Learning QE programming in 18 easy steps-3rd Day

P. Umari
Università degli Studi di Padova, Italy
CNR-IOM Democritos, Trieste, Italy

Aaargh! We forgot the .igk files

1. Go to the pw.x sources in directory PW/src/
2. In close_files.f90 change:

```
IF ( opnd ) CLOSE( UNIT = iunigk, STATUS = 'DELETE' )
```

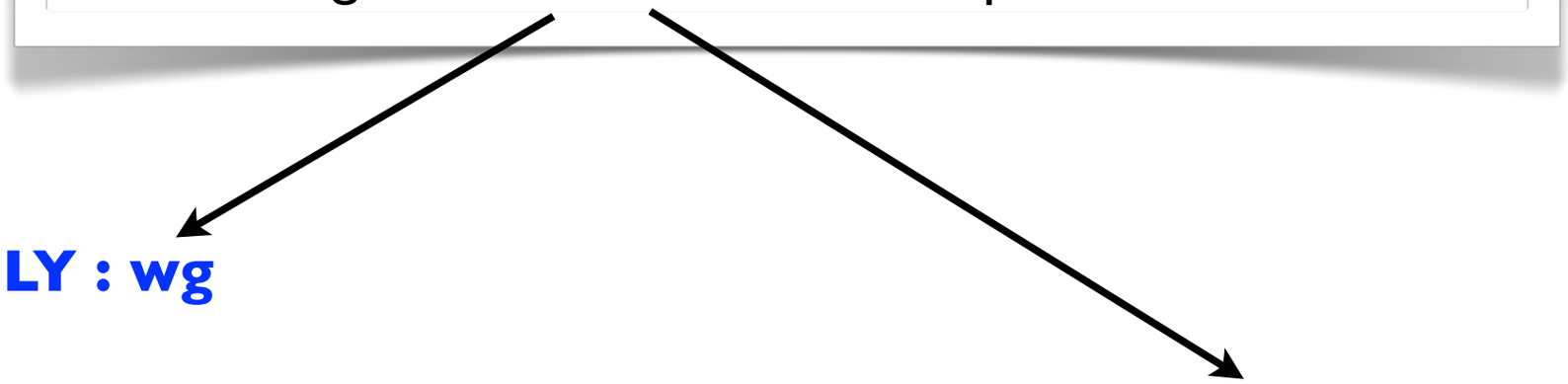
with:

```
IF ( opnd ) CLOSE( UNIT = iunigk)
```

3. Compile again pw.x

Step 13:Calculate electronic charge: Γ-point only case

wg REAL(nbnd,nks)
relative weigth of each band at each k-point



USE wvfct, ONLY : wg

...

charge(1:dfftp%nnr)=charge(1:dfftp%nnr)+gridd(1:dfftp%nnr)*wg(i,is)

Step 14:Calculate electronic charge: k-points case

We need to symmetrize the charge density

```
USE symme, ONLY : sym_rho_init, sym_rho_deallocate,sym_rho
USE gvect, ONLY : ngm,nl
complex(kind=DP), allocatable :: charge_g(:)

...
CALL sym_rho_init (gamma_only )

...

 allocate(charge_g(ngm))
 psic(1:dfftp%nnr)=dcmplx(charge(1:dfftp%nnr),0.d0)
 CALL fwfft ('Dense', psic, dfftp)
 charge_g(1:ngm)=psic(nl(1:ngm))
 call   sym_rho (nspin, charge_g)

...
CALL sym_rho_deallocate
```

sym_rho_init, sym_rho_deallocate
for setting up and clearing symmetry related arrays

ngm INTEGER max local number of G in dense grid
nl INTEGER (ngm) correspondence table for dense grid

call sym_rho (nspin, charge_g)
the charge in G space is symmetrized

Step 15:Write charge along a crystal axis on disk

First we collect all the entire charge density grid on every MPI task:

```
USE mp_global,   ONLY : me_pool
USE cell_base, ONLY: at, alat, tpiba, omega, tpiba2
...
real(kind=DP), allocatable :: x(:,:,:,:),y(:,:,:)
 integer :: nr3_start,nr3_end
 integer :: ir,ii,ix,iy,iz,iqq

 allocate(x(3,dfftp%nr1,dfftp%nr2,dfftp%nr3),y(dfftp%nr1,dfftp%nr2,dfftp%nr3))

 x=0.d0
 y=0.d0

 nr3_start=0
 nr3_end =0
 do ii=1,me_pool + 1
   nr3_start=nr3_end+1
   nr3_end=nr3_end+dfftp%npp(ii)
 end do

 do iz=1,dfftp%npp(me_pool+1)
   do iy=1,dfftp%nr2
     do ix=1,dfftp%nr1
       iqq=(iz-1)*(dfftp%nr1x*dfftp%nr2x)+(iy-1)*dfftp%nr1+ix
       y(ix,iy,iz+nr3_start-1)=charge(iqq)
     enddo
   enddo
 enddo

 call mp_sum(y)

 do iz=1,dfftp%nr3
   do iy=1,dfftp%nr2
     do ix=1,dfftp%nr1
       x(1:3,ix,iy,iz)=at(1:3,1)*dble(ix-1)/dble(dfftp%nr1)+at(1:3,2)*dble(iy-1)/dble(dfftp%nr2)+&
           &  at(1:3,3)*dble(iz-1)/dble(dfftp%nr3)
     enddo
   enddo
 enddo

 x=x*alat
...
```

me_pool INTEGER MPI-ID of the task

at REAL (3,3) crystal axis in alat units
alat REAL lattice length in Bohr
tpiba REAL 2*π/alat
omega REAL cell volume in Bohr**3

dfftp%npp INTEGER (nproc) number of planes of the charge grid belonging to the MPI tasks

Step 16:Calculate the Hartree contribution to the total energy (in Rydberg!!)

```
USE gvect, ONLY : gg
USE cell_base, ONLY :  tpiba2
USE constants, ONLY : e2,  fpi
...
real(kind=DP), allocatable :: fac(:)
...
 allocate(fac(ngm))
   fac=0.d0
   do  ig = gstart, ngm
     fac(ig) = 1.D0 / gg(ig)
   enddo
   fac = fac * 0.5d0* e2 * fpi / tpiba2 / omega
   ene=0.d0
   do ig=gstart,ngm
     ene=ene+fac(ig)*(dble(charge(ig))**2.d0+dimag(charge(ig))**2.d0)
   enddo
   call mp_sum(ene)
```

gg REAL(ngm)modulus square of G vectors in units of tpiba2

tpiba2 REAL tpiba**2

e2 REAL square of elementary charge (=2 in Rydber atomic units)
fpi REAL 4*π

Step 17:Calculate the Hartree contribution to the total energy, a small variation

g REAL(3,ngm) G vectors in units of tpiba

bg REAL(3,3) reciprocal crystal axis in units of tpiba

```
USE gvect, ONLY :  g
USE cell_base, ONLY :  bg

...
vecg(1:3)=g(1:3,ig)
 fac(ig) = 1.D0 /(vecg(1)**2.d0+vecg(2)**2.d0+vecg(3)**2.d0)
```

Step 18:Apply H, and verify that they are really eigenstates with the good eigen-energy

g2kin REAL(npwx)kinetic term of H

```
USE wvfct,              ONLY : g2kin

...
CALL init_us1
CALL init_us_2( npw, igk, xk(1,ik), vkb )
 g2kin(1:npw) = ((g(1,1:npw)+xk(1,ik))**2 +(g(2,1:npw)+xk(2,ik))**2 +  (g(3,1:npw)+xk(3,ik))**2 ) * tpiba2

...
 call h_psi( npwx, npw, nbnd,evc, hevc )
```

call h_psi(npwx, npw, nbnd,evc, hevc)
hevc will contain H|psi> where psi are given in evc