

Nsat_runner allows to prepare input files automatically with tasks for queuing system *qsub* using in CIS. Mentioned task defining in input files it is command execute compiling code *nsat* for giving logarithm of surface gravity and command saving output of numerical code to proper directory. Input parameters for *nsat* code are values of $\log(g)$. In case of sequential calculations, *nsat* make calculations for a given $\log(g)$ and subsequently for the next values. Using CIS resources it is possible to run simultaneously *nsat* code with different values of $\log(g)$. *nsat_runner* prepare instruction, which execute *nsat* code with given value of $\log(g)$. These instructions are wrote as shell scripts for queuing system *qsub*.

*** Description of nsat_runner ***

nsat_runner need three parameters: initial and last value of $\log(g)$ and step.

Additional option is short help for *nsat_runner.py*, calling as follows:

```
python nsat_runner
```

If we do not give them any parameter, then program shows following helping message

Option `-g` and `-s` is mandatory.

usage:

```
%prog [options] arg1 arg2 ...
```

Example:

```
%prog -g 14.0 15.0 -s 0.1  
      [-h] [-g LOG LOG] [-s STEP]
```

optional arguments:

```
-h, --help          show this help message and exit  
-g LOG LOG, --log LOG LOG  
-s STEP, --step STEP
```

So, we should use following command to run *nsat_runner*

```
python nsat_runner.py -g initial_value_log(g) last_value_log(g) -s step_of_log(g)
```

for example:

```
python nsat_runner.py -g 14.0 15.0 -s 0.1
```

As a result, in its home directory *nsat_runner* creates NSATtask directory with subdirectories for each $\log(g)$ with names of *nsat_log(g)*. To such subdirectories *nsat_runner* puts file *tasklog(g).sh*, which contains shell commands for queuing system (see list below). In addition to *nsat_log(g)* directories, there is *tasks_runner* file in the NSATtask directory, containing *qsub* command with input parameters. In this case input parameters are *tasklog(g).sh* files. The next step, which should be done is copying whole NSATtask directory to CIS cluster and execute *nsat_runner* script. Results are written in different directories *nsat_log(g)*, depending on value of $\log(g)$ parameter.