README.md

Examples: Structure Input by Space Group and Motif

The examples in this collection illustrate how to run pscfFieldGen by supplying a representative motif of particles and allowing space group symmetry to generate the full set of positions. This represents the model file setting `coord_input_style motif`. All examples in this collection use PSCF (Fortran) for field format conversions and SCFT calculations after running pscfFieldGen.

Each example in the collection is found in a directory named for the phase geometry being illustrated. In all, there are 4 geometries representing 3 different crystal systems and 4 space groups. All examples use 3D morphologies, but the approach would be comparable in 2 Dimensions.

Available Geometries

Morphology	Directory	Crystal Sys	Space Group
Body Centered Cubic Spheres	`bcc/`	cubic	I m -3 m
Frank-Kasper A15 Spheres	`a15/`	cubic	P m -3 n
Frank-Kasper Sigma Spheres	`sigma/`	tetragonal	P 42/m n m
Frank-Kasper C14 Spheres	`c14/`	hexagonal	P 63/m m c

Running the Examples

To run an example, first make sure that you have PSCF installed (the Fortran version), and have added pscfFieldGen to your python path. When that is done, navigate to the example's directory. Each example has two automated execution options. The first of these focuses on the pscfFieldGen operation, generating only the KGRID initial guess file and converting it to an RGRID format for visualization. The second option does this and continues on to solve the SCFT equations and converge the field.

Quick Run

The first automated run option is a quick generation of the initial guess and conversion to a visualizer-friendly format. This execution option is automated by the executable script *quickrun* available in the directory, and can be launched by typing

```
./quickrun
```

on the command line from within the example's directory. The script acts as a shorthand for the following commands, which could be used directly

```
python -m pscfFieldGen -f model -t > generationLog
pscf < param_kgrid > conversionLog
```

The first line of the script uses the traced invocation of pscfFieldGen to generate a KGRID field file ('rho_kgrid.in'). The traced invocation (flag `-t`) causes the program to echo the input read from *model*. After echoing this data, the program subsequently prints the details of the Lattice it will use, the crystal structure it will use, and output notice of key steps in its calculation. The addition of `> generationLog` at the end of the command redirects this output to a file 'generationLog'.

The second line executes PSCF using the parameter file 'param_kgrid'. This operation convers the KGRID field file data in 'rho_kgrid.in' to RGRID field file format as the file 'rho_rgrid.in'. This file can be used with the visualizer to check the structure of the resulting concentration field. The output from this invocation of pscf is similarly redirected into a log file called `conversionLog` .

Full Run

The second automated run option expands on the first and solves the SCFT equations from the generated initial guess. This execution option is automated by the *fullrun* script present in the example directory. It can be invoked by entering

```
./fullrun
```

on the command line from within the example's directory. This script is equivallent to the following commands

```
python -m pscfFieldGen -f model -t > generationLog
pscf < param_kgrid > conversionLog
pscf < param_converge > solutionLog
```

The first two lines in this execution are identical to those in the quick run, and behave identically. The last command executes PSCF using the parameter file 'param_converge'. This first converts the concentration field in 'rho_rgrid.in' to a basis format file called 'rho.in' and subsequently converts that to a chemical potential field file 'omega.in'. This field file is then used as input to an ITERATE command, which generates the files 'out', 'rho', and 'omega'. After convergence, the concentration field in 'rho' is converted to RGRID format in 'rho_rgrid' for visualization. The output from this is redirected into 'solutionLog'.

If the *quickrun* execution has completed and the directory has not yet been cleaned (see the next section), then the *fullrun* execution pathway can be completed simply by executing the third command listed above.

Resetting the Example

Because of the large number of files produced by both the quick run ('rho_kgrid.in', 'rho_rgrid.in', 'generationLog', 'conversionLog') and the full run ('rho.in', 'omega.in', 'out', 'rho', 'omega', 'rho_rgrid', 'solutionLog'), a script is provided to automatically delete all of these files to reset the example. After running an example, the files generated by it can be removed by entering

```
./clean
```

on the command line. The script checks which of the expected files (as listed above) are present in the directory and deletes those that are. This leaves the example directory empty for another run.