Step 2 — Create Namespaces

- To make it easier to quickly setup parameters for operations we have created cfg-files for each operation that contains default values. The default values can be loaded into a named scope by writing
  - import scope_name = “path to cfg file”

- The scope_name can in principle be any name you decide. There are no rules for what you can call a scope. You may think of a scope like a read-only kind of record/struct.

- GRIT comes with default cfg scope-files for all operation types. One can explore the parameters inside these scope files for learning how to tweak and tune each operation type.

Step 3 — Assign Scopes to Operations

- Scopes do not do anything, one must assign their values to a given operation for a given phase (ie. label). The syntax is as follows

  - assign = operation_name label scope_name

- Notice that the scope is not merely assigned to an operation, but it is assigned to the pairing of an operation and a label. This is because operations can behave differently depending on what phase in the mesh they are invoked on.

- As an example one often turn of mesh optimisation in the ambient phase of the mesh, and have high quality optimisation on the phase representing the object of interest.