

# 地球科学学院大气科学系《诊断分析与绘图实验》报告

## 实验二 数组维数、下标和变量属性

姓名	学号	成绩
马群	20201170333	1190

### 一、目的：

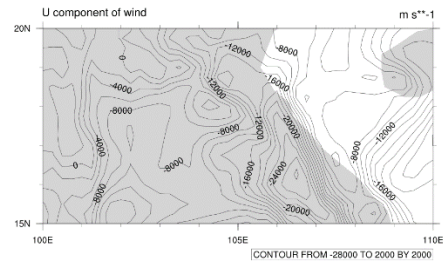
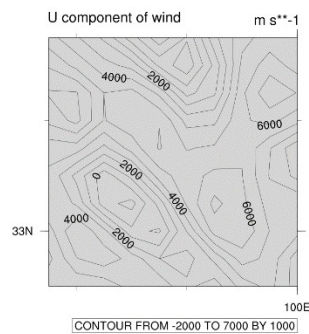
掌握查询数据信息和变量信息的方法；掌握数组维数和下标的操作；掌握获取变量属性和新建变量属性的方法。掌握标准下标和地球坐标的对应关系；了解图形输出的基本操作。

### 二、方法：（见实验指导书）

三、回答习题（可逐题回答，也可以把执行的命令或脚本一次写完，把要说明的内容加成注释或在最后说明）：

```
begin
f = addfile("/home/xiaoma/nc/0328/era5_daily_201901.nc", "r")
print(f)
u = f->u
printVarSummary(u)
a1 = u(0,1:10,11:20)
a2 = u(0,{15:20},{100:110})
print(u@units)
print(a1@units)
wks = gsn_open_wks("X11", "test")
res = True
res@gsnAddCyclic = False
;;;;;;;;;;;;;a1
res@mpMinLatF = 32.5
res@mpMaxLatF = 34.75
res@mpMinLonF = 97.75
res@mpMaxLonF = 100
;;;;;;;;;;;;;a2
;res@mpMinLatF = 15
;res@mpMaxLatF = 20
;res@mpMinLonF = 100
;res@mpMaxLonF = 110
;;;;;;;;;;;;;
plot = gsn_csm_contour_map_ce(wks, a1, res)
```

end



说明:

## 2. 文件中包含

latitude, longitude, time, z (Geopotential), q (Specific humidity), t (Temperature), u (Eastward\_wind), v (Northward\_wind) 共八个变量, 以 latitude, time 和 z 为例:

latitude 包括单位 units, 全称 long\_name; time 包括单位 units, 全称 long\_name, 时间变量 calendar 为 gregorian 世界时; z 包括系数 scale\_factor 和偏移值 add\_offset [类似一次线性函数,  $y = ax + b$ ], \_FillValue 和 missing\_value 缺省值为 -32767, 单位 units 为 (m/s)<sup>2</sup>, 全称 long\_name 和标准名称 standard\_name

```
float latitude ( latitude )
units :      degrees_north
long_name :   latitude

integer time ( time )
units :      hours since 1900-01-01 00:00:00.0
long_name :   time
calendar :    gregorian

short z ( time, latitude, longitude )
scale_factor : 0.04490712217508736
add_offset :  15134.60254643891
_FillValue :  -32767
missing_value : -32767
units :      m**2 s**-2
long_name :   Geopotential
standard_name : geopotential
```

3. 大小: (124\*81\*81), 分别为 time (1043136~1043874), latitude (北纬 15° ~35° ), longitude (东经 95° ~115° ) [ps: 数据实际的储存方式应该 是从左上角开始, 故纬度 (在北半球) (不知道南半球如何, 有数据的话可以试一试) 看起来是倒叙, 从高到低]

同一个数据文件中存储顺序是一致的。所以从 lat 的值排列顺序就可以知道南半球的情况。

```

variable: u
type: short
total Size: 1627128 bytes
      813564 values
Number of Dimensions: 3
Dimensions and sizes: [time | 124] x [latitude | 81] x [longitude | 81]
Coordinates:
  time: [1043136..1043874]
  latitude: [35..15]
  longitude: [95..115]
Number of Attributes: 7
  scale_factor : 0.000523858003941001
  add_offset : -0.691836025681658
  _FillValue : -32767
  missing_value : -32767
  units : m s**-1
  long_name : U component of wind
  standard_name : eastward wind

```

5. 标准下标：按数据实际维数大小；地球坐标：根据实际经纬度（需要用花括号）。这里注意的一点：使用 `nc1_filedump` 或 `print()` 文件的话经纬度看不到具体范围，还需要获取变量后使用 `printVarSummary` 函数。
7. 使用 `res@mpMinlatF` 等四个属性选定范围，对于地球坐标相对方便，标准坐标需要进行换算（根据数据大小和经纬度，且对于纬度来说起始是高纬度）

#### 四、实验小结（本次实验收获的经验、教训、感受等）：

正好在上课前一周，我在使用 `nc1` 做大气物理学的作业（正好用的也是 `era` 的数据 `hhh`），自己摸索学习的大部分知识正好和本次实验重叠，故做起来比较容易。对 `nc1` 大概的感悟就是分为三部分，读取数据、处理数据和 `res` 设置（比较重要）。像百度、气象家园、CSDN 等要重视并常用的（因为我的一个舍友特别喜欢上 `b` 站搜，`b` 站对于这种老软件来说优势并不大）。

做实验内容当时的经验总结有几点：

1. `Ncl_filedump` 是需要更新 `csh` (来自 CSDN)；
2. 使用 `nc1_filedump` 或 `print()` 直接看文件信息的话经纬度看不到具体范围，还需要获取具体变量后使用 `printVarSummary` 函数。这导致当时画图找不到对应经纬度比较难受，不过后来百度的时候找到了 `VarSummary` 函数才解决问题。
3. 数据实际的储存方式直观来说应该是从左上角开始，故纬度（在北半球）（不知道南半球如何，有数据的话可以试一试）看起来是倒叙，从高到低，而使用标准坐标还需要划定范围（匹配经纬度）时 1:10 就不是从  $15^\circ$  开始加，而是从  $35^\circ$  朝下减，不然地图和数据就不匹配。（又想起上周使用 `python` 大创画图也是这个问题，因为 `asc` 头文件里说的是 `left70` 和 `bottom10`，且经度是从 70 开始加，我便以为纬度也是从 10 开始，读取数据也是如此，导致读取出的数据不匹配，我开始以为是地图经纬度的问题，不断尝试才意识到，今天使用 `nc1` 发现也是如此）
4. 当时安装 `nc1` 即是，直接下载的 `anaconda` 文件名很长且复杂，很多同学是一字不漏的全输，经常导致输入错误，可以先改文件名简单一些，再安装会省事许多，这在读取文件的时候也是如此，做测试时完全可以 `copy` 一份数据出来并改名，这样不会对原数据有影响，当完全做好后再把文件名更改回原文件或添加注释即可。
5. 大部分同学的计算机基础（编程思维？）还需要加强，比如本次实验中

取出 lat(1:b)  
就可以知道纬  
度变化的具体  
情况了。

```
var=file->var_in_file
```

不止一位同学来问我，他们写的是 `var=file->var_in_u` 类似的，我觉得可以在方法中多注释清楚一点，不过到后面应该也就好了。

嗯，我以为上课讲的够清楚了，看来还是不清楚，  
让同学们在课下做中学吧。