PS6 Report

吴文浩 12032567

一、 PS6_1

1.1：编写矩阵乘法子程序：由于题目中未强调只能使用 for 循环即加减乘除实现乘法功能，此处直接使用了 Fortran 自带矩阵乘法函数 matmul()实现乘法功能，由于此处 M,N 均为方阵，在代码中传入参数有 x,y 为传入的矩阵，n 为方阵阶数；传出参数为 c

代码截图：

```
subroutine Matrix_multip(x, y, c, n)
implicit none
  integer                    :: n, i, j
  real(4), dimension(n, n) :: x, y
  real(8), dimension(n, n) :: c
  c = matmul(x,y)
  return
end
```

1.2：编写主程序实现读取 M/N.dat：创建两个可分配数组，由于已知两个数据集均为阶数为 3 的方阵，理论上需要获取文件的行数与列数，此处简化直接传入方阵阶数进行分配，读取数据集信息

代码截图：

```
program Main
implicit none

integer                                 :: u, v, n, i, j
real(4), dimension(:,:), allocatable :: a, b
real(8), dimension(:,:), allocatable :: c


! File unit
u = 50
v = 80

! Open the file
open(unit=u, file='M.dat', status='old')
open(unit=v, file='N.dat',status = 'old')

! The first line of the file has the number of values for arrays a and b
! 已提前知道M/N.dat均为方阵，且维数相同，此处代码简化只获取其中一个的行数即可
n = 3

! Allocate the arrays
allocate(a(n,n), b(n,n),c(n,n))

j = 1
do i = 1,n
  read(u, *) a(i,j), a(i,j+1), a(i,j+2)
  read(v, *) b(i,j), b(i,j+1), b(i,j+2)
enddo

! Close the file
close(v)
close(u)
```

1.3：调用子程序实现矩阵乘法：考直接调用子程序，传入所需参数，即可获得传出参数即结果矩阵 C，将该矩阵写入新文件 MN.dat。

代码分析：

```
call Matrix_multip(a,b,c,n)

! Write the values to a new file
open(unit=u, file='MN.dat', status='replace')

do i = 1,n
  write(u,'(f8.1,f8.1,f8.1)') c(i,j), c(i,j+1), c(i,j+2)
enddo
close(u)

! Display the values
do i = 1,n
  write(*,*) "The Answer:"
  write(*,*) "Line ", i, " : ", c(i,j), c(i,j+1), c(i,j+2)
enddo


! Deallocate the arrays
deallocate( a, b, c )

End Program Main
```

结果截图：

```
[ese-wuwh@login03 PS6_1]$ ./Main.x
 The Answer:
 Line           1  :    166.54460144042969        540.46643066406250        256.62811279296875
 The Answer:
 Line           2  :    146.99084472656250        431.39477539062500        208.19314575195312
 The Answer:
 Line           3  :    116.35884094238281        510.89779663085938        198.89994812011719
[ese-wuwh@login03 PS6_1]$ cat MN.dat
   166.5   540.5   256.6
   147.0   431.4   208.2
   116.4   510.9   198.9
```

二、 PS4_2

2.1：模块 Declination_angle: 包含两个函数，计算当年的天数函数 DaysInYear(year, mon, day)；计算 declination angle 函数 DecAngle(year, n)，尽管手册上没有提及闰年时候的倾角公式，但依然认为应当添加对于闰年的天数判断在其中

代码截图：

```
module Declination_angle

implicit none

   real, parameter :: pi = 3.1415926536

contains
   ! 计算当前日期在当年的天数
   Integer Function DaysInYear(year, mon, day)
      Integer :: year, mon , day
      Integer :: DaysInMonth(12) = [31,28,31,30,31,30,31,31,30,31,30,31]
      if ( ( (MOD(year,4)==0).and.(MOD(year,100)/=0) ) .or. (mod(year,400)==0) ) then
         DaysInMonth(2) = 29
      else
         DaysInMonth(2) = 28
      end if
      DaysInYear = sum( DaysInMonth(:mon-1) ) + day
   End Function DaysInYear

   ! 计算declination angle
   real(8) Function DecAngle(year, n)
      Integer :: year, n
      real    :: temp
      ! 转化为弧度制
      if ( ( (MOD(year,4)==0).and.(MOD(year,100)/=0) ) .or. (mod(year,400)==0) ) then
         temp = (n + 284) * 360 * pi / (366 * 180)
      else
         temp = (n + 284) * 360 * pi / (365 * 180)
      end if
      DecAngle = 23.45 *  sin(temp)
   end Function DecAngle

end module Declination_angle
```

2.2：模块 AST：包含一个函数 ASTIME(Long, direct, DMT, year, hour, min, day)，实现包括计算 ET/LSTM/H，根据东经和西经分别计算 AST 时间，将 AST 时间标准化功能，尽管文中对于 D 的公式没有提及闰年，但依然认为应当添加闰年天数的判断。通过查资料也可以知道，均时差 ET 其实会随着年的变化而变化且四年一闰完成一次重置

代码截图：

```
module AST

implicit none

   real, parameter :: pi = 3.1415926536

contains
   ! 计算当前时刻的Local Solar Time
   Function ASTIME(Long, direct, DMT, year, hour, min, day)
      logical :: direct
      Integer :: ASTIME(2)
      Integer :: DMT, year, hour, min, day, LSTM, temp
      real(8) :: D, ET, Long, tt
      ! 根据是否是闰年修改公式
      if ( ( (MOD(year,4)==0).and.(MOD(year,100)/=0) ) .or. (mod(year,400)==0) ) then
         D =360 * (dble(day) - 81) / 366
      else
         D =360 * (dble(day) - 81) / 365
      end if
      ET = 9.87 * sin(2*D*pi/180) - 7.53 * cos(D*pi/180) - 1.5 * sin(D*pi/180)
      if(direct .eqv. .true.) then
    Long = -Long
      else
    Long = Long
      endif
      LSTM = 15 * DMT
      temp = 4 * (Long - LSTM) + ET
      ASTIME(1) = hour + INT(temp/60)
      ASTIME(2) = min + MOD(temp, 60)

      if(ASTIME(2) > 60) then
    ASTIME(1) = ASTIME(1) + 1
    ASTIME(2) = ASTIME(2) - 60
      elseif(ASTIME(2) < 0) then
    ASTIME(1) = ASTIME(1) - 1
    ASTIME(2) = 60 + ASTIME(2)
      else
    ASTIME(1) = ASTIME(1)
    ASTIME(2) = ASTIME(2)
      endif
      if(ASTIME(1) >= 24)then
    ASTIME(1) = ASTIME(1)  - 24
      else
    ASTIME(1) = ASTIME(1)
      endif
      print*, "D = ", D
      print*,  "ET = ", ET
      print*, "LSTM = ", LSTM
   End Function ASTIME
end module AST
```

2.3：主函数 Cal_SZA.f90：用户输入包括（年份，月份，日期，时钟，分钟，方向，经度，纬度）信息，通过调用前两个模块，来计算获取当前对应的 Solar Zenith Angle

代码截图：

```
program FunctionTest
use Declination_angle
use AST
implicit none

logical :: direct
Integer :: DMT, year, month, date, hour, min, day
real(8) :: DA, Long, Lat, H, SAA, SZA
Integer :: ApparentST(2)
real    :: p

p = 3.1415926536

write(*,*) 'Please input the year:'
read(*,*) year

write(*,*) 'Please input the month:'
read(*,*) month

write(*,*) 'Please input the date:'
read(*,*) date

write(*,*) 'Please input the hour:'
read(*,*) hour

write(*,*) 'Please input the minute:'
read(*,*) min

write(*,*) 'Please input the Time Zone(West -12 ---- 12 East):'
read(*,*) DMT

write(*,*) 'In the western longitudes? (please input ".true." or ".false.")'
read(*,*) direct

write(*,*) 'Please input the Longitude:'
read(*,*) Long

write(*,*) 'Please input the Latitude:'
read(*,*) Lat


day = DaysInYear(year, month, date)
write(*,*) 'The day in this year is: ', day

DA = DecAngle(year, day)
write(*,*) 'The declination angle is: ', DA, 'Deg'

ApparentST = ASTIME(Long, direct, DMT, year, hour, min, day)
write(*,*) 'The apparent solar time(AST) is: ', ApparentST(1), ':', ApparentST(2)


H = ((60 * dble(ApparentST(1)) + dble(ApparentST(2))) - 720)/4
write(*,*) 'The hour angle(H) is: ', H

SAA = asin(cos(Lat*p/180)*cos(DA*p/180)*cos(H*p/180) + sin(Lat*p/180) * sin(DA*p/180))
SAA = SAA * 180 / p
write(*,*) 'The  altitude angle is: ', SAA

SZA = 90 - SAA
write(*,*) 'The zenith angle(SZA) is: ', SZA


end program FunctionTest
```

2.4：用库文件的方式编译代码并执行：这里采用手册上的例子进行计算，需要指出的是，当前的代码只能实现对于北半球地区的计算，没有添加南半球计算的算子模块

结果截图：

```
[ese-wuwh@login03 PS6_2]$ gfortran -c AST.f90
[ese-wuwh@login03 PS6_2]$ gfortran -c Declination_angle.f90
[ese-wuwh@login03 PS6_2]$ ar rcvf libsolar.a AST.o Declination_angle.o
r - AST.o
r - Declination_angle.o
[ese-wuwh@login03 PS6_2]$ gfortran Cal_SZA.f90 -o Cal_SZA.x -L. -lsolar
[ese-wuwh@login03 PS6_2]$ ./Cal_SZA.x
 Please input the year:
2019
 Please input the month:
7
 Please input the date:
21
 Please input the hour:
8
 Please input the minute:
0
 Please input the Time Zone(West -12 ---- 12 East):
-7
 In the western longitudes? (please input ".true." or ".false.")
.true
 Please input the Longitude:
112
 Please input the Latitude:
33.43
 The day in this year is:          202
 The declination angle is:    20.441514968872070      Deg
 D =    119.34246575342466
 ET =   -6.0498034063149184
 LSTM =         -105
 The apparent solar time(AST) is:           7 :          26
 The hour angle(H) is:   -68.500000000000000
 The  altitude angle is:    28.621089214275695
 The zenith angle(SZA) is:    61.378910785724301
```

2.5：计算深圳的案例

结果截图：

```
[ese-wuwh@login03 PS6_2]$ ./Cal_SZA.x
 Please input the year:
2020
 Please input the month:
12
 Please input the date:
20
 Please input the hour:
14
 Please input the minute:
35
 Please input the Time Zone(West -12 ---- 12 East):
8
 In the western longitudes? (please input ".true." or ".false.")
.false
 Please input the Longitude:
114.062996
 Please input the Latitude:
22.542883
 The day in this year is:          355
 The declination angle is:   -23.442226409912109      Deg
 D =    269.50819672131149
 ET =    1.7340064615222319
 LSTM =          120
 The Local solar time(LST) is:           14 :          13
 The hour angle(H) is:    33.250000000000000
 The  altitude angle is:    33.787928461860545
 The zenith angle(SZA) is:    56.212071538139455
```

与 https://www.pveducation.org/网站结果进行对比，存在的差异主要来源于三个方面

- 我的代码中包含了对于闰年的判断
- 网站支持的经纬度精度
- 代码计算 Hour Angle 时候的计算精度