CRYSTAL Virtual Machine

Getting Started with CRYSTAL:
From Geometry to Single-Point Calculations

Giacomo Ambrogio

giacomo.ambrogio@unito.it

Advanced School on
QUANTUM MODELLING
of Materials with CRYSTAL

UNIVERSITÀ DI TORINO

Volta Redonda, State of Rio de Janeiro
Brazil

QMMC 2026

CRYSTAL Virtual Machine

Virtual box ⬇ : windows, Linux, Apple (intel)
for any x86 system

DOWNLOAD

1. **Import CRYSTAL_VM_QMMC26.ova** following these steps:
   - Go to **File → Import Appliance**
   - In the "Source" menu
     - Select the file CRYSTAL_VM_QMMC26.ova
   - In the "Settings" menu
     - [Optional] Choose your preferred **Machine Base Folder**
     - For **MAC Address Policy**, select:
       Generate new MAC addresses for all network adapters
     - For **Additional Options,** tick:
       Import Hard Drives as VDI
   - Click **Finish** and wait for the process to complete.
2. **S**elect the imported VM from the left menu and **start** it

Advanced School on
QUANTUM MODELLING
of Materials with CRYSTAL

UNIVERSITÀ
DI TORINO

CRYSTAL Virtual Machine

UTM ⬇️ : Apple silicon

DOWNLOAD

1. **Unzip**: MacOS_AppleSilicon.utm.zip
2. **Drag and drop** CRYSTAL_VM_MSSC2025.utm into UTM
3. Click the **Start** button in UTM

CRYSTAL Virtual Machine

Final Setup Steps:
Once the Virtual Machine (based on **Debian GNU/Linux 12**) starts:

1. Set **keyboard layout**:
   Use the top-right layout menu OR Navigate to:
   Activities → Settings → Keyboard

2. **Update** hands-on folder from **GitHub**
   - Open a terminal
   - Navigate to QMMC26 folder

   ```
   $ cd QMMC26
   ```

   - Update the repository

   ```
   $ git pull
   ```

Command Line Cheat Sheet

Files

Delete `<file>`
```
$ rm <file>
```

Delete `<directory>`
```
$ rm -r <directory>
```

Force-delete `<file>` (add `-r` for a directory)
```
$ rm -f <file>
```

Rename `<old-file>` to `<new-file>`
```
$ mv <old-file> <new-file>
```

Move `<file>` to `<directory>`
```
$ mv <file> <directory>
```

Copy `<file>` to `<directory>`
```
$ cp <file> <directory>
```

Copy `<directory1>` and its content to `<directory2>`
```
$ cp -r <directory1> <directory2>
```

Update file access and modification time (create `<file>` if it does not exist)
```
$ touch <file>
```

Directories

Display path of present working directory
```
$ pwd
```

Change directory to `<directory>`
```
$ cd <directory>
```

Navigate to parent directory
```
$ cd ..
```

List directory content
```
$ ls
```

List detailed directory contents, including hidden files
```
$ ls -la
```

Create new directory named `<new_directory>`
```
$ mkdir <new_directory>
```

Output

Output the contents of `<file>`
```
$ cat <file>
```

Output new lines of `<file>` as it grows
```
$ tail -f <file>
```

Output the contents of `<file>` using the less command (support for pagination)
```
$ less <file>
```

Output the first 10 lines of `<file>`
```
$ head <file>
```

Direct the output of `<cmd>` into `<file>`
```
$ <cmd> > <file>
```

Append the output of `<cmd>` to `<file>`
```
$ <cmd> >> <file>
```

Direct the output of `<cmd1>` to `<cmd2>`
```
$ <cmd1> | <cmd2>
```

Clear the command line window
```
$ clear
```

git

Update local folder
```
$ git pull
```

Clone an entire repository from `<url>`
```
$ git clone <url>
```

Get latest state from remote repository
```
$ git fetch origin
```

Reset local branch to exactly match the remote `main` branch
```
$ git reset -hard origin/main
```

Remove all untracked files, directories and ignored files
```
$ git clean -fdx
```

CRYSTAL

Run a CRYSTAL calculation (omit `<file>` extensions)
```
$ runCRY23 <file>
```

Run a CRYSTAL calculation with a restart option (omit `<file>` and `<restart_file>` extensions)
```
$ runCRY23 <file> <restart_file>
```

Run a PROPERTIES calculation (omit `<file>` and `<restart_file>` extensions)
```
$ runPROP23 <file> <restart_file>
```

Search

Find all files named `<file>` inside `<dir>`
```
$ find <dir> -name "<file>"
```

Output all occurrences of `<text>` inside `<file>` (add `-i` for case insensitive)
```
$ grep "text" <file>
```

Networking

Ping `<host>` and display status
```
$ ping <host>
```

Output `whois` information for `<host>`
```
$ whois <file>
```

Establish an ssh connection to `<host>` with user `<username>`
```
$ ssh <username>@<host>
```

Copy `<file>` to a remote `<host>`
```
$ scp <file> <username>@<host>://remote/path
```

Useful Links

General resources:

🌐 Website

📖 Manual

💬 Forum

Tutorials

Getting started:

▶️ A quick tour of CRYSTAL

▶️ Introduction

Tutorials

Input details:

▶️ Geometry

▶️ Basis Set

▶️ Functionals & Others

▶️ Total energy calculation

Outline

- How to **Run the Code**

- Understand CRYSTAL **input structure**
  - ○ Explore **geometry**, **basis set** and **DFT approximations**

- **Visualize** geometry with external tools

- **Run a first single-point calculation**

- **Exercises**:
  - Construct input for **Silicon**
  - Getting confident with DFT functionals
    + Self-consistent Hybrids
  - Improve your calculation

How to Run CRYSTAL

The CRYSTAL suite is composed of multiple **different executables**:

`crystal`   `properties`

`Pcrystal`   `Pproperties`

`MPPcrystal`

- All versions of **Crystal** are used for the **main computational drivers**
  E.g. Energy and wavefunction, geometry optimizations, frequencies, etc.

- **Properties** is used for **post-processing** of wavefunction (obtained with Crystal)

- Different executables for **different parallelization schemes**

  More on Thursday!

- In the VM we will **only use** the **serial** versions:
  `crystal` and `properties`

How to Run CRYSTAL

- The code needs to be executed from **command line** in a **terminal** shell

- Open a terminal (Navigate to: Activities → Terminal), and enter in 📁 testrun folder using command `cd`

```
$ cd QMMC2026/testrun
```

- We can run the code with the example input file (MgO) called 📋 **test.d12**

```
$ crystal < test.d12
```

How to Run CRYSTAL

- **You have successfully executed your first crystal calculation!**

- However, look at the content of the testrun folder with command `ls`

```
$ ls
```

- There are lots of **temporary files** created by the code (used during the execution) and some **output files**

- To do things in a **clean** way usually we use **scripts** in order to:
  1. generate a **temporary directory**,
  2. move there the input file,
  3. **execute** the code,
  4. and **copy back only** the necessary **ouputs**.

- You can write your own script or **use a premade one**:

```
$ runCRY23 test
```

From now on, always do this

General Structure of the INPUT

As a convention, all CRYSTAL **inputs** have **extension .d12**
The output file usually are **.out**

- Now **navigate** to the following **folder** in the **VM**:
  📁 QMMC2026/input_and_scf
- **Open** the file 📋 **mgo_input.d12**

```
$ open mgo_input.d12
```

- We use **keywords** to control the code execution
  They must be inserted in the *correct* position

- The input file is divided in **3 MAIN BLOCKS** (each closed with an **END** keyword)
  - **GEOMETRY**
  - **BASIS SET**
  - **Methods** and **Parameters**

The **Geometry** Section

d12

```
MgO bulk
CRYSTAL
0 0 0
225
4.21
2
12 0.      0.      0.
 8 0.5     0.5     0.5
END
...
```

1. **Title** section (just a comment, not read by the code)
2. **Dimensionality** of the system:
    CRYSTAL, **SLAB**; **POLYMER** and **MOLECULE**,
    for 3D, 2D, 1D and 0D, respectively.
3. Crystallographic information (3D only), three integer numbers for:
    *   convention for the space group identification:
        sequential number (0) or alphanumeric code (1).
    *   type of cell for rhombohedral groups: hexagonal (0) or rhombohedral (1).
    *   setting of the origin.
4. **Space group**, check Symmetry groups p. 418 of the manual.
5. **Lattice parameters**: the minimal set of crystallographic cell parameters is indicated (in Angstrom and degrees).
    E.g. For cubic systems $a = b = c$ and $\alpha = \beta = \gamma = 90°$. Thus, only a is required.
    Note that CRYSTAL works on the **primitive** cell.
6. Number of **non-equivalent atoms** in the cell and atomic **position** (coordinates in **fractional units**).
7. **END**: Closing the geometry input section.

Geometry from Files

Inputs can be built starting from common **structure files**.
Most used formats: **.xyz** (molecules) and **.cif** (crystals)

Let's see an example for a .xyz file:

xyz

```
3
Water molecule
O  0.00000   0.00000    0.11779
H  0.00000   0.75545   -0.47116
H  0.00000  -0.75545   -0.47116
```

d12

```
Water
MOLECULE
15                      ! C2v
3
8  0.00000    0.00000    0.11779
1  0.00000    0.75545   -0.47116
1  0.00000   -0.75545   -0.47116
END
```

Geometry from Files

And an example for a .cif file:

cif

```
data_MgO
_symmetry_space_group_name_H-M    Fm-3m
_cell_length_a    4.25648400
_cell_length_b    4.25648400
_cell_length_c    4.25648400
_cell_angle_alpha    90.00000000
_cell_angle_beta    90.00000000
_cell_angle_gamma    90.00000000
_symmetry_Int_Tables_number    225
_chemical_formula_structural    MgO
_chemical_formula_sum    'Mg4 O4'
_cell_volume    77.11751306
_cell_formula_units_Z    4
loop_
_symmetry_equiv_pos_site_id _symmetry_equiv_pos_as_xyz
1  'x, y, z'
2  '-x, -y, -z'
…
loop_
_atom_site_type_symbol
_atom_site_label
_atom_site_symmetry_multiplicity
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_occupancy
Mg  Mg0  4  0.00000000  0.00000000  0.00000000  1
O   O1   4  0.00000000  0.00000000  0.50000000  1
```

d12

```
MgO bulk
CRYSTAL
0 0 0
225
4.25648400
2
12  0.    0.    0.
 8  0.    0.    0.5
END
```

Honourable mention:
Editor in cif
Dr² Jefferson Maul
(From Brazil 🇧🇷)

Geometry from Database

The [materials project ↗](#) :
a very useful tool for searching materials and structures

Run a Geometry Test

Before running a full calculation, it's **good practice** to check the geometry.

- CRYSTAL provides the keyword:

    TESTGEOM ➡️ stop after checking the input geometry

- Let's run some testgeoms

```
$ cd QMMC2026/input_and_scf/testgeom
$ ls
SN_polymer.d12  benzene_molecule.d12  graphene_2D.d12
graphite_3D.d12  methane_molecule.d12  mgo.d12  outputs
urea_bulk.d12  water_poymer.d12

$ runCRY23 benzene_molecule
```
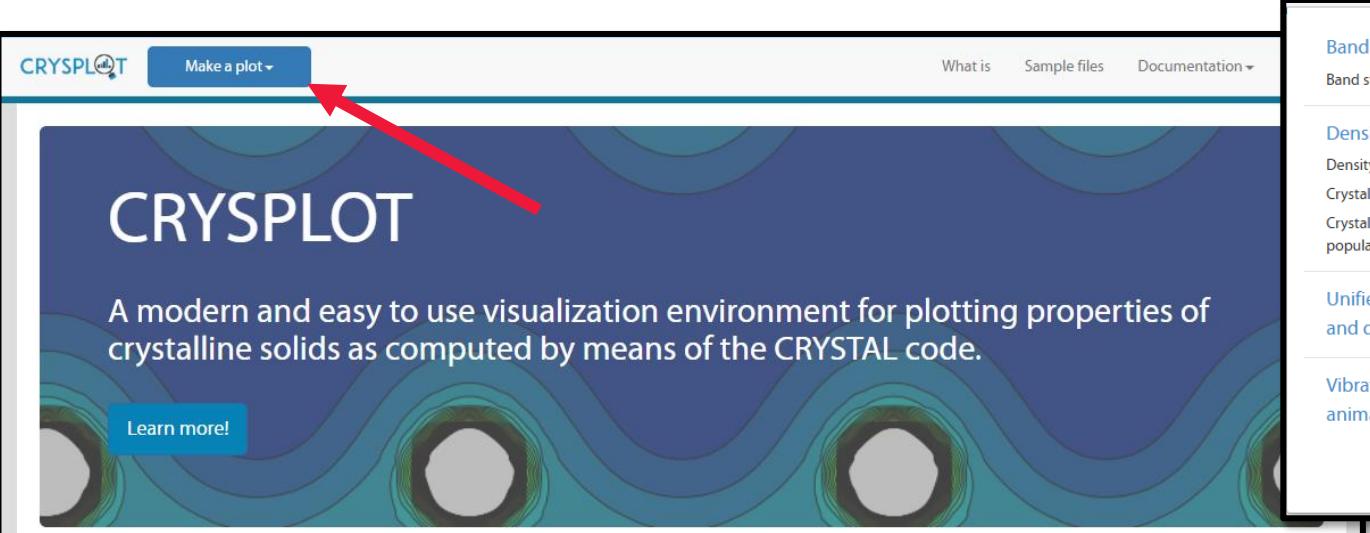
Visualize the Output and Structure

Initially, just look at the output file (.out)

```
$ open benzene_molecule
```

- Try to locate the unit cell structure, all the atoms with their position and the symmetry operators.

- Now visualize the structure with an **external software:**
  - **CRYSPlot** ↗ **:** online tool for quick visualization of CRYSTAL output

Visualize the Output and Structure

There are many **different possibility** to visualize the structure of your testgeom:

- **CRYSplot**: quick, online, fully compatible with Crystal

- **Moldraw**: integrated with crystal, but old GUI

- **Jmol/JsMol**: Cross-platform and lightweight, but limited for advanced density visualization

- **OVITO**: Great for extended systems and animations, not designed for molecular orbitals

- **VESTA**: Excellent for crystals, densities, isosurfaces, quite advanced, requires some setup

We are going to use this!
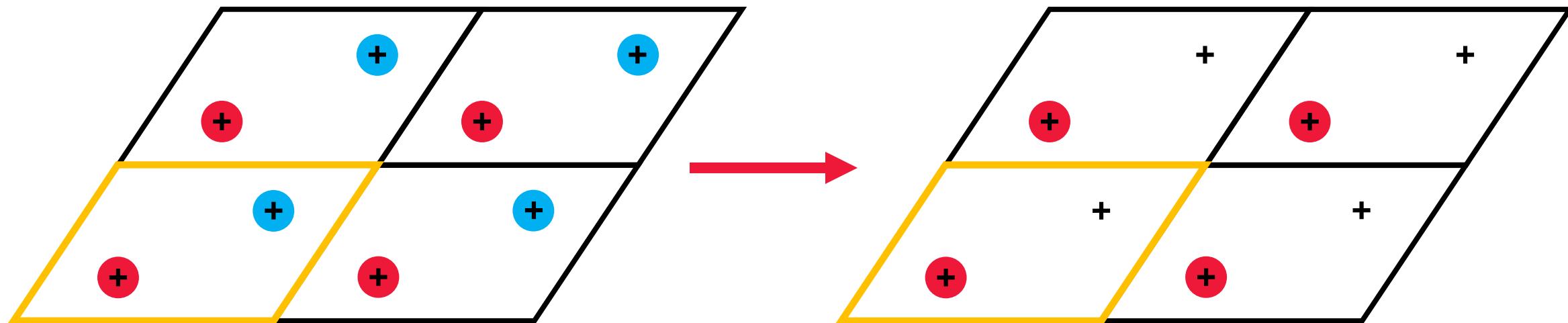
Geometry Editing

After the geometry input, before closing the first block, additional keywords can be used.

- The main functionalities here are the **geometry editing tools**
    The full list is available in the manual

- As an example, we will now create a *defective system*
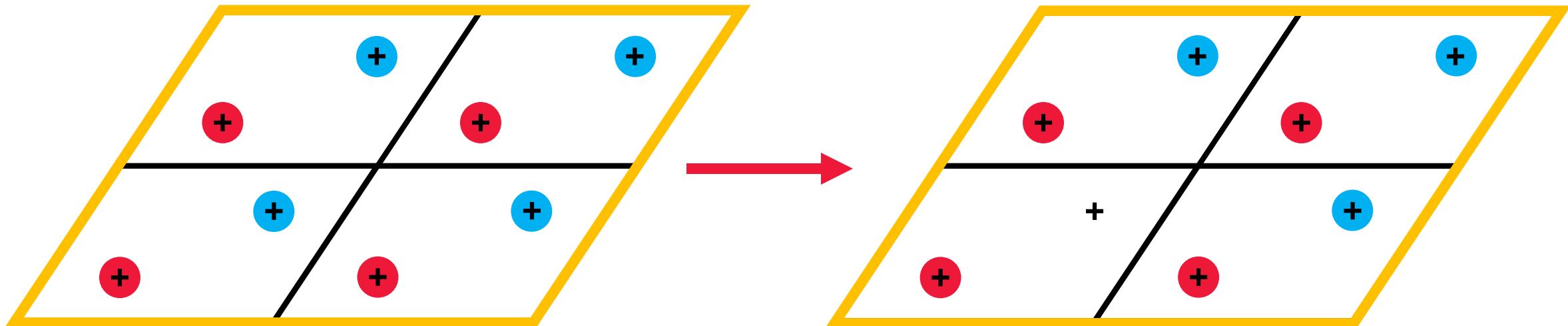
Defective Systems

- To simulate a defect, we will need to increase the size of the system with a **SUPERCELL**

- Then we will remove an atom

Defective Systems

- To simulate a defect, we will need to increase the size of the system with a **SUPERCELL**

- Then we will remove an atom

Defective Systems

- First create a copy of mgo.d12, then let's edit it.

```
$ cp mgo.d12 mgo_defect.d12
```

- Add the **SUPERCELL** keyword
  - Needs the *expansion matrix*

d12

```
MGO BULK
CRYSTAL
0 0 0
225
4.217
2
12  0.     0.     0.
 8  0.5    0.5    0.5
TESTGEOM
END
```

d12

```
MGO BULK
CRYSTAL
0 0 0
225
4.217
2
12  0.     0.     0.
 8  0.5    0.5    0.5
SUPERCELL
2 0 0
0 2 0
0 0 2
TESTGEOM
END
```

Let's just create a
2x2x2 sized supercell

Defective Systems

- Now **run the calculation** and search in the output the new **list of atoms**
- We need to **locate** one oxygen that will be removed (the label)
- Use the `ATOMREMO` keyword

  d12

  - Needs to specify how many atoms to remove
  - Then the list of labels of the atoms

- Look at the resulting geometry in CRYSPlot

```
MGO BULK
CRYSTAL
0 0 0
225
4.217
2
12  0.      0.      0.
 8  0.5     0.5     0.5
SUPERCELL
2 0 0
0 2 0
0 0 2
ATOMREMO
1
9
TESTGEOM
END
```

Other keywords

- Full list available in the **User manual**

Geometry keywords

Symmetry information

Keyword	Description	Page	
ATOMSYMM	printing of point symmetry at the atomic positions	43	–
MAKESAED	printing of symmetry allowed elastic distortions (SAED)	54	–
PRSYMDIR	printing of displacement directions allowed by symmetry.	69	–
SYMMDIR	printing of symmetry allowed geom opt directions	78	–
SYMMOPS	printing of point symmetry operators	78	–
TENSOR	print tensor of physical properties up to order 4	78	I

Symmetry information and control

Keyword	Description	Page	
BREAKELAS	symmetry breaking according to a general distortion	44	I
BREAKSYM	allow symmetry reduction following geometry modifications	44	–
KEEPSYMM	maintain symmetry following geometry modifications	54	–
MODISYMM	removal of selected symmetry operators	55	I
PURIFY	cleans atomic positions so that they are fully consistent with the group	69	–
SYMMREMO	removal of all symmetry operators	78	–
TRASREMO	removal of symmetry operators with translational components	78	–

Modifications without reduction of symmetry

Keyword	Description	Page	
ATOMORDE	reordering of atoms in molecular crystals	41	–
NOSHIFT	no shift of the origin to minimize the number of symmops with translational components before generating supercell	64	–
ORIGIN	shift of the origin to minimize the number of symmetry operators with translational components	64	–
PRIMITIV	crystallographic cell forced to be the primitive cell	68	–
ROTCRY	rotation of the crystal with respect to the reference system cell	70	I

Atoms and cell manipulation - possible symmetry reduction (BREAKSYMM)

Keyword	Description	Page	
ATOMDISP	displacement of atoms	41	I
ATOMINSE	addition of atoms	41	I
ATOMREMO	removal of atoms	42	I
ATOMROT	rotation of groups of atoms	42	I
ATOMSUBS	substitution of atoms	43	I
ELASTIC	distortion of the lattice	47	I
POINTCHG	point charges input	68	I
SCELCONF	generation of supercell for configuration counting	73	I
SCELPHONO	generation of supercell for phonon dispersion	73	I
SUPERCEL	generation of supercell - input refers to primitive cell	75	I
SUPERCON	generation of supercell - input refers to conventional cell	75	I
USESAED	given symmetry allowed elastic distortions, reads δ	78	I

From crystals to slabs (3D→2D)

Keyword	Description	Page	
SLABINFO	definition of a new cell, with $xy \parallel$ to a given plane	72	I
SLABCUT	generation of a slab parallel to a given plane (3D→2D)	71	I

From slabs to single and multi-wall nanotubes (2D→1D)

Keyword	Description	Page	
NANOTUBE	building a nanotube from a slab	59	I
SWCNT	building a nanotube from an hexagonal slab	77	I
NANOMULTI	building a multi-wall nanotube from a slab	62	I

From periodic structures to clusters

Keyword	Description	Page	
CLUSTER	cutting of a cluster from a periodic structure (3D→0D)	45	I
CLUSTSIZE	maximum number of atoms in a cluster	46	I
FULLE	building a fullerene from an hexagonal slab (2D→0D)	52	I
HYDROSUB	border atoms substituted with hydrogens (0D→0D)	54	I

Basis Set Block

- Second block of the input is the basis set.
- Need to specify a basis for **each atomic species** (or more) that you have
- Basis set can be easily found on **databases** as the **Crystal Basis Set Library** ↗

- Let's **add a basis set** to the **mgo_input.d12:**
  - Use the **pob_DZVP_rev2** from the Crystal library
  - Both for **Mg** and **O**

Basis Set Block

pob-DZVP-rev2

$$\chi_\mu(\mathbf{r}) = \sum_{j}^{N_G} d_{\mu j} G(\mathbf{r}; \alpha_{\mu j})$$

$$G(\mathbf{r}; \alpha_\mu) = N\, Y(\theta, \varphi)\, r^{2\mathrm{n}-\mathrm{l}-2} e^{-\alpha_{\mu j} r^2}$$

Mg

```
12 7
0 0 5 2 1
   4953.8339196        -0.0057778967498
   745.18044154        -0.0431247610820
   169.21604972        -0.1926821698700
   47.300672019        -0.4864143911600
   14.461336973        -0.4255089407700
0 0 3 2 1
   24.768174789         0.0879569699840
   2.4940945349        -0.5516505812800
   0.8780758453        -0.5344329483300
0 0 1 2 1
   0.3450688700         1.0000000000000
0 0 1 0 1
   0.1500539900         1.0000000000000
0 2 5 6 1
   98.053010494        -0.0144805646010
   22.586932277        -0.0954957507870
   6.8391509842        -0.3078767265100
   2.2332843818        -0.4993629288600
   0.7160659939        -0.3150347621300
0 2 1 0 1
   0.2469232300         1.0000000000000
0 3 1 0 1
   0.4148382900         1.0000000000000
```

s shell, 5 primitives, 2 electrons

s shell, 2 primitives, 2 electrons **Split-Valence**
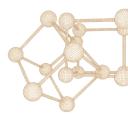
s shell, 1 primitive, 2 electrons
s shell, 1 primitive, 0 electrons **Double Zeta**

p shell, 5 primitives, 6 electrons

Double Zeta

p shell, 1 primitive, 0 electrons

d shell, 1 primitive, 0 electrons **+ polarization**

Basis Set Block

Mg

```
12 7
0 0 5 2 1
    4953.8339196        -0.0057778967498
    745.18044154        -0.0431247610820
    169.21604972        -0.1926821698700
    47.300672019        -0.4864143911600
    14.461336973        -0.4255089407700
0 0 3 2 1
    24.768174789         0.0879569699840
    2.4940945349        -0.5516505812800
    0.8780758453        -0.5344329483300
0 0 1 2 1
    0.3450688700         1.0000000000000
0 0 1 0 1
    0.1500539900         1.0000000000000
0 2 5 6 1
    98.053010494        -0.0144805646010
    22.586932277        -0.0954957507870
    6.8391509842        -0.3078767265100
    2.2332843818        -0.4993629288600
    0.7160659939        -0.3150347621300
0 2 1 0 1
    0.2469232300         1.0000000000000
0 3 1 0 1
    0.4148382900         1.0000000000000
```

O

```
8 6
0 0 5 2 1
    2266.1767785        -0.005343180993
    340.87010191        -0.039890039230
    77.363135167        -0.178539119850
    21.479644940        -0.464276849590
    6.6589433124        -0.443097451720
0 0 1 2 1
    0.8478937000         1.000000000000
0 0 1 0 1
    0.2790534900         1.000000000000
0 2 3 4 1
    17.721504317         0.043394573193
    3.8635505440         0.230941207650
    1.0480920883         0.513753110640
0 2 1 0 1
    0.2623818000         1.000000000000
0 3 1 0 1
    0.4790426200         1.000000000000
```

d12

```
MgO bulk
CRYSTAL
0 0 0
225
4.21
2
12 0.     0.     0.
 8 0.5    0.5    0.5
END
...
```
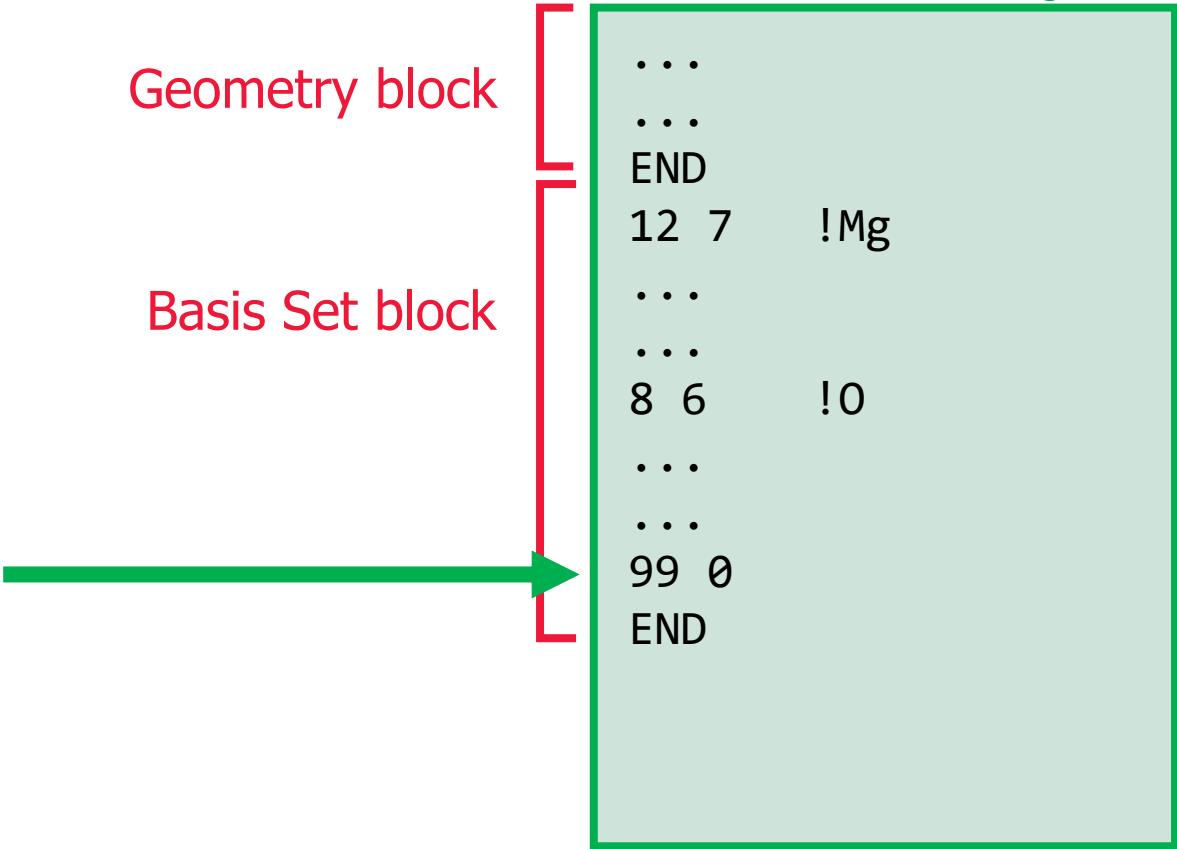
Add them here!

Closing the Basis Set Block

- As a last entry in the basis set you *need* to add a
  99 0

```
...
...
END
12 7     !Mg
...
...
8 6      !O
...
...
99 0
END
```

Geometry block

Basis Set block

- This is there for historical reasons!

The Third Block

Usually used to control the SCF parameters

- The only mandatory keyword here is **SHRINK** (for periodic calculations)

  ➡️ Select a **grid of points** in the reciprocal space for the sapling of the **FBZ**

  - Need to specify a factor for the main **Monkhorst-Pack Grid**
  - and a second one for the **Gilat net**
    (only used for metallic systems, but always necessary)

  ➡️ For example, **8** will generate an **8x8x8** grid

  > The **SHRINK** is related to the **size** of the systems and
  > highly impact the *COST* and the *ACCURACY* of the calculation

- Let's **run the full calculation**
  and look at the **output** of the SCF!

d12

```
MgO bulk
CRYSTAL
0 0 0
225
4.21
2
12 0.    0.
0.
  8 0.5   0.5
0.5
END
12 7
...
99   0
END
SHRINK
8 8
END
```

Third Block

The Third Block

- The third block is also used to select the **computational method**:
  - By default, the code will run a **Restricted Hartree-Fock** (RHF) calculation
  - You can use the keyword UHF for a unrestricted (open shell),
    or ROHF for a restricted-open HF calculation

- If you want to perform a **DFT** calculation you need to add the keyword DFT
  - This keyword will open a **sub-block** (that needs to be closed with an END)
  - Here you must specify a **DFT functional**, either by combining specific
    flavours of *exchange and correlation*, or by using a *premade* functionals (many
    available)
  - And you can add further keywords to control DFT-related parameters
    (e.g. the integration grid size, open/close shell, etc.)

DFT Calculation

- Now **add the DFT block** in the third section of the MgO input

LDA	RSCAN
SVWN	R2SCAN
VWN	M05-2X
VBH	M06
PWLSD	M06-HF
PBE	M06-2XMN15
PBE0	WB97X
RPBE	CAM-B3LYP
B3LYP	HSE06

- **Run** again the calculation

d12

```
MgO bulk
CRYSTAL
0 0 0
225
4.21
2
12 0.     0.
0.
 8 0.5    0.5
0.5
ENDgeom
12 7
...
99  0
ENDbs
DFT
PBE
ENDdft
SHRINK
8 8
END
```

Improving Convergence

- Many keyword can optionally be used to improve your SCF **convergence** and **accuracy**

  - **TOLINTEG**

    6 6 6 6 12

    $\underbrace{\text{6 6}}_{\text{Coulomb}}\ \underbrace{\text{6 6 12}}_{\text{Exchange}}$

    Controls the truncation of the infinite **Coulomb and Exchange** series, the higher the better but can easily increase *computational time*

  - **TOLDEE**

    6

    10^{-6} threshold value for energy (Ha): SCF stops when the absolute energy difference is less than this value (or the number of cycles exceeds 50)

  - **FMIXING**

    30

    Percentage of Hamiltonian matrix mixing. At cycle i: $\mathbf{F}_i = (1-p)\mathbf{F}_i + p\mathbf{F}_{i-1}$ Where p is the FMIXING value. Can help prevent oscillations during SCF

  - **DIIS**
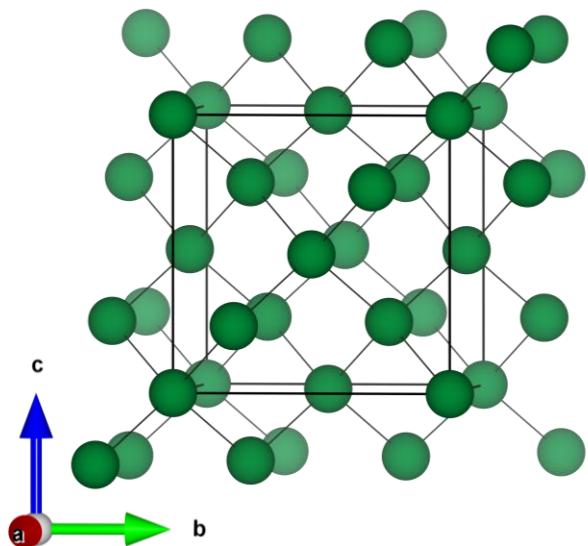
    Direct Inversion of Invariant Subspace convergence accelerator. Active by default, but if you want to turn it off (e.g. for SPINLOCK) use **NODIIS**

Exercise: Silicon

- Go to the folder QMMC2026/exercise_silicon

```
$ cd QMMC2026/exercise_silicon
```

- Here you will find a *blank* input for Si

- Your task is to **fill** the **basis set** and **computational method/parameters** and collect **output** information:
  1. Band gap
  2. Final Energy
  3. number of SCF cycles
  4. Total execution time

d12

```
CRYSTAL
0 0 0
227
5.42
1
14  .125  .125  .125
END
[…basis set…]
99 0
END
[…comp. params…]
SHRINK
8 8
END
```

Effect of the Functional

- Let's start by collecting outputs for different DFT functionals
- For now, we keep the **Basis Set fixed: Si_88-1111G(d)_Towler_1998**

Functional	Energy [Ha]	Bg [eV]	CYC	Time [s]
HF				
PBE				
SCAN				
PBE0				
HSE06				
WB97				
B3LYP				
PBE+15%EXX*				

- Try to be **clean** and **organized**, as we will run lots of calculations.

Experimental value for band gap is **1.12 eV**

Effect of the Functional

advanced School on **QUANTUM MODELLING** of Materials with CRYSTAL

- Let's start by collecting outputs for different DFT functionals
- For now, we keep the **Basis Set fixed: Si_88-1111G(d)_Towler_1998**

Functional	Energy [Ha]	Bg [eV]	CYC	Time [s]
HF	-5.7792311344514E+02	6.2331	7	8.56
PBE	-5.7878743576359E+02	0.6569	6	3.15
SCAN	-5.7912188734300E+02	0.9086	5	3.83
PBE0	-5.7883971170346E+02	1.7889	5	7.25
HSE06	-5.7883815484819E+02	1.2186	5	17.02
WB97	-5.7905494134239E+02	5.5178	7	22.31
B3LYP	-5.7898332474210E+02	1.7941	6	8.42
PBE+15%EXX*				

- Try to be **clean** and **organized**, as we will run lots of calculations.

Experimental value for band gap is **1.12 eV**

Custom Hybrid Functionals

- Need to use the keyword **HYBRID** and then specify the **percentage** of EXX you want to use inside the **DFT** block

d12

```
CRYSTAL
0 0 0
227
5.42
1
14  .125  .125   .125
END
[…basis set…]
99 0
END
DFT
PBE
HYBRID
15
END
SHRINK
8 8
END
```

$$E_{xc} = E_c + (1 - \alpha)E_x + \alpha E_{EXX}$$

Global hybrid functionals include a specific fraction (α) of Exact Exchange (EXX) that can be tuned

Effect of the Functional

- Let's start by collecting outputs for different DFT functionals
- For now, we keep the **Basis Set fixed: Si_88-1111G(d)_Towler_1998**

Functional	Energy [Ha]	Bg [eV]	CYC	Time [s]
HF	-5.7792311344514E+02	6.2331	7	8.56
PBE	-5.7878743576359E+02	0.6569	6	3.15
SCAN	-5.7912188734300E+02	0.9086	5	3.83
PBE0	-5.7883971170346E+02	1.7889	5	7.25
HSE06	-5.7883815484819E+02	1.2186	5	17.02
WB97	-5.7905494134239E+02	5.5178	7	22.31
B3LYP	-5.7898332474210E+02	1.7941	6	8.42
PBE+15%EXX*	-5.7881823442817E+02	1.3184	5	7.02

- Try to be **clean** and **organized**, as we will run lots of calculations.

Experimental value for band gap is **1.12 eV**

Self-Consistent Hybrids

- Usually, the **percentage** of Exact Exchange in hybrid functional is bounded to the **dielectric constat** of the material as in $\alpha = \dfrac{1}{\epsilon_\infty}$

- We can either use the ϵ_∞ from literature or **compute** it with the **CPHF/CPKS**

- However, the ϵ_∞ depends on the value of EXX, so we need to solve this problem self-consistently.

- In CRYSTAL we have a *totally automated* driver that can be easily used with one keyword in the geometry block: **SCHYBRID**

- **Try** to add it to a PBE calculation

d12
```
[...]
SCHYBRID
END
ENDgeom
[...]
PBE0
[...]
```

Self-Consistent Hybrids

- In the output we get something like this:

```
-------------------------------------------------------------
    AVERAGE DIELECTRIC CONSTANT: 11.8953
    OPTIMAL EXCHANGE FRACTION:  8.4023
-------------------------------------------------------------
```

Functional	Energy [Ha]	Bg [eV]	CYC	Time [s]
HF	-5.7792311344514E+02	6.2331	7	8.56
PBE	-5.7878743576359E+02	0.6569	6	3.15
SCAN	-5.7912188734300E+02	0.9086	5	3.83
PBE0	-5.7883971170346E+02	1.7889	5	7.25
HSE06	-5.7883815484819E+02	1.2186	5	17.02
WB97	-5.7905494134239E+02	5.5178	7	22.31
B3LYP	-5.7898332474210E+02	1.7941	6	8.42
PBE+15%EXX	-5.7881823442817E+02	1.3184	5	7.02
PBE+8.4%EXX	-5.7880446821997E+02	1.0202	5	12.76

Effect of the Basis Set

- Now we keep the same method (PBE+8.4%EXX)
  and we **change the basis set**

- Let's use some other BS from the **Crystal Basis Set Library**

Basis Set	Energy [Ha]	Bg [eV]	CYC	Time [s]
Towler	-5.7880446821997E+02	1.0202	5	12.76
Heyd				
POB-TZVP-rev2				

Effect of the Basis Set

- Now we keep the same method (PBE+8.4%EXX) and we **change the basis set**

- Let's use some other BS from the CRYSTAL **Basis Set Library**

Basis Set	Energy [Ha]	Bg [eV]	CYC	Time [s]
Towler	-5.7880446821997E+02	1.0202	5	12.76
Heyd	-5.7879804678833E+02	1.0149	6	59.32
POB-TZVP-rev2	-5.7876747268560E+02	1.1541	5	22.63

Effect of the Tolerances

- A strong impact on the final result can be obtained changing the **tolerances** for the truncation of the **Coulomb** and **Exact Exchange series**.

- Let's improve them with the `TOLINTEG` keyword in the third block of the input.
  ➡ After, insert the **5 integer tolerances** as below
     Keep method (PBE+8.4%EXX) and basis set (pob-TZVP-rev2) constant.

d12

```
...
TOLINTEG
8 8 8 12 24
...
```

TOLINTEG	Energy [Ha]	Bg [eV]	CYC	Time [s]
6 6 6 6 12	-5.7876747268560E+02	1.1541	5	22.63
7 7 7 10 20				
8 8 8 12 24				
9 9 9 15 30				

DEFAULT

Effect of the Tolerances

- A *strong impact* on the final result can be obtained changing the **tolerances** for the truncation of the **Coulomb** and **Exact Exchange series**.

- Let's improve them with the `TOLINTEG` keyword in the third block of the input.
  - → After, insert the **5 integer tolerances** as below
    Keep method (PBE+8.4%EXX) and basis set (pob-TZVP-rev2) constant.

d12

```
...
TOLINTEG
8 8 8 12 24
...
```

TOLINTEG	Energy [Ha]	Bg [eV]	CYC	Time [s]
6 6 6 6 12	-5.7876747268560E+02	1.1541	5	22.63
7 7 7 10 20	-5.7876713536691E+02	1.1666	5	49.81
8 8 8 12 24	-5.7876698770486E+02	1.1699	5	77.71
9 9 9 15 30	-5.7876704233621E+02	1.1728	5	134.80

DEFAULT