

# Instructions

Yu-Ping Yang

December 20, 2017

- The following five files will be output from the interface (under `inputs/` directory):

- `eweld.in`
- `eweld_weld_parameters.in`
- `eweld_boundary_condition.in`
- `eweld_preheat_interpass_temperature.in`
- `eweld_temperature_monitor.in`
- `eweld_mesh_key.txt` (Not need to do now. This option will allow users to input their own meshes. )

- For automatic mesh, the following steps will be run:

1. Check if `pass_coordinates.out` exists in `input` directory, if no, run `utils/determine_passes_arc_v4.exe` to create `inputs/pass_coordinates.out`<sup>1</sup>:  
`./utils/determine_passes_arc_v4.out inputs/eweld.in` will be input.
2. Run `utils/Automesh_v14.py` from SALOME's Python Console to create `Mesh_3D.unv`, or run without Salome GUI:

```
$SALOMEPath/salome start -t -w 1 utils/Automesh_v14.py
```

(a) The files will be input:

- `./inputs/eweld.in`
- `./inputs/eweld_weld_parameters.in`
- `./setting/Setting_arc_efficiency_default.in`

---

<sup>1</sup>On Linux, compile `determine_passes_arc_v4.out`, to get `determine_passes_arc_v4.out` via `gfortran determine_passes_arc_v4.f -o determine_passes_arc_v4.out`

- `./inputs/pass_coordinates.out`
- (b) The files will be output:
- `Mesh_3D.unv`
  - `model_dflux.for`
  - `model_step1.inp`
3. Run
- ```
python2 tools/unv2calculix.py Mesh_3D.unv Model3d
```
- `Model3d.inp` will be created.
4. To generate the `model_film.in` file (using `cgx` and `unical`), run:
- ```
./utils/createFilm.sh Mesh_3D.unv ./utils/write_film.fbd model_film.in
```
5. Run
- ```
python2 utils/Analysis_file_create.py
```
- The files will be input:
    - \* `./inputs/eweld.in`
    - \* `./inputs/eweld_boundary_condition.in`
    - \* `./inputs/eweld_preheat_interpass_temperature.in`
    - \* `./Model3d.inp`
  - The files will be output:
    - \* `model_bc.in`
    - \* `model_ele4.in`
    - \* `model_ele6.in`
    - \* `model_ele8.in`
    - \* `model_group.in`
    - \* `model_ini_temperature.in`
    - \* `model_material.in`
    - \* `model_node.in`
6. Move `model_dflux.for` to the CalculiX directory (to `CalculiX-PW/src/`) and rename to `dflux.f`, and compile CalculiX (see the notes in `CalculiX-PW/README.md` for compilation/installation of CalculiX). Or unzip `tools/CalculiX-PW.tar` (works under Ubuntu 14.04, if all the required packages for CalculiX are installed):
- ```
tar -xf tools/CalculiX-PW.tar
```
7. Run `analysis.inp` with CalculiX