addfile("./data/mslp.mon.mean.r2.nc", "r")
    slp = short2flt(f->mslp(:,:,12,{0:90},:)) ; 读取每年 1 月北半球数据

    dims = dimsizes(slp) ; 返回各维的大小

    year = ispan(1979, 1979+dims(0)-1, 1) ; 要分析的年份

    ;; 计算权重
    w = sqrt(cos(0.01745329 * slp&.lat) ) ; 权重系数
    wp = slp * conform(slp, w, 1)
    copy_VarCoords(slp, wp)

    ;; EOF 分解
    x = wp({lat|20:90}, lon|:, time|:) ; 时间维放最右边
    neof = 2 ; 保留前 2 模态, 用于检验第 1 模态是否显著与其他模态相分离
    eof = eofunc_Wrap(x, neof, False) ; EOF 分解

    ;; north 检验
```

```

ntim = dims(0)
sig_pcv = eofunc_north(eof@pcvar, ntim, False); 详见附录 D 函数 eofunc_north

;; 计算其时间序列,并标准化
eof_ts = eofunc_ts_Wrap(x, eof, False)
eof_ts = dim_standardize_n(eof_ts, 1, 1)

;; 输出至 NetCDF 文件
path_out = "SLP-PC1-jan-"+dims(0)+"yr.nc"
system("rm -f "+ path_out)      ; 若当前路径下有同名文件,则删除
ncdf = addfile(path_out, "c")    ; "c"表示创建 NetCDF 文件

eof_ts!1 = "year" ; 手动修改元数据
eof_ts&year = ispan(1979,2012,1)
ncdf->AOI = eof_ts(0,:) ; 写入变量数值,并将该输出的变量命名为
"AOI"。这里仅存储 PC1

;; 或输出至 binary 文件
fbindirwrite("SLP-PC1-jan-"+dims(0)+"yr.grd", eof_ts(0,:))

;; 回归海平面气压场
xx = regCoef(eof_ts(0,:), slp({lat|:},{lon|:},{time|:})); 注意,海平面气压场
最右边维需为 time

;; t-test
rc_slp = slp(0,:,:) ; 创建新的变量
rc_slp = xx/100 ; 转变为 hPa
confi_rc_slp = rc_slp

P = student_t(xx@tval, xx@nptxy-2)
confi_rc_slp = onedtond(P, (/dims(1), dims(2)/))

; 如下处理的目的是方便绘图(不是必须)
confi_rc_slp = where(confi_rc_slp.eq.0, confi_rc_slp@_FillValue, confi_rc_slp)

```

```

confi_rc_slp = 1/confi_rc_slp
confi_rc_slp = where(rc_slp.lt.0, -1 * confi_rc_slp, confi_rc_slp)

;;;;;;;;;plot;;;;;;;;;
wks = gsn_open_wks("eps", "plot-reg-SLP")
gsn_define_colormap(wks, "rainbow")

```

```

res                                = True
res@gsnAddCyclic                  = True
res@gsnDraw                        = False
res@gsnFrame                      = False
res@gsnLeftString                 = ""
res@gsnRightString                = ""
if(sig_pcv)then
    res@gsnCenterString    = "EOF1 is significant"
end if

```

res@gsnTickMarksOn = False ; 关闭经度标签。虽然默认是绘制经度标签,但由于其经度单位前没有“度”符号,即“°”,因此,关闭其经度标签。若需绘制标准的经度标签,可利用函数 gsn_add_text(详见附录 E 节)以及文本函数符号“~”(附录图 H. 5)进行手动添加

```

resc = res                        ;创建 resc,用以绘制回归场的等值线

```

```

res@gsnPolar                    = "NH"                ;北半球的极射赤面投影

```

```

if(sig_pcv(0))then
    res@gsnCenterString    = "EOF1 is significant"
    res@gsnCenterStringFontHeightF = 0.03
    res@gsnCenterStringOrthogonalPosF = 0.0
else
    res@gsnCenterString    = "EOF1 is not significant"
end if

```

```

res@mpFillOn                    = False

```

```
res@mpMinLatF      = 10.      ; 地图的最南纬度
res@mpMaxLatF      = 90       ; 地图的最北纬度
res@mpCenterLonF   = 180      ; 地图的中心经度
```

```
res@cnLevelSelectionMode = "ExplicitLevels"
res@cnLevels              = (/ -20, 20/) ; -20 与 20 均对应着 0.05 置信
度,但前者对应负值异常,后者对应正值异常
```

```
res@cnFillColors        = (/ "gray", "white", "gray" /)
res@cnFillOn             = True
res@lbLabelBarOn        = False ; 在表示显著性时,一般不需要绘制等值
线线条及其数值,因此关闭以下几项
```

```
res@cnLinesOn           = False
res@cnInfoLabelOn       = False
res@cnLineLabelsOn      = False
```

```
base = gsn_csm_contour_map_polar(wks, confi_rc_slp, res)
```

;; 回归场等值线的设置

```
resc@cnLevelSelectionMode = "ExplicitLevels"
resc@cnLevels              = 1. * ispan(-10, 10, 1)
resc@cnFillOn              = False
resc@cnInfoLabelOn         = False
resc@gsnContourZeroLineThicknessF = 0.
resc@cnLineThicknessF      = 2.
resc@cnLineLabelsOn        = False
plot = gsn_csm_contour(wks, rc_slp, resc)
plot = ColorNegDashZeroPosContour(plot, "black", "black", "black")
```

```
overlay(base, plot) ; 图层叠加
```

;; 添加额外的等值线

```
resc2 = resc
resc2@cnLevels              := (/ -4, 2/) ; 绘制-4 与 2 两个额外的等值线
resc2@cnLineThicknessF      = 7.
```

```

resc2@cnMonoLineDashPattern = False
resc2@cnLineDashPatterns     = (/1,0/) ;两根等值线用不同的线型;也可通过调用函数 ColorNegDashZeroPosContour 实现
resc2@cnLineDashSegLenF      = 0.25 ;虚线上每个线段的长度,线段之间的间隔会自动调整

plot2 = gsn_csm_contour(wks,rc_slp,resc2)

overlay(base,plot2)

draw(base)
frame(wks)
end

```

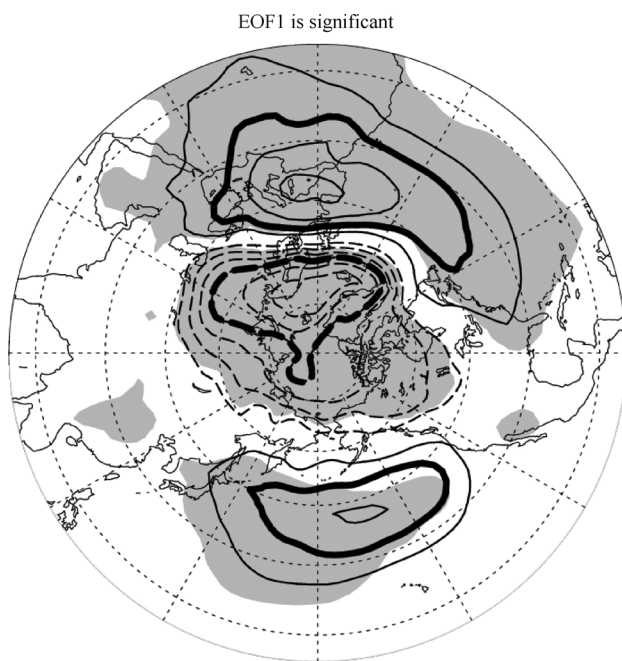


图 5.1 1 月海平面气压场对 EOF1 时间序列的回归场(等值线,单位:hPa)

(细实(细虚)线表示正(负)值,0 线已略去;阴影表示通过 95%显著性检验的区域;粗实(粗虚)线表示 2 (-4) hPa 等值线)

5.3.2 绘制 PC1 及其 9 年滑动平均

```
plot-xy-PC1.ncl(图 5.2):
begin
ncdf = addfile("SLP-PC1-jan-34yr.nc","r")
AOI = ncdf->AOI
```

AOI_mean = runave_Wrap(AOI,9,0) ; 9 表示 9 点滑动平均,本例对应着 9 年滑动平均;0 表示将最前与最后 4 年设为缺省。详见附录 D

```
;;;;;;;;;plot;;;;;;;;;
wks = gsn_open_wks("eps","plot-xy-AOI")
gsn_define_colormap(wks,"rainbow")
```

```
res          = True
res@gsnDraw  = False
res@gsnFrame = False
```

```
res2 = res
```

```
res@vpHeightF = 0.35 ; 改变视图的大小
res@vpWidthF  = 0.6  ;
```

```
res@trXMinF = 1979 ; 设定 X 轴坐标范围
res@trXMaxF = 2012
res@trYMaxF  = 3    ; 设定 Y 轴坐标范围
res@trYMinF  = -3
```

;设定大于与小于 0 时的填色

```
res@gsnYRefLine      = 0.          ; 设定参考值
res@gsnAboveYRefLineColor = "light gray" ; 大于该参考值时所用的颜色
res@gsnBelowYRefLineColor = "white"   ; 小于该参考值时所用的颜色
```

```
res@tiXAxisString= "year" ; 添加 X 轴名称
```

```

res@tiYAxisString= "AOI"      ; 添加 Y 轴名称

plot = gsn_csm_xy(wks,AOI&.year,AOI,res)

res2@xyLineColor      = "black"
res2@xyLineThicknessF = 4.
plot_mean = gsn_csm_xy(wks,AOI&.year,AOI_mean,res2)
overlay(plot,plot_mean)

draw(plot)
frame(wks)
end

```

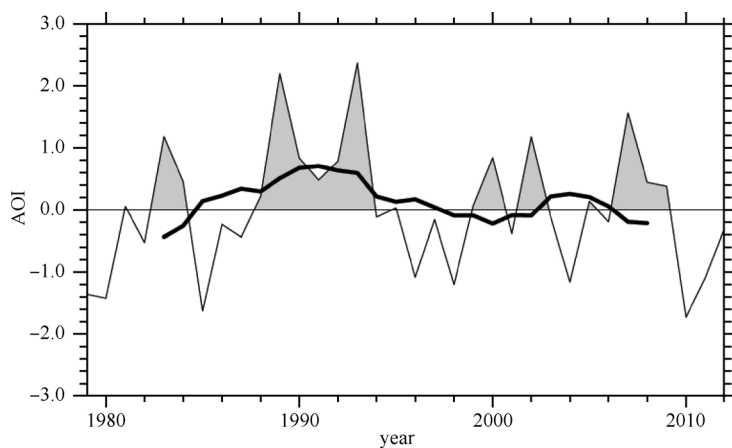


图 5.2 PC1(填色)及其 9 年滑动平均(折线)
(阴影表示数值大于 0)

5.3.3 各层纬向平均气温场与 PC1 的相关系数

```

plot-corr-AO.ncl(图 5.3):
begin
    itimes = 197901 ;起始年月
    itimee = 201001 ;结束年月

    siglvl = 0.05    ;置信度

```

```

;;;read air data
f    = addfile("./data/air.mon.mean.nc", "r")
var  := short2flt(f->air)
time  := f->time
YYYYMM := cd_calendar(time, -1)

ist = ind(itimes.eq.YYYYMM)
ied = ind(itimee.eq.YYYYMM)

air = var(ist:ied:12, :, :, :) ; 每读取第 12 个时次,即挑出每年的 1 月份

air_zm = dim_avg_Wrap(air) ;对数组最右边维平均,保留元数据。这里即是纬
向平均
dims    = dimsizes(air_zm)

;;;read AOI data
ncfile = addfile("SLP-PC1-jan-34yr.nc", "r")
eof_ts = ncfile->AOI
AOI = eof_ts({1979:2010}) ;截取部分时间段,以保证与变量 air 一致

;;相关并检验
mxlag = 0 ;同时相关
corr = esccr(AOI, air_zm(level|:, lat|:, time|:), mxlag) ;输出场为[17]×[73]×[1]

var := (/corr(:, :, 0)/)
copy_VarMeta(air_zm(0, :, :), var)

prob = rtest(var, dims(0), 0)
copy_VarMeta(air_zm(0, :, :), prob)

;; ** plot *****
wks    = gsn_open_wks ("eps", "plot-corr-AO")
gsn_define_colormap(wks, "BlWhRe")

```



```

res                = True
res@gsnDraw        = False
res@gsnFrame       = False
res@gsnLeftString  = ""
res@gsnRightString = ""

```

```

res@tmXBMode = "Explicit"
res@tmXBValues = (/ -90, -60, -30, 0, 30, 60, 90/) ; 指定其纬度
res@tmXBLabels = (/ "90~S~o~N~S", "60~S~o~N~S", "30~S~o~N~S", "0", "30~S~o~N~N", "60~S~o~N~N", "90~S~o~N~N"/) ; 在对应位置上添加带圈号的纬度。这里使用了默认的字符函数码 "~"。见附录图 H. 5

```

```

res@tmXBLabelDeltaF = -0.6 ; 移动其位置,使其更加靠近 X 轴
res@tmXBLabelFontHeightF = 0.015
res@tmYLLLabelFontHeightF = 0.015

```

```

res@tiYAxisString = "Level (hPa)"

```

```

res@cnFillOn = True
res@cnLinesOn = False
res@cnLineLabelsOn = False
res@cnLevelSpacingF = 0.05
res@gsnContourZeroLineThicknessF = 0.
res@cnInfoLabelOn = False

```

```

res@lbLabelBarOn = False

```

```

base = gsn_csm_pres_hgt(wks, prob, res)

```

```

opt = True

```

```

opt@gsnShadeFillType = "color" ; 默认设置

```

```

opt@gsnShadeLow = "gray"

```

base = gsn_contour_shade(base, siglvl, -999., opt) ; 从小于等于 siglvl 的第一个等值线开始用“gray”填色

```

res@cnFillOn          = False
res@cnLinesOn         = True
res@cnLevelSpacingF   = 0.15
res@cnLineDashPattern = 2      ;见附录图 H.6
res@cnLineThicknessF  = 3.
res@cnInfoLabelOn     = True

plot = gsn_csm_pres_hgt(wks, var(level|:, lat|:), res)
plot = ColorNegDashZeroPosContour(plot, "black", "white", "black")
overlay(base, plot)

draw(base)
frame(wks)
end

```

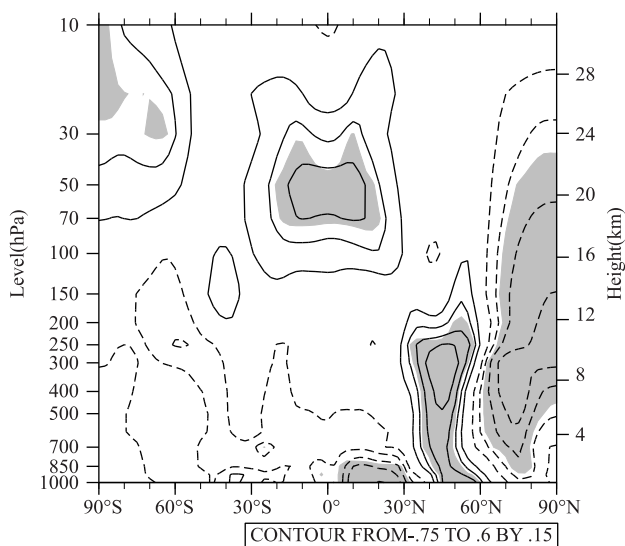


图 5.3 1月纬向平均气温与 AO 的相关系数
(实线(虚线)表示正(负)相关系数,阴影区为通过 95%显著性检验的区域)

5.3.4 AO 指数与极涡指数的关系

plot—scatter—AO.ncl(图 5.4):
begin

```

itimes = 197901
itimee = 201001

siglvl = 0.01

nyear = (itimee - itimes)/100+1

xaxis = fspan(-2,2,nyear) ; X 轴与 Y 轴的数值范围
yaxis = xaxis

;;;read air data
f = addfile("./data/h300-197901-201412.nc", "r")
var := short2flt(f->hgt(:,0,{65:90}),:))
time:= f->time
YYYYMM := cd_calendar(time,-1)

ist = ind(itimes.eq.YYYYMM)
ied = ind(itimee.eq.YYYYMM)

hgt = var(ist:ied:12,:,:)

rad = 3.1415926/180 ;4.0 * atan(1.0)/180.0
lat = hgt&.lat
clat = cos(lat * rad)

vor = hgt(:,0,0)
vor = wgt_areaave(hgt(:,{65:90}),:),clat,1.0,0)
vor = dim_standardize(vor,0)

;;;read AOI data
ncfile = addfile("SLP-PC1-jan-34yr.nc","r")
eof_ts = ncfile->AOI
AOI = eof_ts({1979:2010}) ; 截取部分时间段,以保证与变量 air 一致

```

;;直线拟合 $y = mx + b$, 其中 m 是斜率, b 是截距。 m 与 b 均可从函数 `regline` 的返回值中获得

```
m = regline(AOI,vor)
```

```
data = m * fspan(-2,2,32) + m@yave ; b= m@yave
```

;;显著性检验

```
prob = rtest(m, nyear, 0)
```

```
;; ** plot *****
```

```
wks = gsn_open_wks("eps", "plot-scatter-AO")
```

```
res = True
```

```
res@gsnMaximize = True
```

```
res@gsnDraw = False
```

```
res@gsnFrame = False
```

`if(prob.le. siglvl)` ; 在图上方表明相关系数及是否显著

```
res@gsnCenterString = "r is " + sprintf("%5.2f",m) + "(sig. @" + siglvl  
+ ")"
```

```
else
```

```
res@gsnCenterString = "r is " + sprintf("%5.2f",m) + "(insig. @" +  
siglvl+ ")"
```

```
end if
```

```
res@trYMinF = min(yaxis)
```

```
res@trYMaxF = max(yaxis)
```

```
res@trXMinF = min(xaxis)
```

```
res@trXMaxF = max(xaxis)
```

```
res@tiXAxisString = "AO"
```

```
res@tiYAxisString = "Vor."
```

```
plot = gsn_csm_xy(wks, xaxis, data, res)
```

```

pmres = True
pmres@gsMarkerIndex    = 12 ;见附录图 H.2
pmres@gsMarkerSizeF    = 0.02
pmres@gsMarkerColor    = "black"

xl = eof_ts
yl = vor
dum  = gsn_add_polymarker(wks,plot,xl,yl,pmres)

draw(plot)
frame(wks)
end

```

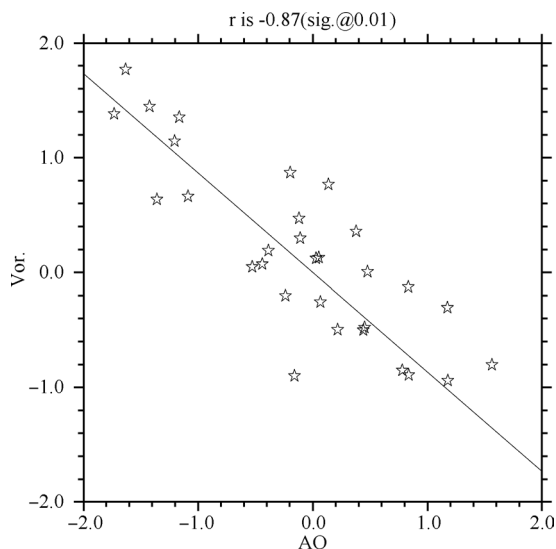


图 5.4 1 月 AO 指数与极涡指数的散点图
(直线为其回归直线)

5.4 思考题

(1) 修改 `plot-reg-AO.ncl` 脚本, 对南半球 20°S 以南地区进行 EOF 分解, 并返回第 1 和第 2 模态, 绘制 SLP 场分别对 PC1 及 PC2 的回归场。

(2) 结合 `plot-xy-PC1.ncl` 与 `plot-scatter-AO.ncl`, 绘制南半球 PC1, 并添加趋势线。

第6章 特殊图形的绘制

——省界内绘图、卫星投影、兰勃特 等角投影和小波图

6.1 实习目的

- (1)掌握在某个不规则区域(比如江苏省)中绘制变量。
- (2)掌握卫星投影图的绘制方法。
- (3)掌握兰勃特等角投影图的绘制方法。
- (4)掌握小波(wavelet)图的绘制方法。

6.2 实习内容及要求

首先下载附录B中“实习资料下载地址”中所有以“.nc”为结尾的数据资料,以及 plot-reg-AO.nc, plot-xy-PC1.nc, plot-corr-AO.nc 和 plot-scatter-AO.nc 四个脚本。再根据以下实习内容与要求,补充完善相应脚本:

(1)省界内绘图。补充 plot-precip-jiangsu.nc 脚本,根据 1979 年至 2008 年中国 160 站夏季平均降水量数据,绘制江苏省夏季气候平均降水量。其中 160 站夏季平均降水量数据可参考第 3 章内容,也可直接使用./data/下的 precip-160-JJA-30yr.grd。

(2)卫星投影,标记高低值系统中心。补充 plot-hgt-Satellite.nc 脚本,利用 NCEP/DOE AMIP-II 再分析资料 300 hPa 位势高度场 h300-197901-201412.nc,绘制 1979 年 1 月位势高度场。要求:采用卫星投影,海洋、陆地用不同颜色填充,在低压与高压中心分别标记“L”和“H”。

(3)兰勃特等角投影。同(2),但补充 plot-hgt-lambert.nc 脚本,并采用“LambertConformal”,即兰勃特等角投影。要求:低值中心标签字体用白色,其背景色用透明色,为地图添加经纬度坐标(采用调用子程序方式实现)。

(4)小波分析。补充 plot-enso-wavelet.nc 脚本,依据第 4 章中计算得到的 1979 年至 2013 年冬季 ENSO 指数(或在./data/目录下的 SLP-PC1-jan-34yr.nc),进行小波分析并检验其显著性(Torrence and Compo, 1998; Liu et al., 2007)。

要求在小波图中用方格表示有边界效应的区域,在其右侧绘制各个周期的功率大小。

6.3 脚本讲解及图形示例

此节给出 6.2 节中所有的完整脚本,其中粗体为关键语句。网上提供的原脚本中已删除这些关键语句,读者可自行参考对照。

6.3.1 台站降水资料的读取并在江苏省内绘制

plot-precip-jiangsu.ncl(图 6.1):

load "\$NCARG_ROOT/lib/ncarg/nclscripts/esmf/ESMF_regridding.ncl" ; 将台站资料插值至格点资料,必须加载该 NCL 文件

begin

regrid_On = True ; 需要插值

;!!注意,本文所用台站资料分辨率较低,若直接用该资料进行绘图,会导致江苏省边界处出现一些空白。因此,为达到较好的图形效果,该脚本首先将降水资料插值到分辨率为 0.1° 的均匀经纬度网格点上。但请读者务必注意,这种由低分辨率插值到高分辨率的结果可能存在较大误差

```
precip = fbindirread("./data/precip-160-JJA-30yr.grd", 0, (/30, 160/), "float")
```

```
precip!0 = "year"
```

```
precip!1 = "stations"
```

```
precip&.year = ispan(1979, 2008, 1)
```

```
precip&.stations = ispan(1, 160, 1)
```

```
var = dim_avg_n_Wrap(precip, 0) ; 气候平均
```

```
;; 读取各个台站的经纬度坐标
```

```
path_station = "./data/160stations.txt"
```

```
station = asciiread(path_station, (/160, 2/), "float")
```

```
lat = station(:, 0)
```

```
lon = station(:, 1)
```

;是否进行插值。这里插值的目的仅为图形美观,其数值可能存在一定误差

```

if(regrid_On)
    minlat = min(lat)
    maxlat = max(lat)
    minlon = min(lon)
    maxlon = max(lon)

    Opt                = True
    Opt@SrcGridLat     = lat
    Opt@SrcGridLon     = lon
    Opt@DstLLCorner    = (/floor(minlat)-0.1, floor(minlon)+0.1/)
    Opt@DstURCorner    = (/ceil(maxlat)-0.1, ceil(maxlon)+0.1/)
    Opt@DstGridType    = "0.1deg" ;插值后的分辨率
    Opt@ForceOverwrite = True
    Opt@Debug          = True

    Opt@InterpMethod   = "patch" ;或者 "bilinear"
    var := ESMF_regrid(var, Opt) ;由于插值后数组维数大小与原 var 不一致,
    所以需在"="前加上":",即重新定义及赋值 var
end if

wks = gsn_open_wks("eps", "plot-precip-jiangsu")
gsn_define_colormap(wks, "GMT_drywet")

res=True
res@gsnDraw      = False
res@gsnFrame     = False
res@gsnAddCyclic = False
res@gsnMaximize  = True

;; 若不插值,则直接用台站资料绘制
if(.not.regrid_On)
    res@sfXArray = lon
    res@sfYArray = lat
end if

```


;; 设置经纬线

```
res@mpGridAndLimbOn      = True
res@mpGridLineColor      = "black"
res@mpGridLatSpacingF     = 1
res@mpGridLonSpacingF     = 1
res@mpGridLineDashPattern = 16      ; 见附录图 H.6
res@mpGridLineThicknessF = 0.2
```

```
res@mpMaxLatF = 36
res@mpMinLatF = 30
res@mpMaxLonF = 123
res@mpMinLonF = 116
```

```
res@mpDataBaseVersion = "Ncarg4_1" ; 地图采用中等分辨率
res@mpDataSetName      = "Earth..4"
```

```
res@mpAreaMaskingOn      = True
res@mpMaskAreaSpecifiers = ("/China:jiangsu"/)
res@mpOceanFillColor      = 0
res@mpInlandWaterFillColor = 0
res@mpLandFillColor       = 0
```

```
res@mpOutlineOn          = True
res@mpOutlineSpecifiers   = "China:jiangsu"
res@mpOutlineBoundarySets = "NoBoundaries"
```

;; 设置坐标标签及字体大小

```
res@pmTickMarkDisplayMode = "Always" ; 坐标标签上添加度符号
res@tmXBLabelFontHeightF  = 0.018    ; YL 标签字体大小会同步变化, 所以不用设置 YL
```

;; 设置等值线

```
res@cnFillOn      = True
res@cnLineColor    = "white"
```

```
res@cnLineThicknessF = 0.5
res@lbOrientation      = "Vertical" ;垂直摆放

res@cnLevelSelectionMode = "ManualLevels"
res@cnMinLevelValF      = 500
res@cnMaxLevelValF      = 560
res@cnLevelSpacingF     = 10

res@cnFillDrawOrder      = "PreDraw"
res@cnLineDrawOrder     = "PreDraw"

plot = gsn_csm_contour_map(wks, var, res)

draw(plot)
frame(wks)
end
```

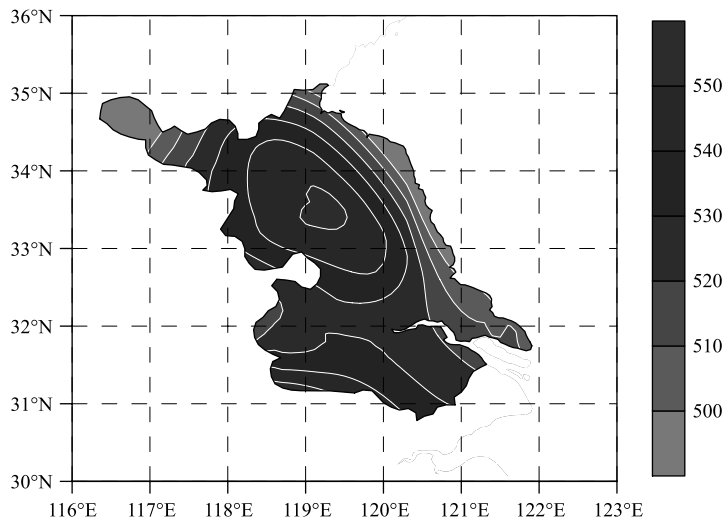


图 6.1 江苏省 1979 年至 2008 年夏季气候平均降水量(单位:mm,附彩图 6.1)

6.3.2 卫星投影图

plot—hgt—Satellite.ncl(图 6.2):

```
begin
```

```
;;;读取数据
```

```
f = addfile("./data/h300-197901-201412.nc", "r")
```

```
var := short2flt(f->hgt(0,{300},:,:))
```

```
wks = gsn_open_wks("eps","plot-hgt-lambert")
```

```
gsn_define_colormap(wks,"amwg256")
```

```
res=True
```

```
res@gsnDraw = False
```

```
res@gsnFrame = False
```

```
res@gsnAddCyclic = True
```

```
res@gsnLeftString = ""
```

```
res@gsnRightString = ""
```

```
res@mpProjection = "Satellite" ; 卫星投影
```

```
res@mpCenterLonF = 120.0 ; 图形中心经度
```

```
res@mpCenterLatF = 50. ; 图形中心纬度
```

```
res@mpFillOn = True
```

```
res@mpOceanFillColor = 108 ; 海洋颜色
```

```
res@mpInlandWaterFillColor = 100 ; 内陆湖泊颜色
```

```
res@mpLandFillColor = 147 ; 陆地颜色
```

```
res@mpOutlineOn = True ; 是否绘制陆地边界线
```

```
res@mpOutlineBoundarySets = "AllBoundaries" ; 绘制所有边界线。其他较常  
用的为"National"、"Geophysical"
```

```
;; 设置等值线
```

```
res@cnFillOn = False
```

```
res@cnInfoLabelOn = False
```

```
; 低值中心用“L”标记
```

```
res@cnLowLabelsOn = True
```

```
res@cnLowLabelFontHeightF = 0.04 ; “L”的字体大小
```

```
res@cnLowLabelBackgroundColor = -1
```

```

; 高值中心用“H”标记
res@cnHighLabelsOn          = True
res@cnHighLabelFontHeightF  = 0.04      ; “H”的字体大小
res@cnHighLabelBackgroundColor = -1

plot = gsn_csm_contour_map(wks, var, res)

draw(plot)
frame(wks)
end

```

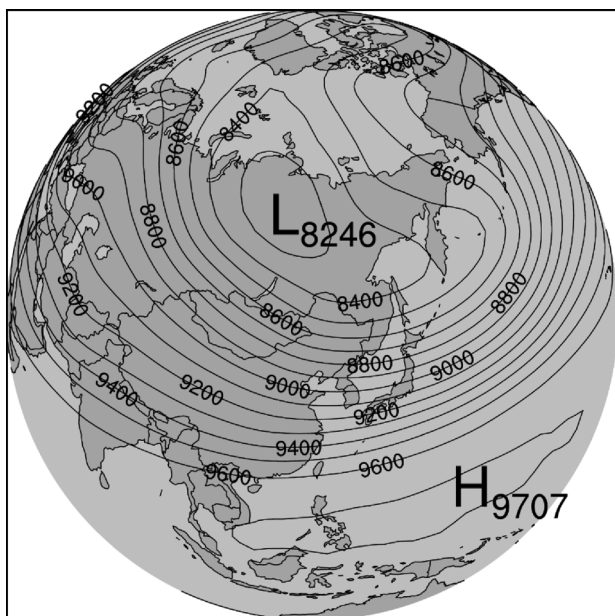


图 6.2 1979 年 1 月 300 hPa 位势高度场(单位:gpm.附彩图 6.2)

6.3.3 兰勃特等角投影图

plot=hgt—lambert.ncl(图 6.3):

; 首先自定义一个名为 add_lc_labels 的程序,用以添加经纬度标签

```

procedure add_lc_labels(wks, map, minlat, maxlat, minlon, maxlon, fontheight)
local lat_values, nlat, lat1_ndc, lat2_ndc, lon1_ndc, lon2_ndc, slope, txres, \
lon_values, PI, RAD_TO_DEG, dum_lft, dum_rgt, dum_bot

```

```

begin
  PI          = 3.14159
  RAD_TO_DEG = 180./PI

  ;挑出“较好”的纬度标签
  lat_values = ispan(toint(minlat),toint(maxlat),10) * 1.
  nlat= dimsizes(lat_values)

  ;分别计算图形左、右线条的斜率(基于标准化坐标)
  lat1_ndc = new(1,float)
  lon1_ndc = new(1,float)
  lat2_ndc = new(1,float)
  lon2_ndc = new(1,float)
  datatondc(map,minlon,lat_values(0),lon1_ndc,lat1_ndc)
  datatondc(map,minlon,lat_values(nlat-1),lon2_ndc,lat2_ndc)
  slope_lft = (lat2_ndc-lat1_ndc)/(lon2_ndc-lon1_ndc)

  datatondc(map,maxlon,lat_values(0),lon1_ndc,lat1_ndc)
  datatondc(map,maxlon,lat_values(nlat-1),lon2_ndc,lat2_ndc)
  slope_rgt = (lat2_ndc-lat1_ndc)/(lon2_ndc-lon1_ndc)

  txres          = True
  txres@txFontHeightF = fontheight
  txres@txPosXF    = 0.1

  dum_lft = new(nlat,graphic)    ; 创建变量,用以绘制经纬度字符
  dum_rgt = new(nlat,graphic)    ;
  do n=0,nlat-1
    ;添加适当空白
    lat_label_rgt = "          " + lat_values(n) + "~S~o~N~" ;见附录
图 H.5

    ;检查是否在北纬、南纬或赤道上
    if(lat_values(n).lt.0) then

```

```

lat_label_lft = lat_values(n) + "~S~o~N~S"           ";见附录图 H.5
lat_label_rgt = lat_label_rgt + "S"
end if
if(lat_values(n).gt.0) then
    lat_label_lft = lat_values(n) + "~S~o~N~N"       ";见附录图 H.5
    lat_label_rgt = lat_label_rgt + "N"
end if
if(lat_values(n).eq.0) then
    lat_label_lft = lat_values(n) + "~S~o~N~"         ";见附录图 H.5
end if

;添加左边坐标标签
txres@txAngleF = RAD_TO_DEG * atan(slope_lft) - 90 ;适当旋转字体以
更加美观
dum_lft(n) = gsn_add_text(wks, map, lat_label_lft, minlon, lat_values(n),
txres)

;添加右边坐标标签
txres@txAngleF = RAD_TO_DEG * atan(slope_rgt) + 90
dum_rgt(n) = gsn_add_text(wks, map, lat_label_rgt, maxlon, lat_values(n),
txres)
end do

;-----
;添加经度标签
delete(txres@txPosXF)
txres@txPosYF = -5.0

;挑出“较好”的经度标签
lon_values = ispan(toint(minlon+10),toint(maxlon-10),10) * 1.
nlon      = dimsizes(lon_values)

dum_bot = new(nlon,graphic)

```

```

do n=0, nlon-1
    ; 对于每个经度标签, 计算其需旋转的角度, 以使字体更加美观
    datatondc(map, lon_values(n)-0.25, minlat, lon1_ndc, lat1_ndc)
    datatondc(map, lon_values(n)+0.25, minlat, lon2_ndc, lat2_ndc)
    slope_bot = (lat1_ndc-lat2_ndc)/(lon1_ndc-lon2_ndc)
    txres@txAngleF = atan(slope_bot) * RAD_TO_DEG

    ; 针对不同的东经、西经或 0 度绘制不同的字符
    lon_label_bot = " ~C~ ~C~" + abs(lon_values(n)) + "~S~o~N~"
; 见附录图 H.5
    if(lon_values(n).lt.0) then
        lon_label_bot = lon_label_bot + "W"
    end if
    if(lon_values(n).gt.0) then
        lon_label_bot = lon_label_bot + "E"
    end if

    dum_bot(n) = gsn_add_text(wks, map, lon_label_bot, lon_values(n), minlat,
txres)
end do
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
begin
    ; 绘制的空间范围
    minlat = 20.
    maxlat = 70.
    minlon = 60.
    maxlon = 140.
    fontheight = 0.012 ; 设置坐标标签字体大小

    ;;; 读取数据
    f = addfile("./data/h300-197901-201412.nc", "r")
    var := short2flt(f->hgt(0, {300}, {minlat:maxlat}, {minlon:maxlon}))

```

```
wks = gsn_open_wks("eps", "plot-hgt-lambert")
gsn_define_colormap(wks, "rainbow")
```

```
res=True
res@gsnDraw      = False
res@gsnFrame     = False
res@gsnAddCyclic = False
```

```
res@gsnLeftString = ""
res@gsnRightString = ""
```

```
res@gsnMaskLambertConformal = True
```

```
res@mpProjection      = "LambertConformal" ; 投影类型
```

```
res@mpMinLatF      = minlat
res@mpMaxLatF      = maxlat
res@mpMinLonF      = minlon
res@mpMaxLonF      = maxlon
```

```
res@mpGridAndLimbOn      = True
res@mpGridLatSpacingF    = 10 ; 注意,由于图形中的经纬度线是由 NCL
自动绘制,因此,可能会出现经纬度线与用户指定的经纬度位置不一致。要解决该问
题,需用户调整子程序或图形经纬度范围
```

```
res@mpGridLonSpacingF    = 10
res@mpGridLineDashPattern = 1
res@mpGridLineColor     = "white"
```

```
;; 设置等值线
```

```
res@cnFillOn = True
res@cnLineLabelsOn = True
res@cnLineLabelFontHeightF = 0.015
res@lbLabelFontHeightF     = 0.015
```

```
; 低值中心用“L”标记
```

```
res@cnLowLabelsOn      = True
```



```
res@cnLowLabelFontColor      = "white"
res@cnLowLabelBackgroundColor = "transparent"
res@cnHighLabelFontHeightF   = 0.03 ; 低值中心标签字体大小只能通过
cnHighLabelFontHeightF 设定,cnLowLabelFontHeightF 无效,希望 NCL 后续版本能
够修正此缺陷
```

```
plot = gsn_csm_contour_map(wks, var, res)
add_lc_labels(wks, plot, minlat, maxlat, minlon, maxlon, fontheight)

draw(plot)
frame(wks)
end
```

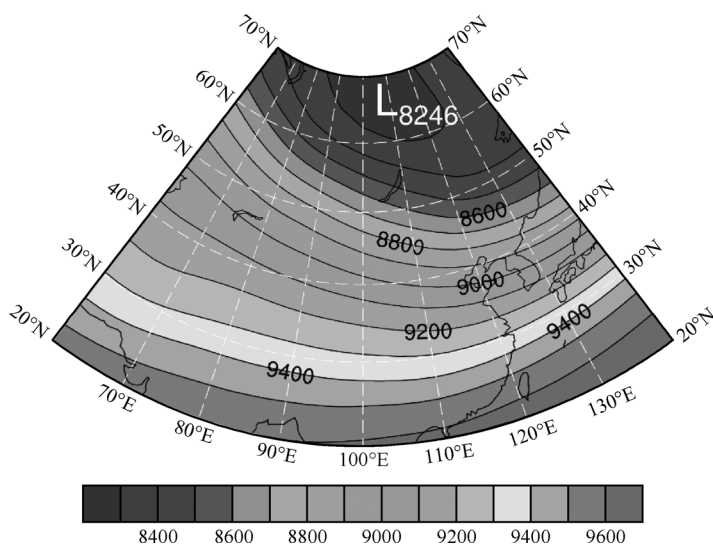


图 6.3 1979 年 1 月 300 hPa 位势高度场(单位:gpm,附彩图 6.3)

6.3.4 小波图

```
plot—enso—wavelet.ncl(图 6.4):
begin
f      = addfile("./data/ENSO—index.nc", "r")
ensoi  = f—>ensoi
```

```

time = ensoi&.year
N     = dimsizes(ensoi)

;; 小波计算
mother = 0      ; 母小波类型,通常为 0,即“Morlet”小波
dt      = 1      ; 数组中数值之间的时间间隔,通常为 1。本例中表示间隔 1 年
param   = -1     ; 母小波参数。如果 param < 0,则使用默认数值,即采用“Mor-
let”小波时为 6
s0      = dt     ; “Morlet”小波 s0 = dt
dj      = 0.25   ; 常用设定
jtot    = 1+floattointeger(((log10(N * dt/s0))/dj)/log10(2.)) ; 常用设定
npad    = N      ; 常用设定
nadof   = 0      ; 常用设定
noise   = 1      ; 常用设定,红噪声检验
siglvl  = .05     ; 置信度水平
isigtest = 0     ; 采用 chi-square 检验

w = wavelet(ensoi,mother,dt,param,s0,dj,jtot,npad,noise,isigtest,siglvl,nadof)

; *****
power      = onedtond(w@power, (/jtot,N/)) ; 功率谱
power!0    = "period"                      ; Y axis
power&.period = w@period

power!1    = "time"                        ; X axis
power&.time  = time

power@long_name = "Power Spectrum"
power@units     = "1/unit-freq"

;计算显著性 ( >= 1 则显著)
SIG          = power                      ; 复制元数据
SIG          = power/conform (power,w@signif,0)
SIG@long_name = "Significance"

```

```
SIG@units      = " "
```

```
; *****
```

```
wks = gsn_open_wks("eps", "plot—enso—wavelet")
gsn_define_colormap(wks, "BlAqGrYeOrReVi200")
```

```
YLValues = (/1,2,4,8,16/)
YLLabels = (/ "1", "2", "4", "8", "16" /)
```

```
res              = True
res@gsnDraw      = False
res@gsnFrame     = False
res@gsnRightString = " "
res@gsnLeftString  = " "
```

```
res@trYReverse   = True                ; 倒置 y—axis
res@tmYLMODE     = "Explicit"
res@tmYLValues   = YLValues
res@tmYLLabels   = YLLabels
res@tmLabelAutoStride = True
```

```
res@trYMaxF      = max(YLValues)
; res@trYMinF     = min(YLValues)
```

```
res@cnLinesOn    = False
res@cnLineLabelsOn = False
res@cnInfoLabelOn = False
```

```
res2 = res
```

```
res@tiXAxisString    = "Year"
res@tiXAxisOffsetYF  = 0.135
res@tiYAxisString    = "Years"
```

```
res@cnFillOn        = True
```

```

res@cnFillMode          = "RasterFill"
res@cnRasterSmoothingOn = True

;;;;;;;;;;;;;
res2@cnLevelSelectionMode = "ManualLevels"
res2@cnMinLevelValF       = 0.00
res2@cnMaxLevelValF       = 2.00
res2@cnLevelSpacingF      = 1.00
res2@cnFillScaleF         = 0.5      ; 增加形状填充的密度(通过调用绘图函数
数 gsn_contour_shade 实现形状填充)

plot = gsn_csm_contour(wks, power, res)
iplot = gsn_csm_contour(wks, SIG, res2)

opt = True
opt@gsnShadeFillType = "pattern" ;改为形状填充
opt@gsnShadeHigh     = 17         ;见附录图 H.3
iplot = gsn_contour_shade(iplot, -999., 1., opt) ;从大于等于 1. 的第一个等
值线开始用形状 17 填充
overlay(plot, iplot) ; 在原图上添加显著性

plot = ShadeCOI(wks, plot, w, time, False) ;用网格表示有边界效应的区域

;;添加各周期的功率
gws = w@gws
resl = True
resl@gsnFrame      = False
resl@gsnDraw       = False
resl@trYAxisType   = "LogAxis"
resl@trYReverse     = True
resl@tmYLMMode     = "Explicit"
resl@tmYLValues     = YLValues
resl@tmYLLabels     = YLLabels
resl@trYMaxF        = max(YLValues)
resl@trYMinF        = min(YLValues)

```

```

resl@tiXAxisString = "Global Wavelet Power"

plotg = gsn_csm_xy(wks,gws,power&.period,resl)

;; 将 plotg 添加至 plot 的右侧
plotc = gsn_attach_plots(plot,plotg,res,resl)

draw(plot)
frame(wks)
end

```

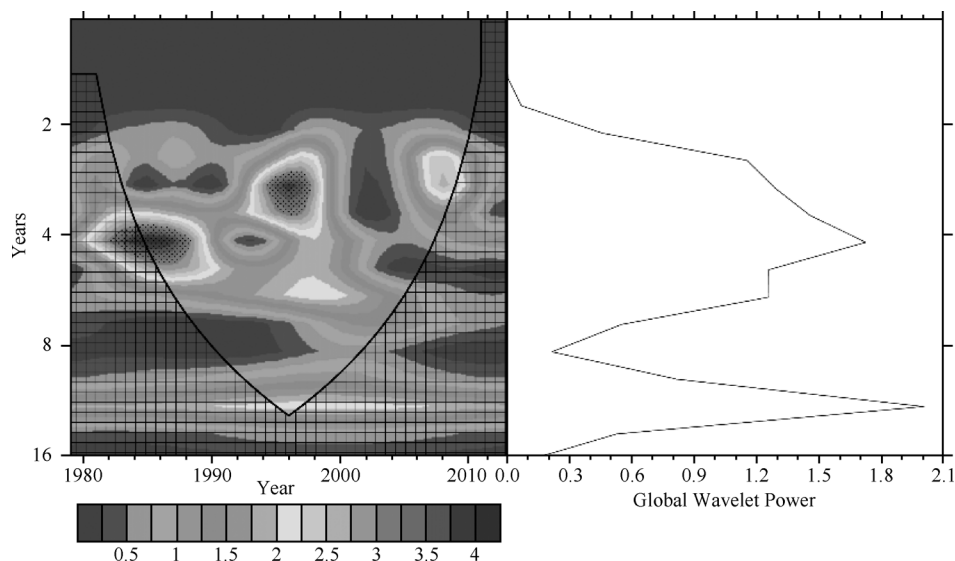


图 6.4 1 月 ENSO 指数 Morlet 小波图(附彩图 6.4)

(左图中网格点处为通过 95% 显著性检验区域, 网格处为边界效应, 右图为功率谱)

6.4 思考题

- (1) 修改 `plot-precip-jiangsu.ncl` 脚本, 绘制安徽省 1998 年夏季异常降水量。
- (2) 修改 `plot-hgt-Satellite.ncl` 脚本, 绘制 1980 年 1 月 `hgt` 异常场的卫星投影图, 同时利用 `data` 中的地表高度资料 `hgt.sfc.nc`, 在图中添加地形高度。
- (3) 修改 `plot-enso-wavelet.ncl` 脚本, 绘制 1979 年至 2014 年 1 月 AO 时间序列的小波图。

第 7 章 天气研究和预报模式输出数据制图

7.1 实习目的

(1)掌握 NCL 中天气研究和预报模式(Weather Research and Forecasting Model, WRF; Skamarock et al., 2008)的扩展函数库,采用该函数库对 WRF 模式输出数据制图。

(2)掌握 WRF 模式的模拟区域及嵌套区域显示。

(3)掌握使用 WRF 模式基础制图类型,并完成绘制剖面图、风玫瑰图、组图等。

(4)掌握利用 gsn 库函数进行 WRF 模式输出数据制图。

7.2 实习内容及要求

首先下载附录 B 中“实习资料地址”中所有以“wrf_”为开头的 NetCDF 格式数据资料,以及以“WRF_”开头的 ncl 脚本。再根据以下实习内容与要求,补充完善相应脚本。

(1)WPS 中模拟区域及嵌套区域显示。要求能利用脚本显示区域设置及嵌套配置。

(2)掌握绘制 WPS 静态数据的方法,利用 WRF 扩展函数库进行多个静态数据的制图。

(3)掌握 WRF 扩展函数中基础的制图技术(等值线、图层叠加等)。

(4)掌握 WRF 中剖面图函数的基本设置,能利用此函数绘制纬线、经线或指定两点之间的剖面。

(5)掌握风玫瑰图绘制方法。

(6)掌握 WRF 扩展库 overlay 的基本方法及 panel 图的绘制方法。

(7)掌握利用系统 gsn 方法绘制 WRF 模式输出结果。

7.3 脚本讲解

WPS 是 WRF Preproc System 的缩写,是 WRF 中预处理模块,其中 geogrid 模

块提供了定义区域、生成静态数据等功能,其产生的文件名格式为“geo_em.d0x.nc”,x 为嵌套的区域号,WPS 参数设置文件名为 namelist.wps;WRF 模式输出结果文件名格式为“wrfout_d0x_yyyy-mm-dd_hh:mi:ss”,其中 x 为嵌套区域号,“yyyymm-dd_hh:mi:ss”为时间标签。

7.3.1 多层嵌套区域显示

```
wrf_ex1.ncl(图 7.1):
begin
type = "png"      ;文件格式,其他可选的有:"eps", "pdf", "ps", "ncgm"
wks = gsn_open_wks(type, "model_simulation_domain")
filename = " wrf_ex_namelist.wps"
mpres@mpDataBaseVersion      = "Ncarg4_1"
mpres@mpGeophysicalLineColor = "Black"
mpres@mpGridLineColor       = "Black"

lnres = True
lnres@gsLineThicknessF      = 3
lnres@domLineColors        = (/ "black", "black" /) ;各个嵌套区域边界的颜色

txres = True
txres@txJust = "TopLeft"
txres@txPerimOn = False
txres@txFontHeightF = 0.015

maxdom = 21
nvar = 19
parent_idn = new (maxdom, integer)
parent_grid_ration = new (maxdom, integer)
i_parent_startn = new (maxdom, integer)
j_parent_startn = new (maxdom, integer)
e_wen = new (maxdom, integer)
e_snn = new (maxdom, integer)

plotvar = wrf_wps_read_nml(filename) ; 读入区域参数设置
```

```
plotvar@_FillValue = -999.0
```

mpres@max_dom = floattointeger(plotvar(0,0)) ; 模式计算的区域数目或嵌套区域的数目

```
mpres@dx = plotvar(0,1) ; 模式格距
```

```
mpres@dy = plotvar(0,2)
```

```
if (.not.ismissing(plotvar(0,3))) then
```

```
    mpres@ref_lat = plotvar(0,3) ; 图形的几何中心所在纬度
```

```
else
```

```
    mpres@ref_lat = 0.0
```

```
end if
```

```
if (.not.ismissing(plotvar(0,4))) then
```

```
    mpres@ref_lon = plotvar(0,4) ; 图形的几何中心所在经度
```

```
else
```

```
    mpres@ref_lon = 0.0
```

```
end if
```

```
if (.not.ismissing(plotvar(0,5))) then
```

```
    mpres@ref_x = plotvar(0,5)
```

```
end if
```

```
if (.not.ismissing(plotvar(0,6))) then
```

```
    mpres@ref_y = plotvar(0,6)
```

```
end if
```

; 注意, 以下 6 行设置应根据投影方式的不同而不同, 详见附录 F 函数 wrf_ll_to_ij

```
mpres@truelat1 = plotvar(0,7) ; 标准纬度 1
```

```
mpres@truelat2 = plotvar(0,8) ; 标准纬度 2
```

```
mpres@stand_lon = plotvar(0,9) ; 中心经度
```

```
mproj_int = plotvar(0,10)
```

```
mpres@pole_lat = plotvar(0,11)
```

```
mpres@pole_lon = plotvar(0,12)
```

```
do i = 0, maxdom-1
```

```
    parent_idn(i) = floattointeger(plotvar(i,13)) ; 读取各个区域编号
```

```
    parent_grid_ration(i) = floattointeger(plotvar(i,14)) ; 各个区域相对于主
```


区域的面积比率

```

        i_parent_startn(i) = floattointeger(plotvar(i,15))      ; 各个区域左下角格
点在主区域中的格点位置(x)
        j_parent_startn(i) = floattointeger(plotvar(i,16))      ; 各个区域左下角格
点在主区域中的格点位置(y)
        e_wen(i) = floattointeger(plotvar(i,17))                ; 各个区域的東西向
网格点个数
        e_snn(i) = floattointeger(plotvar(i,18))                ; 各个区域的南北向
网格点个数
    end do

```

```

mpres@parent_id = parent_idn(0:mpres@max_dom-1)
mpres@parent_grid_ratio = parent_grid_ratio(0:mpres@max_dom-1)
mpres@i_parent_start = i_parent_startn(0:mpres@max_dom-1)
mpres@j_parent_start = j_parent_startn(0:mpres@max_dom-1)
mpres@e_we = e_wen(0:mpres@max_dom-1)
mpres@e_sn = e_snn(0:mpres@max_dom-1)
if(mproj_int .eq. 1) then
    mpres@map_proj = "lambert"
    mpres@pole_lat = 0.0
    mpres@pole_lon = 0.0
else if(mproj_int .eq. 2) then
    mpres@map_proj = "mercator"
    mpres@pole_lat = 0.0
    mpres@pole_lon = 0.0
else if(mproj_int .eq. 3) then
    mpres@map_proj = "polar"
    mpres@pole_lat = 0.0
    mpres@pole_lon = 0.0
else if(mproj_int .eq. 4) then
    mpres@map_proj = "lat-lon"
end if
end if
end if

```

```

end if

if (mpres@dx.lt.1e-10 .and. mpres@dx.lt.1e-10) then
  mpres@dx = 360./(mpres@e_we(0) - 1)
  mpres@dy = 180./(mpres@e_sn(0) - 1)
  mpres@ref_lat = 0.0
  mpres@ref_lon = 180.0
end if
mp = wrf_wps_dom(wks,mpres,lnres,txres)
frame(wks)
end

```

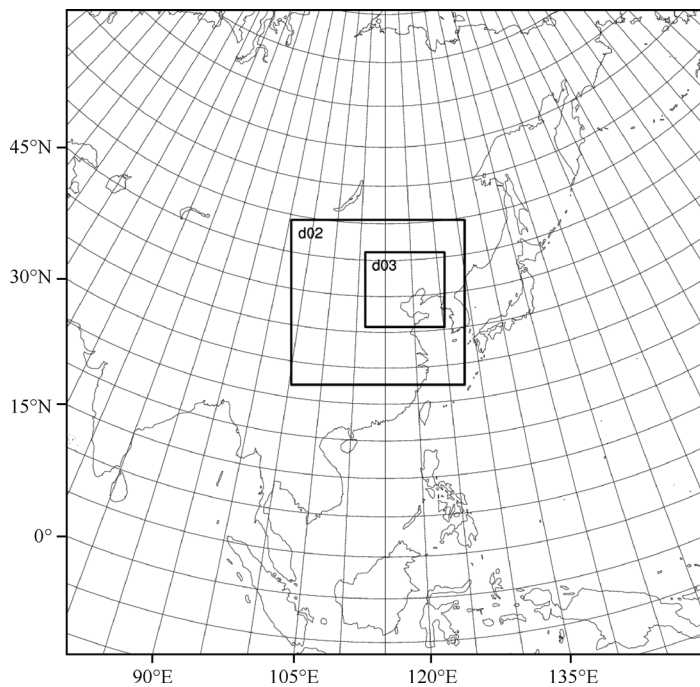


图 7.1 多层嵌套区域图

7.3.2 地表数据制图

```

wrf_ex2.ncl(图 7.2):
begin

```

a = addfile("./data/geo_em.d01.nc", "r"); 打开 WPS 中生成的地形文件或者 WRF 的输出结果

```
type = "eps" ;文件格式,可选的有"pdf","ps","ncgm","png"
wks = gsn_open_wks(type,"wrf_ex2") ; 创建 wks
gsn_define_colormap(wks,"MPL_gist_yarg")
opts = True
opts@MainTitle = " " ; 不设置标题
opts@InitTime = False ; 不画时间标签
opts@Footer = False ; 不画页脚
opts@FieldTitle = ""
opts@tiMainOn = False
opts@lbTitleOn = False
hgt = wrf_user_getvar(a,"HGT_M",0) ; 读取地形高度

res = opts
res@cnFillOn = True ; 使用填色图
```

res@ContourParameters = (/ 0., 7000., 50. /) ;设置等值线参数,这与如下绘图参数设置效果一致 **res@cnLevelSelectionMode = "ManualLevels"**, **res@cnMinLevelValF = 0**, **res@cnMaxLevelValF = 1000**, **res@cnLevelSpacingF = 50.** 也可设定 **res@ContourParameters = 50.**,即等值线间隔取 50,但起止范围由 NCL 自动判定

```
contour = wrf_contour(a, wks, hgt, res)
```

```
pltres = True
pltres@NoTitles = True
mpres = True ; 设置地图选项
mpres@mpGeophysicalLineColor = "Black"
mpres@mpGridLineColor = "Black"
plot = wrf_map_overlays(a, wks, (/contour/), pltres, mpres)
```

end

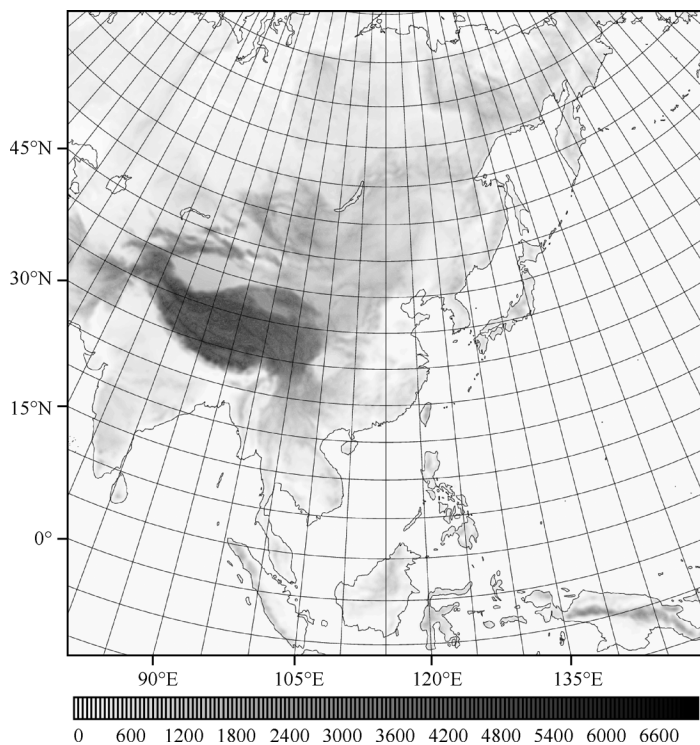


图 7.2 地形高度图(单位:m)

7.3.3 剖面图

wrf_ex3.ncl(图 7.3):

```
begin
```

```
  a = addfile("./data/wrfout_ex1_2014-12-31.nc", "r")
```

```
  wks = gsn_open_wks("png", "wrf_ex3")
```

```
  res = True
```

```
  res@MainTitle = "WRF_EX4"
```

```
  res@Footer = False
```

```
  pltres = True
```

```
  pltres@NoTitles = True
```

```
  times = wrf_user_getvar(a, "times", -1) ; 读取所有时次
```

```
  ntimes = dimsizes(times)
```

```
  FirstTime = True
```

```

mdims = getfilevardimensions(a,"P") ; 获取文件中变量的维数
nd = dimensions(mdims)
it = 1 ; 绘制第 2 时次(it=0 对应第 1 时次),若要绘制多个时次,可用循环 it=
0,ntimes-1,2
print("Working on time: " + times(it) )
res@TimeLabel = times(it) ;设置时间标签
tc = wrf_user_getvar(a,"tc",it) ; 获取指定时次的摄氏温度
rh = wrf_user_getvar(a,"rh",it) ; 获取指定时次的相对湿度
z = wrf_user_getvar(a,"z",it) ; 获取指定的高度
if (FirstTime) then ;计算高度值范围,这里是依据第 1 个时次
进行计算
    zmin = 0.
    zmax = max(z)/1000.
    nz = floattoint(zmax/2 + 1)
    FirstTime = False
end if
; 取经过中心点的南北向的一个剖面
angle = 0
plane = new(2,float)
plane = (/ mdims(nd-1)/2, mdims(nd-2)/2 /) ; 剖面点经过中心点
opts = False

;将三维变量内插至剖面上,剖面经过中心点 plane,与南北夹角为 0°,详见附录 F
函数 wrf_user_intrp3d
rh_plane = wrf_user_intrp3d(rh,z,"v",plane,angle,opts)
tc_plane = wrf_user_intrp3d(tc,z,"v",plane,angle,opts)

dim = dimensions(rh_plane)
zspan = dim(0)
opts_xy = res
opts_xy@tiYAxisString = "Height (km)"
opts_xy@cnMissingValPerimOn = True
opts_xy@cnMissingValFillColor = 0

```

```

opts_xy@cnMissingValFillPattern = 11
opts_xy@tmYLMode    = "Explicit"
opts_xy@tmYLValues  = fspan(0,zspan,nz)
opts_xy@tmYLLabels  = sprintf("%.1f",fspan(zmin,zmax,nz))
opts_xy@tiXAxisFontHeightF    = 0.020
opts_xy@tiYAxisFontHeightF    = 0.020
opts_xy@tmXBMajorLengthF      = 0.02
opts_xy@tmYLMajorLengthF      = 0.02
opts_xy@tmYLLabelFontHeightF  = 0.015
opts_xy@tiMainOn    = False
opts_xy@lbTitleOn   = False

; 相对湿度
opts_rh = opts_xy
opts_rh@ContourParameters      = (/ 10., 90., 10. /)
opts_rh@pmLabelBarOrthogonalPosF = -0.07
opts_rh@cnFillOn                = True
opts_rh@cnFillColors = (/ " White ", " Chartreuse ", " Green ", " Green3 ",
"Green4", " ForestGreen ", "PaleGreen4" /)

; 温度
opts_tc = opts_xy
opts_tc@cnInfoLabelOrthogonalPosF = 0.00
opts_tc@ContourParameters    = (/ -5. /)
opts_tc@cnLineLabelBackgroundColor = "White"

; 分别绘制温度与湿度图
contour_tc = wrf_contour(a,wks,tc_plane,opts_tc)
contour_rh = wrf_contour(a,wks,rh_plane,opts_rh)

; 叠加并出图
plot = wrf_overlays(a,wks,(/contour_rh,contour_tc/),pltres)
end

```

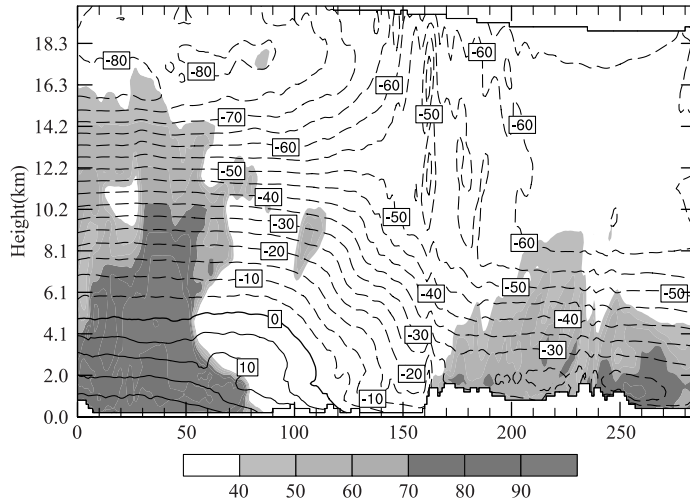


图 7.3 剖面图(附彩图 7.3)

(绿色填色为相对湿度,%; 等值线为温度,单位:℃)

7.3.4 风玫瑰图

wrf_ex4.ncl(图 7.4):

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/wind_rose.ncl"
```

```
begin
```

```
  f = addfile("./data/wrfout_ex1_2014-12-31.nc", "r")
```

```
  uvm10 = wrf_user_getvar(f, "uvm10", -1)
```

```
  u10 = uvm10(0, :, :, :)
```

```
  v10 = uvm10(1, :, :, :)
```

```
  ; 找出离 45°N, 105°E 最近的格点位置
```

```
  loc = wrf_user_ll_to_ij(f, 105., 45., True)
```

```
  mx = loc(0) - 1
```

```
  ny = loc(1) - 1
```

```
  lat1 = f->XLAT(0, ny, mx)
```

```
  lon1 = f->XLONG(0, ny, mx)
```

```
  res = True
```

```

wspd1= ndtooned(sqrt(u10(:,ny,mx)^2 + v10(:,ny,mx)^2))      ; 全风
速,转换为 1D 数组
wdir1= ndtooned(atan2(u10(:,ny,mx),v10(:,ny,mx))/0.01745329 +180.) ;
风向,转换为 1D 数组
numPetals   = 16      ;16 个风向
circFr      = 8.      ;频率圈间隔为 8%
spdBounds   = (/ 1., 3., 5., 10. /)      ;风速区间
colorBounds = (/ "blue", "green", "yellow", "red" /) ;风速区间颜色
wks = gsn_open_wks("png", "wrf_ex4")
plot = WindRoseColor ( wks, wspd1, wdir1, numPetals, circFr, spdBounds, color-
Bounds, res) ;彩色直线顶端数值表示该风向上风速的平均值
end

```

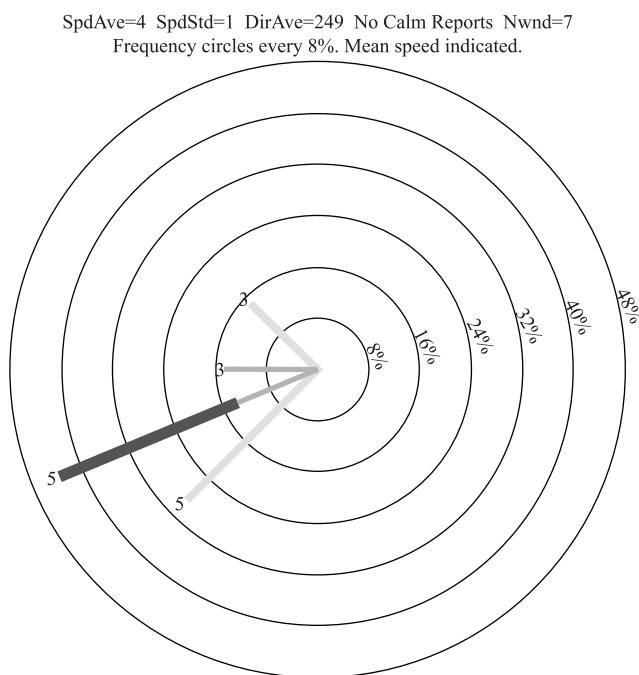


图 7.4 风玫瑰图(附彩图 7.4)

7.3.5 组图

```

wrf_ex5.ncl(图 7.5):
begin

```

```

a = addfile("./data/wrfout_ex1_2014-12-31.nc", "r")
type = "png"
wks = gsn_open_wks(type, "wrf_ex5")
  res = True
  res@NoHeaderFooter          = True
  res@pmLabelBarOrthogonalPosF = -0.1
  res@lbTitleOn = False
  pltres = True
  pltres@PanelPlot = True      ;绘制组图
  mpres = True
  mpres3 = True
  times = wrf_user_getvar(a, "times", -1) ;获取文件中所有时次
  ntimes = dimsizes(times)
  plots = new ( 4, graphic )
  it =1
  print("Working on time: " + times(it) )
  res@TimeLabel = times(it)

  slp = wrf_user_getvar(a, "slp", it)
  wrf_smooth_2d( slp, 3 )          ; 平滑变量

  if ( it .eq. 0 ) then
    tc = wrf_user_getvar(a, "tc", it)
    td = wrf_user_getvar(a, "td", it)
    u = wrf_user_getvar(a, "ua", it)
    v = wrf_user_getvar(a, "va", it)
    tc2 = tc(0, :, :) ;模式最底层气温
    td2 = td(0, :, :) ;模式最底层露点温度
    u10 = u(0, :, :)
    v10 = v(0, :, :)
  else
    tc2 = wrf_user_getvar(a, "T2", it)
    tc2 = tc2-273.16          ; 2 m 气温, 由绝对温度转换为摄氏温度
    td2 = wrf_user_getvar(a, "td2", it) ; 2 m 露点温度

```

```

    u10 = wrf_user_getvar(a,"U10",it)
    v10 = wrf_user_getvar(a,"V10",it)
end if

tf2 = 1.8 * tc2 + 32.    ; 地表温度 "Surface Temperature"
td_f = 1.8 * td2 + 32.   ; 地表露点温度 "Surface Dew Point Temp"
u10 = u10 * 1.94386      ; 单位转化为节
v10 = v10 * 1.94386
u10@units = "kts"        ; 单位为节
v10@units = "kts"

opts = res
opts@cnFillOn = True
opts@ContourParameters = (/ -20., 90., 5./)
opts@gsnSpreadColorEnd = -3
contour_tc = wrf_contour(a,wks,tf2,opts)
contour_tc2 = wrf_contour(a,wks,tf2,opts)
delete(opts)

opts = res
opts@cnFillOn                = True
opts@cnLinesOn                = True
opts@cnLineLabelsOn          = True
opts@ContourParameters        = (/ -20., 90., 5./)
opts@cnLineLabelBackgroundColor = 0
opts@cnLineLabelPerimColor    = "white"
opts@gsnSpreadColorEnd        = -3
contour_td = wrf_contour(a,wks,td_f,opts)
delete(opts)

opts = res
opts@cnFillOn = True
contour_psl2 = wrf_contour(a,wks,slp,opts)
opts@ContourParameters = (/ 996, 1032., 4. /)
opts@cnFillOn = False

```

```

opts@cnInfoLabelOn = False
opts@cnLineColor = "NavyBlue"
opts@cnHighLabelsOn = True
opts@cnLowLabelsOn = True
opts@cnLineLabelBackgroundColor = 0
opts@cnLineLabelPerimColor = "white"
opts@gsnContourLineThicknessesScale = 2.0
contour_psl = wrf_contour(a, wks, slp, opts)
delete(opts)

opts = res
opts@FieldTitle = "Wind"
opts@NumVectors = 35
vector = wrf_vector(a, wks, u10, v10, opts)
delete(opts)
;分别绘制各变量
pltres@NoTitles = True
plots(0) = wrf_map_overlays(a, wks, (/contour_tc/), pltres, mpres)
plots(1) = wrf_map_overlays(a, wks, (/contour_tc2, contour_psl/), pltres, mpres)
plots(2) = wrf_map_overlays(a, wks, (/contour_td/), pltres, mpres)
plots(3) = wrf_map_overlays(a, wks, (/contour_psl2, vector/), pltres, mpres3)

; 生成组图
pnlres = True
;pnlres@txString = "PLOTS for : " + times(it)
pnlres@gsnPanelYWhiteSpacePercent = 5
pnlres@gsnPanelScalePlotIndex = 1
pnlres@gsnPanelFigureStrings = (/ "(a)", "(b)", "(c)", "(d)" /)
pnlres@amJust = "topLeft"
gsn_panel(wks, (/plots/), (/2,2/), pnlres)

```

end

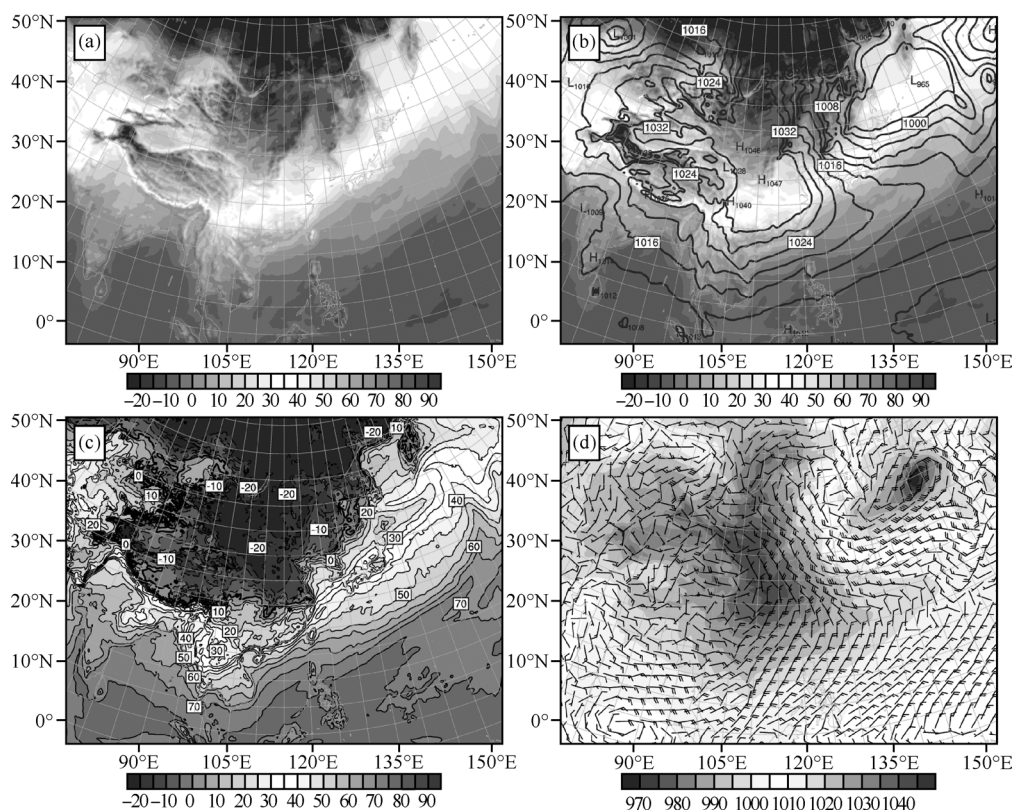


图 7.5 组图(附彩图 7.5)

(a) 地表温度, 单位: $^{\circ}\text{F}$; (b) 地表温度(填色)及海平面气压(等值线, 单位: hPa); (c) 地表露点温度, 单位: $^{\circ}\text{F}$; (d) 海平面气压(填色, 单位: hPa)及风场(风杆, 单位: 节)

7.3.6 利用 gsn 系统函数制作 WRF 图

wrf_ex6.ncl(图 7.6):

begin

```
a = addfile("./data/wrfout_ex1_2014-12-31.nc", "r")
```

```
Rain = (a->RAINNC(5, :, :) + a->RAINNC(5, :, :))/1000.
```

```
HGT = wrf_user_getvar(a, "HGT", 0)
```

```
wks = gsn_open_wks("png", "wrf_ex6")
```

```
res = True ; 绘制地形高度的相关设置
```

```
res@gsnFrame = False
```

```
res@gsnDraw = False
```

```

res@cnFillOn                = True
res@cnFillPalette           = "gsltod"      ; 选择灰度色板
res@cnLinesOn               = False         ; 关闭等值线
res@cnLineLabelsOn          = False         ; 关闭等值线标签
res@cnFillMode              = "RasterFill"
res@cnFillOpacityF          = 1.
res@lbLabelBarOn            = False
res@gsnRightString          = ""

res@mpGeophysicalLineThicknessF = 0.5
res@mpGeophysicalLineColor = "black"
res = wrf_map_resources(a, res)      ; 根据 WRF-ARW 输出数据设置地图参数

res@tfDoNDCOverlay          = True         ; 可保证正确叠加在地图上

; 设置坐标轴刻度
res@pmTickMarkDisplayMode = "Always"      ; 自动添加“较美观”的坐标轴刻度
res@tmXBLLabelFontHeightF = 0.018
res@tmYLLLabelFontHeightF = 0.018
res@tmYLLLabelStride       = 2             ; 每第二个坐标轴刻度上添加标签
res@tmXBLLabelStride       = 2
res@tmYRMajorOutwardLengthF = 0
res@tmYLMajorOutwardLengthF = 0
res@tmXBMajorOutwardLengthF = 0
res@tmXBMinorOutwardLengthF = 0
res@tmXTOn                 = True
res@tmYRON                 = True
res@tmYRLLabelsOn          = False
res@tmXTLabelsOn           = False

res_rain                   = True          ; 绘制降水量的相关设置
res_rain@gsnFrame          = False
res_rain@gsnDraw           = False

```

cmap = read_colormap_file("BlAqGrYeOrReVi200"); 读取 BlAqGrYeOrReVi200 色板

cmap(0,:) = (/0,0,0,0/) ;修改第一个颜色,使其完全透明

```
res_rain@cnFillOn           = True
res_rain@cnFillMode         = "RasterFill"
res_rain@cnFillPalette      = cmap
res_rain@cnLinesOn          = False
res_rain@cnLineLabelsOn     = False
res_rain@cnFillOpacityF     = 1.
res_rain@tfDoNDCOverlay     = True
res_rain@cnLevelSelectionMode = "ManualLevels"
res_rain@cnMaxLevelValF     = 200.
res_rain@cnMinLevelValF     = 40.
res_rain@cnLevelSpacingF    = 15.
res_rain@pmLabelBarHeightF   = 0.08 ;调整色标的大小
res_rain@lbLabelFontHeightF  = 0.014
res_rain@pmLabelBarOrthogonalPosF = -0.008
res_rain@cnInfoLabelOn      = False
res_rain@tiMainFont          = "Helvetica-bold"
res_rain@tiMainFontHeightF   = 0.018
res_rain@gsnRightString      = ""
res_rain@gsnLeftString       = ""
;res_rain@tiMainString       = "Total Rainfall"
```

plotrain = gsn_csm_contour_map(wks,HGT,res)

plot_raintot = gsn_csm_contour(wks,Rain,res_rain)

overlay(plotrain, plot_raintot)

draw(plotrain)

frame(wks)

end

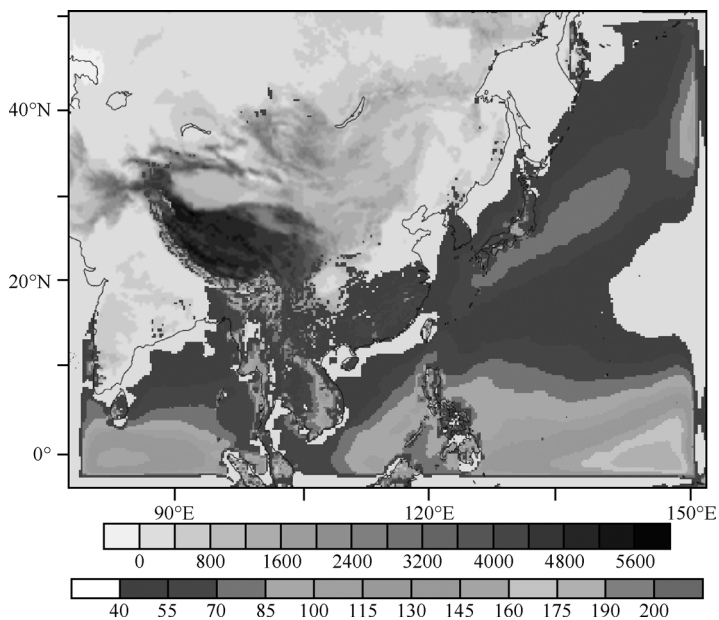


图 7.6 利用 gsn 绘图(附彩图 7.6)

(灰色阴影为地形高度,单位:m; 彩色填色为降水量,单位:mm)

7.4 思考题

- (1)除用户指定色板文件外,考虑选择其他色板文件进一步美化。
- (2)修改等值线设置参数,减少等值线数,使色标更简洁。制作 WPS 土壤、植被数据图,并修改相应的 contour 类型。
- (3)利用剖面图制作方法,完成模拟区指定两点之间的剖面图。
- (4)利用 gsn 库函数制作 WRF 结果的流场图,比较与 WRF 扩展函数库制图的差别。

第 8 章 bash 脚本编程

8.1 实习目的

- (1)掌握 bash 编程,制作动画。
- (2)掌握 bash 编程,实现与其他语言及命令的混合使用。

8.2 实习内容及要求

请首先下载附录 B 中“实习资料地址”中 animation.ncl、animation.sh、precip.f90、station.ncl 及 call-f90.sh 三个脚本。再根据以下实习内容与要求,补充完善相应脚本。

(1)制作动画。补充 animation.ncl 脚本,绘制 1979 年 1 月共 31 天的日平均全球平均地表气温图,输出为 pdf 文件。要求,每幅图上等值线用填色表示,且每种颜色所对应的数值一致。补充 animation.sh 脚本,利用 convert 命令将 animation.ncl 脚本生成的 animation.pdf 文件转换成 animation.gif 文件,最后删除原先的 animation.pdf 文件。注意,要保证 animation.sh 脚本有可执行权限,若无则在终端输入“chmod 700 animation.sh”。

(2)调用 Fortran 程序。首先编译 precip.f90(该脚本主要读入 30 年的 160 站 binary 数据,将其数值乘以 3,再输出)生成一个 precip.exe 可执行文件。补充 call-f90.sh,使得该脚本可实现如下功能:首先进入输出数据的路径,若有同名文件,则删除;调用当前目录下 data-station.ncl 脚本用于计算 160 站近 30 年夏季降水量,即第 3 章中第一个示例脚本;最后调用 precip.exe 命令。注意,要保证 call-f90.sh 脚本有可执行权限,若无则在终端中输入命令“chmod 700 call-f90.sh”。

8.3 脚本讲解

此节给出 7.2 节中所有的完整脚本,其中粗体为关键语句。网上提供的原脚本中已删除这些关键语句,读者可自行参考对照。

8.3.1 利用 bash 脚本制作动画

```
(1)animation.ncl:
begin
  a  = addfile("./data/air2m-197901-201412.nc","r")
  t  = short2flt(a->air(:,0,:,:))      ;读取 2 m 气温

wks = gsn_open_wks("pdf","animation")  ;格式必须为 pdf,才能制作动画

  res                                = True
  res@gsnLeftString  = ""
  res@gsnRightString = ""

;设定同样的等值线,以便比较
res@cnLevelSelectionMode = "ManualLevels"
  res@cnMinLevelValF    = 195.0
  res@cnMaxLevelValF    = 328.0
  res@cnLevelSpacingF   = 2.25

  res@cnFillOn          = True      ; 填色
  res@cnLinesOn         = False     ; 关闭等值线
  res@cnLineLabelsOn    = False     ; 关闭等值线的数值标签

  do ii=0,30
    res@tiMainString    = "January Global Surface Temperature (K) - Day " +
(ii + 1)
    plot = gsn_csm_contour_map(wks,t(ii,:,:),res)
  end do
end

(2)animation.sh:
#! /bin/bash

ncl animation.ncl                # 运行 ncl 脚本,生成 animation.pdf
convert animation.pdf animation.gif # 转换成 gif 图
```

```
rm animation.pdf          # 删除已生成的 animation.pdf 图
exit 0
```

(3)在终端中执行 animation.sh 脚本：
./animation.sh

8.3.2 调用 Fortran 90 脚本

```
(1)precip.f90:
program precip
  implicit none
  integer, parameter :: numb=160, nyear=30
  character * 60 inpath, outpath
  real    preci_new(numb, nyear)
  real    preci(numb, nyear)

! **** 从屏幕上读取输入及输出路径 ****
  read( *, '(A60)') inpath
  read( *, '(A60)') outpath

!input
  open(10, file=inpath, form='binary')
  read(10)  preci
  close(10)

  preci_new = preci * 3

!output
  open(2, file=outpath, form='binary')
  write(2)  preci_new
  close(2)

end
```

对其进行编译得到一个可执行文件,假设其可执行文件名为 precip.exe。

```

(2) call-f90.sh:
#! /bin/bash
path_in='./'          # 输入路径,此处为当前路径
path_out='./'         # 输出路径,此处为当前路径
infilename="precip-160-JJA-30yr.grd"
outfilename="precip_3times.grd"

cd ${path_in}
rm -f ${infilename}

# 通过参数传递变量
ncl 'filename="\${filename}" "${path_in}\data-station.ncl

# 调用 Fortran 程序
./precip <<EOF      # 调用 precip.exe
./${infilename}
./${outfilename}
EOF
exit 0

```

(3)在终端中执行 call-f90.sh 脚本:

```
./call-f90.sh
```

8.4 思考题

(1)修改 animation.ncl 及 animation.sh,绘制 300 hPa 月平均位势高度场,从 1979 年 1 月至 1980 年 12 月的 gif。要求 pdf 与 gif 在不同路径下。

(2)修改本节 precip.f90、call-f90.sh 和 6.3.1 节 plot-precip-jiangsu.ncl 脚本,使 precip.f90 完成台站资料的读取,并计算出各台站的降水异常,plot-precip-jiangsu.ncl 绘制江苏省 1998 年降水异常,最后所有命令封装在 call-f90.sh 中。

参考文献

- Bjerknes J, 1966. A possible response of the atmospheric Hadley circulation to equatorial anomalies of ocean temperature[J]. *Tellus*, **18**(4):820-829.
- Bjerknes J, 1969. Atmospheric teleconnections from the equatorial Pacific[J]. *Monthly Weather Review*, **97**(3):163-172.
- Kalnay E, Kanamitsu M, Kistler R, et al, 1996. The NCEP/NCAR 40-year reanalysis project[J]. *Bulletin of the American Meteorological Society*, **77**:437-470.
- Kanamitsu M, Ebisuzaki W, Woollen J, et al, 2002. NCEP-DOE AMIP-II Reanalysis (R-2)[J]. *Bulletin of the American Meteorological Society*, **83**(11):1631-1643.
- Liu Y, Liang X S, Weisberg R H, 2007. Rectification of the bias in the wavelet power spectrum[J]. *Journal of Atmospheric and Oceanic Technology*, **24**(12):2093-2102.
- North G R, Bell T L, Cahalan R F, et al, 1982. Sampling errors in the estimation of empirical orthogonal functions[J]. *Monthly Weather Review*, **110**(7):699-706.
- Skamarock W C, Klemp J B, Dudhia J, et al, 2008. A Description of the Advanced Research WRF Version 3. NCAR Tech. Note NCAR/TN-475+STR, 113 pp.
- The NCAR Command Language (Version 6. 4. 0) [Software], 2017. Boulder, Colorado: UCAR/NCAR/CISL/TDD. <http://dx.doi.org/10.5065/D6WD3XH5>.
- Thompson D W J, Wallace J M, 1998. The Arctic Oscillation signature in the wintertime geopotential height and temperature fields[J]. *Geophysical Research Letters*, **25**:1297-1300.
- Torrence C, Compo G P, 1998. A practical guide to wavelet analysis[J]. *Bulletin of the American Meteorological Society*, **79**:61-78.

附录

附录 A 官网网址

NCL 官方网址为 <http://www.ncl.ucar.edu/>，它为用户提供了详细的使用说明以及大量的绘图模板，这里列举其中几个较常用的链接地址(表 A. 1)，供读者日后查阅。

表 A. 1 常用链接地址

最新版 NCL 的简要说明	http://www.ncl.ucar.edu/current_release.shtml
NCL 使用手册	http://www.ncl.ucar.edu/Document/Manuals/
NCL 示例脚本	http://www.ncl.ucar.edu/Applications/
函数及程序说明	http://www.ncl.ucar.edu/Document/Function/
Resources 说明	http://www.ncl.ucar.edu/Document/Graphics/Resources/
色板的使用说明	http://www.ncl.ucar.edu/Document/Graphics/color_table_gallery.shtml
WRF 绘图示例脚本	http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/NCL_examples.htm
WRF 函数说明	http://www.ncl.ucar.edu/Document/Functions/wrf.shtml http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/NCL_functions.htm

附录 B 实习资料下载地址

本书所用实习资料可以从气象出版社网站下载。网址为：
<http://www.qxcbs.com/ebook/nclsx/mdata.html>。

附录 C 加载库函数文件

通常而言，在使用 NCL 内置函数或程序时，需在脚本开始处调用对应的库函数文件。但 V6.4 版本之后的 NCL 已对几个常用库函数文件进行了预加载，不需要用户在每个脚本中手动加载。这些预加载的库函数文件为：

```
" $NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
```

```
" $NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
" $NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
" $NCARG_ROOT/lib/ncarg/nclscripts/csm/shear_util.ncl"
" $NCARG_ROOT/lib/ncarg/nclscripts/csm/bootstrap.ncl"
" $NCARG_ROOT/lib/ncarg/nclscripts/csm/extval.ncl"
" $NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
```

由于不同函数或程序可能对应着不同的库函数文件,因此,在实际使用中,需在 NCL 官网上查看脚本中所用函数或程序所需调用的库函数文件,否则会出现类似如下的错误提醒:“fatal: Undefined identifier: (WindRoseColor) is undefined, can't continue”,即由于没有加载对应的库函数文件" \$NCARG_ROOT/lib/ncarg/nclscripts/csm/wind_rose.ncl"导致函数 WindRoseColor 不能被 NCL 识别。

附录 D 常用计算函数

本节将列出常见计算函数及程序,更多函数及程序请参考其官网(附录 A)。需要注意的是,当输入场中存在缺测值时,有些函数或程序仅在出现缺测值的格点上返回缺测值,而另一些函数或程序则不允许输入场中存在缺测值,否则输出场全部为缺测值(比如涉及球谐函数计算的函数或程序)。对于输入场不可存在缺测值的函数或程序,本节进行了标注说明。

1. 函数 `acos(value)`

功能说明:

计算 value 的反余弦值。

参数:

value, 数值(numeric)型,任意维数的一个数值或多个 $[-1, 1]$ 之间的数组。

返回值:

数值类型及维数大小同 value, 反余弦值, 即弧度。若输入值不在 $[-1, 1]$ 之间, 则返回值为 nan。

示例:

```
f = -1.0
F = acos(f)
print(F)    ; 返回值为 3.141593
```

2. 函数 `addfile(file_path, status)`

功能说明:

打开、创建或修改特定的数据文件。特定的数据文件及其对应的后缀名有: NetCDF(.nc, .cdf 或 .netcdf), HDF(.hdf 或 .hd), HDF5(.h5 或 .hdf5), HDF-EOS(.hdfEOS, .he2 或 .he4), HDF-EOS5-HDF5(.he5), CCM(.ccm), GRIB(.gr, .gr1, .grb, .grib, .grb1, .grib1, .gr2, .grb2 或 .grib2), OGR(.shp, .mif, .gmt, .rt1)。

参数:

`file_path`, 字符串(string)型, 数据文件所存放的绝对路径或相对路径。

`status`, 字符串型, 指明打开文件或写入文件。有如下三种设定, ①"r"为读; ②"w"读与写; ③"c"创建。不同类型数据文件支持的 `status` 不同, NetCDF 及 HDF-4 支持"r"、"w"和"c"; HDF5 支持"r"和"w"; GRIB versions 1 和 2、HDFEOS、HDFEOS5、CCM、OGR 仅支持"r"。

示例:

详见本书 4.3 节脚本讲解及图形示例。

3. 函数 `addfiles(file_path, status)`

用法与 `addfile` 一致, 但为对多个支持的数据文件进行读、写或创建。注意, 目前用户主要使用“读”功能, 其“写”或“创建”功能并不完善, 后续 NCL 版本可能会更新与完善。

示例 1:

打开 5 个时间上连续的 NetCDF 文件, 每个文件有 12 个时次, 将它们合并读入一个 4 维数组 `T(ntim, klvl, nlat, mlon)`, 其中 `ntim=12*5`, `klvl=7`, `nlat=48`, `mlon=96`。

```
fils = systemfunc("ls /model/annual*.nc") ; 将路径/model下所有以“annual”为开头、“.nc”为结尾的数据文件名返回至 fils, 这里假定共有 5 个文件
```

```
f = addfiles(fils, "r") ; 5 个文件全部读入
```

```
ListSetType(f, "cat") ; 设定文件合并的方式, “cat”为默认的方式, 将每个数据文件中变量的最左边维进行连接, 即不增加变量的维数, 仅增加变量最左边维的大小
```

```
T = f[:]->T ; 读取 5 个文件中的变量 T
```

```
printVarSummary(T)
```

```
Variable: T
```

```

Type: float
Total Size: 5529600 bytes
          1382400 values
Number of Dimensions: 4
Dimensions and sizes:  [time | 60] x [lev 7] x [lat | 48] x [lon | 96]
Coordinates:
    time: [2349..4143]
    lev: [850..200]
    lat: [-87.15909..87.15909]
    lon: [ 0..356.25]
Number Of Attributes: 2
units :      K
long_name :   temperature

```

示例 2:

同示例 1, 但用“join”方式合并 5 个时间段相同的数据文件, 注意 T 的维数变化

```

fils = systemfunc ("ls /model/annual * .nc")

```

```

f = addfiles(fils, "r")

```

ListSetType(f, "join") ; 增加 1 维, 其他维的大小不变. 在该例中, 可以理解
为对同一个时间或区域进行了 5 次初始扰动不同的数值模拟

```

T = f[:] -> T

```

```

printVarSummary (T)

```

```

Variable: T
Type: float
Total Size: 5529600 bytes
          1382400 values
Number of Dimensions: 5
Dimensions and sizes:  [ncl_join | 5] x [time | 12] x [lev 7] x [lat | 48] x
[lon | 96]
Coordinates:
    time: [2349..4143]
    lev: [850..200]
    lat: [-87.15909..87.15909]
    lon: [ 0..356.25]

```



```
Number Of Attributes: 2
units :           K
long_name :    temperature
```

4. 函数 `all(logical_array)`

功能说明:

判断逻辑数组 `logical_array` 是否全部为真。

参数:

`logical_array`, 逻辑型, 任意维数的逻辑数组。

返回值:

逻辑型, 若 `logical_array` 中所有值均为 True, 则返回值为 True, 否则为 False。
维数大小同 `logical_array`。

5. 函数 `any(logical_array)`

用法与 `all` 一致, 但若 `logical_array` 中有一个值为 True, 则返回值为 True, 否则为 False。

6. 函数 `asciiread(filepath, dimensions, datatype)`

功能说明:

读取 ASCII 文件。

参数:

`filepath`, 字符串型, ASCII 文件存放的绝对路径或相对路径。

`dimensions`, 整(integer)型, 一个标量或一维数组, 表示将要读取数组的维数大小, 同时也表示输出数组的维数大小。如果该值为 -1, 则将所有数据读入至一个 1 维数组中, 其大小与原文件中的数据个数相同。

`datatype`, 字符串型, 表示将要读取数组的数值类型。

示例:

给定当前路径下的一个文件 "data.txt", 它含有 5 行 3 列的共 15 个浮点(float)型数据,

```
1.  2.  3.
4.  5.  6.
7.  8.  9.
10. 11. 12.
13. 14. 15.
```

有如下几种读取方式,

(1) 读取数值 1. 至 15. 全部数据, 返回一个 5×3 的 2 维数组

`z1 = asciiread("./data.txt", (/5, 3/), "float")` ; 其中 `z1(0, 0) = 1.`, `z1(1, 0) = 4.`, `z1(1, 1) = 5.`

(2) 读取 1. 至 8., 返回一个 4×2 的 2 维数组

`z2 = asciiread("./data.txt", (/4, 2/), "float")` ; 其中 `z2(0, 0) = 1.`, `z2(1, 0) = 3.`, `z2(1, 1) = 4.`

(3) 读取数值 1. 至 15. 全部数据, 返回一个 1 维数组

`z3 = asciiread("./data.txt", -1, "float")` ; 其中 `z3(0) = 1.`, `z3(3) = 4.`, `z3(6) = 7.`

7. 函数 `asciiwrite (filepath, var)`

功能说明:

输出数值型或字符串型的 ASC II 文件。在输出的文件中, 每行仅有一个值。用户不能控制其数值的输出格式。若要控制数值格式, 请使用函数 `write_matrix`。

参数:

`filepath`, 字符串型, ASC II 文件存放的绝对路径或相对路径。

`var`, 数值型或字符串型, 要输出至 ASC II 文件的任意维数的数组。

示例:

给定一个数组 `x(5, 7)`, 将其保存至当前路径下, 并命名为 "foo.txt",

`asciiwrite ("foo.txt", x)` ; foo.txt 文件中将有 $5 \times 7 = 35$ 行, 每行仅有一个数值

8. 函数 `asin(value)`

用法与 `acos` 一致, 但为计算 `value` 的反正弦值。

9. 函数 `atan(value)`

功能说明:

计算 `value` 的反正切值。

参数:

`value`, 数值型, 任意维数的一个或多个值。

返回值:

数值类型及维数大小同 `value`, 反正切值, 即弧度。返回值范围为 $(-\pi/2, \pi/2)$ 。

10. 函数 atan2(y, x)

功能说明:

计算 (y/x) 的反正切值。

参数:

y, 数值型, 任意维数的一个或多个值。

x, 数值型, 与 y 相同维数的一个或多个值。

返回值:

如果 x 或 y 是双精度(double)型, 则返回数组为双精度型, 否则为浮点型。 (y/x) 的反正切值, 即弧度。返回值范围为 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ 。维数大小同 y 及 x。

11. 函数 avg(x)

功能说明:

计算数组 x 中所有数值的平均值。

参数:

x, 数值型, 任意维数的数组。

返回值:

如果数组 x 是双精度型, 返回数组为双精度型, 否则为浮点型。一个标量。

12. 函数 cd_calendar(time, option)

功能说明:

将混合 Julian/Gregorian 日期转换为公历日期。

参数:

time, 数值型, 用混合 Julian/Gregorian 时间表示方式的多维数组。必须有一个“units”的字符串属性, 其格式为“units since basetime”, 例如, “days since 1971-1-1”。

option, 整型, 一个标量, 指定输出的格式。

返回值:

维数大小同 time。输出的格式取决于 option 值。通常第 1 维为时间维。具体有如下几种数值:

(1) option = 0

返回的数组为浮点型, 且比数组 time 多一维:

utc_date(:, 0) ——> years

utc_date(:, 1) ——> months

`utc_date(:, 2) --> days`

`utc_date(:, 3) --> hours`

`utc_date(:, 4) --> minutes`

`utc_date(:, 5) --> seconds`

(2) `option = -1` 或 `1`

返回值格式为 `YYYYMM`, `option = 1` 时为双精度型, `option = -1` 时为整型。

(3) `option = -2` 或 `2`

返回值格式为 `YYYYMMDD`, `option = 2` 时为双精度型, `option = -2` 时为整型。

(4) `option = -3` 或 `3`

返回值格式为 `YYYYMMDDHH`, `option = 3` 时为双精度型, `option = -3` 时为整型。

(5) `option = -4` 或 `4`

返回值格式为 `YYYY.fraction_of_year`, 双精度型, `fraction_of_year` 为今年截至目前的总秒数除以当年的总秒数。

(6) `option = -5`

与 `option = 0` 相同, 但返回值为整型。

示例:

```
time = (/17522904, 17522928, 17522952, 17522976, 17523000/)
```

```
time@units = "hours since 1-1-1 00:00:0.0"
```

```
dym = cd_calendar(time, 2)
```

```
print(dym) ; 返回值为(/20000101, 20000102, 20000103, 20000104, 20000105/)
```

13. 函数 `ceil(value)`

功能说明:

返回大于 `value` 的最小整数。

参数:

`value`, 数值型, 任意维数的数组。

返回值:

数值类型、数组维数及数组大小同 `value`。

示例:

```
f = 6.4
```

```
ceil_f = ceil(f)
```

```
print(ceil_f) ; 返回值为 7
```

14. 函数 center_finite_diff_n(q, r, rCyclic, opt, dim)

功能说明:

计算数组 q 第 dim 维的中央差分。

参数:

q, 数值型, 一个多维数组。

r, 数值型, 一个标量、一维或多维数组, 表示数组 q 要进行中央差分的 1 倍格距的数值或坐标值, 可为不等间距。r 有如下三种形式:

(1) 标量: 表示相邻点的间距大小, 即 1 倍格距间隔。

(2) 一维数组: 表示 q 要进行中央差分的第 dim 维所对应的坐标变量, 它与 q 的第 dim 维大小一致。

(3) 多维数组: 同上(2), 但与 q 的维数大小一致。

rCyclic, 逻辑型。若为 True, 则 q 为循环数组, r 中数值均采用中央差分计算。若为 False, 则 q 为非循环数组, r 中的开始及结束点将通过前差或后差格式计算: 开始点, $\text{result}(0) = (q(1) - q(0)) / (r(1) - r(0))$; 结束点, $\text{result}(m) = (q(m) - q(m-1)) / (r(m) - r(m-1))$ 。

opt, 整型, 目前暂未使用, 通常设置为一个整数。

dim, 整型, 对数组 q 的第 dim 维进行中央差分。

返回值:

如果数组 x 是双精度型, 返回数组为双精度型, 否则为浮点型。维数大小同 q。

示例 1:

```
q = (/30, 33, 39, 36, 41, 37/)
```

```
r = 2.0
```

```
dqdr = center_finite_diff_n(q, r, False, 0, 0)
```

; 返回值分别为:

```
dqdr(0) = (33-30)/2 = 1.5
```

```
dqdr(1) = (39-30)/4 = 2.25
```

```
dqdr(2) = (36-33)/4 = 0.75
```

```
dqdr(3) = (41-39)/4 = 0.5
```

```
dqdr(4) = (37-36)/4 = 0.25
```

```
dqdr(5) = (37-41)/2 = -2.0
```

示例 2:

```
theta = (/298, 299, 300, 302, ..., 345, 355, 383/) ; 位温
```

```
p = (/1000, 950, 900, 850, ..., 200, 150, 100/) ; pressure(hPa)
```

`dtdp = center_finite_diff_n(theta, p, False, 0, 0)` ; 利用中央差分计算位温对气压的偏微分

15. 函数 `conform(x, r, ndim)`

功能说明:

将小数组 `r` 中数值扩展复制至与大数组 `x` 维数大小一致的新数组中。

参数:

`x`, 任意维数的数组(大数组)。

`r`, 一个变量或数组(小数组), 它的维必须对应数组 `x` 的一部分维。

`ndim`, 整型, 指明小数组 `r` 与大数组 `x` 的哪些维的大小一致。

返回值:

数值类型同 `r`, 维数大小同 `x`。

示例:

给定一个三维数组 `x(ntim, nlat, mlon)`, 一个二维数组 `t(ntim, nlon)`, `ndim = (/0, 2/)`。可见, 小数组 `t` 的维数大小与大数组 `x` 中的第 0 维与第 2 维大小一致。

`tConform = conform(x, t, ndim)` ; `tConform(ntim, nlat, mlon)`。此时, `t`, `tConform(:, 0, :)`, `tConform(:, nlat-2, :)` 这三个数组在对应位置上的数值完全相同

16. 程序 `copy_VarMeta(var_from, var_to)`

功能说明:

复制一个变量 `var_from` 的属性、行列名称和坐标变量至另一个变量 `var_to` 中。

参数:

`var_from`, 任意维数的数组。有属性、行列名称和坐标变量等元数据。

`var_to`, 任意维数的数组。它的维数必须是 `var_from` 的子集。

示例:

给定一个数组 `U(time, lev, lat, lon)`, 其包含属性、行列名称和坐标变量等元数据。

`Uavg = dim_avg_n(U, 3)` ; 此时仅有 `U` 的 `_FillValue` 这一属性传至 `Uavg`

`copy_VarMeta(U, Uavg)` ; `Uavg` 含有 `U` 的所有属性、行列名称和坐标变量

17. 函数 `cos(value)`

功能说明:

计算 `value` 的余弦值。

参数:

value, 数值型, 表示弧度的任意维数数组。

返回值:

数值类型、数组维数及数组大小同 value。

18. 函数 day_of_year(year, month, day)

功能说明:

返回指定日期在当年所有天数中的序号(从 1 开始计数)。

参数:

year, 整型, 表示年份的多维数组或标量。值必须是 ≥ 0 。

month, 整型, 与 year 相同维数大小的数组, 表示月份。数值范围为 $[1, 12]$ 。

day, 整型, 与 year 相同维数大小的数组, 表示日期。数值范围为 $[1, 31]$ 。

返回值:

数值类型、数组维数及数组大小同 year, 其中每个值代表对应日期在对应年份中的日期顺序。

示例:

```
year = (/1900, 1990, 1990/)
```

```
month = (/1, 2, 3/)
```

```
day = (/1, 1, 1/)
```

```
doy = day_of_year(year, month, day)
```

```
print(doy) ; 返回值为(/1, 32, 60/), 其第 3 个数 60 表示 3 月 1 日在 1990 年中是第 60 天
```

19. 函数 dim_avg_n_Wrap(x, dims)

功能说明:

计算数组 x 第 dims 维的平均值, 并保留元数据。

参数:

x, 数值型, 任意维数的数组。

dims, 整型, 计算 x 的第 dims 维的平均值。dims 可为一维数组。

返回值:

如果输入数组 x 是双精度型, 则返回数组为双精度型, 否则为浮点型。相较于数组 x, 返回数组少一维(对应 x 第 dims 维)。缺测值被忽略。

示例:

给定一个数组 z(time, lev, lat, lon), 计算其时间维上的平均值,

```
zAvgTime = dim_avg_n_Wrap(z, 0) ; zAvgTime(lev, lat, lon)
```

若计算所有 time 及 level 上的平均值,

```
zAvg = dim_avg_n_Wrap(z, (/0, 1/)) ; zAvg(nlat, nlon)
```

20. 函数 `dim_cumsum_n_Wrap(x, opt, dims)`

用法类似于 `dim_avg_n_Wrap`, 但为计算数组 `x` 第 `dims` 维的累加值。其中参数 `opt` 为整型, 有如下三种取值: ① `opt = 0`, 遇到缺测值后的所有累加值均为缺测值; ② `opt = 1`, 遇到缺测值时则返回缺测值, 其余继续累积求和; ③ `opt = 2`, 遇到缺测值时将缺测值计为 0 并参与计算。

示例:

```
x = (/1, 2, -999, 4, 5/)
x@_FillValue = -999
x_cumsum = dim_cumsum_n_Wrap(x, 0, 0) ; 返回值为 (/1, 3, -999, -999,
-999/)
x_cumsum = dim_cumsum_n_Wrap(x, 1, 0) ; 返回值为 (/1, 3, -999, 7, 12/)
x_cumsum = dim_cumsum_n_Wrap(x, 2, 0) ; 返回值为 (/1, 3, 3, 7, 12/)
```

21. 函数 `dim_max_n(x, dims)`

功能说明:

返回数组 `x` 第 `dims` 维中的最大值。

参数:

`x`, 数值型, 任意维数的数组。

`dims`, 整型, `x` 中的第 `dims` 维。`dims` 可为一维数组。

返回值:

数值类型同 `x`; 相较于数组 `x`, 返回数组少 `n` 维(对应 `x` 第 `dims` 维)。

示例 1:

```
f = (/ (/2.1, 3.2, 4.3/), (/ 5.4, 6.5, 7.6/), (/8.7, 9.8, 11.9/), (/1.5,
2.4, 1.8/)/) ; 4×3 数组
fmax = dim_max_n(f, 1)
print(fmax) ; 返回值为 (/4.3, 7.6, 11.9, 2.4/)
```

示例 2:

给定 `x(time, lev, lat, lon)`, 返回每个 time 及 level 上的最大值,

```
xMax = dim_max_n(x, (/2, 3/)) ; 返回值为 xMax(ntim, nlev)
```


22. 函数 `dim_min_n(x, dims)`

用法与 `dim_max_n` 一致,但返回数组 `x` 第 `dims` 维中的最小值。

23. 函数 `dim_rmvmean_n_Wrap(x, dims)`

用法与 `dim_avg_n_Wrap` 一致,但为计算数组 `x` 第 `dims` 维的距平值。

24. 函数 `dim_standardize_n_Wrap(x, opt, dims)`

用法类似于 `dim_avg_n_Wrap`,但为对数组 `x` 的第 `dims` 维进行标准化。`opt=1` 则表示计算 population standard deviation(即分母为非缺测数值的个数 `N`);否则,表示计算样本标准差(即分母为 `N-1`)。

25. 函数 `dim_stddev_n_Wrap(x, dims)`

用法与 `dim_avg_n_Wrap` 一致,但为计算数组 `x` 第 `dims` 维的标准差。

26. 函数 `dim_sum_n_Wrap(x, dims)`

用法与 `dim_avg_n_Wrap` 一致,但为计算数组 `x` 第 `dims` 维的和。

27. 函数 `dim_sum_wgt_n_Wrap(x, w, opt, dims)`

用法类似于 `dim_avg_n_Wrap`,但为计算数组 `x` 第 `dims` 维的加权和。其中 `w` 为数值型,与 `x` 第 `dims` 维大小相同的一维权重数组。`opt` 为整型,一个标量,有如下三种取值:①`opt = 0`,没有缺测值时计算加权总和;②`opt = 1`,有缺测值时仅计算所有非缺测值的加权总和;③`opt > 1`,当且仅当非缺测值的数量大于等于 `opt` 时,返回加权总和。

示例:

给定一个数组 `z(ntime, nlat, nlon)`,计算其时间维上的加权总和。

```
wgt = (/0.08493151, 0.076712325, 0.08493151, 0.08219178/)
```

```
Z = dim_sum_wgt_n_Wrap(z, wgt, 0, 0) ; 返回值为 Z(nlat, nlon)
```

28. 函数 `dim_variance_n_Wrap(x, dims)`

用法与 `dim_avg_n_Wrap` 一致,但为计算数组 `x` 第 `dims` 维的方差。

29. 函数 `dimsizes(data)`

功能说明:

返回数组 `data` 的维数大小。

参数:

data, 任意数值类型的数组。

返回值:

整型或长(long)型, 1 维数组或一个标量, 每个数值代表 data 各维中含有元素的个数。

示例:

```
x = (/ (/1, 2, 3/), (/4, 5, 6/) /) ; 2×3 的数组
print(dimsizes (x)) ; 返回值为(/2, 3/)
```

30. 函数 dtrend_msg_n(x, y, remove_mean, return_info, dim)

功能说明:

计算数组 y 第 dim 维的最小二乘线性趋势并从所有网格点中删除该趋势。

参数:

x, 数值型, 一维数组, 为 y 数组中需要扣除线性趋势的那一维的坐标变量的数值。

y, 数值型, 多维数组。

remove_mean, 逻辑型, 表明是否删除平均值。True, 删除; False 保留。

return_info, 逻辑型, 控制 Y 轴截距 y_intercept 和斜率 slope 是否以属性形式返回。True, 返回; False, 不返回。

dim, 整型, 计算 y 中第 dim 维的趋势并删除。

返回值:

数值类型、数组维数及数组大小同 y, 其数值为去除最小二乘线性趋势后的 y 数组。注意, 数组 y 第 dim 维的平均值也被减去。

示例:

给定一个三维数组 y(time, lat, lon), 其时间维的数组为 t(time), 现计算数组 y 在时间维上的最小二乘线性趋势, 且去除该趋势。

yDtrend = dtrend_msg_n(x, y, True, True, 0) ; 返回值为删除平均值及趋势的 yDtrend(lat, lon, time)。此时, 截距 yDtrend@y_intercept 与斜率 yDtrend@slope 均为 1-D 数组, 大小为 nlat×nlon。如果用户需绘制图形, 则可能需要进一步使用函数 onedtond 将上述数据转换为 2-D 数组。

```
slope2D = onedtond(yDtrend@slope, (/nlat, nlon/))
yInt2D = onedtond(yDtrend@y_intercept, (/nlat, nlon/))
```

31. 函数 dtrend_n(y, return_info, dim)

用法类似于 dtrend_msg_n, 但 y 不可存在缺测值。若存在缺测值, 请使用 dtrend_msg_n。

示例:

给定一个三维数组 U(time, lat, lon), 估计其时间维上的最小二乘线性趋势, 并减去其平均值。

yDtrend = dtrend_n(U, True, 0); 返回值为 yDtrend(time, lat, lon)

32. 函数 dv2uvF_Wrap (dv)

功能说明:

利用球谐函数根据均匀经纬度网格点上的散度场 dv 计算出辐散风场, 并保留元数据。数组 dv 中不可含有缺测值。

参数:

dv, 浮点型或双精度型, 散度场(二维或二维以上, 最右边二维必须为 lat×lon), 需覆盖全球, 其纬度维的数值需单调递增。

返回值:

数值类型同 dv, 维数大小为 $2 \times N \times nlat \times nlon$ 。最左边维的 0 为 u 分量, 1 为 v 分量。N 表示除了 dv 最右边两维的其他维数。如果 dv 的是二维数组, 那么可认为 N 为 1。

示例:

给定一个在均匀经纬度格点上的散度场 div(nlat, nlon)。

uvd = dv2uvF_Wrap(div) ; 返回值分别为纬向辐散风场 uvd(0, nlat, mlon) 及经向辐散风场 uvd(1, nlat, mlon)

33. 函数 dv2uvG_Wrap(dv)

用法与 dv2uvF_Wrap 一致, 但为根据高斯网格点上的散度场计算辐散风场。数组 dv 中不可含有缺测值。

34. 函数 eofunc_n_Wrap(data, neval, optEOF, dim)

功能说明:

经验正交函数分解, 并保留元数据。

参数:

data, 数值型, 多维数组。

neval, 整型, 一个标量, 其数值表明要返回的特征值和特征向量的个数。

optEOF, 逻辑型, 一个标量, 通过设定 optEOF 的属性用以修改函数的默认设置, 在修改前必须将 optEOF 设置为 True:

optEOF = True

optEOF@jopt = 1 ; 利用相关矩阵来计算 EOFs。默认是使用协方差矩阵 (opteof @ jopt = 0)

dim, 指明数组 data 的第 dim 维为观测数据的次数所在维。通常, 该维为时间维。

返回值:

数值类型同 data。相较于数组 x, 返回数组一方面少一维 (对应 x 第 dims 维), 另一方面在最左边维多出一维, 对应各个模态。

示例:

给定一个数组 x(time, lat, lon), 进行经验正交函数分解。

X = eofunc_n_Wrap(x, 3, False, 0) ; 返回值为 X(3, lat, lon), 表示前 3 个主要模态

35. 函数 eofunc_north(eval, N, prinfo)

功能说明:

使用 North 等(1982)中的公式(24)检验特征值是否显著分离。

参数:

eval, 数值型, 将要检验的特征值。它是由函数 eofunc 或者 eofunc_Wrap 计算所得结果的属性, 比如 pcvvar, eval_transpose 或 eval。三种属性中的任一种均可用来检验。如果 prinfo=True, 则最好使用属性 pcvvar。

N, 整型或长型, 特征值的最大可能个数。在气候学研究中, 它通常为时次。

prinfo, 逻辑型, 打印计算所得的“delta lambda”、特征值、最低和最高界限以及分离的显著性(sig)。

返回值:

逻辑型, 一维数组, 表示特征值是否显著与其他特征值相分离。

示例:

给定一个数组 x(time, lat, lon), 计算其各个主要模态的时间序列。

eof = eofunc_n_Wrap(x, 3, False, 0) ; 首先计算前 3 个主要模态

eof_ts = eofunc_ts_n_Wrap(x, eof, False, 0) ; 再计算前 3 个模态所对应的时间序列 eof_ts(3, time)

;可利用不同属性进行检验,若返回值为 True,则表明特征值与其他特征值相分离

```
ntim = dimp(0)
prinfo = True
sig_ev = eofunc_north(eof@eval, ntim, prinfo)
sig_pcv = eofunc_north(eof@pcvar, ntim, prinfo)
sig_evt = eofunc_north(eof@eval_transform, ntim, prinfo)
```

36. 函数 eofunc_ts_n_Wrap(data, evec, optETS, dim)

功能说明:

计算 data 的每个 EOF 的时间序列并保留元数据。

参数:

data,数值型,多维数组。

evec,数值型,多维数组,使用函数 eofunc_n_Wrap 计算出的各个主要 EOFs。

optETS,逻辑型,一个标量。通过 optETS 属性修改函数的默认设置,在修改前必须将 optETS 设置为 True:

```
optETS = True
```

optETS@jopt = 1 ;使用标准化数据矩阵计算时间序列。默认是使用 data 和 evec

dim,指明数组 data 的第 dim 维为观测数据的次数所在维。通常,该维为时间维。

返回值:

数值型,二维数组,维数大小分别为 eofunc 计算出的特征值数目及 data 的时间维大小。

示例:

给定一个数组 x(time, lat, lon),计算其各个主要模态的时间序列。

```
eof = eofunc_n_Wrap(x, 3, False, 0) ; 首先计算前 3 个主要模态
```

```
eof_ts = eofunc_ts_n_Wrap(x, eof, False, 0) ; 再计算前 3 个模态所对应的  
的时间序列 eof_ts(3, time)
```

37. 函数 epflux(u, v, t, plvl, lat, opt)

功能说明:

在等压面上计算准地转 E-P 通量。

参数:

u/v/t,数值型,三个变量分别为纬向风、经向风和气温,它们至少是三维数组

(plvl, lat, lon) 或四维数组 (time, plvl, lat, lon)。注意, 格点的结构必须是直线型 (rectilinear)。

plvl, 数值型, 各气压层数值 (Pa 或 hPa)。

lat, 数值型, 一维数组, 直线型格点的纬度。

opt, 逻辑型。若设为 False, 则采用默认设置。否则可能需要设置以下属性:

opt@raw, 默认是 False, 否则在任何标准化运算前返回变量;

opt@scale_sqrt_p, 默认为 True, 表示用 $\sqrt{P_0/\text{plvl}}$ 标准化 (这样会导致 E-P 通量的数值在高层较小);

opt@magf, 放大某气压层上 (比如 100 hPa) 的 E-P 通量。默认值是不进行放大;

opt@print_var_info, 打印每个变量的信息。

返回值:

list 型, 有 4 个数组结构为 (plvl, nlat) 的数组变量, 它们分别为经向 EP 通量、垂直方向 EP 通量、EP 通量散度和由其散度引起的基本气流加速度, 它们的数值类型同 u 。

示例:

```
;从当前路径下读入 NCEP 再分析资料的各个变量场
```

```
uf = addfile("uwnd.2008.nc", "r")
```

```
vf = addfile("vwnd.2008.nc", "r")
```

```
tf = addfile("air.2008.nc", "r")
```

```
;假定以下 3 个变量的数组结构均是 (time, level, lat, lon)
```

```
U = short2flt(uf->uwnd) ; m/s
```

```
V = short2flt(vf->vwnd)
```

```
T = short2flt(tf->air) ; degK
```

```
lat = U&.lat
```

```
plvl = U&.level
```

```
epf = epflux(U, V, T, plvl, lat, False)
```

```
;从 list 变量中提取各个变量
```

```
Fphi = epf[0] ;经向 EP 通量
```

```
Fp = epf[1] ;垂直向 EP 通量
```

```
EPdiv = epf[2] ; EP 通量散度
```

`dudt = epf[3]` ; 引起的基本气流的加速度

38. 函数 `escr(x, y, mxlag)`

功能说明:

计算序列 x 最右边维及序列 y 最右边维之间的交叉相关系数。

参数:

x , 数值型, 任意维数的数组。最右边维通常是时间维。

y , 数值型, 任意维数的数组。最右边维通常是时间维, 且其大小必须与 x 最右边维相同。

$mxlag$, 整型, 一个标量, 表示 x 超前 y 的时长。建议 $0 \leq mxlag \leq N/4$ 。 N 为数组 x 或 y 最右边维的序列长度。

返回值:

如果输入数组 x 是双精度型, 则返回数组为双精度型, 否则为浮点型。假定 x ($nlev$, $ntime$) 及 y ($nstat$, $ntime$), 则返回数组的维数为 ($nlev$, $nstat$, $mxlag+1$)。

示例:

给定两维数组 $x(M, N)$ 和两维数组 $y(K, N)$, 计算 x 与 y 的交叉相关系数。

`acr = escr(x, y, 10)` ; 返回值为 `acr(M, K, 0:10)`, 其最右边维的 11 个数分别表示 x 超前 y 从 0 至 10 个时次的交叉相关系数

39. 函数 `escorc(x, y)`

功能说明:

计算序列 x 最右边维及序列 y 最右边维的同时线性相关系数。

参数:

x , 数值型, 任意维数的数组。最右边维通常是时间维。

y , 数值型, 任意维数的数组。最右边维通常是时间维。最右边维大小必须与 x 最右边维相同。

返回值:

如果输入数组 x 是双精度型, 则返回数组为双精度型, 否则为浮点型。假定 x ($nlev$, $ntime$) 及 y ($nstat$, $ntime$), 则返回数组的维数为 ($nlev$, $nstat$)。

示例:

给定二维数组 $X(neval, time)$ 和 $Y(time, lat, lon)$, 计算 X 与 Y 的同时相关系数。假定 X 各维的名称从左至右分别为“VAL”、“TIME”, Y 各维的名称为“TIME”、“LAT”和“LON”。

`r = escorc(X, Y(LAT|:, LON|:, TIME|:))` ; 返回值为 `r(neval, lat,`

lon),这里使用了命名的行列对 Y 数据进行转置,以使得其最右边维为时间维,与 X 的最右边维相同

40. 函数 ezfftb(cf, xbar)

功能说明:

利用傅里叶分解得到的实部和虚部系数进行傅里叶合成。

参数:

cf,数值型,由 ezfftf 计算的傅里叶系数。cf(0, ...)是实部系数,cf(1, ...)是虚部系数。

xbar,数值型,一个标量,序列的平均值,即 0 波。

返回值:

如果输入数组 cf 是双精度型,则返回双精度型,否则返回浮点型。返回数组维数大小取决于 cf

示例:

假定对 24 个数值进行傅里叶分解,产生了 12 个实部系数 cf(0, 0:11)和 12 个虚部系数 cf(1, 0:11),即有 12 个波(可查看 ezfftf 示例)。现仅截取前 3 波及平均值以合成重建序列。

```
cReal = cf(0, 0:2)      ; 前 3 波(即 1 波、2 波、3 波)的实部系数
cImag = cf(1, 0:2)      ; 前 3 波(即 1 波、2 波、3 波)的虚部系数
cf_new = (/ (/cReal/), (/cImag/)/)
xbar = cf_new @ xbar
x = ezfftb(cf_new, xbar) ; x 为由 24 数组成的 1 维序列
```

41. 函数 ezfftf(x)

功能说明:

傅里叶分解。

参数:

x,数值型。若为多维数组,则其最右边维将进行傅里叶分解。

返回值:

如果 x 是双精度型,则返回双精度型,否则返回浮点型。假定 x 的最右边维上的格点数为 npts, N 代表其他所有维上的格点数。即 npts×x 个格点,则返回数组的大小为 2×N×(npts/2)。最左边维中的 0 与 1 分别表示实部和虚部系数。同时该数据的均值将以属性 xbar 返回。

示例:

给定一个一维数组 $x = (/1002, 1017, 1018, 1020, 1018, 1027, 1028, 1030, 1012, 1012, 982, 1012, 1001, 996, 995, 1011, 1027, 1025, 1030, 1016, 996, 1006, 1002, 982/)$

$cf = \text{ezfft}(x)$;将得到 cf 为 $[2] \times [12]$, 即 $2 \times (npts/2)$ 的数组。其平均值 $cf@xbar=1011.042.cf(0, :)$ 与 $cf(1, :)$ 分别为傅里叶分解所得的实部与虚部系数

42. 函数 f2fsh_Wrap(grid, outdims)

功能说明:

将一个标量场 $grid$ 从一种分辨率的均匀经纬度网格点上插值到另一种分辨率的均匀经纬度网格点 $outdims$ 上, 并保留元数据。数组 $grid$ 中不可含有缺测值。

参数:

$grid$, 数值型, 二维或多维数组, 其最右边两维必须为 $lat \times lon$ 。数组的纬度坐标值需单调递增。

$outdims$, 整型, 含有两个数值的一维数组, 其数值分别对应新分辨率网格的纬度和经度格点的个数。

返回值:

如果输入数组是双精度型, 则返回数组为双精度型, 否则为浮点型。返回数组维数大小为 $outdims$ 。

示例:

将数组 $x(nt, nlat, nlon)$ 从分辨率为 $2.5^\circ (nlat = 73, nlon = 144)$ 的网格上插值至分辨率为 1° 网格 ($jlat = 181, ilon = 360$)。

$X = \text{f2fsh_Wrap}(x, (/jlat, ilon/))$; 返回值为 $X(nt, jlat, ilon)$

43. 函数 f2gsh_Wrap(grid, outdims, twave)

功能说明:

将一个标量场 $grid$ 从一种分辨率的均匀经纬度网格点上插值到高斯网格点 $outdims$ 上, 并保留元数据。变量 $grid$ 中不可含有缺测值。

参数:

$grid$, 数值型, 二维或多维数组, 其最右边两维必须为 $lat \times lon$ 。数组的纬度坐标值需单调递增。

$outdims$, 整型, 含有两个数值的一维数组, 其数值分别对应新分辨率网格的纬度和经度格点的个数。

$twave$, 整型, 一个标量, 有如下三种数值表示不同的截断方式:

- (1) $twave = 0$, 精确插值(无截断)。
- (2) $twave > 0$, 采用截断至 $twave$ 波的三角形截断。
- (3) $twave < 0$, 采用截断至 $twave$ 波的三角形截断, 且谱系数随波数的增大逐渐减小(平滑数据)。

返回值:

如果输入数组是双精度型, 则返回双精度型, 否则, 返回浮点型。返回数组维数大小为 `outdims`。

示例:

将均匀经纬度网格点分辨率为 2.5° 的数组 `x(nt, nlat, nlon)` (`nlat = 73, nlon = 144`) 插值至高斯网格(`jlat = 64, ilon = 128`), 采用 42 波三角形截断。

`X = f2gsh_Wrap(x, (/jlat, ilon/), 42)` ; 返回值为 `X(nt, jlat, ilon)`

44. 函数 `fbindirread(path, rec_num, rec_dims, rec_type)`

功能说明:

读取无格式二进制数据文件, 其方式等同于 Fortran 的“`form = unformatted, access = direct, recl = ..`”。

参数:

`path`, 字符串型, 无格式二进制数据文件的存放路径。

`rec_num`, 整型, 数据的记录号(从 0 开始)。

`rec_dims`, 整型或长型, 1 维数组, 表示每个记录号中的数组各维的大小。如果设为 -1, 则数据将被一次读入并保存为一个 1 维的数组。

`rec_type`, 字符串型, 数据的类型。

返回值:

`rec_type` 类型, 维数大小为 `rec_dims`。

示例:

`path = "/dummy/file.binary"`; `file.binary` 存放在 `/dummy` 下

`nrec = 5` ; 第 5 记录号(从 0 开始计数)

`x = fbindirread(path, nrec, (/10, 30/), "float")`; `x(10, 30)` 为 2 维数组

`y = fbindirread(path, nrec, (/10, 3, 10/), "float")` ; `y(10, 3, 10)` 为 3 维数组

45. 程序 `fbindirwrite(path, var)`

功能说明:

输出无格式二进制数据, 其方式等同于 Fortran 的“`form = unformatted, access =`

direct, recl= ..”。

参数:

path, 字符串型, 无格式二进制数据文件的存放路径。

var, 任意数值类型的数组变量。

示例:

给定一个 3 维数组 z(100, 64, 128), 将其写入“file.grd”中, 以下两种方式均可,

path = "/dummy/file"

(1) 一次输出

fbindirwrite(path, z)

(2) 按数据的时次循环输出(耗时多)

do nt=0, 99

fbindirwrite(path, z(nt, :, :))

end do

46. 程序 fbignumrec(path)

功能说明:

返回等同于 Fortran 采用顺序方式创建的无格式数据文件的记录号总数。

参数:

path, 字符串型, 无格式二进制数据文件的存放路径。

示例:

nrec = fbignumrec("foo.binary") ; nrec 是一个整型标量值

47. 程序 fbinrecread(path, rec_num, rec_dims, rec_type)

功能说明:

读取用顺序方式创建的数据文件。

参数:

path, 字符串型, 无格式二进制数据文件的存放路径。

rec_num, 整型, 数据的记录号(从 0 开始)。

ec_dims, 整型或长型, 1 维数组, 表示每个记录号中的数组各维的大小。如果设为 -1, 则数据将被一次读入并保存为一个 1 维的数组。

rec_type, 字符串型, 数据的类型。

示例:

首先给出 Fortran77 采用顺序方式创建的无格式文件的代码片段,

integer a(5)

```

real      x(100), y(399), z(128,64)

open (11,file="example01",form="unformatted")

write (11) a           !第 1 记录   [rec_num=0]
write (11) x           !第 2 记录   [rec_num=1]
write (11) y           !第 3 记录   [rec_num=2]
write (11) z           !第 4 记录   [rec_num=3]

```

以下为用 NCL 读取 example01 的代码,

```

fili = "example01"      ; input file
a = fbinrecread (fili, 0, 5, "integer")      ; 第 1 记录号为 0
x = fbinrecread (fili, 1, 100, "float")      ; 第 2 记录号为 1
y = fbinrecread (fili, 2, 399, "float")      ; 第 3 记录号为 2
z = fbinrecread (fili, 3, (/ 64,128 /), "float"); 第 4 记录号为 3。注意这里
的 z 是(64,128)的数组,与 Fortran 中的 z(128,64)不同。这是因为 NCL 中的变量
是最右边维变化最快,而 Fortran 中则是最左边维变化最快

```

data = fbinrecread (fili, 1, -1, "float") ; 从第 1 记录开始读入 x, y, z 并
将其合并为一个 1 维数组,其长度为(100+399+128 * 64)

48. 程序 fbinrecwrite(path, rec_num, var)

功能说明:

为变量 var 分配一个记录号按无格式二进制形式写入 path 文件中,等同于 Fortran 采用顺序方式写入的无格式数据文件。

参数:

path, 字符串型, 无格式二进制数据文件的存放路径。

rec_num, 整型, 要写入的记录号, 从 0 开始。若值为 -1, 则表示在之前的记录号后顺序添加。

var, 任意数值类型的数组变量。

示例:

给定 a(5) 为 integer 数组, x(100)、y(399)、z(64,128) 为浮点型数组。将这些数组以顺序方式写入 test.grd 文件中,

```

filo = "test.grd"      ; 要输出的文件名称

```

```

fbinrecwrite (filo, -1, a)    ; 写入 a
fbinrecwrite (filo, -1, x)
fbinrecwrite (filo, -1, y)
fbinrecwrite (filo, -1, z)

```

以下给出用 Fortran77 读取刚创建的 test.grd 文件的代码片段。注意,要按照写入的顺序进行读取,

```

integer a(5)
real  x(100), y(399), z(128,64)

open (11,file=" test.grd",form="unformatted", access="sequential")

read (11) a
read (11) x
read (11) y
read (11) z

```

49. 函数 filwgt_lanczos(nwt, ihp, fca, fcb, nsigma)

功能说明:

计算 lanczos 滤波器的权重系数。

参数:

nwt, 整型, 一个标量, 表示权重系数的总个数(必须是奇数; $nwt \geq 3$)。权重越多, 滤波效果越好, 但序列两端的数据损失越多。

ihp, 整型, 一个标量。ihp = 0, 表示低通滤波器; ihp = 1, 高通滤波器; ihp = 2, 带通滤波器。

fca, 浮点型或双精度型, 一个标量, 表示滤波器响应函数的高通或低通的截断频率: ($0 < fca < 0.5$)。

fcb, 浮点型或双精度型, 一个标量。只有在进行带通滤波时, 才需设定该第二截断频率($fca < fcb < 0.5$), 否则设为 -999。

nsigma, 数值型, 一个标量。表示 σ 因子的功率($nsigma \geq 0$)。注: 通常设 nsigma = 1。

返回值:

如果 fca 是双精度型, 则该系数为双精度型, 否则为浮点型。返回数组为长度为 nwt 的一维数组。

示例:

给定一个 365 日、1 日 4 次数组 $x(\text{ntime}, \text{nlat}, \text{nlon})$, 构建一个响应函数周期范围在 10 至 30 天的由 91 个权重系数构成的 lanczos 滤波器,

$\text{ihp} = 2$;带通滤波

$\text{fca} = 1./(30 * 4)$;截断的低频频率

$\text{fcb} = 1./(10 * 4)$;截断的高频频率

$\text{wts} = \text{filwgt_lanczos}(91, \text{ihp}, \text{fca}, \text{fcb}, 1)$; 注意,若进行滤波,还需调用 wgt_runave 、 wgt_runave_n 、 wgt_runave_Wrap 或 wgt_runave_n_Wrap 函数。详见下文 wgt_runave_n_Wrap 函数讲解

50. 函数 floor(value)

用法与 ceil 一致,但为返回小于输入值的最大整数。

示例:

$f = 6.4$

$\text{floor_f} = \text{floor}(f)$

$\text{print}(\text{floor_f})$; 返回值为 6

51. 函数 fspan(start, finish, npts)

功能说明:

创建一维浮点型的等差数列。

参数:

start, 数值型, 开始值。

finish, 数值型, 结束值。

npts, 整型, 项数, 需大于等于 2。

返回值:

浮点型, 一个从 start 开始到 finish 结束的共 npts 项的等差数列。

示例:

$x = \text{fspan}(0, 20, 5)$; 返回值为 $x = (/0, 5, 10, 15, 20/)$

52. 函数 g2fsh_Wrap(grid, outdims)

用法同 f2fsh_Wrap, 但为将高斯网格点上的变量插值至均匀经纬度网格上。

53. 函数 g2gsh_Wrap(grid, outdims, twave)

用法同 f2gsh_Wrap, 但为将高斯网格点上的变量插值至另一分辨率的高斯网格

点上。

54. 函数 `gc_latlon (lat1, lon1, lat2, lon2, npts, iu)`

功能说明:

计算球表面两点之间大圆路径的距离并进行插值以加密格点。

参数:

`lat1/lon1`, 数值型, 一个标量或多维数组, 第一个点的纬度/经度。若为多维数组, 则其数组维数及大小必须与 `lat2/lon2` 相同。单位分别为“degrees_north”和“degrees_east”。

`lat2/lon2`, 数值型, 同大小的一个标量或多维数组, 第二个点的纬度/经度。单位分别为“degrees_north”和“degrees_east”。

`npts`, 两点之间沿大圆路径要插值加密的格点数目。实际增加的格点数目为 `npts-2`。

`iu`, 一个有双重作用的控制值。其绝对值表示返回的大圆路径距离所采用的单位, 1 为弧度, 2 为度, 3 为米, 4 为千米; 正号表示经度范围是 $[0, 360)$, 负号则是 $[-180, 180)$ 。

返回值:

双精度型或浮点型, 一个标量或与 `lat2/lon2` 维数及大小相同的数组。

示例 1:

`gcdist = gc_latlon(0., 0., 90., 0., 2, 4)` ; 以千米为单位返回赤道至北极点的大圆路径的距离

示例 2:

`gcdist = gc_latlon(20., -120., 60., -64, 10, 2)` ; 以度为单位返回 $[20^{\circ}\text{N}, 120^{\circ}\text{W}]$ 与 $[60^{\circ}\text{N}, 64^{\circ}\text{W}]$ 两点之间的距离, 同时其中间等距离的插值加密 8 个格点, 其中格点的经度的取值范围为 $[0, 360)$

`print(gcdist@gclat+" "+gcdist@gclon)` ; 输出显示

结果如下:

(0)	20	240	; lat1, lon1 [0-360)
(1)	25.356	243.438	; 该行及以下 7 行的数值均为插值加密的格点
(2)	30.6249	247.194	;
(3)	35.7735	251.377	;
(4)	40.7574	256.131	;
(5)	45.5151	261.637	;
(6)	49.9606	268.124	;

```
(7)      53.9734  275.859      ;  
(8)      57.3891  285.105      ;  
(9)      60       296          ; lat2, lon2
```

55. 函数 get1Dindex (x, wanted_value)

功能说明:

返回数组 x 中所有等于 wanted_value 的下标变量。

参数:

x, 数值型, 一维数组。

wanted_value, 数值型, 与 x 相同类型的一维数组。

返回值:

整型或长型, 一维数组。

示例:

```
year = ispan(1985, 2006, 1)      ;从 1985 至 2006 间隔为 1 的等差序列  
year_exc = (/1989, 2005, 2006 /)  
i = get1Dindex (year, year_exc)  ; i 为 (/4, 20, 21/) , 即 year(4)=1989,  
year(20)=2005, year(21)=2006 这 3 个数的下标变量在返回变量 i 中
```

56. 函数 get1Dindex_Exclude(x, exclude_value)

功能说明:

返回数组 x 中所有不等于 exclude_value 的数值的下标变量。

参数:

x, 数值型, 一维数组。

exclude_value, 数值型, 与 x 相同类型的一维数组, 其数值必须存在于 x 中。

返回值:

整型或长型, 一维数组。

示例:

```
year = ispan(1985, 2006, 1) ;从 1985 至 2006 间隔为 1 的等差序列  
year_exc = (/1985, 1989, 1999, 2005, 2006 /)  
i = get1Dindex_Exclude(year, year_exc) ; i 为 (/1, 2, 3, 5, 6, 7, 8, 9, 10,  
11, 12, 13, 15, 16, 17, 18, 19/) , 或反过来思考 year(0)=1985, year(4)=1989,  
year(14)=1999, year(20)=2005, year(21)=2006 这 5 个数存在于 exclude_  
value 中
```


57. 程序 `gradsf(z, gzx, gzy)`

功能说明:

用球谐函数计算位于均匀经纬度网格点上 z 变量的梯度。输入场 z 中不可含有缺测值。

参数:

z , 数值型, 位于均匀经纬度网格点上的二维或二维以上的数组, 其最右边两维必须为 $nlat \times nlon$ 。数组 z 的网格必须覆盖全球, 其纬度坐标值需单调递增。

gzx 及 gzy (输出), 浮点型或双精度型, z 的梯度场, 其维数大小与类型与 z 相同。

示例:

给定一个分布在均匀经纬度网格点上的数组 $z(nlat, nlon)$, 其纬度坐标数值从南向北单调递增, 计算 z 的纬向和经向梯度。

```
z_grad_lon = z
z_grad_lat = z
gradsf(z, z_grad_lon, z_grad_lat)
z_grad_lon@long_name = "longitudinal gradient"
z_grad_lat@long_name = "latitudinal gradient"
z_grad_lat@units = "K/m"
z_grad_lon@units = "K/m"
```

58. 程序 `gradsg(z, gzx, gzy)`

用法与 `gradsf` 一致, 但为用球谐函数计算高斯网格上变量 z 的梯度。

59. 函数 `ind(larray)`

功能说明:

返回数组中值为 `True` 的下标变量。

参数:

$larray$, 逻辑型, 一维逻辑数组。

返回值:

整型或长型, 一维数组。

示例:

```
z = (/18, -999, 74, -999, 304/)
z@_FillValue = -999
lz = ind(.not.ismissing(z))
```

`print(lz)` ; 返回值为(0, 2, 4/)

60. 函数 `ind_resolve(indices, dsizes)`

功能说明:

将一维数组中的下标变量 `indices` 换算成各维大小为 `dsizes` 的多维数组中的下标变量。

参数:

`indices`, 整型, 一维数组的下标变量。

`dsizes`, 整型, 各维大小的一维数组。

返回值:

整型或长型。2 维数组, 表示原一维数组中的下标变量在各维大小为 `dsizes` 的多维数组中对应的下标变量。该数组左边维的大小对应下标变量中为 `True` 的总个数, 右边维的大小为输入数组下标变量的维数。

示例:

`a = ((/(/(1, 2, 3, 4/), (5, 6, 7, 8/)/), (/(9, 10, 9, 8/), (7, 6, 5, 4/)/)/)` ; `a` 为 $2 \times 2 \times 4$ 的 3 维数组, 即维数为 3

`a1D = ndtooned(a)`

`dsizes_a = dimsizes(a)` ; (2, 2, 4/)

`Indices = ind_resolve(ind(a1D.eq(5)), dsizes_a)` ; `a` 中共有 2 个值等于 5

`print(indices)` ; 返回值为(0, 1, 0/), (1, 1, 2/)/, 即 2×3 数组, 2 表示有 2 个数, 3 则表示 `a` 数组为 3 维数组。`a(0, 1, 0)` 与 `a(1, 1, 2)` 的数值等于 5

61. 函数 `int2p_n_Wrap(pin, xin, pout, linlog, pdim)`

功能说明:

将变量从一种气压坐标系内插至另一气压坐标系下。

参数:

`pin`, 数值型, 多维数组, 表示输入变量 `xin` 的气压层次数值。气压层次维必须在第 `pdim` 维, 且其数值必须单调递增或递减。

`xin`, 数值型, 要插值的多维数组变量。若参数 `pin` 为多维数组, 则 `xin` 与 `pin` 同维、同大小。

`pout`, 数值型, 多维数组, 表示插值后的气压层次数值。气压层次维必须在第 `pdim` 维, 且其数值必须单调递增或递减其单位必须与参数 `pin` 相同。

`linlog`, 整型, 表示插值类型。绝对值为 1 为线性内插, 绝对值不为 1 则为对数内插。若该值为负, 则当气压层在参数 `pin` 范围以外时, 进行外插。

pdim, 整型, 一个标量, 表示 xin 的第 pdim 维进行插值。

返回值:

如果 xin 是双精度型, 返回双精度型, 否则返回浮点型。维数大小同 xin, 但高度层次维已由 pout 替换。

示例:

```
linlog = -1 ; 线性内插和外插
pi = (/ 10., 20., 30., 50., 70., 100., 150., 200., 250., 300., 400., 500.,
600., 700., 850., 925., 1000. /)
xi = (/ -46., -48., -52., -61., -72., -78., -67., -52., -40., -30.,
-15., -4., 2., 10., 18., 23., 28. /)
po = (/ 5., 7., ; 外插
10., 15., 20., 25., 30., 40., 50., 70., 85., 100., 200., 250., 300., \ ; 内插
400., 425., 500., 600., 700., 750., 800., 850., 900., 925., 1000., \ ; 内插
1005., 1012. /) ; 外插
po!0 = "p"
po@units = "hPa"
xo = int2p_n_Wrap (pi, xi, po, linlog, 0)
```

62. 函数 ismissing(data)

功能说明:

判断数组 data 中的每一个值是否为缺测值。

参数:

data, 任意数值类型的数组变量。

返回值:

逻辑型, 维数大小与 data 一致。

63. 函数 ispan(start, finish, stride)

功能说明:

创建一维的整型、长型或 int64 型的等差数列。

参数:

start, 整型, 开始值。

finish, 整型, 结束值。finish 值可小于 start 值。

stride, 整型, 间隔值。

返回值:

整型、长型或 int64 型, 返回一个从 start 开始到 finish 结束的等差数列。

示例:

```
x = ispan(0, 10, 2)
print(x); 返回值为 x = (/0, 2, 4, 6, 8, 10/)
```

64. 函数 linint2_Wrap(xi, yi, fi, fiCyclicx, xo, yo, foOption)

功能说明:

采用双线性插值将变量 fi 从一种直线网格(rectilinear grid) $yi \times xi$ 插值到另一种直线网格 $yo \times xo$, 并保留元数据。这里的直线网格是指网格系的 X 和 Y 方向坐标均为一维变量, 但其不一定为等差数列。

参数:

xi 及 yi, 数值型, 一维数组, 数组 fi 的 X 及 Y 坐标变量, 通常为单调递增的(不)等间距数组。在某些情况下, xi 可以是一个多维数组。xi 与 yi 一般为经度和纬度数组, 其大小分别为 nxi 与 nyi。

fi, 数值型, 二维或二维以上用于插值的数组。最右边两维的大小分别为 nyi 与 nxi。

fiCyclicx, 逻辑型, 设定 fi 最右边维的循环特征。若 fi 最右边维在 X 轴(通常为经度)上循环, 则设置为 True, 否则 False。注意, 当经度值范围为从 -179.75 到 179.75, 或从 0 到 359 等, 均为循环经度。

xo 及 yo, 数值型, 返回数组的 X 及 Y 坐标变量。它必须是单调递增的, 可为不等间距。xo 及 yo 一般为经度和纬度数组。

foOption, 整型, 目前暂未使用, 通常设置为 0。

返回值:

如果 fi 是双精度型, 返回双精度型, 否则返回浮点型。

示例:

给定一个四维数组 fi(ntim, nlvl, nlat, mlon), ntim=50, nlvl=30, nlat=64, mlon=128。fi 最右边维为循环经度维。将所有时间和层次上的数据均插值到 nlat_new=73 及 mlon_new=144 网格中。

```
lon = (0., 2.8125, ... , 357.0125)
lat = (-87.8638, ... , 87.8638)
LON = (0., 2.5, ... , 357.5)
LAT = (-90., -87.5, ... , 90.)
fo = linint2_Wrap(lon, lat, fi, True, LON, LAT, 0)
print(fo); 返回值为 fo(ntim, nlvl, 73, 144)
```

65. 函数 `linmsg_n(x, opt, dim)`

功能说明:

用线性插值的方法填补 x 中第 dim 维上的缺测值。

参数:

x , 数值型, 任意维数的数组。

opt , 整型, 一个标量或具有两个数值的数组, 表明缺测值的开始点和结束点的返回形式。如果 opt 为一个标量, 其数值 ≥ 0 , 则开始点和结束点仍以缺测值返回(最常见的选项); 如果 $opt < 0$, 则返回最接近的非缺测值。如果 opt 有两个数值, 其第一个值的设定方法同 opt 仅为一个标量时的设定方法, 而其第二个值则指定可进行插值的连续缺测值的最大数目。

dim , 整型, 填充数组 x 的第 dim 维的缺测值。

返回值:

双精度型或浮点型, `linmsg` 采用分段线性插值填补缺测值。返回数组维数大小同 x 。输入为缺测值, 结束点可为缺测值或为最接近的非缺测值。

示例:

给定一个四维数组 $x(ntim, klev, nlat, nlon)$, 其各维的名称从左至右分别为“time”、“lev”、“lat”、“lon”。假定 x 在某些时次上(即最左边的第 0 维)有缺测值, 现利用 `linmsg_n` 函数进行插值。

$opt = (/0, 5/)$; 第一个数值设为 0, 表示起始与最终的缺测值不做处理, 仍返回缺测值, 第二个数值设为 5, 表示遇到连续 5 个时次以上缺测数据时, 不做插值填充处理, 仍保留其缺测值

$y = \text{linmsg_n}(x, opt, 0)$; 返回值为 $y(ntim, klev, nlat, mlon)$

66. 函数 `local_max(x, cyclic, delta)`

功能说明:

返回 2 维数组 x 中局地极大值的个数、数值及其在数组中的下标变量。

参数:

x , 浮点型或双精度型, 2 维数组。

$cyclic$, 逻辑型。数组的最右边维若是循环维, 则设为 True, 否则为 False。

$delta$, 数值型, 容差度。如果与周边差值在 $delta$ 以内, 它将不会被计为一个局地极大值。一般情况下, 设置 $delta = 0$ 。

返回值:

整型, 一个标量。同时, 以属性形式返回如下三个变量, 局地极大值的 ① 数值

maxval 及其在②X 轴和 Y 轴中的坐标位置 xi 及 yi。

示例:

给定一个数组 x(nlat, nlon), nlat=73, nlon=144, 数组在经度维循环, 找出其局地极大值的数值及其位置。

```
nmax = local_max(x, True, 0.0)
print(nmax@maxval)    ; 局地极大值的数值,(/5854, 5875, 5892/)
print(nmax@xi)        ; 局地极大值的 X 轴坐标,(/95, 142, 7/)
print(nmax@yi)        ; 局地极大值的 Y 轴坐标,(/31, 43, 45/)
```

67. 函数 local_max_1d(x, cyclic, delta, iopt)

功能说明:

返回一维数组的局地极大值或其下标变量。

参数:

x, 数值型, 一维数组。

cyclic, 逻辑型。如果数组是循环数组, 则设为 True, 否则为 False。

delta, 数值型, 容差度。如果值在 delta 以内, 它将不会被计为一个局地极大值。

iopt, 整型。iopt = 0 返回局地极大值; iopt ≠ 0, 则返回局地极大值的下标变量。

返回值:

数值型, 一维数组。

示例:

给定一个数组 H(lat), 数组不循环, nlat=73。

```
Hmax=local_max_1d(H, False, 1, 0)    ; 返回值为(/5858, 5892, 5166/)
Hmax=local_max_1d(H, False, 1, 1)    ; 返回值为(/34, 45, 68/)
```

68. 函数 local_min(x, cyclic, delta)

用法与 local_max 一致, 但返回 2 维数组 x 中的局地极小值的个数、数值及其在数组中的下标变量。

69. 函数 local_min_1d(x, cyclic, delta)

用法与 local_max_1d 一致, 但返回一维数组 x 的局地极小值或其下标变量。

70. 函数 log(value)

功能说明:

计算数值 value 的自然对数值。

参数:

value, 数值型, 任意维数的大于 0 的值。

返回值:

如果输入数组是双精度型, 返回双精度型, 否则, 返回浮点型。返回数组维数大小同 value。

示例:

```
f = 3.6
log_f = log(f)
print(log_f)    ; 返回值为 1.280934
```

71. 函数 log10(value)

用法与 log 一致, 但为计算以 10 为底的对数值。

72. 函数 max(value)

功能说明:

返回多维数组 value 中的最大值。

参数:

value, 数值型, 一维或多维数组。

返回值:

数值类型同 value, 一个标量。

示例:

```
f = (/ (/2.1, 3.2, 4.3/), (/ 5.4, 6.5, 7.6/), (/8.7, 9.8, 11.9/)/)
print(max (f))    ; 返回值为 11.9
```

73. 函数 maxind(arg)

功能说明:

返回数组 arg 中第一个最大值的下标变量。

参数:

arg, 数值型, 一维数组。

返回值:

整型或长型, 一个标量。

示例:

```
x = (/3, 1, 5, 1, 5, 2, 5, 1, 3, 2/)
```

```
print(maxind(x)) ; 返回值为 2
```

74. 函数 min(value)

用法与 max 一致,但返回多维数组 value 中的最小值。

75. 函数 minind(value)

用法与 maxind 一致,但返回数组 value 中第一个最小值的下标变量。

76. 函数 mod(n, m)

功能说明:

计算 n 除以 m 的余数。

参数:

n,数值型,一个标量或数组。

m,数值型,一个标量或数组,最好与数组 n 类型相同。

返回值:

数值类型、数组维数及数组大小同 n。

示例:

```
x=mod(17, 3)
print(x) ; 返回值为 2
```

77. 函数 month_to_season(xMon, season)

功能说明:

计算月平均数据 xMon 指定季节 season 的季节平均。

参数:

xMon,数值型,可为一维(time)、三维(time, lat, lon)或四维(time, lev, lat, lon)的月平均数组。time 维必须为 12 的倍数,NCL 将第一个时次识别为 1 月。

season,字符串型,表明计算季节平均的季节。可设定为 DJF,JFM,FMA,MAM,AMJ,MJJ,JJA,JAS,ASO,SON,OND,NDJ。注意,在 NCL 的实际计算中,DJF = JF,即仅取 1 月 2 月平均。同样 NDJ = ND。

返回值:

数值类型、数组维数及数组大小同 xMon,但最左边维(即时间维)的大小将被除以 12。如果 xMon 包含元数据,则输出场保留元数据。

示例:

给定一个数组 z(ntime, nlat, nlon),ntime 为 12 月×10 年 = 120 个时次,假定

第一个时次是 1990 年 1 月,现计算其夏季平均。

`Z = month_to_season(z, "JJA")` ; 返回值为 `Z(ntime/12, nlat, nlon)`

78. 函数 `monthday(year, day)`

功能说明:

转换 year 年中第 day 天为公历日期(月日)。

参数:

year, 整型, 表示年份的多维数组或标量。值必须是 ≥ 0 。

day, 整型, 与 year 相同维数大小的数组, 表示一年中第 day 天。数值范围为 $[1, 366]$ 。

返回值:

数值类型、数组维数及数组大小同 year, 每个数值表示由月与日组成的整型数值。

示例:

```
year = (/1933, 1996, 1997/)
```

```
day = (/245, 366, 365/)
```

```
md = monthday (year, day)
```

`print(md)` ; 返回值为 `(/902, 1231, 1231/)`, 即 1933 年的第 245 天是 9 月 2 日, 1996 年的第 366 天是 12 月 31 日, 1997 年的第 365 天是 12 月 31 日

79. 函数 `ndtooned(val)`

功能说明:

将多维数组 val 转换为一维数组。

参数:

val, 任意维数或类型的数组。

返回值:

数值类型同 val, 将多维数组转换为相同大小的一维数组。

示例:

```
a = (/ (/1, 2, 3/), (/4, 5, 6/), (/7, 8, 9/) /) ; a 为 3×3 数组
```

```
b = ndtooned(a)
```

`print(b)` ; 返回值为 `(/1, 2, 3, 4, 5, 6, 7, 8, 9/)`, 即 1 维数组, 共 9 个数值

80. 函数 `new(dimension_sizes, vartype, parameter)`

功能说明:

创建一个数组变量。

参数:

`dimension_sizes`, 整型或长型, 要创建的数组变量的维数大小。

`vartype`, 字符串型, 数组变量的数值类型。通常有“float”、“integer”、“string”、“graphic”等。

`parameter`, 缺测值的数值。该参数为可选参数。若设置, 则可以是一个数值或者字符串“No_FillValue”。“No_FillValue”表示数组中无缺测值。

返回值:

数值类型 `vartype`, 维数大小为 `dimension_sizes`。

示例:

创建一个没有缺测值的浮点型数组。

```
x = new((/64, 128/), float, "No_FillValue")
```

81. 函数 `num(val)`

功能说明:

统计数组 `val` 中值为 `True` 的个数。

参数:

`val`, 逻辑型, 任意维数数组。

返回值:

整型或长型, 一个标量。

示例:

```
a = (/1, 2, 3, 4, 5/)
```

```
print(num(a.gt.3)) ; 返回值为 2
```

82. 函数 `onedtond(val, dims)`

功能说明:

将一维数组 `val` 转换为多维数组。

参数:

`val`, 任意数值类型的一维数组。

`dims`, 整型, 输出的多维数组的维数大小。

返回值:

数值类型同 `val`, 维数大小为 `dims`。如果输出数组的维数小于输入数组, 则会出

现警告,此时按先后顺序仅返回输入数组中与输出数组维数大小相匹配的数组。如果输出数组的维数大于输入数组,则用输入数组中的数值进行重复填充直至完全匹配输出数组的维数大小。

示例:

```
a = (/1, 2, 3, 4, 5, 6, 7, 8/)
b = onedtond(a, (/2, 4/))
print(b)      ; 返回值为(/(/1, 2, 3, 4/), (/5, 6, 7, 8/)/)
```

83. 函数 pot_vort_isobaric (p, u, v, t, lat, gridType, opt)

功能说明:

在全球直线网格点的等压面上计算位涡、静力稳定性和位温。

参数:

p, 数值型, 一维数值, 表示各气压层, 单位为 Pa。

u 和 v, 数值型, 直线网格点上的纬向及经向风场, 必须为三维数组 (lev, lat, lon) 或四维数组 (time, lev, lat, lon), 其纬度方向必须为从南至北。

t, 数值型, 数组维数和大小同 u 和 v, 表示气温, 单位为 K。

lat, 数值型, 一维数值, 表示从南至北的纬度坐标。

gridType, 整型, 0 表示高斯格点, 1 表示均匀间隔网格点。

opt, 整型, 0 表示仅返回位涡, 1 表示返回目录(list), 将包含变量位涡、静力稳定性、位温。

返回值:

数值类型、数组维数及大小同 u 和 v。

示例:

```
; 假定数据在纬度方向上从南至北存放
f  = addfile ("foo.nc", "r")
U  = f->U      ; (time, lev, lat, lon) 或者 (lev, lat, lon)
V  = f->V
T  = f->T      ; K
lat = f->lat
lev = f->lev    ; hPa
lev = lev * 100 ; 转换为 Pa
lev@units = "Pa"
```

; 如果 U/V/T 为从北往南存放, 则

```

;U   = U(:, :, ::-1, :)    ; 转换为从南往北存放
;V   = V(:, :, ::-1, :)
;T   = T(:, :, ::-1, :)

gridType = 0    ; 高斯格点
opt = 1
PV = pot_vort_isobaric(lev, U, V, T, lat, gridType, opt) ; 目录变量
pv = PV[0]    ; 位涡,  $\text{K m}^2 \text{ kg}^{-1} \text{ s}^{-1}$ 
s = PV[1]    ; 静力稳定度,  $\text{K Pa}^{-1}$ 
theta = PV[2] ; 位温

```

84. 程序 print(data)

功能说明:

输出一个变量或表达式的值。

参数:

data, 任意数值类型的数组变量或表达式。

85. 程序 printVarSummary(data)

功能说明:

输出一个变量的基本信息, 如类型、属性、维数大小、行列名称(如果存在)、坐标信息(如果存在)、缺测值等。

参数:

data, 任意数值类型的数组变量。

86. 函数 product(x)

功能说明:

计算数组 x 中所有数值的乘积。

参数:

x, 数值型, 任意维数的数组。

返回值:

数值类型同 x, 一个标量。缺测值被忽略。

示例:

```

x = (/ (/1, -99/), (/3, 4/), (/ -99, 6/)/)
x@_FillValue = -99

```

```
xproduct = product(x)
print(xproduct)      ; 返回值为  $1 \times 3 \times 4 \times 6 = 72$ 
```

87. 函数 reshape(val, dims)

功能说明:

将多维数组转换为另一多维数组(即改变维数大小,而数组大小不变)。

参数:

val,数值型,任意维数的数组。

dims,整型或长型,一维的正值数组,表示转换后的数组各维的大小。

示例:

;假定数组 u3d 的结构为 [time|360] X [lat|73] X [lon|144]

u4d = reshape(u3d, (/30,12,73,144/)) ; 即转换为 30 X 12 X 73 X 144 的四维数组

88. 函数 readAsciiTable(filename, ncol, data_type, opt)

功能说明:

读取 ASCII 文件 filename 中指定的行段。

参数:

filename,字符串型,任意维数的数组。

ncol,integer 或长型,文件中的总行数。

data_type,字符串型,文件中数值的类型。

opt,如果是整型或长型,则 opt 有 1 或 2 个数值。第 1 个数值 n1 表示文件前 n1 行将被忽略;第 2 个数值 n2 如果存在,则表示文件最后 n2 行将被忽略。如果 opt 为字符串型,则表示忽略从文件第 1 行开始至文件第 1 列中出现该字符串的行。

返回值:

数值类型为 data_type,一维或两维的数组。

示例:

给定一个名为“darwin”的文件,

DARWIN SEA LEVEL PRESS (1000 MB SUBTRACTED)

YEAR	JAN	APR	JUL	OCT
1951	5.3	9.4	13.3	11.4
1952	6.7	10.3	13.0	9.9
[... ...]				

```
ncol = 5
data = readAsciiTable("darwin", ncol, "float", 2); 读入数据文件,但跳过前
2 行
data = readAsciiTable("darwin", ncol, "float", "YEAR"); 读入数据文件,但
跳过第 1 行至第 1 列中出现“YEAR”的行,与上行代码效果相同
```

89. 函数 reg_multlin(y, x, option)

功能说明:

多元线性回归系数。

参数:

y, 数值型, 长度为 N 的一维数组, 包含因变量 y(N)。

x, 数值型, 维数大小为 (/M+1, N/) 的二维数组, M 为独立变量的个数。

option, 逻辑型, 暂不使用, 设为 False。

返回值:

如果 x 或 y 是双精度型, 返回双精度型, 否则返回浮点型。一维数组, 其维数大小与 x 最左边维相同。

示例:

假定已知 $y = 28 + 0.038 \times x_1 + 0.83 \times x_2$, 现根据 y 及 x1、x2 数值求取其回归系数,

```
y = (/40, 50, 50, 70, 65, 65, 80/)
N = dimsizes(y)
x1 = (/100, 200, 300, 400, 500, 600, 700/) ; 长度为 N
x2 = (/10, 20, 10, 30, 20, 20, 30/); 长度为 N
M = 2
X = new( (/M+1, N/), "float" )
X(0, :) = 1.0
X(1, :) = x1
X(2, :) = x2
beta = reg_multlin(y, X, False)
print(beta) ; 返回值为 beta (/28.09524, 0.03809524, 0.8333333/)
```

90. 函数 regCoef_n(x, y, dims_x, dims_y)

功能说明:

计算变量 x 和变量 y 之间的一元线性回归系数。

参数:

x 及 y , 数值型, 任意维数的数组。 x 数组的最后一维与 y 数组的最后一维大小必须相同。

dims_x 及 dims_y , 整型, 一个标量, 分别表示数组 x 的第 dims_x 维与数组 y 的第 dims_y 维用以计算回归系数。

返回值:

如果 x 或 y 是双精度型, 返回双精度型, 否则返回浮点型。同时以属性形式返回如下四个数值, ①Y 轴截距“ $yintercept$ ”; ②t 值统计量“ $tval$ ”; ③估计回归系数的标准差“ $rstd$ ”和 ④样本个数“ $nptxy$ ”。“ $tval$ ”与“ $rstd$ ”数值类型同返回值一致, “ $nptxy$ ”为整型。

示例:

给定一个浮点型的一维数组 $x(\text{ntim})$ 及一个浮点型的四维数组 $y(\text{ntim}, \text{nlev}, \text{nlat}, \text{nlon})$ 。计算 y 对 x 的回归系数。

```
rc = regCoef_n(x,y,0,0) ; rc 大小为(nlev, nlat, nlon)
```

; 若要进行显著性检验, 则需要调用 `student_t` 函数进一步计算

```
df = rc@nptxy-2 ; 自由度
```

```
tval = rc@tval ; t 值统计量
```

`prob = student_t(tval, df)` ; 若其值小于 0.05, 则表明通过置信度为 0.05 的显著性检验

91. 函数 regline(x, y)**功能说明:**

计算一维变量 x 与一维变量 y 之间的一元线性回归系数。如果需对多维数组进行回归, 则参考 `regCoef_n`。

参数:

x 和 y , 数值型, 长度相同的一维数组。

返回值:

如果 x 或 y 是双精度型, 返回双精度型, 否则, 返回浮点型。同时, 该函数还以属性形式返回如下数值:

- (1) `xave` (一个标量, float 或双精度型), x 的平均值;
- (2) `yave` (一个标量, float 或双精度型), y 的平均值;
- (3) `tval` (一个标量, float 或双精度型), t 值统计量;
- (4) `rstd` (一个标量, float 或双精度型), 估计回归系数的标准差;
- (5) `yintercept` (一个标量, float 或双精度型), $x=0$ 时对应的 y 值, 即 y 截距;

(6) `nptxy` (一个标量, 整型), 样本个数。

示例:

构建数组 `y` 对 `x` 的线型回归直线,

```
x = (/1190., 1455., 1550., 1730., 1745., 1770., 1900., 1920., 1960.,
2295., 2335., 2490., 2720., 2710., 2530., 2900., 2760., 3010./)
```

```
y = (/1115., 1425., 1515., 1795., 1715., 1710., 1830., 1920., 1970.,
2300., 2280., 2520., 2630., 2740., 2390., 2800., 2630., 2970./)
```

```
x@_FillValue = -999.
```

```
y@_FillValue = -999.
```

```
rc = regline(x, y) ; 返回值为 0.9745615
```

```
yReg = rc * x + rc@yintercept ; 构建线性回归直线, 其序列长度与 x 和 y 相同
```

92. 函数 `rtest(r, nr, opt)`

功能说明:

计算线性相关系数的双侧(two-tailed)概率值(用以显著性检验)。

参数:

`r`, 数值型, 表示线性相关系数的一个标量或任意维数的数组。

`nr`, 整型, 一个标量或与 `r` 相同维数的数组, 表明用于计算相关系数时所用到的样本数目。如果 `nr` 是一个标量, `r` 是任意维数的数组, 则在所有检验中使用标量 `nr`。`nr` 至少为 3。

`opt`, 整型, 暂不使用, 设置为零。

返回值:

如果输入数组 `r` 是双精度型, 返回双精度型, 否则, 返回浮点型。维数大小同 `r`。

示例:

```
pr = rtest(0.2, 100, 0)
```

```
print(pr) ; 返回值为 0.4603628, 由于  $pr > 0.1$ , 所以不能通过置信度为 0.1 的显著性检验
```

93. 函数 `runave_n_Wrap(x, nave, opt, dim)`

功能说明:

对数组 `x` 的第 `dim` 维等权重滑动平均, 并保留元数据。

参数:

`x`, 数值型, 一维或多维数组。

nave, 整型, 等权重滑动平均时所用的格点数目。

opt, 整型, 边界点选项(通常 $\text{opt} = 0$)。假定 N 为序列中最后一个格点, x_i 为输入序列, x_o 为输出序列, 则当 opt 设定为如下三种数值时, 其对应的计算过程分别为:

(1) $\text{opt} < 0$; 采用循环边界点

例 nave = 2

$$x_o(0) = (x_i(0) + x_i(1)) / \text{nave}$$

$$x_o(N) = (x_i(N) + x_i(0)) / \text{nave}$$

例 nave = 3

$$x_o(0) = (x_i(N) + x_i(0) + x_i(1)) / \text{nave}$$

$$x_o(N) = (x_i(N-1) + x_i(N) + x_i(0)) / \text{nave}$$

例 nave = 4

$$x_o(0) = (x_i(N) + x_i(0) + x_i(1) + x_i(2)) / \text{nave}$$

$$x_o(N) = (x_i(N-1) + x_i(N) + x_i(0) + x_i(1)) / \text{nave}$$

(2) $\text{opt} = 0$; 起始和结束点设置为 x_i 的缺测值

例 nave = 2

$$x_o(0) = (x_i(0) + x_i(1)) / \text{nave}$$

$$x_o(N) = x_i@_FillValue$$

例 nave = 3

$$x_o(0) = x_i@_FillValue$$

$$x_o(1) = (x_i(0) + x_i(1) + x_i(2)) / \text{nave}$$

$$x_i(N) = x_i@_FillValue$$

例 nave = 4

$$x_o(0) = x_i@_FillValue$$

$$x_o(1) = (x_i(0) + x_i(1) + x_i(2) + x_i(3)) / \text{nave}$$

$$x_o(N-2) = (x_i(N-3) + x_i(N-2) + x_i(N-1) + x_i(N)) / \text{nave}$$

$$x_o(N-1) = x_i@_FillValue$$

$$x_o(N) = x_i@_FillValue$$

(3) $\text{opt} > 0$; 采用反射(对称)条件

例 nave = 2

$$x_o(0) = (x_i(0) + x_i(1)) / \text{nave}$$

$$x_o(N) = (x_i(N) + x_i(N-1)) / \text{nave}$$

例 nave = 3

$$x_o(0) = (x_i(1) + x_i(0) + x_i(1)) / \text{nave}$$

$$xo(N) = (xi(N - 1) + xi(N) + xi(N - 1)) / nave$$

例 $nave = 4$

$$xo(0) = (xi(2) + xi(1) + xi(0) + xi(1)) / nave$$

$$xo(N) = (xi(N - 1) + xi(N) + xi(N - 1) + xi(N - 2)) / nave$$

dim , 整型, 一个标量, 对 x 的第 dim 维进行加权滑动平均。

返回值:

如果输入数组 x 是双精度型, 返回双精度型, 否则, 返回浮点型。返回数组维数同 x 。

示例:

给定一个数组 $x(ktimes, nlat, mlon)$, $ktimes=1000$, $nlat=64$, $mlon=128$ 。对经度维进行 3 点等权重滑动平滑, 其经度维为循环维。

$X = runave_n_Wrap(x, 3, -1, 2)$; 输出值为 $X(ktimes, nlat, mlon)$

94. 函数 short2flt(x)

功能说明:

将短型数据转换为浮点型。

参数:

x , 短型数组。

返回值:

返回数组的维数及大小同 x , 同时也保留元数据。

示例:

```
f = addfile("air.2m.gauss.1979.nc", "r")
t2m = short2flt(f->air)
```

95. 函数 sin(value)

用法与 \cos 一致, 但为计算 $value$ 的正弦值。

96. 函数 smth9_Wrap(x, p, q, wrap)

功能说明:

对数组 x 进行二维九点平滑, 并保留元数据。

参数:

x , 数值型, 二维或二维以上的数组。最右边两维将进行平滑。

p 和 q , 数值型, 控制平滑效果的两个标量。通常设 $p = 0.50$ 。若 $p = 0.50$, $q = -0.25$ 为“轻”平滑; 若 $p = 0.50$, $q = 0.25$ 为“重”平滑。若 $q = 0$, 则为二维五点平滑。

wrap, 逻辑型, 如果最右边维是循环维, 则设为 True; 否则设为 False。

返回值:

如果输入数组 x 是双精度型, 返回双精度型, 否则, 返回浮点型。返回数组维数大小同 x 。

示例:

给定一个数组 $x(nx, ny)$, $nx=20, ny=40$ 。

$X = \text{smth9_Wrap}(x, 0.50, -0.25, \text{False})$; 返回值为 $X(nx, ny)$

97. 函数 sqrt(value)

用法与 log 一致, 但为计算 value 的平方根。

98. 函数 stddev(x)

用法与 avg 一致, 但为计算数组 x 中所有数值的标准差。

99. 函数 student_t(t, df)

功能说明:

计算两侧(two-tailed)Student-t 分布概率值(用以显著性检验)。

参数:

t , 数值型, 表示 t 值的一个标量或数组。

df , 数值型, 一个标量或与 t 相同维数大小的数组, 表示自由度的个数。如果 t 是一个数组, df 是一个标量, 则 df 的维数大小需与 t 相同。

返回值:

如果输入数组 t 是双精度型, 返回双精度型, 否则, 返回浮点型。维数大小同 t 。

示例:

$t = (/12.7, 2.572, 2.228, 2.12, 2.086, 1.980, 1.96/)$

$df = (/1, 5, 10, 16, 20, 120, 500/)$; 维数大小同 t

$P = \text{student_t}(t, df)$

$\text{print}(P)$; 输出值为 $P(/0.05002433, 0.04991403, 0.05001178, 0.04999094, 0.04999636, 0.04999207, 0.05055078/)$, 可见, 只有第 1/3/4/5 个数(从第 0 个数开始)能通过 95% 的显著性检验。

100. 函数 sum(x)

用法与 avg 一致, 但为计算数组 x 中所有数值之和。

101. 函数 `svdcov(x, y, nsvd, homlft, hetlft, homrgt, hetrgt)`

功能说明:

奇异值分解, 返回数组 `x` 与数组 `y` 的左同质 `homlft`、左异质 `hetlft`、右同质 `homrgt` 和右异质 `hetrgt` 数组。输入场 `x` 和 `y` 中不可含有缺测值。

参数:

`x` 和 `y`, 浮点型或双精度型, 二维数组, 由空间维及时间维构成。

`nsvd`, 整型, 一个标量, 指定要返回的 SVD 模态的数目。

`homlft`(输出), 数值型, 左同质数组, 大小为 `nsvd` × 数组 `x` 的空间格点数, 用户必须预先定义该数组。

`hetlft`(输出), 数值型, 左异质数组, 大小为 `nsvd` × 数组 `x` 的空间格点数, 用户必须预先定义该数组。

`homrgt`(输出), 数值型, 右同质数组。大小为 `nsvd` × 数组 `y` 的空间格点数, 用户必须预先定义该数组。

`hetrgt`(输出), 数值型, 右异质数组。大小为 `nsvd` × 数组 `y` 的空间格点数, 用户必须预先定义该数组。

返回值:

如果输入数组 `x` 或 `y` 是双精度型, 返回双精度型, 否则, 返回浮点型。该函数返回每个模态的方差解释贡献, 是长度为 `nsvd` 的一维数组。此外, `homlft`、`hetlft`、`homrgt` 及 `hetrgt` 也为输出场。

示例:

```

ntime = 8           ; 时次
ncols = 3           ; 数组 s 的空间格点数
ncolz = 6           ; 数组 z 的空间格点数
nsvd = 3            ; 需返回的 SVD 模态的数目
xmsg = -999.9       ; 缺测值
s = asciiread("svdrdm_S.asc", (/ntime, ncols/), "float"); 读入数组 s
z = asciiread("svdrdm_Z.asc", (/ntime, ncolz/), "float"); 及 z
s!0 = "time"        ; 命名行列, 为行列调整做准备
s!1 = "col"
z!0 = "time"
z!1 = "col"
homlft = new((/nsvd, ncols/), float) ; 预先定义输出场
hetlft = new((/nsvd, ncols/), float)

```

```

homrgt = new((/nsvd,ncolz/),float)
hetrgt = new((/nsvd,ncolz/),float)
x = svdcov(s(col|:, time|:), z(col|:, time|:), nsvd, homlft, hetlft, hom-
rgt, hetrgt) ;对 s 与 z 的行列进行了调整
print("svdcov: percent variance= " + x) ; 屏幕上输出各个模态的方差解
释贡献

```

102. 函数 svdcov_sv(x, y, nsvd, svLeft, svRight)

功能说明:

奇异值分解,返回输入场 x 及 y 的左奇异向量 svLeft、右奇异向量 svRight。输入场 x 和 y 中不可含有缺测值。

参数:

x 和 y,浮点型或双精度型,二维数组,由空间维及时间维构成。

nsvd,整型,一个标量,指定返回奇异向量的数目。

svLeft(输出),浮点型或双精度型,左奇异向量。大小为 nsvd \times 数组 x 的空间格点数,用户必须预先定义该数组。

svRight(输出),浮点型或双精度型,右奇异向量。大小为 nsvd \times 数组 y 的空间格点数,用户必须预先定义该数组。

返回值:

如果输入数组 x 或 y 是双精度型,返回双精度型,否则返回浮点型。该函数返回长度为 nsvd 的一维数组,对应各模态的方差解释贡献。svLeft 及 svRight 分别为左右奇异向量,长度分别为 nsvd \times 数组 x 的空间格点数及 nsvd \times 数组 y 的空间格点数。

示例:

```

ntime = 8 ; 时次
ncols = 3 ; 数组 s 的空间格点数
ncolz = 6 ; 数组 z 的空间格点数
nsvd = 3 ; 需返回的 SVD 模态的数目
froot = "/fs/scd/home1/shear/ncldata_input/"
s = asciiread(froot+"svd_ex01_S.asc", (/ntime, ncols/), "float")
z = asciiread(froot+"svd_ex01_Z.asc", (/ntime, ncolz/), "float")
s!0 = "time"
s!1 = "col"
z!0 = "time"

```

```

z!1 = "col"
svLeft = new((/nsvd, ncols/), float) ; 需预先定义
svRight = new((/nsvd, ncolz/), float)
pcVar = svdcov_sv(s(col|:, time|:), z(col|:, time|:), nsvd, svLeft,
svRight) ; 通过命名的行列将变量 s 及 z 的时间维放至最右边。pcVar 为方差
解释贡献,pcVar@sv 为奇异向量值

```

103. 函数 `svdstd(x, y, nsvd, homlft, hetlft, homrgt, hetrgt)`

用法与 `svdcov` 一致,但该函数在进行 SVD 分解前会先对数组 `x` 和 `y` 进行标准化。

104. 函数 `svdstd_sv(x, y, nsvd, svLeft, svRight)`

用法与 `svdcov_sv` 一致,但该函数在进行 SVD 分解前会先对数组 `x` 和 `y` 进行标准化。

105. 程序 `system(command)`

功能说明:

执行一个 `shel` 命令。

参数:

`command`, 字符串型, 一个 `shell` 命令。

示例:

在屏幕上输出当前的日期和时间。

```
system("date") ; 返回值为“2016 年 07 月 01 日 10:55:43”
```

106. 函数 `systemfunc(command)`

用法与 `system` 一致,但返回一个字符串数组。

示例:

将当前时间和日期作为字符串数组返回。

```
TimeDate = systemfunc("date")
```

```
print(TimeDate)
```

; 返回值为

Variable: TimeDate

Type: string

Total Size:4 bytes

```

1values
Number of Dimensions: 1
Dimensions and sizes: [1]
Coordinates:
(0)2016 年 07 月 01 日 11:08:21

```

107. 函数 tan(value)

用法与 cos 一致, 但为计算 value 的正切值。

108. 函数 ttest(ave1, var1, s1, ave2, var2, s2, iflag, tval_opt)

功能说明:

返回显著性水平估计值和 t 值。

参数:

ave1 和 ave2, 数值型, 一个标量或任意维数的数组(它们维数大小必须相同)。它们分别是两个样本的均值。

var1 和 var2, 数值型, 一个标量或与 ave1 和 ave2 维数相同的数组。它们分别是两个样本的方差。

s1 和 s2, 数值型, 必须与 ave1 维数相同或者为一个标量, 表示样本的数目。如果两个样本内各自的数据显著相关, 则 s1 和 s2 需取为等价的样本大小(the equivalent sample sizes)。

iflag, 逻辑型。False 表示假定两个样本有相同的总体方差; True 表示假定它们有不同的总体方差。后者应用了改进的 Welsh 自由度 (Welsh's t-test)。

tval_opt, 逻辑型, True 表示返回 Student t-values 及统计概率; False 表示只返回统计概率。

返回值:

如果 ave1, var1, var2 或 ave2 是双精度型, 返回双精度型, 否则, 返回浮点型。如果 tval_opt 是 False, 则返回的数组将与 ave1 维数相同。否则, 返回的数组大小为 $2 \times \text{dimsizes}(\text{ave1})$ 。

示例:

给定一维数组 X 与 Y(两者的维数大小不一定相同)。假定 X 与 Y 中的每个值都是独立的, 且 X 与 Y 有相同的总体方差。

```

siglvl = 0.05           ; 设定置信度水平
aveX = avg(X)
aveY = avg(Y)

```

```

varX = variance(X)
varY = variance(Y)
sX = dimsizes(X)      ; X 与 Y 维数大小不一定相同
sY = dimsizes(Y)
iflag = False
;(1)
tval_opt=False
prob = ttest(aveX, varX, sX, aveY, varY, sY, iflag, False) ; prob 为一个
标量,其数值范围为(0, 1)。如果 prob<siglvl,则零假定(两组数来自同一个母体)
被拒绝,而接受其相反的假定
if(prob.lt.siglvl) then
;两组数之间差异显著,通过置信度为 siglvl 的显著性检验
end if
;(2)
;tval_opt=True
probt = ttest(aveX, varX, sX, aveY, varY, sY, iflag, True) ;此时 probt
为一个长度为 2 的 1 维数组,probt(0)为其概率,probt(1)为 t-value

```

109. 函数 uv2dv_cfd(u, v, lat, lon, boundOpt)

功能说明:

利用中央差分计算风场的散度场。

参数:

u 和 v,数值型,均匀经纬度网格点上的纬向及经向风场,最右边两维必须为纬度和经度。

lat 和 lon,数值型,u 和 v 数组的纬度及经度坐标值。

boundOpt,整型,边界条件的设定,有如下 4 个选项:

(1) 0,边界点设为缺测值。

(2) 1,u 和 v 数组在经度维上循环(但 u 和 v 数组不包含循环点)。南北边界点将被设置为缺测值。

(3) 2,所有边界点使用前差或后差差分格式。

(4) 3,u 和 v 数组在经度维上循环(u 和 v 数组不可包括循环点)。南北边界使用前差或后差差分格式。

返回值:

数值类型与维数大小同 u 和 v。

示例:

给定某个区域 $[20^{\circ}-60^{\circ}\text{N}, 90^{\circ}-140^{\circ}\text{E}]$ 内的纬向风场 $u(\text{ntim}, \text{nlat}, \text{mlon})$ 和经向风场 $v(\text{ntim}, \text{nlat}, \text{mlon})$, 计算风场散度场。

$\text{dv} = \text{uv2dv_cfd}(u, v, \text{lat}, \text{lon}, 2)$; 返回值为 $\text{dv}(\text{ntim}, \text{nlat}, \text{mlon})$

110. 函数 $\text{uv2vrdivF_Wrap}(u, v)$

功能说明:

利用球谐函数, 根据均匀经纬度网格点上的风场值 u 和 v 计算其涡度场及散度场, 并保留元数据。输入场 u 和 v 中不可含有缺测值。

参数:

u 和 v , 数值型, 纬向及经向风场(二维或二维以上, 最右边二维必须为 $\text{lat} \times \text{lon}$), 需覆盖全球, 其纬度维数值需单调递增。

返回值:

数值类型、数组维数及数组大小同 u 和 v 。

示例:

; 假定全球的纬向风场 u 和经向风场 v 均为三维数组($\text{ntim}, \text{nlat}, \text{mlon}$)

$\text{vrdiv} = \text{uv2vrdivF_Wrap}(u, v)$; 返回数组的结构为 $(2, \text{ntim}, \text{nlat}, \text{mlon})$

$\text{vr} = \text{vr}(0, :, :, :)$; 涡度场

$\text{dv} = \text{dv}(1, :, :, :)$; 散度场

111. 函数 $\text{uv2vrdivG_Wrap}(u, v)$

用法与 uv2dvF_Wrap 一致, 但为计算高斯网格点上风场的涡度与散度。

112. 函数 $\text{uv2vr_cfd}(u, v)$

用法与 uv2dv_cfd 一致, 但为计算涡度。

113. 函数 $\text{uv2sfvpF}(u, v)$

功能说明:

根据均匀经纬度点上的风场, 通过球谐函数计算流函数和速度势。输入场 u 和 v 中不可含有缺测值。

参数:

u 和 v , 数值型, 纬向和经向风场(二维或二维以上, 最右边二维必须为 $\text{lat} \times \text{lon}$), 需覆盖全球, 其纬度维的数值需单调递增。

返回值:

数值类型、数组维数及数组大小同 u 和 v 。

示例:

```
; 假定全球的纬向风场 u 和经向风场 v 均为三维数组(ntim, nlat, mlon)
sfvp = uv2sfvpF(u, v)      ; 返回数组的结构为(2, ntim, nlat, nlon)
sf    = sfvp(0, :, :, :)   ; 流函数
vp    = sfvp(1, :, :, :)   ; 速度势
```

114. 函数 uv2sfvpG(u, v)

用法与 uv2sfvpF 一致, 但为计算高斯网格点上的流函数和速度势。

115. 函数 vibeta(p, x, linlog, psfc, pbot, ptop)

功能说明:

计算变量 x 的垂直积分。

参数:

p , 数值型, 含有各气压层数值的多维数组。它最右边维的长度至少大于等于 3, 其方向为从下向上。

x , 数值型, 要积分的多维数组。如果 p 为一维数组, 则 x 最右边维的大小必须与 p 相同。否则, x 与 p 必须相同维数大小。 x 的气压维必须是自下而上, 即与 p 的顺序一致。

linlog, 数值型, 1 表示线型插值, 2 为对数插值。

psfc, 数值型, 地表气压的多维数组。与 x 相比, 无气压层这一维。

pbot, 数值型, 表示积分的下边界。

ptop, 数值型, 表示积分的上边界。

返回值:

如果 p 、 $psfc$ 或 x 是双精度型, 返回双精度型, 否则, 返回浮点型。维数大小同 $psfc$ 。

示例:

```
给定位于各气压层 p(17) 上某水平格点上的气温场为 x(17), 求其垂直积分值,
p = (/1000., 925., 850., 700., 600., 500., 400., 300., 250., 200., 150.,
100., 70., 50., 30., 20., 10./) ; 气压 (hPa): 从底部至顶部
x = (/292., 285., 283., 277., 270., 260., 250., 235., 225., 215., 207.,
207., 213., 220., 225., 228., 230./) ; 各个气压层上的温度
linlog = 1
psfc = 1013.
```

```

pbot = 1100.
ptop = 10
vint = vibeta(p, x, linlog, psfc, pbot, ptop) ; 单位 K · hPa
print("vint=" + vint) ; vint=255368 K · hPa

```

116. 函数 **vinth2p(datai, hbcofa, hbcofb, plevo, psfc, intyp, p0, ii, kxtrp)**

功能说明:

将 CAM(Community Atmosphere Model)模式的混合坐标系插值至气压坐标系中。由于混合坐标系是 σ 坐标系与气压坐标系的一种混合坐标系。所以,若参数 hbcofa 设为 0, p0 设为 1000, 该函数即可将变量从 σ 坐标系插值至气压坐标系中。

参数:

datai, 数值型, 数组维数为 3、4 或 5。必须有一维对应混合坐标层, 其方向为从上至下。最右边三维必须是 level X lat X lon [比如, T(time, lev, lat, lon)]。

hbcofa, 数值型, 与 datai 的高度层次维同大小的一维数组, 表示混合系数 A。其方向为从上至下。该数组中数值应标准化, 若否, 请除以参数 p0。

hbcofb, 同 hbcofa, 但为混合系数 B。

plevo, 数值型, 一维数组, 用以输出的气压层, 单位百帕。方向既可从上至下, 也可从下至上。

psfc, 数值型, 地表气压的多维数组, 单位为帕。它比 datai 少高度层次维。

intyp, 整型, 表示插值类型。1 为线性插值, 2 为对数插值, 3 为对数对数插值。

p0, 数值型, 一个标量, 表示地表参考气压, 单位为百帕。

ii, 目前暂未使用, 通常设置为 1。

kxtrp, 逻辑型。False 时表示当气压层在参数 psfc 范围外时, 不做外插。

返回值:

如果 p、psfc 或 x 是双精度型, 返回双精度型, 否则返回浮点型。维数大小同 datai, 但高度层次维已由 plevo 替换。元数据会自动保留。

示例 1:

```

; 混合坐标系插值至气压坐标系
fccm = addfile("dummy.ccm", "r") ;
pnew = (/ 850.0, 200.0 /) ; 转换后的气压层[hPa/mb]
hyam = fccm->hyam ; A 系数
hybm = fccm->hybm ; B 系数
T = fccm->T ; 假定 "T" 为 (lat, lev, lon)
psfc = fccm->PS ; 地表气压, 单位 Pa

```

```
P0mb = 1000. ; 参考地表气压,单位 hPa
Tnew = vint2p (T(lev|:,lat|:,lon|:),hyam,hybm,pnew,psfc,1,P0mb,1,
True)
```

示例 2:

```
;  $\sigma$  坐标系插值至气压坐标系
f = addfile ("dummy.nc" , "r") ; 假定是在  $\sigma$  坐标系下
pnew = (/ 850.0,200.0 /) ; 转换后的气压层[hPa/mb]
sigma = f->sigma ; sigma 即为 B 系数
fake = sigma ; 将 A 系数设为 0
fake = 0.0 ;
T = f->T ; 假定 "T" 为 (lat,lev,lon)
psfc = fccm->PS ; 地表气压,单位 Pa
P0mb = 1000. ; 参考地表气压,单位 hPa
fluxp = vint2p(flux,fake,sigma,plevs,ps,1,P0mb,1,True)
```

117. 函数 vr2uvF_Wrap(u, v)

用法与 dv2uvF_Wrap 一致,但为计算旋转风。

118. 函数 vr2uvG_Wrap(u, v)

用法与 dv2uvG_Wrap 一致,但为计算旋转风。

119. 函数 wavelet(y, mother, dt, param, s0, dj, jt看, npad, noise, isigtest, sigigtest, nadof)

功能说明:

对时间序列 y 进行小波变换并计算其显著性水平。

参数:

y,数值型,一维数组(长度为 N)。

mother,整型,指定母波的类型:

(1) 0 = 'Morlet'(通常,mother 设为 0)

(2) 1 = 'Paul'

(3) 2 = 'DOG'(高斯导数)

如果 mother<0 或>2,则默认用'Morlet'。

dt,数值型,y 数组的采样时间(通常,dt = 1.0)。

param,数值型,母波参数。如果 param<0,则自动用如下默认值替代:

母波为'Morlet'时,该值表示 k_0 (波数),默认 6。

母波为'Paul'时,该值表示 m (阶),默认 4。

母波为'DOG'时,该值表示 m (第 m 阶导数),默认 2。

s_0 ,数值型,小波的最小尺度,通常设为 $2 \times dt$ 。(注:为了准确地重建和计算方差,Morlet 母波设置为 $s_0 = dt$,而 Paul 母波则设置 $s_0 = dt/4$)。

dj ,数值型,离散尺度的间距,通常设为 0.25。值越小,分辨率越好,但计算更慢。

jt_{tot} ,整型,尺度的个数。尺度范围从 s_0 到 $s_0 \times 2^{\lceil (jt_{tot}-1) \times dj \rceil}$ 。最常见的设定是, $jt_{tot} = 1 + \text{floattointeger}(((\log_{10}(N \times dt / s_0)) / dj) / \log_{10}(2.0))$ 。

$npad$,整型,小波变换的(包括填充 padding)的总数点。通常,它的数值为 2 的乘方,必须大于等于 N 。如果 $npad > N$,零值将被填充至时间序列的末端。通常, $npad = N$,即无填充。

$noise$,整型。0 表明使用白噪声背景显著性检验,1 表明使用红噪声背景显著性检验。通常 $noise = 1$ 。

$isigtest$,整型。0 表示进行正则卡方检验,1 表示在全域小波谱中做“时间平均”检验。

$sigigtest$,数值型,置信度。通常,设为 0.05。

$nadof$,数值型,目前暂未使用,通常设置为零。

返回值:

如果数组 y 是双精度型,返回双精度型,否则,返回浮点型。返回三维数组(称之为波),维数大小为 $2 \times jt_{tot} \times N$ 。该数组为随尺度及时间变化的小波变换实部(0,::,)及虚部(1,::,)。

示例:

详见本书 6.3.4 节。

120. 函数 `wgt_areaave(q, wgt_y, wgt_x, opt)`

功能说明:

以 wgt_x 及 wgt_y 为权重计算数组 q 的区域平均值。

参数:

q ,数值型,二维或二维以上数组。若 q 位于地球球面上时,则 $q([\dots], lat, lon)$ 的最右边两维应为“纬度”(lat)和“经度”(lon),否则, $q([\dots], y, x)$ 最右边两维应为“y”与“x”。

wgt_y ,数值型,标量(通常为 1.0)或者是表示“lat”/“y”维的一维权重数组。

wgt_x ,数值型,标量(通常为 1.0)或者是表示“lon”/“x”维的一维权重数组。

opt ,整型,缺测值处理选项。 $opt = 0$,则计算区域平均时可有缺测值; $opt = 1$,

若 q 中存在缺测值,则在局地返回缺测值。

返回值:

如果输入数组是双精度型,返回双精度型,否则返回浮点型。若 q 是二维数组,则返回值为一个标量;否则,输出数组维数大小等于 $n-2$, n 为输入数组 q 的维数大小。

示例:

给定数组 $u(lat, lon)$, $nlat=64$, $m lon=128$, lat 和 lon 为高斯格点上的经度和纬度,权重为高斯权重。使用如下几种权重方法计算区域平均:(1) 格点所占网格面积;(2) 高斯权重;(3) 纬度的余弦。

```
lat = f->lat
lon = f->lon
jlat = dimsizes( lat )
rad = 4.0 * atan(1.0)/180.0
re = 6371220.0
rr = re * rad ; 转换为弧度
dlon = abs(lon(2)-lon(1)) * rr ; 经度间隔
dx = dlon * cos(lat * rad)
dy = new(jlat, typeof(dx))
dy(0) = abs(lat(2)-lat(1)) * rr
dy(1:jlat-2) = abs(lat(2:jlat-1)-lat(0:jlat-3)) * rr * 0.5
dy(jlat-1) = abs(lat(jlat-1)-lat(jlat-2)) * rr
; (1) 格点面积为权重
area = dx * dy
uAve_area = wgt_areaave(u, area, 1.0, 1)
; (2) 高斯格点权重
gwgt = f->gwgt
uAve_gwgt = wgt_areaave(u, gwgt, 1.0, 1) ; (2)高斯格点为权重
; (3) 纬度余弦为权重
clat = cos(lat * rad)
uAve_clat = wgt_areaave(u, clat, 1.0, 1)
print(" uAve_area = " + uAve_area + " uAve_gwgt = " + uAve_gwgt +
"uAve_clat = " + uAve_clat) ; 返回为 uAve_area=15.1826 uAve_gwgt=15.1828
uAve_clat = 15.1818
```

121. 函数 `wgt_runave_n_Wrap(x, wgt, opt, dim)`

功能说明:

对数组 `x` 第 `dim` 维以 `wgt` 为权重进行滑动平均,并保留元数据。

参数:

`x`, 数值型, 一维或多维数组。

`wgt`, 数值型, 一维权重数组。其序列长度通常为奇数。

`opt`, 整型, 端点选项(通常 `opt = 0`)。假定 `N` 为序列中最后一个格点, `xi` 为输入序列, `xo` 为输出序列, 则当 `opt` 设定为如下三种数值时, 其具体计算过程分别为:

(1) `opt < 0` ; 采用循环边界点

例 `nave = 2`

$$xo(0) = w(0) \times xi(0) + w(1) \times xi(1)$$

$$xo(N) = w(0) \times xi(N) + w(1) \times xi(0)$$

例 `nave = 3`

$$xo(0) = w(0) \times xi(N) + w(1) \times xi(0) + w(2) \times xi(1)$$

$$xo(N) = w(0) \times xi(N-1) + w(1) \times xi(N) + w(2) \times xi(0)$$

例 `nave = 4`

$$xo(0) = w(0) \times xi(N) + w(1) \times xi(0) + w(2) \times xi(1) + w(3) \times xi(2)$$

$$xo(N) = w(0) \times xi(N-1) + w(1) \times xi(N) + w(2) \times xi(1) + w(3) \times xi(2)$$

(2) `opt = 0` ; 起始和结束点设置为 `xi` 的缺测值

例 `nave = 2`

$$xo(0) = w(0) \times xi(0) + w(1) \times xi(1)$$

$$xo(N) = xi_@_FillValue$$

例 `nave = 3`

$$xo(0) = xi_@_FillValue$$

$$xo(1) = w(0) \times xi(0) + w(1) \times xi(1) + w(2) \times xi(2)$$

$$xi(N) = xi_@_FillValue$$

例 `nave = 4`

$$xo(0) = xi_@_FillValue$$

$$xo(1) = w(0) \times xi(0) + w(1) \times xi(1) + w(2) \times xi(2) + w(3) \times xi(3)$$

$$xo(N-2) = w(0) \times xi(N-3) + w(1) \times xi(N-2) + w(2) \times$$

$$xi(N - 1) + w(3) \times xi(N)$$

$$xo(N - 1) = xi@_FillValue$$

$$xo(N) = xi@_FillValue$$

(3) $opt > 0$;利用反射(对称)条件

例 $nave = 2$

$$xo(0) = w(0) \times xi(0) + w(1) \times xi(1)$$

$$xo(N) = w(0) \times xi(N) + w(0) \times xi(0)$$

例 $nave = 3$

$$xo(0) = w(0) \times xi(1) + w(1) \times xi(0) + w(2) \times xi(1)$$

$$xo(N) = w(0) \times xi(N - 1) + w(1) \times xi(N) + w(2) \times xi(N - 1)$$

例 $nave = 4$

$$xo(0) = w(0) \times xi(1) + w(1) \times xi(0) + w(2) \times xi(1) + w(3) \times xi(2)$$

$$xo(N) = w(0) \times xi(N - 1) + w(1) \times xi(N) + w(2) \times xi(0) + w(3) \times xi(2)$$

dim , 整型, 一个标量, x 数组中给定维。值从零开始。

返回值:

如果输入数组是双精度型, 返回双精度型, 否则, 返回浮点型。维数大小同 x 。

示例:

给定一个数组 $x(nlat, mlon, ktimes)$, $nlat=64, mlon=128, ktimes=1000$ 。计算三点平滑, 令 $opt = 0$ 。

$X = wgt_runave_n_Wrap(x, (/0.25, 0.50, 0.25/), 0, 2)$; 返回值为 $X(nlat, mlon, ktimes)$

122. 函数 `wgt_vert_avg_beta(p, datai, psfc, punits, opt)`

功能说明:

以气压层厚度和 β 因子计算变量 $datai$ 的垂直平均或和。

参数:

p , 数值型, 含有各气压层数值的多维数组。如果它是一维数组, 则它表示 $datai$ 的气压层坐标数值。如果它是一个多维数组, 则它必须与 $datai$ 具有相同的维数大小和排列顺序。气压层的顺序(比如自上而下或自下而上)必须与 $datai$ 的顺序一致。

$datai$, 数值型, 三维或四维数组。最右边三维必须是 $level \times lat \times lon$ 。

$psfc$, 数值型, 地表气压的多维数组。与 $datai$ 相比, $psfc$ 少气压层这一维。

$punits$, 整型, 一个标量, 表示 p 和 $psfc$ 的单位。 $punits = 0$, 表示 hPa 或 mb,

punits = 1, 表示 Pa。

opt, 整型。其值有如下两种设定:

(1) 一个标量。如果 opt = 0, 计算数组 datai 在垂直方向上的以气压层厚度为权重的和; 如果 opt = 1, 则计算数组 datai 在垂直方向上的以气压层厚度为权重的平均值。

(2) 长度为 3 的一维数组。opt(0) 与 (1) 中 opt 为一个标量时的设定一致, 即 opt(0) = 0, 计算权重和; opt(0) = 1, 计算权重平均值。opt(1) 和 opt(2) 是用户指定某气压层的起始与结束范围, 用以计算垂直积分或垂直和。

返回值:

如果 psfc 或 datai 是双精度型, 返回双精度型, 否则返回浮点型。

示例 1:

给定一个数组 T(time, lev_p, lat, lon) 和 ps(time, lat, lon), 单位为 hPa, 其中 lev_p = (/1, 2, 3, 5, 7, 10, 20, 30, 50, 70, 100, 150, 200, 250, 300, 400, 500, 600, 700, 775, 850, 925, 1000/)。

; 以下 5 行的设定不是必须

lev_p!0 = "lev_p" ; 行列命名

lev_p&lev_p = lev_p ; 创建坐标变量

lev_p@long_name = "pressure" ; 添加属性

lev_p@units = "hPa" ; 单位

lev_p@positive = "down" ; 方向

T_wva = wgt_vert_avg_beta (lev_p, T, PS, 0, 1) ; 返回值为 T_wva (time, lat, lon), 即垂直平均的 T 值

如果变量 T 有坐标变量 lev_p, 则上式也可写为:

T_wva = wgt_vert_avg_beta(T&lev_p, T, PS, 0, 1)

示例 2:

同示例 1, 但仅计算 (1) 10 至 100hPa 和 (2) 650 至 900 hPa 范围内的权重平均值。

opt = (/ 1, 10, 100 /)

T_wva_10_100 = wgt_vert_avg_beta(lev_p, T, PS, 0, opt)

opt = (/ 1, 650, 900 /)

T_wva_650_900 = wgt_vert_avg_beta(lev_p, T, PS, 0, opt)

123. 函数 where(condtnl_expr, true_value, false_value)

功能说明:

根据逻辑数组 condtnl_expr 的 True 值和 False 值分别返回 true_value、false_value。

参数:

condtnl_expr, 逻辑型, 任意维数的数组或表达式。

true_value, 与 condtnl_expr 相同维数大小的数组或值。

false_value, 与 condtnl_expr 相同维数大小的数组或值。

返回值:

逻辑型, 数组维数大小与 condtnl_expr 一致。

示例:

`q = where(x.lt.0, x+200, x@_FillValue)` ; 对于 `x` 中小于 0 的数值, 则返回 `x+200` 至 `q` 数组对应位置上; 对于 `x` 中大于等 0 的数值, 则返回 `x` 的缺测值至 `q` 数组对应位置上。

124. 程序 write_matrix(data, fmtf, option)

功能说明:

按指定格式 `fmtf` 输出二维数组 `data` 至屏幕或 ASCII 文件。

参数:

`data`, 数值型, 要输出的二维数组。

`fmtf`, 字符串型, 指定输出的格式。所使用的字符串格式必须与变量类型一致。

`option`, 逻辑型, `option = False` 时, 根据指定的格式标准输出; `option = True` 时, 则该变量可有如下属性:

(1)`fout`, 指定输出文件名的字符串。如果不存在, 标准输出 (即在屏幕上显示)。
`fout` 的最大长度是 96 个字符。

(2)`title`, 标题的字符串。如果不存在, 则无标题。

(3)`tspace`, 整数, 标题之前的空格数。只适用于属性 `title` 存在的时候。如果设定属性 `title` 而不设定属性 `tspace`, 则标题将靠左对齐。默认是 `tspace = 0`。

(4)`row`, 若设定为 `True`, 则打印行数, 最大行数为 99999。如果不设定 `row` 或 `row` 为 `False`, 则不输出行编号。

示例:

```
N = 5
M = 7
ave = 0.0
std = 5.0
x = random_normal(ave, std, (/N, M/))
opt = True
opt@title = "Normal"
```

```
opt@tspace = 22
write_matrix(x, "7f7.2", opt)
print(x)
;返回值为
```

Normal						
4.35	4.36	9.73	4.91	1.77	-0.63	-4.29
4.39	4.66	-5.84	4.59	3.68	-14.12	0.07
0.27	3.77	0.89	-3.09	5.08	-2.51	5.85
-3.35	-1.66	8.46	7.55	0.14	1.76	0.87
-6.90	4.06	10.39	4.56	-5.63	-1.43	8.65

附录 E 常用绘图函数

本附录只例举部分函数,更多函数请参考 NCL 官网内容(附录 A 节)。

1. 函数 ColorNegDashZeroPosContour (plot, ncolor, zcolor, pcolor)

功能说明:

用实线表示 0 值及正值等值线、虚线表示负值等值线,同时用不同颜色表示负值、正值以及 0 值等值线。

参数:

- plot,要修改的图形。
- ncolor,绘制负值等值线所用颜色。
- zcolor,绘制 0 值等值线所用颜色。

示例:

详见 2.3.5 节与 5.3.1 节。

2. 函数 gsn_add_polygon(wks, plot ,x, y, res)

功能说明:

在图形 plot 中添加多边形。

参数:

- wks,工作站标识等(通常由函数 gsn_open_wks 创建)。
- plot,要添加多边形的图形。
- x 和 y,构成多边形各个端点的 X 和 Y 坐标(1 维数组)。如果是在地图中绘制,则 x 和 y 分别为经度和纬度。

res, 修改图形的相关绘图参数 (见附录 G)。

其他说明:

返回变量名必须唯一。x 与 y 的第一个数值与最后一个数值必须相同, 即最后一个点的位置与第一个点的位置重合, 使得多边形重合。

示例:

详见 2.3.8 节。

3. 函数 gsn_add_polyline(wks, plot ,x, y, res)

功能说明:

在图形 plot 中添加任意折线。

参数:

wks, 工作站标识符 (通常由函数 gsn_open_wks 创建)。

plot, 要添加任意折线的图形。

x 和 y, 构成任意折线的各个点的 X 和 Y 坐标 (1 维数组)。如果是在地图中绘制, 则 x 和 y 分别为经度和纬度。

res, 修改图形的相关绘图参数 (见附录 G)。

其他说明:

返回变量名必须唯一。

示例:

详见 2.3.8 节。

4. 函数 gsn_add_polymarker(wks, plot ,x, y, res)

功能说明:

在图形 plot 中添加标识。

参数:

wks, 工作站标识符 (它通常由函数 gsn_open_wks 创建)。

plot, 要添加标记的图形。

x 和 y, 添加标记的 X 和 Y 坐标 (1 维数组)。如果是在地图中绘制, 则 x 和 y 分别为经度和纬度。

res, 修改图形的相关绘图参数 (见附录 G)。

其他说明:

返回变量名必须唯一。

示例:

详见 2.3.8 节、5.3.4 节。

5. 函数 `gsn_add_shapefile_polygons(wks, plot, shp_name, res)`

功能说明:

依据 shapefile 文件中的数据信息在图形 plot 中添加多边形。

参数:

wks, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

plot, 要添加多边形的图形。

shp_name, 包含多边形位置信息的文件(后缀为 .shp 的文件)。

res, 修改图形的相关绘图参数(见附录 G)。

其他说明:

返回变量名必须唯一。

6. 函数 `gsn_add_shapefile_polylines(wks, plot, shp_name, res)`

同 `gsn_add_shapefile_polygons`, 但为添加任意折线。相关绘图参数见附录 G。

7. 函数 `gsn_add_shapefile_polymarkers(wks, plot, shp_name, res)`

同 `gsn_add_shapefile_polygons`, 但为添加标识。相关绘图参数见附录 G。

8. 函数 `gsn_add_text(wks, plot, text, x, y, res)`

同 `gsn_add_polyline`, 但为添加文本字符串。相关绘图参数见附录 G。

示例:

详见 2.3.8 节与 3.3.4 节。

9. 函数 `gsn_attach_plots(base_plot, plots, res_base, res_plots)`

功能说明:

在图形 base_plot 旁添加一系列附属图形 plots(使各图形连接在一起)。

参数:

base_plot, 图形文件。

plots, 需添加至 base_plot 某坐标轴上的其他附属图形文件。

res_base, 修改 base_plot 的相关绘图参数。

res_plots, 修改 plots 的相关绘图参数。

其他说明:

默认是将附属图形 plots 添加至图形 base_plot 右边 Y 轴的右侧。如果设定 `res_base@gsnAttachPlotsXAxis=True`, 则附属图形的上部 X 轴将与前一幅图形的底

部 X 轴重合。每次使用该函数时,函数的返回变量名必须唯一。

示例:

详见 6.3.4 节。

10. 函数 `gsn_contour_shade(plot, lowval, highval, opt)`

用法说明:

用不同的颜色或图案填充图形 `plot` 中等值线的高值区和低值区。

参数:

`plot`, 需要填充的图形。

`lowval/highval`, 等值线的最小值/最大值(用于指定填充的范围)。

`opt`, 修改图形的相关绘图参数。有如下 4 种:

(1)`opt@gsnShadeFillType`: 可以为 "color" 用颜色填充, 或 "pattern" 用形状填充。默认是 "color"。注意, "color" 或 "pattern" 不可大写。

(2)`opt@gsnShadeLow`: 一个整型数值或字符串(比如 "blue"), 表明用哪种颜色或形状填充小于或等于 `lowval` 的区域。

(3)`opt@gsnShadeMid`: 一个整型数值或字符串(比如 "blue"), 表明用哪种颜色或形状填充大于等于 `lowval` 及小于等于 `highval` 之间的区域。

(4)`opt@gsnShadeHigh`: 一个整型数值或字符串(比如 "blue"), 表明用哪种颜色或形状填充大于或等于 `highval` 的区域。

示例:

详见 5.3.3 节与 6.3.4 节。

11. 函数 `gsn_csm_attach_zonal_means(wks, map, data, res)`

用法说明:

在一个等值线图/地图的右侧添加一个纬向平均图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`map`, 要添加纬向平均图的地图。

`data`, 用于计算纬向平均的数组。

`res`, 修改图形的相关绘图参数(见附录 G)。

12. 函数 `gsn_csm_blank_plot(wks, res)`

用法说明:

绘制一个刻度线指向外侧的空白图形。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。
res, 修改图形的相关绘图参数。

13. 函数 gsn_csm_contour(wks, data, res)

用法说明:

绘制等值线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。
data, 一维或者二维数组。
res, 修改图形的相关绘图参数(见附录 G)。

示例:

详见 2.3.9 节与 5.3.1 节。

14. 函数 gsn_csm_contour_map(wks, data, res)

用法说明:

在地图上绘制等值线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。
data, 一维或者二维数组。如果是两维, 则最左边维对应 Y 轴(纬度)。
res, 修改图形的相关绘图参数(见附录 G)

其他说明:

默认为地图采用圆柱投影。

示例:

详见 2.3.6 节、6.3.1 节、6.3.2 节和 6.3.3 节。

15. 函数 gsn_csm_contour_map_ce(wks, data, res)

用法说明:

在圆柱投影地图上绘制等值线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。
data, 一维或者二维数组。如果数据是两维, 则最左边维对应 Y 轴(纬度)。
res, 修改图形的相关绘图参数(见附录 G)。

示例:

详见 2.3.2 节。

16. 函数 `gsn_csm_contour_map_overlay(wks, data1, data2, res1, res2)`

用法说明:

在地图上叠加两组等值线图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`data1` 和 `data2`, 第 1 幅和第 2 幅等值线图的数据;必须是一维或者二维。如果数据为二维,则最左边维对应 Y 轴(纬度)。

`res1`, 修改第一个图形(含地图)的相关绘图参数(见附录 G)。

`res2`, 修改第二个图形的相關绘图参数(见附录 G)。

其他说明:

修改地图的相关绘图参数只能在 `res1` 中设定,不可在 `res2` 中设定。

17. 函数 `gsn_csm_contour_map_polar(wks, data, res)`

用法说明:

在极射赤面投影地图上绘制等值线图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`data`, 二维数组。最左边维对应 Y 轴(纬度)。

`res`, 修改图形的相关绘图参数(见附录 G)。

示例:

详见 5.3.1 节。

18. 函数 `gsn_csm_hov(wks, data, res)`

用法说明:

绘制 Hovmueller(时间—经度)图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`data`, 二维数组。最右边维对应经度。

`res`, 修改图形的相关绘图参数(见附录 G)。

19. 函数 `gsn_csm_lat_time(wks, data, res)`

用法说明:

绘制时间—纬度图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`data`, 二维数组。最左边维对应纬度, 最右边维对应时间。

`res`, 修改图形的相关绘图参数(见附录 G)。

20. 函数 `gsn_csm_map(wks, res)`

用法说明:

绘制地图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`res`, 修改图形的相关绘图参数(见附录 G)。

其他说明:

默认为地图采用圆柱投影。

示例:

详见 2.3.8 节。

21. 函数 `gsn_csm_map_ce(wks, res)`

用法说明:

绘制圆柱投影地图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`res`, 修改图形的相关绘图参数(见附录 G)。

其他说明:

默认 0 度为地图的中心经度。

22. 函数 `gsn_csm_map_polar(wks, res)`

用法说明:

绘制极射赤面投影地图。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

res, 修改图形的相关绘图参数(见附录 G)。

其他说明:

默认北半球。

23. 函数 gsn_csm_pres_hgt(wks, data, res)

用法说明:

绘制经度、纬度或者时间随气压/高度的垂直剖面图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

data, 二维数组, 最左边维对应气压。

res, 修改图形的相关绘图参数(见附录 G)。

示例:

详见 5.3.3 节。

24. 函数 gsn_csm_pres_hgt_streamline(wks, data, xcomp, zcomp, res)

用法说明:

在气压/高度剖面图中绘制流线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

data, 二维数组, 最左边维为气压。

xcomp 和 zomp, 矢量场的 x 及 z 分量, 必须均为二维数组。

res, 修改图形的相关绘图参数(见附录 G)。

25. 函数 gsn_csm_pres_hgt_vector(wks, data, xcomp, zcomp, res)

用法说明:

在气压/高度剖面图中绘制箭头。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

data, 二维数组。最左边维为气压。

xcomp 和 zomp, 矢量场的 x 及 z 分量, 必须均为二维数组。

res, 修改图形的相关绘图参数(见附录 G)。

26. 函数 gsn_csm_streamline(wks, u, v, res)

用法说明:

绘制流线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

u 和 v, 二维数组。

res, 修改 streamline 的相关绘图参数(见附录 G)。

27. 函数 gsn_csm_streamline_contour_map(wks, u, v, data, res)

用法说明:

在地图上同时绘制等值线图和流线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

u 和 v, 二维数组。

data, 二维数组。最左边一维代表 Y(纬度)轴。

res, 修改图形的相关绘图参数(见附录 G)。

28. 函数 gsn_csm_streamline_contour_map_ce(wks, u, v, data, res)

用法说明:

在圆柱投影地图上同时绘制流线图和等值线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

u 和 v, 二维数组。

data, 二维数组。最左边维对应 Y 轴(纬度)。

res, 修改图形的相关绘图参数(见附录 G)。

29. 函数 gsn_csm_streamline_contour_map_polar(wks, u, v, data, res)

用法说明:

在极射赤面投影地图上同时绘制等值线图和流线图。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

u 和 v, 二维数组。

data, 二维数组, 最左边维对应 Y 轴(纬度)。

res, 修改图形的相关绘图参数(见附录 G)。

其他说明:

默认北半球。

30. 函数 `gsn_csm_streamline_map(wks, u, v, res)`

用法说明：

在地图上绘制流线图。

参数：

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`u` 和 `v`, 二维数组。

`res`, 修改图形的相关绘图参数(见附录 G)。

31. 函数 `gsn_csm_streamline_map_ce(wks, u, v, res)`

用法说明：

在圆柱投影地图上绘制流线图。

参数：

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`u` 和 `v`, 二维数组。

`res`, 修改图形的相关绘图参数(见附录 G)。

32. 函数 `gsn_csm_vector_scalar_map_polar(wks, u, v, data, res)`

用法说明：

在极射赤面投影地图上同时绘制矢量图及标量图。

参数：

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`u` 和 `v`, 二维数组。

`data`, 二维数组。

`res`, 修改图形的相关绘图参数(见附录 G)。

其他说明：

设置等值线时, 需设定 `res@gsnScalarContour = True`。

33. 函数 `gsn_csm_x2y(wks, x1, x2, y, res1, res2)`

用法说明：

绘制两个 X 轴的 XY 图。

参数：

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`x1`, 第一条折线的 X 坐标, 位于图形底部。

x2, 第二条折线的 X 坐标, 位于图形顶部。

y, 两条折线公用的 Y 坐标。

res1, 修改第一条折线以及图形中其他要素的相关绘图参数 (见附录 G)。

res2, 修改第二条折线的相关绘图参数 (见附录 G)。

34. 函数 `gsn_csm_x2y2(wks, x1, x2, y1, y2, res1, res2)`

用法说明:

绘制有两个 X 轴和两个 Y 轴的 XY 图。

参数:

wks, 工作站标识符 (它通常由函数 `gsn_open_wks` 创建)。

x1, 第一条折线的 X 坐标, 位于图形底部。

x2, 第二条折线的 X 坐标, 位于图形顶部。

y1, 对应 x1 的 Y 坐标, 位于图形左侧。

y2, 对应 x2 的 Y 坐标, 位于图形右侧。

res1, 修改与第一条折线以及整个图形的相关绘图参数 (见附录 G)。

res2, 修改第二条折线的相关绘图参数 (见附录 G)。

35. 函数 `gsn_csm_xy(wks, x, y, res)`

用法说明:

绘制 XY 图。

参数:

wks, 工作站标识符 (它通常由函数 `gsn_open_wks` 创建)。

x/y, 折线的 X 坐标和 Y 坐标。如果是二维, 则最左边维的大小对应着折线的数目。

res, 修改折线的相关绘图参数 (见附录 G)。

示例:

详见 2.3.4 节、3.3.2 节、3.3.3 节、3.3.4 节和 5.3.2 节。

36. 函数 `gsn_csm_xy2(wks, x, y1, y2, res1, res2)`

用法说明:

绘制有两个 Y 轴的 XY 图。

参数:

wks, 工作站标识符 (它通常由函数 `gsn_open_wks` 创建)。

x, 两根折线公用的 X 坐标。

y1, 第一根折线的 Y 坐标, 位于图形左侧。

y2, 第二根折线的 Y 坐标, 位于图形右侧。

res1, 修改与第一根折线以及图形中其他要素的相关绘图参数(见附录 G)。

res2, 修改第二根折线的相关绘图参数(见附录 G)。

37. 函数 `gsn_csm_xy3(wks, x, yL, yR, yR2, resL, resR, resR2)`

用法说明:

绘制有三个 Y 轴的 XY 图。

参数:

wks, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

x, 三根折线的 X 坐标。

yL, 第一根折线的 Y 坐标, 位于图形左侧。

yR, 第二根折线的 Y 坐标, 位于图形右侧。

yR2, 第三根折线的 Y 坐标, 独立于图形外, 平行于 yL 和 yR 轴, 位于 yR 坐标轴的右侧。

resL, 修改与第一根折线以及图形中其他要素的相关绘图参数(见附录 G)。

resR, 修改第二根折线的相关绘图参数(见附录 G)。

resR2, 修改第三根折线的相关绘图参数(见附录 G)。

38. 函数 `gsn_csm_y(wks, y, res)`

用法说明:

绘制 XY 图, 其中 X 轴的数值为数组 y 中各个数值的下标变量(从 0 开始)。

参数:

wks, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

y, 折线的 Y 坐标值。如果是二维, 则最左边维对应着折线的数目。

res, 修改图形的相关绘图参数(见附录 G)。

示例:

详见 2.3.1 节和 2.3.3 节。

39. 程序 `gsn_define_colormap(wks, color_map)`

用法说明:

更换默认色板。

参数:

wks, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`color_map`, 可以是已命名的颜色, 也可以是一组 RGB 的数组, 或者是预先定义好的色板。

示例:

详见 2.3.2 节。

40. 程序 `gsn_draw_colormap(wks)`

功能说明:

绘制当前所用色板。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

41. 程序 `gsn_labelbar_ndc(wks, nboxes, labels, x, y, res)`

功能说明:

在单位坐标系内绘制色标。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`nboxes`, 图形颜色的个数。

`labels`, 表示各种颜色的字符串数组。

`x` 和 `y`, 在单位坐标系内, 色标左上角的 `x` 及 `y` 坐标位置。

`res`, 修改图形的相关绘图参数。值得注意的是 `nboxes` 和 `labels` 数组大小不一定相同, 这取决于色标和文字搭配情况。(见附录 G)

示例:

详见 3.3.4 节。

42. 程序 `gsn_legend_ndc(wks, nitems, labels, x, y, res)`

功能说明:

在单位坐标系内绘制图例。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`nitems`, 图例的数量。

`labels`, 图例名称的字符串数组。

`x` 和 `y`, 在单位坐标系内, 图例左上角的 `x`, `y` 坐标位置。

`res`, 修改图形的相关绘图参数(见附录 G)。

43. 程序 `gsn_merge_colormaps(wks, color_map1, color_map2)`

功能说明:

合并两个色板。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`color_map1` 和 `color_map2`, 需要合并的色板。它们可以是预先定义好的色板, 也可以是一组 RGB 数组, 也可以是一组颜色名称的字符串数组。

44. 函数 `gsn_open_wks(type, name)`

功能说明:

建立一个图形文件。

参数:

`type`, 绘制图形文件的类型, 如 `ps`, `eps`, `pdf`, `x11`, `png`, `ncgm`。

`name`, 图形文件的名字。

45. 程序 `gsn_panel(wks, plot, dims, res)`

功能说明:

在一个页面中绘制及排列多个大小相同的图形。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`plot`, 大小相同的多个图形。

`dims`, 表明多个图形 `plot` 排列方式的数组。可以为两个数字(默认), 分别代表行和列的数目; 也可以通过设定属性 `gsnPanelRowSpec` 为 `True`, 分别指定每行中绘制图形的数目, 此时 `dims` 为 1D 数组。

`res`, 修改图形的相关绘图参数。画图顺序从左到右, 从上到下。(见附录 G)

示例:

详见 2.3.9 节和 4.3.1 节。

46. 程序 `gsn_polygon_ndc(wks, x, y, res)`

功能说明:

在单位坐标系内绘制多边形。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

x 和 y , 多边形各个端点在单位坐标系内的 X 和 Y 轴坐标值(1 维数组)。
 res , 修改图形的相关绘图参数(见附录 G)。

示例:

; 在单位坐标系中绘制一个三角形, 用颜色填充。注意, 函数 `gsn_polygon_ndc` 在绘制三角形时需设定三个顶点的位置, 这与函数 `gsn_add_polygon` 需设定四个点的位置不同(第一点与第四个点的位置重合)

```

cirx=(/0.415, 0.326, 0.225/)           ; x 轴坐标位置
ciry=(/0.846, 0.898, 0.880/)           ; y 轴坐标位置
gsres=True
gsres@gsFillColor=4                     ; 修改其填充颜色.
gsn_polygon_ndc(wks, cirx, ciry, gsres) ; 绘图
frame(wks)                             ; 翻页

```

47. 程序 `gsn_polyline_ndc(wks, x, y, res)`

同 `gsn_polygon_ndc`, 但为添加任意折线。相关绘图参数见附录 G。

48. 程序 `gsn_polymarker_ndc(wks, x, y, res)`

同 `gsn_polygon_ndc`, 但为添加标识。相关绘图参数见附录 G。

49. 程序 `gsn_text_ndc(wks, text, x, y, res)`

同 `gsn_polygon_ndc`, 但为添加字符串, 相关绘图参数见附录 G。

50. 程序 `overlay(base_id, transform_id)`

功能说明:

合并图形。

参数:

`base_id`, 底图。地图可绘制在该图形中。

`transform_id`, 要叠加至底图的图形。该图形中不可有地图。可通过连续调用该函数, 叠加多个图形。

示例:

详见 4.3.1 节、5.3.1 节和 5.3.2 节。

51. 函数 `read_colormap_file(filename)`

功能说明:

读取 NCL 的色板文件。

参数:

filename, 色板文件。

返回值:

数组大小为 $n \times 4$ 的 2 维浮点型数组。 n 为色板文件中颜色的数目, 4 为红色、绿色、蓝色和透明数值, 数值范围均在 $[0., 1.]$ 内。如果色板文件中不含有透明值, 则透明数值 = 1, 表示不透明。

52. 函数 ShadeCOI(wks, plot, w, time, res)

功能说明:

用填充的多边形表示边界效应。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

plot, 要添加边界效应的图形。

w, (从函数 wavelet 中返回的)数据, 必须含有“coi”属性。

time, 与“coi”相对应的表示时间的序列。

res, 修改图形的相关绘图参数(见附录 G)。一般通过 `res = True` 及 `res@gs-FillIndex`(见附录图 H. 3)设定。

示例:

详见 6.3.4 节。

53. 函数 WindRoseColor(wks, wspd, wdir, numPetals, circFr, spdBounds, colorBounds, res)

功能说明:

绘制风玫瑰图, 同时用不同颜色区分不同风速区间。

参数:

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

wspd, 表示全风速大小的一维数组。

wdir, 表示风向的一维数组。这里风向是气象风向, 比如数值 90 表示风从东吹来。

numPetals, 要绘制的风方向的个数。

circFr, 要绘制的频率圈的间隔。

spdBounds, 不同风速所对应的风速范围。

colorBounds, 为 spdBounds 中每个数值设置特定的颜色。

res, 修改图形的相关绘图参数。最常见的是仅设定 `res@tiMainString`, 即设定

主标题。

示例：

详见 7.3.4 节。

附录 F WRF 相关的计算函数、绘图函数及绘图参数

在使用以 wrf 开头的 NCL 绘图函数对 WRF 模式生成的 NetCDF 数据进行绘图时,可能用到两种绘图参数:一是 NCL 官网提供的所有绘图参数(或本书附录 G 中列出的部分常见绘图参数),二是仅针对 WRF 数据的绘图参数。本节介绍的绘图参数即是第二类。

1. 函数 wrf_contour(nc_file, wks, data, res)

功能说明:

绘制 nc_file 中的变量 data 等值线。

参数:

nc_file,要读取的 NetCDF 文件。

wks,工作站标识符(它通常由函数 gsn_open_wks 创建)。需注意,本函数仅创建一个图形,并不输出至 wks。该功能可使用户创建多个图形,并用 overlay 进行叠加。创建的图形并不含有地图,因此,即可用于实际数据的绘制,也可用于“理想”数据的绘制。该函数既可绘制等值线,也可绘制填色图。

data,要绘制的数据。

res,修改等值线的相关绘图参数。具体如下:

res@MainTitle,设置图形主标题。

res@MainTitlePos,主标题的位置,可设为“Left”、“Right”和“Center”。默认为“Left”。

res@NoHeaderFooter,关闭页眉与页脚。

res@Footer,添加页脚。默认为“True”。

res@InitTime,绘制起始时次。默认为“True”。

res@ValidTime: 绘制有效时次。默认为“True”。若为“True”,用户需将 opts@TimeLabel 设一个正确的时间。

res@TimeLabel,要绘制的时次。

res@TimePos,时次的显示位置,可设为“Left”和“Right”。默认为“Right”。

res@ContourParameters,若为一个值,则表示等值线间隔;若为三个值,则表示等值线起始值、结束值和间隔。

res@FieldTitle,更新变量场图题。若不设定,则变量的说明将用作图题。

res@UnitLabel,更新变量场单位。

res@PlotLevelID,添加层次信息至变量图题。

示例:

详见 7.3.3 节。

2. 函数 wrf_interp_1d(v_in ,z_in ,z_out)

功能说明:

在垂直方向上线性内插一维变量。

参数:

v_in,数值型,要在垂直方向上内插的一维数组。

z_in,数值型,表示数组 v_in 垂直结构(高度/气压)的一维数组。数组的维数和大小与 v_in 的相同。

z_out,数值型,数组内插后的常值高度数组。必须与 z_in(高度/气压)数值类型相同。

返回值:

如果输入双精度型,返回双精度型,否则返回浮点型。返回数组的维数和大小与 z_out 的相同。

示例:

```
a = addfile("wrfout_d01_2000-01-24_12:00:00.nc", "r")
time = 0
; 气温
t = a->T(time, :, :, :)      ; 扰动位温
theta = t + 300.              ; 位温
p = a->P(time, :, :, :)
pb = a->PB(time, :, :, :)
pf = p + pb                   ; 全气压
tk = wrf_tk(pf, theta)
; 模式高度
ph = a->PH(time, :, :, :)     ; 扰动位势
phb = a->PHB(time, :, :, :)   ; 基本态位势
tmp = (ph+phb)/9.81
dimv = dimsizes(tmp)
z = 0.5 * (tmp(0:dimv(0)-2, :, :) + tmp(1:dimv(0)-1, :, :))
```

```

z@description = "height"
z@units = "m"
lat = a->XLAT(time, :, :)
lon = a->XLONG(time, :, :)
; 无线电探空仪观测
radiosonde = asciiread("raob_OUN.txt", (/2, 39/), "float")
z_raob = radiosonde(0, :) ; 高度
z_raob@_FillValue = -999.99
; 站点信息
olat = 35.2
olon = -97.4
; 找出 WRF 格点中最靠近无线电探空仪位置的格点
obsij = wrf_latlon_to_ij(lat, lon, olat, olon)
obsi = obsij(0)
obsj = obsij(1)
tin = tk(:, obsi, obsj)
zin = z(:, obsi, obsj)
; 内插
t_wrf = wrf_interp_1d(tin, zin, z_raob)
t_wrf@_FillValue = -999.99

```

3. 函数 wrf_interp_2d_xy(v3d, xy)

功能说明:

从指定输入场中提取横截面。

参数:

v3d, 数值型, 至少 3 维的数组, 最右边三维的大小是 $nz \times ny \times nx$ 。

xy, 数值型, 一个比 v3d 小一维的数组, 其最右边维的大小为 $nxy \times 2$ 。此数组表示一根横切模型区域的线条, 它表示该线条的 x/y 位置。返回的垂直横截面将沿着该线从 v3d 中提取。

返回值:

该函数沿着 xy 所描述的线条从数组 v3d 提取垂直横截面。如果 v3d 具有属性“description”或“units”, 则它们将与输出量一起输出。否则, “description”属性将被设为“”。

示例:

```
a = addfile("wrfout_d01_2000-01-24_12:00:00.nc", "r")
time = 0
t = a->T(time, :, :, :)
theta = t + 300.
p = a->P(time, :, :, :)
pb = a->PB(time, :, :, :)
pf = p + pb
tk = wrf_tk(pf, theta)
dim = dimsizes(tk)
xy = new((/ dim(1), 2 /), float)
xy(:, 0) = dim(2)/2
xy(:, 1) = ispan(0, dim(1)-1, 1)
var2dcr = wrf_interp_2d_xy(tk, xy); 提取变量 tk 沿 xy 线的横截面
```

4. 函数 wrf_interp_3d_z(v3d, vert, loc)

功能说明:

内插至指定的气压层/高度。

参数:

v3d, 数值型, 需内插的数组, 此数组至少 3 维, 最右边三维的大小是 $nz \times ny \times nx$ 。

vert, 数值型, 数组的维数和大小与 v3d 的相同。数组必须是气压值(hPa)或高度值(m), 这取决于在哪个垂直坐标系下进行内插。单位需与参数 loc 的单位一致。

loc, 数值型, 一个标量, 气压值(hPa)或高度值(m), 这取决于在哪个垂直坐标下进行内插。数组 v3d 将被内插到此层。如果 loc 是气压(高度), 则 vert 必须是气压(高度), 并且单位必须相同。

返回值:

返回数组的维数和大小与 v3d 和 vert 的相同, 但没有 nz 维。

示例:

```
a = addfile("wrfout_d01_2000-01-24_12:00:00.nc", "r")
time = 0
t = a->T(time, :, :, :)
theta = t + 300.
p = a->P(time, :, :, :)
pb = a->PB(time, :, :, :)
```

```

pf = p + pb
tk = wrf_tk(pf, theta)
pf = pf * 0.01           ; 转换为 hPa
var2d = wrf_interp_3d_z(tk, pf, 850.); 内插 tk 至 850 hPa

```

5. 函数 wrf_ll_to_ij(lon, lat, opt)

功能说明:

寻找离指定格点经纬度位置最近的模式网格点的下标变量(i,j)。

参数:

lon/lat, 数值型, 在 ARW WRF 中的经度和纬度数组。

opt, 逻辑型, 包含以下属性的逻辑标量。

MAP_PROJ: 地图投影方式(1 = 兰勃特投影, 2 = 极射赤面投影, 3 = 墨卡托投影, 6 = lat-lon 投影)(必填)。

TRUELAT1: 标准纬度 1。若 MAP_PROJ = 1, 2, 3, 必填; 否则, 默认为 0。

TRUELAT2: 标准纬度 2。若 MAP_PROJ = 6, 必填; 否则, 默认为 0。

STAND_LON: 地图投影使用的标准经度(必填)。

REF_LON, REF_LAT: 投影区域中心点的经度与纬度(必填)。

KNOWNI, KNOWNJ: REF_LON 和 REF_LAT 中 i 和 j 的位置(必填)。

POLE_LAT: 若 MAP_PROJ = 6, 可不设定; 否则, 默认为 90。

POLE_LON: 若 MAP_PROJ = 6, 可不设定; 否则, 默认为 0。

DX, DY: 若 MAP_PROJ = 1, 2, 3, 必填; 否则, 默认为 0。

LATINC, LONINC: 若 MAP_PROJ = 6, 必填; 否则, 默认为 0。

返回值:

返回值是一个 float 或双精度型数组, 它表示离输入的经纬度位置最近的网格点的位置 i, j 的值。其大小为 $2 \times N$ 。

N 表示 lon 和 lat 的维数

loc(0, :, ..., :) \rightarrow i 值

loc(1, :, ..., :) \rightarrow j 值

示例:

```

opt = True
opt@MAP_PROJ = 1
opt@TRUELAT1 = 30.0
opt@TRUELAT2 = 60.0
opt@STAND_LON = -98.0

```

```

opt@REF_LAT = 34.83
opt@REF_LON = -81.03
opt@KNOWNJ = 37.0
opt@KNOWNI = 30.5
opt@DX = 30000.
opt@DY = 30000.
loc = wrf_ll_to_ij( -80.0, 30.0, opt )
print(loc)          ; 返回值为 37.45699,20.38181

```

6. 函数 wrf_map(nc_file, wks, res)

功能说明:

绘制地图。

参数:

nc_file, 要读取的 NetCDF 文件。

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。

res, 修改地图的相关绘图参数。详见附录 G。

7. 函数 wrf_map_overlays(nc_file, wks, (/graphics/), pltres, mpres)

功能说明:

合并由 wrf_contour 绘制的等值线图与由 wrf_vector 绘制的矢量图,并添加地图。

参数:

nc_file, 要读取的 NetCDF 文件。

wks, 工作站标识符(它通常由函数 gsn_open_wks 创建)。可叠加任意数量的图形。地图会自动添加至图形上。

graphics, 要绘制的图形。图形按照先后顺序进行叠加。

pltres, 修改 panel 的相关绘图参数。详见附录 G。此外, pltres@NoTitles=True 表示不绘制所有变量场的图题, 若想在翻页前添加字符或线条, 则设 pltres@FramePlot=False, 在调用函数 wrf_map_overlays 后再直接添加字符或线条, 最后再调用命令“frame(wks)”。

mpres, 修改地图的相关绘图参数。详见附录 G。如果想放大图形, 则设定 mpres@ZoomIn=True, 以及放大后图形的位置: mpres@Xstart, mpres@Xend, mpres@Ystart, mpres@Yend。

示例:

详见 7.3.2 节。

8. 函数 `wrf_overlays(nc_file, wks, (/graphics/), pltres, mpres)`

同 `wrf_map_overlays`, 但不绘制地图。

示例:

详见 7.3.3 节。

9. 程序 `wrf_smooth_2d(fld, iter)`

功能说明:

平滑变量场。

参数:

`fld`, 数值型, 要平滑的变量场, 至少包含两维。从 NCL 的 V5.1.0 开始, 该函数不会对数组中缺测值进行平滑。

`iter`, 整型, 指定平滑的次数。

示例:

```
a = addfile("wrfout_d01_2000-01-24_12:00:00.nc", "r")
time = 1
slp = wrf_user_getvar(a, "slp", time)
wrf_smooth_2d(slp, 3)
```

10. 函数 `wrf_time_c(Times, opt)`

功能说明:

将字符型变量“Times”转换为用户指定的数字单位。

参数:

`Times`, 数值型, 表示时间的 WRF 变量, 它必须为字符变量, 数组大小为 `Times (Time, DateStrLen)`。

`opt`, 整型, 指定返回变量的单位。

`opt=0`: 返回 “hours since” 在当前文件上的初始时间(双精度型)。

`opt=1`: 返回 “hours since 1901-01-01 00:00:00”(双精度型)。

`opt=2`: 返回 “yyyymmddhhmmss”(双精度型)。

`opt=3`: 返回 “yyyymmddhh”(整型)。

返回值:

一维数组, 数值对应于 `Times` 字符变量。

11. 函数 wrf_user_getvar(nc_file, fld, it)

功能说明:

从一个由 ARW WRF 模式生成的 NetCDF 文件中读取变量,并计算一些基本的诊断量。

参数:

nc_file,要读取的 NetCDF 文件变量,它可以是函数 addfile 打开的一个文件变量,也可以是由函数 addfiles 打开的多个文件变量。

fld,要读取或获取的变量名。该变量既可以是 ARW WRF 模式的输出文件中的变量,也可以是一系列所支持的诊断量名(以下诊断量名中出现的“/”表示该诊断量的多种名称,比如 lon/long/XLONG/XLONG_M,即为经度的四种名称,任意一种均可被函数 wrf_user_getvar 识别):

(1) avo,绝对涡度 $[10^{-5} \text{ s}^{-1}]$ 。

(2) eth,相当位温 $[\text{K}]$ 。

(3) cape_2d,3 维数组,其最左边维的维数为 4,分别对应 mcape(最大对流有效位能)、mcin(CIN)、lcl(抬升凝结高度)及 lfc(自由对流高度),其剩余的两维对应着两维的水平空间。

(4) cape_3d,2 维数组,其最左边维的维数为 2,分别对应 mcape(最大对流有效位能)、mcin(CIN),其剩余的三维对应着三维空间。

(5) ctt,云顶的温度 $[\text{°C}]$ 。

(6) dbz,反射率 $[\text{dBz}]$ ^①。

(7) mdbz,最大反射率 $[\text{dBz}]$ 。

(8) geopt/geopotential,位势 $[\text{m}^2 \text{ s}^{-2}]$ 。

(9) helicity,风暴相对螺旋度 $[\text{m}^{-2} \text{ s}^{-2}]$ 。

(10) lat/XLAT/XLAT_M,纬度。

(11) lon/long/XLONG/XLONG_M,经度。

(12) omg,垂直速度 $[\text{Pa s}^{-1}]$ 。

(13) p/pres,气压 $[\text{Pa}]$ 。

(14) pressure,气压 $[\text{hPa}]$ 。

(15) pvo,位涡 $[\text{PVU}]$ 。

① 注意,该函数中关于 dbz 的计算仅应用了暖云方案。实际上,WRF 模式的 namelist 文件中有 do_radar_ref 选项,通过该选项可设置特定的微物理方案,从而直接输出对应的 dBz,并不需后期计算。换言之,该函数计算出的 dbz 可能与 WRF 模式计算出的 dbz 存在一定的差异。

- (16)pw, 可降水量[mm]。
- (17)rh2, 2 米相对湿度[%]。
- (18)rh, 相对湿度[%]。
- (19)slp, 海平面气压[hPa]。
- (20)ter/HGT/HGT_M, 地形高度[m]。
- (21)td2, 2 米露点温度[°C]。
- (22)td, 露点温度[°C]。
- (23)tc, 气温[°C]。
- (24)th/theta, 位温[K]。
- (25)tk, 气温[K]。
- (26)times, 数据的时次(返回字符串, 较为常用)。
- (27)Times, 数据的时次(返回字符, 二维的数组)。
- (28)tv, 虚温[K]。
- (29)twb, 湿球温度[K]。
- (30)updraft_helicity, 上升螺旋度[$\text{m}^{-2} \cdot \text{s}^2$]。
- (31)ua, 风场在(地图投影坐标系下)位温所在格点(mass grid)的 X 方向分量。
- (32)va, 风场在(地图投影坐标系下)位温所在格点上的 Y 方向上的分量。
- (33)wa, 风场在(地图投影坐标系下)位温所在格点上的 Z 方向上的分量。
- (34)uvmet10, 转换至地球坐标系下的 10 米 U 和 V。
- (35)uvmet10, 转换至地球坐标系下的 U 和 V。
- (36)z/height, 高度[m]。

it, 要读取的时次。可以是一个时次;也可用起止范围及间隔表示等差数列 (/start_time, end_time, interval/), 如 (/0, 10, 2/); 也可是多个特定时次, 如 (/1, 3, 7, 10/); 若该值为 -1, 则表示读取文件中的所有时次。

示例:

```
ter = wrf_user_getvar(a, "HGT", 0) ;从 a 文件中读取第 0 时次的变量"HGT"
vor= wrf_user_getvar(a, "avo", 0) ;依据 a 文件中的数据,计算出绝对涡度"avo"
```

其他说明:

除上述 36 个诊断量外, 读者还可以通过修改文件“\$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl”使函数 wrf_user_getvar 计算更多的诊断量。具体做法如下:

首先, 复制备份 WRFUserARW.ncl 文件。

其次, 在该文件中找到函数 wrf_user_getvar 的代码段, 它以如下两行开头,

```
undef("wrf_user_getvar")
```

```
function wrf_user_getvar( file_handle, varin[*]:string, time:integer )
```

以如下两行结尾,

```
return(var)
end
```

假定读者想添加的诊断量名为“newvar”,则在该代码段中加入相应计算代码:

```
if( variable .eq. "newvar" ) then
    ;添加相应计算代码
end if
```

最后,通过如下两种方式的任一种方法,即可使用修改后的新的函数:

(1)如果用户是在原 WRFUserARW.ncl 进行修改,则

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
xxx = wrf_usr_getvar(f, "newvar", 0)
```

(2)如果用户将修改后的函数 wrf_usr_getvar 改名为“wrf_usr_getvar2”,同时将该函数存放在一个单独文件中,比如“my_new_script.ncl”,则用户需要加载原先的 WRFUserARW.ncl 文件以及 my_new_script.ncl 文件才能正确调用:

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/wrf/WRFUserARW.ncl"
load "my_new_script.ncl"
xxx = wrf_usr_getvar2(f, "newvar", 0)
```

12. 函数 wrf_user_intrp2d(var2d, loc_param, angle, res)

功能说明:

沿一根线条对 2D 数组进行内插。

参数:

var2d,要进行内插的数据。它最多可以是一个三维数组。最右边 2 维必须是 south_north x west_east。。

loc_param,一个标量,或一个由 2 个或 4 个数值构成的数组。2 个数值的 loc_param 表示要绘制一根线条的中心点的 x/y 坐标;4 个数值的 loc_param 则表示截断的起始和结束位置的 x/y 坐标。

angle,若参数 loc_param 有 4 个数值,则该值设为 0。如果参数 loc_param 仅有 2 个数值,则 angle 表示线条穿过中心点的角度,此时 0. 表示自南向北,90. 表示自西向东。

res,更多参数设置。参数 loc_param 有 4 个数值时,res 设为 True;否则,设为 False。

13. 函数 `wrf_user_intrp3d (var3d, H, plot_type, loc_param, angle, res)`

功能说明:

水平或垂直方向内插。

参数:

`var3d`, 要进行内插的数据。它最多可以是一个五维数组。最右边 3 维必须是 `bottom_top x south_north x west_east`。

`H`, 输出场, 与 `var3d` 同维数。垂直方向上或者是气压层(hPa 或 Pa)或者是几何高度(m)。

`plot_type`, 若为“h”, 则为水平内插图形, 若为“v”, 则为垂直内插图形。

`loc_param`, 一个标量, 或一个由 2 个或 4 个数值构成的数组。若 `plot_type = “h”`, 则标量 `loc_param` 表示要内插的目标层次。当内插至气压层时, 该值的单位为 hPa 或 Pa, 比如 500, 代表内插至 500 hPa。当内插至几何高度时, 该值的单位为 m, 比如 2000, 代表内插至 2000 m。若 `plot_type = “v”`, 2 个数值的 `loc_param` 表示要绘制一根线条的中心点的 x/y 坐标; 4 个数值的 `loc_param` 则表示截断的起始和结束位置的 x/y 坐标。

`angle`, 若 `plot_type = “h”` 或者 `plot_type = “v”` 时参数 `loc_param` 有 4 个数值, 则该值设为 0。如果参数 `loc_param` 仅有 2 个数值, 则 `angle` 表示线条穿过中心点的角度, 此时 0. 表示自南向北, 90. 表示自西向东。

`res`, 更多参数设置。若 `plot_type = “h”`, 或者 `plot_type = “v”` 且 `loc_param` 只有 2 个数值, 则设为 False。若 `plot_type = “h”` 且 `loc_param` 有 4 个数值, 则设为 True。

14. 函数 `wrf_user_list_times(file_handle)`

功能说明:

提取 ARW WRF 模式输出中的可用时间列表。

参数:

`file_handle`, 文件变量或 list 变量, 可以是通过 `addfile` 打开的文件变量, 也可以是通过 `addfiles` 打开的 list 变量。

返回值:

返回文件变量中可用的时间列表。

示例:

```
a = addfile("wrfout_d01_2000-01-24_12:00:00.nc", "r")
times = wrf_user_list_times(a)
```

```

ntimes = dimsizes(times)
print("There are " + ntimes + " times in this file")
print( times )

```

15. 函数 wrf_user_ll_to_ij(nc_file, lons, lats, res)

功能说明:

将模式中经度/纬度位置转换为离该位置最近的 x/y 位置。该函数使用了地图信息以确定最近点的位置,因此,该函数返回值可能超出了模式区域的范围。

参数:

nc_file,要读取的 NetCDF 文件。

lons/lats,要转换的经度/纬度。可以为一个标量,也可以为数组。

res,更多参数设置。res@returnInt=False,则函数返回值为浮点型。否则,返回值为整型(默认)。res@useTime 可设定指定时间的经纬度用以转换。默认为 0。

返回值:

一个表示 x/y 位置的数组。如果输入场为一个点的经度/纬度位置,则返回值数组仅有两个数,第一个数表示 x 格点位置,第二个数表示 y 格点位置。如果输入场是多个格点的经度/纬度位置,则返回值大小为 2Xn,n 表示格点的数目。

其他说明:

若返回值还需在其后的 NCL 脚本中使用,按照 NCL 的语法,该返回值还需减去 1(从 0 开始)。

示例:

```

loc = wrf_user_ll_to_ij (a, 100., 40., res)
loc = wrf_user_ll_to_ij(a, (/100., 120./), (/40., 50./), res) ;loc(0,:)为
x (自西向东)位置,loc(1,:)为 y (自南向北)位置

```

16. 函数 wrf_user_ij_to_ll(nc_file, i, j, res)

功能说明:

对模式中指定位置 x/y 的格点转换为离其最近的经度/纬度位置。由于该函数使用了地图信息以确定最近点的位置,因此该函数返回值可能超出了模式区域的范围。

参数:

nc_file,要读取的 NetCDF 文件。

i/j,(地图投影坐标系下)X/Y 方向上的格点数。注意,从 1 开始。

res,更多参数设置。res@useTime 可设定指定时间的经纬度用以转换。默认

为 0。

返回值:

一个表示经度与纬度的数组。如果输入场为一个位置点,则返回为一个由两个数构成的 1 维数组。其中第一个数表示经度位置,第二个数表示纬度位置。如果 i 与 j 是大小为 n 的数组,则返回大小为 $2 \times n$ 的数组。

示例:

```
loc = wrf_user_ij_to_ll(a, (/10.,12,16/), (/30.,40,40/), True)
```

17. 函数 `wrf_user_vert_interp(file_handle, field, vert_coordinate, interp_levels, opts)`

功能说明:

将 ARW WRF 变量垂直插值到新的高度层次数组中。

参数:

`file_handle`, 文件变量或 list 变量, 可以通过 `addfile` 打开的文件变量, 也可以通过 `addfiles` 打开的 list 变量。

`field`, 数值型, 需要内插的变量场, 必须是三维数组 $\text{bottom_top} \times \text{south_north} \times \text{west_east}$ 或四维数组 $\text{time} \times \text{bottom_top} \times \text{south_north} \times \text{west_east}$ 。

`vert_coordinate`, 字符串型, 一个标量字符串, 表示要内插至高度层次的类型。有效字符串:

"pressure", "pres": 气压(hPa)

"ght_msl": 网格点高度 msl(km)

"ght_agl": 网格点高度 agl(km)

"theta": 位温(K)

"theta-e": 相当位温(K)

`interp_levels`, 数值型, 要内插至的一维垂直层次数组。

`opts`, 逻辑型, 为 True 时, 可以以属性的形式添加更多的选项。有效属性是:

`opts@extrapolate` (逻辑型): 对地面以下数值是否外插(默认为 False)。

`opts@field_type` (字符串型): 变量场的类型。有效字符串为:"ght", "pressure", "pres", "p", "t", "z", "none"。

`opts@logP` (逻辑型): 内插时使用气压对数而不是气压(默认为 False)。

`opts@time` (逻辑型): 用于提取内插变量的时间指数, 它应被设置为用于提取正在进行内插的变量场的同一时间指数(默认 = -1)。

返回值:

返回的变量将与输入变量 `field` 具有相同的维数大小, 除了其 level 维被替换为

新的 levels。属性“vert_interp_type”将被添加,以表明内插的类型。

示例:

```
filename = "wrfout_d01_1991-01-01_00:00:00"
fin = addfile(filename+".nc", "r")
fld = wrf_user_getvar(fin, "tk", -1) ;要插值的变量
vert_coord = "theta"
interp_levels = ispan(200, 1000, 50)
opts = True
pts@extrapolate = True
opts@field_type = "T"
opts@logP = True
fld_intrp = wrf_user_vert_interp(fin, fld, vert_coord, interp_levels, opts)
printVarSummary(fld_intrp)
```

;输出为

Variable: fld_intrp

Type: float

Total Size: 7493940 bytes

1873485 values

Number of Dimensions: 4

Dimensions and sizes: [Time | 1] x [interp_levels | 17] x [south_north | 279]

x [west_east | 395]

Coordinates:

interp_levels: [200..1000]

Number Of Attributes: 8

_FillValue :9.96921e+36

description :Temperature

FieldType :104

MemoryOrder :XYZ

units :K

stagger :

coordinates :XLONG XLAT

vert_interp_type :theta

18. 函数 `wrf_vector(nc_file, wks, data_u, data_v, res)`

功能说明:

根据 `nc_file` 中的变量 `data_u` 和 `data_v`, 绘制矢量箭头。

参数:

`nc_file`, 要读取的 NetCDF 文件。

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。需注意, 本函数仅创建一个图形, 并不输出至 `wks`。该功能可使用户创建多个图形, 并用 `overlay` 进行叠加。创建的图形并不含有地图, 因此, 既可用于实际数据的绘制, 也可用于“理想”数据的绘制。

`data_u/data_v`, 要绘制的数据。

`res`, 修改等值线的相关绘图参数。除函数 `wrf_contour` 中列出的各种绘图参数外, 还有 `opts@NumVectors`, 设定风矢量在图形中的密度。

示例:

详见 7.3.5 节。

19. 函数 `wrf_wps_close_int(istatus)`

功能说明:

关闭打开的 WPS 中间文件。在读完 WPS 文件后, 一般无需关闭 WPS 文件, 除非需要在同一脚本中重新打开该文件。

参数:

`istatu`, 整型。

20. 函数 `wrf_wps_dom(wks, mpres, lnres, txres)`

功能说明:

预览将要分析的区域。

参数:

`wks`, 工作站标识符(它通常由函数 `gsn_open_wks` 创建)。

`mpres`, 修改地图的相关绘图参数, 详见附录 G。

`lnres`, 修改线条的相关绘图参数, 详见附录 G。

`txres`, 修改文本的相关绘图参数, 详见附录 G。

示例:

详见 7.3.1 节。

21. 函数 `wrf_wps_open_int(filename)`

功能说明:

打开 WPS 中间文件并返回一个状态。如果成功则返回 0, 否则返回 1。它表示从 WPS 中间文件读取数据的三步过程中的第一步:

- (1) `wrf_wps_open_int`: 打开一个 WPS 文件。
- (2) `wrf_wps_rdhead_int`: 读取当前变量场的标题信息。
- (3) `wrf_wps_rddata_int`: 读取当前变量场的数据。

参数:

`filename`, 字符串型, 要打开的 WPS 中间文件的名称。

22. 函数 `wrf_wps_rddata_int(istatus, nx, ny)`

同 `wrf_wps_open_int`, 但为从打开的 WPS 中间文件读取 2D 变量场, 即从 WPS 中间文件读取数据的三步过程中的第三步。其中 `nx/ny` 为整型数值。

23. 程序 `wrf_wps_rdhead_int(istatus, rhead, field, date, units, map_source, description)`

功能说明:

从打开的 WPS 中间文件读取标题信息。即从 WPS 中间文件读取数据的三步过程中的第二步。

参数:

`istatus`, 整型, 打开的 WPS 文件的状态。每次调用此过程时, 此变量都将更新。

如果 `istatus = 0`, 表示可能有更多的数据要从文件读取。

如果 `istatus = 1`, 表示无法读取标题信息, 和/或没有更多数据要从文件读取。

`rhead`, 浮点型, 一个预先分配的长度为 14 的浮点型数组, 它将包含每个场的标题值。

`field`, 字符串型, 当前场名称的标量字符串。

`date`, 字符串型, 当前场日期的标量字符串。

`units`, 字符串型, 当前场单位的标量字符串。

`map_source`, 字符串型, 当前场地图源的标量字符串。

`description`, 字符串型, 当前场描述的标量字符串。

24. 函数 `wrf_wps_read_int(filename)`

功能说明:

从 WPS 中间文件读取数据。

参数:

filename, 字符串型, 要打开的 WPS 中间文件的名称。

返回值:

此函数从指定的 WPS 中间文件读取数据, 并返回一个浮点型、大小为 $(nfields \times ny \times nx)$ 的三维数组, 其中每个大小为 $(ny \times nx)$ 的数组子集是一个要写入文件的二维变量场。它还返回附加在变量上的几个属性, 每个属性为长度为 $nfields$ 的字符串数组, 包含了以下几个变量场的信息:

hdate: 每个变量场的日期。

field: 每个变量场的名称。

units: 每个变量场的单位。

description: 每个变量场的描述。

map_source: 每个变量场的地图源。

25. 程序 wrf_wps_write_int(filename, field, units, description, data, opt)**功能说明:**

将数据写入 WPS 中间文件。

参数:

filename, 字符串型, 输出 WPS 文件的名称。

field, 字符串型, 正在写入变量场的名称。

units, 字符串型, 正在写入变量场的单位。

description, 字符串型, 正在写入变量场的描述。

date, 字符串型, 写入文件名的二维变量场。

opt, 逻辑型, 包含了将要写入文件的附加元数据。此逻辑必须设置为 True, 并且有必须设置的几个必需属性。

附录 G 常用绘图参数

以下两个方面需读者注意。一是 NCL 虽将绘图参数按类型进行了分类(比如 xy 类、cn 类、gsn 类等), 但在实际使用中, 对某一特定图形要素的设置可能涉及到不同类的绘图参数。例如在绘制等值线时, 除需设定 cn 类绘图参数外, 部分 gsn 类的绘图参数可能也需设定。二是绘图参数相互之间并不完全独立, 有些绘图参数的设定需要以另一些绘图参数的设定为前提。因此, 本节在编排常用绘图参数时, 不仅考虑其类型, 而且兼顾了其使用时的逻辑顺序。更多绘图参数请参考其官网内容(附录 A 节)。

1. 等值线 contour(cn)

;等值线数值

res@cnLevels = (/ -4, 3, 20/) ; 绘制特定等值线。仅在 cnLevelSelectionMode 设为 "ExplicitLevels" 有效

res@cnLevelSelectionMode = "AutomaticLevels" ; 默认值, 表示自动选择等值线; 其余设定为 "ExplicitLevels"、"ManualLevels"

res@cnMaxLevelValF = 20 ; cnLevelSelectionMode 设为 "ManualLevels" 时的最大值

res@cnMinLevelValF = -20 ; cnLevelSelectionMode 设为 "ManualLevels" 时的最小值

res@cnLevelSpacingF = 0.5 ; 等值线数值间隔 0.5。仅在 cnLevelSelectionMode 设为 AutomaticLevels 及 ManualLevels 有效

; 等值线线型

res@cnLinesOn = True ; 绘制等值线, 为 True 时以下设定才有效

res@cnLineDashPattern = 2 ; 线型为 2 的虚线, 见附录图 H.6

res@cnLineDashSegLenF = 0.25 ; 各个虚线段的长度

res@cnLineColor = "black" ; 颜色

res@cnLineThicknessF = 0.3 ; 0.3 倍粗细

res@gsnContourNegLineDashPattern = 16 ; 正值等值线线型为 16。默认是 0

res@gsnContourPosLineDashPattern = 0 ; 负值等值线线型。默认是 0, 实线

res@gsnContourZeroLineThicknessF = 0 ; 不画 0 值等值线。默认是 1

; 等值线颜色(或形状)填充

res@cnFillOn = True ; 颜色或图形填充等值线, 为 True 时以下设定才有效(为 True 时, 默认用颜色填充, 除非设置 cnFillPattern, 即可改为形状填充)

res@cnFillColors = (/5, 7, -1/) ; -1 为透明。等值线很多时, 可不设定, 让 NCL 在色板中自动挑选颜色

res@cnFillMode = "RasterFill" ; 默认是 "AreaFill", 第三种选项为 "CellFill", 即在格点上填充

res@cnFillOpacityF = 0.5 ; 50% 透明

```
res@cnFillPalette      = "GMT_gray" ; 用“GMT_gray”色板填色
res@cnFillPattern      = 5          ; 形状填充。共有 18 种类型可选,请参考附录图 H.3
res@cnFillScaleF       = 0.5        ; (为形状填充时)修改形状填充的密度
```

;等值线数值标签

```
res@cnLineLabelsOn      = True          ; 打开等值线数值标签
res@cnLineLabelAngleF   = 0.0          ; 标签角度
res@cnLineLabelBackgroundColor = "white" ; 边框内部颜色
res@cnLineLabelDensityF = 0.5          ; 修改等值线上标签数目的相对密度
res@cnLineLabelPerimOn  = True          ; 添加边框
res@cnLineLabelPerimColor = "white"     ; 边框颜色
res@cnLineLabelPlacementMode = "Computed" ; 绘制标签的算法。设为"Computed"时,可进一步通过设定 cnLineLabel-DensityF 修改等值线上标签数目
```

;等值线数值信息标签

```
res@cnInfoLabelOn      = True          ; 打开等值线信息标签。默认 True
res@cnInfoLabelOrthogonalPosF = 0.05   ; 垂直移动等值线信息标签
res@cnInfoLabelParallelPosF  = 0.05    ; 水平移动等值线信息标签
```

;绘制顺序(有三个值:“PreDraw”、“Draw”、“PostDraw”)

```
res@cnLineDrawOrder = "PostDraw" ; 最后绘制等值线
res@cnLabelDrawOrder = "PostDraw" ; 最后绘制数值标签
```

;添加高值中心标签“H”。

```
res@cnHighLabelsOn      = True          ; 添加“H” 标签,为 True 时以下设定才有效
res@cnHighLabelBackgroundColor = -1     ; 背景色为透明
res@cnHighLabelFontHeightF = 0.024     ; 字体大小
```

;添加低值中心标签“L”

```
res@cnLowLabelsOn      = True          ; 添加“L” 标签,为 True 时以下设定才有效
```

res@cnLowLabelBackgroundColor = -1 ; 背景色为透明
 res@cnLowLabelFontHeightF = 0.024 ; 字体大小

;其他

res@gsnAddCyclic = True ; 对于有经度坐标变量的数据而言,添加循环点,使等值线能在 0 度左右连续。如果数据在经度维上的第一个点与最后一点一致,则需要设为 False

res@gsnScalarContour = True ; 在调用 gsn_csm_vector_scalar **** 绘图函数时,除绘制矢量箭头外,还绘制等值线表示第三个变量

2. 标识、多边形、任意折线 (gs)

;标识

pmres@gsMarkerColor = "black" ; 颜色
 pmres@gsMarkerIndex = 4 ; 类型,见附录图 H.2
 pmres@gsMarkerSizeF = 8 ; 大小
 pmres@gsMarkerThicknessF = 2.0 ; 粗细

;多边形

pmres@gsFillColor = "black" ; 颜色
 pmres@gsFillIndex = 2 ; 类型,见附录图 H.3
 pmres@gsFillLineThicknessF = 0.5 ; 填充区内线的粗细
 pmres@gsFillOpacityF = 0.5 ; 50%透明

;任意折线

pmres@gsLineColor = "black" ; 颜色
 pmres@gsLineDashPattern = 16 ; 线型,见附录图 H.6
 pmres@gsLineDashSegLenF = 0.12 ; 虚线段的长度
 pmres@gsLineLabelString = "ENSO" ; 字符
 pmres@gsLineLabelFontColor = "red" ; 字符颜色
 pmres@gsLineLabelFontHeightF = 0.02 ; 字符大小
 pmres@gsLineLabelFontThicknessF = 0.5 ; 字符粗细
 pmres@gsLineLabelFuncCode = ":" ; 字符函数码,用以显示特殊形式字符

```

pmres@gsLineOpacityF          = 0.5          ; 50%透明
pmres@gsLineThicknessF        = 2            ; 粗细

```

3. gsn 高级接口(gsn)

; 水平参考线

```

res@gsnXRefLine                = 0.          ; X 轴参考线的数值
res@gsnXRefLineColor           = "black"     ; 颜色
res@gsnXRefLineDashPattern     = 16          ; 线型
res@gsnXRefLineThicknessF      = 2.          ; 2 倍粗

```

; 垂直参考线。

```

res@gsnYRefLine                = 0.          ; 设定 Y 轴参考线的数值
res@gsnYRefLineColor           = "black"     ; 颜色
res@gsnYRefLineDashPattern     = 16          ; 线型
res@gsnYRefLineThicknessF      = 2.          ; 2 倍粗

```

; bar 图。

```

res@gsnXYBarChart              = True
res@gsnXYBarChartBarWidth      = 0.5        ; bar 的宽度
res@gsnXYBarChartFillOpacityF = 0.5        ; 50%透明

```

; 以下对折线或 bar 图均适用(需设定 res@gsnYRefLine 和 res@gsnXYBarChart = True)

```

res@gsnAboveYRefLineBarColors  = "red"      ; 大于 Y 轴参考值时填充的颜色
res@gsnAboveYRefLineBarPatterns = 16        ; 大于 Y 轴参考值时填充的形状,
                                                见附录图 H.3
res@gsnAboveYRefLineColor      = "black"; 大于 Y 轴参考值时折线的填充
                                                颜色
res@gsnBelowYRefLineBarColors  = "blue"     ; 小于 Y 轴参考值时填充所用的
                                                颜色
res@gsnBelowYRefLineBarPatterns = 3         ; 小于 Y 轴参考值时填充的形状
res@gsnBelowYRefLineColor      = "black"; 小于 Y 轴参考值时折线的填充
                                                颜色

```

;绘制纬向平均图。仅在使用 `gsn_csm_contour_map_ce` 或 `gsn_csm_contour_map` 绘图时有效

`res@gsnZonalMean` = True ; 在图形的右侧绘制纬向平均图
`res@gsnZonalMeanXMaxF` = 30 ; X 轴的最大值
`res@gsnZonalMeanXMinF` = -30 ; X 轴的最小值
`res@gsnZonalMeanYRefLine` = 0 ; 添加一个纬向平均的参考值

; Lambert Mask

`res@gsnMaskLambertConformal` = True ; labert 投影 mask,若不设定则图形通常为正方形。通常需要设定投影类型 `mpProjection` 为 "LambertConformal"。同时设定如下绘图参数以指定地图的范围: `mpMinLonF` , `mpMaxLonF`, `mpMinLatF`, `mpMaxLatF`
`res@gsnMaskLambertConformalOutlineOn` = True ;绘制边界线

; 排列多幅图形(panel)

`res@gsnPanelCenter` = True ; 在用 `gsn_panel` 排列多幅图时,居中对齐每行图
`res@gsnPanelDebug` = False ; 不在图形中显示图形的坐标等信息
`res@gsnPanelFigureStrings` = ("/(a)","(b)","(c)"/) ; 为每幅图添加序号标签。若要调整标签的位置,则可设置 `res@amJust` = "TopLeft",默认"CenterCenter"
`res@gsnPanelFigureStringsBackgroundFillColor` = "white"; 序号标签的背景颜色
`res@gsnPanelFigureStringsFontHeightF` = 0.02 ; 序号标签字体大小
`res@gsnPanelFigureStringsPerimOn` = False ; 序号标签周边加框
`res@gsnPanelLabelBar` = False ; 多幅图公用一个色标。默认 False
`res@gsnPanelBottom` = 0.0 ; 最下边图形绘制结束的位置。默认 0.0
`res@gsnPanelLeft` = 0.0 ; 最左边图形开始绘制的位置。默认 0.0
`res@gsnPanelRight` = 1.0 ; 最右边图形绘制结束的位置。默认 1.0
`res@gsnPanelRowSpec` = False ; 手动指定每行绘制图形的个数
`res@gsnPanelTop` = 1.0 ; 最上边图形开始绘制的位置。默认 1.0
`res@gsnPanelXWhiteSpacePercent` = 1. ; 每两个相邻子图之间沿 X 轴方向的空白大小。默认 1
`res@gsnPanelYWhiteSpacePercent` = 1. ; 每两个相邻子图之间沿 Y 轴方向的空白大小

; 图形正上方字符。适用于每个 gsn_csm 绘图函数。

```
res@gsnCenterString          = "title" ; 字符
res@gsnCenterStringFontColor = "black" ; 颜色
res@gsnCenterStringFontHeightF = 0.02 ; 大小
res@gsnCenterStringFuncCode   = ":"    ; 字符函数码
res@gsnCenterStringOrthogonalPosF = 0.02 ; 调整其垂直位置
res@gsnCenterStringParallelPosF  = 0.02 ; 调整其水平位置
```

; 图形左上方字符。适用于每个 gsn_csm 绘图函数。

```
res@gsnLeftString           = "title" ; 字符
res@gsnLeftStringFontColor  = "black" ; 颜色
res@gsnLeftStringFontHeightF = 0.02 ; 大小
res@gsnLeftStringFuncCode   = ":"    ; 字符函数码
res@gsnLeftStringOrthogonalPosF = 0.02 ; 调整其垂直位置
res@gsnLeftStringParallelPosF  = 0.02 ; 调整其水平位置
```

; 图形右上方字符。适用于每个 gsn_csm 绘图函数。

```
res@gsnRightString          = "title" ; 字符
res@gsnRightStringFontColor = "black" ; 颜色
res@gsnRightStringFontHeightF = 0.02 ; 大小
res@gsnRightStringFuncCode   = ":"    ; 字符函数码
res@gsnRightStringOrthogonalPosF = 0.02 ; 调整其垂直位置
res@gsnRightStringParallelPosF  = 0.02 ; 调整其水平位置
```

; 色板中颜色的选取

```
res@gsnSpreadColorStart = 2 ; 指定色板中第 2 个颜色用于绘制第一个等值线
                           或箭头。默认 2
res@gsnSpreadColorEnd   = -1 ; 色板的最后一个颜色绘制最后一个等值线或
                           箭头。默认 -1
```

; 以下是在使用 gsn_histogram 绘图函数时使用

```
res@gsnHistogramHorizontal = False ; 水平绘制。默认 False
res@gsnHistogramBarWidthPercent = 66 ; histogram 的宽度
res@gsnHistogramBinWidth      = 10 ; 设定 bin 的等间距间隔
```

`res@gsnHistogramBinIntervals` = (/1,2,5,7/) ; 设定 bin 的不等间距间隔。该例中将统计数值在 1-2, 2-5, 5-7 这三个区间中的个数

`res@gsnHistogramCompare` = True ; 绘制两类数据的频率统计结果。输入数据的最左边维的大小需为 2

`res@gsnHistogramComputePercentages` = True ; 将在右侧 Y 轴显示百分比。对于水平绘制的图, 则在上部 X 轴显示百分比

`res@gsnHistogramMinMaxBinsOn` = True ; 当设定 `gsnHistogramBinIntervals` 时, 该绘图参数将添加两个 bins。在该例中, 它将添加小于 1 及大于 7 的 bins

`res@gsnHistogramPercentSign` = True ; 设定 `gsnHistogramComputePercentages` 为 True 后, 该绘图参数将在右侧 Y 轴标签上添加符号%, 取消字符“percent”的显示

; 绘图及图形设置

`res@gsnDraw` = True ; 绘图, 默认 True

`res@gsnFrame` = True ; 翻页, 默认 True

`res@gsnMaximize` = False ; 最大化页面, 默认 False

`res@gsnPolar` = "NH" ; 北半球极射赤面投影。通常需在 `gsn_csm_xxxx_polar` 一类函数中使用

`res@gsnPolarLabelFontHeightF` = 0.02 ; 极射赤面投影经度/纬度标签字符的大小

`res@gsnScalarContour` = True ; 在调用 `gsn_csm_vector_scalar ****` 绘图函数时, 除绘制矢量箭头外, 还绘制等值线以表示第三个变量

`res@gsnShape` = False ; 默认 False。对于没有地图的图形而言, 若设为 True, 则 X 轴与 Y 轴的长度将根据各自的取值范围等比例绘制

4. 地图 map(mp)

; 地图的投影方式

`res@mpProjection` = "LambertConformal" ; 默认 "CylindricalEquidistant", 其他选项为 "Orthographic", "Stereographic", "Gnomonic", "AzimuthalEquidistant", "Satellite", "PseudoMollweide", "Mercator", "LambertEqualArea", "Robinson", "

RotatedMercator", "Aitoff", "Hammer", "Mollweide", "WinkelTripel"

; 地图的长宽比例

res@mpShapeMode = "FreeAspect" ; 默认是固定比例"FixedAspectFitBB"

; 地图的范围

res@mpCenterLonF = 180 ; 地图的中心经度

res@mpMaxLonF = 180 ; 地图的最东经度

res@mpMinLonF = 0. ; 地图的最西经度

res@mpMaxLatF = 90 ; 地图的最北纬度

res@mpMinLatF = 10. ; 地图的最南纬度

; 地图经纬度线及标签

res@mpGridAndLimbOn = True ; 绘制经纬度线

res@mpGridLatSpacingF = 15 ; 纬度线间隔

res@mpGridLineDashPattern = 2 ; 线型, 见附录图 H.6

res@mpGridLineThicknessF = 0.5 ; 线型粗细

res@mpGridLonSpacingF = 15 ; 经度线间隔

res@mpGridPolarLonSpacingF = 15 ; 极射赤面投影的经度线间隔

res@pmTickMarkDisplayMode = "Always" ; 坐标标签上添加符号度

; 地图数据

res@mpDataBaseVersion = "MediumRes" ; 中等分辨率地图。默认是低分辨率(可满足一般需求), 高分辨率则需另行下载

res@mpDataSetName = "Earth..4" ; 2008 年生成的数据, 包含中国的省界。详见 http://www.ncl.ucar.edu/Document/HLUs/Classes/MapPlotData4_1_earth_4.shtml

; 地图边界线

res@mpGeophysicalLineColor = "black" ; 地图边界线的颜色

res@mpGeophysicalLineDashPattern = 16 ; 地图边界线的线型, 见附录图 H.6

res@mpGeophysicalLineDashSegLenF = 0.2 ; 虚线线段的长度

res@mpGeophysicalLineThicknessF = 0.5 ; 地图边界线的粗细

res@mpMaskOutlineSpecifiers = ("/China:Jiangsu","Japan"/) ; 不绘制的边界名称

res@mpOutlineBoundarySets = "NoBoundaries" ; 无边界。默认是“Geophysical”。若为“National”,则在大陆边界线上增加国界线

res@mpOutlineMaskingOn = True ; 打开边界线的 mask

res@mpOutlineOn = True ; 绘制边界线

res@mpOutlineSpecifiers = ("/China : Jiangsu"/) ; 指定绘制的边界名称。若到省界,需指定地图数据。见上文 mp-DataSetName

; 地图填色

res@mpFillOn = True ; 填色地图

res@mpFillColor = "gray" ; 灰色填色

res@mpInlandWaterFillColor = 0 ; 内陆湖泊, 白色

res@mpLandFillColor = 0 ; 陆地, 白色

res@mpMonoFillColor = True ; 默认 False

res@mpOceanFillColor = 0 ; 海洋, 白色

;通过下列绘图参数,可实现在指定区域绘图的效果(需首先设定 res@mpFillOn = True)

res@mpAreaMaskingOn = True ; mask 某些区域

res@mpMaskAreaSpecifiers = ("/China","Taiwan","Arunachal Pradesh","Disputed area between India and China"/) ; 这是中国的陆地领土。图形效果是除中国陆地领土外的其他所有区域将被抹去(不绘制)

res@mpFillDrawOrder = "PostDraw" ; 最后填色地图

res@mpFillAreaSpecifiers = ("/China : Jiangsu","water","land"/) ; 指定填色的地图区域

res@mpSpecifiedFillColors = ("/transparent", "white","white"/) ; 多用颜色。此例的效果为仅在江苏省内绘制

5. 色标 labelbar(lb)

; 绘制色标

```
res@lbLabelBarOn      = True
res@lbOrientation      = "Horizontal" ; "Horizontal" 摆放, 默认 "Vertical"
res@gsnPanelLabelBar  = True          ; 绘制公用的色标。在 panel 绘图时使用。
                                通常在绘制每个子图时, 需 res@lbLabelBarOn
                                = False
```

; 色标边框

```
res@lbBoxLinesOn      = False          ; 绘制色标边框, 默认 True
res@lbBoxLineColor     = "black"       ; 边框的颜色
res@lbBoxLineThicknessF = 1.5          ; 边框线的粗细
res@lbBoxLineDashPattern = 16          ; 边框线的线型, 见附录图 H.6
```

; 色标标签

```
res@lbLabelOffsetF     = 0.1           ; 色标边框与标签之间的距离。默认 0.1
res@lbLabelFontHeightF = 0.01         ; 字体大小
res@lbLabelFontColor    = "black"     ; 字体颜色
res@lbLabelFontThicknessF = 1.5       ; 1.5 倍粗
res@lbLabelStride       = 1           ; 色标标签间隔数
```

; 若色标与坐标相重叠时:

```
res@pmLabelBarOrthogonalPosF = .50    ; 调整色标垂直方向的位置
res@pmLabelBarParallelPosF   = 0.5    ; 调整色标水平方向的位置
```

; 色标大小

```
res@pmLabelBarWidthF    = 0.15       ; 宽度
res@pmLabelBarHeightF   = 0.6        ; 高度
```

6. 图例 legend(lg)

; 绘制图例

```
res@pmLegendDisplayMode = "Always"    ; 默认 "NoCreate"
```

; 图例摆放的方向

res@lgOrientation = "horizontal"

; 图例的位置及大小

res@pmLegendOrthogonalPosF = -0.8 ; 垂直方向移动的距离

res@pmLegendParallelPosF = .90 ; 水平方向移动的距离

res@pmLegendSide = "Top" ; 调整其位置

res@pmLegendWidthF = 0.12 ; 图例的宽度

res@pmLegendHeightF = 0.25 ; 图例的长度

; 图例的边框

res@lgPerimOn = False ; 不绘制边框

res@lgPerimFill = "SolidFill" ; 对图例填色

res@lgPerimFillColor = "white" ; 图例填充的颜色

; 图例的标签

res@xyExplicitLegendLabels = ("/20N", "30N", "40N", "50N"/)

res@lgLabelFontColor = "black" ; 标签颜色

res@lgLabelFontHeightF = 0.015 ; 标签字体大小

res@lgLabelFontThicknessF = 1.5 ; 标签字体粗细

res@lgLabelOffsetF = 0.02 ; 标签与边框的距离

7. 图题 title(ti)

; 主标题字符。整体上与 tx 绘图参数类似

res@tiMainOn = True ; 为 True 时以下设定才有效

res@tiMainFontColor = "black"

res@tiMainFontHeightF = 0.025

res@tiMainFontThicknessF = 1.

res@tiMainFuncCode = ":" ; 字符函数码,用以显示特殊形式
字符

res@tiMainJust = "CenterCenter" ; 字符对齐的位置,见附录图 H.5

res@tiMainOffsetXF = 0.1 ; 调整其水平位置

res@tiMainOffsetYF = 0.1 ; 调整其垂直位置

res@tiMainPosition = "Center" ; 其余设定为"Left"和"right"

```
res@tiMainSide          = "Top"          ; 默认 Top,即该图形的上方绘制。
                           其余为“Bottom”
```

```
res@tiMainString        = "Main string"
```

; X 轴设置.

```
res@tiXAxisOn           = True            ; 给 X 轴添加名称
res@tiXAxisFontColor    = "black"        ; 颜色
res@tiXAxisFontHeightF  = 0.025          ; 大小
res@tiXAxisFontThicknessF = 1.           ; 粗细
res@tiXAxisFuncCode     = ":"            ; 字符函数码,用以显示特殊形式
                           字符
res@tiXAxisJust          = "CenterCenter" ; 字符对齐的位置,见附录图 H.5
res@tiXAxisOffsetXF      = 0.1           ; 调整其水平位置
res@tiXAxisOffsetYF     = 0.1           ; 调整其垂直位置
res@tiXAxisPosition      = "Center"      ; 其余两个为"Left"和"right"
res@tiXAxisSide          = "Bottom"      ; 默认 Bottom,即在图形的下方绘
                           制。可改为“Top”
res@tiXAxisString        = "X Axis"
```

; Y 轴设置。

```
res@tiYAxisOn           = True            ; 给 Y 轴添加名称
res@tiYAxisFontColor    = "black"        ; 颜色
res@tiYAxisFontHeightF  = 0.025          ; 大小
res@tiYAxisFontThicknessF = 1.           ; 粗细
res@tiYAxisFuncCode     = ":"            ; 字符函数码,用以显示特殊形式
                           字符
res@tiYAxisJust          = "CenterCenter" ; 字符对齐的位置,见附录图 H.5
res@tiYAxisOffsetXF      = 0.1           ; 调整其水平位置
res@tiYAxisOffsetYF     = 0.1           ; 调整其垂直位置
res@tiYAxisPosition      = "Center"      ; 其余设定为"Bottom"和"Top"
res@tiYAxisSide          = " Left "      ; 默认 Left,即在图形左方绘制。
                           也可在“Right”绘制
res@tiYAxisString        = "Y Axis"
```

8. 坐标(tm 及 tr)

; X 轴设置

res@trXLog=True ; 对数形式显示, 要求 X 轴的最大最小值均大于 0

res@trXReverse = True ; 逆序显示

res@trXMinF=91 ; X 轴的最小值

res@trXMaxF=304 ; X 轴最大值(该绘图参数仅是设定数值范围, 其数值大小不一定在图形中显示)

res@tiXAxisOn = True ; 添加 X 轴名称

res@tiXAxisString = "Level"

res@tmXMajorGrid = True ; 在 X 轴主刻度线位置上绘制直线

res@tmXMajorGridLineDashPattern = 16 ; 线型, 见附录图 H.6

res@tmXMajorGridLineColor = "gray" ; 颜色

res@tmXMajorGridThicknessF = 0.5 ; 粗细

res@tmXMinorGrid = True ; 在 X 轴次刻度线位置上绘制直线

res@tmXMinorGridLineDashPattern = 3 ; 线型, 见附录图 H.6

res@tmXMinorGridLineColor = "red" ; 颜色

res@tmXMinorGridThicknessF = 0.3 ; 粗细

; Y 轴与 X 轴一致, 只需将上述绘图参数中“X”改为“Y”(略)

; XB 轴设定

res@tmXBOn = True ; 默认 True

res@tmXBStyle = "Log" ; 对数形式显示

res@tmXBMode = "Manual" ; 默认“Automatic”

res@tmXBTickStartF = -150 ; 最小刻度值

res@tmXBTickEndF = 400 ; 最大刻度值

res@tmXBTickSpacingF = 50 ; 刻度的数值间隔


```

res@tmXBMode = "Explicit"
res@tmXBValues = (/0, 90, 180, 270/) ;若是在
                        地图上绘制,则这些数值指定其经度
res@tmXBLabels = (/ "0", "90~S~o~N~E", "180", "90~S~o~
                        N~W" /) ;在对应位置上添加带圈号的经
                        度。这里使用了默认的字符函数码 "~"
res@tmXBLabelDeltaF = -0.6 ;移动其位置,使
                        其更加靠近 X 轴
res@tmXBLabelFontHeightF = 0.02 ;字体大小
res@tmXBLabelFontThicknessF = 2 ;粗细

res@tmXBMajorLengthF = 0.01 ;XB 坐标轴上主刻度线的长度
res@tmXBMajorOutwardLengthF = 0.01 ;主刻度线向外的长度,此例表明主
                        刻度线在 X 轴下方

res@tmXBMinorOn = True ;设定次刻度线
res@tmXBMinorValues = (/100, 130/) ;次刻度线所对应的 X 轴的数值
res@tmXBMinorLengthF = 0.006 ;通常设定其长度小于主刻度线
res@tmXBMinorOutwardLengthF = 0.006 ;次刻度线向外的长度

```

;XT、YL、YR 轴的设定与 XB 轴设定一致,只需将上述绘图参数中的“XB”分别改为“XT”、“YL”及“YR”

;地图上经度与纬度线的间隔

```

res@gsnMajorLatSpacing = 10 ;主要纬度线间隔
res@gsnMajorLonSpacing = 10 ;主要经度线间隔
res@gsnMinorLatSpacing = 5 ;次要纬度线间隔
res@gsnMinorLonSpacing = 5 ;次要经度线间隔

```

;其他设置

```

res@gsnPolarLabelFontHeightF = 0.02 ;修改极射赤面投影图坐标标签字体的大小
res@gsnTickMarksOn = True ;绘制坐标刻度及其标签。默认 True

```

9. 字符 text(tx)

;任意添加的字符。整体上与 ti 绘图参数类似

`res@txAngleF` = 30 ; 字体相对于 X 轴方向的倾斜角度, 默认 0 度
`res@txFuncCode` = ":" ; 字符函数码, 用以显示特殊形式字符
`res@txFontColor` = "black" ; 颜色
`res@txFontHeightF` = 0.05 ; 大小
`res@txFontThicknessF` = 1. ; 粗细
`res@txString` = "example" ; 字符

; 字符位置

`res@txJust` = "CenterCenter" ; 字符对齐的位置, 见附录图 H.4
`res@txPosXF` = 0.5 ; 水平方向的位置(单位坐标系中)
`res@txPosYF` = 0.5 ; 垂直方向的位置(单位坐标系中)

; 字符的边框设定

`res@txPerimColor` = "black" ; 颜色
`res@txPerimDashPattern` = 16 ; 线型, 见附录图 H.6
`res@txPerimDashLengthF` = 0.15 ; 虚线线段长度
`res@txPerimOn` = True ; 默认 False
`res@txPerimThicknessF` = 1. ; 粗细

10. 箭头 `vector(vc)`

`resv@vcMapDirection` = False ; 在绘制垂直剖面图时, 一定要设为 False, 默认 True

`resv@vcMaxMagnitudeF` = 160. ; 大于该数值则不绘制
`resv@vcMinDistanceF` = 0.03 ; 箭头之间的最小距离(单位坐标系中)
`resv@vcMinMagnitudeF` = 1.0 ; 小于该数值则不绘制
`resv@vcGlyphOpacityF` = 0.5 ; 50% 透明
`resv@vcGlyphStyle` = "FillArrow" ; 其余设定为 "LineArrow" (默认)、
 "WindBarb"、"CurlyVector"

`resv@vcPositionMode` = "ArrowTail" ; 格点位置对应箭头的尾部。其余设定为 "ArrowHead" 及 "ArrowCenter" (默认)

`resv@vcVectorDrawOrder` = "PostDraw" ; 绘制的顺序。其余设定为: "PreDraw"、"Draw"

; 设定 FillArrow (需设定 `resv@vcGlyphStyle` = "FillArrow")

```

resv@vcFillArrowEdgeThicknessF = 2           ; 箭头边界线的粗细
resv@vcFillArrowEdgeColor       = "white"     ; 箭头边界线的颜色
resv@vcFillArrowFillColor       = "black"     ; 箭头内部填充颜色
resv@vcFillArrowHeadInteriorXF  = 0.25        ; 见附录图 H.1
resv@vcFillArrowHeadXF          = 0.6         ; 见附录图 H.1
resv@vcFillArrowHeadYF          = 0.2         ; 见附录图 H.1
resv@vcFillArrowWidthF          = 0.1         ; 见附录图 H.1

```

;也可设定所有 FillArrow 的头部大小一致

```

resv@vcFillArrowMinFracWidthF = 1.0
resv@vcFillArrowHeadMinFracXF = 1.0
resv@vcFillArrowHeadMinFracYF = 1.0

```

;设定 LineArrow(需设定 resv@vcGlyphStyle = "LineArrow")

```

resv@vcMonoLineArrowColor      = True         ; 默认 True
resv@vcLineArrowColor          = "black"
resv@vcLineArrowThicknessF     = 4.0         ; 4 倍粗

```

;也可设定所有 LineArrow 的头部大小一致

```

resv@vcLineArrowHeadMaxSizeF = 0.01
resv@vcLineArrowHeadMinSizeF = 0.01

```

;参考箭头

```

resv@vcRefAnnoOn                = True         ; 为 True 时以下设定才有效
resv@vcRefAnnoBackgroundColor   = "white"     ; 背景颜色
resv@vcRefAnnoFontHeightF       = 0.015       ; 参考箭头标签字体大小
resv@vcRefAnnoOrthogonalPosF    = -0.12       ; 调整其垂直位置
resv@vcRefAnnoParallelPosF      = 0.95        ; 调整其水平位置
resv@vcRefAnnoPerimOn           = False        ; 关闭边框
resv@vcRefAnnoSide               = "Top"       ; 参考箭头放至图形上方,默认为
                                         "Bottom",其余设定为"Left"与"Right"
resv@vcRefAnnoString1On         = False        ; 不设定参考箭头上方的字符
resv@vcRefAnnoString2On         = True         ; 设定参考箭头下方的字符
resv@vcRefAnnoString2           = "30 m/s"

```

resv@vcRefMagnitudeF = 30 ; 单位长度箭头所表示的实际物理量的大小

resv@vcRefLengthF = 0.045 ; 单位长度箭头在图形中的长度 (单位坐标系中)

11. 视图 viewport(vp)

res@vpHeightF = 0.25 ; 视图的高度

res@vpWidthF = 0.8 ; 视图的宽度

res@vpXF = 0.1 ; 视图左上角在单位坐标系中的水平位置

res@vpYF = 0.9 ; 视图左上角在单位坐标系中的垂直位置

12. 折线(xy)

;折线类型

resxy@xyMarkLineMode = "Lines" ; 用线条表示。默认“Lines”，其余设定为“Markers”、“MarkLines”

resxy@xyMarkLineModes = ("/Markers", "MarkLines", "Lines"/) ; 分别用符号、线条与符号、线条表示

;折线采用相同线型、颜色、粗细

resxy@xyDashPattern = 0 ; 实线。默认值

resxy@xyLineColor = "blue" ; 颜色。也可用色板中对应颜色的数字表示

resxy@xyLineThicknessF = 3 ; 3 倍粗

;折线采用不同线型、颜色、粗细

resxy@xyDashPatterns = (/0,1,3,16/) ; 见附录图 H.6

resxy@xyMonoDashPattern = False ; 默认 False

resxy@xyMonoLineColor = False ; 默认 True

resxy@xyMonoLineThickness = False ; 默认 False

resxy@xyLineColors = ("/blue", "red"/) ; 也可用色板中对应颜色的数字表示

resxy@xyLineThicknesse = (/1., 1., 1., 3./) ; 不同粗细

;折线上的标识采用相同的形状、颜色、大小

resxy@xyMarker = 3 ; 见附录图 H.2

resxy@xyMarkerColor = "black" ; 颜色

```
resxy@xyMarkerSizes = 0.02 ; 大小
```

;折线上的标识采用不同形状、颜色、粗细

```
resxy@xyMarkers = (/3,5,16/) ; 不同类型,见附录图 H.2
```

```
resxy@xyMarkerColors = (/ "blue", "yellow", "red" /) ; 不同颜色
```

```
resxy@xyMarkerSizes = (/1.,2.,1./) ; 不同大小
```

```
resxy@xyMonoMarkerColor = False ; 默认 False
```

```
resxy@xyMonoMarker = False ; 默认 False
```

```
resxy@xyMonoMarkerSize = False ; 默认 False
```

13. 流线 streamline (st)

; ; 流线箭头

```
resst@stArrowLengthF = 0.02 ; 箭头大小
```

```
resst@stArrowStride = 3 ; 每 3 个格点绘制一个箭头,默认是 2
```

```
resst@stMinArrowSpacingF = 0.004 ; 箭头之间的空格
```

; ; 流线的设置

```
resst@stLevelPalette = "ncl_default" ; 设定色板
```

```
resst@stLevelSpacingF = 1. ; 间隔
```

```
resst@stLevelSelectionMode = "ManualLevels" ; 其余设定“AutomaticLevels”(默认),“ExplicitLevels”,“EqualSpacedLevels”
```

```
resst@stLineOpacityF = 0.7 ; 70%透明
```

```
resst@stLineThicknessF = 1.5 ; 流线粗细
```

```
resst@stLineColor = "orange" ; 颜色
```

```
resst@stMaxLevelValF = 10. ; 最大值
```

```
resst@stMinLevelValF = 1 ; 最小值
```

```
resst@stMonoLineColor = False ; 关闭仅使用一种颜色,即使用多种颜色
```

; ; 设定流线之间的距离

```
resst@stMinDistanceF = 0.03
```

```
resst@stMinLineSpacingF = 0.007
```

附录 H 常用绘图参数图示

该附录列出常用绘图参数图示,图 H.1—H.6 分别为填充箭头、标识类型、填充类型、文本对齐位置、字符函数、虚线类型。

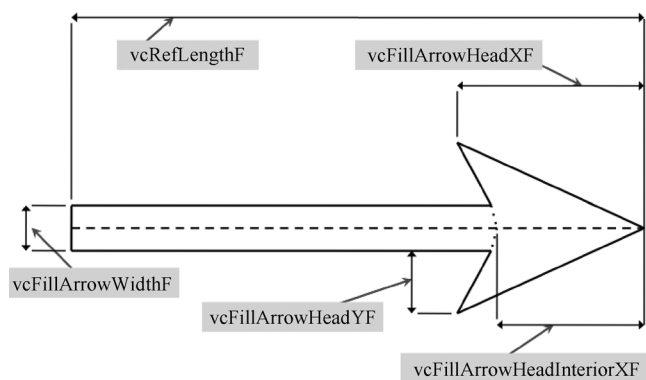


图 H.1 填充箭头

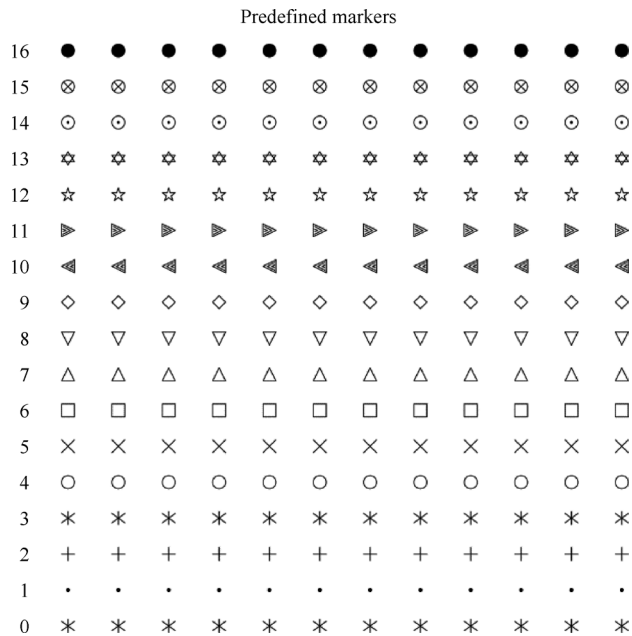


图 H.2 标识类型

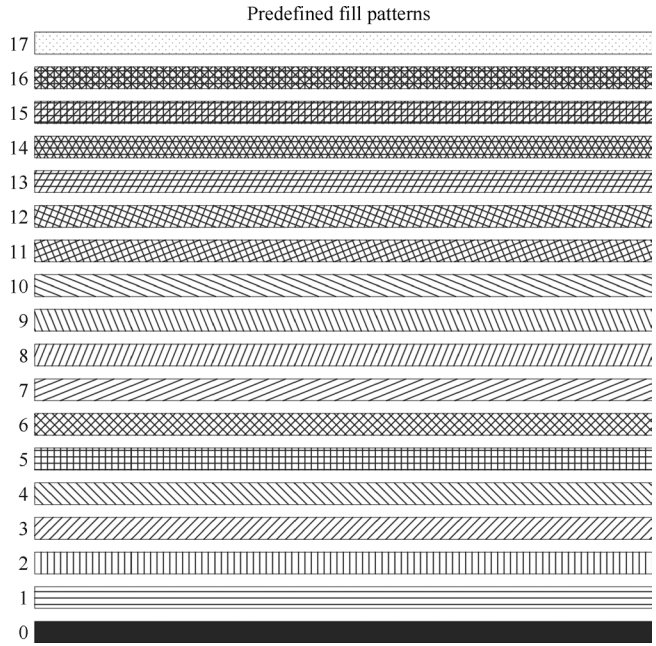


图 H.3 填充类型

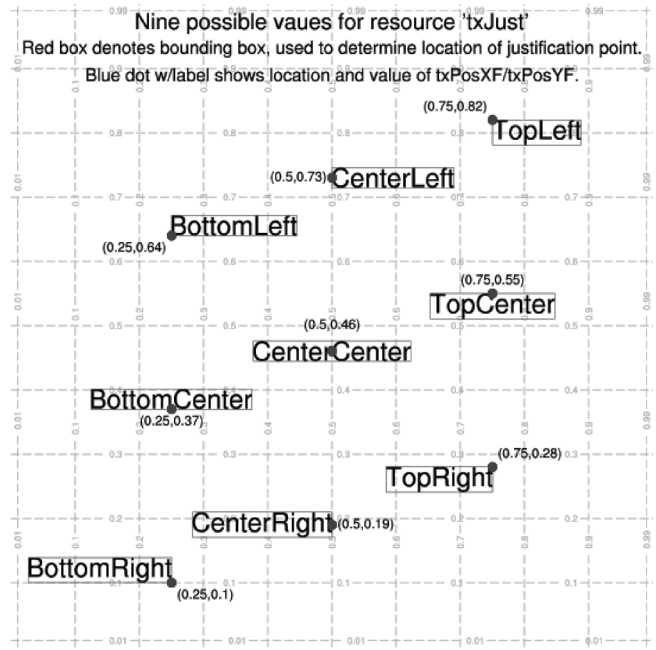


图 H.4 文本对齐位置

Function Codes

"Superscripts: x~S~2~N~ + y~S~2~N~"
→ Superscripts: $x^2 + y^2$

"Subscripts: CH~B~4~N~ + N~B~2~N~O"
→ Subscripts: $\text{CH}_4 + \text{N}_2\text{O}$

"Carriage~C~Return"
→ Carriage
Return

"Happy ~F19~%~F~ Valentine's Day"
→ Happy ♥ Valentine's Day

Can be used for any
string in a graphic
tiMainString
txString
xylLineLabel
etc

图 H. 5 字符函数

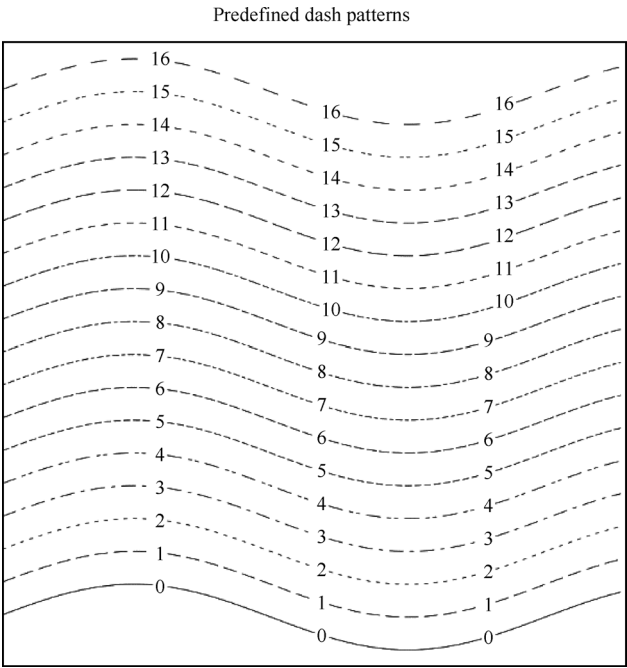


图 H. 6 虚线类型

附录 I 常用色板

这里给出几种常用色板,如图 I. 1—I. 4 分别给出三种彩虹色色板、四种地球海

洋色板、两种灰色色板和五种白色居中色板。每个颜色上的数字表示该颜色在该色板中的位置。例如读者使用的是“WhBlGrYeRe”色板(图 I. 1(b)), 则 `res@xyLineColors = (/1, 100/)` 表示绘制黑色和红色折线。色板名称已标注在色板下方, 注意其大小写, 不可写错, 否则 NCL 不能识别。

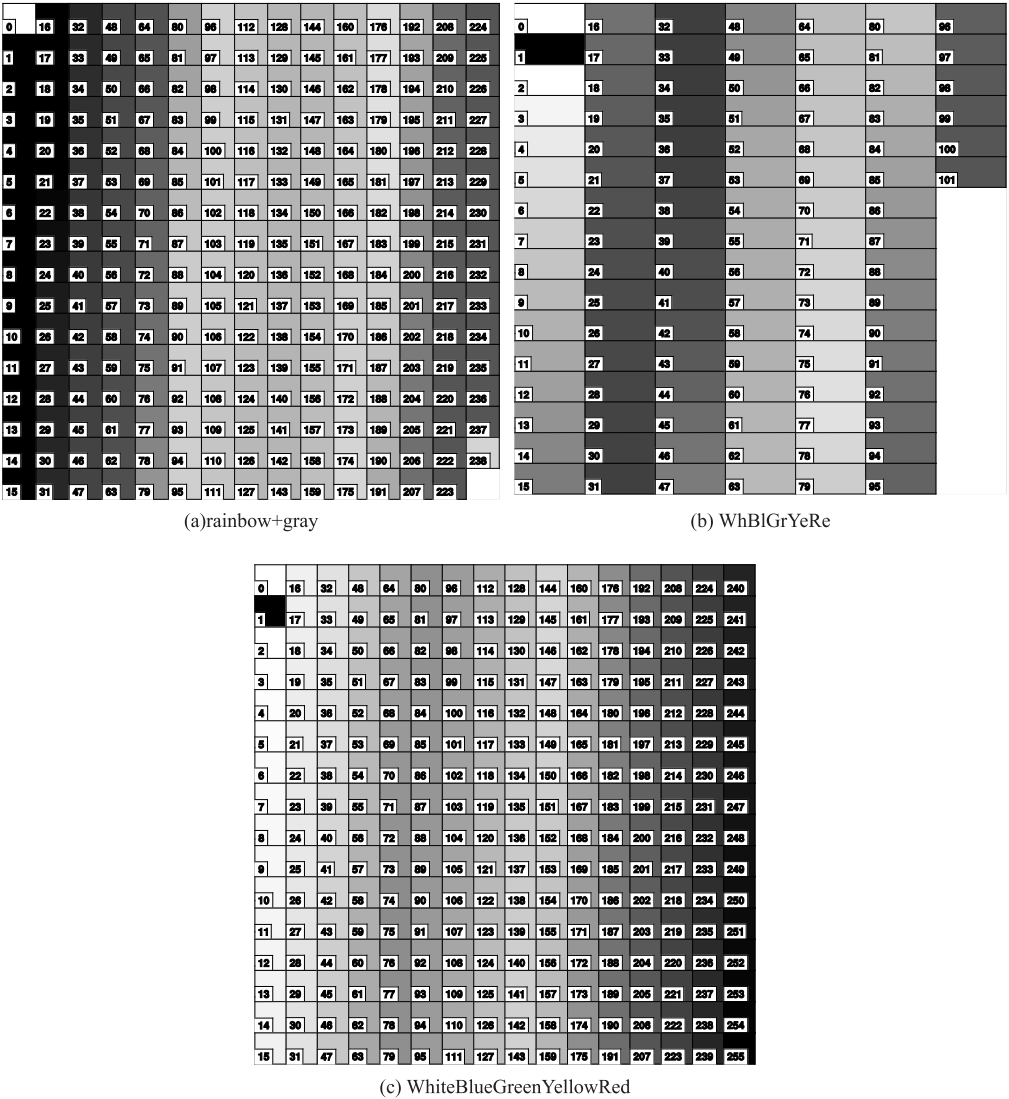


图 I. 1 三种彩虹色色板(附彩图 I. 1)

0		16		32		48		64	
1		17		33		49		65	
2		18		34		50			
3		19		35		51			
4		20		36		52			
5		21		37		53			
6		22		38		54			
7		23		39		55			
8		24		40		56			
9		25		41		57			
10		26		42		58			
11		27		43		59			
12		28		44		60			
13		29		45		61			
14		30		46		62			
15		31		47		63			

(a) cmp_haxby

0		16		32		48			
1		17		33		49			
2		18		34		50			
3		19		35		51			
4		20		36		52			
5		21		37		53			
6		22		38		54			
7		23		39		55			
8		24		40		56			
9		25		41		57			
10		26		42		58			
11		27		43		59			
12		28		44		60			
13		29		45		61			
14		30		46					
15		31		47					

(b) GMT_drywet

0		16		32		48		64		80		96		112		128
1		17		33		49		65		81		97		113		129
2		18		34		50		66		82		98		114		
3		19		35		51		67		83		99		115		
4		20		36		52		68		84		100		116		
5		21		37		53		69		85		101		117		
6		22		38		54		70		86		102		118		
7		23		39		55		71		87		103		119		
8		24		40		56		72		88		104		120		
9		25		41		57		73		89		105		121		
10		26		42		58		74		90		106		122		
11		27		43		59		75		91		107		123		
12		28		44		60		76		92		108		124		
13		29		45		61		77		93		109		125		
14		30		46		62		78		94		110		126		
15		31		47		63		79		95		111		127		

(c) MPL_ocean

0		16		32		48		64		80		96		112		128		144		160		176		192		208		224		240
1		17		33		49		65		81		97		113		129		145		161		177		193		209		225		241
2		18		34		50		66		82		98		114		130		146		162		178		194		210		226		242
3		19		35		51		67		83		99		115		131		147		163		179		195		211		227		243
4		20		36		52		68		84		100		116		132		148		164		180		196		212		228		244
5		21		37		53		69		85		101		117		133		149		165		181		197		213		229		245
6		22		38		54		70		86		102		118		134		150		166		182		198		214		230		246
7		23		39		55		71		87		103		119		135		151		167		183		199		215		231		247
8		24		40		56		72		88		104		120		136		152		168		184		200		216		232		248
9		25		41		57		73		89		105		121		137		153		169		185		201		217		233		249
10		26		42		58		74		90		106		122		138		154		170		186		202		218		234		250
11		27		43		59		75		91		107		123		139		155		171		187		203		219		235		251
12		28		44		60		76		92		108		124		140		156		172		188		204		220		236		252
13		29		45		61		77		93		109		125		141		157		173		189		205		221		237		253
14		30		46		62		78		94		110		126		142		158		174		190		206		222		238		254
15		31		47		63		79		95		111		127		143		159		175		191		207		223		239		255

(d) OceanLakeLandSnow

图 I.2 四种地球海洋色板(附彩图 I.2)

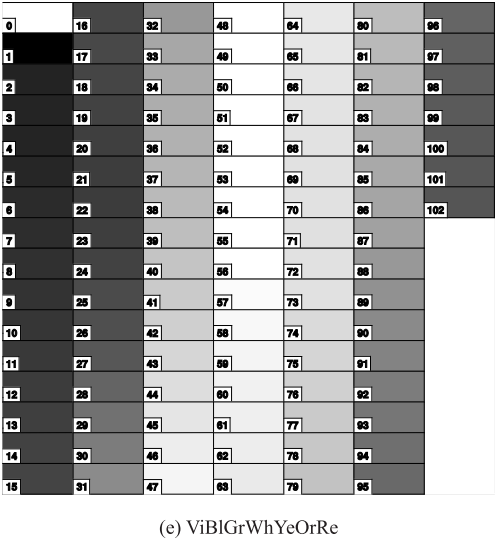
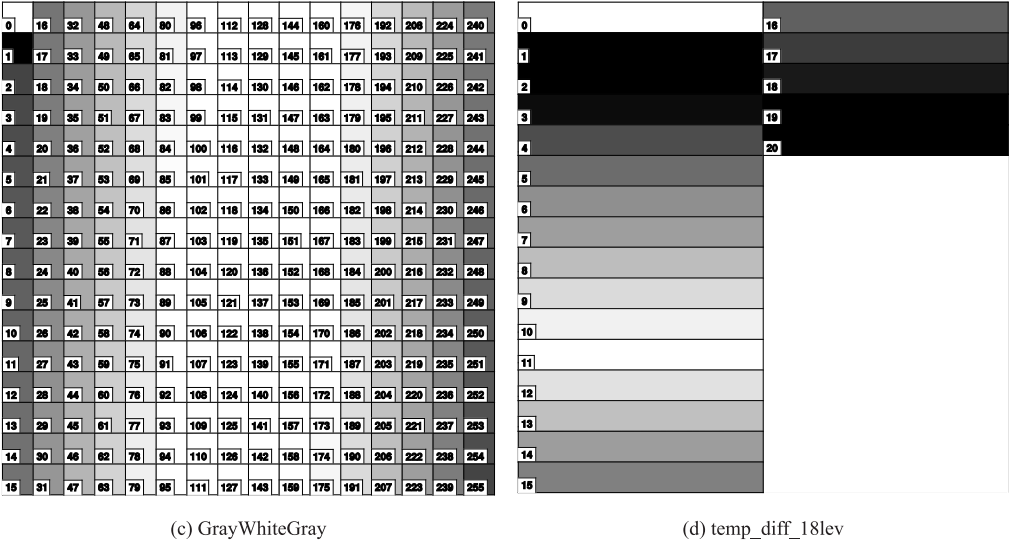


图 I. 4 五种白色居中色板(附彩图 I. 4)

附录 J 常见错误提醒

要想避免出现错误提醒,正如 Dennis Shea 常说的那样:请查看你的数据! 一旦你熟悉你所使用的数据格式,那么就可以避免很多不必要的警告或错误提醒。因此,我

们建议用户在遇到问题时,请用使用程序 printVarSummary(或命令 ncl_filedump、软件 CDO)对数据进行检查。以下列出几个常见的警告和错误提醒及其解决方案(表 J. 1、J. 2)。

表 J. 1 常见语法错误提醒

<p>错误信息:fatal:Subscript out of range, error in subscript #0</p> <p>原因:数组下标数值超出数组的大小范围。注意,NCL 中数组是从 0 开始,而不是 1</p> <p>例如:<code>x = (/1,2,3,4,5/)</code> <code>print(x(5))</code> ;打印第 6 个数(实际上 x 数组仅有 5 个数)</p> <p>解决方案:检查数组下标值,以确保它们在数组大小范围内。用 命令"print" 和"printVarSummary"获取更多数组变量信息</p>
<p>错误信息:fatal: Subscript out of range, error in subscript #1</p> <p>原因:数组下标数值超出数组的大小范围。下标数值从 0 开始且方向为从数组的左边维指向右边维,因此该错误信息中的"#1"指的是该数组中从左向右数第 2 维</p> <p>例如:<code>x = random_uniform(-100,100,(/10,20,30/))</code> <code>print(x(5,20,5))</code> ;指数'20'实际是第 21 个数,超出该维的大小</p> <p>解决方案:检查数组下标值,已确保它们在数组大小范围内。用 命令"print" 和"printVarSummary"获取更多数组变量信息</p>
<p>错误信息:fatal:Number of subscripts on right-hand-side do not match number of dimensions of variable: (4), Subscripts used: (3)</p> <p>原因:使用了错误的数组维数</p> <p>例如:<code>x = random_uniform(-50,50,(/5,32,64/))</code> ; x 是一个三维的数组 <code>y = x(0, :, :, :)</code></p> <p>解决方案:检查数组维数</p>
<p>错误信息:fatal:Assignment type mismatch, right hand side can't be coerced to type of left hand side</p> <p>原因:在重新赋值时,使用了不同的数值类型或数组大小及维数</p> <p>例如:<code>x = 5</code> ;整型 <code>x = "Now I'm a string"</code> ;字符型</p> <p>解决方案:使用重新赋值操作符":=",或先删除原先的变量,再赋值</p>
<p>错误信息:fatal:syntax error: possibly an undefined procedure</p> <p>原因:所用的函数或程序并不存在</p> <p>例如:<code>i = 5</code> <code>prnt(i)</code> ;应该是 print(i)</p> <p>解决方案:检查所写函数或程序是否存在拼写错误或进一步确认该函数或程序需要实现加载某个库函数文件</p>
<p>错误信息:fatal:syntax error: function fspan expects 3 arguments, got 2</p> <p>原因:调用函数或程序时使用了错误的参数个数</p> <p>例如:<code>x = fspan(0,10)</code> ;缺少生成数组大小的参数</p> <p>解决方案:检查该函数或程序的参数设置</p>

续表

错误信息: fatal:syntax error: line -1

原因: 程序代码块中没有闭合, 比如使用了“begin”, 但在脚本后却没有出现对应的“end”, 或使用了“if”, 但没有“end if”

例如: if (x.lt.0) then
 x = 5

解决方案: 闭合程序块

错误信息: fatal:Dimension sizes of left hand side and right hand side of assignment do not match

原因: 在不同大小的数组间进行赋值

例如: x = (/1,2,3,4/)
 y = (/1,2,3,4,5/)
 x = y

解决方案: 检查等号“=”前两边数组的大小

错误信息: fatal:The result of the conditional expression yields a missing value. NCL can not determine branch, see ismissing function

原因: 在使用 if 语句或其他条件判断语句时, 出现了缺省值

例如: x = new(1,float) ; 此时 x 的数值为默认的缺省值
 if(x.eq.5) then
 print("x is 5")
 end if

解决方案: 使用函数“ismissing”以检查缺省值

 x = new(1,float)
 if(.not.ismissing(x).and.x.eq.5) then
 print("x is 5")
 end if

错误信息: fatal:Variable (x1) is undefined

原因: 变量未定义

例如: x = 5
 print(x1)

解决方案: 检查变量名的拼写

错误信息: warning:Attempt to reference attribute (FillValue) which is undefined

原因: 变量的属性不存在

例如: x = new(1,float)
 print(x@FillValue); 应为 print(x@_FillValue)

解决方案: 检查变量属性的拼写

续表

<p>错误信息:Argument 0 of the current function or procedure was coerced to the appropriate type and thus will not change if the function or procedure modifies its value</p> <p>原因:调用函数或程序时所使用的参数的数据类型错误。比如当函数需要的是一个字符型参数,而脚本传递的是一个数值</p> <p>例如:<code>x = 12.34</code> <code>str = str_split(x, ".")</code></p> <p>解决方案:使用诸如以“to”开头的“toXXX”等函数转换参数的类型</p> <code>x = tostr(12.34)</code> <code>str = str_split(x, ".")</code>

表 J.2 常见绘图错误提醒

<p>错误信息:<code>gsn_csm_contour_map</code>: Fatal: the input data array must be 1D or 2D fatal:Illegal right-hand side type for assignment</p> <p>原因:调用函数 <code>gsn_csm_dontour_map</code> 时,所使用的变量的维数有误(超过了 2D)</p> <p>例如:<code>plot = gsn_csm_contour_map(wks, tsurf, res)</code> ; 这里假定 <code>tsurf(time, lat, lon)</code></p> <p>解决方案:检查你的数据。变量的维数是多少? 可用函数 <code>printVarSummary</code> 查看。你可以挑出一个时次进行绘图,例如:</p> <code>plot = gsn_csm_contour_map(wks, tsurf(0, :, :), res)</code> ; 仅绘制第 1 时次
<p>错误信息:<code>gsn_add_cyclic</code>: Warning: The range of your longitude data is not 360. You may want to set <code>gsnAddCyclic</code> to False to avoid a warning message from the Spline function.</p> <p>原因:变量的经度范围小于 360 度</p> <p>解决方案:<code>res@gsnAddCyclic = False</code></p>
<p>错误信息:<code>is_valid_lat_ycoord</code>: Warning: The units attribute of the Y coordinate array is not set to one of the allowable units values (i.e. 'degrees_north'). Your latitude labels may not be correct. <code>is_valid_lat_xcoord</code>: Warning: The units attribute of the X coordinate array is not set to one of the allowable units values (i.e. 'degrees_east'). Your longitude labels may not be correct.</p> <p>原因:经度和纬度数组的单位没有被正确设置</p> <p>解决方案:用程序 <code>printVarSummary</code> 检查要绘图的变量,或者直接检查坐标变量经度、纬度的单位是否正确。若没有添加相应的单位,则</p> <code>lat@units = "degrees_north"</code> <code>lon@units = "degrees_east"</code>

续表

错误信息: warning:ScalarFieldSetValues: 2d coordinate array sfXArray has an incorrect dimension size: defaulting sfXArray
warning:ScalarFieldSetValues: 2d coordinate array sfYArray has an incorrect dimension size: defaulting sfYArray

原因:若数组不是在全球范围格点上,则该问题是指函数 gsn_csm_xxxx 将 gsnAddCyclic 这一 resource(默认)设置为 True

例如: res@sfXArray = lon ; -- range of lon values < 360
res@sfYArray = lat

解决方案: res@gsnAddCyclic = False

错误信息: warning:tmXBStride is not a valid resource in ex07-1_xy at this time

原因:resource 的名称拼写错误,或者在使用当前绘图函数时,该 resource 不起作用

例如 1: res@tmXBStride = 2

解决方案 1: 检查 resource 是否名称错误。该例中应写为 res@tmXBLabelStride = 2

例如 2: res@xyLineColor = "grey"
plot = gsn_csm_contour_map(wks, t, res)

解决方案 2: 在使用绘图函数“gsn_csm_contour_map”时,“xyLineColor”这一 resource 不起作用。如需将等值线设置为灰色,则设置“cnLineColor”: res@cnLineColor = "grey"

函数索引

鉴于 NCL 提供的函数及程序十分庞杂,这里将本书中所提及的常见函数及程序按其针对的对象大致分为地球科学、应用数学、绘图、输入输出、统计分析、数组操作及查询和其他几部分给出,并分别按音序排列。

地球科学

- 大圆路径的距离(gc_latlon) 123
- EP 通量(epflux) 37
- 辐散风(高斯格点,dv2uvG_Wrap) 111
- 辐散风(均匀经纬度网格点,dv2uvF_Wrap) 111
- 复制元数据(copy_VarMeta) 37,38,46,53,106
- 公历日期(由一年中的序号转换,monthday) 133
- 公历日期在一年中的序号(day_of_year) 107
- 季节平均(month_to_season) 37,38,132
- 静力稳定度(等压面,pot_vort_isobaric) 135
- 流函数和速度势(高斯格点,uv2sfvpG) 150
- 流函数和速度势(均匀经纬度格点,uv2sfvpF) 149
- 日期格式的转换(cd_calendar) 53,103
- 散度场(由球谐函数计算,高斯格点 uv2vrdivG_Wrap) 149
- 散度场(由球谐函数计算,经纬度格点 uv2vrdivF_Wrap) 149
- 散度场(由中央差分计算,uv2dv_cfd) 148
- 时间的字符型转换(wrf_time_c) 181
- 时间列表(wrf_user_list_times) 185
- 梯度(高斯格点,gradsg) 125
- 梯度(均匀经纬度网格点,gradsf) 125
- 网格点的下标变量(wrf_ll_to_ij) 179
- 网格点的下标变量(使用了地图信息,wrf_user_ll_to_ij) 186
- 位温(等压面,pot_vort_isobaric) 135
- 位涡(等压面,pot_vort_isobaric) 135

- 涡度场(高斯格点,uv2vrdrvG_Wrap) 149
- 涡度场(由球谐函数计算,均匀经纬度网格点,uv2vrdrvF_Wrap) 149
- 涡度场(由中央差分计算,uv2vr_cfd) 149
- WRF 模式中物理诊断量计算(wrf_user_getvar) 182
- 旋转风(高斯格点,vr2uvG_Wrap) 152
- 旋转风(均匀经纬度网格点,vr2uvF_Wrap) 152

应用数学

- 垂直插值(不同气压层之间,int2p_n_Wrap) 126
- 垂直插值(混合坐标系至气压层,vinth2p) 151
- 垂直插值(至多种高度层中,wrf_user_vert_interp) 187
- 垂直积分(vibeta) 150
- 垂直内插(输入场为一维变量,至新的垂直坐标系下,wrf_interp_1d) 176
- 垂直内插(至指定高度层,wrf_interp_3d_z) 178
- 垂直平均或和(wgt_vert_avg_beta) 156
- 对数(e 为底,log) 131
- 对数(10 为底,log10) 131
- 反余弦值(acos) 98
- 反正切值(atan,atan2) 102,103
- 反正弦值(asin) 102
- 横截面(wrf_interp_2d_xy) 177
- 滑动平滑(9 点二维,smth9_Wrap) 142
- 滑动平滑(不等权重,wgt_runave_n_Wrap) 155
- 滑动平滑(等权重,runave_n_Wrap) 140
- 内插(水平或垂直,wrf_user_intrp3d) 185
- 内插(沿直线,wrf_user_intrp2d) 184
- 平方根(sqrt) 39,143
- 平滑变量场(wrf_smooth_2d) 181
- 区域面积平均(wgt_areaave) 56,153
- 水平插值(由高斯格点插值至均匀经纬度网格点,由球谐函数计算,g2fsh_Wrap) 122
- 水平插值(由均匀经纬度网格点插值至高斯格点,由球谐函数计算,f2gsh_Wrap) 117
- 水平插值(在两种高斯格点之间进行,由球谐函数计算,g2gsh_Wrap) 123
- 水平插值(在两种均匀经纬度网格点之间进行,由球谐函数计算,f2fsh_Wrap) 117

水平插值(在两种直线网格系之间进行,由双线性插值方法计算,linint2_Wrap) 128
填补缺测值(线性插值方法,linmsg_n) 129
有限中央差分(center_finite_diff_n) 105
余数(mod) 132
余弦值(cos) 12,106
正切值(tan) 147
正弦值(sin) 142

绘图

创建图形文件(gsn_open_wks) 8,172
等值线的线型及颜色设置(ColorNegDashZeroPosContour) 42,159
合并矢量图及标量图(wrf、不叠加地图,wrf_overlays) 181
合并矢量图及标量图(wrf、叠加地图,wrf_map_overlays) 180
合并图形(overlay) 43,173
绘制标识(单位坐标系内,gsn_polymarker_ndc) 173
绘制垂直剖面图(gsn_csm_pres_hgt) 54,166
绘制垂直剖面图(箭头,gsn_csm_pres_hgt_vector) 166
绘制垂直剖面图(流线图,gsn_csm_pres_hgt_streamline) 166
绘制等值线图(gsn_csm_contour) 14,163
绘制等值线图(wrf,wrf_contour) 80,175
绘制等值线图(叠加地图,gsn_csm_contour_map) 16,163,196,219
绘制等值线图(叠加地图的两组等值线图,gsn_csm_contour_map_overlay) 164
绘制等值线图(叠加地图、等圆柱投影,gsn_csm_contour_map_ce) 10,163,196
绘制等值线图(叠加地图、极射赤面投影,gsn_csm_contour_map_polar) 49,164
绘制地图(gsn_csm_map) 19,165
绘制地图(wrf,wrf_map) 180
绘制地图(等圆柱投影,gsn_csm_map_ce) 165
绘制地图(极射赤面投影,gsn_csm_map_polar) 165
绘制多边形(单位坐标系内,gsn_polygon_ndc) 172
绘制风玫瑰图(WindRoseColor) 85,98,174
绘制空白图(gsn_csm_blank_plot) 162
绘制流线及等值线图(叠加地图,gsn_csm_streamline_contour_map) 167
绘制流线及等值线图(叠加地图、等圆柱投影,gsn_csm_streamline_contour_map_ce) 167

- 绘制流线及等值线图(叠加地图、极射赤面投影,gsn_csm_streamline_contour_map_polar) 167
- 绘制流线图(gsn_csm_streamline) 166
- 绘制流线图(叠加地图,gsn_csm_streamline_map) 168
- 绘制流线图(叠加地图、等圆柱投影,gsn_csm_streamline_map_ce) 168
- 绘制任意折线(单位坐标系内,gsn_polyline_ndc) 173
- 绘制色标(单位坐标系内,gsn_labelbar_ndc) 33,171
- 绘制时间—经度图(gsn_csm_hov) 164
- 绘制时间—纬度图(gsn_csm_lat_time) 165
- 绘制矢量图(wrf,wrf_vector) 88,189
- 绘制矢量图及标量图(叠加地图、极射赤面投影,gsn_csm_vector_scalar_map_polar) 168
- 绘制填充等值线(gsn_contour_shade) 73,162
- 绘制图例(单位坐标系内,gsn_legend_ndc) 42
- 绘制文本(单位坐标系内,gsn_text_ndc) 173
- 绘制折线图(两个 X 轴和两个 Y 轴,gsn_csm_x2y2) 169
- 绘制折线图(两个 X 轴和一个 Y 轴,gsn_csm_x2y) 168
- 绘制折线图(一个 X 轴和两个 Y 轴,gsn_csm_xy2) 169
- 绘制折线图(一个 X 轴和三个 Y 轴,gsn_csm_xy3) 170
- 绘制折线图(一个 X 轴和一个 Y 轴,gsn_csm_xy) 13,30,32,169
- 绘制折线图(一个 X 轴和一个 Y 轴,X 轴从 0 开始,gsn_csm_y) 9,11,170
- 绘制组图(单位坐标系内,gsn_panel) 22,43,88,172
- 色板(读取,read_colormap_file) 73,91
- 色板(更换,gsn_define_colormap) 10,170
- 色板(合并,gsn_merge_colormaps) 172
- 色板(绘制,gsn_draw_colormap) 171
- 添加标识(gsn_add_polymarker) 20,160
- 添加标识(从 shapefile 文件中读取,gsn_add_shapefile_polymarkers) 161
- 添加并连接附属图形(gsn_attach_plots) 74,160
- 添加多边形(gsn_add_polygon) 20,42,159
- 添加多边形(从 shapefile 文件中读取,gsn_add_shapefile_polygons) 161
- 添加任意折线(gsn_add_polyline) 20,42,160
- 添加任意折线(从 shapefile 文件中读取,gsn_add_shapefile_polylines) 161
- 添加纬向平均图(gsn_csm_attach_zonal_means) 162
- 添加文字(gsn_add_text) 20,161

小波分析的边界效应(ShadeCOI) 73,174

输入输出

打开、创建或修改多个支持的数据文件(NetCDF 文件、HDF 文件、HDF-EOS 文件、CCM 文件、GRIB 文件或 OGR 文件,addfiles) 99

打开、创建或修改一个支持的数据文件(NetCDF 文件、HDF 文件、HDF-EOS 文件、CCM 文件、GRIB 文件或 OGR 文件,addfile) 37,99

读取 ASC II 文件(asciiread) 60,101

读取 ASC II 文件指定的行段(readAsciiTable) 137

读取 WRF 模式输出变量(wrf_user_getvar) 82,182

读取无格式二进制文件(顺序读取方式,fbinrecread) 47

读取无格式二进制文件(直接读取方式,fbindirread) 29,118

屏幕输出变量(print) 82,136

屏幕输出变量概要信息(printVarSummary) 136,217

输出二维数组至屏幕或 ASC II 文件(write_matrix) 26,158

输出为 ASC II 文件(asciiwrite) 102

输出为无格式二进制文件(顺序输出方式,fbinrecwrite) 120

输出为无格式二进制文件(直接输出方式,fbindirwrite) 47,118

WPS 中间文件(打开,wrf_wps_open_int) 190

WPS 中间文件(读取,wrf_wps_rddata_int) 190

WPS 中间文件(读取,wrf_wps_read_int) 190

WPS 中间文件(读取标题信息,wrf_wps_rdhead_int) 190

WPS 中间文件(关闭,wrf_wps_close_int) 189

WPS 中间文件(写入,wrf_wps_write_int) 191

统计分析

多元线性回归系数(reg_multlin) 138

EOF 检验(eofunc_north) 47,112

EOF 空间模态(eofunc_n_Wrap) 111

EOF 时间序列(eofunc_ts_n_Wrap) 113

傅里叶分解(ezfftf) 116

傅里叶合成(ezfftb) 116

滤波器系数(filwgtls_lanczos) 121
去除线性趋势(输入场不可存在缺测值,dtrend_n) 111
去除线性趋势(输入场可存在缺测值,dtrend_msg_n) 110
student-t 检验(student_t) 39,143
SVD 分解(输出为左同质、左异质、右同质、右异质场,svdcov) 144
SVD 分解(输出为左、右奇异向量,svdcov_sv) 145
SVD 分解(输入为标准化场,输出为左同质、左异质、右同质、右异质场 svdstd) 146
SVD 分解(输入为标准化场,输出为左、右奇异向量,svdstd_sv) 146
t 检验(ttest) 108
相关系数(交叉相关,escrr) 53,115
相关系数(同时相关,escorc) 115
相关系数的显著性检验(rtest) 53,140
小波变换(wavelet) 71,152
一元线性回归系数(输入场为多维数组,regCoef_n) 138
一元线性回归系数(输入场为一维数组,regline) 57,139

数组操作及查询

创建变量(new) 12,22,76,134
创建等差数列(浮点型,fspan) 12,122,217
创建等差数列(整型,ispans) 9,127
短型数据转换为浮点型(short2flt) 9,142
个数统计(值为 True,num) 134
数值小数部分的舍取(ceil,floor) 104,122
数组的局地极大值(输入为多维数组,local_max) 129
数组的局地极大值(输入为一维数组,local_max_1d) 130
数组的局地极小值(输入为多维数组,local_min) 130
数组的局地极小值(输入为一维数组,local_min_1d) 130
数组的逻辑判断(全为 True,all) 101
数组的逻辑判断(缺测,ismissing) 77,127,218
数组的逻辑判断(任一为 True,any) 101
数组的逻辑判断(值为 True,where) 39,157
数组结构(dimsize) 31,109
数组扩展(conform) 46,71,106

- 数组所有元素的标准差(stddev) 143
- 数组所有元素的乘积(product) 136
- 数组所有元素的和(sum) 143
- 数组所有元素的平均(avg) 103
- 数组所有元素的最大值(max) 131
- 数组所有元素的最小值(min) 132
- 数组指定维的标准差(dim_stddev_n_Wrap) 109
- 数组指定维的标准化(dim_standardize_n_Wrap) 109
- 数组指定维的方差(dim_variance_n_Wrap) 39,109
- 数组指定维的和(dim_sum_n_Wrap) 109
- 数组指定维的加权和(dim_sum_wgt_n_Wrap) 109
- 数组指定维的距平值(dim_rmvmean_n_Wrap) 37,109
- 数组指定维的累加(dim_cumsum_n_Wrap) 108
- 数组指定维的平均(dim_avg_n_Wrap) 38,107
- 数组指定维的最大值(dim_max_n) 108
- 数组指定维的最小值(dim_min_n) 109
- 数组转换(多维至多维,reshape) 137
- 数组转换(多维至一维,ndtooned) 85,133
- 数组转换(一维至多维,onedtond) 71,134
- 元素的下标变量(不等于指定值,get1Dindex_Exclude) 124
- 元素的下标变量(等于指定值,get1Dindex) 124
- 元素的下标变量(值为 True,ind) 36,125
- 元素下标变量的转换(ind_resolve) 126
- 最大值的下标变量(输入场为一维数组,maxind) 131
- 最小值的下标变量(输入场为一维数组,minind) 132

其他

- shell 命令(system、systemfunc) 146

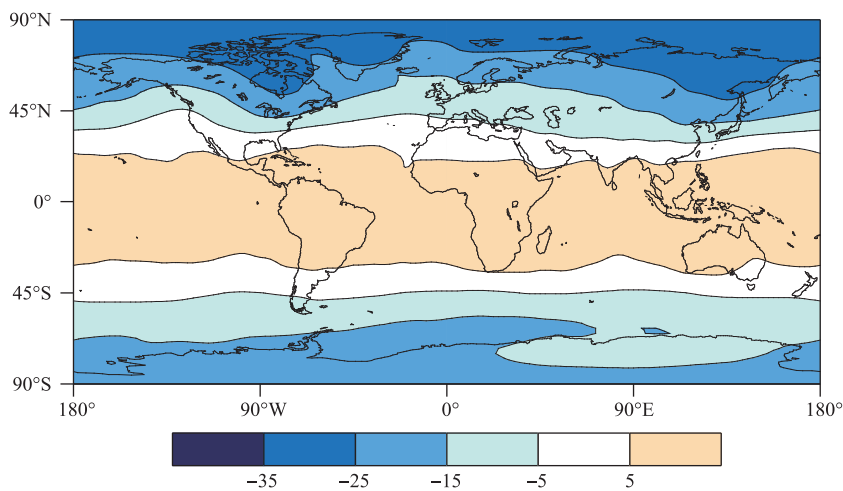


图 2.2 气温图(单位:°C)

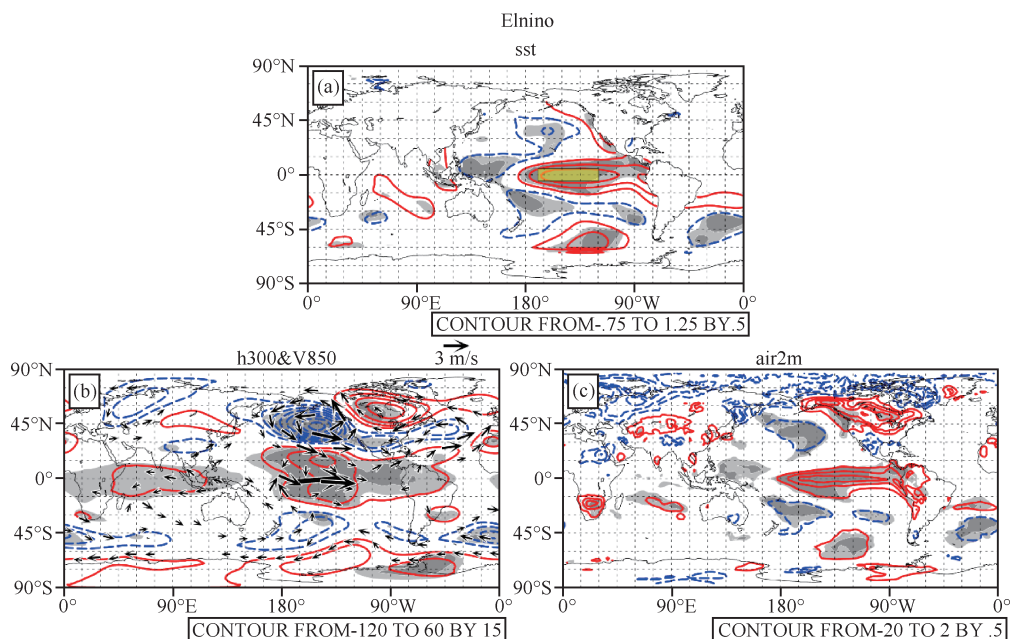


图 4.1 厄尔尼诺年环流合成图

(a) sst, (b) h300 (等值线)与 V850(箭头), (c) air2m;图中深(浅)阴影表示通过置信度为 0.01(0.05)显著性检验的区域;红色实线(蓝色虚线)表示正(负)异常

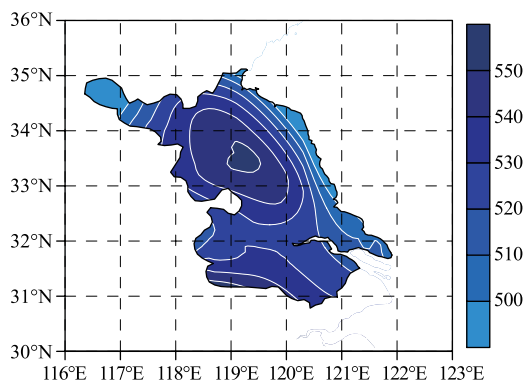


图 6.1 江苏省 1979 年至 2008 年夏季
气候平均降水量(单位:mm)

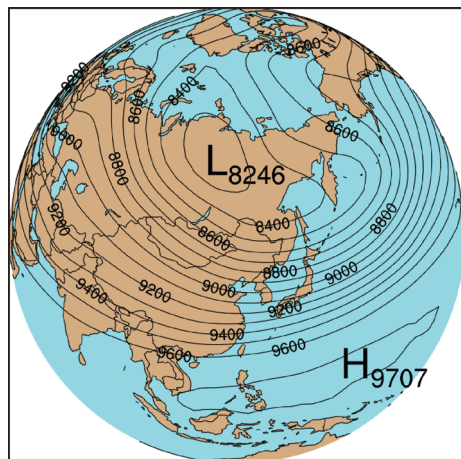


图 6.2 1979 年 1 月 300 hPa 位势
高度场(单位:gpm)

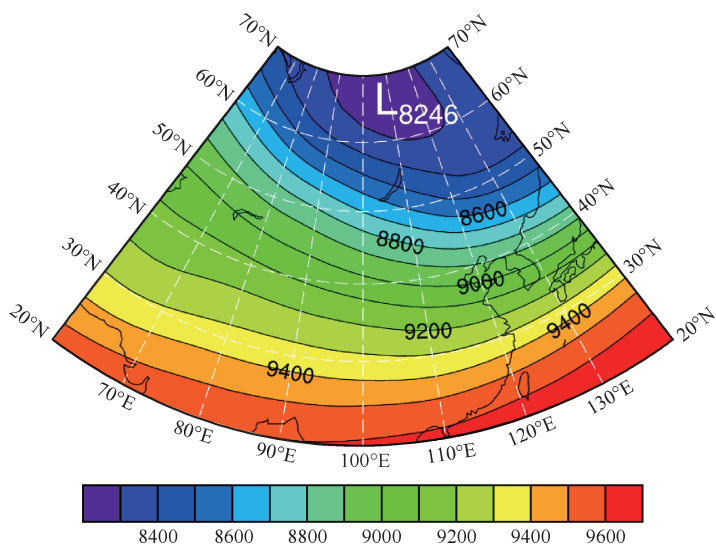


图 6.3 1979 年 1 月 300 hPa 位势高度场(单位:gpm)

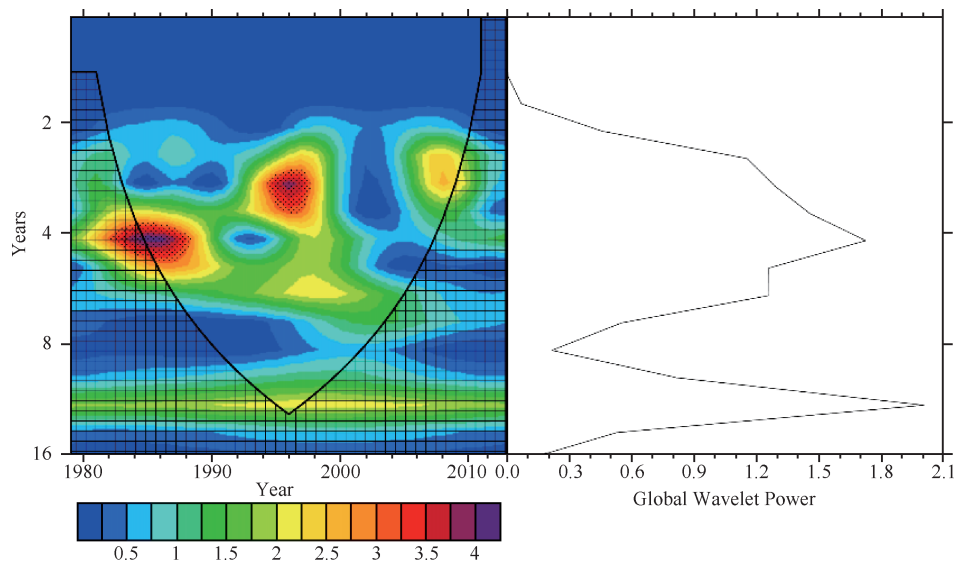


图 6.4 1 月 ENSO 指数 Morlet 小波图

(左图中网格点处为通过 95% 显著性检验区域, 网格处为边界效应, 右图中为功率谱)

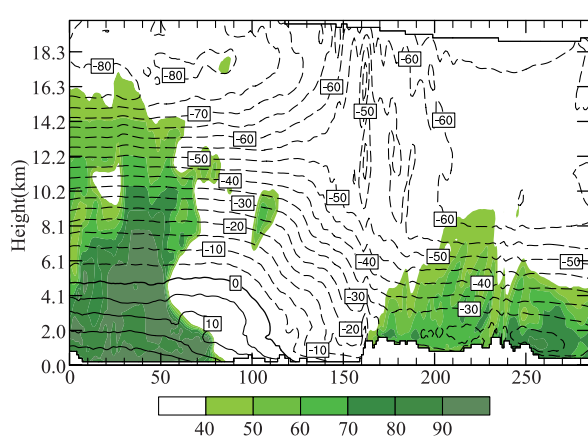


图 7.3 剖面图

(绿色填色为相对湿度, %; 等值线为温度, 单位: °C)

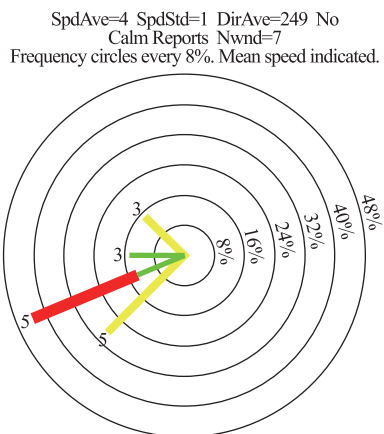


图 7.4 风玫瑰图

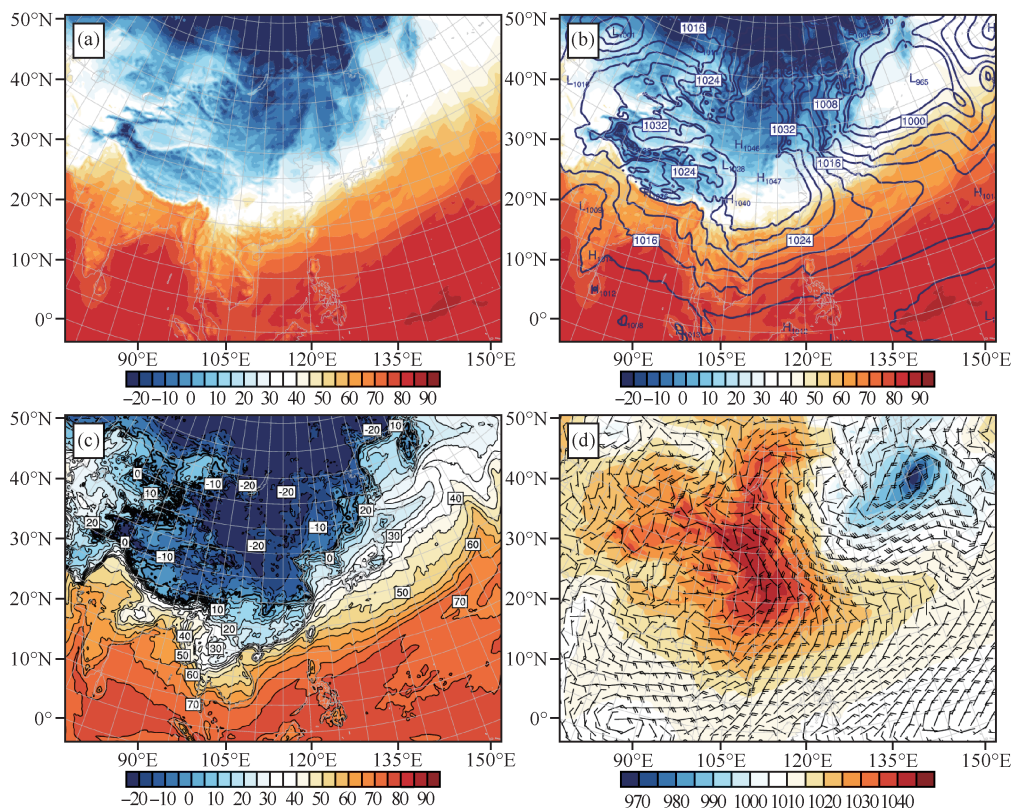


图 7.5 组图

(a) 地表温度, 单位: $^{\circ}\text{F}$; (b) 地表温度 (填色) 及海平面气压 (等值线, 单位: hPa); (c) 地表露点温度, 单位: $^{\circ}\text{F}$; (d) 海平面气压 (填色, 单位: hPa) 及风场 (风杆, 单位: 节)

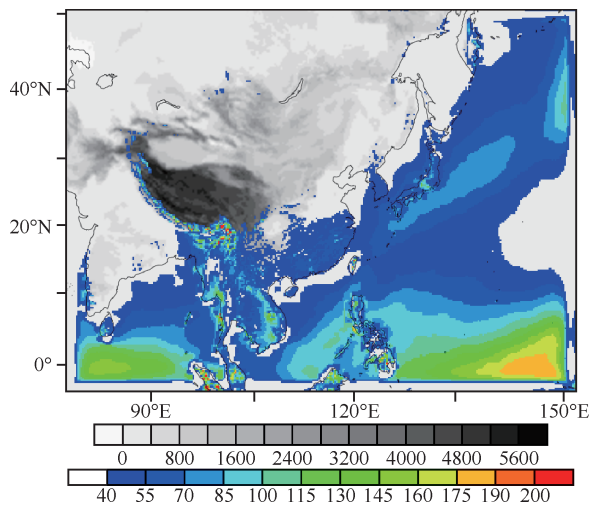
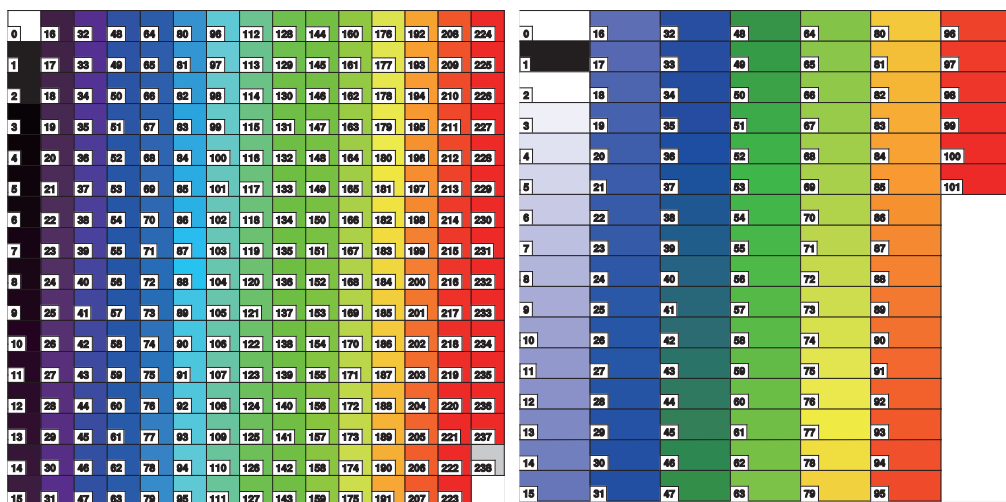


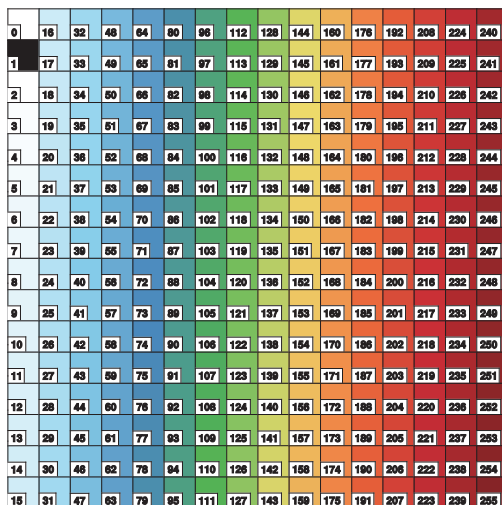
图 7.6 利用 gsn 绘图

(灰色阴影为地形高度, 单位: m; 彩色填色为降水量, 单位: mm)



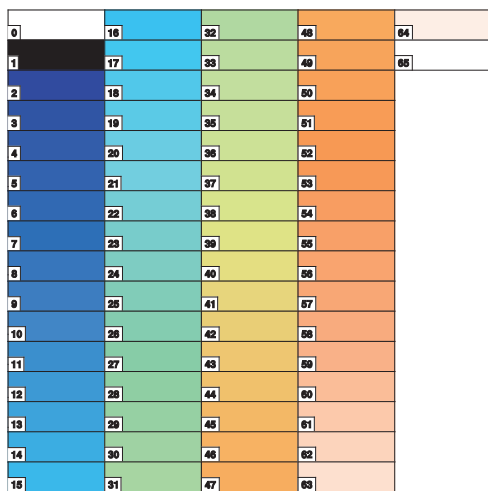
(a) rainbow+gray

(b) WhBlGrYeRe

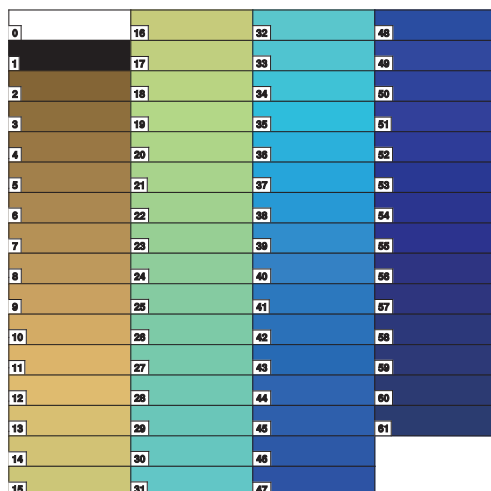


(c) WhiteBlueGreenYellowRed

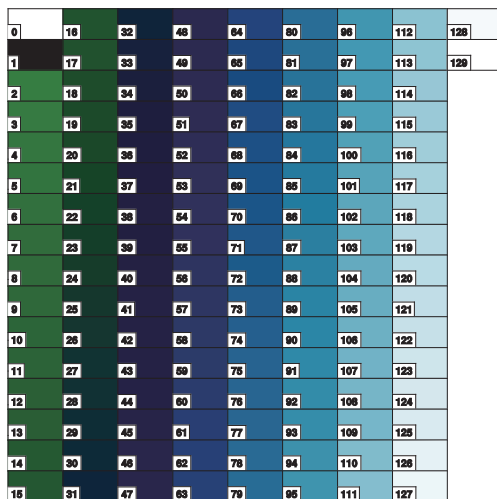
图 I. 1 三种彩虹色色板



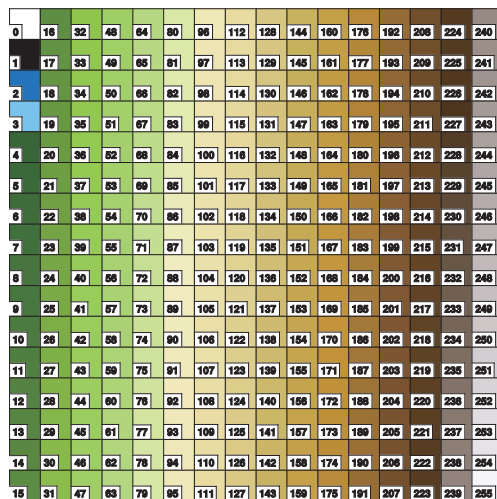
(a) cmp_haxby



(b) GMT_drywet



(c) MPL_ocean

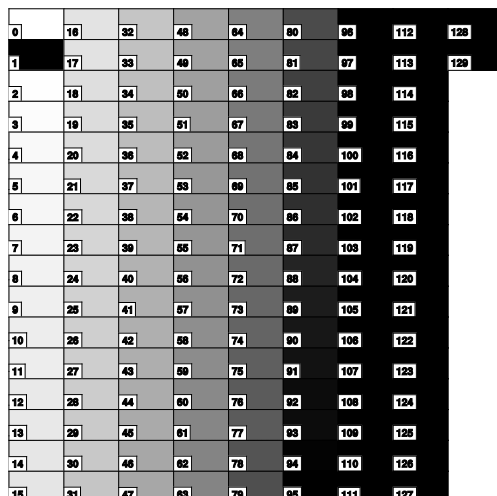


(d) OceanLakeLandSnow

图 I. 2 四种地球海洋色板

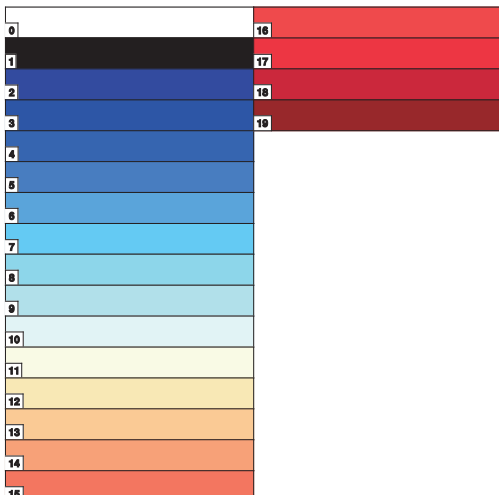


(a) gsltod

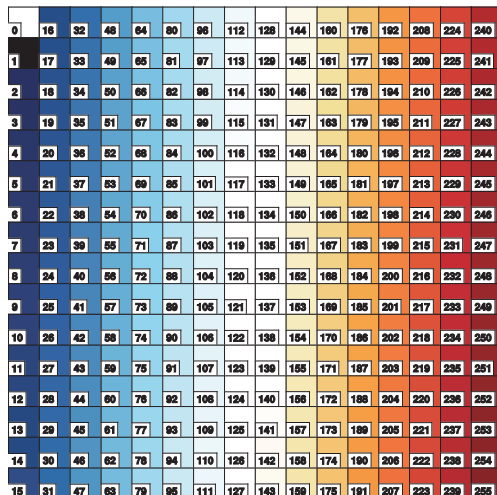


(b) MPL_gist_yarg

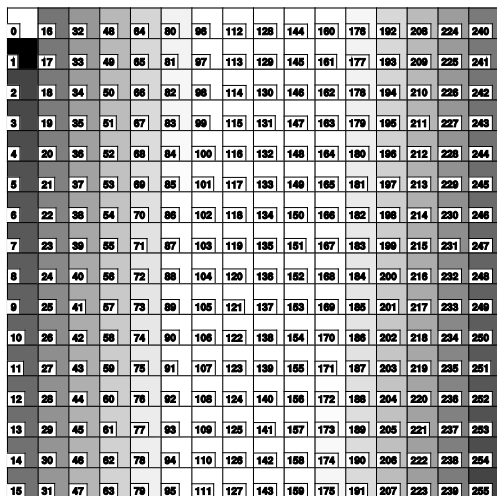
图 I.3 两种灰色色板



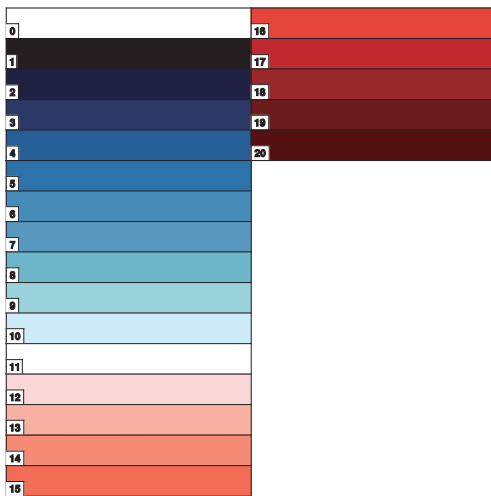
(a) BlueDarkRed18



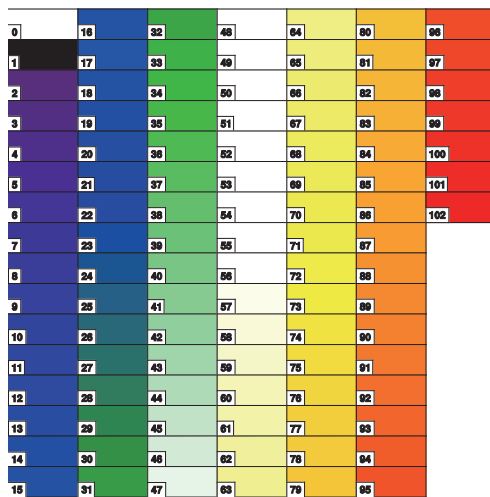
(b) BlueWhiteOrangeRed



(c) GrayWhiteGray



(d) temp_diff_18lev



(e) ViBiGrWhYeOrRe

图 I. 4 五种白色居中色板

责任编辑：黄红丽

NCL SHUJU CHULI YU HUITU SHIXI JIAOCHENG

ISBN 978-7-5029-6567-9



定价：40.00元