

# Getting Started with OpenMX Ver. 3.5

1. Getting the package and its installation
2. Running of test jobs
3. Input files
4. Output files
5. Basis sets and pseudopotentials
6. Functionals
7. Cutoff energy for regular mesh
8. Getting SCF
9. Geometry optimization
10. Initial spin moments
11. Visualization of cube files

# The PDF file

is available at

[http://www.openmx-square.org/tech\\_notes/tech\\_notes.html](http://www.openmx-square.org/tech_notes/tech_notes.html)

# Get the package

(1) Go to

<http://www.openmx-square.org/download.html>

(2) Download OpenMX Ver. 3.5

Up

## Download of OpenMX

Available packages in terms of [GNU-GPL](#)

- [openmx3.5 \(release date: 10/Sep./2009\)](#)
- [openmx3.4 \(release date: 23/June/2008\)](#)    [+ patch \(17/Mar./2009\)](#)
- [openmx3.3 \(release date: 30/July/2007\)](#)    [+ patch \(25/Sep./2007\)](#)
- [openmx3.2 \(release date: 01/Apr./2007\)](#)    [+ patch \(02/July/2007\)](#)

# Installation No.0

Before the installation of OpenMX, the installation of **LAPACK, BLAS, and FFTW** is indispensable.

Of course, **variants** of LAPACK/BLAS can be used instead, such as

ATLAS,  
ACML  
MKL

Our recommendation is **ACML** which is fast and stable. In addition, ACML can be used for not only AMD but also Intel processors.

# Installation No.1

(1) Decompress the tar file.

```
X kterm  
[ozaki@vtpcc01 ozaki]$ tar zxvf openmx3.5.tar.gz
```

(2) You will find four directories below openmx3.5.

```
X kterm  
[ozaki@vtpcc01 ozaki]$ cd openmx3.5  
[ozaki@vtpcc01 openmx3.5]$ ls  
DFT_DATA  DFT_DATA06  source  work  
[ozaki@vtpcc01 openmx3.5]$
```

# Installation No.2

## (1) Modify makefile in openmx3.5/source

```
kterm
[ozaki@vtpcc01 ozaki]$ cd openmx3.5/source/
[ozaki@vtpcc01 source]$ emacs -nw makefile

# (4) make install
#
CC      = gcc -O3 -Dnomp -Dnoomp -I/usr/local/include
LIB     = -L/usr/local/lib -lfftw3 -llapack -lblas -lc
```

**The two lines “CC” and “LIB” must be changed depending on your computational environment. Several examples can be found in makefile**

## (2) Compile the sources

```
kterm
[ozaki@vtpcc12 source]$ make install
/usr/local/mpich-1.2.5-intel/bin/mpicc -openmp -O3 -I/home/ozaki/include -I/home/ozak
32bit/ifort32_mp/include -c openmx.c
/usr/local/mpich-1.2.5-intel/bin/mpicc -openmp -O3 -I/home/ozaki/include -I/home/ozak
32bit/ifort32_mp/include -c openmx_common.c
```

# Installation No.3

Compiler options (not all, see the manual for more details)

-Dnomp      for the serial version

-Dnoomp      In case that compiler does  
not support openmp.

-Dfftw2      for use of FFTW2

# Installation No.4

You can find several examples for setting CC and LIB in makefile.

```
#####  
#  
# Please set a proper CC and LIB for the compilation. #  
# Examples of CC and LIB on several platforms are shown below. #  
#  
#####  
  
#  
# Cygwin, IBM ThinkPAD X40 (Pentium M 1.0GHz)  
#  
# CC      = gcc -Dnomp -Dblaswrap -O3 -I/home/ozaki/include  
# STACK  = -Wl,--heap,9000000,--stack,9000000  
# LIB     = -L/home/ozaki/lib -llapack -lblas -lg2c -li77 -lfftw3 -static
```

Also, tips for installation can be found in the OpenMX Forum as

## ⚠ Tips for installation of OpenMX

Date: 2007/09/25 17:12

Name: **T.Ozaki**

Dear All,

The following is tips for installation of OpenMX.

Most problems in installation of OpenMX come from compilation of LAPACK and BLAS and its linking. Thus, I shall show tips for installation of OpenMX on several platforms.

(1) Intel Pentium 4 and Xeon (32 bit)

# Running of test jobs No.1

Move to the directory 'work', and perform the program, openmx, using an input file, *Methane.dat*, which can be found in the directory 'work' as follows:

```
X kterm  
[ozaki@vtpcc12 work]$ pwd  
/home/ozaki/openmx3.5/work  
[ozaki@vtpcc12 work]$ ./openmx Methane.dat > met.std &
```

For the MPI version, you can perform as follows:

```
X kterm  
[ozaki@vtpcc12 work]$ pwd  
/home/ozaki/openmx3.5/work  
[ozaki@vtpcc12 work]$ mpirun -np 4 openmx Methane.dat > met.std &
```

# Running of test jobs No.2

After the calculation, **11 files** and **one directory** will be generated

met.std	standard output of the SCF calculation
met.out	input file and standard output
met.xyz	final geometrical structure
met.cif	structure file for Material Studio
met.ene	quantities computed at every MD step
met.memory0	analysis for used memory
met.md	geometrical structures at every MD step
met.md2	geometrical structure of the final MD step
met.tden.cube	total electron density in the Gaussian cube format
met.v0.cube	Kohn-Sham potential in the Gaussian cube format
met.vhart.cube	Hartree potential in the Gaussian cube format
met_rst/	directory storing restart files

# Running of test jobs No.3

**met.out** gives the standard information about the calculation such as the SCF convergence history, total energy, Mulliken charge, etc.

```
*****
*****
                      SCF history at MD= 1
*****
*****
SCF=   1  NormRD=  1.000000000000  Uele= -3.799184452246
SCF=   2  NormRD=  0.294505017736  Uele= -3.180922853695
SCF=   3  NormRD=  0.088735677904  Uele= -3.371991788325
SCF=   4  NormRD=  0.021096020030  Uele= -3.435330322081
SCF=   5  NormRD=  0.006019683783  Uele= -3.449516147411
SCF=   6  NormRD=  0.000784960501  Uele= -3.452522027174
SCF=   7  NormRD=  0.000002534854  Uele= -3.453266301957
SCF=   8  NormRD=  0.000000871028  Uele= -3.453266590490
SCF=   9  NormRD=  0.000000229718  Uele= -3.453266630447
SCF=  10  NormRD=  0.000000531457  Uele= -3.453266655128
SCF=  11  NormRD=  0.000000265107  Uele= -3.453266657644
*****
*****
                      Total energy (Hartree) at MD = 1
*****
*****
Uele.          -3.453266657644
Ukin.           5.824571433675
UH0.          -14.517598384684
UH1.           0.012112580577
Una.           -6.365977491981
Unl.           0.681047545213
Uxc0.          -1.609135574153
Uxc1.          -1.609135574153
Ucore.         9.551521413583
Uhub.          0.000000000000
Ucs.           0.000000000000
Uzs.           0.000000000000
Uzo.           0.000000000000
Uef.           0.000000000000
Utot.          -8.032594051921
```

# Automatic running test

To check whether most functionalities of OpenMX have been successfully installed, it is possible to perform an automatic running test.

For the serial case

```
kterm  
[ozaki@vtppcc12 work]$ ./openmx -runtest
```

For the MPI case

```
kterm  
[ozaki@vtppcc12 work]$ mpirun -np 6 openmx -runtest
```

The result can be found in a file `runtest.result` as

```
kterm  
[ozaki@vtppcc12 work]$ more runtest.result  
 1 input_example/Benzene.dat      Elapsed time(s)=   9.94 diff Utot= 0.000000026206 diff Force= 0.000000000082  
 2 input_example/C60.dat          Elapsed time(s)=  42.02 diff Utot= 0.000000153045 diff Force= 0.000000001243  
 3 input_example/CO.dat           Elapsed time(s)=  32.40 diff Utot= 0.000000001580 diff Force= 0.0000000009634  
 4 input_example/Cr2.dat          Elapsed time(s)=  22.21 diff Utot= 0.000000016815 diff Force= 0.000000000031  
 5 input_example/Crys-MnO.dat     Elapsed time(s)=  66.06 diff Utot= 0.000000015036 diff Force= 0.000000001301  
 6 input_example/GaAs.dat         Elapsed time(s)=  83.70 diff Utot= 0.000000002263 diff Force= 0.000000000002  
 7 input_example/Glycine.dat      Elapsed time(s)=  14.27 diff Utot= 0.000000001423 diff Force= 0.000000000384  
 8 input_example/Graphite4.dat    Elapsed time(s)=   8.32 diff Utot= 0.000000003735 diff Force= 0.000000000055  
 9 input_example/H2O-EF.dat       Elapsed time(s)=  13.65 diff Utot= 0.000000014457 diff Force= 0.000000000349  
10 input_example/H2O.dat          Elapsed time(s)=  10.64 diff Utot= 0.000000014569 diff Force= 0.0000000008071  
11 input_example/HYb.dat          Elapsed time(s)=  49.90 diff Utot= 0.000000000481 diff Force= 0.000000000107  
12 input_example/Methane.dat      Elapsed time(s)=   8.45 diff Utot= 0.000000037886 diff Force= 0.000000000036  
13 input_example/Mo1_MnO.dat     Elapsed time(s)=  43.24 diff Utot= 0.000000005192 diff Force= 0.000000000038  
14 input_example/Ndia2.dat       Elapsed time(s)=   8.07 diff Utot= 0.000000011974 diff Force= 0.000000000005  
  
Total elapsed time (s)      412.86
```

Also, the results on several platforms can be found in `openm3.5/work/input_example`

# Input file

```
kterm
[ozaki@vtpcc01 work]$ more Methane.dat
#
# SCF calculation of a methane molecule by the LDA
# and the cluster method
#
#
# File Name
#
System,CurrentDirectory      ./      # default=./
System.Name                  met
level.of.stdout              1      # default=1 (1-3)
level.of.fileout             1      # default=1 (0-2)
#
# Definition of Atomic Species
#
Species.Number               2
<Definition.of.Atomic.Species
H   H4,0-s1                  H_TM
C   C4,5-s1p1                C_TM_PCC
Definition.of.Atomic.Species>
#
# Atoms
#
Atoms,Number                 5
Atoms,SpeciesAndCoordinates,Unit Ang # Ang|AU
<Atoms,SpeciesAndCoordinates
1  C   0.000000  0.000000  0.000000  2.0  2.0
2  H   -0.889981 -0.629312  0.000000  0.5  0.5
3  H   0.000000  0.629312 -0.889981  0.5  0.5
4  H   0.000000  0.629312  0.889981  0.5  0.5
5  H   0.889981 -0.629312  0.000000  0.5  0.5
Atoms,SpeciesAndCoordinates>
```

- (1) Value **behind** keyword
- (2) The order is **arbitrarily**.
- (3) Put # to the head of line for comment

## Selected keywords:

The name of system

The number of atoms

Definition of species

Atomic coordinates

# Output files

The amount of the output files is controlled by the keyword, **'level.of.fileout'** (0,1,2). The typical default output files is as follows:

met.std	standard output of the SCF calculation
met.out	input file and standard output
met.xyz	final geometrical structure
met.cif	structure file for Material Studio
met.ene	values computed at every MD step
met.memory0	analysis for used memory
met.md	geometrical structures at every MD step
met.md2	geometrical structure of the final MD step
met.tden.cube	total electron density in the Gaussian cube format
met.v0.cube	Kohn-Sham potential in the Gaussian cube format
met.vhart.cube	Hartree potential in the Gaussian cube format
met_rst/	directory storing restart files

For more details, see the section **'Output files'** of the manual.



# Basis sets (\*.pao) and pseudopotentials (\*.vps)

The downloaded vps and pao files have to be stored in the following way:

**\*.vps**

If Ver. 2004,

openmx3.5/DFT\_DATA/VPS

If Ver. 2006,

openmx3.5/DFT\_DATA06/VPS

**\*.pao**

If Ver. 2004,

openmx3.5/DFT\_DATA/PAO

If Ver. 2006,

openmx3.5/DFT\_DATA06/PAO

## Specification of the directory storing \*.pao and \*.vps

The directory *DFT\_DATA* can be specified by the keyword in your input file as

```
DATA.PATH    ../DFT_DATA2006/    # default=../DFT_DATA/
```

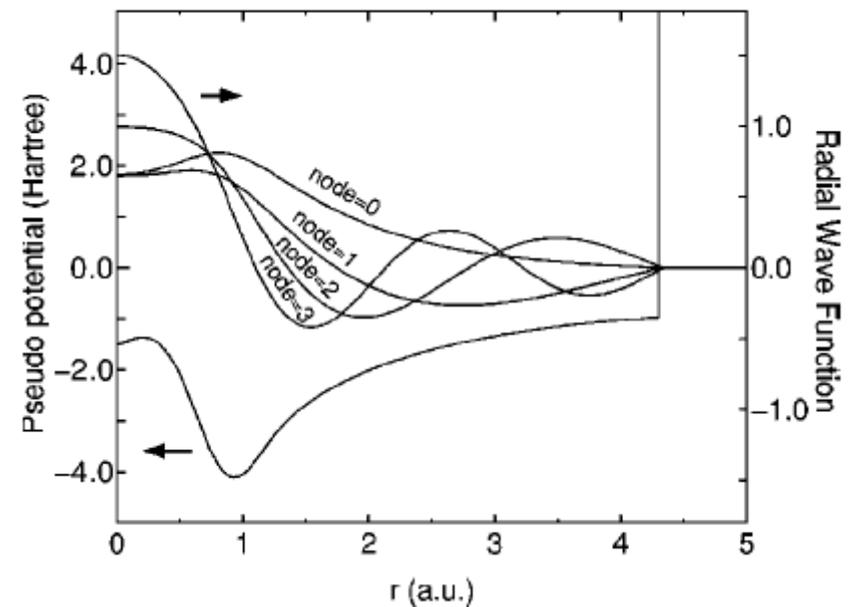
The keyword is useful to avoid using PAO files for pseudopotentials of the different versions

# Basis sets No.1

The species of atoms can be specified by

```
Species,Number      2
<Definition,of,Atomic,Species
H   H4.0-s1         H_TM
C   C4.5-s1p1       C_TM_PCC
Definition,of,Atomic,Species>
```

where H4.0-s1 means that the first s-orbital of H4.0.pao is used, and C4.5-s1p1 means that the first s- and p-orbitals in C4.5.pao is used as basis functions.



# Basis sets No.2

The proper choice of basis functions depends on elements and chemical environment. Although detailed analyses can be found In PRB 69, 195113 (2004) and JCP 121, 10879 (2004), the following can be good starting points.

Examples

H4.5-s2	Li8.0-s2	Ti5.5-s2p2d1	Fe5.5-s2p2d1
B4.5-s2p1	Na9.0-s2	V5.5-s2p2d1	Co5.5-s2p2d1
C4.5-s2p1	K9.0-s2	Cr5.5-s2p2d1	Ni5.5-s2p2d1
N4.5-s2p2		Mn5.5-s2p2d1	Cu5.5-s2p2d1
O4.5-s2p2d1			
F4.5-s2p2d1			

Trends:

- (1) Elements located in the right side of the periodic table requires basis sets with higher angular momentum.
- (2) Alkali metals require a long tail of basis functions.
- (3) 3d-transition metals are well described by TM5.5-s2p2d1 in their oxide.

# Pseudopotentials

The species of atoms can be specified by

```
Species,Number      2
<Definition,of,Atomic,Species
H   H4,0-s1          H_TM
C   C4,5-s1p1        C_TM_PCC
Definition,of,Atomic,Species>
```

H\_TM and C\_TM\_PCC mean that H\_TM.vps and C\_TM\_PCC.vps stored in the directory specified by DATA.PATH are used as pseudopotentials.

# Functionals

Three kinds of functional are now available by the following keywords

```
scf.XcType          LDA          # LDA|LSDA-CA|LSDA-PW|GGA-PBE
scf.SpinPolarization off         # On|Off|NC
```

In case of LDA+U

```
scf.Hubbard.U       on           # On|Off , default=off
scf.Hubbard.Occupation dual      # onsite|full|dual, default=dual
```

On-site Us are specified by

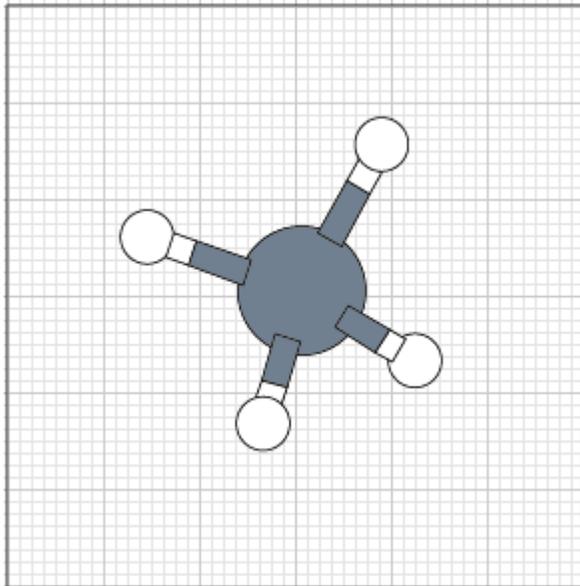
```
Species.Number      2
<Definition.of.Atomic.Species
Ni Ni5.5-s2p2d2f1 Ni_LDA
O  O5.0-s2p2d1    O_LDA
Definition.of.Atomic.Species>

<Hubbard.U.values          # eV
Ni 1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 7.0 2d 0.0 1f 0.0
O  1s 0.0 2s 0.0 1p 0.0 2p 0.0 1d 0.0
Hubbard.U.values>
```

# Cutoff energy for regular mesh No.1

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.

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

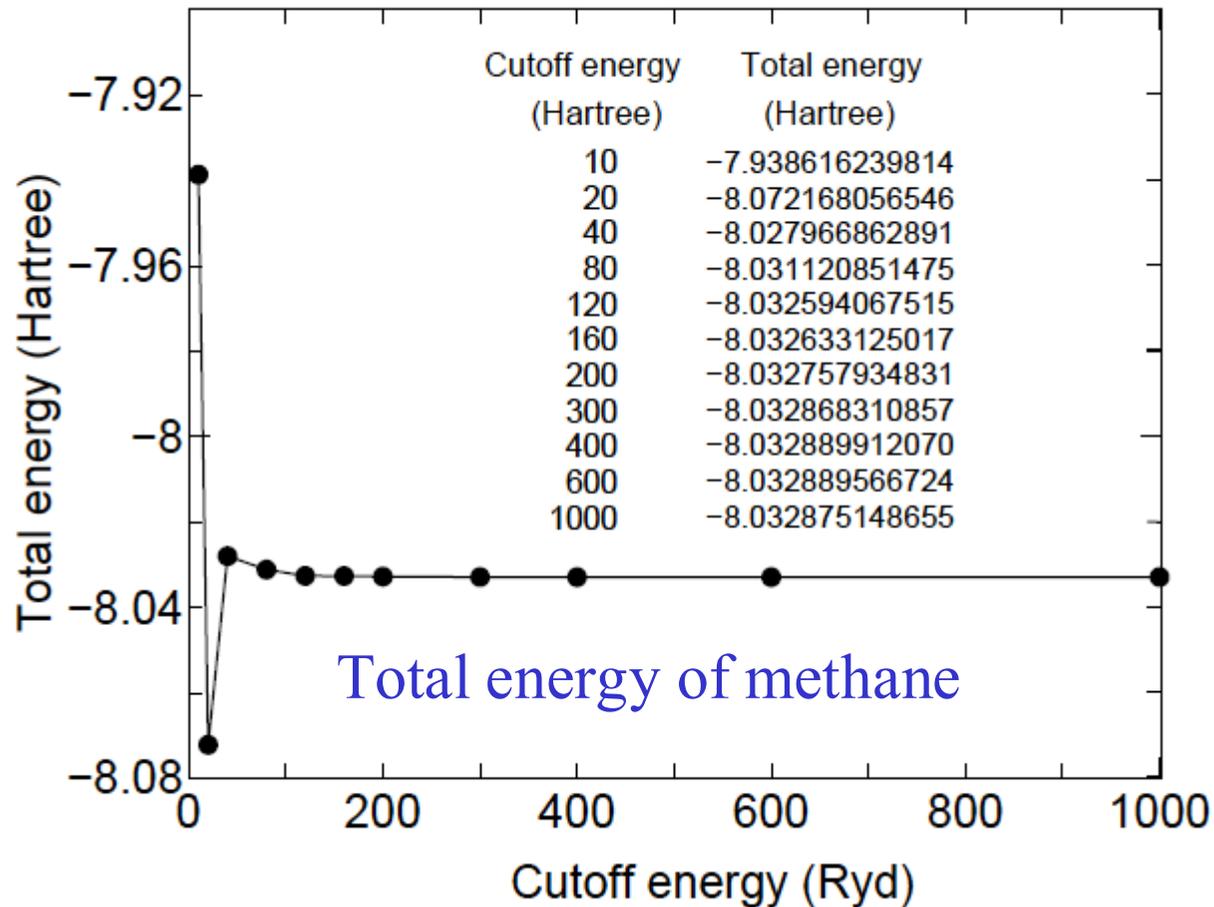
$$\mathbf{ga}_1 = \frac{\mathbf{a}_1}{N_1}, \quad \mathbf{ga}_2 = \frac{\mathbf{a}_2}{N_2}, \quad \mathbf{ga}_3 = \frac{\mathbf{a}_3}{N_3},$$

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

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

# Cutoff energy for regular mesh No.2

The proper choice of the cutoff energy depends on system.  
150-250 Ryd is a reasonable choice in most cases.



# RMM-DIIS for obtaining SCF

In most cases, the Residual Minimization Method in the direct Inversion of Iterative subspace (RMM-DIIS) in momentum space works well.

Residual vectors

$$R_n(\mathbf{q}) \equiv \rho_n^{(\text{out})}(\mathbf{q}) - \rho_n^{(\text{in})}(\mathbf{q}),$$

Kerker metric

$$\langle R_m | R_{m'} \rangle \equiv \sum_{\mathbf{q}} \frac{R_m^*(\mathbf{q}) R_{m'}(\mathbf{q})}{w(\mathbf{q})},$$

Let us assume the residual vector at the next step is expressed by

with the Kerker factor

$$w(\mathbf{q}) = \frac{|\mathbf{q}|^2}{|\mathbf{q}|^2 + q_0^2},$$

$$\bar{R}_{n+1} = \sum_{m=n-(p-1)}^n \alpha_m R_m,$$

**Assume an optimum charge is given by**

Minimize

$$\langle \bar{R}_n | \bar{R}_n \rangle \longrightarrow \text{optimum } \alpha \text{ s}$$

$$\rho_{n+1}^{(\text{in})} = \sum_{m=n-(p-1)}^n \alpha_m \rho_m^{(\text{in})} + \beta \sum_{m=n-(p-1)}^n \alpha_m R_m$$

with respect to  $\alpha$

# Comparison of mixing methods

## Anderson mixing

$$\bar{\rho}_n^{(\text{in})} = \rho_n^{(\text{in})} + \sum_{m=n-(p-1)}^{n-1} \theta_m (\rho_m^{(\text{in})} - \rho_n^{(\text{in})}), \quad \bar{R}_n^{(\text{in})} = R_n^{(\text{in})} + \sum_{m=n-(p-1)}^{n-1} \theta_m (R_m^{(\text{in})} - R_n^{(\text{in})}).$$

$$\rho_{n+1}^{(\text{in})} = \bar{\rho}_n^{(\text{in})} + \beta \bar{R}_n^{(\text{in})} \quad \rightarrow \quad \text{equivalent to RMM-DIIS}$$

## Broyden mixing

$$E = |G_n - G_{n-p}|^2 + \sum_{m=l+1-p}^{n-1} \langle \lambda_m | \{ (|n_{m+1}\rangle - |n_m\rangle) - G_n (|R_{m+1}\rangle - |R_m\rangle) \}$$

$$\frac{\partial E}{\partial G} = 0 \quad \frac{\partial E}{\partial \lambda} = 0$$

$$|n_{n+1}\rangle = |n_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) - G_{n-p} \left\{ |R_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) \right\}$$

$$G_{n-p} = -\beta \quad \rightarrow \quad \rho_{n+1}^{(\text{in})} = \bar{\rho}_n^{(\text{in})} + \beta \bar{R}_n^{(\text{in})} \quad \rightarrow \quad \text{equivalent to RMM-DIIS}$$

**RMM-DIIS, Anderson, Broyden methods are all equivalent from the mathematical point of view and based on a quasi-Newton method.**

# A way for improving the SCF convergence

## Broyden method

$$|n_{n+1}\rangle = |n_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) - G_{n-p} \left\{ |R_n\rangle - \sum_{m=n-p}^{n-1} \gamma_m^n (|n_{m+1}\rangle - |n_m\rangle) \right\}$$

$$G_{n-p} = -\beta \quad \rightarrow \quad \rho_{n+1}^{(\text{in})} = \bar{\rho}_n^{(\text{in})} + \beta \bar{R}_n^{(\text{in})}$$

If  $G$  can be stored, the Broyden method may be the best method among them. However,  $G$  is too large to be stored. Thus, from the theoretical point of view a reasonable improvement of the convergence can be obtained by increasing the number of the previous steps.

In fact, the convergent results were obtained using 30-50 previous steps in the RMM-DIIS for 20 difficult systems that the SCF is hardly obtained using a smaller number of previous steps.

The results can be found in [http://www.jaist.ac.jp/~t-ozaki/large\\_example.tar.gz](http://www.jaist.ac.jp/~t-ozaki/large_example.tar.gz)

# Getting SCF No.1

Five methods for getting SCF are available, the **RMM-DIISK** is the best choice among them in most cases.

```
scf.Mixing.Type          rmm-diisk      # Simple|Rmm-Diis|Gr-Pulay
                        # Kerker|Rmm-Diisk
```

The **RMM-DIISK** is controlled by the following keywords:

```
scf.Init.Mixing.Weight  0.0100          # default=0.30
scf.Min.Mixing.Weight   0.0010          # default=0.001
scf.Max.Mixing.Weight   0.1000          # default=0.40
scf.Kerker.factor       3.000          # default=1.00
scf.Mixing.History      20             # default=5
scf.Mixing.StartPulay   12             # default=6
scf.Mixing.EveryPulay   1              # default=6
scf.criterion           1.0e-10        # default=1.0e-6 (Hartree)
```

# Getting SCF No.2

If you **cannot** obtain the SCF convergence, try the following prescriptions:

- **Decrease `scf.Min.Mixing.Weight`**. The large `scf.Min.Mixing.Weight` may lead to unfavorable charge sloshing
- **Use a rather larger value for `scf.Mixing.StartPulay`**. Before starting the Pulay type mixing, achieve a convergence at some level. An appropriate value may be 10 to 30 for `scf.Mixing.StartPulay`.
- **Use a rather larger value for `scf.ElectronicTemperature`** in case of metallic systems. When `scf.ElectronicTemperature` is small, numerical instabilities appear often.
- **Increase `scf.Mixing.History`**. `scf.Mixing.History=25-40` could lead to the SCF convergence in most cases.

# Getting SCF No.3

Those parameters for 20 systems difficult to get the SCF convergence and the results can be found at

[http://www.jaist.ac.jp/~t-ozaki/large\\_example.tar.gz](http://www.jaist.ac.jp/~t-ozaki/large_example.tar.gz)

# Geometry optimization No.1

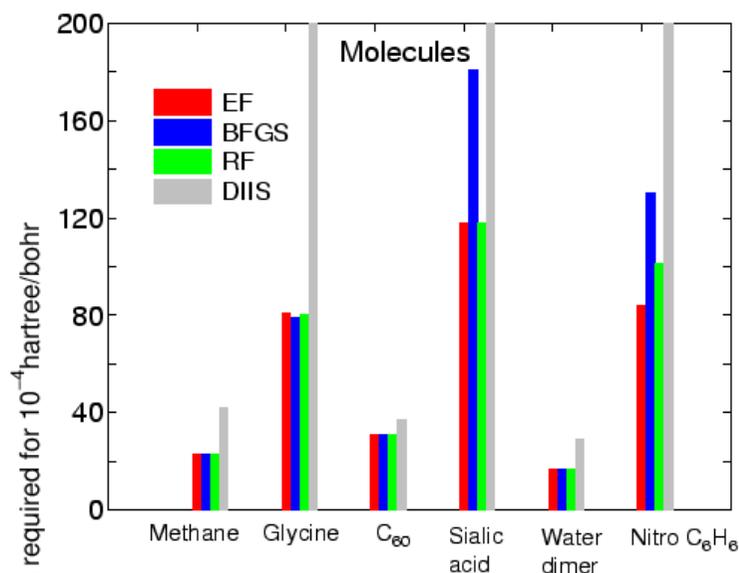
Five methods for geometry optimization are available, the **EF** is the best choice among them in most cases.

```
MD.Type                EF                # Opt|DIIS|BFGS|RF|EF
```

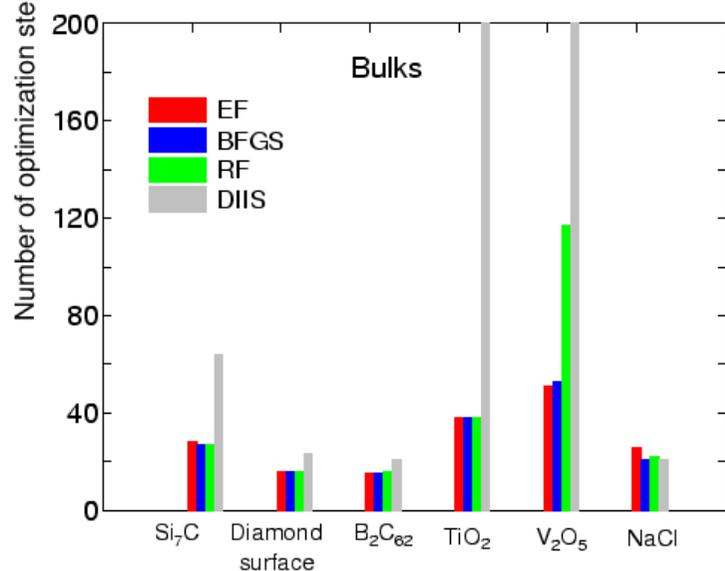
The **EF** is controlled by the following keywords:

```
MD.Opt.DIIS.History    3                # default=3
MD.Opt.StartDIIS       5                # default=5
MD.Opt.EveryDIIS       200              # default=200
MD.maxIter             100              # default=1
MD.Opt.criterion       1.0e-4           # default=0.0003 (Hartree/bohr)
```

# Geometry optimization No.2



The **EF** gives a faster convergence, while the **RF** is also a good choice.



For your convenience, the input files and out files used in the calculations shown in the figure can be found in "[openmx3.5/work/geoopt\\_example](#)".

# Geometry optimization No.3

If you **cannot** obtain the optimize geometry, try the following prescriptions:

- **Increase MD.Opt.DIIS.History**. MD.Opt.DIIS.History=10-20 could lead to the SCF convergence **in some cases (not all)**.
- **Use a rather larger value for MD.OptStart.DIIS**. Before starting the Pulay type mixing, achieve a convergence at some level. An appropriate value may be 10 to 30 for MD.OptStart.DIIS.
- **Use a rather larger value for MD.Opt.criterion**. There is a case that the maximum force does not decrease below  $10^{-4}$  Hartree/bohr. Then, a compromise is to increase MD.Opt.criterion to  $3.0 \times 10^{-4}$  Hartree/bohr or more.

# Initial spin moments

The initial spin moments can be controlled by the last **two columns** in the keyword **Atoms.SpeciesAndCoordinates**.

Example: MnO in the NaCl structure

## Ferromagnetic order

```
<Atoms.SpeciesAndCoordinates      # Unit=AU
 1 Mn  0.000  0.000  0.000      8.0  5.0
 2 Mn  4.1905 4.1905  0.000      8.0  5.0
 3 O   4.1905 0.000  0.000      3.0  3.0
 4 O   4.1905 4.1905  4.1905     3.0  3.0
Atoms.SpeciesAndCoordinates>
```

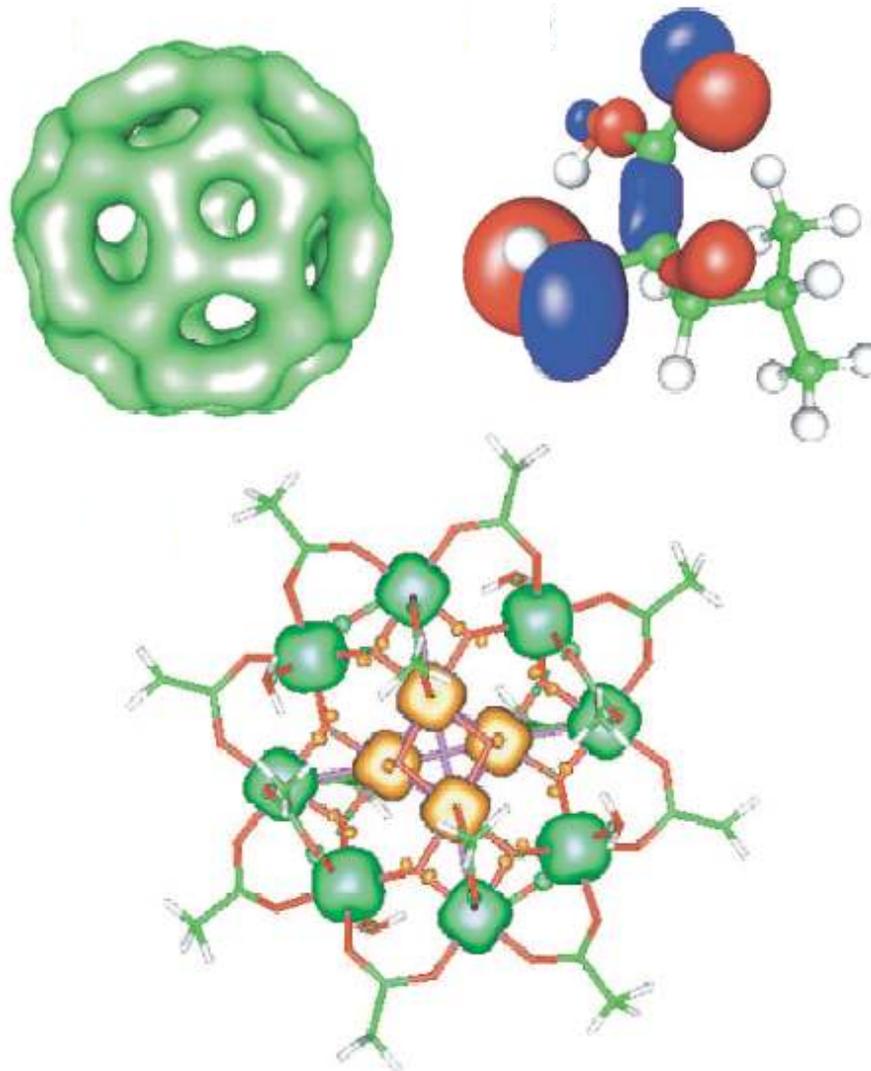
## Antiferromagnetic order

```
<Atoms.SpeciesAndCoordinates      # Unit=AU
 1 Mn  0.000  0.000  0.000      8.0  5.0
 2 Mn  4.1905 4.1905  0.000      5.0  8.0
 3 O   4.1905 0.000  0.000      3.0  3.0
 4 O   4.1905 4.1905  4.1905     3.0  3.0
Atoms.SpeciesAndCoordinates>
```

# Visualization of cube files

The generated **cube files** (\*.cube) can be visualized by many software such as

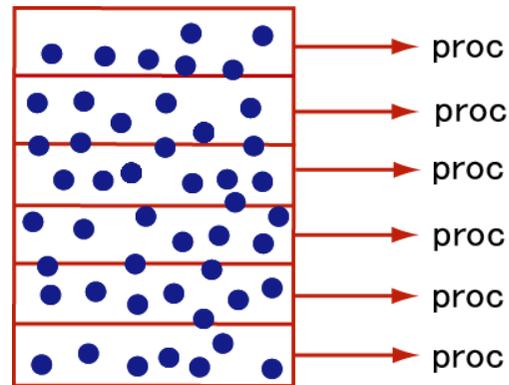
XCrysDen  
gOpenMol  
Molkel  
etc.



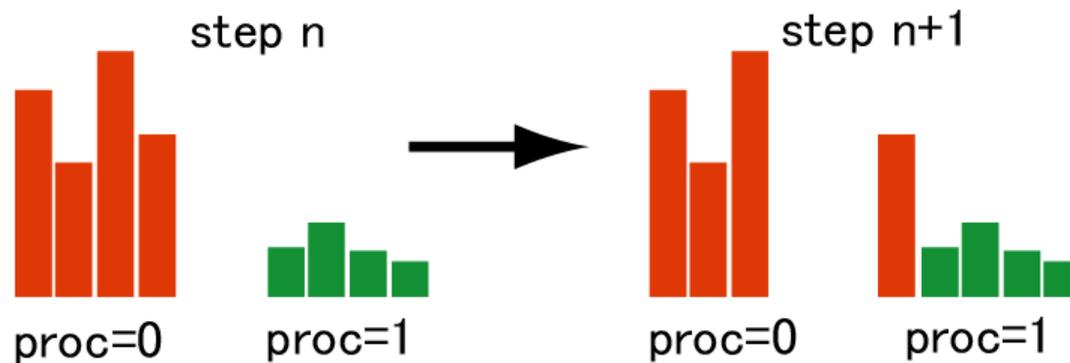
# Parallelization

- The parallelization is basically done by a 1D-domain decomposition.
- Also a different parallelization scheme is considered depending on the data structure in each subroutine.
- The dynamic load balancing is attempted at every MD step.

## 1-D domain decomposition



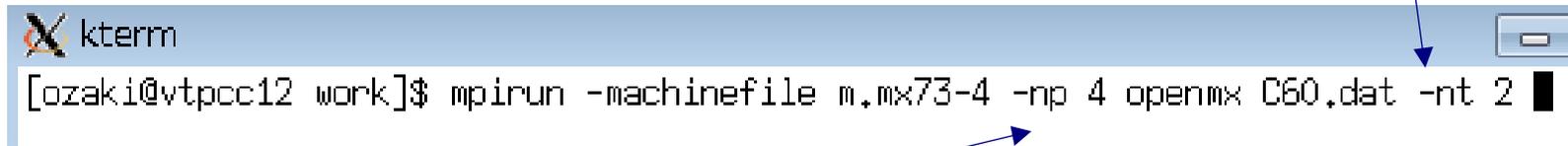
## Dynamic load balancing



# OpenMP/MPI hybrid parallelization

The numbers of processes and threads can be typically specified by the following command:

The number of threads for OpenMP parallelization

A terminal window titled 'kterm' with a blue header bar. The terminal text shows a command: [ozaki@vtpcc12 work]\$ mpirun -machinefile m.mx73-4 -np 4 openmx C60.dat -nt 2. A blue arrow points from the text 'The number of threads for OpenMP parallelization' to the '-nt 2' part of the command. Another blue arrow points from the text 'The number of processes for MPI parallelization' to the '-np 4' part of the command.

```
[ozaki@vtpcc12 work]$ mpirun -machinefile m.mx73-4 -np 4 openmx C60.dat -nt 2
```

The number of processes for MPI parallelization

Also the number of processes per node should be controlled by properly giving a machinefile which depends on computational environment.

# Parallel efficiency

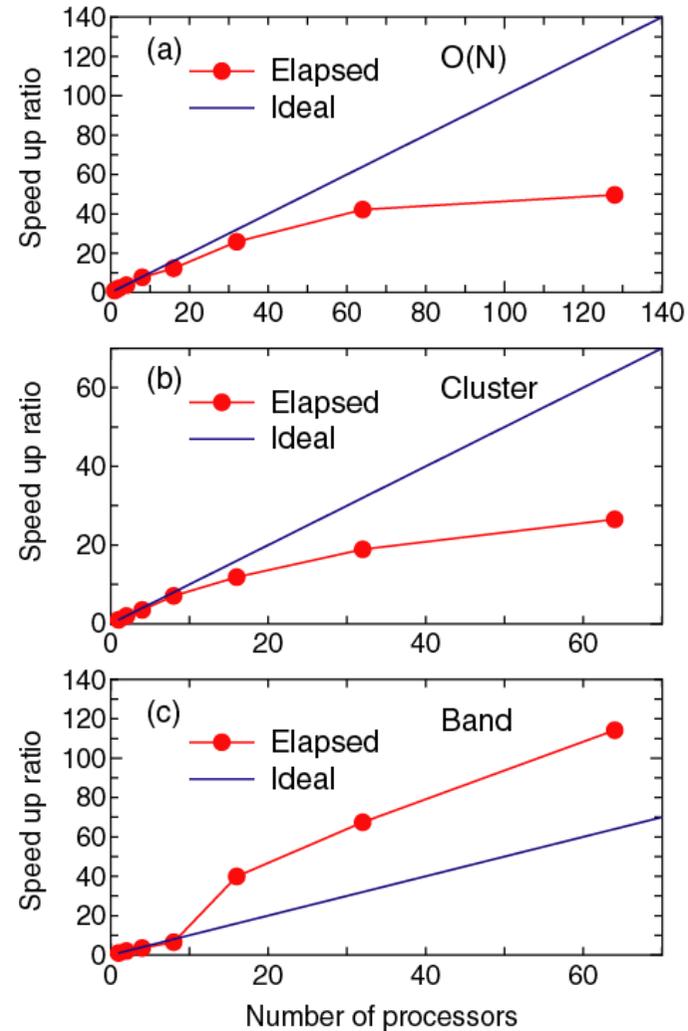
- (a) Diamond (512 atoms)
- (b) SMM (148 atoms)
- (c) Diamond (64 atoms,  
k-points=3x3x3)

Cray-XT3

2.4 GHz

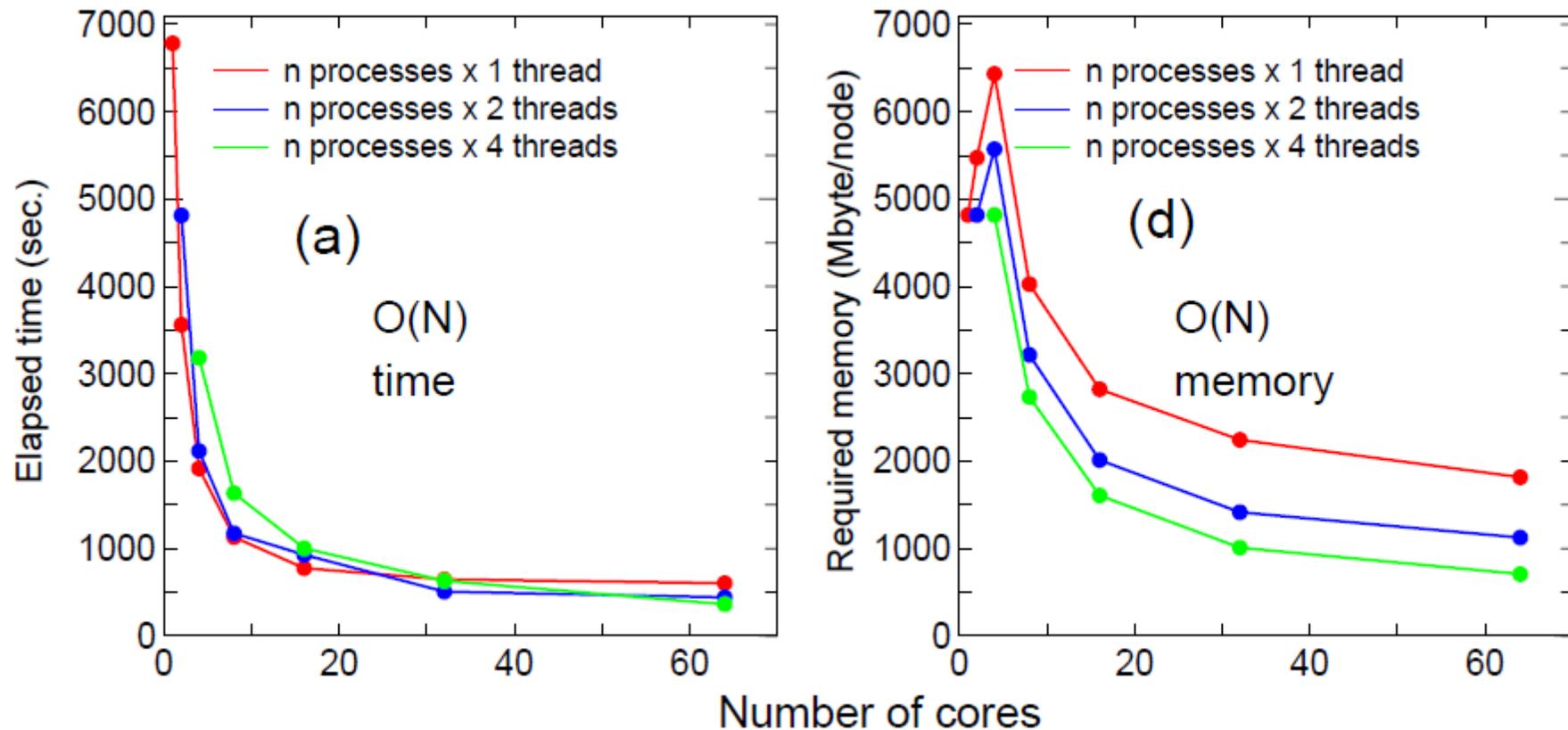
Interconnect

actual performance 1.0GB/s



# OpenMP/MPI hybrid parallelization

Almost all parts are parallelized by the hybrid method.



The memory size can be reduced by the hybrid method, while the parallel efficiency is comparable to the flat MPI.

# Other things

See the manual about

Density of states

Band dispersion

Charge analysis

Non-collinear DFT

Spin-orbit interaction

Linear scaling methods

Molecular dynamics

Applying electric field

Zeeman term

Electric polarization

etc.....

# OpenMX Forum

If you **cannot** solve your problem, please post your problem to the OpenMX Forum.