Data structure of OpenMX and ADPACK

- Structure of ADPACK
- Solving the 1D Dirac equation
- Pseudopotentials
- Localized basis functions
- Structure of OpenMX
- Input_std()
- Total energy
- Data structure for parallelization

Taisuke Ozaki (Univ. of Tokyo)

OpenMX developer's meeting, Dec. 21-22, 2015

What is ADPACK?

ADPACK (Atomic Density functional program PACKage) is a software to perform density functional calculations for a single atom

The features are listed below:

- All electron calculation by the Schrödinger or Dirac equation
- LDA and GGA treatment to exchang-correlation energy
- Finite element method (FEM) for the Schrödinger equation
- Pseudopotential generation by the TM, BHS, MBK schemes
- Pseudopotential generation for unbound states by Hamann's scheme
- Kleinman and Bylander (KB) separable pseudopotential
- Separable pseudopotential with Blöchl multiple projectors
- Partial core correction to exchange-correlation energy
- Logarithmic derivatives of wave functions
- Detection of ghost states in separable pseudopotentials
- Scalar relativistic treatment
- Fully relativistic treatment with spin-orbit coupling
- Generation of pseudo-atomic orbitals under a confinement potential

The pseudopotentials and pseudo-atomic orbitals can be the input data for OpenMX.

ADPACK is freely available

Welcome to OpenMX

Open source package for Material eXplorer

Google™ Custom Search

Contents

- What's new
  - The 1st OpenMX developer's meeting
  - Patch (Ver.3.7.10) to OpenMX Ver. 3.7
- What is OpenMX?
- Download
- Manual
- Technical Notes
- Video Lectures
- Publications
- OpenMX Forum
- Workshop
- Database of VPS and PAO
  - Ver. 2013
- ADPACK
- Miscellaneous informations
- Contributors
- Acknowledgment
- Opening positions
- Links

ADPACK can be available under GNU-GPL.

http://www.openmx-square.org/

Programs of ADPACK

Programs: 65 C rounties and 5 header files (50,000 lines)

Link: LAPACK and BLAS

Main routine: adpack.c

Input: readfile.c, Inputtool.c

Output: Output.c

All electron calculations: All_Electron.c, Initial_Density.c, Core.c

Numerical solutions for Schroedinger and Dirac eqs.: Hamming_I.c, Hamming_O.c

Density: Density.c, Density_PCC.c, Density_V.c

Exchange-Correlation: XC_CA.c, XC_EX.c, XC_PW91.c, XC_VWN.c, XC_PBE.c

Mixing: Simple_Mixing.c

Pseudopotentials: MBK.c, BHS.c, TM.c

Pseudo-atomic orbitals: Multiple_PAO.c

The global variables are declared in adpack.h.

1D-Dirac equation with a spherical potential

1-dimensional radial Dirac equation for the majority component G is given by

$$\left[\frac{1}{2m_{nlj}(r)}\left(\frac{d^2}{dr^2}+\frac{\alpha^2}{2m_{nlj}(r)}\frac{dv_{\rm eff}}{dr}\frac{d}{dr}+\frac{\alpha^2}{2m_{nlj}(r)}\frac{\kappa}{r}\frac{dv_{\rm eff}}{dr}-\frac{\kappa(\kappa+1)}{r^2}\right)+\varepsilon_{nlj}-v_{\rm eff}\right]G_{nlj}=0,$$

The mass term is given by

$$m_{nlj}(r) = 1+\frac{\alpha^2(\varepsilon_{nlj}-v_{\rm eff})}{2}.$$

Minority component

$$F_{nlj}=\frac{\left(\frac{d}{dr}+\frac{\kappa}{r}\right)G_{nlj}}{\alpha\left[\frac{2}{a^2}+\varepsilon_{nlj}-v_{\rm eff}(r)\right]F_{nlj}}.$$

By expressing the function G by the following form,

$$G_{nlj}(r)=r^{l+1}L_{nlj}(r).$$

One obtain a set of equations:

$$\frac{dL_{nlj}}{dx} = M_{nlj},$$

$$\frac{dM_{nlj}}{dx} = -(2l+1+\frac{r\alpha^2}{2m_{nlj}}\frac{dv_{\rm eff}}{dr})M_{nlj}-\frac{r\alpha^2}{2m_{nlj}}\frac{dv_{\rm eff}}{dr}(l+1+\kappa)L_{nlj}+2m_{nlj}r^2(v_{\rm eff}-\varepsilon_{nlj})L_{nlj}.$$

The charge density is obtained from

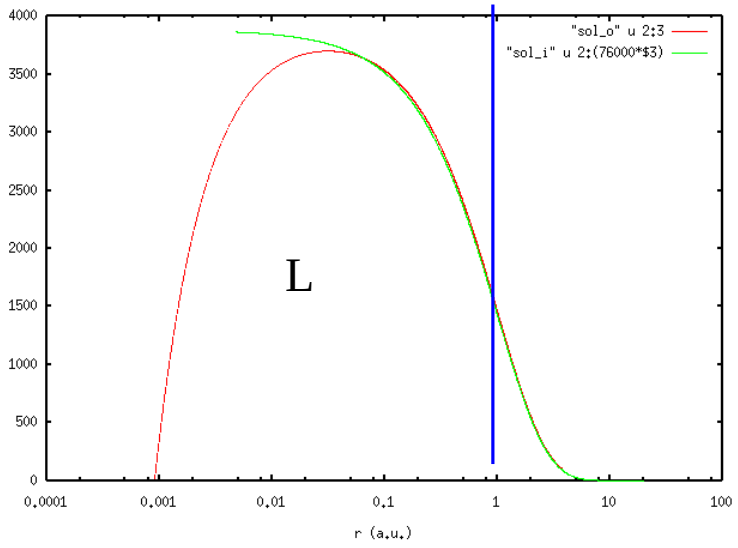
$$\rho(r) = \sum_{n,l,j} q_{nlj}\frac{G_{nlj}(r)^2+F_{nlj}(r)^2}{4\pi r^2},$$

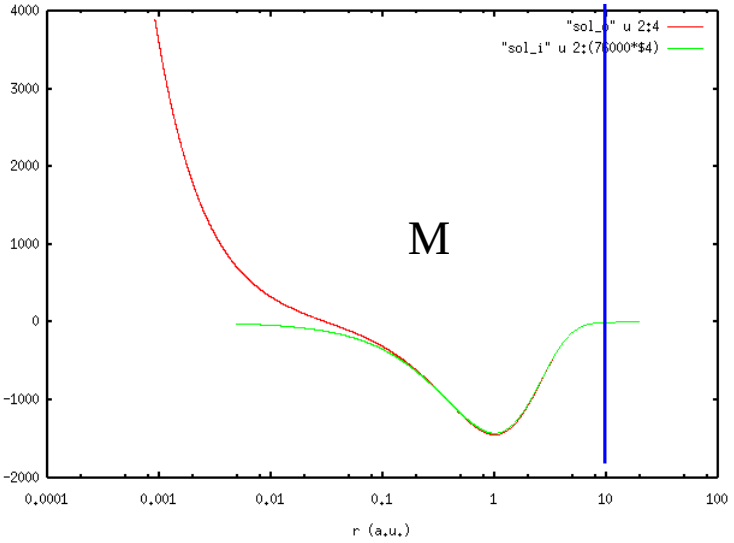
Solving the 1D-Dirac equation

By changing the varial r to x wit $r = e^x$, and applying a predictor and corrector method, we can derive the following equations:

$$
\begin{aligned}
L_{i+1}^{(p)} &= 32L_i - 31L_{i-1} - dx(16M_i + 14M_{i-1}) + dx^2(4M_i' - 2M_{i-1}'), \\
M_{i+1}^{(p)} &= -4M_i + 5M_{i-1} + dx(4M_i' + 2M_{i-1}'), \\
M_{i+1}^{(c)} &= M_i + \frac{dx}{12}(8M_i' - M_{i-1}' + 5M_{i+1}'^{(p)}), \\
L_{i+1}^{(c)} &= L_i + \frac{dx}{12}(8M_i - M_{i-1} + 5M_{i+1}^{(c)}).
\end{aligned}
$$

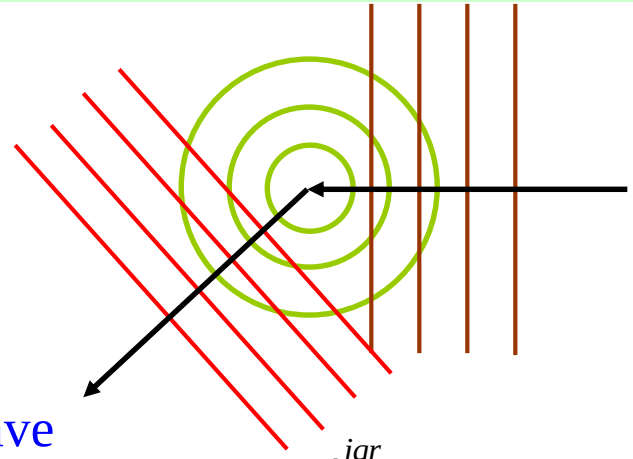
For a given E, the L and M are solved from the origin and distant region, and they are matched at a matching point.





In All_Electron.c, the calculation is performed.

Scattering by a spherical potential



Incident wave

$$e^{i\mathbf{q}\cdot\mathbf{r}}$$

$$\varepsilon = \frac{1}{2}q^2$$

Scattered wave

$$\psi_{\text{sca}}(\varepsilon, r) = e^{i\mathbf{q}\cdot\mathbf{r}} + \frac{e^{iqr}}{qr}\sum_l (2l+1)e^{i\eta_l}\sin\eta_l P_l(\cos\theta)$$

Phase shift η_l

$$\tan\eta_l(\varepsilon, r_0) = \frac{r_0 \frac{d}{dr} j_l(kr)|_{r_0} - D_l(\varepsilon) j_l(kr_0)}{r_0 \frac{d}{dr} n_l(kr)|_{r_0} - D_l(\varepsilon) n_l(kr_0)}$$

Logarithmic derivative of ψ

$$D_l(\varepsilon, r) = r\frac{d}{dr}\ln\psi_l(r)$$

$$\frac{\partial}{d\varepsilon} D_l(\varepsilon, r)|_{r_0} = -\frac{2}{r_0 \psi_l^{\,2}(r_0)} \int_0^{r_0} dr r^2 |\psi_l(r)|^2$$

If the norm of pseudized wave is conserved within r_0 and the logarithmic derivative coincides with that for the all electron case, the phase shift coincides with the all electron case to first order.

Norm-conserving pseudopotential by Troullier and Matins

N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).

$$\left[-\frac{1}{2}\frac{d^2}{dr^2}+\frac{l(l+1)}{2r^2}+V(r)\right]u_l(r)=\varepsilon_l u_l(r) \qquad R_l(r)=\frac{u_l(r)}{r}$$

For $u_l(r)$, the following form is used.

$$u_l(r)=\begin{cases} u_l^{(\mathrm{AE})}(r) & r\geq r_{cl} \\ r^{l+1}\exp[p(r)] & r\leq r_{cl}\end{cases} \qquad p(r)=\sum_{i=0}^{6}c_{2i}r^{2i}$$

Putting u_l into radial Schroedinger eq. and solving it with respect to V, we have

$$\begin{aligned} V^{(\mathrm{scr})}(r) &= \varepsilon_l-\frac{l(l+1)}{2r^2}+\frac{1}{2}\frac{[u_l(r)]''}{u_l(r)} \\ &= \varepsilon_l+\frac{l(l+1)p'(r)}{r}+\frac{1}{2}\left[p''(r)+[p'(r)]^2\right]\end{aligned}$$

$c_0 \sim c_{12}$ are determined by the following conditions:

- Norm-conserving condition within the cutoff radius
- The second derivatives of $V^{(scr)}$ is zero at r=0
- Equivalence of the derivatives up to 4^{th} orders of u_l at the cutoff radius

Ultrasoft pseudopotential by Vanderbilt

D. Vanderbilt, PRB 41, 7892 (1990).

The phase shift is reproduced around multiple reference energies by the following non-local operator.

$$V_{\mathrm{NL}} = \sum_{i,j} B_{ij} |\beta_i\rangle\langle\beta_j|$$

$$|\chi_i\rangle = V_{\mathrm{NL}}^{(i)} |\phi_i\rangle = (\varepsilon_i - T - V_{\mathrm{loc}})|\phi_i\rangle$$

$$B_{ij} = \langle\phi_i|\chi_j\rangle$$

$$|\beta_i\rangle = \sum_j (B^{-1})_{ji} |\chi_j\rangle$$

If the following generalized norm conserving condition is fulfilled, the matrix B is Hermitian. Thus, in the case the operator V_{NL} is also Hermitian.

$$Q_{ij} = \langle\psi_i|\psi_j\rangle_R - \langle\phi_i|\phi_j\rangle_R$$

$$B_{ij} - B^*_{ji} = (\varepsilon_i - \varepsilon_j) Q_{ij}$$

If Q=0, then B-B*=0

How the non-local operator works?

Operation of the non-local operator to pseudized wave function

$$
\begin{aligned}
\hat{v}^{(\mathrm{NL})}|\phi_k^{(\mathrm{PS})}\rangle &= \sum_{ij}|\beta_i\rangle B_{ij}\langle\beta_j|\phi_k^{(\mathrm{PS})}\rangle \\
&= \sum_{ij}|\beta_i\rangle B_{ij}\sum_{k'}\left(B^{-1}\right)_{k'j}\langle\chi_{k'}|\phi_k^{(\mathrm{PS})}\rangle, \\
&= \sum_{ij}|\beta_i\rangle B_{ij}\sum_{k'}\left(B^{-1}\right)_{k'j}B_{kk'}, \\
&= \sum_{ij}|\beta_i\rangle B_{ij}\delta_{kj}, \\
&= \sum_{i}\left(\sum_{j}\left(B^{-1}\right)_{ji}|\chi_j\rangle\right)B_{ik}. \\
&= |\chi_k\rangle
\end{aligned}
$$

Note that

$$v^{(\mathrm{SL})}(r) = v_{\mathrm{L}}(r) + v_{\mathrm{H}}^{(\mathrm{v})}(r) + v_{\mathrm{xc}}^{(\mathrm{v+pcc})}(r).$$

$$|\chi_i\rangle = \left(\varepsilon_i + \frac{1}{2}\nabla^2 - v^{(\mathrm{SL})}(r)\right)|\phi_i^{(\mathrm{PS})}\rangle,$$

$$B_{ij} = \langle\phi_i^{(\mathrm{PS})}|\chi_j\rangle$$

$$|\beta_i\rangle = \sum_j\left(B^{-1}\right)_{ji}|\chi_j\rangle.$$

It turns out that the following Schroedinger equation is satisfied.

$$\left(-\frac{1}{2}\nabla^2 + v^{(\mathrm{SL})}(r) + \hat{v}^{(\mathrm{NL})}\right)|\phi_i^{(\mathrm{PS})}\rangle = \varepsilon_i|\phi_i^{(\mathrm{PS})}\rangle.$$

The matrix B and the generalized norm conserving condition

The matrix B is given by

$$B_{ij} = \int_0^{r_c} dr P_i^{(\mathrm{PS})}(r) \left(\varepsilon_j + \frac{1}{2}\frac{d^2}{dr^2} - \frac{l(l+1)}{2r^2} - v^{(\mathrm{SL})}(r) \right) P_j^{(\mathrm{PS})}(r),$$

$$B_{ji}^* = \int_0^{r_c} dr P_i^{(\mathrm{PS})}(r) \left(\varepsilon_i + \frac{1}{2}\frac{d^2}{dr^2} - \frac{l(l+1)}{2r^2} - v^{(\mathrm{SL})}(r) \right) P_j^{(\mathrm{PS})}(r),$$

Thus, we have

$$B_{ij} - B_{ji}^* = (\varepsilon_j - \varepsilon_i) \int_0^{r_c} dr P_i^{(\mathrm{PS})}(r) P_j^{(\mathrm{PS})}(r)$$
$$+\frac{1}{2}\int_0^{r_c} dr P_i^{(\mathrm{PS})}(r) P_j''^{(\mathrm{PS})}(r) - \frac{1}{2}\int_0^{r_c} dr P_i''^{(\mathrm{PS})}(r) P_j^{(\mathrm{PS})}(r).$$

By integrating by parts

$$B_{ij} - B_{ji}^* = (\varepsilon_j - \varepsilon_i)\langle \phi_i^{(\mathrm{PS})} | \phi_j^{(\mathrm{PS})} \rangle_{r_c} + \frac{1}{2}\left[P_i^{(\mathrm{PS})}(r) P_j'^{(\mathrm{PS})}(r) \right]_0^{r_c} - \frac{1}{2}\left[P_i'^{(\mathrm{PS})}(r) P_j^{(\mathrm{PS})}(r) \right]_0^{r_c},$$
$$= (\varepsilon_j - \varepsilon_i)\langle \phi_i^{(\mathrm{PS})} | \phi_j^{(\mathrm{PS})} \rangle_{r_c} + \frac{1}{2} P_i^{(\mathrm{PS})}(r_c) P_j'^{(\mathrm{PS})}(r_c) - \frac{1}{2} P_i'^{(\mathrm{PS})}(r_c) P_j^{(\mathrm{PS})}(r_c). \quad \cdots (1)$$

By performing the similar calculations, we obtain for the all electron wave functions

$$0 = (\varepsilon_j - \varepsilon_i)\langle \phi_i^{(\mathrm{AE})} | \phi_j^{(\mathrm{AE})} \rangle_{r_c} + \frac{1}{2} P_i^{(\mathrm{AE})}(r_c) P_j'^{(\mathrm{AE})}(r_c) - \frac{1}{2} P_i'^{(\mathrm{AE})}(r_c) P_j^{(\mathrm{AE})}(r_c). \quad \cdots (2)$$

By subtracting (2) from (1), we have the following relation.

$$B_{ij} - B_{ji}^* = (\varepsilon_i - \varepsilon_j)\left(\langle \phi_i^{(\mathrm{AE})} | \phi_j^{(\mathrm{AE})} \rangle_{r_c} - \langle \phi_i^{(\mathrm{PS})} | \phi_j^{(\mathrm{PS})} \rangle_{r_c} \right).$$

Norm-conserving pseudopotential by MBK

I. Morrion, D.M. Bylander, and L. Kleinman, PRB 47, 6728 (1993).

If $Q_{ij} = 0$, the non-local operator can be transformed to a diagonal form.

$$V_{\mathrm{NL}} = \sum_{i,j} B_{ij} |\beta_i\rangle\langle\beta_j|,$$
$$= \sum_{i} \lambda_i |\alpha_i\rangle\langle\alpha_i|$$

The form is exactly the same as that for the Blochl expansion, resulting in no need for modification of OpenMX.

To satisfy $Q_{ij}=0$, the pseudized wave function is written by

$$\phi_i = \phi_{\mathrm{TM},i} + f_i \qquad f_i = \sum_{i=0} c_i \left[r j_l \left(\frac{r}{r_c} u_{li} \right) \right]$$

The coefficients can be determined by matching up to the third derivatives to those for the all electron, and $Q_{ij}=0$. Once c's are determined, χ is given by

$$\chi_i = V_{\mathrm{TM}}^{(i)} \phi_{\mathrm{TM},i} + \varepsilon_i f_i - V_{\mathrm{loc}} \phi_i - \frac{1}{2} \sum_i c_i \left(\frac{u_{li}}{r_c} \right)^2 \left[r j_l \left(\frac{r}{r_c} u_{li} \right) \right]$$

The form of MBK pseudopotentials

The pseudopotential is given by the sum of a local term V_{loc} and non-local term V_{NL}.

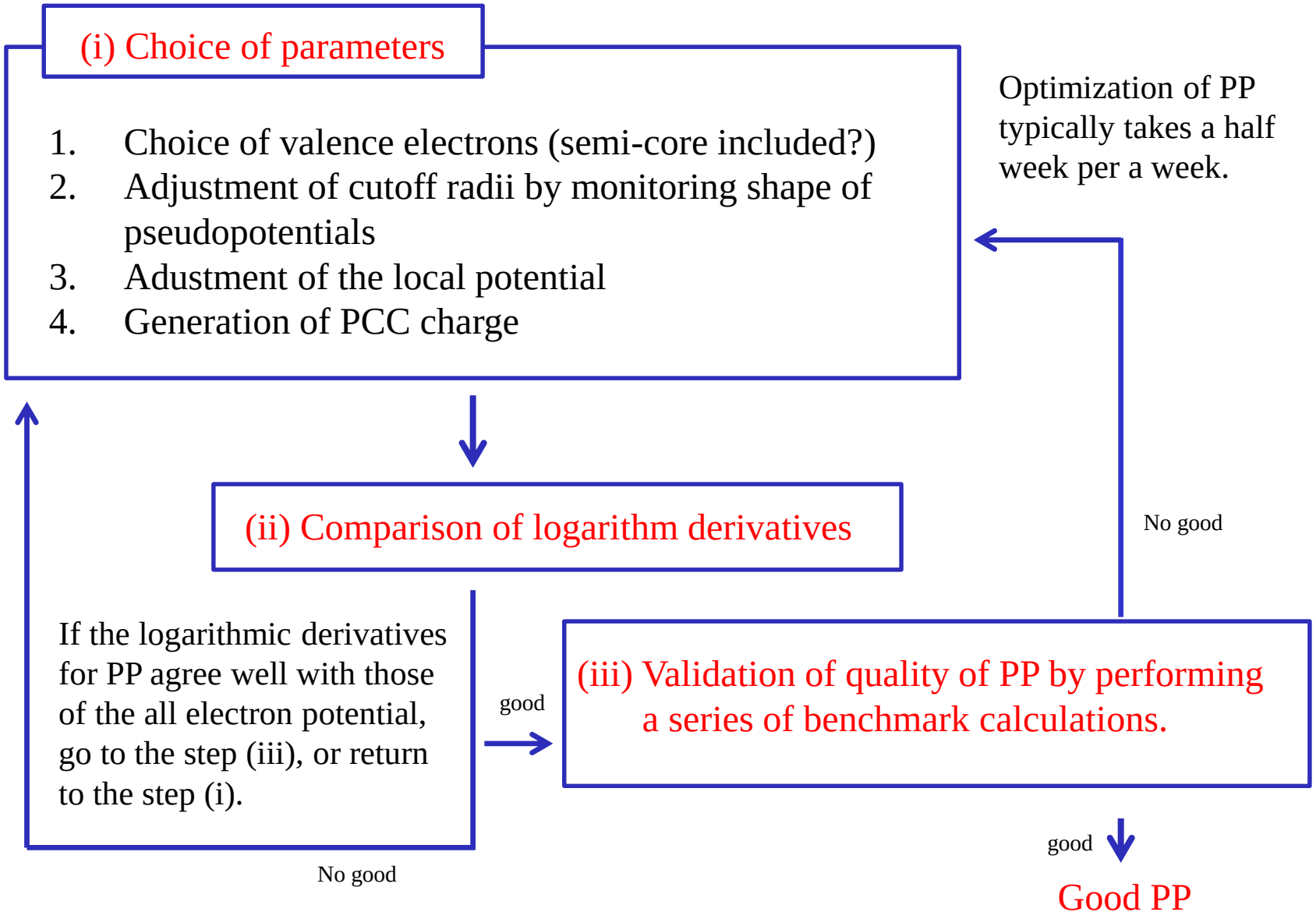
$$V^{(\mathrm{ps})} = V_{\mathrm{loc}}(r) + V_{\mathrm{NL}}$$

The local term V_{loc} is independent of the angular channel l. On the other hand, the non-local term V_{NL} is given by projectors

$$V_{\mathrm{NL}} = \sum_i \lambda_i \, |\alpha_i\rangle\langle\alpha_i|$$

The projector consists of radial and spherical parts, and depends on species, radial and l-channels.

Optimization of pseudopotentials



Database (2013)

Database (2013) of optimized VPS and PAO

The database (2013) of fully relativistic pseudopotentials (VPS) and pseudo-atomic orbitals (PAO), generated by ADPACK, which could be an
The data of elements with the underline are currentl
VPS and PAO, in the program package, OpenMX, t
openmx*.*/DFT_DATA13/VPS/ and openmx*.*/DFT
The delta factor of OpenMX with the database (201

Fully relativistic pseudopotentials

Fully relativistic pseudopotentials generated by the M
(PBE13) which contain a partial core correction and

- Pt_CA13.vps
- Pt_PBE13.vps

Optimized VPS and PAO are available, which includes relativistic effect with spin-orbit coupling.

E																	
H																	
Li	Be																
Na	Mg																
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba	L	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra	A															
	L	La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu	
	A	Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr	

http://www.openmx-square.org/

Close look at “vps” files #1

Input file

```
*************************************************
                  Input file
*************************************************

#
# File Name
#

System.CurrrentDir    ./              # default=./
System.Name            Pt_PBE13
Log.print             OFF             # ON|OFF

System.UseRestartfile  yes            # NO|YES, default=NO
System.Restartfile     Pt_PBE13       # default=null

#
# Calculation type
#

eq.type               dirac           # sch|sdirac|dirac
calc.type             vps             # ALL|VPS|PAO
xc.type               GGA             # LDA|GGA

#
# Atom
#

AtomSpecies             78
max.occupied.N           6
total.electron        78.0
valence.electron      16.0
<occupied.electrons
 1   2.0
 2   2.0  6.0
 3   2.0  6.0  10.0
 4   2.0  6.0  10.0 14.0
 5   2.0  6.0   9.0  0.0  0.0
 6   1.0  0.0   0.0  0.0  0.0  0.0
occupied.electrons>
```

In the header part, the input file for the ADPACK calculations are shown, which maybe helpful for the next generation of pseudopotentials.

Close look at “vps” files #2

Eigenvalues for all electron calculation

```
*****************************************************
 Eigenvalues (Hartree) in the all electron calculation
*****************************************************

                              j=l+1/2                 j=l-1/2
 n=  1  l=  0     -2868.8969439503935     -2868.8969439503935
 n=  2  l=  0      -503.1142921617339      -503.1142921617339
 n=  2  l=  1      -419.1546670411121      -482.3720670651586
 n=  3  l=  0      -118.0771871048129      -118.0771871048129
 n=  3  l=  1       -94.8406495857986      -108.7310226714485
 n=  3  l=  2       -76.1768324508852       -79.1659107610239
 n=  4  l=  0       -25.3345514773071       -25.3345514773071
 n=  4  l=  1       -18.0570286040002       -21.3625609002204
 n=  4  l=  2       -10.9124371484223       -11.5257406345024
 n=  4  l=  3        -2.4568191349790        -2.5821360560534
 n=  5  l=  0        -3.6982650446879        -3.6982650446879
 n=  5  l=  1        -1.8910921655761        -2.4338351652662
 n=  5  l=  2        -0.2019516934267        -0.2496588022815
 n=  6  l=  0        -0.2079456213222        -0.2079456213222
```

The eigenvalues with $j=l\pm1/2$ for the all electron calculations by the Dirac equation are included, which can be used to estimate the splitting by spin-orbit coupling

Close look at “vps” files #3

Information for pseudopotentials

```
vps.type                    MBK
number.vps                   5
<pseudo.NandL
  0     5    1    1.1000   0.0
  1     5    2    1.7000   0.0
  2     6    0    2.0000   0.0
  3     6    1    2.8000   0.0
  4     7    0    2.9000  -0.1
pseudo.NandL>
```

The specification for the pseudopotentials is made by vps.type, number.vps, and pseudo.NandL.

The project energies λ is shown as follows:

```
<project.energies
 7
 0     1.000000000000000e+00    1.000000000000000e+00
 0     1.000000000000000e+00    1.000000000000000e+00
 1    -1.000000000000000e+00   -1.000000000000000e+00
 1    -1.000000000000000e+00   -1.000000000000000e+00
 2    -1.000000000000000e+00   -1.000000000000000e+00
 2    -1.000000000000000e+00   -1.000000000000000e+00
 2    -1.000000000000000e+00   -1.000000000000000e+00
project.energies>
```

Close look at “vps” files #4

The generated pseudopotentials are output by Pseudo.Potentials

```
<Pseudo.Potentials
-1.00000000000000e+01 4.53999297624849e-05  -1.56274493573701e+01
1.29016191735297e+00  -3.53576793162279e-04 -2.74501529071506e-04
-2.44604572903325e-08  4.46045901596636e-08 5.09718887954710e-08
-9.97354909819639e+00 4.66168218439265e-05  -1.56274493573701e+01
1.29016191758754e+00  -3.63054006042743e-04 -2.81859221925009e-04
-2.57892987562848e-08  4.70277839806157e-08 5.37409931750746e-08
-9.94709819639279e+00 4.78663312960476e-05  -1.56274493573701e+01
1.29016191783448e+00  -3.72785244541990e-04 -2.89414129143262e-04
-2.71903310082123e-08  4.95826204925540e-08 5.66605322202532e-08
-9.92064729458918e+00 4.91493323893655e-05  -1.56274493573701e+01
1.29016191809451e+00  -3.82777317519879e-04 -2.97171536827178e-04
-2.86674758905822e-08  5.22762513305807e-08 5.97386784609548e-08
-9.89419639278557e+00 5.04667228281981e-05  -1.56274493573701e+01
1.29016191836861e+00  -3.93037216339845e-04 -3.05136872765505e-04
-3.02248683064018e-08  5.51162166502627e-08 6.29840484088342e-08
-9.86774549098196e+00 5.18194243787784e-05  -1.56274493573701e+01
1.29016191865760e+00  -4.03572119760603e-04 -3.13315710232509e-04
```

1st column: x

2nd column: r=exp(x) in a.u.

3rd column: radial part of local pseudopotential

4th and later columns: radial part of non-local pseudopotentials.

Close look at “vps” files #5

Charge density for partial core correction

```
<density.PCC
-1.0000000000000e+01  4.5399929762484 9e-05  6.8411607762813 6e-01
-9.9735490981963 9e+00  4.6616821843926 5e-05  6.8411607762813 6e-01
-9.9470981963927 9e+00  4.7866331296047 6e-05  6.8411607762813 6e-01
-9.9206472945891 8e+00  4.9149332389365 5e-05  6.8411607762813 6e-01
-9.8941963927855 7e+00  5.0466722828198 1e-05  6.8411607762813 6e-01
-9.8677454909819 6e+00  5.1819424378778 4e-05  6.8411607762813 6e-01
-9.8412945891783 6e+00  5.3208383514206 6e-05  6.8411607762813 6e-01
-9.8148436873747 5e+00  5.4634572076688 9e-05  6.8411607762813 6e-01
-9.7883927855711 4e+00  5.6098987957526 1e-05  6.8411607762813 6e-01
-9.7619418837675 4e+00  5.7602655795330 1e-05  6.8411607762813 6e-01
-9.7354909819639 3e+00  5.9146627692953 5e-05  6.8411607762813 6e-01
-9.7090400801603 2e+00  6.0731983953636 7e-05  6.8411607762813 6e-01
-9.6825891783567 1e+00  6.2359833836887 0e-05  6.8411607762813 6e-01
-9.6561382765531 0e+00  6.4031916233406 9e-05  6.8411607762813 6e-01
```

1st column: x

2nd column: r=exp(x) in a.u.

3rd column: charge density for PCC

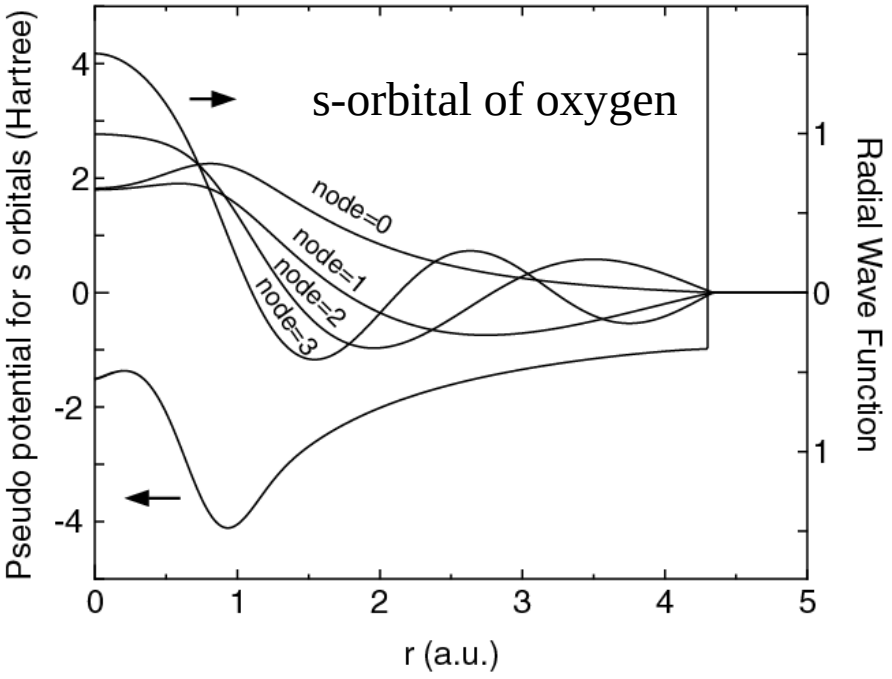
Primitive basis functions

1. Solve an atomic Kohn-Sham eq. under a confinement potential:

$$V_{\text{core}}(r)=\begin{cases} -\dfrac{Z}{r} & \text{for } r\leqslant r_1 \\ \displaystyle\sum_{n=0}^{3} b_n r^n & \text{for } r_1<r\leqslant r_c \\ h & \text{for } r_c<r, \end{cases}$$

2. Construct the norm-conserving pseudopotentials.

3. Solve ground and excited states for the the peudopotential for each L-channel.



In most cases, the accuracy and efficiency can be controlled by

Cutoff radius

Number of orbitals

PRB 67, 155108 (2003)
PRB 69, 195113 (2004)

Variational optimization of basis functions

One-particle wave functions

$$\psi_\mu(\mathbf{r}) = \sum_{i\alpha} c_{\mu,i\alpha}\phi_{i\alpha}(\mathbf{r}-\mathbf{r}_i)$$

Contracted orbitals

$$\phi_{i\alpha}(\mathbf{r}) = \sum_q a_{i\alpha q}\chi_{i\eta}(\mathbf{r})$$

The variation of E with respect to *c* with fixed *a* gives

$$\partial E_{\mathrm{tot}}/\partial c_{\mu,i\alpha} = 0 \quad \rightarrow \quad \sum_{j\beta} \langle \phi_{i\alpha}|\hat{H}|\phi_{j\beta}\rangle c_{\mu,j\beta} = \varepsilon_\mu \sum_{j\beta} \langle \phi_{i\alpha}|\phi_{j\beta}\rangle c_{\mu,j\beta}$$

Regarding c as dependent variables on *a* and assuming KS eq. is solved self-consistently with respect to *c*, we have

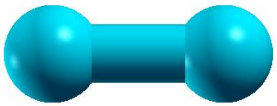
$$\frac{\partial E_{\mathrm{tot}}}{\partial a_{i\alpha q}} = \frac{\delta E_{\mathrm{tot}}}{\delta \rho(\mathbf{r})}\frac{\delta \rho(\mathbf{r})}{\delta a_{i\alpha q}}$$

$$= 2\sum_{j\beta} \left(\Theta_{i\alpha,j\beta}\langle \chi_{i\eta}|\hat{H}|\phi_{j\beta}\rangle - E_{i\alpha,j\beta}\langle \chi_{i\eta}|\phi_{j\beta}\rangle\right)$$

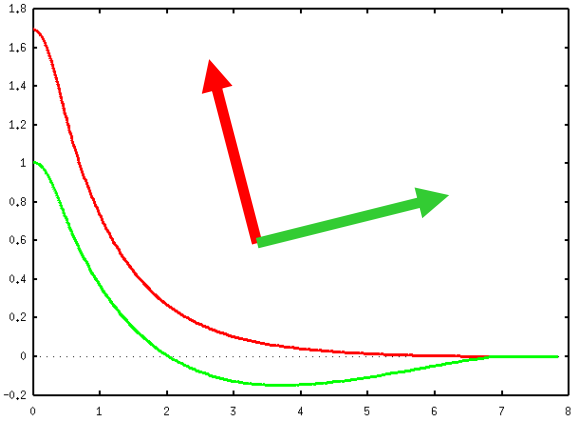
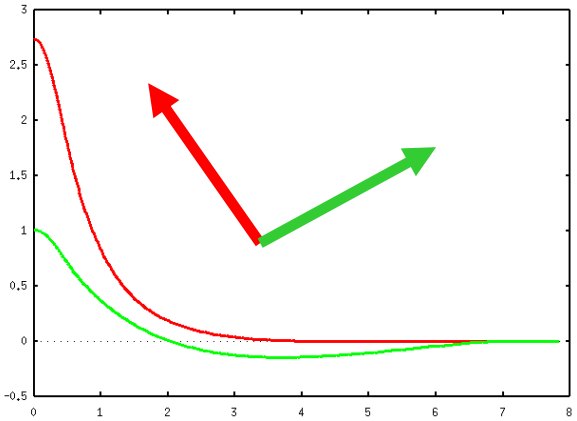
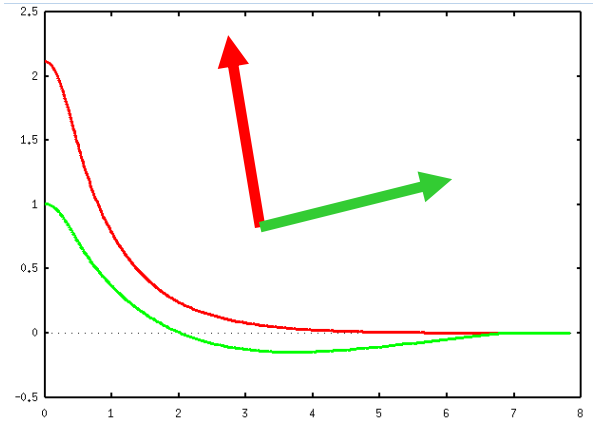
Ozaki, PRB 67, 155108 (2003)

Optimization of basis functions

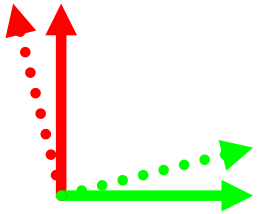
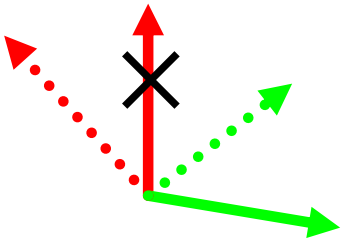
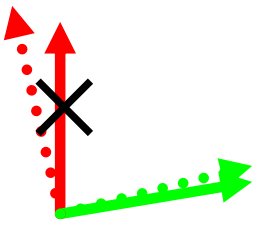
1. Choose typical chemical environments


2. Optimize variationally the radial functions

3. Rotate a set of optimized orbitals within the subspace, and discard the redundant funtions

Close look at “pao” files #1

Contraction coefficients

```
number.optpao  3

#
#  Pt_opt.dat, Pt7a_1.pao
#

<Contraction.coefficients1
  45
  Atom=  1  L= 0  Mul= 0  p=  0   0.999972378719334
  Atom=  1  L= 0  Mul= 0  p=  1   0.004243624172614
  Atom=  1  L= 0  Mul= 0  p=  2   0.001121755031727
  Atom=  1  L= 0  Mul= 0  p=  3  -0.001404518456681
  Atom=  1  L= 0  Mul= 0  p=  4   0.003486437962913
  Atom=  1  L= 0  Mul= 0  p=  5   0.002540946071787
  Atom=  1  L= 0  Mul= 0  p=  6   0.003333832045599
  Atom=  1  L= 0  Mul= 0  p=  7   0.001154904378998
  Atom=  1  L= 0  Mul= 0  p=  8   0.001380635454875
  Atom=  1  L= 0  Mul= 0  p=  9  -0.000385484804772
  Atom=  1  L= 0  Mul= 0  p= 10   0.000256086343292
  Atom=  1  L= 0  Mul= 0  p= 11  -0.000666201042052
  Atom=  1  L= 0  Mul= 0  p= 12   0.000286598921484
  Atom=  1  L= 0  Mul= 0  p= 13  -0.000378947534286
```

The optimized PAO are generally obtained by two or three calculations for orbital optimization. The contraction coefficients are attached in the header of the pao file.

Close look at “pao” files #2

Input file

```
************************************************
                  Input file
************************************************

#
# File Name
#

System.CurrrentDir    ./              # default=./
System.Name            Pt7.0p
Log.print             OFF             # ON|OFF

System.UseRestartfile  yes            # NO|YES, default=NO
System.Restartfile     Pt7.0p         # default=null

#
# Calculation type
#

eq.type              sdirac           # sch|sdirac|dirac
calc.type             pao             # ALL|VPS|PAO
xc.type               GGA             # LDA|GGA

#
# Atom
#

AtomSpecies             78
max.occupied.N           6
total.electron        78.0
valence.electron      16.0
<occupied.electrons
 1   2.0
 2   2.0  6.0
 3   2.0  6.0  10.0
 4   2.0  6.0  10.0 14.0
 5   2.0  6.0   9.0  0.0  0.0
 6   1.0  0.0   0.0  0.0  0.0  0.0
occupied.electrons>
```

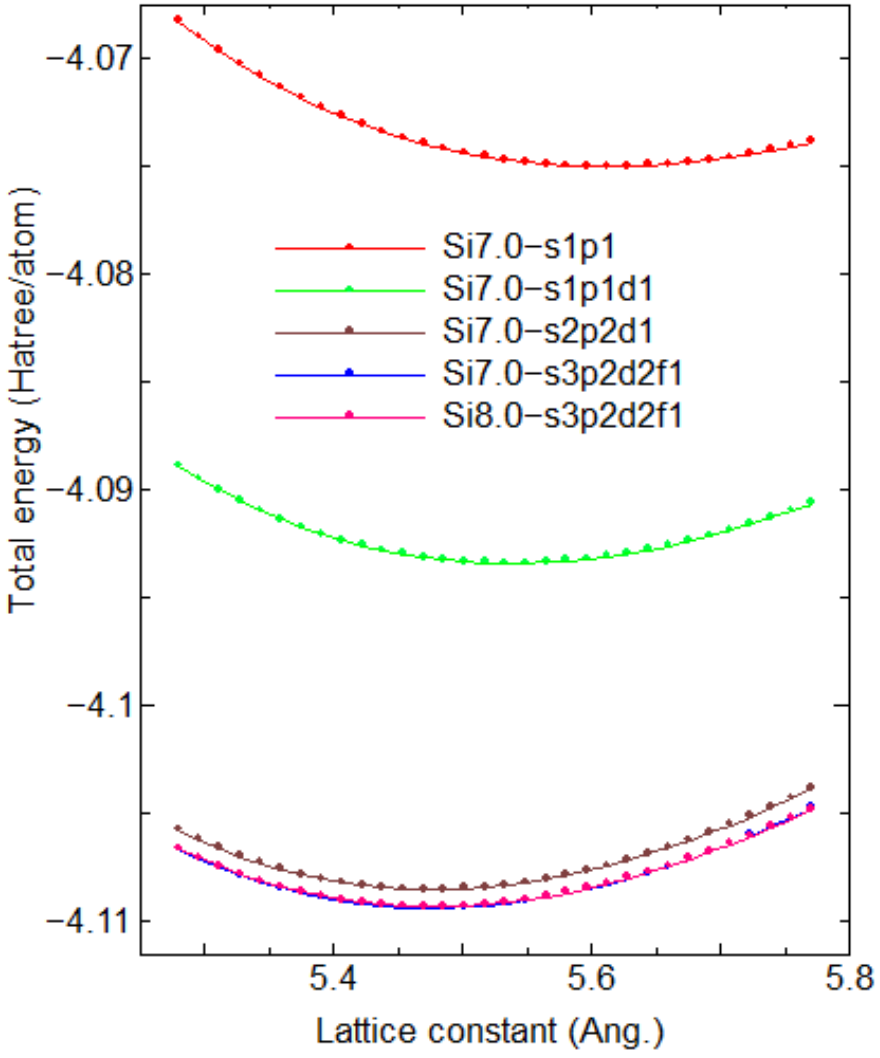
The input file for the ADPACK calculations is attached which can be helpful for checking the computational condition.

Close look at “pao” files #3

e.g., Si7.0.pao

```
Eigenvalues
Lmax= 3 Mul=15
 | mu    0  0     -0.59320968623145 1
 | mu    0  1      0.01654247265872 4
 | mu    0  2      0.57915688461708 8
 | mu    0  3      1.39915528592998
 | mu    0  4      2.43106703909053
 | mu    0  5      3.63498884739274
 | mu    0  6      4.99885012002783
 | mu    0  7      6.55485759476897
 | mu    0  8      8.33541526908920
 | mu    0  9     10.34055461939939
 | mu    0 10     12.55818051232040
 | mu    0 11     14.98369777887096
 | mu    0 12     17.62014479571136
 | mu    0 13     20.47011746372617
 | mu    0 14     23.53245309073866
 | mu    1  0     -0.31460013133101 2
 | mu    1  1      0.12937640798489 5
 | mu    1  2      0.71637380083701
 | mu    1  3      1.54488697995006
 | mu    1  4      2.59211686526084
 | mu    1  5      3.84687324239366
 | mu    1  6      5.31246180826158
 | mu    1  7      6.99509799702661
 | mu    1  8      8.89454075291723
 | mu    1  9     11.00602102880752
 | mu    1 10     13.32692005826443
 | mu    1 11     15.85857455673626
 | mu    1 12     18.60272353142246
 | mu    1 13     21.55894843635971
 | mu    1 14     24.72602323954447
 | mu    2  0      0.01886411574821 3
 | mu    2  1      0.35893514996065 7
 | mu    2  2      0.94629918754692
 | mu    2  3      1.76201765644983
 | mu    2  4      2.81418723624388
 | mu    2  5      4.10656012645961
 | mu    2  6      5.63661971875011
 | mu    2  7      7.39522693820483
 | mu    2  8      9.37222098331867
 | mu    2  9     11.56227937764802
 | mu    2 10     13.96620568880690
 | mu    2 11     16.58651812641581
 | mu    2 12     19.42296574188659
 | mu    2 13     22.47308081363278
 | mu    2 14     25.73538272482773
 | mu    3  0      0.28356769151846 6
 | mu    3  1      0.79082836114569
 | mu    3  2      1.52992065308726
 | mu    3  3      2.52537261496132
```

The eigenvalues of pseudized and confined states are shown.
The information can be used to choose basis functions step by step.



One may find that the structural properties can be obtained by Si7.0-s2p2d1.

Also, it is found that the cutoff of 7.0 (a.u) reaches the convergent result from the comparison between Si7.0-s3p2d2f1 and Si8.0-s3p2d2f1.

Close look at “pao” files #4

Valence density

```
<valence.charge.density
-10.000000000000000  0.000045399929762   0.001745659462571
-9.975841116916641   0.000046510097569   0.001745659561254
-9.951682233833282   0.000047647412391   0.001745659664823
-9.927523350749921   0.000048812538055   0.001745659773518
-9.903364467666561   0.000050006154622   0.001745659887594
-9.879205584583202   0.000051228958783   0.001745660007317
-9.855046701499841   0.000052481664263   0.001745660132967
-9.830887818416482   0.000053765002242   0.001745660264838
-9.806728935333123   0.000055079721779   0.001745660403236
-9.782570052249763   0.000056426590249   0.001745660548486
-9.758411169166404   0.000057806393792   0.001745660700926
-9.734252286083043   0.000059219937772   0.001745660860913
-9.710093402999684   0.000060668047247   0.001745661028819
-9.685934519916325   0.000062151567448   0.001745661205038
-9.661775636832964   0.000063671364278   0.001745661389980
```

1st column: x

2nd column: r=exp(x) in a.u.

3rd column: valence density for the chosen states

Close look at “pao” files #4

The generated radial functions are output by pseudo.atomic.orbitals.L=0,1,…

```
PAO.Lmax    4
PAO.Mul     15
<pseudo.atomic.orbitals.L=0
-10.000000000000000  0.000045399929762  0.166091619003741 -0.1610
1.60669396756814 4  1.612886733723762  1.837547017444685  2.242568
3.168493033883155  3.429512980423414  3.703471339309258
-9.975841116916641  0.000046510097569  0.166091619004696 -0.16107
1.606693967375123  1.612886733470841  1.837547017050928  2.242568
3.168493032006391  3.429512978059319  3.703471336381002
-9.951682233833282  0.000047647412391  0.166091619005698 -0.16107
1.60669396717254 8  1.612886733205399  1.837547016637681  2.242568
3.168493030036719  3.429512975578192  3.703471333307784
-9.92752335074992 1  0.000048812538055  0.166091619006750 -0.16107
1.606693966959944  1.612886732926817  1.837547016203976  2.242568
3.16849302796954 0  3.429512972974238  3.703471330082429
-9.903364467666561  0.000050006154622  0.166091619007854 -0.16107
1.60669396673681 5  1.612886732634444  1.837547015748801  2.242568
3.168493025800029  3.429512970241378  3.703471326697407
-9.879205584583202  0.000051228958783  0.166091619009013 -0.16107
1.606693966502641  1.612886732327596  1.837547015271093  2.242568
3.168493023523118  3.429512967373229  3.703471323144812
-9.855046701499841  0.000052481664263  0.166091619010229 -0.16107
1.606693966256874  1.612886732005560  1.837547014769735  2.242568
```

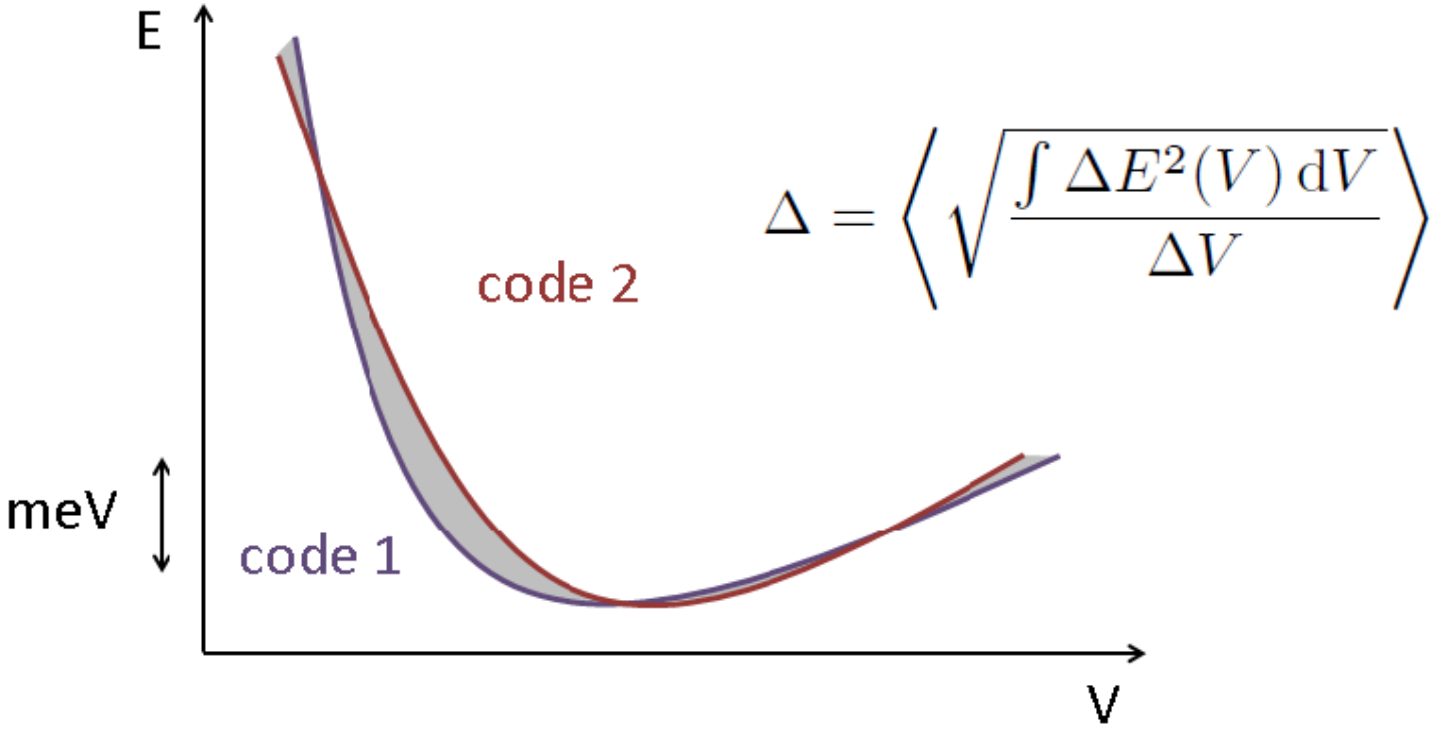
1st column: x

2nd column: r=exp(x) in a.u.

3rd and later columns: radial part of basis functions

Δ-factor

The delta factor is defined as difference of total energy between Wien2k (FLAPW+LO) and a code under testing, which is shown as shaded region in figure below, where the volume is changed by plus and minus 6 % taken from the equilibrium V_0 .



$$\Delta = \left\langle \sqrt{\frac{\int \Delta E^2(V)\,\mathrm{d}V}{\Delta V}} \right\rangle$$

Lejaeghere et al., Critical Reviews in Solid State and Materials Sciences 39, 1-24 (2014).

Comparison of codes in terms of Δ-factor

Code	Version	Basis	Potentials	Δ-factor	Authors
WIEN2k	13.1	LAPW/APW+lo	full	0 meV/atom	S. Cottenier
VASP	5.2.12	plane waves	PAW 2012	0.7 meV/atom	K. Lejaeghere
VASP	5.2.12	plane waves	PAW 2012 GW-ready	0.8 meV/atom	K. Lejaeghere
Abinit	7.5.3	plane waves	PAW JTH	1.2 meV/atom	F. Jollet *et al.* [3]
Abinit	7.1.2	plane waves	GPAW PAW 0.9 (40 Ha cut-off)	1.3 meV/atom	F. Jollet *et al.* [3]
GPAW	0.9.1	plane waves	PAW 0.9	1.5 meV/atom	ASE [2]
Quantum ESPRESSO	5.0.2	plane waves	PAW 0.3.1	1.8 meV/atom	Quantum ESPRESSO [5]
OpenMX	3.7	pseudo-atomic orbitals	Morrison-Bylander-Kleinman norm-conserving (2013)	2.0 meV/atom	OpenMX [4]
VASP	5.2.2	plane waves	PAW 2011	2.1 meV/atom	K. Lejaeghere *et al.* [1]
GPAW	0.8.0	grid-based	PAW 0.6	3.8 meV/atom	K. Lejaeghere *et al.* [1]
Dacapo	2.7.16	plane waves	Vanderbilt ultrasoft version 2	6.2 meV/atom	ASE [2]
Abinit	5.4.4p	plane waves	Troullier-Martins norm-conserving (FHI)	14.5 meV/atom	ASE [2]

http://molmod.ugent.be/deltacodesdft

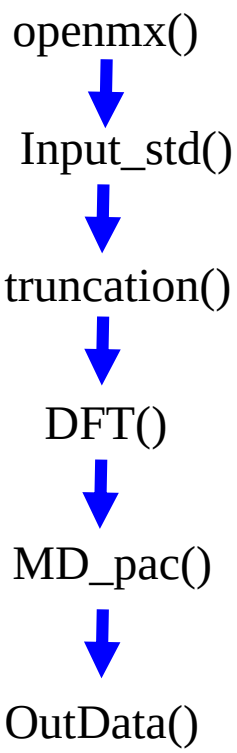
Structures of OpenMX

- Language: C, fortran90
- 265 sub files, about 1000 sub routines
- 21 header files
- About 300,000 lines
- Compilation by makefile
- Eigenvalue solver: ELPA included
- Linking of LAPACK, BLAS, FFTW3
- Hybrid parallelization by MPI,OpenMP

Basic structure

openmx_common.h

Common functions and common variables are declared in the header file.



Main computational flow

Reading input file

Analysis of geometrical structure, memory allocation, analysis of communication pattern

SCF, total energy, forces

Molecular dynamics and geometry optimization

Output

Input_std.c

The input file is analyzed in Input_std() which employs Inputtool.c. After searching keywords, the value after the keyword is set for the keyword. The variables for the keywords are declared in openmx_common.h.

```
/*************************************************************
                        open a file
*************************************************************/

if (input_open(file)==0){
  MPI_Finalize();
  exit(0);
}

input_string("System.CurrrentDirectory",filepath,"./");
input_string("System.Name",filename,"default");
input_string("DATA.PATH",DFT_DATA_PATH,"../DFT_DATA13");
input_int("level.of.stdout", &level_stdout,1);
input_int("level.of.fileout",&level_fileout,1);
input_logical("memory.usage.fileout",&memoryusage_fileout,0); /* default=off */

if (level_stdout<0 || 3<level_stdout){
  printf("Invalid value of level.of.stdout\n");
  po++;
}

if (level_fileout<0 || 3<level_fileout){
  printf("Invalid value of level.of.fileout\n");
  po++;
}
```

Inputtool.c allows us to write the input file in arbitrary order for the keywords.

Implementation: Total energy (1)

The total energy is given by the sum of six terms, and a proper integration scheme for each term is applied to accurately evaluate the total energy.

$$E_{\rm tot} = E_{\rm kin} + E_{\rm ec} + E_{\rm ee} + E_{\rm xc} + E_{\rm cc} = E_{\rm kin} + E_{\rm na} + E_{\rm ec}^{\rm (NL)} + E_{\delta\rm ee} + E_{\rm xc} + E_{\rm scc}.$$

$$E_{\rm kin} = \sum_{\sigma}\sum_{\rm n}^{N}\sum_{i\alpha,j\beta}\rho_{\sigma,i\alpha j\beta}^{(\mathbf{R}_{\rm n})}h_{i\alpha j\beta,{\rm kin}}^{(\mathbf{R}_{\rm n})}.$$ Kinetic energy

$$E_{\rm ec} = E_{\rm ec}^{\rm (L)} + E_{\rm ec}^{\rm (NL)},$$ Coulomb energy with external potential

$$= \sum_{\sigma}\sum_{\rm n}^{N}\sum_{i\alpha,j\beta}\rho_{\sigma,i\alpha j\beta}^{(\mathbf{R}_{\rm n})}\langle\phi_{i\alpha}(\mathbf{r}-\tau_i)|\sum_I V_{{\rm core},I}(\mathbf{r}-\tau_I)|\phi_{j\beta}(\mathbf{r}-\tau_j-\mathbf{R}_{\rm n})\rangle$$

$$+ \sum_{\sigma}\sum_{\rm n}^{N}\sum_{i\alpha,j\beta}\rho_{\sigma,i\alpha j\beta}^{(\mathbf{R}_{\rm n})}\langle\phi_{i\alpha}(\mathbf{r}-\tau_i)|\sum_I V_{{\rm NL},I}(\mathbf{r}-\tau_I)|\phi_{j\beta}(\mathbf{r}-\tau_j-\mathbf{R}_{\rm n})\rangle,$$

$$E_{\rm ee} = \frac{1}{2}\int dr^3 n(\mathbf{r})V_{\rm H}(\mathbf{r}),$$ Hartree energy

$$= \frac{1}{2}\int dr^3 n(\mathbf{r})\{V_{\rm H}^{\rm (a)}(\mathbf{r}) + \delta V_{\rm H}(\mathbf{r})\},$$

$$E_{\rm xc} = \int dr^3\{n_\uparrow(\mathbf{r}) + n_\downarrow(\mathbf{r}) + n_{\rm pcc}(\mathbf{r})\}\epsilon_{\rm xc}(n_\uparrow + \frac{1}{2}n_{\rm pcc}, n_\downarrow + \frac{1}{2}n_{\rm pcc}),$$ Exchange-correlation energy

$$E_{\rm cc} = \frac{1}{2}\sum_{I,J}\frac{Z_I Z_J}{|\tau_I - \tau_J|}.$$ Core-core Coulomb energy

TO and H.Kino, PRB 72, 045121 (2005)

Implementation: Total energy (2)

The reorganization of Coulomb energies gives three new energy terms.

$$E_{\rm ec}^{(L)} + E_{\rm ee} + E_{\rm cc} = E_{\rm na} + E_{\delta\rm ee} + E_{\rm scc},$$

The neutral atom energy

$$\begin{aligned} E_{\rm na} &= \int dr^3 n(\mathbf{r}) \sum_I V_{{\rm na},I}(\mathbf{r}-\tau_I), \\ &= \sum_\sigma \sum_{\rm n}^{N} \sum_{i\alpha,j\beta} \rho_{\sigma,i\alpha j\beta}^{(\mathbf{R_n})} \sum_I \langle \phi_{i\alpha}(\mathbf{r}-\tau_i) | V_{{\rm na},I}(\mathbf{r}-\tau_I) | \phi_{j\beta}(\mathbf{r}-\tau_j-\mathbf{R_n}) \rangle, \end{aligned}$$

Short range and separable to two-center integrals

Difference charge Hartree energy

$$E_{\delta\rm ee} = \frac{1}{2}\int dr^3 \delta n(\mathbf{r}) \delta V_{\rm H}(\mathbf{r}),$$

Long range but minor contribution

Screened core-core repulsion energy

$$E_{\rm scc} = \frac{1}{2}\sum_{I,J}\left[\frac{Z_I Z_J}{|\tau_I-\tau_J|} - \int dr^3 n_I^{\rm (a)}(\mathbf{r}) V_{{\rm H},J}^{\rm (a)}(\mathbf{r})\right].$$

Short range and two-center integrals

Difference charge

$$\begin{aligned} \delta n(\mathbf{r}) &= n(\mathbf{r}) - n^{\rm (a)}(\mathbf{r}), \\ &= n(\mathbf{r}) - \sum_i n_i^{\rm (a)}(\mathbf{r}), \end{aligned}$$

Neutral atom potential

$$V_{{\rm na},I}(\mathbf{r}-\tau_I) = V_{{\rm core},I}(\mathbf{r}-\tau_I) + V_{{\rm H},I}^{\rm (a)}(\mathbf{r}-\tau_I).$$

Implementation: Total energy (3)

So, the total energy is given by

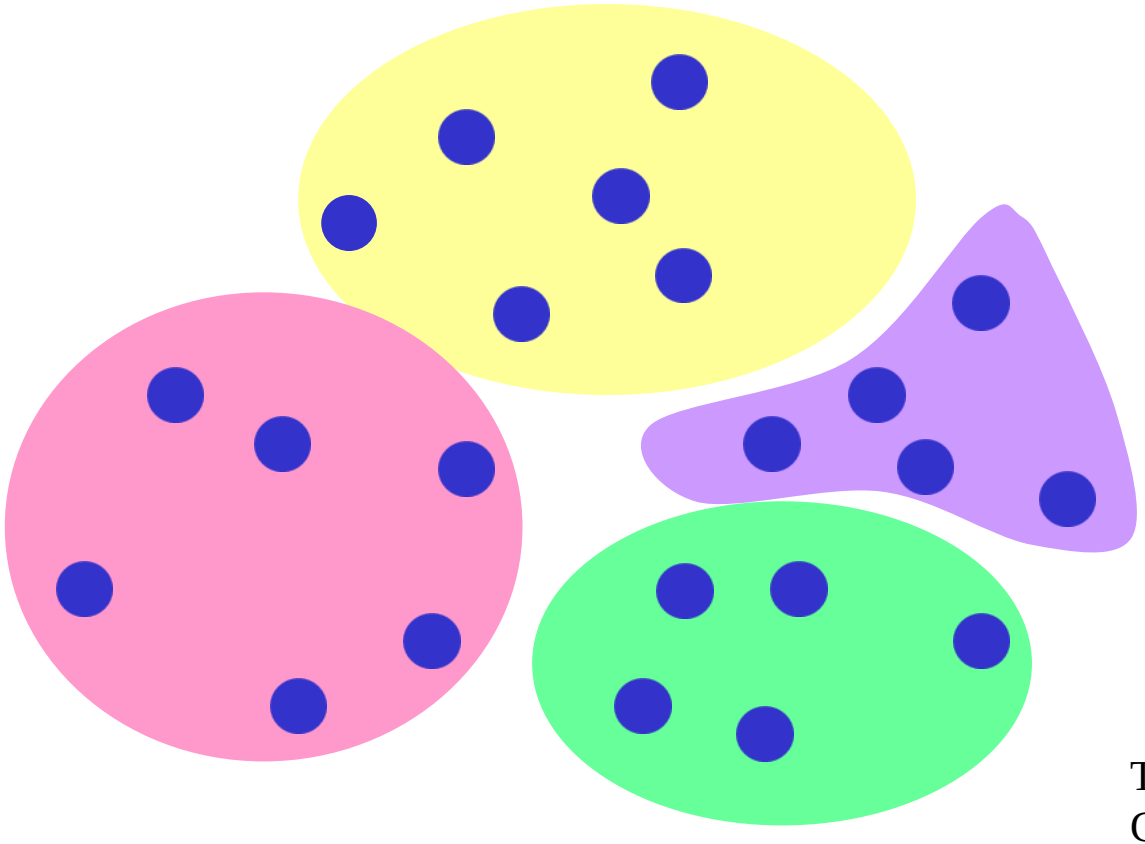
$$E_{\mathrm{tot}} = E_{\mathrm{kin}} + E_{\mathrm{na}} + E_{\mathrm{ec}}^{(\mathrm{NL})} + E_{\delta\mathrm{ee}} + E_{\mathrm{xc}} + E_{\mathrm{scc}}.$$

Each term is evaluated by using a different numerical grid with consideration on accuracy and efficiency.

		The relevant subroutines
E_{kin}		Set_OLP_Kin.c, Total_Energy.c
E_{na}	Spherical coordinate in momentum space	Set_ProExpn_VNA.c, Total_Energy.c
$E_{\mathrm{ec}}^{(\mathrm{NL})}$		Set_Nonlocal.c, Total_Energy.c
$E_{\delta\mathrm{ee}}$	Real space regular mesh	Poisson.c, Total_Energy.c
E_{xc}		Set_XC_Grid.c, Total_Energy.c
E_{scc}	Real space fine mesh	Total_Energy.c

Atomic 3D atomic partitioning

How one can partition atoms to minimize communication and memory usage in the parallel calculations ?



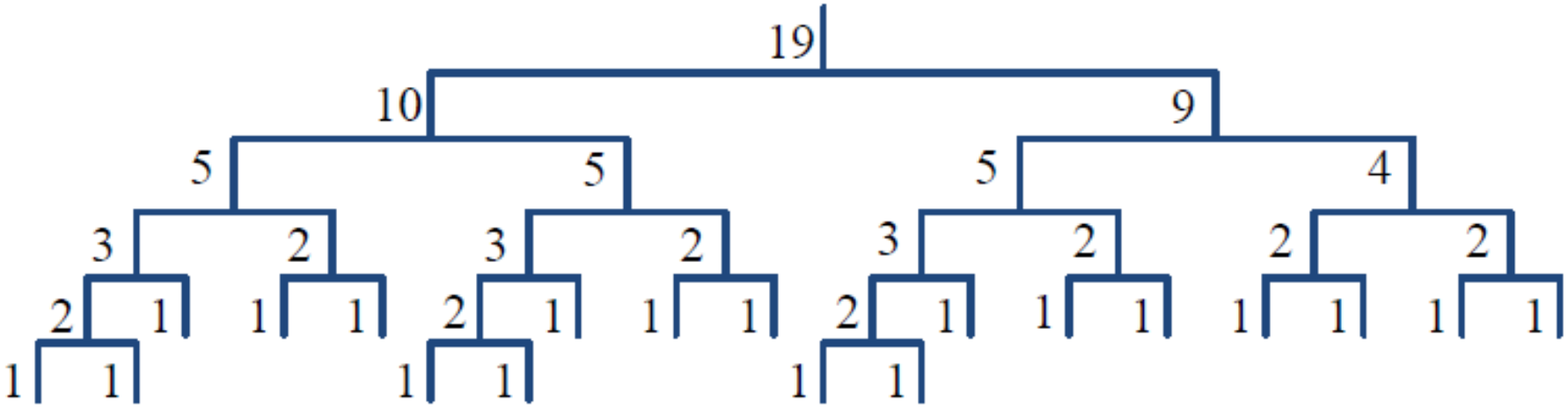
Requirement:

- Locality
- Same computational cost
- Applicable to any systems
- Small computational overhead

T.V.T. Duy and T. Ozaki, Comput. Phys. Commun. 185, 777-789 (2014).

Modified recursive bisection

If the number of MPI processes is 19, then the following binary tree structure is constructed.

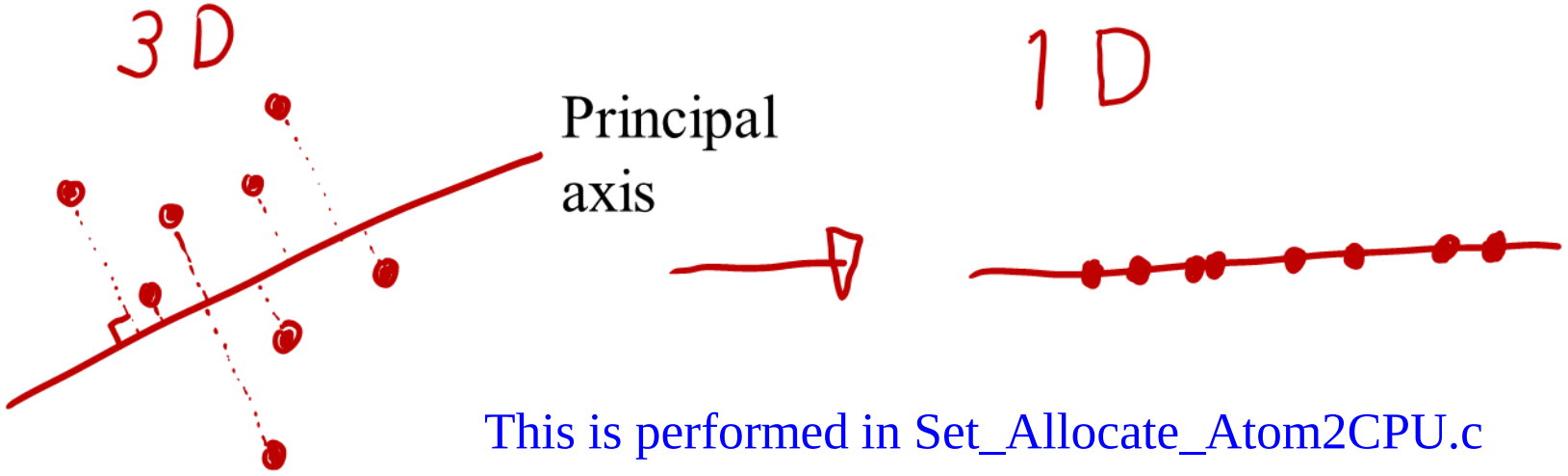


In the conventional recursive bisection, the bisection is made so that a same number can be assigned to each region. However, the modified version bisects with weights as shown above.

This is performed in Set_Allocate_Atom2CPU.c

Reordering of atoms by an inertia tensor

Atoms in an interested region are reordered by projecting them onto a principal axis calculated by an inertia tensor.



This is performed in Set_Allocate_Atom2CPU.c

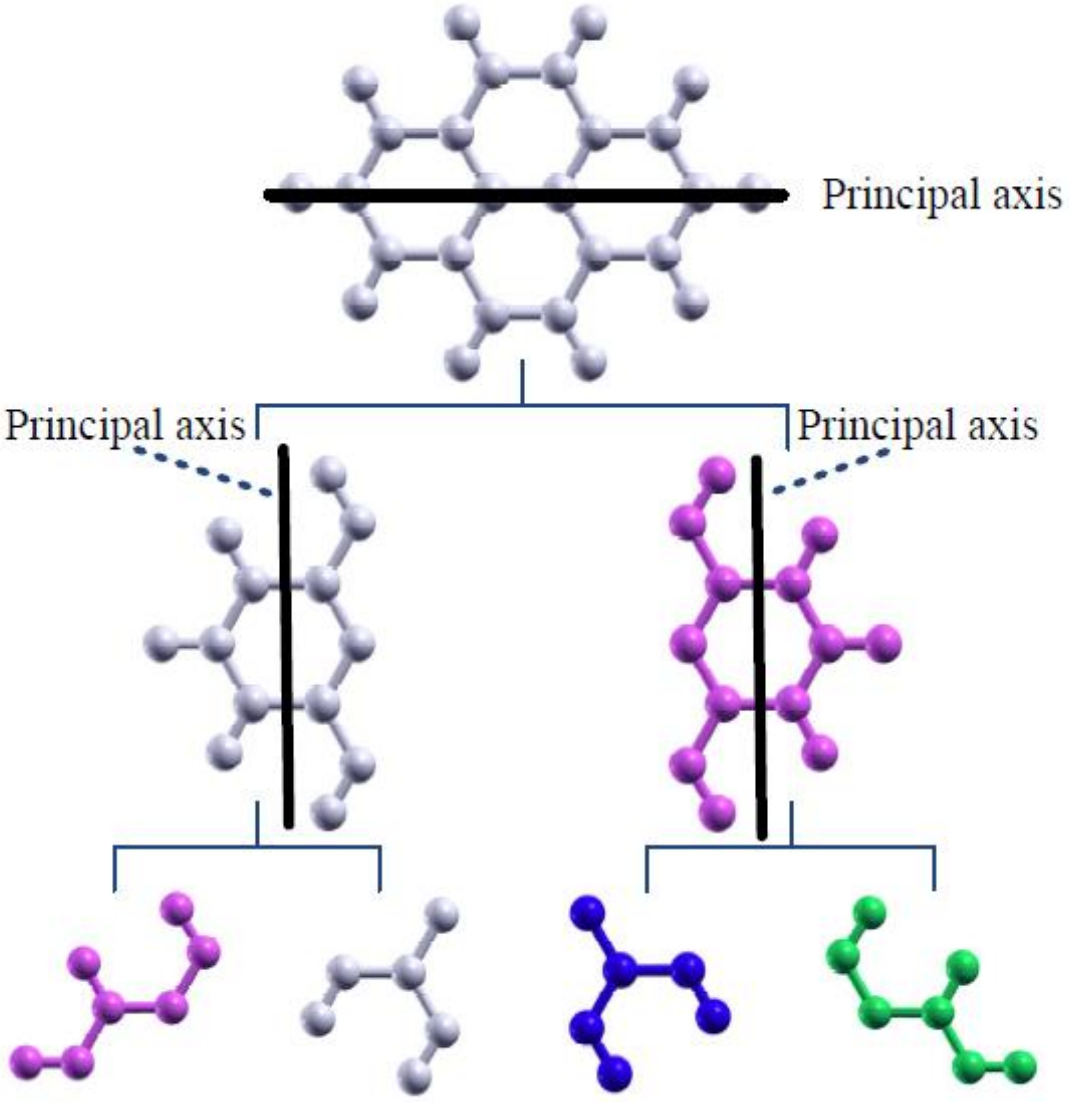
The principal axis is calculated by solving an eigenvalue problem with an inertia tensor:

$$\begin{pmatrix} \sum_i w_i (Y_i^2+Z_i^2) & -\sum_i w_i X_i Y_i & -\sum_i w_i X_i Z_i \\ -\sum_i w_i Y_i X_i & \sum_i w_i (X_i^2+Z_i^2) & -\sum_i w_i Y_i Z_i \\ -\sum_i w_i Z_i X_i & -\sum_i w_i Z_i Y_i & -\sum_i w_i (X_i^2+Y_i \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = -\lambda \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

Recursive atomic partitioning

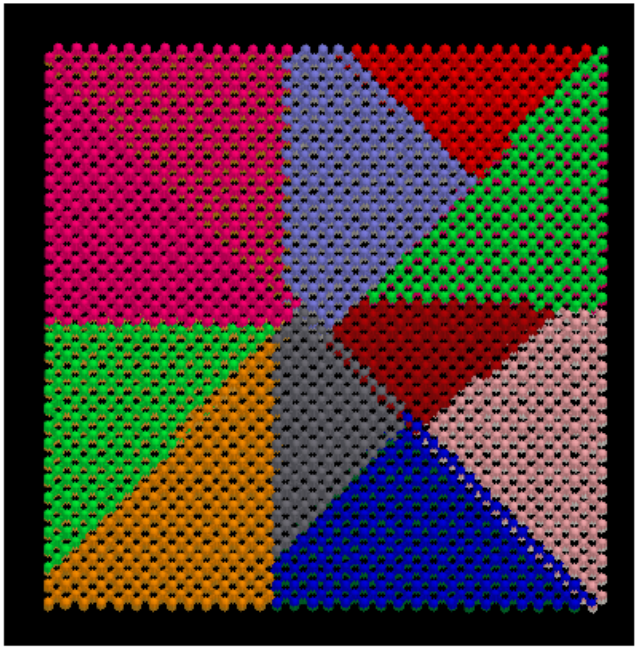
The method guarantees

- Locality of atomic partitioning
- Balanced computational cost
- Applicability to any systems
- Small computational cost

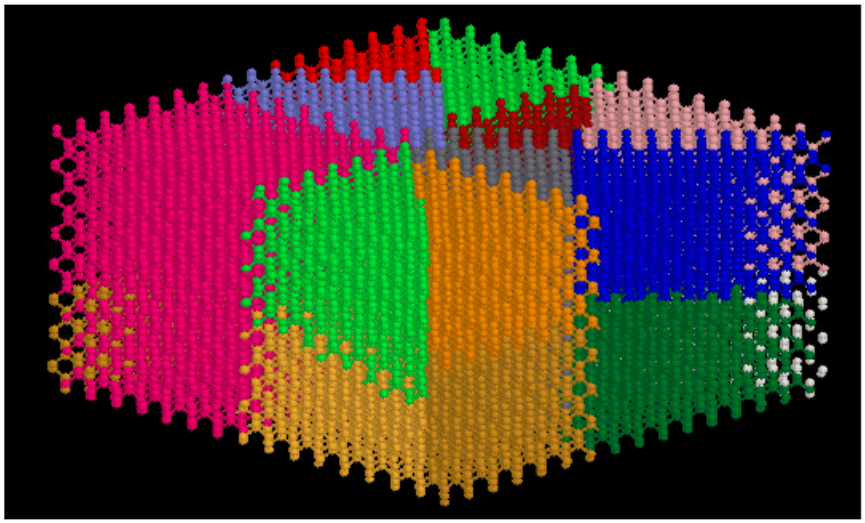


T.V.T. Duy and T. Ozaki, CPC 185, 777 (2014).

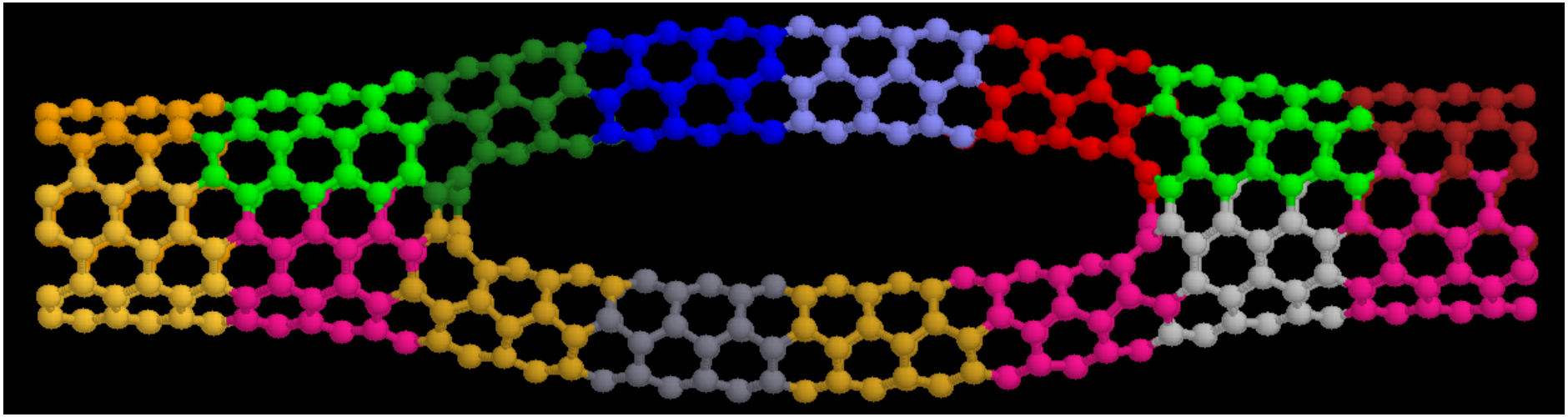
Allocation of atoms to processess



Diamond 16384 atoms, 19 processes



Multiply connected CNT, 16 processes

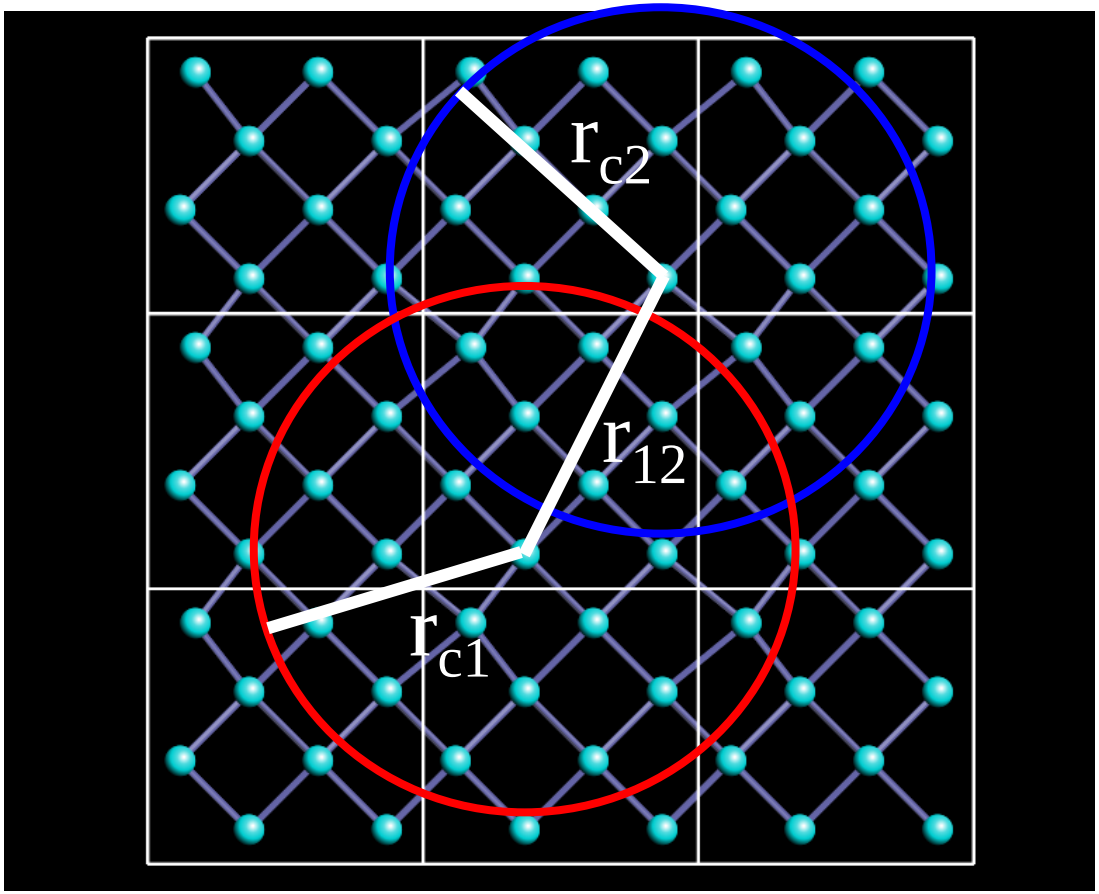


Definition of neighboring atoms

Each atom has strictly localized basis functions. Thus, the non-zero overlap between basis functions occurs if $r_{12} < (r_{c1} + r_{c2})$.

The analysis is performed in Trn_System() of truncation.c, and relevant variables for the information are

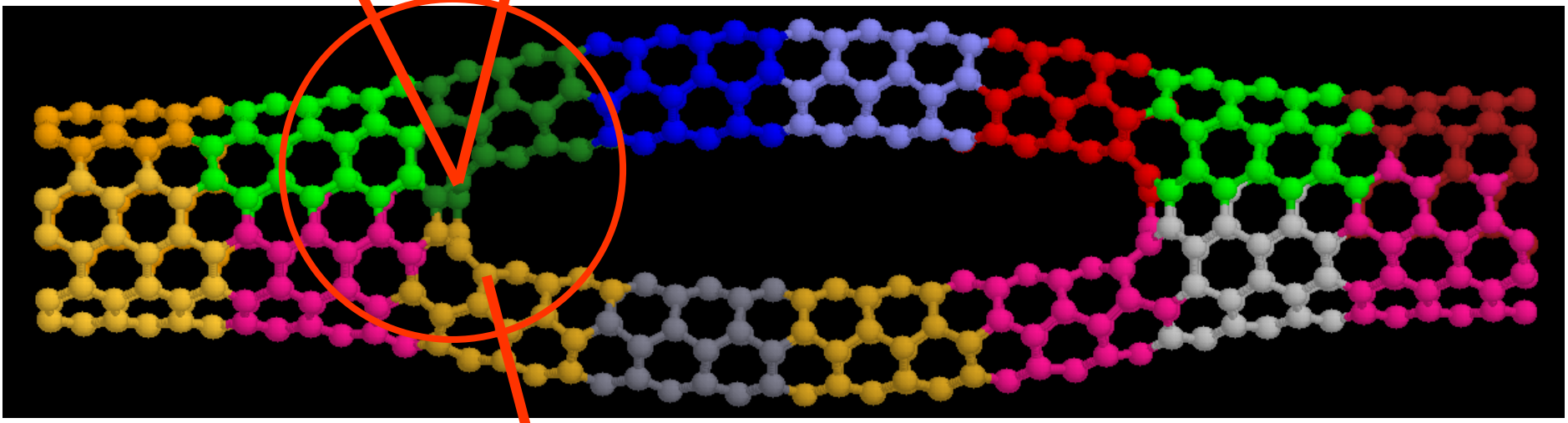
FNAN[]: # of neighboring atoms
natn[][]: global index of the neighboring atom
ncn[][]: cell index of the neighboring atom



Three indices for atoms

Global index: if there are N atoms in the unit cell, then each atom has a global index which is within 1 to N.

interMediate index: a set of atoms (N_p atoms) are assigned to each MPI process. Then index within the MPI process each atom has an intermediate which is within 1 to N_p.



Local index: The second index of natn[][] and ncn[][] runs for a local index which is within 1 to FNAN[].

Example: the three indices of atoms

Only the non-zero Hamiltonian matrix elements are stored in H. An example is given below to show how the conversion among the three indices is made.

```
for (Mc_AN=1; Mc_AN<=Matomnum; Mc_AN++){
  Gc_AN = M2G[Mc_AN];
  Cwan = WhatSpecies[Gc_AN];
  for (h_AN=0; h_AN<=FNAN[Gc_AN]; h_AN++){
    Gh_AN = natn[Gc_AN][h_AN];
    Hwan = WhatSpecies[Gh_AN];
    for (i=0; i<Spe_Total_NO[Cwan]; i++){
      for (j=0; j<Spe_Total_NO[Hwan]; j++){
        for (spin=0; spin<=SpinP_switch; spin++){

          if (ProExpn_VNA==0){
            H[spin][Mc_AN][h_AN][i][j] = F_Kin_flag*H0[0][Mc_AN][h_AN][i][j]
                                       + F_NL_flag*HNL[spin][Mc_AN][h_AN][i][j];
          }
          else{
```

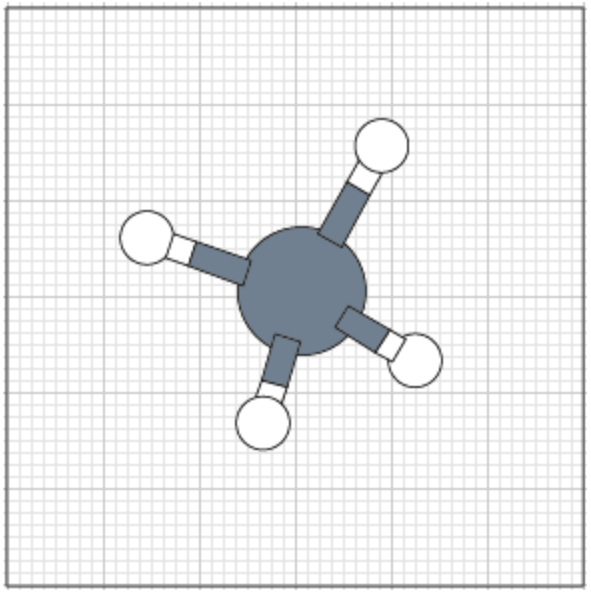
where Mc_AN is the intermediate index, and Gc_AN is the global index, h_AN is the local index, respectively.

M2G, F_G2M, and S_G2M can be used to convert the indices.

Cutoff energy for regular mesh

The two energy components $E_{\delta ee} + E_{xc}$ are calculated on real space regular mesh. The mesh fineness is determined by plane-wave cutoff energies.

```
scf.energycutoff          150.0          # default=150 (Ry)
```



The cutoff energy can be related to the mesh fineness by the following eqs.

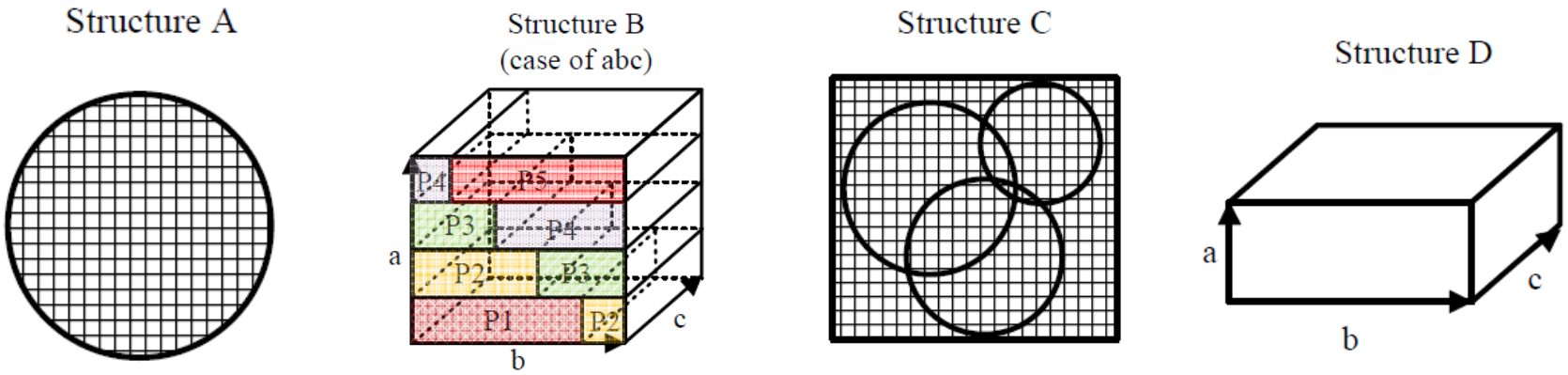
$$E_{\rm cut}^{(1)} = \frac{1}{2}\mathbf{gb}_1 \cdot \mathbf{gb}_1, \quad E_{\rm cut}^{(2)} = \frac{1}{2}\mathbf{gb}_2 \cdot \mathbf{gb}_2, \quad E_{\rm cut}^{(3)} = \frac{1}{2}\mathbf{gb}_3 \cdot \mathbf{gb}_3,$$

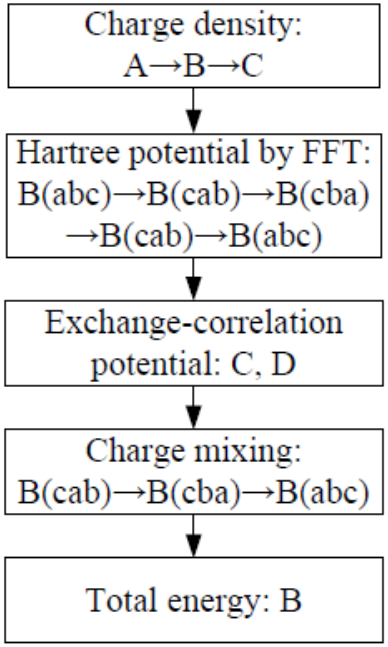
$$\mathbf{ga}_1 = \frac{\mathbf{a}_1}{N_1}, \quad \mathbf{ga}_2 = \frac{\mathbf{a}_2}{N_2}, \quad \mathbf{ga}_3 = \frac{\mathbf{a}_3}{N_3},$$

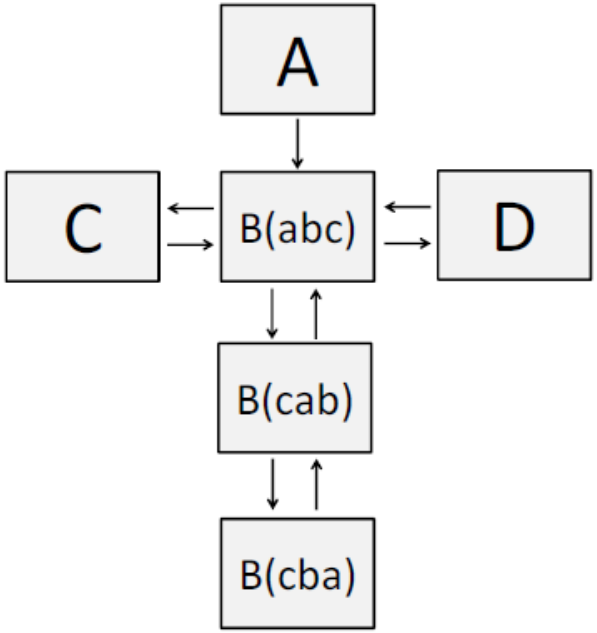
$$\mathbf{gb}_1 = 2\pi\frac{\mathbf{ga}_2 \times \mathbf{ga}_2}{\Delta V}, \quad \mathbf{gb}_2 = 2\pi\frac{\mathbf{ga}_3 \times \mathbf{ga}_1}{\Delta V}, \quad \mathbf{gb}_2 = 2\pi\frac{\mathbf{ga}_1 \times \mathbf{ga}_2}{\Delta V},$$

$$\Delta V = \mathbf{ga}_1 \cdot (\mathbf{ga}_2 \times \mathbf{ga}_3),$$

Partitioning of grids







Uniform grid is used to calculate matrix elements and solve Poisson's equation. A hundred million grid points for a few dozen thousand atoms.

A proper one of four data structures for grid is used for each calculation.

They are designed to minimize the MPI communication.

These data structure are all constructed in Construct_MPI_Data_Structure_Grid() and Set_Inf_SndRcv() of truncation.c.

Case study #1

Values of basis functions are calculated on grids in the structure A.

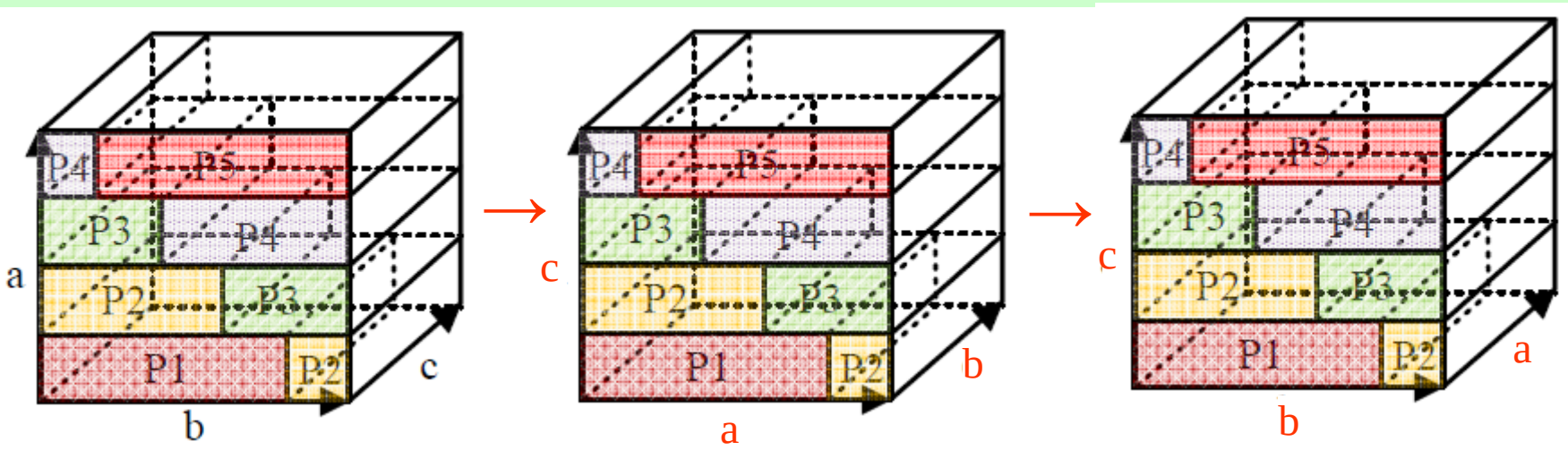
In Set_Orbitals_Grid.c

```
/* get info. on OpenMP */

OMPID = omp_get_thread_num();
Nthrds = omp_get_num_threads();
Nprocs = omp_get_num_procs();

for (Nc=OMPID*GridN_Atom[Gc_AN]/Nthrds; Nc<(OMPID+1)*GridN_Atom[Gc_AN]/Nthrds; Nc++){

  GNc = GridListAtom[Mc_AN][Nc];
  GRc = CellListAtom[Mc_AN][Nc];

  Get_Grid_XYZ(GNc, Cxyz);
  x = Cxyz[1] + atv[GRc][1] - Gxyz[Gc_AN][1];
  y = Cxyz[2] + atv[GRc][2] - Gxyz[Gc_AN][2];
  z = Cxyz[3] + atv[GRc][3] - Gxyz[Gc_AN][3];

  if (Cnt_kind==0){

    /* Get_Orbitals(Cwan, x, y, z, Chi0); */
    /* start of inlining of Get_Orbitals */
```

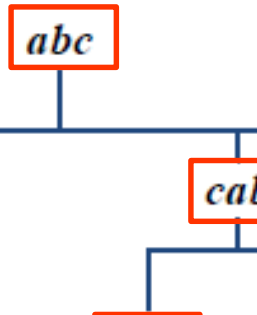
Nc runs over grids in the structure A.

Case study #2:

2D-parallelization of 3D-FFT in Poisson.c



Compared to 1D-parallelization, no increase of MPI communication up to N. Even at N^2, just double communication.



MPI Processes	cab	acb	acb	cab	***cba***	bca	cba	bca
N/2	N^3-2N	$2N^3$-4N	$2N^3$-4N	$2N^3$-4N	***N^3-2N***	$2N^3$-4N	N^3-2N	N^3-2N
N	N^2(N-1)	$2N^2$(N-1)	$2N^2$(N-1)	$2N^2$(N-1)	***N^2(N-1)***	$2N^2$(N-1)	N^2(N-1)	N^2(N-1)
2N	$1.5N^3$-N^2	$2N^2$(N-1)	$2N^2$(N-1)	$2N^2$(N-1)	***$1.5N^3$-N^2***	$2N^2$(N-1)	$1.5N^3$-N^2	$1.5N^3$-N^2
N^2	$2N^2$(N-1)	$2N^2$(N-1)	$2N^2$(N-1)	$2N^2$(N-1)	***$2N^2$(N-1)***	$2N^2$(N-1)	$2N^2$(N-1)	$2N^2$(N-1)

Case study #3

Each MPI process calculates a part of the energy terms evaluated on the regular mesh using the structure B(ABC) as follows:

```
/*****************************************************************
      calculations of Eva, Eef, EH1, and EXC
*****************************************************************/

My_Eva = 0.0;
My_Eef = 0.0;
My_EH1 = 0.0;
My_EXC[0] = 0.0;
My_EXC[1] = 0.0;

for (BN=0; BN<My_NumGridB_AB; BN++){

  sden[0] = Density_Grid_B[0][BN];
  sden[1] = Density_Grid_B[1][BN];
  tden = sden[0] + sden[1];
  aden = ADensity_Grid_B[BN];
  pden = PCCDensity_Grid_B[BN];

  /* if (ProExpn_VNA==off), Ena is calculated here. */
  if (ProExpn_VNA==0) My_Eva += tden*VNA_Grid_B[BN];

  /* electric energy by electric field */
  if (E_Field_switch==1) My_Eef += tden*VEF_Grid_B[BN];
```

In Total_Energy.c

BN runs over grids in the structure B(ABC).

Case study #4

Each MPI process calculates exchange-correlation potential and energy density using the structure D as follows:

In Set_XC_Grid.c

```
OMPID = omp_get_thread_num();
Nthrds = omp_get_num_threads();

Ng1 = Max_Grid_Index_D[1] - Min_Grid_Index_D[1] + 1;
Ng2 = Max_Grid_Index_D[2] - Min_Grid_Index_D[2] + 1;
Ng3 = Max_Grid_Index_D[3] - Min_Grid_Index_D[3] + 1;

for (MN=OMPID; MN<My_NumGridD; MN+=Nthrds){

  i = MN/(Ng2*Ng3);
  j = (MN-i*Ng2*Ng3)/Ng3;
  k = MN - i*Ng2*Ng3 - j*Ng3;

  if ( i==0 || i==(Ng1-1) || j==0 || j==(Ng2-1) || k==0 || k==(Ng3-1) ){

    dDen_Grid[0][0][MN] = 0.0;
    dDen_Grid[0][1][MN] = 0.0;
    dDen_Grid[0][2][MN] = 0.0;
    dDen_Grid[1][0][MN] = 0.0;
    dDen_Grid[1][1][MN] = 0.0;
    dDen_Grid[1][2][MN] = 0.0;
  }
```

MN runs over grids in the structure D.

Summary

The data structure of OpenMX is carefully designed so that the size of memory and MPI communication can be minimized.

The first step to construct the data structure is how atoms are allocated to each MPI process by using the modified bisection and inertia tensor projection methods.

The second step to construct the data structure is how four sorts of grid structure are constructed.

Each of calculations are performed by choosing one of the four grid structures, and changing the grid structure requires the MPI communication, of which pattern is determined beforehand in truncation().