

NCL

程序设计入门

蔡宏珂 陈权亮 范广洲 衡志炜 花家嘉 编著

NCL 程序设计入门

蔡宏珂 陈权亮 范广洲 衡志炜 花家嘉 编著

内容简介

本书介绍了大气科学领域当前流行的高级程序设计语言 NCAR Command Language(NCL)。本书内容详实,包括语法基础、常用函数和过程简介、图形属性简介、应用技巧专题等部分,以较大篇幅全面介绍了 NCL 具有鲜明特色的常用函数、过程和图形属性,并着重讲述了安装和运行、Linux 系统操作技巧、官方网站目录结构、代码的一般结构、读写文件、日期时间的处理等区别于其他程序设计语言的 NCL 应用技巧。

本书语言简练、结构清晰,面向 NCL 程序设计的初、中级用户,适用于学习气象数据分析与可视化的本科生、研究生,也适用于从事科学研究和业务应用的技术人员,既可作为 NCL 程序设计入门教材,也可作为 NCL 程序设计备查手册。

图书在版编目(CIP)数据

NCL 程序设计入门 / 蔡宏珂等编著. — 北京 : 气象出版社, 2017. 11(2018. 3 重印)

ISBN 978-7-5029-6684-3

I. ①N… II. ①蔡… III. ①气象观测-应用程序-程序设计 IV. ①P41-39

中国版本图书馆 CIP 数据核字(2017)第 284387 号

NCL 程序设计入门

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

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

网 址: <http://www.qxcbs.com>

责任编辑: 简学东

责任校对: 王丽梅

封面设计: 八 度

印 刷: 三河市百盛印装有限公司

开 本: 787 mm×1092 mm 1/16

字 数: 506 千字

版 次: 2017 年 11 月第 1 版

定 价: 68.00 元

邮政编码: 100081

E-mail: qxcbs@cma.gov.cn

终 审: 张 斌

责任技编: 赵相宁

印 张: 21

彩 插: 3

印 次: 2018 年 3 月第 2 次印刷

前 言

NCL 是目前大气科学领域常用的程序设计语言,具有语法简练、功能丰富的特点,尤其适于本领域的文件读写、数据分析和图形绘制。本书紧跟 NCL 发展动态,介绍了 NCL 语法基础,搜集整理了 NCL 的逾千条函数和全部图形属性,以功能用途分类介绍,并针对应用专题介绍了使用技巧。

一般的高级语言具有相似的语法特征,往往可以一通百通,而 NCL 的特色功能主要通过函数或过程、图形属性实现,这两方面是 NCL 应用的重点和难点。本书的重心就放在函数和过程以及图形属性的介绍上,全面列举了常用函数过程和图形属性。在介绍函数时,详细列举了函数的功能、参数、描述,部分函数给出了应用示例。在介绍图形属性时,详细列举了图形属性的功能、默认值和备用值。本书力求简化 NCL 应用,希望能够像做填空题一样使用函数过程、像做选择题一样设置图形属性。

NCL 初学者一个很大的症结就是不熟悉 NCL 的函数和图形属性。他们不知道有这个函数可以实现这个功能、有那个属性可以修改那个图形,不知道 NCL 有这么多现成的实用语句可以使用。他们一动手写程序,下意识地回到以往 C 语言或者 Fortran 语言课堂,习惯性地用“加减乘除与或非”来实现算法。鉴于此,笔者有心做出一个有相当容量的目录,分门别类地罗列了函数的用途和图形属性的对象,便于读者检索。希望这是一条从“hello world”通往数据分析和可视化的终南捷径,能够帮助读者少走弯路。笔者真诚建议本书读者熟悉本书目录。

本书可以帮助读者全面掌握 NCL 语法,并快速熟悉函数和图形属性的使用方法,在进行 NCL 程序设计时能方便快捷地使用内置功能实现数据分析和可视化。因此,本书虽名为入门,其实也可以作为语法手册而常备书案。然而,本书基于笔者笔记,最早的内容可以追溯到十年前,屡经补充更新,仍不脱本人使用习惯。本书并非有意识地编撰,想到哪儿就写到哪儿,没有计划、没有提纲、没有针对性,存在先天不足。实际上,在付梓之前的这一刻,我仍希望能有精力和时间将其完善,尤其是增补应用专题技巧这一部分,但这需要在教学中积累更多实用经

验和调试案例。因此,本书必须名为入门,在 NCL 程序设计的较高应用层面上尚存很大待挖掘的潜力。

这本书是笔者使用 NCL 开展大气科学数据分析和可视化、从事科研和教学工作的经验总结,可以说是凝聚了笔者多年的心血。尽管才疏学浅,但仍希望能够以过来人的一点点经验,为后来者清理前行路上的障碍,帮助新人尽快上手。本书最初的出发点,是给笔者自己一个交待,而最根本的目的,是希望能够为人所用。如果读者觉得本书有一点点用处,能够对读者的学习和工作有所促进,那么编写本书的目的便达到了。“自闭桃源称太古,愿栽大木拄长天”,我把读者的收获看作我的功德。能够为本领域同仁做出一点贡献,是我多年的心愿,即使笔者本人眇眇之身不入方家法眼。此次斗胆腆颜自荐出版,蒙气象出版社俯允,让我有机会完成这个心愿,实在是感激不尽。本书也得到了成都信息工程大学的大力支持,感谢大气科学学院和研究生处长期的指导和鼓励。

对于本书的面世,我必须特别感谢我的老师——中国科学技术大学周任君先生。他是我在科学研究、程序设计两方面的启蒙老师,我是他的开山弟子。能够获得周老师手把手地指导、将我引入 NCL(当然也包括别的程序设计方法)的大门,是我在中国科学技术大学学习期间在个人能力方面最有实用价值的收获,只可惜我所学不如周老师之万一。谨以此书向周老师表达诚挚、深切的感谢!

感谢我最亲爱的学生飘飘、嘉雯和桃子一直以来给予本书的支持和帮助。投我以木瓜,报之以琼琚。匪报也,永以为好也!投我以木桃,报之以琼瑶。匪报也,永以为好也!投我以木李,报之以琼玖。匪报也,永以为好也!谨以此书纪念我们的淳朴友谊并寄托我对她们的美好祝福!

最后,以“己欲立而立人,己欲达而达人”来剖白心迹,诚邀读者和我交流,指正本书疏漏,充实本书内涵。“沉舟侧畔千帆过,病树前头万木春”,期待本书能陪伴读者更进一步。

蔡宏珂@CUIT
2017 年 10 月

目 录

前言

第一章 语法基础	(1)
一、数据类型	(1)
二、保留字	(1)
三、语法符号	(1)
四、运算符	(2)
五、变量	(2)
六、顺序结构、循环结构和分支结构	(2)
七、数组	(3)
八、运行方式	(4)
九、自定义函数和过程	(4)
第二章 文件 I/O	(5)
一、文件格式	(5)
二、常用函数	(5)
三、输入变量	(5)
四、特殊文件类型读写方式	(6)
第三章 数据显示和分析	(10)
一、概述	(10)
二、扩展函数库	(10)
三、常用函数	(10)
第四章 外部 FORTRAN/C 程序调用	(12)
一、WRAPIT	(12)
二、指定外部函数位置	(12)
三、参数传递	(12)
四、调用 F77(Fortran77)	(12)
五、调用 F90(Fortran90)	(13)
六、调用商业库	(13)

第五章 数据可视化	(15)
一、概述	(15)
二、一般步骤	(15)
三、图形概念	(16)
四、颜色	(16)
五、图形叠加	(16)
六、子图组合	(17)
七、文本和格式文本修饰符	(18)
八、多边形、多边框、图形符号	(18)
九、命名规范	(19)
第六章 常用函数和过程简介	(20)
一、变量操作和显示	(20)
(一)变量(数据)的显示输出	(20)
(二)其他	(22)
二、数组建立、查询和操作	(24)
(一)数组变形和元素整改	(24)
(二)查询元素索引	(27)
(三)建立特殊数组	(30)
(四)列表	(31)
(五)其他	(33)
三、数据类型测试	(34)
(一)变量类型	(34)
(二)元数据	(35)
(三)文件	(37)
四、数据类型转换	(38)
五、字符串操作	(41)
(一)特殊符号	(41)
(二)字符串分割连接	(43)
(三)子字符串	(44)
(四)字符串格式化	(48)
(五)其他	(49)
六、时间日期	(49)
(一)时间日期查询	(49)
(二)建立时间日期数组	(50)
(三)时间日期形式转化	(51)
(四)按日期统计	(54)
七、元数据	(55)

八、文件输入/输出	(57)
(一)受支持格式文件的打开	(57)
(二)受支持格式文件的查询与定义	(58)
(三)ASCII 文本文件	(64)
(四)二进制文件	(65)
(五)Vis5D+格式文件	(71)
(六)ARW WRF 模式输出文件	(71)
九、系统	(71)
十、数学分析	(74)
(一)通用函数	(74)
(二)数据排序	(76)
(三)微积分	(77)
(四)基本统计量	(78)
(五)统计	(80)
(六)概率分布	(89)
(七)假设检验	(92)
(八)傅里叶分析	(93)
(九)线性回归	(96)
(十)线性代数	(99)
(十一)特殊函数	(101)
(十二)相关系数	(102)
(十三)维度分析	(104)
(十四)滤波	(109)
(十五)谱分析	(111)
(十六)小波分析	(113)
(十七)平滑	(114)
十一、图形对象操作	(116)
十二、颜色	(122)
十三、绘图	(128)
十四、绘图空间	(156)
十五、格点化	(157)
十六、插值	(164)
(一)球面无规则格点(站点)三次样条插值	(164)
(二)改进 Shepard 算法计算三维格点插值	(166)
(三)二维平面的自然邻点插值	(167)
(四)无结构格点(站点)数据的反距离加权插值	(170)
(五)线性插值	(171)
(六)样条插值	(173)
(七)一维结构的三次样条插值	(178)

(八)二维结构的三次样条插值	(179)
(九)三维结构的三次样条插值	(183)
(十)其他	(186)
十七、经纬度	(189)
十八、累积分布	(198)
(一)二项分布	(198)
(二) χ^2 分布	(199)
(三) Γ 分布	(200)
(四)正态分布	(201)
十九、经验正交函数(EOF)分解	(201)
二十、奇异值分解(SVD)	(204)
二十一、气候分析	(206)
二十二、气象分析	(212)
二十三、海洋学分析	(223)
二十四、随机数生成器	(223)
二十五、球谐函数	(226)
二十六、CESM	(226)
二十七、WRF	(227)
第七章 图形属性简介	(228)
一、图形属性命名规律	(228)
二、页面控制	(229)
三、拼图控制	(229)
四、折线图 and 散点图常用属性	(231)
五、柱状图 and 直方图控制	(234)
六、矢量场图常用属性	(235)
七、流场图常用属性	(243)
八、等值线图常用属性	(245)
九、地图设置常用属性	(254)
十、图形标注 and 标题控制	(261)
十一、坐标轴常用属性	(264)
十二、色标常用属性	(270)
十三、图例常用属性	(275)
十四、图形符号常用属性	(280)
十五、文本常用属性	(281)
第八章 应用技巧专题	(283)
一、安装和运行	(283)
二、Linux 系统操作技巧	(289)

三、NCL 官方网站目录结构	(293)
四、代码的一般结构	(294)
五、读写文件	(296)
六、日期时间的处理	(301)
参考文献	(304)
附录	(305)

第一章 语法基础

一、数据类型

有符号数			无符号数			非数值型
数据类型	位	后缀	数据类型	位	后缀	
double	64	d 或 D				string
int64	64	q	uint64	64	Q	character
float	32					graphic
long	32	l	ulong	32	L	file
integer	32		uint	32	I	logical
short	16	h	ushort	16	H	list
char	8	B	byte	8	b	

数值类型转换:隐式转换以不丢失信息为原则,否则会产生致命错误;若必须丢失信息,则使用显式转换。

科学记数法:单精度 float 型采用 e 或 E,双精度 double 型采用 d 或 D。

二、保留字

以下为 NCL 保留字,定义函数和变量名,请勿使用。

begin, break, byte, character, continue, create, defaultapp, do, double, else, end, external, False, file, float, function, getvalues, graphic, if, integer, load, local, logical, long, new, noparent, numeric, procedure, quit, Quit, QUIT, record, return, setvalues, short, string, then, True, while

三、语法符号

- ;
开始注释
- @
创建/引用属性
- !
创建/引用命名维度
- &
创建/引用坐标变量
- { ... }
用于坐标带下标
- \$
通过内置函数 addfile 输入输出变量时封装字符串
- (/ ... /)
创建数组,只包含值,不包含属性、命名维度等
- :
用于数组索引
- |
用于命名维度分隔符

\	续行号(行尾)
::	用于调用外部代码分隔符
—>	用于输入输出支持的数据格式
[...]	类型列表下标变量

四、运算符

代数运算符：

+	加法/字符串连接	/	除法
-	减法	%	模(整数运算)
*	乘法	#	矩阵乘法
^	乘方	>和<	大于/小于

逻辑运算符：

.lt.	小于	.le.	小于等于
.gt.	大于	.ge.	大于等于
.ne.	不等于	.eq.	等于
.and.	与	.or.	或
.xor.	异或	.not.	非

五、变量

变量定义必须以字母开头,允许字母、数字和下划线,请注意,不要使用保留字做变量名。
元数据为变量的辅助信息。元数据可以用 delete()函数删除。

缺失数据属性 @_FillValue 为元数据的保留属性。变量的缺失值用 ismissing()函数测试,而不能用 Data.eq. Data@_FillValue 判断。

值分配:赋值源为常量、表达式或数组创建符号(/.../)。只将值赋给目标变量,忽略属性、命名维度等。

变量分配:赋值源为变量,不但将值赋给目标变量,还包含属性、命名维度等。将一个源变量复制到另一个目标变量,如 y=x,要求二者形状和数据类型相同或源变量强制转换以匹配目标变量,目标变量的元数据将被覆盖。

六、顺序结构、循环结构和分支结构

begin 语句体 end	do n=start,end(,step) 循环体 end do	do while(标量逻辑表达式) 循环体 end do	if(标量逻辑表达式) then 语句体 else 语句体 end if
	break 退出循环	continue 跳过本次循环	

同其他解释性语言一样,NCL 循环效率低下,尽量使用数组或内置函数替代。如果必须使用多层循环,可以用外部 C 或 Fortran 程序替代。

逻辑判断从左至右执行,因此将最可能的“假”放在左边。

七、数组

1. 建立数组

(/.../)和函数 new(数组大小/形状,数据类型,[缺失值])手工建立数组。

2. 下标

起始索引(默认为 0):结束索引(默认为 N-1):可选步长(默认为 1)。

维数——最右边维度变化最快(类似于 C 语言)。

3. 引用

{...}形式表示使用命名维度,数组索引不用{...}。

假定 T 为三维数组(nt,ny,nx),命名维度为(time,lat,lon):

T	整个数组,不必使用 T(:, :, :)
T(0, :, :, :5)	0 号 time,所有 lat,lon 间隔为 5
T(0, :, :-1, :50)	0 号 time,反转 lat,0~50 号 lon
T(:, 1, 45, 10:20)	0~1 号 time, 46 号 lat, 11~21 号 lon
X=T(:, {-20:20}, {90:290:2})	所有 time, lat 值-20~20,lon 值 90~290 间隔为 2

4. 缩维

假定 T(nt,nz,ny,nx):

T1=T(5, :, 12, :)	产生 T1(nz,nx)
T2=T(:, :, :, 0)	产生 T2(nt,nz,ny)
T3=T(5:5, :, 12, :)	产生 T3(1,nz,nx)
T4=T(5:5, :, 12:12, :)	产生 T4(1,nz,1,nx)

5. 变形

假定 T(time,lat,lon):

t=T(lat :,lon :,time :)	→t(lat,lon,time)
t=T(time :{lon 90:120},{lat -20:20})	→t(time,lon,lat)所有 time,lon 值 90~120,lat 值-20~20

6. 命名维度

假定 T(ntime,nlat,nlon):

使用 T! 0="time",T! 1="lat",T! 2="lon"命名;
使用 TIME=T! 0,LAT=T! 1,LON=T! 2 获取名字字符串"time","lat","lon"。
命名维度可以用 delete()函数删除。
使用命名维度做数组变形:

X=T(lat :,lon :,time :)	变为(lat,lon,time)
X=T({lat -20:20},lon 30:42,time :)	变为(lat,lon,time),并选择 lat 值为-20~20,lon 为 30~42 号元素,时间为全部时间

八、运行方式

1. 命令行交互

ncl [选项] [命令行参数]

ncl> commands

ncl> quit

将命令保存为文件

ncl> record "file_name"

ncl> stop record

2. 脚本

ncl [选项] [参数] Script.ncl

ncl [选项] [参数] Script.ncl >&! out

ncl [选项] [参数] Script.ncl >&! out &

>&! 和 & 是 csh 和 tcsh 语法

选项包括 -hnpv, 含义为:

-h: 输出提示信息并退出;

-n: 在 print() 函数中不枚举维度, 即只依次显示数值, 数值前不显示其数组索引;

-p: 函数 system() 的输出不分页;

-x: 显示 NCL 命令, 用于调试;

-V: 显示 NCL 版本并退出。

参数为变量列表, 形如 VarName1=VarValue1, VarName2=VarValue2, ..., 通过此种形式从 shell 向 NCL 脚本输入参数。

九、自定义函数和过程

- 使用函数和过程前必须先用 load 加载函数脚本, 或定义环境变量 NCL_DEFAULT_SCRIPTS_DIR;

- 函数和过程的参数传递为引用传递, 可以严格限定参数的数据类型和数组维度, 也可以不加任何限制;

- 自定义函数和过程基本结构:

undef("function_name")

function

function_name(declaration_list)

local local_variable_list

begin

[statement(s)]

return (return_value)

end

undef("procedure_name")

procedure

procedure_name(declaration_list)

local local_variable_list

begin

[statement(s)]

end

注: undef 用于清除以前的同名函数或过程, 以避免多次加载, local 用于声明局部变量, 二者非必须但建议使用。

第二章 文件 I/O

一、文件格式

主要文件格式包括：

二进制文件；

ASCII 文本文件；

NetCDF("r","w","c"):".nc",".cdf",".netcdf";

GRIB1 和 GRIB2 ("r"):".gr",".gr1",".grb",".grib",".grb1",".grib1",".gr2","grb2",".grib2";

HDF("r","w","c"):".hdf",".hd",只支持 ScientificDataSet;

HDFEOS("r"):".hdfEOS","he2","he4";

CCM("r"):".ccm",只支持 Cray。

二、常用函数

ncl_filedump:查看文件信息。

ncl_convert2nc:将 GRIB 文件转换成同名 netCDF 文件。

setfileoption,filevardef,filevarattdef,filedimdef,fileattdef:设置文件及数据属性。

addfile,addfiles:打开文件获取文件句柄。文件句柄包含文件全局属性,可以用 getvaratts 获取。

isfilepresent:测试文件存在。

getfilevarnames,getfilevardims,getfilevaratts,getfilevardimsizes,getfilevartypes:查询文件数据属性。

isfilevar,isfilevaratt,isfilevardim,isfilevarcoord:测试文件数据属性。

三、输入变量

1. 用文件句柄和 addfile 函数打开文件

f=addfile(fname,ext,status)

文件定位优先使用 ext 指定的格式,若找不到 ext,则以 ext 格式打开 fname.* 文件;

status 包含:r(只读)、c(创建 netCDF 或 HDF4)、w(读写 netCDF 或 HDF4)。

2. 使用 f-> 指向文件数据集,向变量赋值

x=f->X:读取数据和所有元数据;

x=(f->X):读取数据和元数据@_FillValue。

3. \$ 封装变量名

文件数据引用包含非数字字母字符:

```
x=f->$A+B...C$
```

文件数据引用本身就是变量:

```
vars=("/T","U","V")
```

```
x=f->$vars(n)$
```

文件数据名称由字符串变量和字符串常量做连接运算,还须添加括号,以保证运算优先级:

```
x=f->$("var"+vars(n))$
```

4. 打开多个文件

打开文件夹内所有 ann*.nc 文件:

```
files=systemfunc("ls ann*.nc")
```

```
f=addfiles(files,"r") f 为 list 数据类型
```

```
ListSetType(f,"cat") 多个文件中数据默认连接方式,不必写出
```

```
T=f[:]->t t 必须存在于各个文件且形状相同
```

多个文件中数据连接方式用 ListSetType() 设定。ListSetType(f,"cat") 为默认连接方式,直接连接数据;ListSetType(f,"join") 新建维度存放数据。netCDF 文件有相应函数 ncread() 和 nccat()。

四、特殊文件类型读写方式

1. 二进制文件读写

Fortran 接口读取无格式顺序文件 fbinrecread, fbinread, 读取直接存取文件 fbindirread, 获取无格式顺序数据数量 fbinnumrec 和 C 接口读取文件 cbinread, 以及 Cray-COS 系统接口 craybinrecread, craybinnumrec。

Fortran 接口写无格式顺序文件 fbinrecwrite, fbinwrite, 写直接存取文件 fbindirwrite 和 C 接口写文件 cbinwrite。

示例:读取 GrADS 二进制数据文件

GrADS 控制文件 365 天日降水量

```
DSET gpcp_1dd_v1.lnx.1999
UNDEF -999.0
TITLE GPCP 1 Degree Daily (1DD) Precip Analysis (Version 1)
OPTIONS little_endian
XDEF 360 LINEAR 0.5 1.0
YDEF 180 LINEAR -89.5 1.0
ZDEF 01 LEVELS 1 2
TDEF 365 LINEAR 1JAN1999 1dy
```



```

VARS 1
rain 1 99 ch01 GPCP 1DD Precip (mm/day)
ENDVARS
NCL 代码
nlat=180
nlon=360
ntim=365
setfileoption("bin","ReadByteOrder","LittleEndian")
x=fbindirread("gpcp_1dd_v1.lnx.1999",0,(/ntim,nlat,nlon/),"float")
x! 0="time"
x! 1="lat"
x! 2="lon"
x&time=time
x&lat=latGlobeFo(nlat,"lat","latitude","degrees_north")
x&lon=lonGlobeFo(nlon,"lon","longitude","degrees_east")
x@long_name="GPCP 1DD Precip"
x@units="mm/day"

```

2. ASCII 文件

读取 ASCII 文件: `asciiread`, `readAsciiHeader` 和 `readAsciiTable`;

写入: `asciiwrite` 和 `write_matrix`。

示例: 读取简单 ASCII 表

ASCII 数据表

1881	-999.9	0.2	-999.9	-999.9	1.5	-999.9	-999.9	-0.2
1882	-1.7	-0.5	0.6	0.1	0.9	-1.9	-3.5	-4.6
1995	-1.0	-0.8	0.4	-1.8	-1.2	-0.4	0.6	-0.1

NCL 代码:

```

ncols=9
nrows=3
ksoi=asciiread("ascii.in",(/nrows,ncols/),"float")
yrs=ksoi(:,0)
data=ksoi(:,1:)
data@_FillValue=-999.9

```

示例: 读取带表头的 ASCII 表

ASCII 数据表