```
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! This code finds the eigenvalues and eigenstates of a single GOE matrix.
!
! IPR for all eigenstates, density of states, and the level spacing
! distribution are computed.
!
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!             VARIABLES to be used in the whole code
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

       module variables

       implicit none

       integer (kind=4) :: i,j,k,ii
       integer (kind=4) :: dimTotal


! Eigenvectors and eigenvalues of a GOE Hamiltonian
       real (kind=8), dimension(:), allocatable :: Eiggoe
       real (kind=8), dimension(:,:), allocatable :: Vecgoe

! for the DIAGONALIZATION
       INTEGER (kind=4) :: INFO
       real (kind=8), dimension(:), allocatable :: work

! RANDOMNESS
       INTEGER (kind=4) :: idum
       real (kind=8) :: aux

! IPR
       real (kind=8) :: IPRgoe

! Density of states
       real (kind=8) :: Emin,Emax,bin
       INTEGER (kind=4), PARAMETER :: Nbin = 40
       real (kind=8), dimension(Nbin) :: Eint
       INTEGER (kind=4), dimension(Nbin) :: NinBin

! Level spacing distribution
       real (kind=8) :: percentage,average
       INTEGER (kind=4) :: half
       real (kind=8), dimension(:), allocatable :: spacing
       real (kind=8), PARAMETER :: spcmin = 0.0d0
```

```
        real (kind=8), PARAMETER :: spcmax = 8.0d0
        real (kind=8), PARAMETER :: binS = 0.1d0
        INTEGER (kind=4), PARAMETER :: NofBINs = 80
        INTEGER (kind=4), PARAMETER :: NofBINsPlus = 81
        real (kind=8), dimension(NofBINs) :: Nhist
        real (kind=8), dimension(NofBINsPlus) :: SPChist
        real (kind=8) :: normaliza


! For the OUTPUT files
        character(len=70) saida,saiHIS,saiPs


        end module


!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!              Program starts here
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        Program GOEmatrix

        use variables
        implicit none

        real (kind=8) :: ran1,gasdev


!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! PARAMETERS
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        dimTotal=3000

! Output File   ccccccccccccccccccc
        saida='GOE_En_IPR.dat'
        OPEN(unit=20, FILE=saida,STATUS='UNKNOWN')
        saiHIS='GOE_DOS.dat'
        OPEN(unit=30, FILE=saiHIS,STATUS='UNKNOWN')
        saiPs='GOE_Ps.dat'
        OPEN(unit=40, FILE=saiPs,STATUS='UNKNOWN')


        allocate(Eiggoe(dimTotal))
        allocate(Vecgoe(dimTotal,dimTotal))
        allocate(work(7*dimTotal))
```

```
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!c CREATE GOE RANDOM MATRIX and DIAGONALIZE it
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       Do i=1,dimTotal
         Do j=1,dimTotal
           Vecgoe(i,j)=gasdev(idum)
         Enddo
       Enddo
       Do i=1,dimTotal
         Do j=1,dimTotal
           aux=Vecgoe(i,j)+Vecgoe(j,i)
           Vecgoe(i,j)=aux
         Enddo
       Enddo
      CALL
DSYEV('V','U',dimTotal,Vecgoe,dimTotal,Eiggoe,WORK,7*dimTotal,INFO)


!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! IPR of the EIGENSTATES of the GOE RANDOM MATRIX
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

       DO i=1,dimTotal
         aux=0.0d0
         Do j=1,dimTotal
           aux=aux+Vecgoe(j,i)**4
         Enddo
         IPRgoe=1.0d0/aux
         write(20,120) i,Eiggoe(i),IPRgoe
       ENDDO
120    Format (I5,2E18.9)


!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! DENSITY OF STATES
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

       Emin=Eiggoe(1)
       Emax=Eiggoe(dimTotal)
       bin = (Emax-Emin)/dfloat(Nbin-1)
       Do i=1,Nbin+1
         Eint(i)=Emin-bin-bin/2.0d0 + bin*dfloat(i)
       Enddo

       Do i=1,Nbin
         NinBin(i) = 0
```

```
       Enddo

       Do i=1,dimTotal
        do k=1,Nbin
          If(Eiggoe(i) >= Eint(k) .AND. Eiggoe(i) < Eint(k+1)) then
          NinBin(k) = NinBin(k)+1
          Endif
        enddo
       Enddo

! Normalized density of states
        write(30,130) Eint(1),0.0d0
       Do k=1,Nbin
        write(30,130) Eint(k),dfloat(NinBin(k))/(bin*dfloat(dimTotal))
        write(30,130) Eint(k+1),dfloat(NinBin(k))/(bin*dfloat(dimTotal))
       Enddo
130    Format (2E18.9)


!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
! LEVEL SPACING DISTRIBUTION
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       percentage=0.1d0*dfloat(dimTotal)
       half=int(percentage/2.0d0)
       aux=(dfloat(dimTotal)-percentage)
       allocate(spacing(int(aux)))

! Unfolded Spacings
       aux=(dfloat(dimTotal)-percentage)/(10.0d0)
       Do j=1,int(aux)
         average=(Eiggoe(half+10*j)-Eiggoe(half+10*(j-1)))/(10.0d0)
         Do i=1+10*(j-1),10*j
       spacing(i)=(Eiggoe(half+i)-Eiggoe(half-1+i))/average
         Enddo
       Enddo
! For the Histogram
       SPChist(1)=spcmin
       Do i=1,NofBINs
         SPChist(i+1)=SPChist(i)+binS
         Nhist(i)=0.0d0
       Enddo
! Histogram
       Do k=1,10*int(aux)
         Do j=1,NofBINs
          If(spacing(k) >= SPChist(j) .AND. spacing(k) < SPChist(j+1)) then
            Nhist(j) = Nhist(j) + 1.0d0
```

```
          Endif
         Enddo
       Enddo
! Normalization
       normaliza=0.0d0
       Do i=1,NofBINs
         normaliza=normaliza+binS*Nhist(i)
       Enddo
! Output
        write(40,130) SPChist(1),0.0d0
       Do i=1,NofBINs
        write(40,130) SPChist(i),Nhist(i)/normaliza
        write(40,130) SPChist(i+1),Nhist(i)/normaliza
       Enddo



!ccccccccccccccccccccccccccccccccccccccccccccccccc
!c END END END END END END END END END END END END
!ccccccccccccccccccccccccccccccccccccccccccccccccc
!      STOP
       END



!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!ccccccccccccccc FUNCTIONS FUNCTIONS FUNCTIONS ccccccccccccccccc
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

!  Functions to generate random numbers from uniform and Gaussian
distributions
!  using Numerical Recipes ran1 and gasdev routines.
!  ran1 gives uniform random numbers between 0 and 1
!  gasdev gives random numbers from a Gaussian distribution with zero mean
!  and unit variance

! Random number
       FUNCTION ran1(idum)
       INTEGER (kind=4) :: idum,IA,IM,IQ,IR,NTAB,NDIV
       REAL (kind=8) :: ran1,AM,EPS,RNMX
       PARAMETER (IA=16807,IM=2147483647,AM=1.0d0/IM,IQ=127773,IR=2836)
       PARAMETER (NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
       INTEGER (kind=4) :: j,k,iv(NTAB),iy
       SAVE iv,iy
       DATA iv /NTAB*0/, iy /0/
       if (idum.le.0.or.iy.eq.0) then
```

```
        idum=max(-idum,1)
        do 11 j=NTAB+8,1,-1
        k=idum/IQ
        idum=IA*(idum-k*IQ)-IR*k
        if (idum.lt.0) idum=idum+IM
        if (j.le.NTAB) iv(j)=idum
11       continue
        iy=iv(1)
        endif
        k=idum/IQ
        idum=IA*(idum-k*IQ)-IR*k
        if (idum.lt.0) idum=idum+IM
        j=1+iy/NDIV
        iy=iv(j)
        iv(j)=idum
        ran1=min(AM*iy,RNMX)
        return
        END

! Gaussian random number
        FUNCTION gasdev(idum)
        INTEGER (kind=4) :: idum
        REAL (kind=8) :: gasdev
!       USES ran1
        INTEGER (kind=4) :: iset
        REAL (kind=8) :: fac,gset,rsq,v1,v2,ran1
        SAVE iset,gset
        DATA iset/0/
        if (iset.eq.0) then
1        v1=2.0d0*ran1(idum)-1.0d0
         v2=2.0d0*ran1(idum)-1.0d0
         rsq=v1**2+v2**2
         if(rsq.ge.1..or.rsq.eq.0.)goto 1
         fac=sqrt(-2.0d0*log(rsq)/rsq)
         gset=v1*fac
         gasdev=v2*fac
         iset=1
        else
         gasdev=gset
         iset=0
        endif
        return
        END
```

```
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!ccccccccccccccc SUBROUTINES SUBROUTINES SUBROUTINES ccccccccccccc
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```