各位同学：

这是我自己的一段代码，大家好好看看我编写代码时注意的格式。重点注意格式和注释。这个程序不长，但是我也会注重这些细节。

龚明/LQCC

```
! Author info: Ming Gong, 2021/8/12
! The purpose of this program is to calculate a random matrix
! with coupling between localized modes and extended modes, using

!       | diagonal              |                              !
!       | Localized modes       | coupling between them        !
! H = ! --------------------------------------------!
!       | coupling              | Extended modes               !
!       |                       | diagonal matrix              !

program TwoMatrixCoupling
    implicit none
    double precision, parameter :: b = 0.43

    ! N total dimension of the matrix A
    ! Nk the averaged time of the random potential
    ! NS is used to calculate the distribution of level spacings
    integer, parameter :: N = 4000
    integer, parameter :: Nk = 20
    integer, parameter :: NS = 500

    ! N1 and N2 are the dimensions of the subspaces
    ! N1 + N2 = N, and 0 <= b <= 1.
    integer, parameter :: N1 = int(N * b)
    integer, parameter :: N2 = N - N1
    double precision     :: A(N, N), W(N)
    double precision     :: A1(N1, N1), W1(N1)

    ! used for lapack subroutines
    integer, parameter :: LWORK = 3 * N
    double precision     :: WORK(LWORK)
    integer              :: INFO

    ! distribution of eigenvalues
    ! means: mean level spacing
    ! ds is the increase spacing
    double precision     :: dis(1:NS)
    double precision     :: means, ds, tmp, sumdis
    double precision     :: sigma1, sigma2

    integer              :: i, j, k

    sigma1 = 0.0
    sigma2 = 0.1
    dis    = 0.0
```

```
       ! check the input data
       If( b. gt. 1.0d0 .or. B .lt. 0.0d0) then
           write(*, *) “ b should in [0, 1], error “
           call abort
       endif

       ds = 2.0d0 /(dble(NS) - 1.0d0)

       open(11, file='dis.dat', status='unknown')

       do k = 1, Nk
       A = 0.0d0

       ! init the random matrix of A1, which is Hermit.
       do i = 1, N1
           do j=i, N1
           A1(i,j) = (rand()-0.5)
           A1(j,i) = A1(i,j)
           end do
       end do

       !DSYEV computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix
A1.
       !call dsyev('V', 'U', N1, A1, N1, W1, WORK, LWORK, INFO)

       !
       !
       !        INFO
       !
       !                      INFO is INTEGER
       !                      = 0:   successful exit
       !                      < 0:   if INFO = -i, the i-th argument had an illegal value
       !                      > 0:   if INFO = i, the algorithm failed to converge; i
       !                              off-diagonal elements of an intermediate tridiagonal
       !                              form did not converge to zero.
       !

       If(INFO .ne. 0) then
           write(*, *) “ Failed output of dsyev (A1 matrix), info = “, INFO
           write(*, *) “ Please check the details from LAPACK “
           call abort
       endif


       do i = 1, N1
           !A(i, i) = W1(i) / W1(1)
           A(i, i) = 0.0d0
       end do

       do i = N1+1, N
           !A(i, i) = (rand() - 0.5d0)
           A(i, i) = 0.0d0
       end do

       do i = 1, N1
            do j = i, N1
            A(i, j) = A(i, j) + (rand()-0.5) * sigma1 / dble(N)
            A(j, i) = A(i, j)
            end do
        end do


        ! setup the random coupling between the localized modes and extended modes
        ! with coupling strength about x * sigma /Sqrt[L],
        ! where L is the chain length.

        do i = 1, N
         do j = N1 +1, N
           !A(i, j) = A(i, j) + (rand()-0.5) * sigma2 / dble(N)
           A(i, j) = A(i, j) + (rand()-0.5) * sigma2 / sqrt(dble(N))
           A(j, i) = A(i, j)
         end do
        end do

        !DSYEV computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix A.
        call dsyev('V', 'U', N, A, N, W, WORK, LWORK, INFO)


        If(INFO .ne. 0) then
           write(*, *) “ Failed output of dsyev (A matrix), info = “, INFO
           write(*, *) “ Please check the details from LAPACK “
           call abort
        endif

        ! with the eigenvalues, calculate the distribution function
        ! store the distribution /averaged in dis(i)

        means = (W(N) - W(1)) / (N-1.0d0)
        write(*, *) " # mean separation <s> = ", means, "      k = ", k

        do i = 1, N-1
        tmp = (W(i+1) - W(i))/means
        j = int(tmp/ds) + 1
        if(j .le. NS .and. j .ge. 1) then
            dis(j) = dis(j) + 1.0d0 / (N - 1.0d0)/Nk
        endif
        end do

        end do    ! end of k

        sumdis = 0.0d0
        do i = 1, NS
        sumdis = sumdis + dis(i)
        write(11, '(2f16.5)') (i-0.5)*ds, dis(i)/ds
        sumdis = sumdis + dis(i)
        end do

        write(*, *) " Sum of distribution = ", sumdis
        close(11)

end program
```