Practical Session:

SCALE-UP and its auxiliary tools

Pablo García-Fernández

Santander July 2017

Universidad de Cantabria

Goals for the practical session

Overall objective: Run SCALE-UP and utils in its different modes

1. Running SCALE-UP in the terminal
   - Single point calculation: basic keywords
   - Output and files
   - Plotting the bands and the DOS
   - Visualizing the geometry
   - Visualizing other properties
2. Running SCALE-UP python interface
   - Plotting energy surfaces

First run

Go to the first exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/Materials/1-Hello_world
```

First run

Go to the first exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/Materials/1-Hello_world
```

Check the input:

```
$ vi input.fdf
```

or (if you prefer a gui-ed editor),

```
$ gedit input.fdf &
```

First run

Go to the first exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/Materials/1-Hello_world
```

Check the input:

```
$ vi input.fdf
```

or (if you prefer a gui-ed editor),

```
$ gedit input.fdf &
```

Check that the SCALE-UP bin directory is in the path:

```
$ echo $PATH
```

First run

Go to the first exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/Materials/1-Hello_world
```

Check the input:

```
$ vi input.fdf
```

or (if you prefer a gui-ed editor),

```
$ gedit input.fdf &
```

Check that the SCALE-UP bin directory is in the path:

```
$ echo $PATH
```

Run the code:

```
$ scaleup.x < input.fdf > output
```

First run

Go to the first exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/Materials/1-Hello_world
```

Check the input:

```
$ vi input.fdf
```

or (if you prefer a gui-ed editor),

```
$ gedit input.fdf &
```

Check that the SCALE-UP bin directory is in the path:

```
$ echo $PATH
```

Run the code:

```
$ scaleup.x < input.fdf > output
```

Check the output:

```
$ ls
```

```
$ vi output
```

The human-readable output file

The output file has some fixed sections and some optional ones:

Fixed sections

1. Header & input summary
2. Indications of file printout
3. Summary of keywords read
4. Summary of time & memory
5. Closing message

The human-readable output file

The output file has some fixed sections and some optional ones:

Fixed sections

1. Header & input summary
2. Indications of file printout
3. Summary of keywords read
4. Summary of time & memory
5. Closing message

Optional sections

1. Energy decomposition
2. SCF For BO calculations
3. Montecarlo step For MC runs
4. Dynamic step For MD simulations
5. ...

First run

Go to the first exercise of the session:

```
$ cd ~/workshop/Day1/02_First_run/0-Hello_world
```

Check the input:

```
$ vi input.fdf
```

or (if you prefer a gui-ed editor),

```
$ gedit input.fdf &
```

To run the code:

```
$ $SCALEUPHOME/scaleup.x < input.fdf > output
```

Check the output:

```
$ ls
```

```
$ vi output
```

Several of the keywords are typical of any *ab initio* code:

First run

Go to the first exercise of the session:

```
$ cd ~/workshop/Day1/02_First_run/0-Hello_world
```

Check the input:

```
$ vi input.fdf
```

or (if you prefer a gui-ed editor),

```
$ gedit input.fdf &
```

To run the code:

```
$ $SCALEUPHOME/scaleup.x < input.fdf > output
```

Check the output:

```
$ ls
```

```
$ vi output
```

Several of the keywords are typical of any *ab initio* code:

Knowing it is a linear chain in z, converge energy with k-sampling

```
$ vi input.fdf
```

Check change of energy components with k-sampling

The output files

The output is controlled by the Print_... number keywords

higher number gives more output, -1 supresses it

The output files

The output is controlled by the Print_... number keywords

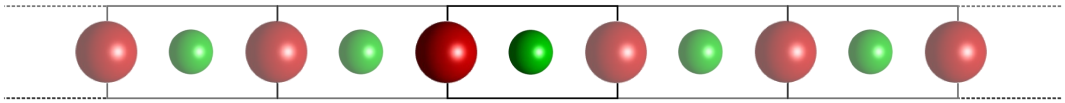
higher number gives more output, -1 supresses it

1. Geometry systemname.REF/systemname.restart
   Print_Geometry 1
2. Electrostatic info systemname.eltic
   Print_electrostatic 1
3. Electronic eigenvalues systemname.ener
   Print_energies 1
4. Electronic bands systemname.bands
   Print_bands 1
5. Orbital occupation systemname.orbocc
   Print_orbocc 1
6. Real-space Hamiltonian systemname.rham
   Print_Space_Hamiltonian 1

They can be read by the SCALE-UP python utils!

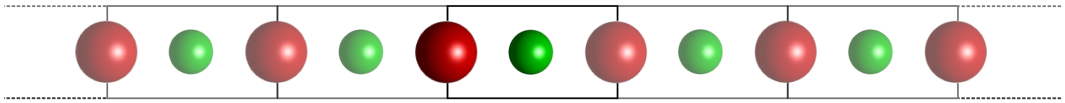
Check the manual!

Exercise 2-Output_files



Goal: Learn to request various output files studying ionic chain

Exercise 2-Output_files



Goal: Learn to request various output files studying ionic chain

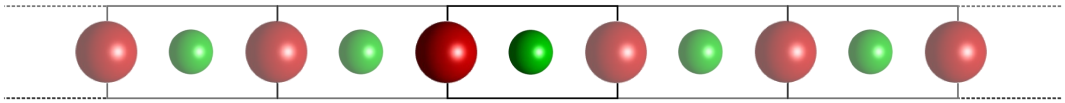
Go to the second exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/2-Output_files
```

Check the input to observe the various output possibilities:

```
$ vi input.fdf
$ scaleup.x < input.fdf > output
$ vi _biatomic_chain.orbocc
$ ...
```

Exercise 2-Output_files



Goal: Learn to request various output files studying ionic chain

Go to the second exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/2-Output_files
```

Check the input to observe the various output possibilities:

```
$ vi input.fdf
$ scaleup.x < input.fdf > output
$ vi _biatomic_chain.orbocc
$ ...
```

- Find:
  -Difference occupation of the pz orbital on the anion
  -Difference occupation of the s orbital on the cation
- Find the total charge and the difference charge on the atoms
  Relationship between the orbital occupation and charges?
- Find the electrostatic potential and dipoles on the atoms
- How is the charge of the cation during convergence?

The utils python script

The scaleup_utils.py script can read many output files:

```
$ python $SCALEUPHOME/scripts/scaleup_utils.py -mymode -file myfile.ext
```

The utils python script

The scaleup_utils.py script can read many output files:

```
$ python $SCALEUPHOME/scripts/scaleup_utils.py -mymode -file myfile.ext
```

Density of states Plots the DOS

```
$ python $PATH/scaleup_utils.py -dos -file file.ener
```

Bands Plots the bands

```
$ python $PATH/scaleup_utils.py -bands -file file.bands
```

The utils python script

The scaleup_utils.py script can read many output files:

```
$ python $SCALEUPHOME/scripts/scaleup_utils.py -mymode -file myfile.ext
```

Density of states Plots the DOS

```
$ python $PATH/scaleup_utils.py -dos -file file.ener
```

Bands Plots the bands

```
$ python $PATH/scaleup_utils.py -bands -file file.bands
```

Geometry Exports a xsf (xcrysden) file

```
$ python $PATH/scaleup_utils.py -geom -sysname filename (options)
```

This requires filename.REF (RAG) and filename.restart $(\{\vec{u}_{\boldsymbol{\lambda}}\},\eta)$

Absolute (default) or distortion (options=-distort) possible

The utils python script

The scaleup_utils.py script can read many output files:

```
$ python $SCALEUPHOME/scripts/scaleup_utils.py -mymode -file myfile.ext
```

Density of states Plots the DOS

```
$ python $PATH/scaleup_utils.py -dos -file file.ener
```

Bands Plots the bands

```
$ python $PATH/scaleup_utils.py -bands -file file.bands
```

Geometry Exports a xsf (xcrysden) file

```
$ python $PATH/scaleup_utils.py -geom -sysname filename (options)
```

This requires filename.REF (RAG) and filename.restart $(\{\vec{u}_{\boldsymbol{\lambda}}\},\eta)$

Absolute (default) or distortion (options=-distort) possible

Properties Exports a xsf (xcrysden) file with scalar or vector fields

```
$ python $PATH/scaleup_utils.py -geom -sysname filename (prop) (option)
```

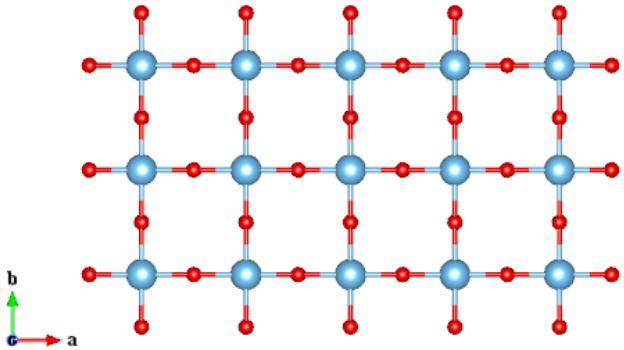
Property can be -mag or -diffcharge

Check the help:

```
$ python $PATH/scaleup_utils.py -help
```

Exercise 3-Visual_dos_bands

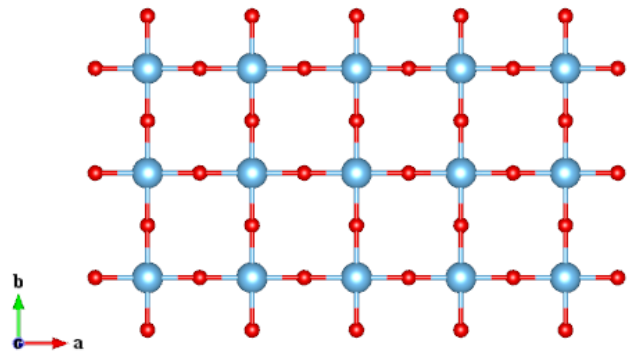
Goal: Learn to use the utils script to make simple plots



System is a square-planar lattice

Exercise 3-Visual_dos_bands

Goal: Learn to use the utils script to make simple plots



System is a square-planar lattice

Go to the third exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/3-Visual_dos_bands
```

Check the input:

```
$ vi input.fdf
```

and edit it to complete the following exercises:

- Obtain a DOS diagram
- Converge the DOS diagram with k-sampling
- Plot a band diagram
- Plot the band diagram
  $\Gamma(0,0,0) \rightarrow X(0.5,0,0) \rightarrow M(0.5,0.5,0) \rightarrow \Gamma(0,0,0)$

Geometry and property visualization

http://www.xcrysden.org/

We can use xcrysden to visualize geometry and properties:

```
$ xcrysden
```

Geometry and property visualization

http://www.xcrysden.org/

We can use xcrysden to visualize geometry and properties:

```
$ xcrysden
```

Open the geometry file:
File→Open structure...→filename.xsf

Geometry and property visualization

http://www.xcrysden.org/

We can use xcrysden to visualize geometry and properties:

```
$ xcrysden
```

Open the geometry file:
File→Open structure...→filename.xsf

To plot vector fields:
Display→Forces or Press f
Change arrow scale by Modify→Force Settings or Press shift-f

Geometry and property visualization

http://www.xcrysden.org/

We can use xcrysden to visualize geometry and properties:

```
$ xcrysden
```

Open the geometry file:
File→Open structure...→filename.xsf

To plot vector fields:
Display→Forces or Press f
Change arrow scale by Modify→Force Settings or Press shift-f

To plot scalar fields:
Tools→Data grid and enter info on isosurface

Geometry and property visualization

http://www.xcrysden.org/

We can use xcrysden to visualize geometry and properties:

```
$ xcrysden
```

Open the geometry file:
File→Open structure...→filename.xsf

To plot vector fields:
Display→Forces or Press f
Change arrow scale by Modify→Force Settings or Press shift-f

To plot scalar fields:
Tools→Data grid and enter info on isosurface

Useful:
Modify→Number of units drawn...

Exercise 4-Visual_properties

Goal: Learn to visualize various properties with xcrysden

System is a square-planar lattice

It has been doped with one electron... Is it localized? If so, where?

Exercise 4-Visual_properties

Goal: Learn to visualize various properties with xcrysden

System is a square-planar lattice

It has been doped with one electron... Is it localized? If so, where?

Go to the fourth exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/4-Visual_properties
```

Check the input and edit it to complete the following exercises:

- Represent the full geometry

  ```
  $ python -geom -sysname _2d_lattice_SP
  ```

  ```
  $ xcrysden &
  ```

- Represent the distortions of the system Load _2d_lattice_SP.xsf and show the distortions (Press f)
- Represent the difference charge as a scalar field
- Combine the plots of difference charge and distortions
- Represent the magnetization to find whether the system is spin-polarized

Python interface

In the python interfase SCALE-UP is essentially a class that, once created, allows you to interact with the model

- Init the model saving the SCALE-UP calculation in a variable:

```
mycalc = sclup.scaleup('srtio3_full_lat.xml',supercell=[2,2,1],\
                       lattice=True,electrons=False)
```

Python interface

In the python interfase SCALE-UP is essentially a class that, once created, allows you to interact with the model

- Init the model saving the SCALE-UP calculation in a variable:
  mycalc = sclup.scaleup('srtio3_full_lat.xml',supercell=[2,2,1],\
  lattice=True,electrons=False)
- Once you init the model you can modify the distortions:
  mycalc.displacements[i]=α
  displacements is a vector:

i=0	*i=1*	*i=2*	*i=3*	*i=4*	*i=5*	*i=6*	*i=7*	*i=8*	*i=9*	*i=10*	*i=11*
$at_1(x)$	$at_1(y)$	$at_1(z)$	$at_2(x)$	$at_2(y)$	$at_2(z)$	$at_3(x)$	$at_3(y)$	$at_3(z)$	$at_4(x)$	$at_4(y)$	$at_4(z)$ ···

Python interface

In the python interfase SCALE-UP is essentially a class that, once created, allows you to interact with the model

- Init the model saving the SCALE-UP calculation in a variable:
  mycalc = sclup.scaleup('srtio3_full_lat.xml',supercell=[2,2,1],\
  lattice=True,electrons=False)
- Once you init the model you can modify the distortions:
  mycalc.displacements[i]=α
  displacements is a vector:

i=0	*i=1*	*i=2*	*i=3*	*i=4*	*i=5*	*i=6*	*i=7*	*i=8*	*i=9*	*i=10*	*i=11*
$at_1(x)$	$at_1(y)$	$at_1(z)$	$at_2(x)$	$at_2(y)$	$at_2(z)$	$at_3(x)$	$at_3(y)$	$at_3(z)$	$at_4(x)$	$at_4(y)$	$at_4(z)$ ...

- Change the strain
  mycalc.strain[i]=η
  Index i (0-5) runs Voight indexes ($\eta_{xx}, \eta_{yy}, \eta_{zz}, \eta_{zy}, \eta_{zx}, \eta_{xy}$)

Python interface

In the python interfase SCALE-UP is essentially a class that, once created, allows you to interact with the model

- Init the model saving the SCALE-UP calculation in a variable:
  mycalc = sclup.scaleup('srtio3_full_lat.xml',supercell=[2,2,1],\
  lattice=True,electrons=False)
- Once you init the model you can modify the distortions:
  mycalc.displacements[i]=α
  displacements is a vector:

i=0	*i=1*	*i=2*	*i=3*	*i=4*	*i=5*	*i=6*	*i=7*	*i=8*	*i=9*	*i=10*	*i=11*
$at_1(x)$	$at_1(y)$	$at_1(z)$	$at_2(x)$	$at_2(y)$	$at_2(z)$	$at_3(x)$	$at_3(y)$	$at_3(z)$	$at_4(x)$	$at_4(y)$	$at_4(z)$ ···

- Change the strain
  mycalc.strain[i]=η
  Index i (0-5) runs Voight indexes ($\eta_{xx}, \eta_{yy}, \eta_{zz}, \eta_{zy}, \eta_{zx}, \eta_{xy}$)
- Get the energy:
  energy=mycalc.get_energy()

Python interface

In the python interfase SCALE-UP is essentially a class that, once created, allows you to interact with the model

- Init the model saving the SCALE-UP calculation in a variable:
  mycalc = sclup.scaleup('srtio3_full_lat.xml',supercell=[2,2,1],\
  lattice=True,electrons=False)
- Once you init the model you can modify the distortions:
  mycalc.displacements[i]=α
  displacements is a vector:

i=0	*i=1*	*i=2*	*i=3*	*i=4*	*i=5*	*i=6*	*i=7*	*i=8*	*i=9*	*i=10*	*i=11*
$at_1(x)$	$at_1(y)$	$at_1(z)$	$at_2(x)$	$at_2(y)$	$at_2(z)$	$at_3(x)$	$at_3(y)$	$at_3(z)$	$at_4(x)$	$at_4(y)$	$at_4(z)$ ···

- Change the strain
  mycalc.strain[i]=η
  Index i (0-5) runs Voight indexes ($\eta_{xx}, \eta_{yy}, \eta_{zz}, \eta_{zy}, \eta_{zx}, \eta_{xy}$)
- Get the energy:
  energy=mycalc.get_energy()

Suggestion: How to calculate a phonon band diagram?

Exercise 5-Python_surfaces

Goal: Use the python interface to visualize the energy surface along a path

How does strain alter $SrTiO_3$ phase transition?

Exercise 5-Python_surfaces

Goal: Use the python interface to visualize the energy surface along a path

How does strain alter $SrTiO_3$ phase transition?

Go to the fifth exercise of the session:

```
$ cd ~/Workshop/Day1/02_First_run/5-Python_surfaces
```

Check the script that plots the energy for octahedral rotation (M point):

```
$ vi get_energy_rotation.py
```

- Run the script and look at the resulting energy surface

  Is there instability?
- Change the strain of the system isotropically to enlarge the cell

  Remember strain comes in Voight notation ($\eta_{xx}, \eta_{yy}, \eta_{zz}, \eta_{zy}, \eta_{zx}, \eta_{xy}$)

  Is the mode associated to the distortion softer or harder?
- Add an extra loop to explore the behavior with strain and plot your results

  Is the instability destroyed at some point?

Practical session summary

- 1-Hello_world
  Converge the energy with k-sampling! Check energy terms
- 2-Output_files
  Check the input for various output files
- 3-Visual_dos_and_bands
  Play with SCF options and plot the DOS and bands for system
- 4-Visual_properties
  Use utils to explore the plotting modes for the geometry
  Represent magnetization, electron and holes to study the electronic structure
- 5-Python_surfaces
  How does strain affect $SrTiO_3$ rotations?

```
$ python $SCALEUPHOME/scripts/scaleup_utils.py -mymode -file myfile.ext
```