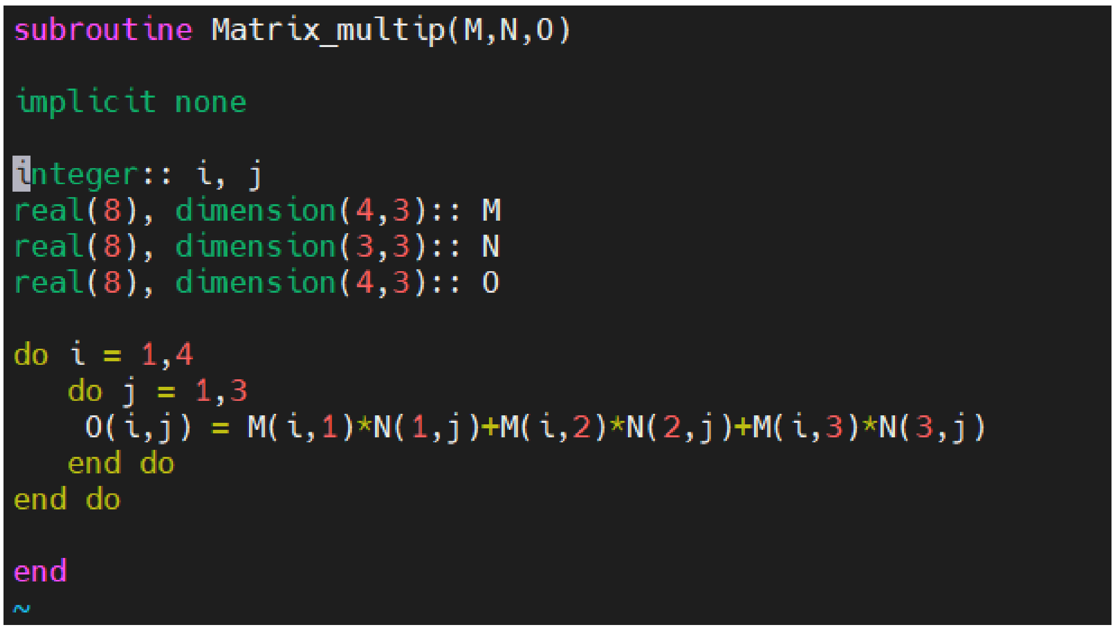
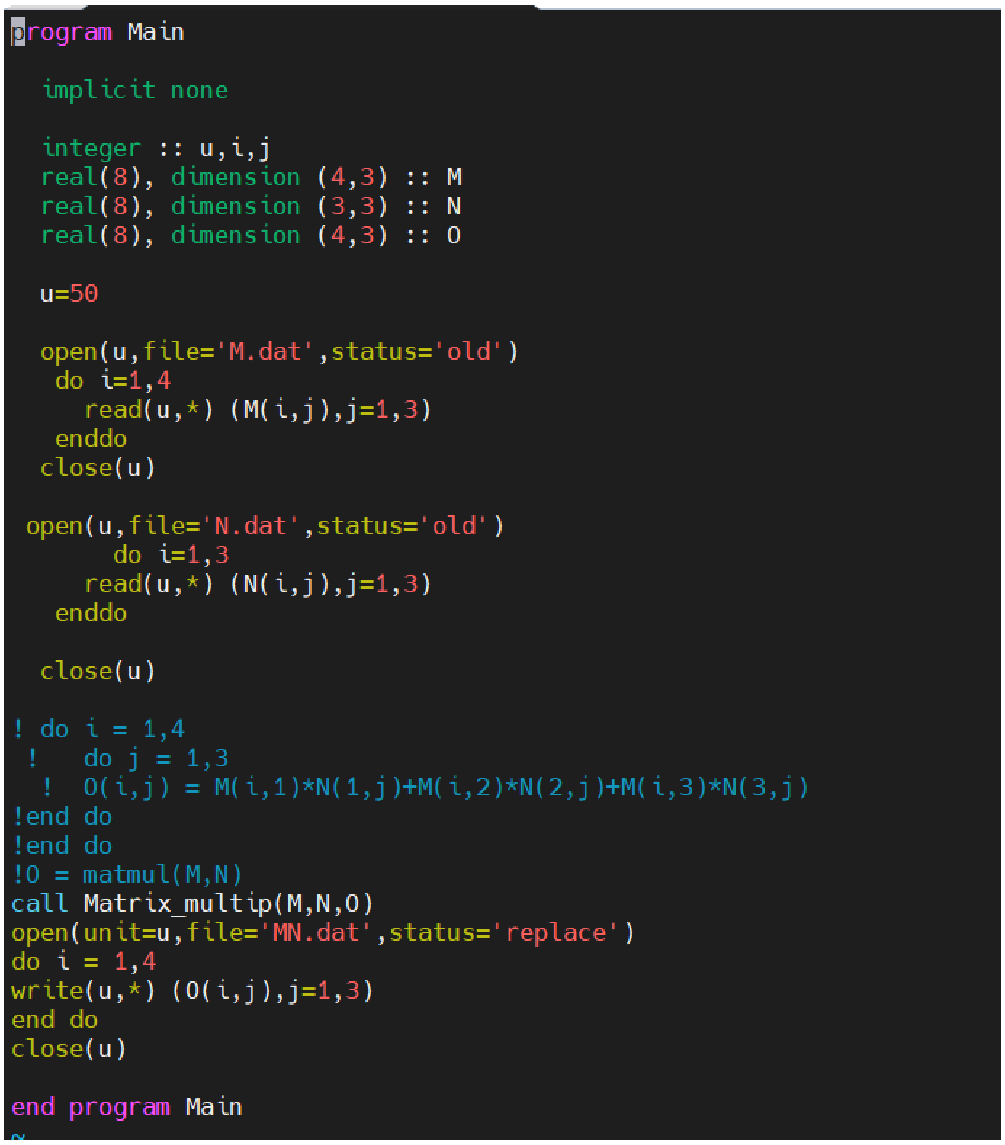
1.1 subroutine



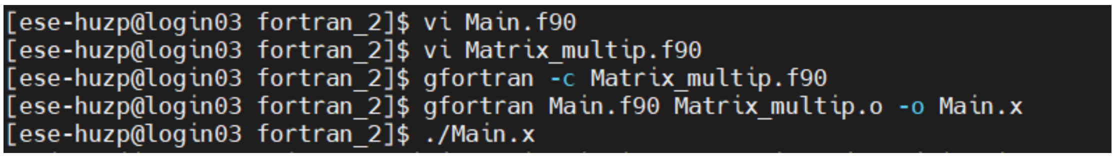
```fortran
subroutine Matrix_multip(M,N,O)

implicit none

integer:: i, j
real(8), dimension(4,3):: M
real(8), dimension(3,3):: N
real(8), dimension(4,3):: O

do i = 1,4
   do j = 1,3
    O(i,j) = M(i,1)*N(1,j)+M(i,2)*N(2,j)+M(i,3)*N(3,j)
   end do
end do

end
~
```

1.2 Main

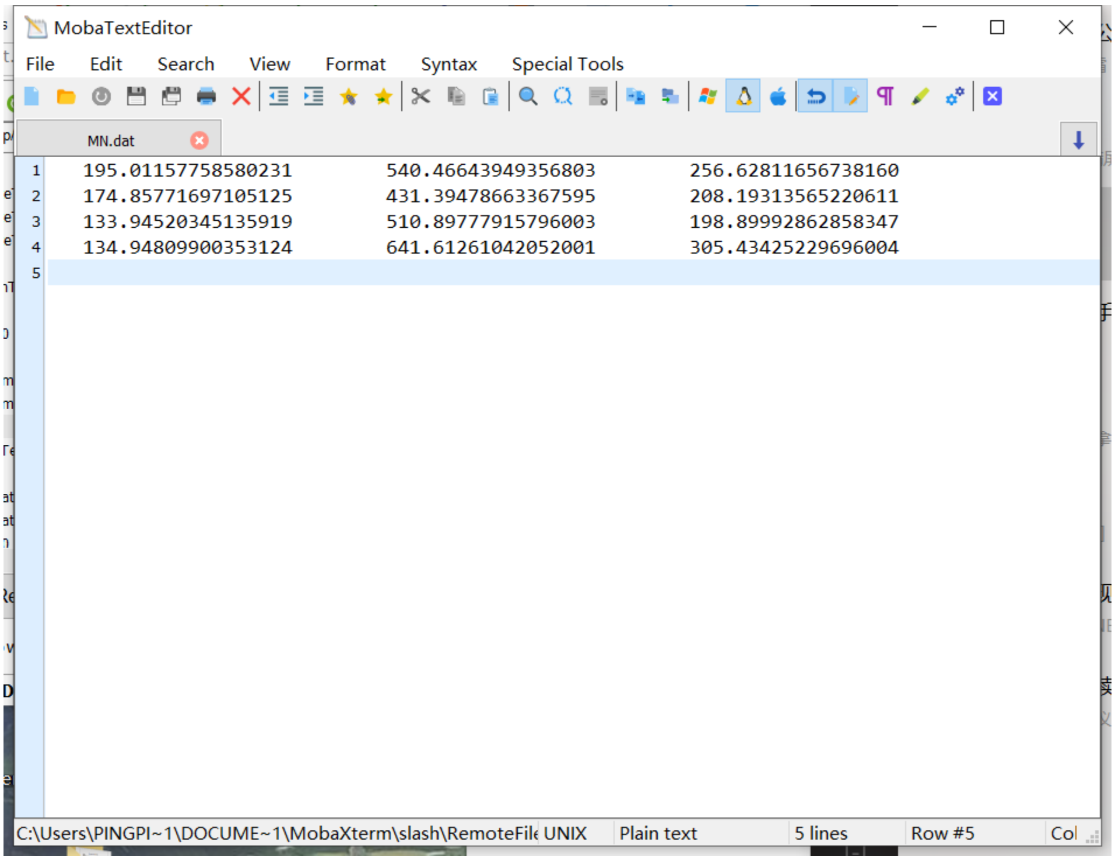


```fortran
program Main

   implicit none

   integer :: u,i,j
   real(8), dimension (4,3) :: M
   real(8), dimension (3,3) :: N
   real(8), dimension (4,3) :: O

   u=50

   open(u,file='M.dat',status='old')
    do i=1,4
      read(u,*) (M(i,j),j=1,3)
    enddo
   close(u)

  open(u,file='N.dat',status='old')
        do i=1,3
      read(u,*) (N(i,j),j=1,3)
    enddo

   close(u)

! do i = 1,4
 !   do j = 1,3
  !  O(i,j) = M(i,1)*N(1,j)+M(i,2)*N(2,j)+M(i,3)*N(3,j)
!end do
!end do
!O = matmul(M,N)
call Matrix_multip(M,N,O)
open(unit=u,file='MN.dat',status='replace')
do i = 1,4
write(u,*) (O(i,j),j=1,3)
end do
close(u)

end program Main
~
```

1.3 编译和运行结果



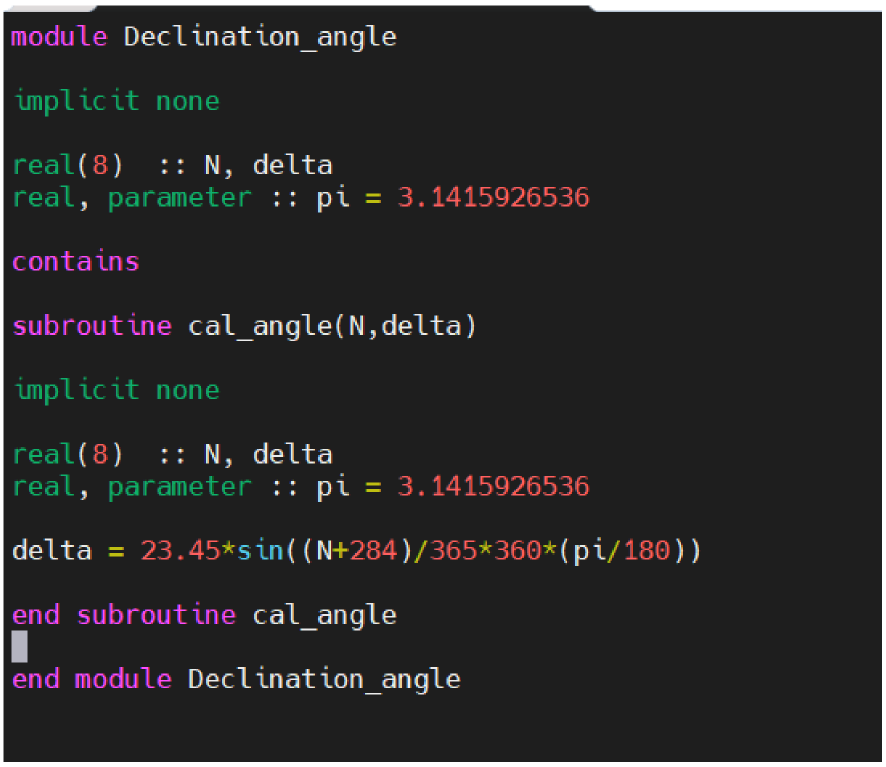
```
[ese-huzp@login03 fortran_2]$ vi Main.f90
[ese-huzp@login03 fortran_2]$ vi Matrix_multip.f90
[ese-huzp@login03 fortran_2]$ gfortran -c Matrix_multip.f90
[ese-huzp@login03 fortran_2]$ gfortran Main.f90 Matrix_multip.o -o Main.x
[ese-huzp@login03 fortran_2]$ ./Main.x
```


MobaTextEditor
File Edit Search View Format Syntax Special Tools
MN.dat

```
195.01157758580231        540.46643949356803        256.62811656738160
174.85771697105125        431.39478663367595        208.19313565220611
133.94520345135919        510.89777915796003        198.89992862858347
134.94809900353124        641.61261042052001        305.43425229696004
```

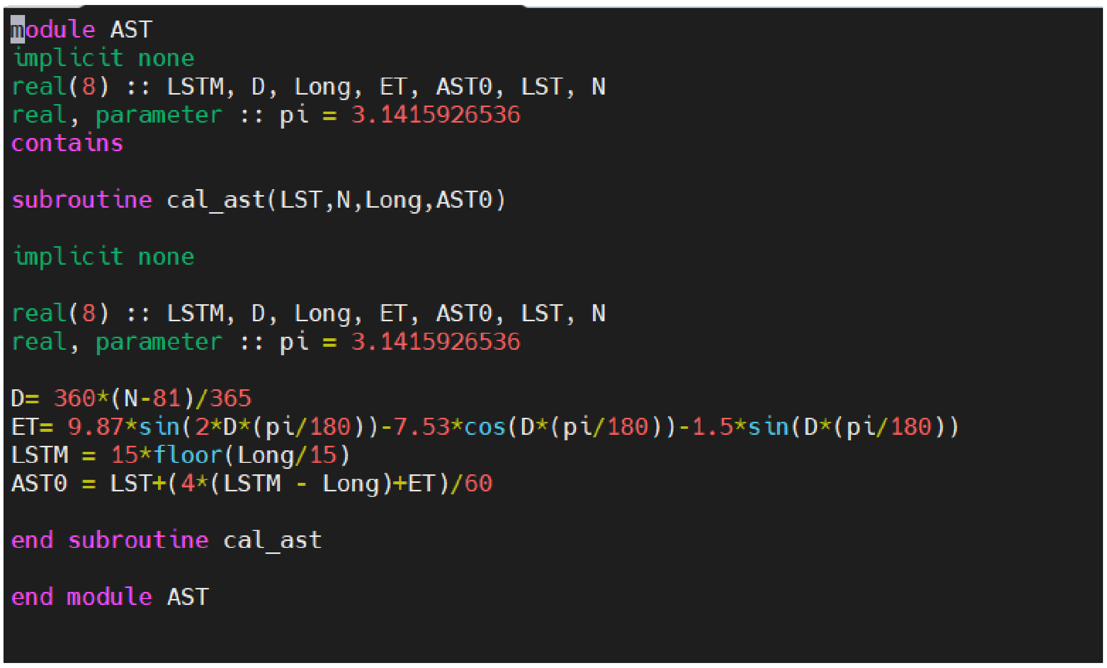
C:\Users\PINGPI~1\DOCUME~1\MobaXterm\slash\RemoteFile UNIX Plain text 5 lines Row #5 Col

2.1



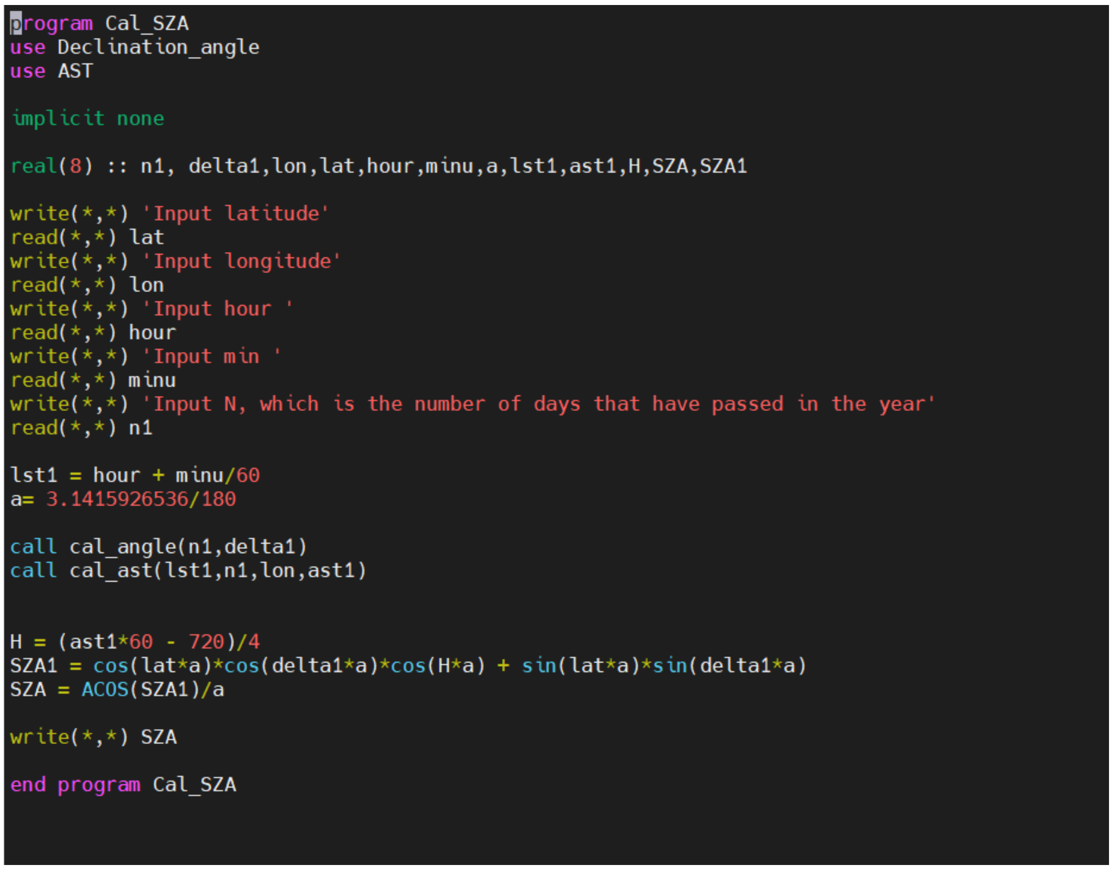
```
module Declination_angle

implicit none

real(8)  :: N, delta
real, parameter :: pi = 3.1415926536

contains

subroutine cal_angle(N,delta)

implicit none

real(8)  :: N, delta
real, parameter :: pi = 3.1415926536

delta = 23.45*sin((N+284)/365*360*(pi/180))

end subroutine cal_angle

end module Declination_angle
```

2.2



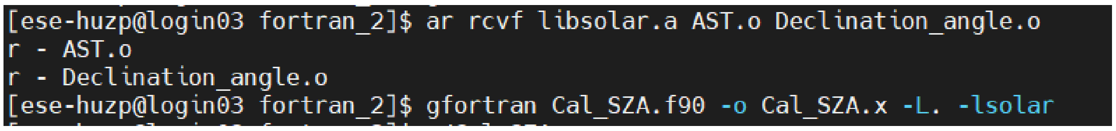
```
module AST
implicit none
real(8) :: LSTM, D, Long, ET, AST0, LST, N
real, parameter :: pi = 3.1415926536
contains

subroutine cal_ast(LST,N,Long,AST0)

implicit none

real(8) :: LSTM, D, Long, ET, AST0, LST, N
real, parameter :: pi = 3.1415926536

D= 360*(N-81)/365
ET= 9.87*sin(2*D*(pi/180))-7.53*cos(D*(pi/180))-1.5*sin(D*(pi/180))
LSTM = 15*floor(Long/15)
AST0 = LST+(4*(LSTM - Long)+ET)/60

end subroutine cal_ast

end module AST
```

2.3



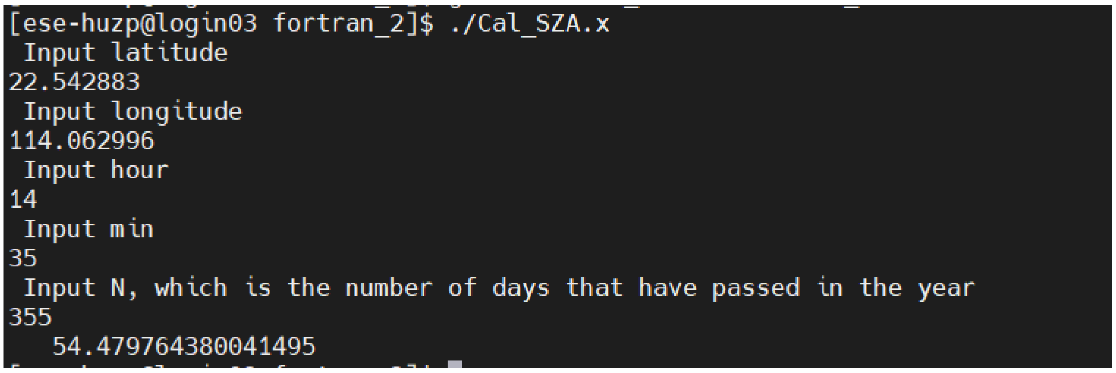
```
program Cal_SZA
use Declination_angle
use AST

implicit none

real(8) :: n1, delta1,lon,lat,hour,minu,a,lst1,ast1,H,SZA,SZA1

write(*,*) 'Input latitude'
read(*,*) lat
write(*,*) 'Input longitude'
read(*,*) lon
write(*,*) 'Input hour '
read(*,*) hour
write(*,*) 'Input min '
read(*,*) minu
write(*,*) 'Input N, which is the number of days that have passed in the year'
read(*,*) n1

lst1 = hour + minu/60
a= 3.1415926536/180

call cal_angle(n1,delta1)
call cal_ast(lst1,n1,lon,ast1)


H = (ast1*60 - 720)/4
SZA1 = cos(lat*a)*cos(delta1*a)*cos(H*a) + sin(lat*a)*sin(delta1*a)
SZA = ACOS(SZA1)/a

write(*,*) SZA

end program Cal_SZA
```

2.4



```
[ese-huzp@login03 fortran_2]$ ar rcvf libsolar.a AST.o Declination_angle.o
r - AST.o
r - Declination_angle.o
[ese-huzp@login03 fortran_2]$ gfortran Cal_SZA.f90 -o Cal_SZA.x -L. -lsolar
```

2.5



```
[ese-huzp@login03 fortran_2]$ ./Cal_SZA.x
 Input latitude
22.542883
 Input longitude
114.062996
 Input hour
14
 Input min
35
 Input N, which is the number of days that have passed in the year
355
   54.479764380041495
```