User's manual for Cluster construction model program

Zeyin Yan

May 9, 2022

This cluster construction program (called *Cry_DM* program) reproduces electron properties based on the CRYSTAL 14 *ab initio* package, it allows to conduct the one-electron reduced density matrix, Moyal function computations and cross-term analysis. This program is not an *ab initio*, it retrieves the geometry (cell parameters and atomic coordinates in the primitive cell), basis sets and population matrices (orbital coefficient is not available to be retrieved in CRYSTAL 14). An "iterative cluster" can be constructed that all the first neighbouring cells contributions (for 27 cells case) are accounted for the primitive cell (center cell) (shown as Figure 1). This program is coded in *Fortran*, using *OpenMP* parallel algorithm. It corresponds to the *Chapter 2* (theoretical framework) of the thesis "Reconstruction of momentum density and determination of one-electron reduced density matrix" by Zeyin YAN.

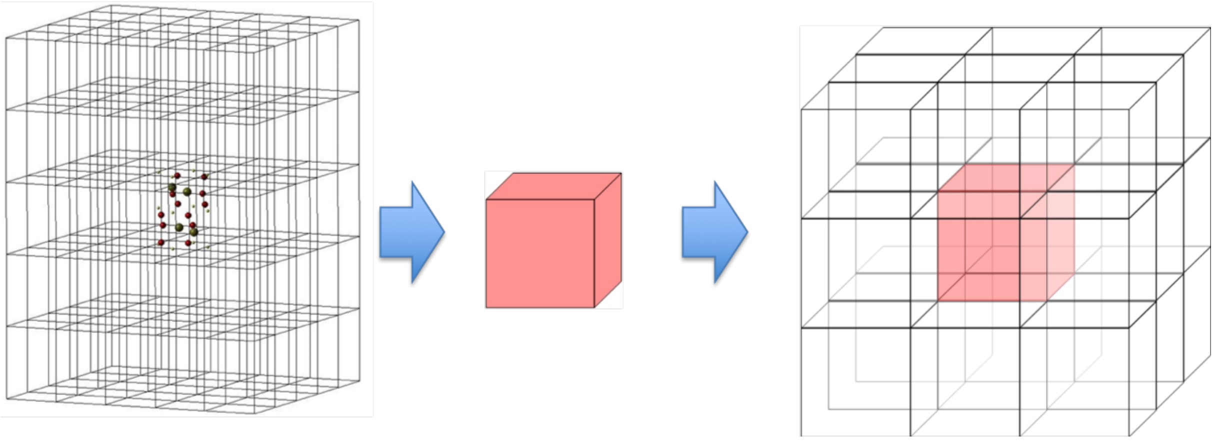


Figure 1: Cluster construction

=

1 Introduction

The first thing is the *ab initio* computation with CRYSTAL 14. According to the theoretical framework, the **geometry**, **basis sets** and **population matrices** are necessary to reproduce the electronic properties in the cluster. These information can be obtained by using the key word ***CRYAPI_OUT***, these data can be read after the treatment by the utility program *cryapi_inp* =[1] with modifications.
As more than 27 population matrices are retrieved, they represent the population matrices between the primitive cell and all the cells in the cluster ($P^{11}, P^{12}, \ldots, P^{1,27}$). However, for the cluster computation:

$$\Gamma(\mathbf{r}, \mathbf{r}') = \sum_{\mathbf{g},\mathbf{l}}^{N_{cells}} \Phi^{\mathbf{g}}_{Cart}(\mathbf{r}) P_c^{gl} \Phi^{\mathbf{l}}_{Cart}(\mathbf{r}') \tag{1}$$

the interactions between neighbouring cells should be accounted for. As cells are identical in the periodic computation, the population matrix between two cells is identical with the same relative cell positions. Therefore, G-vector is defined by a vector for the cells:

$$G = [m \quad k \quad l] \tag{2}$$

and the position of the cell origin can be expressed as

$$\mathbf{R_G} = m\mathbf{a} + k\mathbf{b} + l\mathbf{c} \tag{3}$$

where **a**, **b** and **c** are the Bravais lattice vectors, $G = [0 \quad 0 \quad 0]$ is the center cell position. The relative cell positions between cell i and j can be written as:

$$\mathbf{G}_{relative} = \mathbf{G(i)} - \mathbf{G(j)} \tag{4}$$

For the relative cell positions which do not exists in the cluster, between the primitive cell and other cells, the population matrices are approximated to 0.

2 File preparations

So, the necessary files for cluster construction program are **G_Vector**, **output** and **xxx.d3** files:

2.1 G_Vector

To obtain the G-vector list, two methods are proposed:

1 (**recommended**) In the printing option *SETPRINT*, the subroutine 59 can output the G-vector values. Insert the following cards in the block 3 of the SCF input file (.d12).

```
SETPRINT
1
59 N
```

where N is the number of G_Vector. As the G_Vector are output with other data (eg. P(g) matrix...), the out file can be very large. The advice is to use a simple crystal to generate the list, for example, using the same cell parameters, but put only a H atom in the center of the cell. Search the key word "*G=* " in the output file (eg: notepad++), the resulting lines are:

...

P(G) +++ G= n (m k l) +++

...

where *n* is the number of cell, and *m k l* is the G_Vector of cell. List the G-vector according to the ascending order of *n*.

[1] http://www.crystal.unito.it/documentation.php

=

2 Using following algorithm

As the G_Vector are based on the cell parameters, the list can be generated independently from the a, b, c, α, β and γ.

(a) Calculate the lattice vectors. The CRYSTAL14 convention is fixing the b axis, so

$$V = abc\sqrt{1 + 2cos(\alpha)cos(\beta)cos(\gamma) - cos^2(\alpha) - cos^2(\beta) - cos^2(\gamma)} \tag{5a}$$

$$\mathbf{a} = [a * sin(\gamma), b * cos(\gamma), 0] \tag{5b}$$

$$\mathbf{b} = [0, b, 0] \tag{5c}$$

$$\mathbf{c} = [\frac{c * (cos(\beta) - cos(\gamma) * cos(\alpha))}{sin(\gamma)}, c * cos(\alpha), \frac{V}{|\mathbf{a} \times \mathbf{b}|}] \tag{5d}$$

(b) List all possible G_Vector (m, k, $l \in \{-3, -2 - 1, 0, 1, 2, 3\}$), and calculate the norm values. As the G-vector space is center-symmetric, for m index only half will be accounted for ($m \in \{-3, -2 - 1, 0\}$) at the beginning. A 196 number G-vectors list with the module value $|G|$ will be obtained,

$$\begin{array}{cccc} -3 & -3 & -3 & |G_1| \\ \vdots & \vdots & \vdots & \vdots \\ m & k & l & |G_i| \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 3 & 3 & |G_{196}| \end{array}$$

(c) Rank the list with $|G|$ ascending order.

(d) Double the list, insert the $-G$ just behind the G.

(e) Unique the list (the G with 0 could be listed several times).

Here an example coded by *MATLAB*:

```
%********************************************************
clc
clear
close all
%input cell parameters
cell_a=5.69;
cell_b=7.609;
cell_c=5.335;
cell_alpha=90;
cell_beta=90;
cell_gamma=90;

%step a, Calculate the lattice vectors.
V_cell=cell_a*cell_b*cell_c*sqrt(1+2*cosd(cell_alpha)*...
    cosd(cell_beta)*cosd(cell_gamma)-cosd(cell_alpha)^2-...
    cosd(cell_beta)^2-cosd(cell_gamma)^2);

Vec_a=[cell_a*sind(cell_gamma),cell_a*cosd(cell_gamma),0];
Vec_b=[0,cell_b,0];
Vec_c=[cell_c*(cosd(cell_beta)-cosd(cell_gamma)*...
    cosd(cell_alpha))/sind(cell_gamma),cell_c*cosd(cell_alpha),...
    V_cell/norm(cross(Vec_a,Vec_b))];

%step b, List all possible G-vectors
G_vec_temps=zeros(196,4);
num_vec=0;
for m=-3:0
```

```
=
   for k=-3:3
      for l=-3:3
          num_vec=num_vec+1;
          G_vec_temps(num_vec,1:3)=[m k l];
          G_vec_temps(num_vec,4)=norm(Vec_a*m+Vec_b*k+Vec_c*l);
      end
   end
end

%step c, Rank the list
G_vec_temps=sortrows(G_vec_temps,4);

%step d, Double the list
G_vec_double=zeros(2*num_vec,3);

G_vec_double(1:2:2*num_vec-1,:)=G_vec_temps(:,1:3);
G_vec_double(2:2:2*num_vec,:)=-G_vec_temps(:,1:3);

%step e, Unique the list
G_vec=unique(G_vec_double,'rows','stable');
%*****************************************************
```

G_vec is the final G-Vector list.

Example:
Here is an example of **G_Vector** file (for $YTiO_3$ case):

```
  27
  0  0  0
  0  0 -1
  0  0  1
 -1  0  0
  1  0  0
  0 -1  0
  0  1  0
 -1  0 -1
  1  0  1
 -1  0  1
  1  0 -1
  0 -1 -1
  0  1  1
  0 -1  1
  0  1 -1
 -1 -1  0
  1  1  0
 -1  1  0
  1 -1  0
  0  0 -2
  0  0  2
 -1 -1 -1
  1  1  1
 -1 -1  1
  1  1 -1
 -1  1 -1
  1 -1  1
  ...
  ...
```

The first line is the number of cells in the cluster, it should agree with the number of population matrices. The following lines are the **G_Vector** (with ascending order).

=
*Tips: A long **G_Vector** list can be prepared at the beginning, only change the first line number if you want to focus on a larger cluster.*

2.2 Output file of CRYSTAL 14

To obtain the output file, the first step is to conduct the *ab initio* SCF calculation, a property computation is necessary to output the basic information using key word ***CRYAPI_OUT*** (see user's manual of CRYSTAL14). Here is the *out.d3* file:

```
CRYAPI_OUT
END
```

The useful information (but unformatted) is saved in *GRED.DAT* file. The **NEWK** can be inserted if you want to get the k points information and eigenvectors (saved in *KRED.DAT* file which is not needed for our computations).
With the *GRED.DAT* file, the useful information can be transferred to formatted file by the utility program *cryapi_inp* (called Cry_out.exe in our cases).
Therefore, to obtain the output file easily, I suggest to conduct the property and output computations together. Here is a example of script file (eg: in fusion):

```
#!/bin/bash

#PBS -S /bin/bash
#PBS -N out
#PBS -l walltime=00:20:00
#PBS -l select=n:ncpus=12:mpiprocs=12
#PBS -M xxx.xx@ecp.fr
#PBS -m abe
#PBS -q haswellq
#PBS -P project name

module load intel-compilers/17.0.4
module load sgi-mpt/2.14

cd $PBS_O_WORKDIR

#path of CRYSTAL14
CRYSTAL_BIN_PATH=/gpfs/opt/softwares/crystal14/1.0.4/bin

#path of cryapi_inp program
CRY_OUT=/workdir/xxx/xxx/Cry_out

#property computation
INPUT_PROPERTIES=/workdir/xxx/.../xxx/xxx.d3

ln -sf $INPUT_PROPERTIES INPUT
mpiexec_mpt -n 12*n $CRYSTAL_BIN_PATH/Pproperties 2> /workdir/xxx/.../xxx/xxx.out
rm -f *pe*

#output computation
$CRY_OUT/Cry_out.exe
```

As the output file is a type of log file which is saved with file name *out.oxxxx* (*out* comes from the name of task, the third line of script file), where *xxxx* is the task number in fusion. I suggest to rename this file into xxx.out. It contains geometry, basis set and multiples population matrices, the size of file can be hundreds MB.
For now, both G-vector list and output file are prepared, put both files in the same folder, so do the *xxx.d3* file to define the property computations and script file to submit the computation. Here is an example of script file:

=

```
#!/bin/bash

#PBS -S /bin/bash
#PBS -N property
#PBS -l walltime=xx:00:00
#PBS -l select=1:ncpus=24:mpiprocs=24:mem=10gb
#PBS -M xxx.xx@ecp.fr
#PBS -m abe
#PBS -q haswellq
#PBS -P project name

module load intel-compilers/17.0.4

cd $PBS_O_WORKDIR

#path of model construction program
CRYSTAL_DM_PATH=/workdir/xxx/xxx/DM_Cry


export OMP_NUM_THREADS=24
# command of program: xxx.exe xxx.out xxx.d3 U/R
$CRYSTAL_DM_PATH/Cry_DM.exe YTiO3_ECP_27.out ECHG_YTiO3_axial1.d3 U

# change the default result file names
#(necessary if you conduct multiple computations in the same script)
# for example the electron density computation
mv ECHG.data ECHG_axial1.data
mv ECHG_PATO.data ECHG_PATO_axial1.data
```

So the last thing is to define the xxx.d3 file, which use the habits of CRYSTAL 14 to define the property computations. These key words will be introduced in the next section.

3 Property keywords

3.1 Properties

ECHG - Electronic charge density maps

=

rec	variable	meaning
• ∗	0	the electron density (and spin density for open-shell case)
• ∗	NPY	number of points on the B-A segment
•	A keyword	enter a keyword to choose the type of coordinate
•	**COORDINA**	*the atomic unit is used to simplify the computation*
• ∗	XA, YA, ZA	cartesian coordinates of point A
• ∗	XB, YB, ZB	cartesian coordinates of point B
• ∗	XC, YC, ZC	cartesian coordinates of point C
		or
•	**ATOMS**	
• ∗	IA	label of the atom at point A
	AL, AM, AN	indices (direct lattice, input as reals) of the cell where atom is located
• ∗	IB	label of the atom at point B
	BL, BM, BN	indices (direct lattice, input as reals) of the cell where atom is located
• ∗	IC	label of the atom at point C
	CL, CM, CN	indices (direct lattice, input as reals) of the cell where atom is located
•	**RECTABGU**	definition of a new A'B'C'D' rectangular window (see Figure 2 and CRYSTAL14 manual)
•	**MARGINS**	definition of a new A"B"C"D" window including A'B'C'D'
• ∗	ABM	margins along AB
	CDM	margins along CD
	ADM	margins along AD
	BCM	margins along BC
•	**END**	end of input

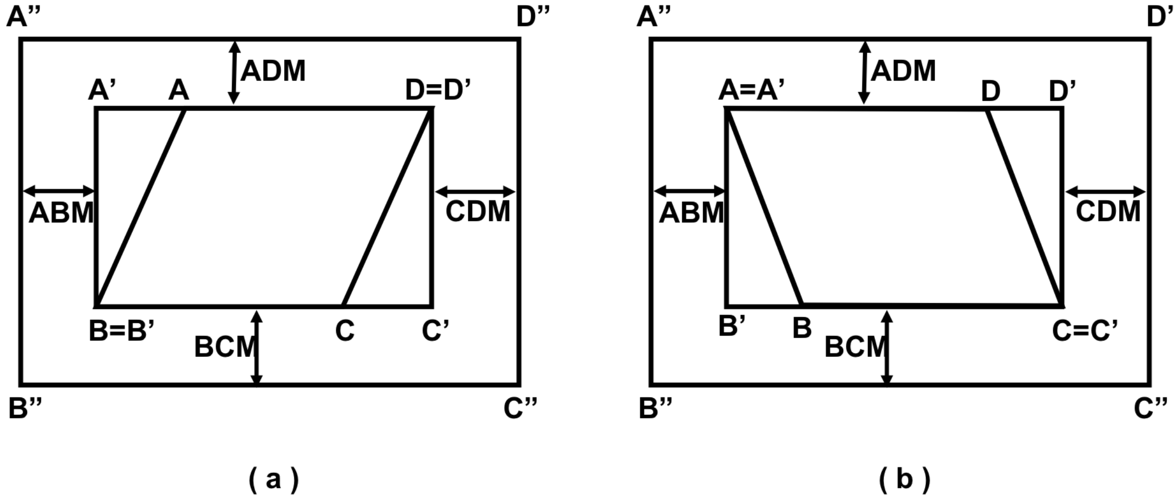


Figure 2: Definition of ECHG plane
(a) $\measuredangle$ ABC $\leq \frac{\pi}{2}$; (b)$\measuredangle$ ABC $\geq \frac{\pi}{2}$

Here is an example of *ECHG.d3* file:

```
ECHG
0
100
ATOMS
14 0 0 0
5 0 0 0
9 0 0 0
RECTANGU
```

```
=
MARGINS
3.5 1.5 1.5 3.5
END
```

The output files are ECHG.data and ECHG_PATO.data, the format as CRYSTAL 14. However, the PATO results are obtained by using Clement functions, for now there are only Y, Ti, O, C, N, H, S in the program.

Format in *Fortran* (the same as in CRYSTAL 14):

1ST RECORD :	-%-,IHFERM,TYPE,NROW,NCOL,DX,DY,COSXY	format: A3,I1,A4,2I5,3E12.5
2ND RECORD :	XA,YA,ZA,XB,YB,ZB	format: 1P,6E12.5
3RD RECORD :	XC,YC,ZC,NAF,LDIM	format: 1P,3E12.5,4X,2I4
4TH RECORD AND FOLLOWING :	((RDAT(I,J),I=1,NROW),J=1,NCOL)	format: 1P,6E12.5

Meaning of the variables:

1	'-%-'	3 character string marks the beginning of a block of data;
1	IHFERM:	0 : closed shell, insulating system 1 : open shell, insulating system
1	TYPE	4 characters string corresponding to the type of data "MAPN"
1	NROW	number of rows of the data matrix RDAT
1	NCOL	number of columns of the data matrix RDAT
1	DX	increment of x (Å) in the plane of the window
1	DY	increment of y (Å) in the plane of the window
1	COSXY	cosine of the angle between x and y axis;
2	XA,YA,ZA	coordinates of the points A,B (see keyword MAPNET) (Å)
2	XB,YB,ZB	defining the window where the functions is computed (Å)
3	XC,YC,ZC	coordinates of point C (Å)
3	NAF	number of atoms in the cell
3	LDIM	dimensionality (0 molecule; 1 polymer, 2 slab, 3 bulk), only 3 for our cases
4−>		ncol*nrow values of the function (a.u.) at the nodes of the grid

naf records follow, with atomic number, symbol, coordinates (Å) of the atoms in the cell:

NAT,SYMBAT,XA,YA,ZA format: I4,1X,A,1P,3E20.12

NAT	atomic number
SYMBAT	Mendeleev symbol
XA,YA,ZA	cartesian coordinates of the atoms in the cell (Å)

Cartesian components of cell parameters follow (Å)

AX, AY, AZ	cartesian component of vector a	format: 3E20.12
BX, BY, BZ	cartesian component of vector b	format: 3E20.12
CX, CY, CZ	cartesian component of vector c	format: 3E20.12

BIDIERD- Reciprocal form factors

The objective here is to calculate the directional Compton profile

=

rec	variable	meaning
• ∗	NDIR	number of directions
	NPU	number sampling points along each direction
	STEP	step along each direction
	IMODO 0:	the direction is defined by the Cartesian coordinates (bohr) of a point
	≠ 0:	the direction is defined by the atom label and indices of the cell where the atom is located
	ICASO 1:	the total density matrix is used
•	**PROF**	Compton profiles computed as Fourier transforms of the $B(r)$
• ∗	NPOIP	number of point along each direction
	STEPC	step along each direction (a.u.)
• ∗	**DIR**	the directions are specified
if $IMODO = 0$ insert NDIR records		
• ∗	X Y Z	the explored direction is defined by the straight line going from the origin to (X,Y,Z)
if $IMODO \neq 0$ insert NDIR records		
• ∗	I XG YG ZG	label of the atom and indices of the cell where the atom is located. The explored direction is defined by the straight line going from the origin to the atom position
•	**END**	end of input

Here is an example of *BIDIERD.d3* file:

```
BIDIERD
3 3000 0.01 0 1
PROF
1200 0.01
DIR
1 0 0
0 1 0
0 0 1
END
```

The output files are B_total.data, CP_total.data and B_spin.data, CP_spin.data for open-shell cases.
Format in *Fortran* :
1ST AND FOLLOWING: SP, RDAT(1, SP),..., RDAT(NDIR, SP) format: F6.2, $|NDIR|$F20.12

The file is a matrix (NPU × (NDIR+1) for B...data, and NPOIP × (NDIR+1) for CP...data)

Meaning of the variables:

SP :	s and p values for B(s) and J(p), respectively
RDAT(i, SP):	B($\mathbf{u}_i$, s) or J($\mathbf{u}_i$, p)

DENSMAT- One electron reduced density matrix

=

rec	variable	meaning
• ∗	NKN	number of knots in the path (=number of segments+1)
	NPU	number sampling points along each direction
	IPLOT 1	data are saved
	IMODO 0:	knots coordinates (x,y,z) in a.u.
	$\neq 0$:	knots are defined through atom labels
	FAC	prolongation factor for the first and last segments: $0 \sim 1$ (only for $IMODO \neq 0$ case) In Figure 3, $\frac{A'A}{AB} = \frac{DD'}{CD} = FAC$
if $IMODO = 0$ *insert NKN records*		
• ∗	X Y Z	Cartesian coordinates (bohr) of the ith knot
if $IMODO \neq 0$ *insert*		
• ∗	DX DY DZ	displacement (bohr, float type) applied to all atoms defining the path
insert NKN records		
• ∗	I XG YG ZG	label of the atom and indices of the cell where the atom is located
•	**END**	end of input

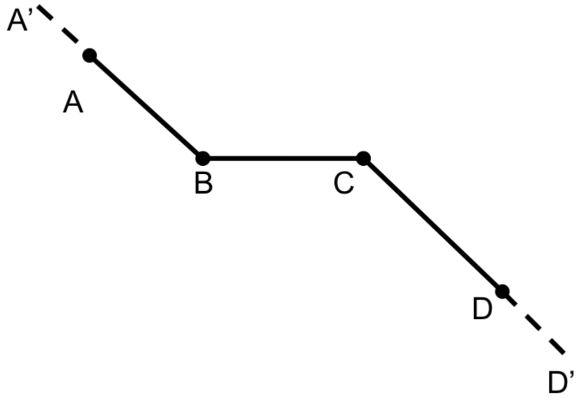


Figure 3: Prolongation factor for DENSMAT

Here is an example of *DENSMAT.d3* file:

```
DENSMAT
5 200 1 1 0.5
0. 0. 0.
12 1 1 0
8 0 0 0
9 0 0 0
5 0 0 0
12 1 0 0
END
```

The output files are DM_CRY.data.
Format in *Fortran* (I try to use the same format of ECHG):

1ST	RECORD :	-%-,IHFERM,TYPE,NROW,NCOL,DX,DY,COSXY	format: A3,I1,A4,2I5,3E12.5
2ND	RECORD :	XA,YA,ZA,XB,YB,ZB	format: 1P,6E12.5
3RD	RECORD :	XC,YC,ZC,NAF,LDIM	format: 1P,3E12.5,4X,2I4
4TH	RECORD		
AND FOLLOWING :		((RDAT(I,J),I=1,NROW),J=1,NCOL)	format: 1P,6E12.5

Meaning of the variables:

=

1	'-%-'	3 character string marks the beginning of a block of data;
1	IHFERM:	0 : closed shell, insulating system
		1 : open shell, insulating system
1	TYPE	4 characters string corresponding to the type of data "MAPN"
1	NROW	number of rows of the data matrix RDAT, NPU
1	NCOL	number of columns of the data matrix RDAT, the same with NROW
1	DX	increment of x (Å) in the plane of the window, step length
1	DY	increment of y (Å) in the plane of the window, the same with DX
1	COSXY	cosine of the angle between x and y axis, zero
2	XA,YA,ZA	coordinates of the points A,B (see keyword MAPNET) (Å), (0, (NROW-1)*DX, 0),
2	XB,YB,ZB	defining the window where the functions is computed (Å), (0, 0, 0)
3	XC,YC,ZC	coordinates of point C (Å), ((NROW-1)*DX, 0, 0)
3	NAF	number of atoms in the cell
3	LDIM	dimensionality (0 molecule; 1 polymer, 2 slab, 3 bulk), only 3 for our cases
4− >		ncol*nrow values of the function (a.u.) at the nodes of the grid

naf records follow, with atomic number, symbol, coordinates (Å) of the atoms in the cell:
NAT,SYMBAT,XA,YA,ZA format: I4,1X,A,1P,3E20.12

NAT	atomic number
SYMBAT	Mendeleev symbol
XA,YA,ZA	cartesian coordinates of the atoms in the cell (Å)

Cartesian components of cell parameters follow (Å)

AX, AY, AZ	cartesian component of vector a	format: 3E20.12
BX, BY, BZ	cartesian component of vector b	format: 3E20.12
CX, CY, CZ	cartesian component of vector c	format: 3E20.12

MOYAL2- Two-dimensional Moyal function

The 2D plane is defined by two vectors.

rec	variable	meaning
• ∗	NPU1	number of points along direction 1
	NPU2	number of points along direction 2
	STEP1	step along direction 1
	STEP2	step along direction 2
• ∗	**DIR**	the directions are specified
• ∗	SX1 SY1 SZ1	
	KX1 KY1 KZ1	the explored direction is defined by the vectors in phase space $(\vec{s},\vec{k})$
• ∗	SX2 SY2 SZ2	
	KX2 KY2 KZ2	
•	**END**	end of input

Here is an example of *MOYAL2.d3* file:

```
MOYAL2
100 100 0.05 0.05
DIR
0. 1. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0.
END
```

The output files are Moyal_Imag.dat and Moyal_Real.dat.
Format in *Fortran* (I try to use the same format of ECHG):

1ST	RECORD :	-%-,IHFERM,TYPE,NROW,NCOL,DX,DY, NAF	format: A3,I1,A4,2I5,2E12.5, I4
2ND	RECORD :	SX1, SY1, SZ1, KX1, KY1, KZ1	format: 1P,6E12.5
3RD	RECORD :	SX2, SY2, SZ2, KX2, KY2, KZ2	format: 1P,6E12.5
4TH	RECORD		
AND FOLLOWING :		((RDAT(I,J),I=1,NROW),J=1,NCOL)	format: 1P,6E12.5

=
Meaning of the variables:

1	'-%-'	3 character string marks the beginning of a block of data;
1	IHFERM:	0 : closed shell, insulating system 1 : open shell, insulating system
1	TYPE	4 characters string corresponding to the type of data "MAPN"
1	NROW	number of rows of the data matrix RDAT
1	NCOL	number of columns of the data matrix RDAT
1	DX	increment of x (Å) in the plane of the window
1	DY	increment of y (Å) in the plane of the window
1	NAF	number of atoms in the cell
2	SX1, SY1, SZ1, KX1, KY1, KZ1	unite vector of the first direction
3	SX2, SY12, SZ2, KX2, KY2, KZ2	unite vector of the second direction
4− >		ncol*nrow values of the function (a.u.) at the nodes of the grid

naf records follow, with atomic number, symbol, coordinates (Å) of the atoms in the cell:
NAT,SYMBAT,XA,YA,ZA format: I4,1X,A,1P,3E20.12

NAT	atomic number
SYMBAT	Mendeleev symbol
XA,YA,ZA	cartesian coordinates of the atoms in the cell (Å)

Cartesian components of cell parameters follow (Å)

AX, AY, AZ	cartesian component of vector a	format: 3E20.12
BX, BY, BZ	cartesian component of vector b	format: 3E20.12
CX, CY, CZ	cartesian component of vector c	format: 3E20.12

MOYAL- Directional Moyal function $A(\vec{s})_{\vec{k}}$

rec	variable	meaning
• ∗	NPU	number of point along each direction
	NDIR	number of direction along each direction
	STEP	step (a.u.) along each direction
• ∗	**DIR**	direction $\vec{s}$ is specified
	SX SY SZ	vector is defined by 3 float
• ∗	**LIST**	list of $\vec{k}$ (hkl) are specified
	h k l	NDIR records of hkl values
•	**END**	end of input

Here is an example of *MOYAL.d3* file:

```
MOYAL
100 8 0.05
DIR
1. 0. 0.
LIST
 0 0  0
 2 0  0
 4 0  0
 6 0  0
 8 0  0
10 0  0
12 0  0
14 0  0
END
```

The output files are Moyal_Imag.dat and Moyal_Real.dat.

=

Format in *Fortran* (I try to use the same format of ECHG):

1ST	RECORD :	-%-,IHFERM,TYPE,NROW,NCOL,DX, NAF	format: A3,I1,A4,2I5,E12.5,I4
2ND	RECORD :	XS,YS,ZS	format: 1P,3E12.5
3TH	RECORD		
AND FOLLOWING :		((RDAT(I,J),I=1,NROW),J=1,NCOL)	format: 1P,6E12.5

Meaning of the variables:

1	'-%-'	3 character string marks the beginning of a block of data;
1	IHFERM:	0 : closed shell, insulating system 1 : open shell, insulating system
1	TYPE	4 characters string corresponding to the type of data "MAPN"
1	NROW	number of rows of the data matrix RDAT, NPU
1	NCOL	number of columns of the data matrix RDAT, NDIR
1	DX	increment of x (Å) in the plane of the window, STEP
2	XS,YS,ZS	unite vector of **s**
3− >		ncol*nrow values of the function (a.u.) at the nodes of the grid

naf records follow, with atomic number, symbol, coordinates (Å) of the atoms in the cell:

NAT,SYMBAT,XA,YA,ZA   format: I4,1X,A,1P,3E20.12

NAT	atomic number
SYMBAT	Mendeleev symbol
XA,YA,ZA	cartesian coordinates of the atoms in the cell (Å)

Cartesian components of cell parameters follow (Å)

AX, AY, AZ	cartesian component of vector a	format: 3E20.12
BX, BY, BZ	cartesian component of vector b	format: 3E20.12
CX, CY, CZ	cartesian component of vector c	format: 3E20.12

3.2 Orbital resolved properties

The cross-term analysis is based on the orbital separation, the keywords are changed with a 'S' before such as 'SECHG', 'SBIDIERD', 'SDENSMAT', 'SMOYAL2' and 'SMOYAL', and number of orbitals and a list of orbitals should be defined at the end of *.d*3 file.

The cross-term contributions can be also calculated directly by the keywords with a 'C' before such as 'CECHG', 'CBIDIERD', 'CDENSMAT', 'CMOYAL2' and 'CMOYAL', and two lists of orbitals should be defined (with the number of orbitals respectively).

SECHG - Orbital resolved electronic charge density maps

=

rec	variable	meaning
• ∗	0	the electron density (and spin density for open-shell case)
• ∗	NPY	number of points on the B-A segment
•	A keyword	enter a keyword to choose the type of coordinate
•	**COORDINA**	*the atomic unit is used to simplify the computation*
• ∗	XA, YA, ZA	cartesian coordinates of point A
• ∗	XB, YB, ZB	cartesian coordinates of point B
• ∗	XC, YC, ZC	cartesian coordinates of point C
		or
•	**ATOMS**	
• ∗	IA	label of the atom at point A
	AL, AM, AN	indices (direct lattice, input as reals) of the cell where atom is located
• ∗	IB	label of the atom at point B
	BL, BM, BN	indices (direct lattice, input as reals) of the cell where atom is located
• ∗	IC	label of the atom at point C
	CL, CM, CN	indices (direct lattice, input as reals) of the cell where atom is located
•	**RECTABGU**	definition of a new A'B'C'D' rectangular window (see Figure 2 and CRYSTAL14 manual)
•	**MARGINS**	definition of a new A"B"C"D" window including A'B'C'D'
• ∗	ABM	margins along AB
	CDM	margins along CD
	ADM	margins along AD
	BCM	margins along BC
•	**ORBITAL**	list of orbitals are specified
• ∗	NUM	total number of orbitals
	I	insert the NUM orbital number
•	**END**	end of input

Here is an example of *SECHG.d3* file:

```
SECHG
0
100
ATOMS
14 0 0 0
5 0 0 0
9 0 0 0
RECTANGU
MARGINS
3.5 1.5 1.5 3.5
ORBITAL
276
1
2
...
276
END
```

CECHG - Orbital resolved electronic charge density maps

=

rec	variable	meaning
• ∗	0	the electron density (and spin density for open-shell case)
• ∗	NPY	number of points on the B-A segment
•	A keyword	enter a keyword to choose the type of coordinate
•	**COORDINA**	*the atomic unit is used to simplify the computation*
• ∗	XA, YA, ZA	cartesian coordinates of point A
• ∗	XB, YB, ZB	cartesian coordinates of point B
• ∗	XC, YC, ZC	cartesian coordinates of point C
		or
•	**ATOMS**	
• ∗	IA	label of the atom at point A
	AL, AM, AN	indices (direct lattice, input as reals) of the cell where atom is located
• ∗	IB	label of the atom at point B
	BL, BM, BN	indices (direct lattice, input as reals) of the cell where atom is located
• ∗	IC	label of the atom at point C
	CL, CM, CN	indices (direct lattice, input as reals) of the cell where atom is located
•	**RECTABGU**	definition of a new A'B'C'D' rectangular window (see Figure 2 and CRYSTAL14 manual)
•	**MARGINS**	definition of a new A"B"C"D" window including A'B'C'D'
• ∗	ABM	margins along AB
	CDM	margins along CD
	ADM	margins along AD
	BCM	margins along BC
•	**ORBITAL1**	list of orbitals are specified
• ∗	NUM1	total number of orbitals 1
	I	insert the NUM1 orbital number
•	**ORBITAL2**	list of orbitals are specified
• ∗	NUM2	total number of orbitals 2
	I	insert the NUM2 orbital number
•	**END**	end of input

Here is an example of *SECHG.d3* file:

```
SECHG
0
100
ATOMS
14 0 0 0
5 0 0 0
9 0 0 0
RECTANGU
MARGINS
3.5 1.5 1.5 3.5
ORBITAL1
276
1
2
...
276
ORBITAL2
100
277
278
...
376
END
```

The output files are ECHG.data and ECHG_PATO.data, the format as CRYSTAL 14.

=

SBIDIERD- Orbital resolved reciprocal form factors
The objective here is to calculate the directional Compton profile

rec	variable	meaning
• *	NDIR	number of directions
	NPU	number sampling points along each direction
	STEP	step along each direction
	IMODO 0:	the direction is defined by the Cartesian coordinates (bohr) of a point
	≠ 0:	the direction is defined by the atom label and indices of the cell where the atom is located
	ICASO 1:	the total density matrix is used
•	**PROF**	Compton profiles computed as Fourier transforms of the $B(r)$
• *	NPOIP	number of point along each direction
	STEPC	step along each direction (a.u.)
• *	**DIR**	the directions are specified
if $IMODO = 0$ *insert NDIR records*		
• *	X Y Z	the explored direction is defined by the straight line going from the origin to (X,Y,Z)
if $IMODO \neq 0$ *insert NDIR records*		
• *	I XG YG ZG	label of the atom and indices of the cell where the atom is located. The explored direction is defined by the straight line going from the origin to the atom position
•	**ORBITAL**	list of orbitals are specified
• *	NUM	total number of orbitals
	I	insert the NUM orbital number
•	**END**	end of input

Here is an example of *SBIDIERD.d3* file:

```
SBIDIERD
3 3000 0.01 0 1
PROF
1200 0.01
DIR
1 0 0
0 1 0
0 0 1
ORBITAL
276
1
2
...
276
END
```

CBIDIERD- Orbital resolved reciprocal form factors
The objective here is to calculate the directional Compton profile

=

rec	variable	meaning
• ∗	NDIR	number of directions
	NPU	number sampling points along each direction
	STEP	step along each direction
	IMODO 0:	the direction is defined by the Cartesian coordinates (bohr) of a point
	$\neq 0$:	the direction is defined by the atom label and indices of the cell where the atom is located
	ICASO 1:	the total density matrix is used
•	**PROF**	Compton profiles computed as Fourier transforms of the $B(r)$
• ∗	NPOIP	number of point along each direction
	STEPC	step along each direction (a.u.)
• ∗	**DIR**	the directions are specified
if $IMODO = 0$ *insert NDIR records*		
• ∗	X Y Z	the explored direction is defined by the straight line going from the origin to (X,Y,Z)
if $IMODO \neq 0$ *insert NDIR records*		
• ∗	I XG YG ZG	label of the atom and indices of the cell where the atom is located. The explored direction is defined by the straight line going from the origin to the atom position
•	**ORBITAL1**	list of orbitals are specified
• ∗	NUM1	total number of orbitals 1
	I	insert the NUM1 orbital number
•	**ORBITAL2**	list of orbitals are specified
• ∗	NUM2	total number of orbitals 2
	I	insert the NUM2 orbital number
•	**END**	end of input

Here is an example of *SBIDIERD.d3* file:

```
SBIDIERD
3 3000 0.01 0 1
PROF
1200 0.01
DIR
1 0 0
0 1 0
0 0 1
ORBITAL1
276
1
2
...
276
ORBITAL2
100
277
278
...
376
END
```

The output files are B_total.data, CP_total.data and B_spin.data, CP_spin.data for open-shell cases.

SDENSMAT- Orbital resolved one electron reduced density matrix

=

rec	variable	meaning
• ∗	NKN	number of knots in the path (=number of segments+1)
	NPU	number sampling points along each direction
	IPLOT 1	data are saved
	IMODO 0:	knots coordinates (x,y,z) in a.u.
	$\neq 0$:	knots are defined through atom labels
	FAC	prolongation factor for the first and last segments: $0 \sim 1$ (only for $IMODO \neq 0$ case)
		In Figure 3, $\frac{A'A}{AB} = \frac{DD'}{CD} = FAC$
if $IMODO = 0$ *insert NKN records*		
• ∗	X Y Z	Cartesian coordinates (bohr) of the ith knot
if $IMODO \neq 0$ *insert*		
• ∗	DX DY DZ	displacement (bohr, float type) applied to all atoms defining the path
insert NKN records		
• ∗	I XG YG ZG	label of the atom and indices of the cell where the atom is located
•	**ORBITAL**	list of orbitals are specified
• ∗	NUM	total number of orbitals
	I	insert the NUM orbital number
•	**END**	end of input

Here is an example of *SDENSMAT.d3* file:

```
SDENSMAT
5 200 1 1 0.5
0. 0. 0.
12 1 1 0
8 0 0 0
9 0 0 0
5 0 0 0
12 1 0 0
ORBITAL
276
1
2
...
276
END
```

CDENSMAT- Orbital resolved one electron reduced density matrix

=

rec	variable	meaning
• ∗	NKN	number of knots in the path (=number of segments+1)
	NPU	number sampling points along each direction
	IPLOT 1	data are saved
	IMODO 0:	knots coordinates (x,y,z) in a.u.
	$\neq 0$:	knots are defined through atom labels
	FAC	prolongation factor for the first and last segments: 0 ∼ 1 (only for $IMODO \neq 0$ case)
		In Figure 3, $\frac{A'A}{AB} = \frac{DD'}{CD} = FAC$
if $IMODO = 0$ *insert NKN records*		
• ∗	X Y Z	Cartesian coordinates (bohr) of the ith knot
if $IMODO \neq 0$ *insert*		
• ∗	DX DY DZ	displacement (bohr, float type) applied to all atoms defining the path
insert NKN records		
• ∗	I XG YG ZG	label of the atom and indices of the cell where the atom is located
•	**ORBITAL1**	list of orbitals are specified
• ∗	NUM1	total number of orbitals 1
	I	insert the NUM1 orbital number
•	**ORBITAL2**	list of orbitals are specified
• ∗	NUM2	total number of orbitals 2
	I	insert the NUM2 orbital number
•	**END**	end of input

Here is an example of *SDENSMAT.d3* file:

```
SDENSMAT
5 200 1 1 0.5
0. 0. 0.
12 1 1 0
8 0 0 0
9 0 0 0
5 0 0 0
12 1 0 0
ORBITAL1
276
1
2
...
276
ORBITAL2
100
277
278
...
376
END
```

The output files are DM_CRY.data.

SMOYAL2- Orbital resolved two-dimensional Moyal function

The 2D plane is defined by two vectors.

=

rec	variable	meaning
• ∗	NPU1	number of points along direction 1
	NPU2	number of points along direction 2
	STEP1	step along direction 1
	STEP2	step along direction 2
• ∗	**DIR**	the directions are specified
• ∗	SX1 SY1 SZ1	
	KX1 KY1 KZ1	the explored direction is defined by the vectors in phase space $(\vec{s}, \vec{k})$
• ∗	SX2 SY2 SZ2	
	KX2 KY2 KZ2	
•	**ORBITAL**	list of orbitals are specified
• ∗	NUM	total number of orbitals
	I	insert the NUM orbital number
•	**END**	end of input

Here is an example of *SMOYAL2.d3* file:

```
SMOYAL2
100 100 0.05 0.05
DIR
0. 1. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0.
ORBITAL
276
1
2
...
276
END
```

CMOYAL2- Orbital resolved two-dimensional Moyal function
The 2D plane is defined by two vectors.

rec	variable	meaning
• ∗	NPU1	number of points along direction 1
	NPU2	number of points along direction 2
	STEP1	step along direction 1
	STEP2	step along direction 2
• ∗	**DIR**	the directions are specified
• ∗	SX1 SY1 SZ1	
	KX1 KY1 KZ1	the explored direction is defined by the vectors in phase space $(\vec{s}, \vec{k})$
• ∗	SX2 SY2 SZ2	
	KX2 KY2 KZ2	
•	**ORBITAL1**	list of orbitals are specified
• ∗	NUM1	total number of orbitals 1
	I	insert the NUM1 orbital number
•	**ORBITAL2**	list of orbitals are specified
• ∗	NUM2	total number of orbitals 2
	I	insert the NUM2 orbital number
•	**END**	end of input

Here is an example of *SMOYAL2.d3* file:

```
SMOYAL2
100 100 0.05 0.05
DIR
0. 1. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0.
ORBITAL1
276
1
```

```
=
2
...
276
ORBITAL2
100
277
278
...
376
END
```

The output files are Moyal_Imag.dat and Moyal_Real.dat.

SMOYAL- Orbital resolved directional Moyal function $A(\vec{s})_{\vec{k}}$

rec	variable	meaning
• ∗	NPU	number of point along each direction
	NDIR	number of direction along each direction
	STEP	step (a.u.) along each direction
• ∗	**DIR**	direction $\vec{s}$ is specified
	SX SY SZ	vector is defined by 3 float
• ∗	**LIST**	list of $\vec{k}$ (*hkl*) are specified
	h k l	NDIR records of hkl values
•	**ORBITAL**	list of orbitals are specified
• ∗	NUM	total number of orbitals
	I	insert the NUM orbital number
•	**END**	end of input

Here is an example of *SMOYAL.d3* file:

```
SMOYAL
100 8 0.05
DIR
1. 0. 0.
LIST
 0 0  0
 2 0  0
 4 0  0
 6 0  0
 8 0  0
10 0  0
12 0  0
14 0  0
ORBITAL
276
1
2
...
276
END
```

CMOYAL- Orbital resolved directional Moyal function $A(\vec{s})_{\vec{k}}$

=

rec	variable	meaning
• ∗	NPU	number of point along each direction
	NDIR	number of direction along each direction
	STEP	step (a.u.) along each direction
• ∗	**DIR**	direction $\vec{s}$ is specified
	SX SY SZ	vector is defined by 3 float
• ∗	**LIST**	list of $\vec{k}$ (hkl) are specified
	h k l	NDIR records of hkl values
•	**ORBITAL1**	list of orbitals are specified
• ∗	NUM1	total number of orbitals 1
	I	insert the NUM1 orbital number
•	**ORBITAL2**	list of orbitals are specified
• ∗	NUM2	total number of orbitals 2
	I	insert the NUM2 orbital number
•	**END**	end of input

Here is an example of *SMOYAL.d3* file:

```
SMOYAL
100 8 0.05
DIR
1. 0. 0.
LIST
 0 0  0
 2 0  0
 4 0  0
 6 0  0
 8 0  0
10 0  0
12 0  0
14 0  0
ORBITAL1
276
1
2
...
276
ORBITAL2
100
277
278
...
376
END
```

The output files are Moyal_Imag.dat and Moyal_Real.dat.

4 Compilations

4.1 DM_Cry program

The makefile is prepared with the source code, to compile the program, the process is:

```
xxx@fusion ~]$ cd xx/xx/xx  #enter the folder of source code
```

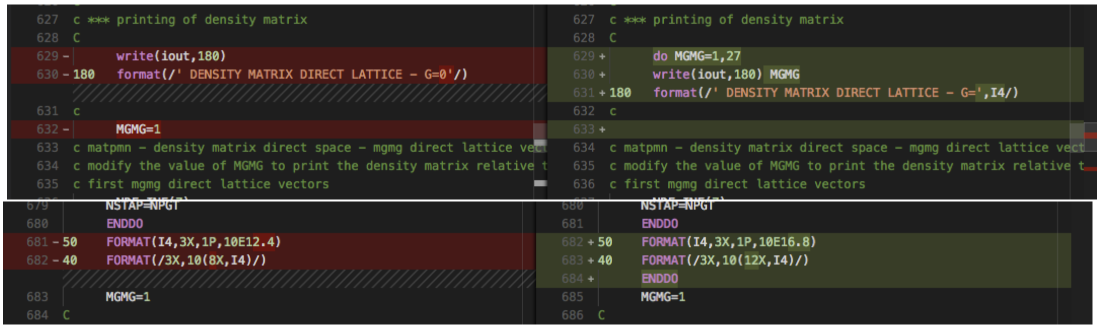
```
=
xxx@fusion DM_Cry]$ module load intel-compilers/17.0.4
xxx@fusion DM_Cry]$ make
ifort -qopenmp -c Fbase_Mod.f90 -o Fbase_Mod.o
ifort -qopenmp -c Reading_Mod.f90 -o Reading_Mod.o
ifort -qopenmp -c DM_Mod.f90 -o DM_Mod.o
ifort -qopenmp -c ECD_Mod.f90 -o ECD_Mod.o
ifort -qopenmp -c EMD_Mod.f90 -o EMD_Mod.o
ifort -qopenmp -c Clementi.f90 -o Clementi.o
ifort -qopenmp -c Clementi_DM.f90 -o Clementi_DM.o
ifort -qopenmp -c Bs_Mod.f90 -o Bs_Mod.o
ifort -qopenmp -c CP_BS_Mod.f90 -o CP_BS_Mod.o
ifort -qopenmp -c Wigner_Mod.f90 -o Wigner_Mod.o
ifort -qopenmp -c Moyal_Mod.f90 -o Moyal_Mod.o
ifort -qopenmp -c Parity_Mod.f90 -o Parity_Mod.o
ifort -qopenmp -c D3_Mod.f90 -o D3_Mod.o
ifort -qopenmp -c main.f90 -o main.o
ifort  -qopenmp Fbase_Mod.o Reading_Mod.o DM_Mod.o ECD_Mod.o EMD_Mod.o Clementi.o
Clementi_DM.o Bs_Mod.o CP_BS_Mod.o Wigner_Mod.o Moyal_Mod.o Parity_Mod.o D3_Mod.o main.o -o Cry_DM.e
```

As the new compiler should use the *qopenmp* rather than *openmp* for parallelism, it is possible to change the second and third lines of in $Makefile$.
If you want to recompile the program:

```
xxx@fusion DM_Cry]$ make clean
rm -f *.exe *.o *.mod *.dat
xxx@fusion DM_Cry]$ make
...
...
...
xxx@fusion DM_Cry]$ pwd
 /workdir/xxx/xxx/DM_Cry    #this the CRYSTAL_DM_PATH in the script file
```

4.2 Cry_out

The compile process is similar, here one point should be clear is that, this program can be downloaded directly from the site web. However, to conduct our computation, there is some lines should be modified. The original program output only the primitive information, to generalize to a lager cluster, the keyword $MGMG$ in cryapi_inp.f should be changed (as presented in Figure 4): a *do* cycle is added. For the future work, if a larger cluster (more than 27 cells) is need, just change the number 27 to N (up right in Figure 4), and also $G - vector$ file should also be change as previously presented (number in the first line).
In fact, there is another modification, the format for α (basis set parameter) is changed with more preciser number (as presented in Figure 5), it should be compatible with the format definition in $Reading_{M}od.f90$ in the cluster program (1120 format, line 48).



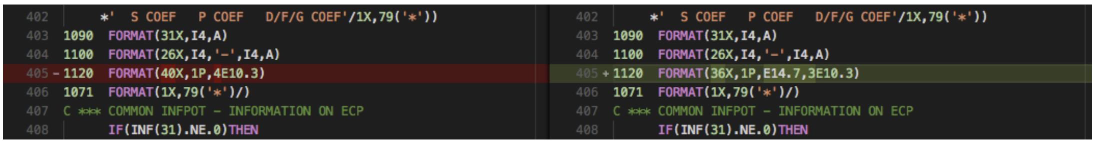
```
  c *** printing of density matrix
  C
-       write(iout,180)
- 180   format(/' DENSITY MATRIX DIRECT LATTICE - G=0'/)

  c
-       MGMG=1
  c matpmn - density matrix direct space - mgmg direct lattice vec
  c modify the value of MGMG to print the density matrix relative t
  c first mgmg direct lattice vectors
        NSTAP=NPGT
        ENDDO
- 50    FORMAT(I4,3X,1P,10E12.4)
- 40    FORMAT(/3X,10(8X,I4)/)

        MGMG=1
  C
```

```
  c *** printing of density matrix
  C
+       do MGMG=1,27
+       write(iout,180) MGMG
+ 180   format(/' DENSITY MATRIX DIRECT LATTICE - G=',I4/)
  c
+
  c matpmn - density matrix direct space - mgmg direct lattice vect
  c modify the value of MGMG to print the density matrix relative t
  c first mgmg direct lattice vectors
        NSTAP=NPGT
        ENDDO
+ 50    FORMAT(I4,3X,1P,10E16.8)
+ 40    FORMAT(/3X,10(12X,I4)/)
+       ENDDO
        MGMG=1
  C
```

Figure 4: Modifications in cryapi_inp.f



```
     *'  S COEF   P COEF   D/F/G COEF'/1X,79('*'))
1090  FORMAT(31X,I4,A)
1100  FORMAT(26X,I4,'-',I4,A)
- 1120  FORMAT(40X,1P,4E10.3)
1071  FORMAT(1X,79('*')/)
C *** COMMON INFPOT - INFORMATION ON ECP
      IF(INF(31).NE.0)THEN
```

```
     *'  S COEF   P COEF   D/F/G COEF'/1X,79('*'))
1090  FORMAT(31X,I4,A)
1100  FORMAT(26X,I4,'-',I4,A)
+ 1120  FORMAT(36X,1P,E14.7,3E10.3)
1071  FORMAT(1X,79('*')/)
C *** COMMON INFPOT - INFORMATION ON ECP
      IF(INF(31).NE.0)THEN
```

Figure 5: Modification of precision

≣