SSCHA input file cheatsheet

&inputsscha

t : The temperature used to generate the ensemble (in Kelvin). [REQUIRED]

tg : The temperature that will be used in the minimization (in Kelvin).

lambda_a : Force constant minimization step. Step for the force constant matrix

lambda_w : Force constant minimization step. Step for the structure

minim_struc : Do SSCHA minimize the structure? [.true./.false.]

precond_wyck : Preconditioning variable [.true./.false.]

preconditioning : Preconditioning dynamical martix variable for a very fast minimization [.true./.false.] (preconditioning can be used only if root_representation = "normal")

root_representation : Set root representation, a trick to increase the speed of the minimization and to avoid the imaginary frequency error. (The python new code is not able to do simultaneously precondition and root_representation.)

1. "normal" : normal minimization, can lead to imaginary frequencies
2. "sqrt" : square root representation.
3. "root4" : fourth-root represnetation, the best one (and the slowest).

neglect_symmetries : **********

n_random_eff : **********

n_random : The dimension of the ensemble

meaningful_factor : The stopping criteria. The code will end the minimization after the gradient is lower than meaningful_factor times its stochastic error.

eq_energy : Set the equilibrium energy. It should be the energy of the structure without fluctuations, it is used to separate the electronic and the vibrational energy, since they are usually of different order of magnitude. It is measured in Ry.

&inputsscha

fildyn_prefix : [REQUIRED] **********

nqirr : [REQUIRED] The number of irriducible q points (just look how many dynamical-matrixes files are there).

data_dir : The position of the ensemble (where the data are stored). Unit of measurements must be in bohr for displacements and $Ry/bohr$ for forces and $Ry/bohr^3$ for stress tensors. Energy is in Ry.

load_bin : **********

supercell_size : " The supercell size.

max_ka : Maximum number of steps after which the code is automatically stopped

stress_offset : **********

gradi_op : Which gradient is used to trigger the stopping condition. By default, both of them should satisfy the meaningufl criteria. Options are "all", "gc", and "gw".

1. "all" - both the gradient should satisfy the meaningulf (default)
2. "gw" - only the wyckoff (i.e. structure) gradient.
3. "gc" - only the force-constant matrix gradient.

population : The population id. This is an integer that distinguish different ensembles and allows for use the same data_dir for several minimizations.

print_stress : Legacy flag, not used anymore, now it automatically print the stress if it finds the stress tensor inside the ensemble.

use_spglib : **********

&relax

type : [REQUIRED] **********

- "sscha"
- "relax"
- "vc-relax"

n_configs : [REQUIRED] **********

max_pop_id : **********

start_pop : **********

ensemble_datadir : **********

generate_ensemble : **********

target_pressure : **********

fix_volume : **********

bulk_modulus : **********

sobol_sampling : **********

sobol_scatter : **********

&calculator

k_points : [REQUIRED] **********

k_offset : **********

disable_check : **********

program : [REQUIRED] **********

binary : **********

pseudo_ : **********

1. "quantum-espresso"

   - "ecutrho", "ecutwfc", "smearing", "degauss", "occupations", "conv_thr", "tstress", "tprnfor", "verbosity", "disk_io", "input_dft", "use_all_frac"

&utils

save_freq_filename :

save_rho_filename :

mu_lock_start :

mu_lock_end :

mu_free_start :

mu_free_end :

project_dyn :

project_structure :

&cluster

template :

CHA_CLUSTERS_DIR :

hostname :

pwd :

account :

binary_path :

mpicmd :

reconnect_attempts :

port :

shell :

submit_cmd :

queue_directive :

v_nodes :

n_nodes :

use_nodes :

v_cpu :

n_cpu :

use_cpu :

v_time :

n_time :

n_pools :

use_time :

v_memory :

max_ram :

: "use_memory" "v_partition" "partition_name"
"use_partition" "init_script" "max_recalc"
"batch_size" "local_workdir" "v_account"
"use_account" "sshcmd" "scpcmd" "timeout"
"job_numbers" "n_together"
"workdir"

Abstract

The input file perform the minimization.

To run the SSCHA code with the input file use:

```
>>> sscha -i simple_input.in --save-data simple_input.out
```

The file can have any name (often *.in).

An example of an input file:

```
!
! * * * * * * * * * * * * * * *
! *                           *
! *    VC - RELAX EXAMPLE     *
! *                           *
! * * * * * * * * * * * * * * *
!
!
! This is the input to perform the sscha minimization, followed
! by the change of the unit cell given by the stress step.
!
! This is not the recommended way to do it (you can do everything automatically)
! But usefull if you want to control manually each submission
!

&relax
type = "vc-relax"
start_pop = 2
max_pop_id = 2
generate_ensemble = .false.
fix_volume = .false.
target_pressure = 0 ! [GPa]
bulk_modulus = 15 ! [GPa]
n_configs = 1000
&end

&inputscha
n_random = 1000
data_dir = "../ensemble_data_test"
population = 2
fildyn_prefix = "../ensemble_data_test/dyn"
nqirr = 1
supercell_size =  1 1 1
Tg = 0
T = 0
meaningful_factor = 1e-4
gradi_op = "all"
n_random_eff = 500
print_stress = .true.
eq_energy = -144.40680397
lambda_a = 1
lambda_w = 1
root_representation = "normal"
preconditioning = .true.
max_ka= 20
/
```