Second ABINIT Workshop 10-12 May 2004

# IMPLEMENTATION OF THE PAW FORMALISM IN ABINIT: PRESENT FEATURES AND PERSPECTIVES

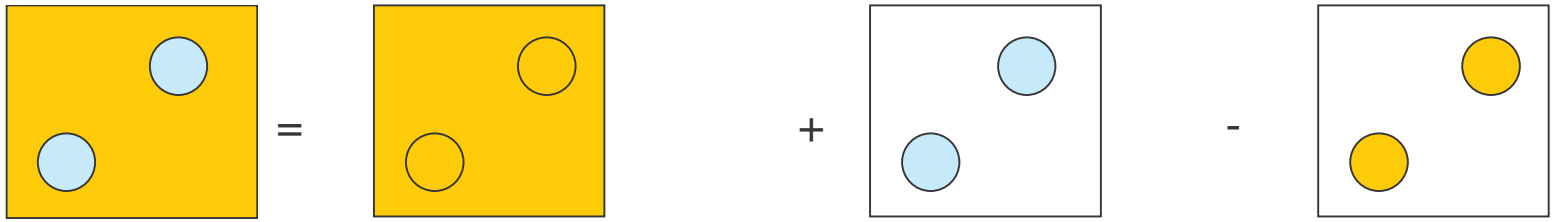
F. Jollet, M. Torrent, F. Bottin,G. Zérah

CEA-DAM, Centre d'Etudes de Bruyères le Châtel

# The PAW method

Wavefunction:

$$|\Psi_n\rangle = |\tilde{\Psi}_n\rangle + \sum_i \left(|\phi_i\rangle - |\tilde{\phi}_i\rangle\right)\langle \tilde{p}_i | \tilde{\Psi}_n\rangle$$




Operators:

$$\langle A\rangle = \sum_n f_n \langle \Psi_n | A | \Psi_n\rangle = \sum_n f_n \langle \tilde{\Psi}_n | \tau^* A \tau | \tilde{\Psi}_n\rangle$$

Density:

$$n(\mathbf{r}) = \tilde{n}(\mathbf{r}) + \sum_R \left(n_R^1(\mathbf{r}) - \tilde{n}_R^1(\mathbf{r})\right)$$

Energy:

$$E = \tilde{E} + \sum_R \left(E_R^1 - \tilde{E}_R^1\right)$$

Hamiltonian:

$$H = \frac{dE}{d\tilde{\rho}} = -\frac{1}{2}\Delta + \tilde{v}_{eff} + \sum_{i,j} |\,\tilde{p}_i\rangle D_{i,j} \langle \tilde{p}_j\,| \qquad \tilde{v}_{eff} = v_H[\tilde{n} + \hat{n} + \tilde{n}_{Zc}] + v_{xc}[\tilde{n} + \hat{n} + \tilde{n}_c]$$

$$D_{ij} = D_{ij}^0 + \sum_{kl} \rho_{kl} E_{ijkl} + D_{ij}^{xc} + \sum_L \int \tilde{v}_{eff}(\mathbf{r}) \hat{Q}_{ij}^L(\mathbf{r}) d\mathbf{r} \qquad \rho_{i,j} = \sum_n f_n \langle \tilde{\Psi}_n | \tilde{p}_j \rangle \langle \tilde{p}_i | \tilde{\Psi}_n \rangle$$

Ultrasoft approximation

➢ $$D_{i,j} = D_{i,j}^0 + \sum_L \int \tilde{v}_{eff}(\mathbf{r}) \hat{Q}_{i,j}^L(\mathbf{r}) d\mathbf{r}$$

Norm conserving approximation

$$\hat{Q}_{i,j}^L(r) = 0$$

➢ $$D_{i,j} = D_{i,j}^0 \qquad S=I$$

# Implementation in ABINIT

## Norm conserving

$$H\varphi_i = \varepsilon_i \varphi_i$$

$$H = \frac{dE}{d\tilde{\rho}} = -\frac{1}{2}\Delta + \tilde{v}_{eff} + \sum_i |p_i\rangle D_i^0 \langle p_i|$$

i over quantum numbers l

$D_i^0$ is constant

## PAW

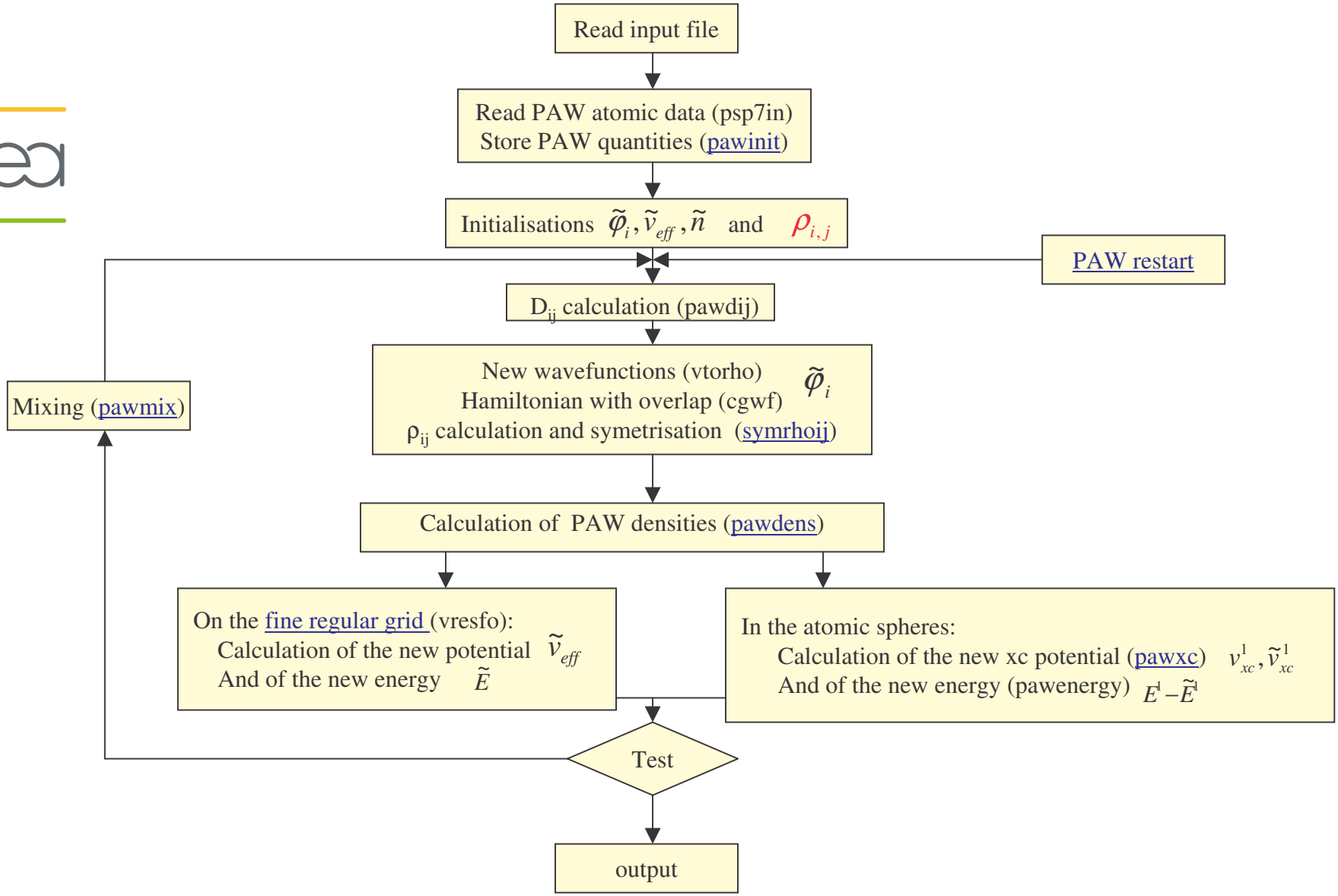
$$H\varphi_i = \varepsilon_i S \varphi_i$$

$$H = \frac{dE}{d\tilde{\rho}} = -\frac{1}{2}\Delta + \tilde{v}_{eff} + \sum_{i,j} |\tilde{p}_i\rangle D_{i,j} \langle \tilde{p}_j|$$

i over quantum numbers l and m

$D_{i,j}$ is updated at each iteration

# PAW- ABINIT


Read input file
Read PAW atomic data (psp7in)
Store PAW quantities (pawinit)
Initialisations $\tilde{\varphi}_i, \tilde{v}_{eff}, \tilde{n}$ and $\rho_{i,j}$
PAW restart
$D_{ij}$ calculation (pawdij)
New wavefunctions (vtorho) $\tilde{\varphi}_i$
Hamiltonian with overlap (cgwf)
$\rho_{ij}$ calculation and symetrisation (symrhoij)
Mixing (pawmix)
Calculation of PAW densities (pawdens)
On the fine regular grid (vresfo):
Calculation of the new potential $\tilde{v}_{eff}$
And of the new energy $\tilde{E}$
In the atomic spheres:
Calculation of the new xc potential (pawxc) $v^1_{xc}, \tilde{v}^1_{xc}$
And of the new energy (pawenergy) $E^1 - \tilde{E}^1$
Test
output

# Efficiency

☺ Good ultrasoft PAW atomic data
☺ Adapted radial grids (psp7in)
☺ Store real gaunt coefficient (pawinit)
☺ Store atomic data related quantities ($E_{ijkl}$ in pawinit)
☺ Use the non-zero radial part of the density (pawdens)
☺ Use the radialpart of the XC potentials (pawxcm)
☺ Symetrize $\rho_{i,j}$ (symrhoij)
☺ Mix the spherical part (pawmix)

→ the spherical part is efficient

☺Possibility to calculate forces at the end of electronic steps

☹ More use of BLAS
☹ Improve the eigenvalue search with overlap (RMM-DIIS ?)

# Accuracy

☺ Use XC potentials without second order development (LDA only, pawxc)
input keyword: pawxcdev

☺ Use good PAW atomic data

☺ Increase pawmqgrid

# How does a PAW calculation work?

- At first order:
  - Only change all the pseudopotential files
- At second order:
  - test the convergency of the fine grid (ecutdg or ngfftdg)
  - Change pawsphmix to tune the spherical mixing
  - Change pawxcdev to desactivate the second order expansion of XC potentials (spherical grid: pawntheta and pawnphi)
  - Activate lcutd to cut spherical harmonics expansion
- At third order:
  - Increase pawmqgrid

# Example of fcc Ca

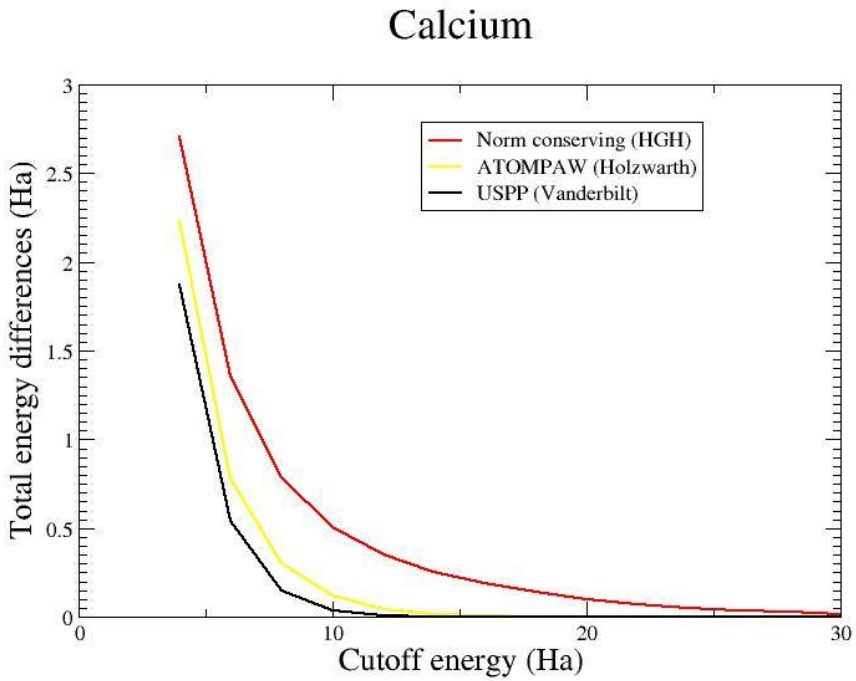
For Ca:

Pseudos:

HGH, PAW-uspp,PAW-atompaw

Cutoff for a convergency < 1mHa

HGH: >40 Ha
PAW-uspp: 20 Ha
PAW-atompaw: 20 Ha




$a_{0\_HGH}$=10.3 a.u.

$a_{0\_PAW\text{-}uspp}$=10.2 a.u.

# Example of fcc oxygen

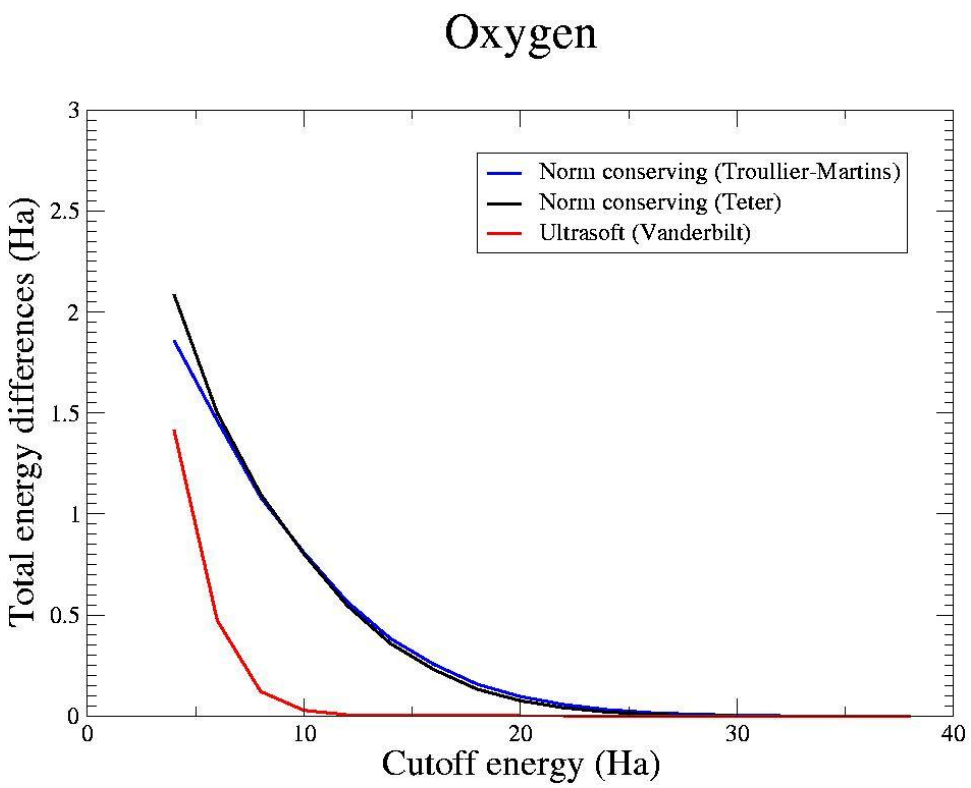
## For O:

Pseudos:

Teter, TM, PAW-uspp

Cutoff for a convergency < 1mHa

Teter: 30 Ha
TM: 33 Ha
PAW-uspp: 21 Ha




$a_{0\_TM}$=3. 11 Å

$a_{0\_PAW\text{-}uspp}$=3.06 Å

# Example of $BaTiO_3$

## For $BaTiO_3$:

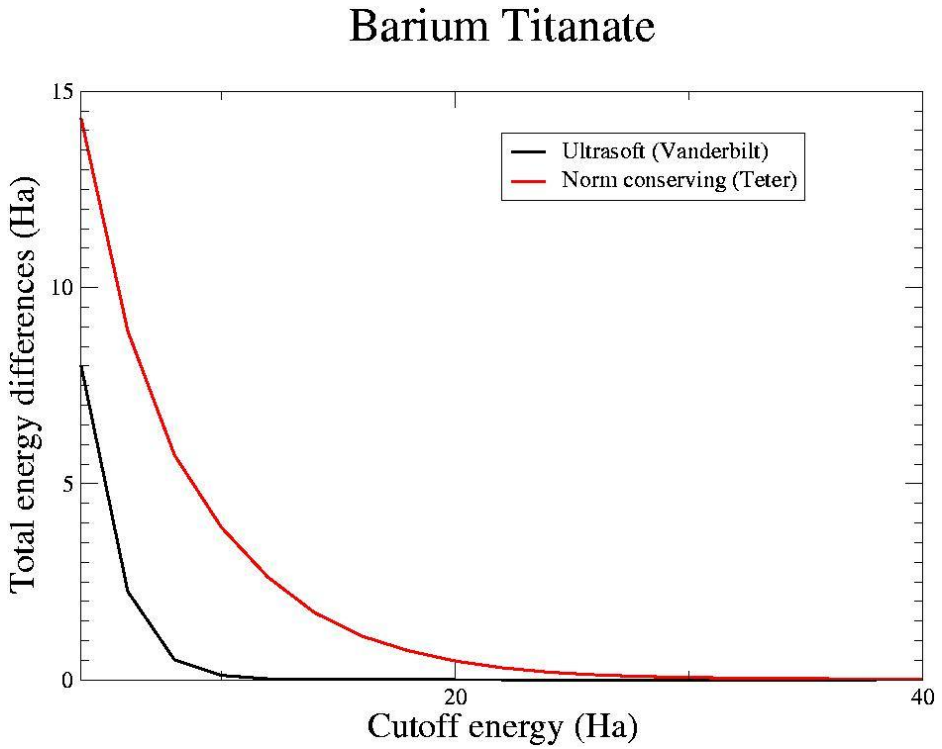
Pseudos:

Teter, PAW-uspp

Cutoff for a convergency < 1mHa

Teter: 56 Ha

PAW-uspp: 30 Ha




$a_0$_Teter=7.45 a.u.

$a_0$_PAW-uspp=7.48 a.u.

# Conclusion

- What is done:
  - Calculation of the total energy
  - Coding of two interfaces to generate atomic data
- What is to be done
  - Calculation of forces and stresses
  - Calculation of linear response
  - Latex Documentation
- What is to be improved
  - The intrinsic efficiency of the code (sequential)
  - The parallelisation (on atoms ?)

# pawinit

```
!! FUNCTION
!! Initialize some starting values for several arrays used by PAW calculation
!!
!! 1-Initialize data related to angular mesh
!! 2-Tabulate normalized shape function g(r)
!! 3-Compute dltij factors (used in sums over klmn=(ilmn,jlmn)))
!! 4-Compute $q_ijL=\displaystyle
!!                  \int_{0}^{r_c}{(\phi_i\phi_j-\widetilde{\phi_i}\widetilde{\phi_j}) r^l\,dr}
!!                   Gaunt(l_i m_i,l_j m_j,l m))$
!!           $S_ij=\displaystyle \sqrt{4 \pi} q_ij0$
!! 5-Compute $e_ijkl= vh1_ijkl - Vhatijkl - Bijkl - Cijkl$
!!     With:
!!       $vh1_ijkl =\sum_{L,m} {vh1*Gaunt(i,j,Lm)*Gaunt(k,l,Lm)}$
!!       $Vhat_ijkl=\sum_{L,m} {vhatijL*Gaunt(i,j,Lm)*q_klL}$
!!       $B_ijkl   =\sum_{L,m} {vhatijL*Gaunt(k,l,Lm)*q_ijL}$
!!       $C_ijkl   =\sum_{L,m} {intvhatL*q_ijL*q_klL}$
!!     and:
!!       vh1 according to eq. (A17) in Holzwarth et al., PRB 55, 2005 (1997)
!! 6-Compute Ex-correlation energy for the core density
!!
!! OUTPUT
!!  pawang
!!     %ngnt=number of non-zero Gaunt coefficients
!!     %realgnt(pawang%ngnt)=non-zero real Gaunt coefficients
!!  pawtab(ntypat) <type(pawtab_type)>=paw tabulated data read at start:
!!     %eijkl(lmn2_size,lmn2_size)=Part of the Dij that depends only from the projected occupation coeffs
!!     %gnorm(l_size)=Normalization factor of radial shape function
!!     %qijl(l_size**2,lmn2_size)=Moments of the difference charge density between AE and PS partial wave
!!     %shapefunc(mesh_size,l_size)=Normalized radial shape function
!!     %sij(lmn2_size)=Nonlocal part of the overlap operator
```

Return

# PAW restart

- Aim: to be able to start from a previous calculation with PAW (irdwfk, getwfk)

- The header of i/o files has been modified (hdrio)
  - usepaw=0 or 1
  - ecutdg (ecuteff or ecutdg)
  - lmnsize (lmnmax or lmnsize)
  - when PAW: $\rho_{i,j}$ on one line at the end of the header

- The initialisation of the calculation is made, taking into account the content of the header

Return

# symrhoij

```
!! FUNCTION
!! Symmetrize rhoij
!!
!! INPUTS
!!  indlmn(6,lmnmax,ntypat)= array giving l,m,n,ln,lm,spin for i=lmn
!!  indsym(4,nsym,natom)=indirect indexing array for atom labels
!!  lmnmax=maximum number of PAW radial wavefunctions
!!  natom=number of atoms in cell
!!  nsym=number of symmetry elements in space group
!!  ntypat=number of types of atoms in unit cell.
!!  paw_ij(natom) <type(paw_ij_type)>=paw arrays given on (i,j) channels
!!  pawang <type(pawang_type)>=angular mesh discretization and related data
!!  typat(natom)=type for each atom
!!
!! OUTPUT
!!  paw_ij(iatom)%rhoij(:,:)= symetrized paw rhoij quantities
!!
!! SIDE EFFECTS
!!
!! NOTES
!!  This file comes from the file SymWij.f by N.A.W. Holzwarth and A.
!!  Tackett for the code pwpaw
!!
!! PARENTS
!!      vtorho
!!
!! CHILDREN
!!      print_ij
!!
```

Return

# pawdens

```
!! FUNCTION
!! Compute various PAW densities inside spheres around atoms
!! (rho1, trho1 and nhat)
!!
!! INPUTS
!!  nzlmopt= if 0, compute all LM-moments of densities
!!           if 1, compute only non-zero LM-moments of densities (indexes of
!!                 non-zero moments were stored before)
!!
!! OUTPUT
!!  if nzlmopt==0,
!!      paw_an(iatom)%lnmselect(lm_size,nspden)=select the non-zero LM-moments of rho1 and trho1
!!  pawnhat1(nfft,ispden)=nhat on fine rectangular grid        n̂(x,y,z)
!!  paw_an(iatom)%rho1 (ir,inpt,ispden)=rho1 on spherical mesh        n^1_LM(r)
!!  paw_an(iatom)%trho1hat(ir,inpt,ispden)=trho1+nhat on spherical mesh    ñ^1_LM(r)+n̂_LM(r)
!!
!! PARENTS
!!      scfcv
!!
```

$$n(r,\theta,\varphi) = \sum_{LM} n_{LM}(r) S_{LM}(\theta,\varphi)$$

Return

# pawxc and pawxcm

```
!! pawxc
!!
!! FUNCTION
!! PAW only
!! Start from the density or spin-density, and compute xc potential and energies inside a paw sphere.
!! Driver of XC functionals.
!! Only treat colinear spins.
!!
!! INPUTS
!!  option=0 for xc from a generic density
!!         1 for xc from core density (spherical symetry + no dble-counting terms)
!! OUTPUT
!!  enxc=returned exchange and correlation energy (hartree)
!!  enxcdc=returned exchange-cor. contribution to double-counting energy
!!  vxc(pawrad%mesh_size,pawang%angl_size,nspden)=xc potential
!!         (spin up in 1st half and spin-down in 2nd half if nspden=2)

!! pawxcm
!!
!!  vxc(pawrad%mesh_size,lm_size,nspden)=xc potential
!!         (spin up in 1st half and spin-down in 2nd half if nspden=2)
```

$$v_{xc}(r,\theta,\varphi)=\sum_{LM} v_{LM}^{xc}(r)S_{LM}(\theta,\varphi)=v_{xc}\left[n_0(\vec{r})\right]+\left[n(\vec{r})-n_0(\vec{r})\right]\frac{dv_{xc}}{dn}\left[n_0\right]+\frac{\left[n(\vec{r})-n_0(\vec{r})\right]^2}{2}\frac{d^2v_{xc}}{dn^2}\left[n_0\right]$$

➔ GGA

Return

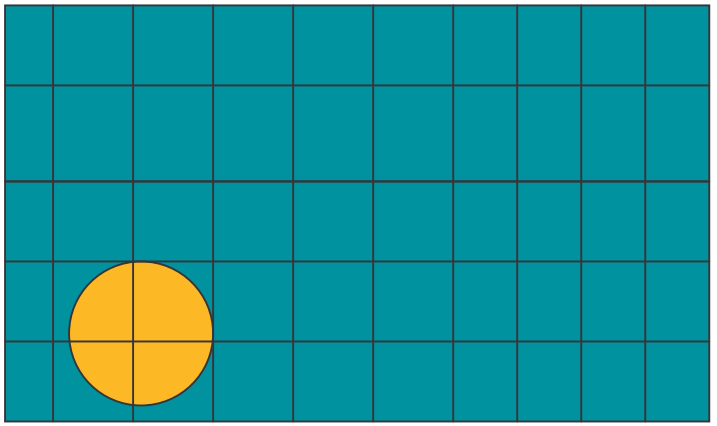
# pawmix

```
!! FUNCTION
!! Mix several (PAW) quantities before sending them to the next SCF cycle.
!! Only simple mixing is available.
!!
!! INPUTS
!!  istep=step number in the main loop of scfcv
!!  ixc= choice of exchange-correlation scheme (see above, and below)
!!  mixtyp=type of mixing (0=no mixing, 1=mixing based on rhoij, 2=simple mixing)
!!  natom=number of atoms in cell.
!!  nspden=number of spin-density components
!!  ntypat=number of types of atoms in unit cell.
!!  paw_an(natom) <type(paw_an_type)>=paw arrays given on angular mesh
!!  paw_ij(natom) <type(paw_ij_type)>=paw arrays given on (i,j) channels
!!  pawang <type(pawang_type)>=paw angular mesh and related data
!!  pawrad(ntypat) <type(pawrad_type)>=paw radial mesh and related data
!!  pawtab(ntypat) <type(pawtab_type)>=paw tabulated starting data
!!  pawxcdev=Choice of XC development (0=no dev. (use of angular mesh) ; 1=dev. on moments)
!!  sphmix=mixing factor for the spherical part of total PAW energy
!!  typat(natom)=type (integer) for each atom
!!
!! SIDE EFFECTS
!!  paw_an(natom)%vxc1[m](mesh_size,lm_size,nspden)=XC potential calculated from spherical density
!!  paw_an(natom)%vxct1[m](mesh_size,lm_size,nspden)=XC potential calculated from spherical pseudo
                                                                                    density
!!  paw_ij(natom)%veijkl(lmn2_size)=factors usefull for Hartree energy calculation
!!
!! PARENTS
!!      scfcv
!!
!! CHILDREN
!!      pawxc,pawxcm,wrtout
```

Return

# The fine regular grid

- A coarse grid is used to obtain wavefunctions $\tilde{\varphi}$
- We need $\hat{n}$ on the regular and on the radial grid (nhatgrid + pawdens)
- For accuracy, a fine grid is used
  to calculate $\tilde{v}_{eff} = v_H[\tilde{n} + \hat{n} + \tilde{n}_{Zc}] + v_{xc}[\tilde{n} + \hat{n} + \tilde{n}_c]$



- In transgrid: $\tilde{n}_{CG}(G) \rightarrow \tilde{n}_{FG}(G) \rightarrow \tilde{n}_{FG}(\vec{r})$
- In vresfo: $\tilde{n}_{FG}(\vec{r}) + \hat{n}_{FG}(\vec{r}) \rightarrow \tilde{v}_{eff}^{FG}$
- In transgrid: $\tilde{v}_{eff}^{FG} \rightarrow \tilde{v}_{eff}^{CG}$

Return