

❑ GrADS (注意方框內為可調整的文字或參數)

■ 開啟檔案

開啟 ctl 檔 open filename.ctl

開啟網頁或 nc 檔 sdfopen filename.nc

■ 作畫邊框

set parea 1 10 1 7.5

左上分割畫面

set vpage 0 5.5 4.25 8.5

set parea 1 10 1 7.5

右上分割畫面

set vpage 5.5 11 4.25 8.5

set parea 1 10 1 7.5

左下分割畫面

set vpage 0 5.5 0 4.25

set parea 1 10 1 7.5

右下分割畫面

set vpage 5.5 11 0 4.25

set parea 1 10 1 7.5

■ 戳記控制

控制繪製 GrADS logo 戳記 set grads on/off

控制繪製海岸線 set mpdraw on/off

控制繪製輸出時間 set timelab on/off

■ 地圖投影

麥卡托投影 set mproj latlon

北極極座標投影 (範圍必須在北半球) set mproj nps

南極極座標投影 (範圍必須在南半球) set mproj sps

藍伯特錐狀投影 (範圍必須在某半球內) set mproj lambert

羅賓森投影 (經度必須用-180~180 設定) set mproj robinson

關閉地圖顯示 set mproj off

■ 設定座標範圍及刻度

經度範圍 set lon , set lon , set x , set x

緯度範圍 set lat , set lat , set y , set y

高度範圍 set lev , set lev , set z

時間範圍 set t , set t , set time

縱軸範圍 set vrange

控制置換 xy 軸 set xyrev

設定等距 x 座標刻度 set xlint

設定等距 y 座標刻度 set ylint

設定不等距 x 座標刻度 set xlevs

設定不等距 y 座標刻度 set ylevs

■ 繪圖種類

等值線 `set gxout contour`

色塊等值線 `set gxout shaded`

向量 `set gxout vector`

流線圖 `set gxout stream`

填色網格 `set gxout grfill`

一維圖形 `set gxout line`

■ 圖形設定

繪圖顏色 `set ccolor` `1`

線條樣式 `set cstyle` `1`

線條粗細 `set cthick` `1`

資料點樣式 `set cmark` `1`

等值線最小值 `set cmin` `1`

等值線最大值 `set cmax` `1`


等值線固定間距 `set cint` `1`

等值線特定間距 `set clevs` `1 10 100`

流場線條密度 `set strmden` `1`

單位風標大小 `set arrscl` `0.3 15`

風標間隔及用顏色代表風速 `d skip(u,2); skip(v,2); mag(u,v)`

0	background		0	0	0 (black by default)
1	foreground		255	255	255 (white by default)
2	red		250	60	60
3	green		0	220	0
4	dark blue		30	60	255
5	light blue		0	200	200
6	magenta		240	0	130
7	yellow		230	220	50
8	orange		240	130	40
9	purple		160	0	200
10	yellow/green		160	230	50
11	medium blue		0	160	255
12	dark yellow		230	175	45
13	aqua		0	210	140
14	dark purple		130	0	220
15	gray		170	170	170

■ 繪製文字

字體大小 `set strsiz 1`

設定文字顏色、對齊中心、粗細、旋轉 `set string 1 c 10 0`

將文字繪製於特定座標 `draw string 1 1 content`

圖上方加標題文字 `draw title content`

橫軸說明文字 `draw xlab content`

縱軸說明文字 `draw ylab content`

■ 儲存及關閉檔案

儲存檔案及控制類型 `printim filename.png black/white x1280 y768`

關閉檔案 `close 1`

■ 消除指令及設定適用類型

	clear	undefine	reset	reinit
清除x window畫面	○		○	○
清除自訂變數		○		○
時間空間範圍還原到預設值			○	○
繪圖設定還原到預設值			○	○
關閉已開啟的檔案				○

set gxout	contour	shaded	vector	stream	line
set ccolor 線條顏色	○		○	○	○
set cthick 線條粗細	○		○	○	○
set cstyle 線條格式	○				○
set cmin, set cmx 等值線範圍	○	○	○	○	
set cint 等值線間距（等距）	○	○	○	○	
set clevs 等值線間距（不等距）	○	○	○	○	
set strmden 流線密度				○	
set mproj 地圖投影	○	○	○	○	○

■ 平均

對經度平均 set lat 0 , ave (var, lat=-90, lat=90, -b)

對緯度平均 set lon 0 , ave (var, lon=0, lon=360, -b)

對時間平均 set t 1 , ave (var, t=1, t=12)

對區間時間平均 set t 1 , ave (var, t=1, t=12, 3)

對經度緯度平均 set lon 0 , set lat 0 , aave (var, lon=0, lon=360, lat=-90, lat=90)

對經度緯度時間平均 ave (aave (var, lon=0, lon=360, lat=-90, lat=90), t=1, t=12)

■ mask 指令

maskout (var, mask) 在 mask 條件中數值 ≥ 0 的網格位置，其值與 var 變數相同；而數值 < 0 的網格位置，其值為 missing value。

const (var, n) 在 var 為有效值的網格，填入常數 n。

const (var, n, -u) 在 var 為無效值的網格填入常數 n。

const (var, n, -a) 將所有網格都填入常數 n。

■ 氣候值及距平值 (anomaly)

以溫度年循環為例 (一個月一筆資料，共十年)：

```
set t 1 12
define aveTa = ave (Ta, t=0, t=120, 12)
define seasonalTa = aveTa
modify seasonalTa seasonal
set t 1 120
define anomalyTa = Ta - seasonalTa
```

■ 輸出至二進位檔

```
set gxout fwrite  
set fwrite filename.dat  
d var  
disable fwrite  
set gxout contour
```

■ ctl 檔

```
DSET ^ex1.dat  
TITLE 2010 monthly mean at 850hPa  
UNDEF 99999.  
XDEF 144 LINEAR 0 2.5  
YDEF 73 LINEAR -90 2.5  
ZDEF 1 levels 850  
TDEF 12 LINEAR 01JAN2010 1mo  
VARS 6  
Ta 1 99 air temperature (° C)  
u 1 99 u-wind (east-west) (m/s)  
v 1 99 v-wind (north-south) (m/s)  
rh 1 99 relative humidity (%)  
qv 1 99 specific humidity (%)  
prcp 1 99 precipitation (mm)  
ENDVARS
```

請注意第一行不要打成 DEST。

維度宣告區：維度 / 資料格數 / 線性分隔 / 起始點 / 間格大小。

變數宣告區：變數 / 在高度上的層數 / 二進位檔為 99 / 變數描述。

□ Fortran

■ 編譯與執行

編譯檔案：f95 `-O1` filename.f95 (加入 -O1 ~ -O3 可增加運算速度)

f95 `-O1` `filename`.f95 -o `filename`.exe

f95 -O1 filename.f95 filename_sub.f95 -o filename.exe (編譯主副程式)

更改檔名：mv a.out `filename`.exe

執行檔案：./filename.exe

■ Fortran 基本規則

每行！之後的文字會被編譯為註解。

每行最多不可超過 132 個字元，一行的最後出現&符號，代表要與下行連接。

■ 變數與常數

變數與常數名字最多 31 個字元，只能有字母、數字、底線，首字須為字母。

變數可在宣告區或執行區被賦值，且其值在執行區可隨時更動。

常數在宣告區就要被賦值 (設為 `PARAMETER`)，其值之後不可更動。

■ 整數與實數

整數無小數點，小數點以下無條件捨去，實數若用科學記號表示，指數需是整數。

`INT(x)`：實數轉成整數，小數點下無條件捨去、`NINT(x)`：小數點下四捨五入。

`REAL(x)`：整數轉換成實數。

■ 字串

字串變數宣告時需設定變數要儲存的字數 (`len=?`)，文字部分要用引號夾住。

字串變數被賦值的內容比宣告時的長度短，不足的部分會自動補上空格。

`TRIM()` 可以去除字串尾端的空格。

雙斜線 `//` 可以連接兩字串內容或變數，例：`'ersstv5_' // TRIM (str_i) // '.txt'`。

■ 代數運算

等號左邊只能有一個變數；右邊可以是數字或代數，運算符號要用括號分開。

當計算的精確度到小數點以下，最好將數字及變數都設為實數，而非整數。

總和：`SUM (x)` (數值)

平方根：`SQRT (x)` (n 次方根要用 `x ** (1. / n.)`求得)

絕對值：`ABS (x)`

指數與對數：指數函數 `EXP (x)`、自然對數 `LOG (x)`、以十為底的對數 `LOG10 (x)`

三角函數：`SIN (x)`、`COS (x)`、`TAN (x)` (弧度制)

反三角函數：`ASIN (x)`、`ACOS (x)`、`ATAN (x)` (弧度制)

π ：`4. * ATAN (1.)`

餘數：`MOD (x, y)`

較大值：`MAX (x, y)`

較小值：`MIN (x, y)`

■ 格式碼

字串變數：A / 總字元數。若字數不足前方用空格補滿。

科學記號 ($a \times 10^n$, $0.1 \leq |a| < 1$) : E / 總字元數 / . / 小數點後位數 / 指數位數。

科學記號 ($a \times 10^n$, $1 \leq |a| < 10$) : ES / 總字元數 / . / 小數點後位數 / 指數位數。

浮點數：F / 總字元數 / . / 小數點後位數。

整數：I / 總字元數 / . / 至少要輸出的位數。若字數不足前方用空格補滿。

空白：總空白數 / x。

■ 陣列

Fortran 讀取的順序是先直行再橫列(Column major order)。

宣告陣列大小時，可以設定行列的範圍，例：REAL, DIMENSION(3 : 7, 4 : 5) :: A。

在執行區設定陣列大小時，可以設定行列的範圍及間格，例：A(3 : 7 : 2, 5 : 4 : -1)。

重設行列範圍：RESHAPE (A, (/2, 5/))

總和：SUM (A, dimension, mask)，將陣列元素的值加總，可設定只對特定的維度作

運算 (維度為整數)，以及增加篩選條件。

記數：COUNT (mask, dimension)，計算被篩選出的元素總數。

最大值：MAXVAL(array, dimension, mask)，找出陣列元素中的最大的值。

最小值：MINVAL(array, dimension, mask)，找出陣列元素中的最小的值。

最大值位置：MAXLOC(array, dimension, mask)，找出陣列元素中的最大值的位置。

最小值位置：MINLOC(array, dimension, mask)，找出陣列元素中的最小值的位置。

■ 宣告區範例

! Declaration Section

PROGRAM filename

! No implicit variables.

IMPLICIT NONE ! not 'implicit' :)

! Declare variables and constants.

INTEGER :: i, j, k, l, m = 1, n ! m is a variable, for there is no 'PARAMETER.'

INTEGER, PARAMETER :: k = 1

INTEGER, DIMENSION (10, 10) :: A

REAL :: x

REAL, PARAMETER :: y = 1.0

REAL, DIMENSION (10, 10) :: B

CHARACTER (len = 10) :: str1

CHARACTER (len = 10), PARAMETER :: str2 = ' character ' ! The blank is involved.

CHARACTER (len = 10), DIMENSION (10) :: C

LOGICAL :: D = (y > 0.0)

■ 終止區範例

! Termination Section

END PROGRAM filename

■ 常用常數與公式範例 (注意常數數值和變數單位可能有變)

$C_p = 1004.0$

$L_v = 2500000.0$

$P_0 = 1000.0$

$R_d = 287.0$

$R_v = 461.5$

$T_0 = 273.15$

$RH = Q_v / Q_s * 100$

$T_h = T * (P_0 / P) ** (R_d / C_p)$

$T = T_h * (P / P_0) ** (R_d / C_p)$

$E_s = 6.11 * \exp (L_v / R_v * (1 / T_0 - 1 / T))$

$Q_s = 0.622 * E_s / (P - E_s)$

$Q_c = Q_v - Q_s$

$CWV = CWV + 0.5 * (Q_v(i) + Q_v(i + 1)) * (P(i) - P(i + 1)) / 9.81$

■ 邏輯判斷範例

! The program below will do the commands if it meets the logical expression.

i = 1

IF ((i > 1) .AND. (j >= 1)) THEN

 x = 0.0

ELSEIF ((i /= 1) .OR. (k == 1)) THEN

 x = 1.0

ELSEIF (.NOT. (i < 1)) THEN

 x = 2.0

ELSE

 x = 3.0

END IF

■ 迴圈範例

! The program below will repeat n times till END DO is declared.

! The third parameter is the interval, and the default is 1.

DO i = 1, n, 1

! The order of (i, j) would be (1, 1), (1, 2), ... , (1, n), (2, 1), ... , (n, n).

 DO j = 1, n

! Skip the following commands till END DO and do the next loop.

 CYCLE

 END DO

! The loop will keep repeating until it doesn't meet the logical expression.

DO WHILE (n > 0)

 x = i

! Skip the whole loop and do the commands behind END DO.

 EXIT

 END DO

END DO

■ 文字檔案讀寫範例

! Open the file input.txt and number it 10.

```
OPEN (10, FILE = 'input.txt', FORM = 'FORMATTED', STATUS = 'UNKNOWN')
```

! Skip the header.

```
READ (10, *)
```

! Read data into array A, B and C from file 10.

```
DO i = 1, n
```

```
    READ (10, *) A(i), B(i), C(i)
```

```
END DO
```

! Open the file output.txt and number it 20.

```
OPEN (20, FILE = 'output.txt', FORM = 'FORMATTED', STATUS = 'UNKNOWN')
```

! Write the header into file 20.

```
WRITE (20, *) 'It is a header.'
```

! WRITE data into file 20 from array A, B and C.

```
DO i = 1, n
```

```
    WRITE (20, *) A(i), B(i), C(i)
```

```
END DO
```

! Close file 10 and file 20.

```
CLOSE (10)
```

```
CLOSE (20)
```

■ 格式化讀寫範例

! Open the file input.txt and number it 10.

```
OPEN (10, FILE = 'input.txt', FORM = 'FORMATTED', STATUS = 'UNKNOWN')
```

! Skip the header.

```
READ (10, *)
```

! Read data into array A, B and C from file 10 in format.

```
DO i = 1, n
```

```
    READ (10, 100) A(i), B(i), C(i)
```

```
    100 FORMAT (I3.1, 2x, F5.2, 2x, A8)
```

```
END DO
```

! Open the file output.txt and number it 20.

```
OPEN (20, FILE = 'output.txt', FORM = 'FORMATTED', STATUS = 'UNKNOWN')
```

! Write the header into file 20 in format.

```
WRITE (20, 200) 'It is a header.'
```

```
200 FORMAT (A15)
```

! Write data into file 20 from array A, B and C in format.

```
DO i = 1, n
```

```
    WRITE (20, 201) A(i), B(i), C(i)
```

```
    201 FORMAT (I3.1, 2x, F5.2, 2x, A8)
```

```
END DO
```

! Close file 10 and file 20.

```
CLOSE (10)
```

```
CLOSE (20)
```

■ 二進位檔案讀寫範例

! Open the file input.txt and number it 10.

OPEN (10, FILE = 'input.txt', FORM = 'FORMATTED', STATUS = 'UNKNOWN')

! Skip the header.

READ (10, *)

! Read data into array A from file 10.

DO i = 1, n

 READ (10, *) A(i)

END DO

! Inquire the record length of array A in l.

INQUIRE (IOLENGTH = l) (A(i), i = 1, n)

! Open the file output.dat and number it 20.

OPEN (20, FILE = 'output.dat', FORM = 'UNFORMATTED', STATUS = 'UNKNOWN',
ACCESS = 'DIRECT', RECL = l)

! Write data into section 1 of file 20 from array A in opposite order.

WRITE (20, REC = 1) (A(i), i = n, 1, -1)

! Close file 10 and file 20.

CLOSE (10)

CLOSE (20)

二進位檔案輸出必須用隱式迴圈，用顯式迴圈會導致前一次的輸出被覆蓋。

■ 由大到小排序範例

! Sort the data.

n = 10

DO i = 1, n-1

 k = i

 max = A (i)

! Find the maximum value in A (i) to A (n) and store it temporarily in the variable max.

 DO j = i+1, n

 IF (A (j) > max) THEN

 k = j

 max = A (k)

 ENDIF

 ENDDO

! Swap the maximum value with A (i).

A (k) = A (j)

A (j) = max

■ 由小到大排序範例

! Sort the data.

n = 10

DO i = 1, n-1

 k = i

 min = A (i)

! Find the minimum value in A (i) to A (n) and store it temporarily in the variable min.

 DO j = i+1, n

 IF (A (j) < min) THEN

 k = j

 min = A (k)

 ENDIF

 ENDDO

! Swap the minimum value with A (i).

A (k) = A (j)

A (j) = min

如果將排序演算法中的 IF 判斷式從原本的 < (或 >) 改為 ≤ (或 ≥)，若有數值相同

的資料，它們的順位可能會變得不一樣。

■ 副程式範例——梯形積分

! SUBROUTINE

! Declaration Section

SUBROUTINE integration (P, Q, R, m)

! No implicit variables.

IMPLICIT NONE

! Declare variables and constants

INTEGER :: i

INTEGER, INTENT (in) :: m

REAL :: g = 9.8

REAL, DIMENSION (m), INTENT (in) :: P, Q

REAL, INTENT (out) :: R

! Execution Section

! Do loop to calculate the integral of R

Do i = 1, m - 1

 R = R + 0.5 * (Q (i) + Q (i + 1)) * (P (i) - P (i + 1))

END DO

R = -R * 100. / g

! Termination Section

RETURN

END SUBROUTINE integration

■ 副程式範例——灰階轉換

! SUBROUTINE

! Declaration Section

SUBROUTINE grayscale (OutIm, InIm, x, y, z)

! No implicit variables.

IMPLICIT NONE

! Declare variables and constants

INTEGER :: i, j

INTEGER, INTENT (in) :: x, y, z

INTEGER, DIMENSION (x, y, z), INTENT (in) :: InIm

REAL, DIMENSION (x, y), INTENT(out) :: OutIm

! Execution Section

! Do loop to calculate the gray scale.

DO i = 1, x

DO j = 1, y

OutIm (i, j) = (0.299 * InIm (i, j, 1) + 0.587 * InIm (i, j, 2) + 0.114 * InIm (i, j, 3)) / 255

END DO

END DO

! Termination Section

RETURN

END SUBROUTINE grayscale

■ 副程式範例——溫度遞減率

! SUBROUTINE

! Declaration Section

SUBROUTINE lapse_rate (Zt, Zb, Tt, Tb, G)

! No implicit variables.

IMPLICIT NONE

! Declare variables and constants

REAL, INTENT (in) :: Zt, Zb, Tt, Tb

REAL :: dZ, dT

REAL, INTENT (out) :: G

! Execution Section

! Calculate the value of dT, dZ, and G.

dT = Tt - Tb

dZ = Zt - Zb

G = -1*(dT / dZ)

! If the height, temperature, or thickness of height ≤ 0 , G is defined as missing value.

IF ((Zt <= 0.) .OR. (Zb <= 0.) .OR. (Tt <= 0.) .OR. (Tb <= 0.) .OR. (Zt - Zb <= 0.)) THEN

 G = +1.0E+30

END IF

! Termination Section

RETURN

END SUBROUTINE lapse_rate