```
! TwoDim.F
! 2003-12-14
! NISHIMATSU Takeshi
!
! TwoDim.F and TwoDim.gp produce a band diagram of non-interacting
! electrons in a simple two-dimensional square-lattice potential.
! See Charles Kittel: "Introduction to Solid State Physics",
! 7th ed., Chapter 7 "Energy Bands", Problem 6.
!
! Compilation
! * Writen in Fortran90 free format. You may need a proper compile option,
!   such as -free, -FR, -ffree-form, etc. It depends on your compiler.
! * With a '-DDEBUG' compile option,
!   the Hamiltonian and its eigenvalues will be printed out.
!
! Execution:
! % echo '200 -3 3 0.05' | ./TwoDim
!!
#define MATRIX_DIM 100
program TwoDim
  implicit none
  integer n_samples
  integer min_G, max_G
  real*8 U
  real*8 x, k(2), k_plot
  real*8 G(2,MATRIX_DIM)
  integer nG, i, j, status

  read(5,*) n_samples, min_G, max_G, U

  nG=0
  do i=min_G, max_G
     do j=min_G, max_G
        nG=nG+1
        G(1,nG)=i
        G(2,nG)=j
     end do
  end do

  do i=0,n_samples
     x=1.0d0/2/n_samples*i
     k(1)=x;      k(2)=0
     call solve(k, x,               nG, G, U)
     k(1)=0.5d0; k(2)=x
     call solve(k, x+0.5d0,         nG, G, U)
     k(1)=x;      k(2)=x
     call solve(k, -x*sqrt(2.0d0), nG, G, U)
  end do
end program TwoDim
```

```
subroutine solve(k, k_plot,        nG, G, U)
  implicit none
  real*8 k(2), k_plot
  integer nG   ! The number of G's.
  real*8 G(2,*)
  real*8 U

  real*8 Hamiltonian(MATRIX_DIM,MATRIX_DIM)
  real*8 eigenvalues(MATRIX_DIM)
  real*8 work(MATRIX_DIM*3)
  real*8 k_minus_G(2)
  real*8 GG(2)
  integer i, j, info

  Hamiltonian=0.0d0   !<k-G_i|H|k-G_j> Compute lower triangle only
  do j=1,nG
   !Diagonal term
     k_minus_G = k - G(:,j)
     Hamiltonian(j,j) = ( k_minus_G(1)**2 + k_minus_G(2)**2 )/2 !Kinetic energy
   !Off-diagonal term
     do i=j+1,nG
        GG= -G(1:2,i) +G(1:2,j)
        if (    GG(1).eq. 1.0d0 .and. GG(2).eq. 1.0d0 .or. &
             &  GG(1).eq. 1.0d0 .and. GG(2).eq.-1.0d0 .or. &
             &  GG(1).eq.-1.0d0 .and. GG(2).eq. 1.0d0 .or. &
             &  GG(1).eq.-1.0d0 .and. GG(2).eq.-1.0d0 ) then
           Hamiltonian(i,j) = -U
        else
           Hamiltonian(i,j) = 0.0d0
        end if
     end do
  end do

#ifdef DEBUG
  do i=1,nG   ! Print the Hamiltonian
     write(6,'(30f5.2)') Hamiltonian(i,1:nG)
  end do
  write(6,*)
#endif

  ! call the LAPACK subroutine DSYEV() to compute all
  !  eigenvalues and eigenvectors of a real symmetric matrix
  call DSYEV('V', 'L', nG, Hamiltonian, MATRIX_DIM, eigenvalues, &
       & work, MATRIX_DIM*3, info)

  do j=1,nG
     write(6,'(f18.15,f20.15)') k_plot, eigenvalues(j)
#    ifdef DEBUG
     write(6,'(f11.7)') Hamiltonian(1:nG,j)   ! Print eigenvalues
     write(6,*)
#    endif
  end do
end subroutine solve

!Local variables:
!  compile-command: "f90 -free -o TwoDim TwoDim.F -lcxml && echo '1 0 1 0.05' |
./TwoDim"
!End:
```