Notes on generalized coordinates

N. A. W. Holzwarth 12-02-01 (updated 07-19-05)

For some structures it is convenient to introduce generalized coordinates to represent that atomic positions $\mathbf{R}^a$, where a denotes an atomic index. We consider a set of NPARM parameters $\{C_p\}$ which are related to the atomic positions according to:

$$\mathbf{R}^a = \mathbf{R}^a_0 + \sum_{p \in \mathrm{map}} C_p \mathbf{V}^{\mathbf{a}}_{\mathbf{p}}, \tag{1}$$

where $\mathbf{R}^a_0$ denotes a local (fixed) origin for atom a. The summation over the parameters p includes at most 3 coefficients C_p with unit vectors $\mathbf{V}^{\mathbf{a}}_{\mathbf{p}}$. The parameters behave as generalized coordinates and it is possible to introduce generalized forces in terms of derivatives of the total energy $E \equiv E(\{\mathbf{R}^a\})$ according to

$$F_p = -\frac{\partial E}{\partial C_p} = -\sum_a \frac{\partial E}{\partial \mathbf{R}^a} \cdot \frac{\partial \mathbf{R}^a}{\partial C_p} = -\sum_a \frac{\partial E}{\partial \mathbf{R}^a} \cdot \mathbf{V}^a_p \tag{2}$$

.

The generalized coordinates $\{C_p\}$ are only used to move the atoms and the PAW calculation is entirely done in terms of the atomic positions $\{\mathbf{R}^a\}$. To determine the generalized coordinates, we calculate

$$C_p = \frac{1}{N_p} \sum_a (\mathbf{R}^a - \mathbf{R}^a_0) \cdot \mathbf{V}^a_p, \tag{3}$$

where N_p denotes the number of atoms which depend on C_p. The generalized forces $\{F_p\}$ are used to update the generalized coordinates $\{C_p\}$ which are then used to update the atomic positions $\{\mathbf{R}^a\}$ according to Eq. (1). Equation (3) works well when the vectors $\{\mathbf{V}^a_p\}$ are all orthogonal to eachother. However, for hexagonal, monoclinic, or other structures, the direction vectors may not be orthogonal. In those cases, we must construct a set of auxiliary vectors $\{\bar{\mathbf{V}}^a_p\}$ with the property

$$\mathbf{V}^a_p \cdot \bar{\mathbf{V}}^a_{p'} = \delta_{pp'}, \tag{4}$$

and Eq. (3) becomes

$$C_p = \frac{1}{N_p} \sum_a (\mathbf{R}^a - \mathbf{R}^a_0) \cdot \bar{\mathbf{V}}^a_p. \tag{5}$$

The auxiliary vectors $\{\bar{\mathbf{V}}^a_p\}$ are calculated internally within the program from a knowledge of the directional vectors $\{\mathbf{V}^a_p\}$.

In the *pwpaw* program the generalized coordinates are activated by a keyword *USESHELLS* for the case that all of the local origins $\mathbf{R}^a_0$ are zero or *USESHELLS_WITHORIGIN* for the more general case. There are two data structures and two global variables which are defined for this purpose:

```
Type Shell_Map                        !** Map atomic positions to parameters
  Integer :: MapParams                !** Total parameters of this atom (<=3)
  Real              :: Origin(3)      !** Constant position vector for origin
```

```
  Integer, Pointer     :: Map(:)   !** Map to parameter indices
  Real, Pointer     :: V(:,:)      !** MapParams unit vectors to define pos
  Real, Pointer     :: IV(:,:)     !** Inverse unit vectors
End Type

Type Shell_Structure               !** Atom position parametrization
  Integer :: NPARAMS               !** Total number of shell parameters
  Real, Pointer :: C(:)            !** List of parameter values
  Real, Pointer :: ShellForce(:)   !** Generalized forces
  Type (Shell_Map), Pointer :: AtomMap(:) !Parameters map to atoms
End Type

Logical :: UseShells               !** Flag for using shell map

Type (Shell_Structure) :: Shell
```

If either of the *USESHELLS* or *USESHELLS_WITHORIGIN* keywords are included in the input file, then the global variable *UseShells* is set to *true*. The input file then reads in the value of *Shell%NPARMS*. The arrays Shell%C(:) and Shell%ShellForce(:) are then allocated to the value of *NPARMS* and array Shell%AtomMap(:) is allocated to the number of atoms in the unit cell. The input file then lists all of the atoms in the unit cell. For each atom a, Shell%AtomMap(a)%MapParams≥ 0 must be given. If the value of MapParams is greater than 0, then Shell%AtomMap(a)%Map(:) and Shell%AtomMap(a)%V(3,:) are allocated to the value of MapParams. Then the local origin is optionally read in, followed by the MapParams values of the parameter map coefficient (corresponding to p) and unit vectors $\mathbf{V_p^a}$ are read in. An example of an input file for a 20 atom unit cell of a CdSe slab, not using any local origin shifts, is given below.

```
USESHELLS   12
   Cd1  0
   Cd3  0
   Cd5  0
   Cd7  0
   Se2  0
   Se4  0
   Se6  0
   Se8  0
   Cd9  0
   Cd11 0
   Cd13 3  1 (1,0,0) 2 (0,1,0) 3 (0,0,1)
   Cd15 3  4 (1,0,0) 5 (0,1,0) 6 (0,0,1)
   Se10 0
   Se12 0
   Se14 3  7 (1,0,0) 8 (0,1,0) 9 (0,0,1)
   Se16 3  10 (1,0,0) 11 (0,1,0) 12 (0,0,1)
   Cd17 3  1 (1,0,0) 2 (0,1,0) 3 (0,0,-1)
```

```
  Cd19 3  4 (1,0,0) 5 (0,1,0) 6 (0,0,-1)
  Se18 3  7 (1,0,0) 8 (0,1,0) 9 (0,0,-1)
  Se20 3  10 (1,0,0) 11 (0,1,0) 12 (0,0,-1)
 End
```

In this case, only the 4 atoms in the surface layers are allowed to move and the xyz and $xy\bar{z}$ symmetry is maintained.

Another example, with an explicit origin is given for the wurtzite structure unit cell of CdSe. In this case there is one degree of freedom in the structure which determines the separation (uc) of nearest neighbor Cd and S atoms along the c axis.

```
USESHELLS_WITHORIGIN 1
  Cd1 0
  Cd2 0
  Se1 1  O_FRAC (0.3333333,-0.3333333,0.0) 1 (0,0,1)
  Se2 1  O_FRAC (-0.3333333,0.3333333,0.5) 1 (0,0,1)
End
```

In this case, the generalized parameter $C_p \equiv C_1 = uc$.

We have tried this approach for several types of crystal structures and found it to be helpful. One could imagine other generalized coordinate mappings which would involve more complicated data structures but which might also be useful.

A more complicated example is the quartz-like form of $FePO_4$ with lattice translation vectors

$$\mathbf{T_1} = a\left(\frac{\sqrt{3}}{2}\hat{\mathbf{x}} - \frac{1}{2}\hat{\mathbf{y}}\right), \tag{6}$$

$$\mathbf{T_2} = a\hat{\mathbf{y}}, \tag{7}$$

$$\mathbf{T_3} = c\hat{\mathbf{z}}, \tag{8}$$

with a and c lattice constants. The corresponding fractional positions file takes the form:

```
#6/21/05 -- Using experimental structure as listed in
#Ping's lapw run -- GGACR10NEW.struct
#
Atom_List FRAC_POSITION
 Fe1    Fe       0.46744520  0.0         0.33333333
 Fe2    Fe       0.0         0.46744520  0.66666667
 Fe3    Fe       0.53255480  0.53255480  0.00000000

 P1     P        0.46843155  0.0         0.83333333
 P2     P        0.0         0.46843155  0.16666667
```

```
 P3      P         0.53156845  0.53156845  0.50000000

 O11     O         0.42006151  0.29910885  0.40171601
 O12     O         0.70089115  0.12095266  0.73504934
 O13     O         0.29910885  0.42006151  0.59828399
 O14     O         0.87904734  0.57993849  0.06838268
 O15     O         0.57993849  0.87904734  0.93161732
 O16     O         0.12095266  0.70089115  0.26495066

 O21     O         0.41763934  0.24615371  0.88250761
 O22     O         0.75384629  0.17148563  0.21584094
 O23     O         0.24615371  0.41763934  0.11749239
 O24     O         0.82851437  0.58236066  0.54917428
 O25     O         0.58236066  0.82851437  0.45082572
 O26     O         0.17148563  0.75384629  0.78415906
End
```

The corresponding shell structure file is given by:

```
USESHELLS_WITHORIGIN 8
 Fe1  1  O_FRAC (0.0, 0.0, 0.3333333333333) 1 (0.866025403,-0.5,0)
 Fe2  1  O_FRAC (0.0, 0.0, 0.6666666666667) 1 (0,1,0)
 Fe3  1  O_FRAC (1.0, 1.0, 0.0) 1 (-0.866025403,-0.5,0)

 P1   1  O_FRAC (0.0, 0.0 ,0.8333333333333) 2 (0.866025403,-0.5,0)
 P2   1  O_FRAC (0.0, 0.0 ,0.1666666666667) 2 (0,1,0)
 P3   1  O_FRAC (1.0, 1.0, 0.5) 2(-0.866025403,-0.5,0)

 O11  3  O_FRAC (0.0, 0.0, 0.0) 3 (0.866025403,-0.5,0) 4 (0,1,0) 5 (0,0,1)
 O12  3  O_FRAC (1.0, 0.0, 0.3333333333333) 3 (0,1,0) 4 (-0.866025403,-0.5,0) 5 (0,0,1)
 O13  3  O_FRAC (0.0, 0.0, 1.0) 3 (0,1,0) 4 (0.866025403,-0.5,0) 5 (0,0,-1)
 O14  3  O_FRAC (1.0, 1.0,-0.3333333333333) 3 (-0.866025403,-0.5,0) 4 (0.866025403,-0.5,0) 5 (0,0,1)
 O15  3  O_FRAC (1.0, 1.0, 1.3333333333333) 3 (-0.866025403,-0.5,0) 4 (0,1,0) 5 (0,0,-1)
 O16  3  O_FRAC (0.0, 1.0, 0.6666666666667) 3 (0.866025403,-0.5,0) 4 (-0.866025403,-0.5,0) 5 (0,0,-1)

 O21  3  O_FRAC (0.0, 0.0, 0.0) 6 (0.866025403,-0.5,0) 7 (0,1,0) 8 (0,0,1)
 O22  3  O_FRAC (1.0, 0.0,-0.6666666666667) 6 (0,1,0) 7 (-0.866025403,-0.5,0) 8 (0,0,1)
 O23  3  O_FRAC (0.0, 0.0, 1.0) 6 (0,1,0) 7 (0.866025403,-0.5,0) 8 (0,0,-1)
 O24  3  O_FRAC (1.0, 1.0,-0.3333333333333) 6 (-0.866025403,-0.5,0) 7 (0.866025403,-0.5,0) 8 (0,0,1)
 O25  3  O_FRAC (1.0, 1.0, 1.3333333333333) 6 (-0.866025403,-0.5,0) 7 (0,1,0) 8 (0,0,-1)
 O26  3  O_FRAC (0.0, 1.0, 1.6666666666667) 6 (0.866025403,-0.5,0) 7 (-0.866025403,-0.5,0) 8 (0,0,-1)
End
```