

Table of Contents

Table of Contents	1
Restarting a stopped calculation	2
Saving the checkpoint file	2
Specifying the location of the checkpoint file	3
Restarting the calculation from the checkpoint file	3
Restarting the original script	4
Restarting geometry optimizations	4

[Try it!](#)[QuantumATK](#)[Contact](#)[Docs](#) » [Tutorials](#) » [Tips and tricks](#) » Restarting a stopped calculation

Restarting a stopped calculation

Version: P-2019.03

Downloads & Links

[PDF](#)

In this tutorial, you will learn how to restart a calculation that was terminated before converging (for example due to power outage, exceeding walltime in a queue, or exceeding maximum number of iterations) without having to start all over.



Saving the checkpoint file

It may happen that a calculation terminates before converging, for instance due to a power outage, or if a job runs over its allocated walltime in a queue. It may also be that convergence is not reached within the set number of maximum iterations. In these cases you will want to restart the job from the point it stopped.

For this purpose, QuantumATK saves the current state of the calculation to a **checkpoint file** at regular intervals. The default is to save it every 30 minutes. The name of the checkpoint file is always written in the log file.

```
+-----+
| Checkpoint Handler                                     |
| Filename : /tmp/checkpoint28146777.nc                 |
| Interval : 0.5 h                                     |
+-----+

Calculating Eigenvalues : =====
Calculating Density Matrix : =====

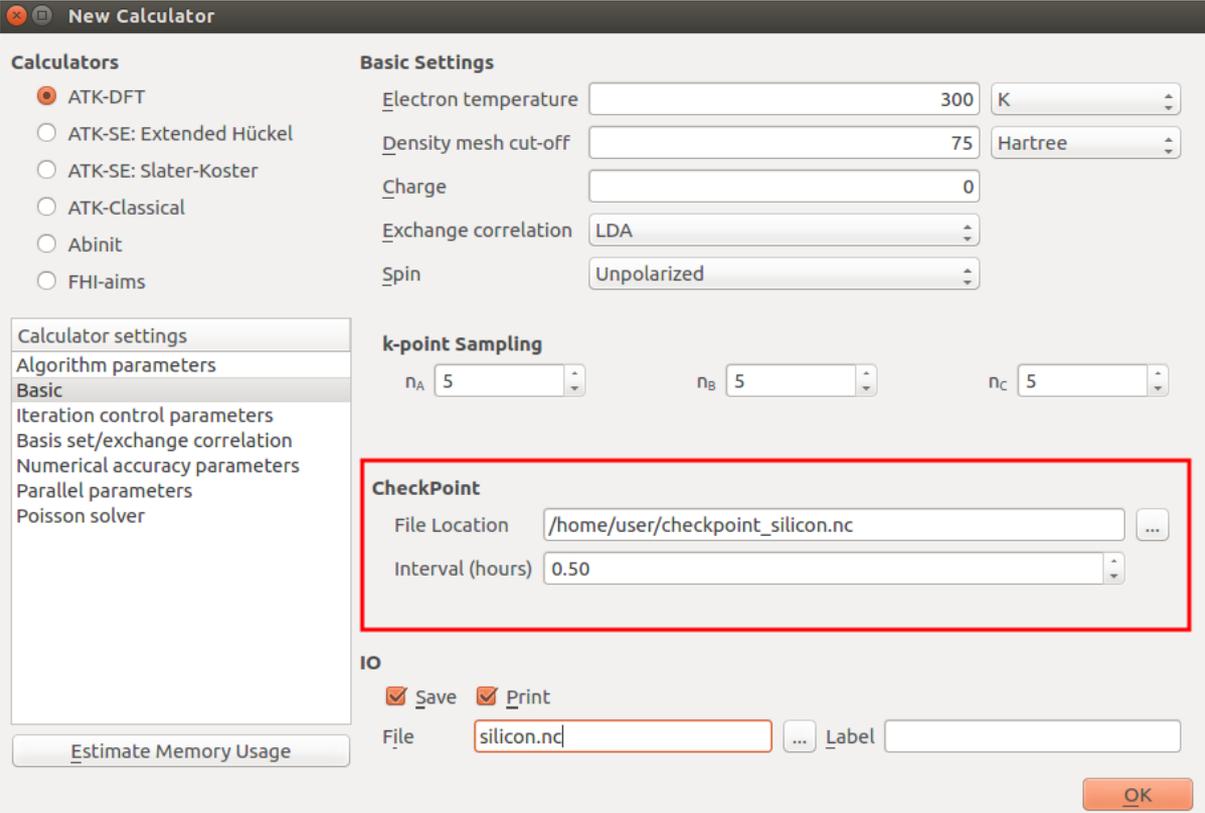
+-----+
| Density Matrix Report                                |
|              DM[U]   DM[D]   DD                   |
+-----+
| 0 Fe [ 0.717 , 0.717 , 0.716 ] 5.39789 2.59547 -0.00664 |
+-----+
```

Note

The default location of the checkpoint file is in the directory specified by the environment variable TEMP. If you are running on a large cluster, you may not have permission to write to the TEMP directory, and even if you do, any files you create in this directory may be deleted automatically when your job finishes – even if the QuantumATK calculation did not converge. In this case it is important to specify the location of the checkpoint file manually, e.g. in your HOME directory.

Specifying the location of the checkpoint file

The image below shows how you can set the saving interval and the name of the checkpoint file using the  **New Calculator** block in the QuantumATK  **Script Generator**.



Calculators

- ATK-DFT
- ATK-SE: Extended Hückel
- ATK-SE: Slater-Koster
- ATK-Classical
- Abinit
- FHI-aims

Basic Settings

Electron temperature: 300 K

Density mesh cut-off: 75 Hartree

Charge: 0

Exchange correlation: LDA

Spin: Unpolarized

k-point Sampling

n_A: 5 n_B: 5 n_C: 5

CheckPoint

File Location: /home/user/checkpoint_silicon.nc

Interval (hours): 0.50

IO

Save Print

File: silicon.nc Label:

Estimate Memory Usage

OK

If you wish, you can refer to the [`Reference Manual`](#) to get more information on how to set these parameters or turn off the checkpoint handler.

Restarting the calculation from the checkpoint file

The quickest way to restart a calculation from a checkpoint file is to read the configuration from the checkpoint file and then call update on the configuration.

```
configuration = nload("checkpointfile.hdf5")[0]
configuration.update()
nlsave("file.hdf5", configuration)
```

The argument to `nload()` should of course be set to the actual checkpoint file name.

The disadvantage of this approach is that if the *original* script contained any analysis blocks (e.g. to compute the band structure), you need to manually insert those blocks to the bottom of the restart script.

Restarting the original script

Conceptually, a better approach would be to rerun the script you already have, but tell it to start not from scratch, but from the checkpoint file. This would also retain all analysis blocks, as defined in the original script. This is possible; you just need to insert the lines of code shown above in the appropriate way.

Let us assume that you have a “standard” script, produced by the Script Generator, without too many elaborate steps. That is, a straightforward sequence of “Configuration” and “New Calculator”, followed by analysis blocks. In other cases, you can always modify the script in the same way as described here, but you have to take more care to preserve the logic. Special care needs to be taken if the script contains an  **InitialState** block.

Open your original script in the  **Editor** and locate the line

```
device_configuration.update()
```

Tip

For bulks or molecular calculations, the variable will be called `bulk_configuration` or `molecule_configuration` instead.

Then add the following line before that line:

```
device_configuration = nload("checkpointfile.hdf5")[0]
```

Again, the argument to `nload()` should of course be the actual checkpoint file name.

Now you can rerun the script.

Note

- The checkpoint file is not written exactly at the specified interval, but only when a step in the self-consistent loop has been completed and the requested time interval has passed.
- The history of the self-consistent loop is not written to the checkpoint file. Therefore, convergence might become more difficult when restarting, since the mixing algorithm has less information to work with than normally.

Restarting geometry optimizations

Restarting a geometry optimization is much more complicated. For a lengthy relaxation it is therefore always a good idea to use a trajectory file; if the calculation is interrupted you can take out some of the later images and set up a new optimization using this geometry as a starting point. Note, however, that some images in a QuasiNewton geometry optimization are “test balloons”, which may correspond to very large forces (i.e. a very bad guess), especially during the first 5–10 steps. So, it can be important to choose an image that does not have too large forces.

 Previous

Next 