Assignment 06

Assignment 06 is difficult for me. So, I asked Yuan Li for help. He explained the two problems in detail. Then, I finished the homework. Furthermore, I got help from the website (http://blog.sina.com.cn/s/blog_4cf1445d0100x1m1.html).

1. Matrix multiplication

1.1 Write a subroutine Matrix_multip.f90 to do matrix multiplication.

1) set matrix M, N and T, T=M*N.

2) design a loopse to obtain T (Here, the front number represents the column while the latter number represents the row. It is different from my common sense.)

```
subroutine Matrix_multip(M, N, T)
implicit none
real(4), dimension(3,4) :: M
real(4), dimension(3,3) :: N
real(4), dimension(3,4) :: T
integer :: i, j, k
real(4) :: value
do j=1,3
do i=1,4
value=0
do k=1,3
value=value+N(j,k)*M(k,i)
enddo
T(j,i)=value
enddo
enddo
end subroutine Matrix_multip
```

1.2 Write a program Main.f90 to read /work/ese-ouycc/fortran_2/M.dat as the matrix M, and /work/ese-ouycc/fortran_2/N.dat as the matrix N.

```fortran
Program Main

implicit none

integer :: i
real(4), dimension(4) :: a, b, c
real(4), dimension(3,4) :: M
real(4), dimension(3,3) :: N
real(4), dimension(3,4) :: T

open(3, file='M.dat', status='old')

do i=1,4
read(3, *) a(i), b(i), c(i)
M(1,i)=a(i)
M(2,i)=b(i)
M(3,i)=c(i)
enddo
close(3)

do i=1,4
write(*,*) M(:,i)
enddo

open(3, file='N.dat', status='old')
do i=1,3
read(3, *) a(i),b(i), c(i)
N(1,i)=a(i)
N(2,i)=b(i)
N(3,i)=c(i)
enddo
close(3)

do i=1,3
write(*,*) N(:,i)
enddo

call Matrix_multip(M, N, T)
open(3, file='MN.dat', status='replace')
do i=1,4
write(3, '(f8.1,f8.1,f8.1,f8.1)') T(:,i)
enddo
close(3)

End Program Main
```

```
[ese-wangjq@login03 Assignment6]$ gfortran Matrix_multip.f90 Main.f90
[ese-wangjq@login03 Assignment6]$ ./a.out
   9.48898888       15.7995195       9.28895760
   9.28895760       12.9239597       5.86212111     M
   5.86212111       11.2947102       14.0426903
   1.93569267       18.6091709       18.2320099
   7.72341394       14.1155996       1.44496036
   5.55180502       14.8062401       14.0426903     N
  0.596554220       18.5803604       2.26603913
```

*1.3 Call subroutine Matrix_multip() from Main.f90 to compute M*N; write the output to a new file MN.dat, values are in formats of f8.1.*

$$T = \begin{pmatrix} 166.5 & 540.5 & 256.6 \\ 147.0 & 431.4 & 208.2 \\ 116.4 & 510.9 & 198.9 \\ 129.1 & 641.6 & 305.4 \end{pmatrix}$$

```
[ese-wangjq@login03 Assignment6]$ vi Main.f90
[ese-wangjq@login03 Assignment6]$ gfortran Matrix_multip.f90 Main.f90
[ese-wangjq@login03 Assignment6]$ ./a.out
   9.48898888       15.7995195       9.28895760
   9.28895760       12.9239597       5.86212111
   5.86212111       11.2947102       14.0426903
   1.93569267       18.6091709       18.2320099
   7.72341394       14.1155996       1.44496036
   5.55180502       14.8062401       14.0426903
  0.596554220       18.5803604       2.26603913
[ese-wangjq@login03 Assignment6]$ vi MN.dat
```

```
  166.5   540.5   256.6
  147.0   431.4   208.2
  116.4   510.9   198.9
  129.1   641.6   305.4
```

2. Calculate the solar zenith angle

2.1 Write a module Declination_angle to calculate the declination angle on a certain date.

Here, declination angle has been converted to radian for calculations in the latter steps.

$$\delta = 23.45°\sin\left[\frac{N+284}{365}\times 360°\right] \tag{2}$$

```
module Delination_angle
contains
subroutine cal_delination_angle(N, delta)
implicit none
real(4) :: pi=3.1415926536
Integer,intent(in) :: N
real(4),intent(out) :: delta
delta=(23.45/180*pi)*sin((N+284)/365*2*pi)
print *, "delta=", delta
end subroutine cal_delination_angle
end module Delination_angle
```

2.2 Write a module AST to calculate the apparent solar time (AST; or local solar time) in a certain location for a certain date and time.

The unit of AST is minute.

$$AST = LST + (4 \text{ min/ deg})(LSTM - Long) + ET \tag{3}$$

$$LSTM = 15° \times \left(\frac{Long}{15°}\right)_{\text{round to integer}} \tag{4}$$

$$ET = 9.87 \sin(2D) - 7.53 \cos(D) - 1.5 \sin(D)$$
$$\text{where} \quad D = 360° \frac{(N-81)}{365} \tag{5}$$

```
module AST
contains
subroutine cal_AST(Long, N, LST, AST_time)
implicit none
real(4) :: pi=3.1415926536
integer :: LSTM
real(4),intent(in) :: Long
real(4) :: D
real(4) :: ET
integer,intent(in) :: N
real(4),intent(in) :: LST
real(4),intent(out) :: AST_time
LSTM=15*INT(Long/15)
print *, "LSTM=", LSTM
D=2*pi*(N-81)/365
ET=9.87*sin(2*D)-7.53*cos(D)-1.5*sin(D)
print *, "D=", D
print *, "ET=", ET
AST_time=LST+4*(LSTM-Long)+ET
print *, "AST_time=", AST_time
end subroutine cal_AST

end module AST
```

2.3 Write a main program (Cal_SZA.f90) that uses module Declination_angle and AST to print the SZA in a certain location for a certain date and time.

Here, I assume N=100, longitude=100°, LST=360 min.

Radian is need when calculation. But radian should be converted to angle at last. (angle= radian/pi*180)

$$H = \frac{(No.\ of\ \ minutes\ \ past\ \ midnight, AST) - 720\ \text{mins}}{4\ \text{min}/\text{deg}} \tag{6}$$

$$\cos(\theta_z) = \sin(\beta_1) = \cos(L)\cos(\delta)\cos(H) + \sin(L)\sin(\delta) \tag{7}$$

```
program Cal_SZA

use Delination_angle
use AST
implicit none
real(4) :: pi
real(4) :: H
real(4) :: L
real(4) :: zenith_angle, beta_1


integer :: N
real(4) :: delta
real(4) :: Long, LST
real(4) :: AST_time

N=100

call cal_delination_angle(N, delta)
print *, "delta=", delta

Long=100
LST=360

call cal_AST(Long, N, LST, AST_time)
print *, "AST_time=", AST_time

L=60
pi=3.1415926536
H=(AST_time-720)/4
zenith_angle=ACOS(cos(pi*L/180)*cos(delta)*cos(pi*H/180)+sin(pi*L/180)*sin(delta))/pi*180
beta_1=ASIN(cos(pi*L/180)*cos(delta)*cos(pi*H/180)+sin(pi*L/180)*sin(delta))/pi*180
print *, "zenith angle=", zenith_angle
print *, "beta_1=", beta_1

end program Cal_SZA
```

2.4 Create a library (libsolar.a) that contains Declination_angle.o and AST.o. Compile Cal_SZA.f90 using libsolar.a.

1) obtain Declination_angle.o and AST.o

2) Create libsolar.a

```
[ese-wangjq@login03 Assignment6]$ gfortran -c Delination_angle.f90
[ese-wangjq@login03 Assignment6]$ gfortran -c AST.f90
[ese-wangjq@login03 Assignment6]$ ar rcvf libsolar.a Delination_angle.o AST.o
r - Delination_angle.o
r - AST.o
```

3) compile Cal_SZA.f90 using libsolar.a

```
[ese-wangjq@login03 Assignment6]$ gfortran Cal_SZA.f90 -o Cal_SZA_lib.x -L. -lsolar
[ese-wangjq@login03 Assignment6]$ ./Cal_SZA_lib.x
 delta=   7.15607413E-08
 delta=   7.15607413E-08
 LSTM=          90
 D=  0.327069938
 ET=  -1.60705566
 AST_time=   318.392944
 AST_time=   318.392944
 zenith angle=   95.1794128
 beta_1=  -5.17940807
```

```
[ese-wangjq@login03 Assignment6]$ ll
total 647
-rwxr-xr-x 1 ese-wangjq ese-ouycc 13320 Dec 22 19:38 a.out
-rw-r--r-- 1 ese-wangjq ese-ouycc   495 Dec 22 20:09 AST.f90
-rw-r--r-- 1 ese-wangjq ese-ouycc  1047 Dec 19 23:47 ast.mod
-rw-r--r-- 1 ese-wangjq ese-ouycc  3464 Dec 22 20:35 AST.o
-rw-r--r-- 1 ese-wangjq ese-ouycc   667 Dec 22 20:42 Cal_SZA.f90
-rwxr-xr-x 1 ese-wangjq ese-ouycc 13488 Dec 22 20:44 Cal_SZA_lib.x
-rw-r--r-- 1 ese-wangjq ese-ouycc   679 Dec 22 20:47 Cal_SZA_SZ.f90
-rwxr-xr-x 1 ese-wangjq ese-ouycc 13496 Dec 22 20:47 Cal_SZA_SZ.x
-rwxr-xr-x 1 ese-wangjq ese-ouycc 13488 Dec 19 23:49 Cal_SZA.x
-rw-r--r-- 1 ese-wangjq ese-ouycc   303 Dec 22 19:49 Delination_angle.f90
-rw-r--r-- 1 ese-wangjq ese-ouycc   890 Dec 19 23:44 delination_angle.mod
-rw-r--r-- 1 ese-wangjq ese-ouycc  2248 Dec 22 20:34 Delination_angle.o
-rw-r--r-- 1 ese-wangjq ese-ouycc  5974 Dec 22 20:35 libsolar.a
-rw-r--r-- 1 ese-wangjq ese-ouycc   636 Dec 22 19:38 Main.f90
-rw-r--r-- 1 ese-wangjq ese-ouycc   293 Dec 22 19:33 Matrix_multip.f90
-rw-r--r-- 1 ese-wangjq ese-ouycc   114 Dec 19 16:31 M.dat
-rw-r--r-- 1 ese-wangjq ese-ouycc   100 Dec 22 20:23 MN.dat
-rw-r--r-- 1 ese-wangjq ese-ouycc    86 Dec 19 16:37 N.dat
```

2.5 Print the SZA for Shenzhen (22.542883N, 114.062996E) at 14:35 (Beijing time; UTC+8) on 2020-12-20.

Here, N=355, longitude=114.062996, LST=875, latitude=22.542883. The result shows that zenith angle is 36.91° .

```
program Cal_SZA

use Delination_angle
use AST
implicit none
real(4) :: pi
real(4) :: H
real(4) :: L
real(4) :: zenith_angle, beta_1


integer :: N
real(4) :: delta
real(4) :: Long, LST
real(4) :: AST_time

N=355

call cal_delination_angle(N, delta)
print *, "delta=", delta

Long=114.062996
LST=875

call cal_AST(Long, N, LST, AST_time)
print *, "AST_time=", AST_time

L=22.542883
pi=3.1415926536
H=(AST_time-720)/4
zenith_angle=ACOS(cos(pi*L/180)*cos(delta)*cos(pi*H/180)+sin(pi*L/180)*sin(delta))/pi*180
beta_1=ASIN(cos(pi*L/180)*cos(delta)*cos(pi*H/180)+sin(pi*L/180)*sin(delta))/pi*180
print *, "zenith angle=", zenith_angle
print *, "beta_1=", beta_1

end program Cal_SZA
```

```
[ese-wangjq@login03 Assignment6]$ gfortran Cal_SZA_SZ.f90 -o Cal_SZA_SZ.x -L. -lsolar
[ese-wangjq@login03 Assignment6]$ ./Cal_SZA_SZ.x
 delta=   7.15607413E-08
 delta=   7.15607413E-08
 LSTM=          105
 D=   4.71669245
 ET=   1.38263178
 AST_time=   840.130676
 AST_time=   840.130676
 zenith angle=   36.9088898
 beta_1=   53.0911102
```