SMITER GUI work notes on file conversion

Ivona Vasileska

February 2017

1 What is SMITER?

SMITER is technically SMARDDA (SMART DATA) for ITER, which is a software for predicting and calculation of power deposition by a tokamak plasma on limiters and divertors. SMARDDA uses a specially designed algorithm involving a special multi-octree type of hierarchical data structure (HDS). The SMITER algoritham is divided in two cases limiter case and divertor case. For both limiter and divertor cases, the geometry is divided into two types, the first is the *"results geometry"* the part where is calculated the power deposition and the *"shadowing geometry"* the part which protects the edges of the first by fieldline shadowing.

How SMITER algorithm works?

1. step: CAD database input using CADfix → Meshing → Mesh output of part body (.CATpart file → .dat file)

2. step: Conversion of .dat format .vtk format (OpenSource) by SMITER datvtk (geometry data)

3. step: .eqsk file that describes a tokamak magnetic field equilibrium (equilibrium data) which format is a "non-standard" standard for solutions of the Grad-Shafranov equation, with combination of .datvtk (geometry data) give the GEOQ code.

4. step: Processing the *"shadowing geometry"* with GEOQ code give the HDSGEN code. This code computes the multi-octree HDS (hierarchical data structure) introduced, which is designed to accelerate the computation of track/ray-triangle intersection.

5. step: The *"results geometry"* cannot be struck by a field line, the same tile triangulation can in fact be part of both launch and *"shadowing geometry"*. The fieldline tracing and power deposition calculations are performed by the POWRES code, or the POWCAL code. This is the code that actually performs the power surface deposition calculation, following fieldlines using transformed geometry from .geoq and the HDS from .hdsgen.

This tutorial describes how SMITER GUI can be used to perform SMITER simulation starting from (i) CAD `.stl` file describing limiter or divertor geometry, (ii) meshing by SMITER GUI mesher, and (iii) converting into legacy `.vtk` file for final simulation run. In the following tutorial we will be using the `smiter-aux` tests and by replacing the `.vtk` files with the mesh produced by using SMITER GUI. Then we compare output files resulting from POWCAL. With SMITER GUI we can confirm that the conversion of the geometry, meshing and file transforms in SMITER GUI worked right.

2 SMITER GUI - geometry and meshing module

In this section we present workflow starting with the `.stl` file describing geometry and meshing module of the SMITER GUI for the limiter case and divertor case. The SMITER GUI has a couple of modules which one can select by clicking on the icons at the taskbar. By choosing the geometry module and FILE → IMPORT TO → STL, we are importing the `.stl` file from CAD export. Figure 1 presented the `.stl` files in the SMITER GUI geometry module for (a) the limiter case from the test Test-EQ3-inrshad1-inres1 test-deck, and (b) divertor from the test Val-HR-bott-baff test-deck.

For conversion into the `.vtk` file format the geometry needs to be meshed. To select the *meshing module* click on CREATE MESH. By observing test-decks in `smiter-aux` are using triangle meshes we can create similar meshing density for our meshing to get comparable input meshes. With COMPUTE MESH the mesh is easily meshed from the geometry for both parts

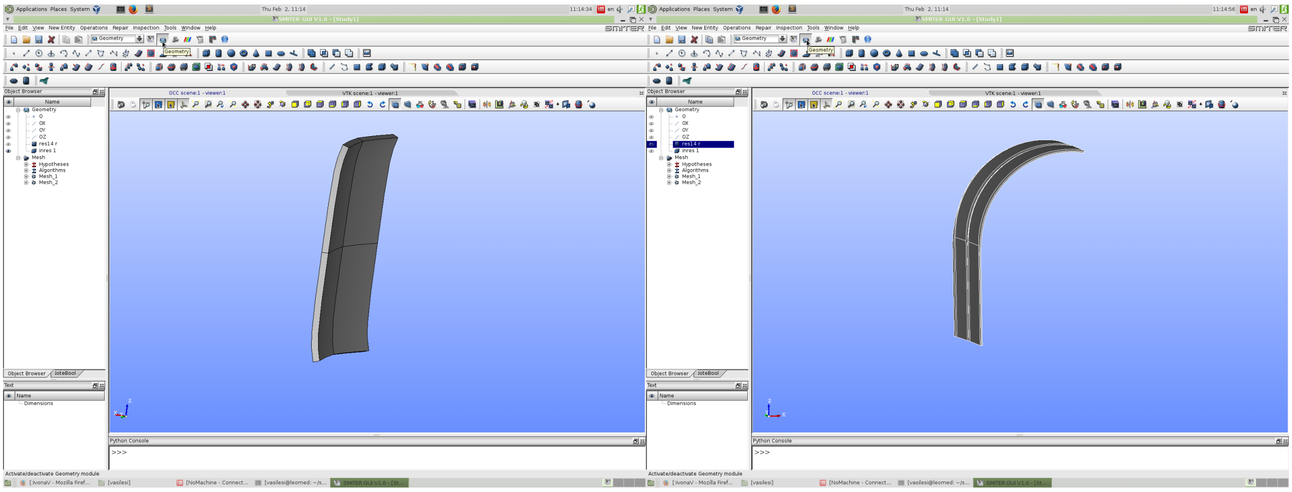

Figure 1: (a) - Geometry module for limiter .stl file, (b) - Geometry module for divertor .stl file

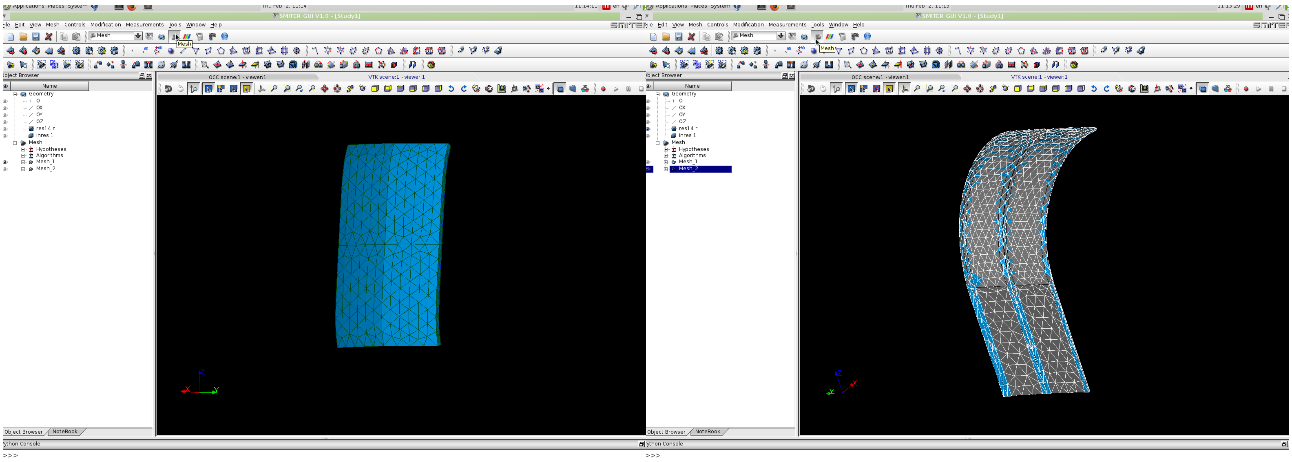

Figure 2: (a) - Mesh module for limiter, (b) - Mesh module for divertor

of the ITER tokamak (see Figure 2). In this first step we have described how the SMITER GUI can be used for relevant geometries.

3 SMITER GUI - VTK conversion

The second step is the part where the meshed geometry needs to be converted into `.vtk` file for SMITER simulation. For that we use the *Python console* in the SMITER GUI, where the user can write the Python code for any kind of interactive scripting which will help user. In our case we will write a Python code for converting `.stl` into `.vtk`.

The following `.stl` to `.vtk` converter script is used in Figure 3:

```
>>> import vtk
```

```
>>> reader = vtk.vtkSTLReader()
>>> reader.SetFileName("*location of the .stl file*")
>>> reader.Update()
>>> reader.GetOutput()
>>> *name = reader.GetOutput()
>>> print *name
>>> writer = vtk.vtkDataSetWriter()
>>> writer.SetFileName("*location of the .vtk file*")
>>> writer.SetInputData(*name)
>>> writer.Write()
>>>
```

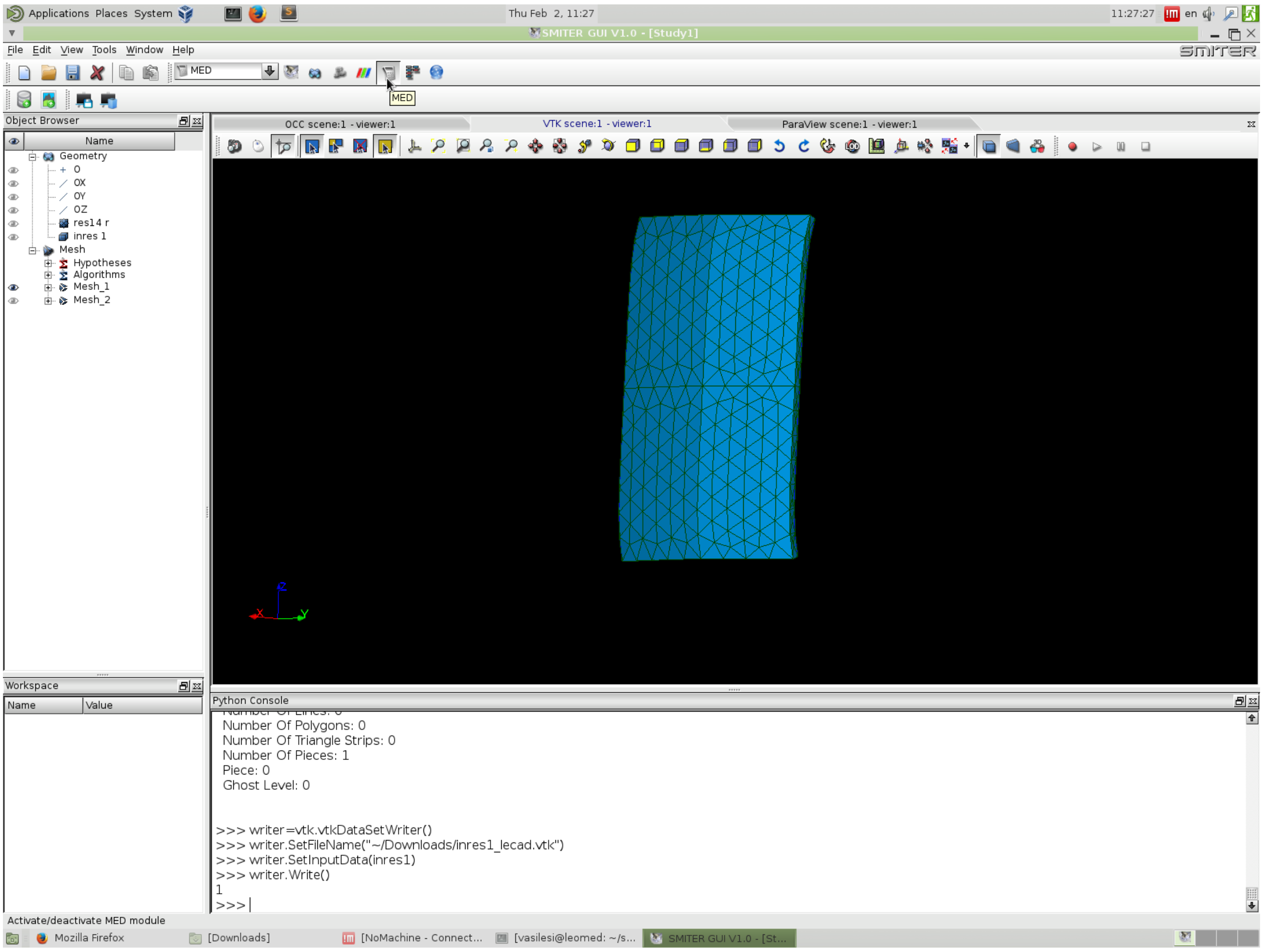

Figure 3: MED module for converting .stl to .vtk file for limiter case in Test-EQ3-inrshad1-inres1

The Python console can be used for typing a different codes. For example converter the .med file into `.vtk`, which can help us to convert the geometry from CAD into `med` file. The code for this type of conversion is:

```
>>> from MEDLoader import *
>>> cm = MEDLoader.ReadUMeshFromFile ("*location of the .med file*",
         "Mesh\textunderscore 1",0)
>>> cm.writeVTK("*folder place of the .vtk file*")
```

To convert `.med` to `.vtk.vtu` file we use

```
>>> import vtk
>>> reader = vtk.vtkXMLUnstructuredGridReader()
>>> reader.SetFileName("*location of the .vtk.vtu file*")
>>> reader.Update()
>>> print(reader)
>>> writer = vtk.vtkUnstructuredGridWriter()
>>> writer.SetFileName("*location of the .vtk file*")
>>> output = reader.GetOutput()
>>> print output
>>> writer.SetInputData(output)
>>> writer.Write()
>>>
```

4 SMITER GUI - VTK module

After the `.stl` file convertsion into legacy `.vtk` file we switch to ParaVis module in the taskbar of the SMITER GUI and open the `.vtk` file. We can compare it with the existing smiter-aux test deck. From the Figure 4 it is obvious that the workflow from the CAD filem, meshing and conversion worked correct. For the SMITER simulation we have replaced the `.vtk` files in the DATA sub-folder of the `smiter-aux` test deck and then run the tests with our meshes. Then we have compared the output of the power deposition test saved in .vtk files. In Figure 5 we present the power deposition results of the limiter case, (a) using original SMITER aux test (a), and (b) the meshes produced with the SMITER GUI.

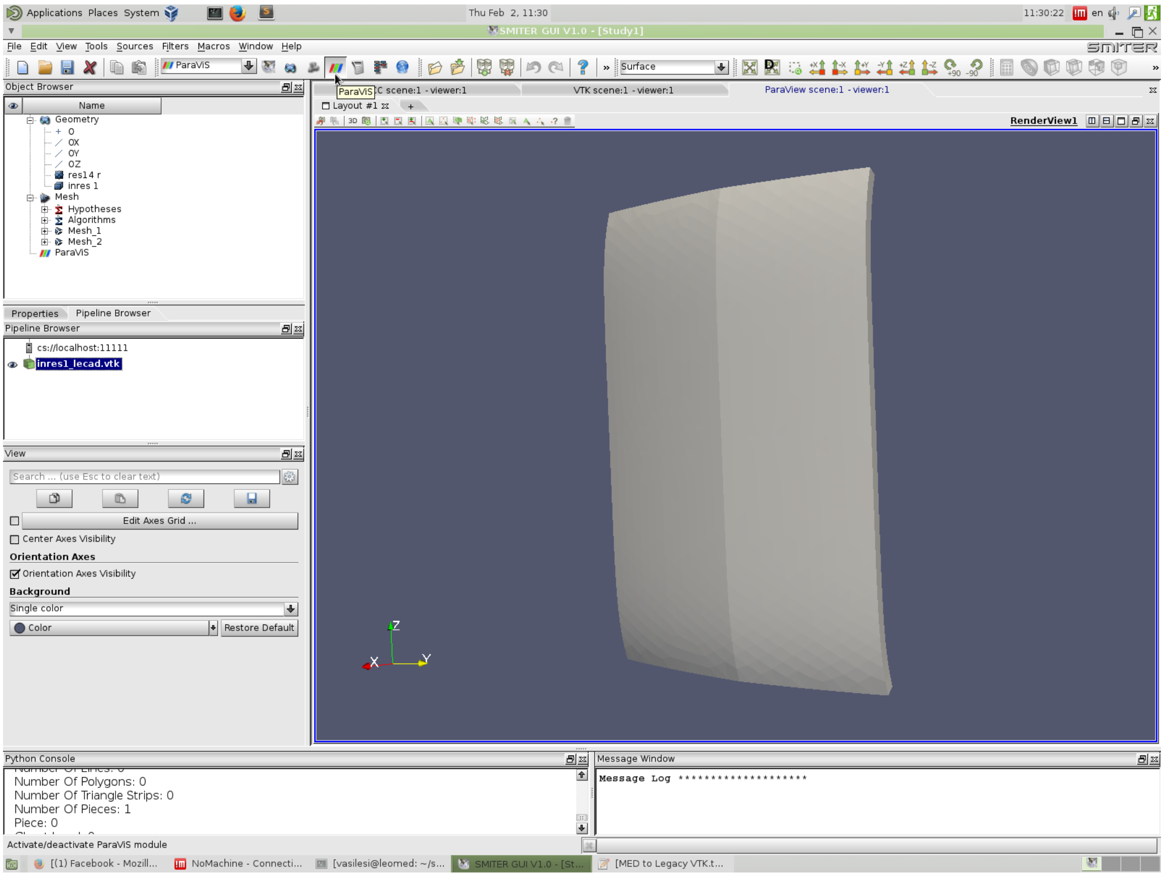

Figure 4: .vtk file for limiter case in Test-EQ3-inrshad1-inres1

5 Meshing comparison

Also there was done second type of meshing using ICEM mesher. For this meshing the CAD model was already meshed before with ICEM and then was saved in `.stl` file. After that was done the same steps which are explained before. We convert to `.vtk` file and we do not need to use the SMITER mesher. Replaced the this file in `smiter-aux` test deck and then run the test. The resultants of the power deposition are presented in Figure 6, where (a) is the original SMITER aux test, (b) the meshes produced at ICER mesher and (c) the meshes produced with SMITER GUI.

6 SMITER TUI

SMITER TUI is Text-based User Interface which can provided the POWCAL files. This interface helps the user to prepare the input files for the power calculation which is the base for determination of the power deposition and fildline data. To get the POWCAL files in this interface, the user need just to input the CAD files and the EQUILIBRIUM files. To run

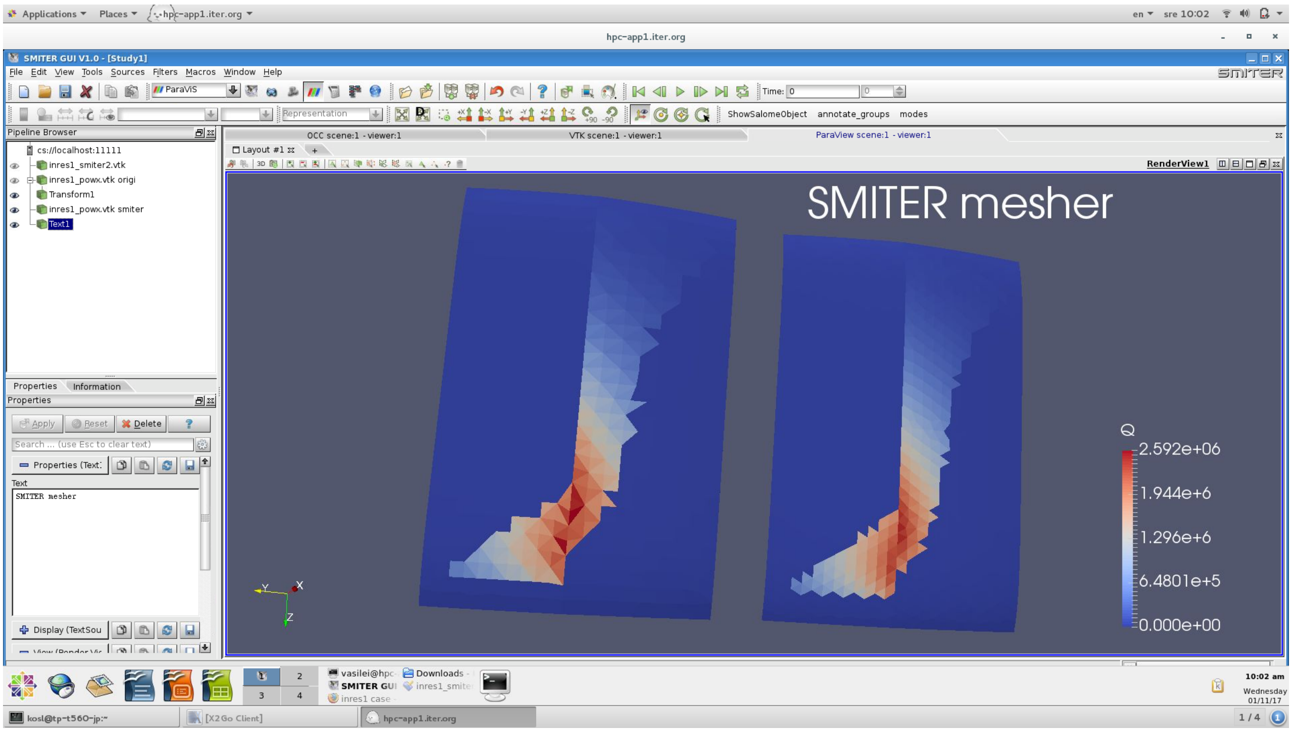

Figure 5: Power deposition files of the limiter case in (a) smiter-aux Test-EQ3-inrshad1-inres1 test deck, and (b) SMITER GUI meshing.

SMITER-TUI simply need to go to the folder that contains the executable file ismiter. For e.g. with the command: cd /smiter/exec./ismiter In Figure 7 is shown the beginning window where the user can choose what kind of mark actions want to use it. If the user chose the `Select geometry and equil files` mark action it is open the follow window 8. In this window easily can be changed the parameters depending what kind of case the user is running for providing the POWCAL files.

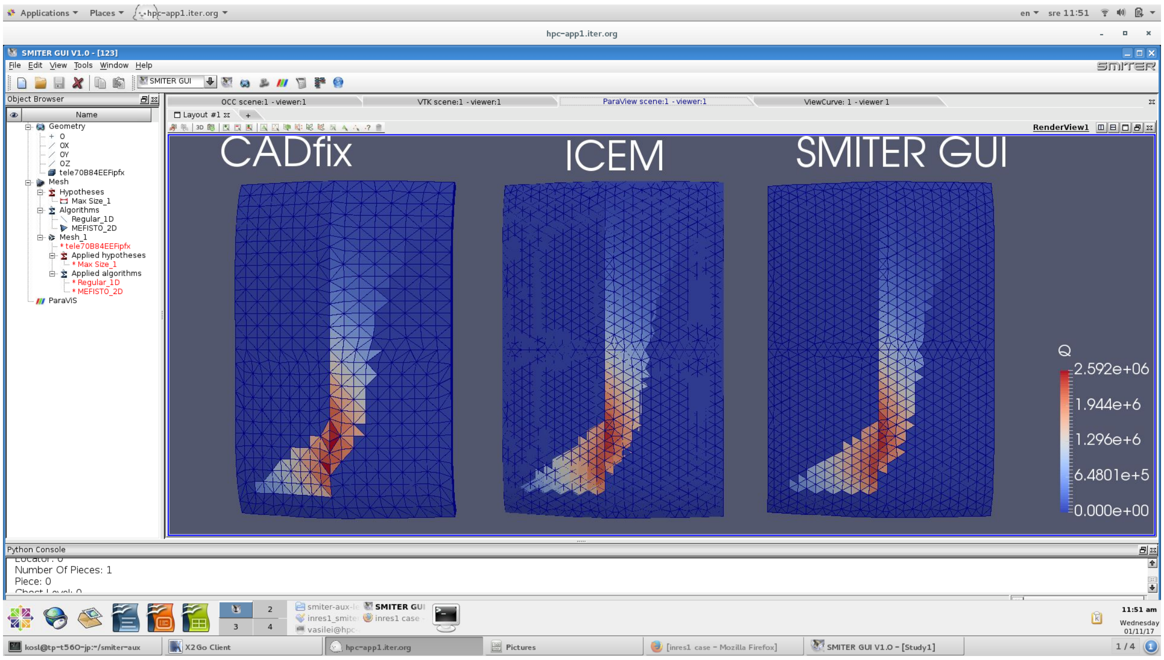

Figure 6: Power deposition files of the limiter case in (a) smiter-aux Test-EQ3-inrshad1-inres1 test deck, (b) ICEM meshing, (c) SMITER GUI meshing

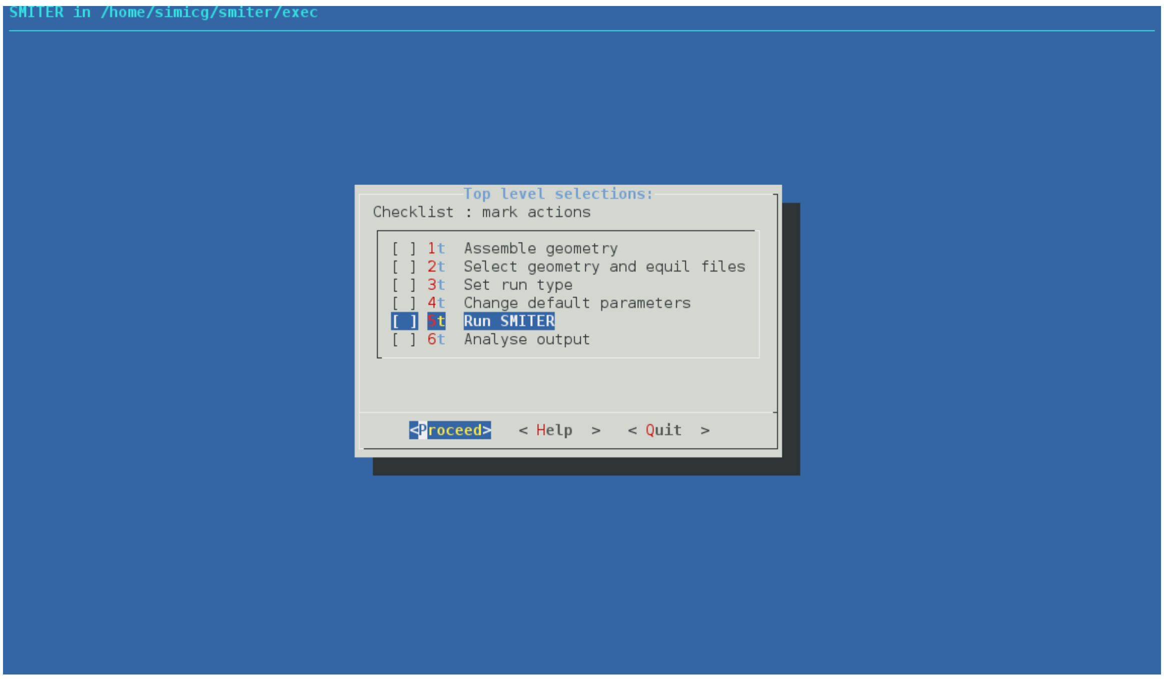

Figure 7: Beginning window of SMITER TUI

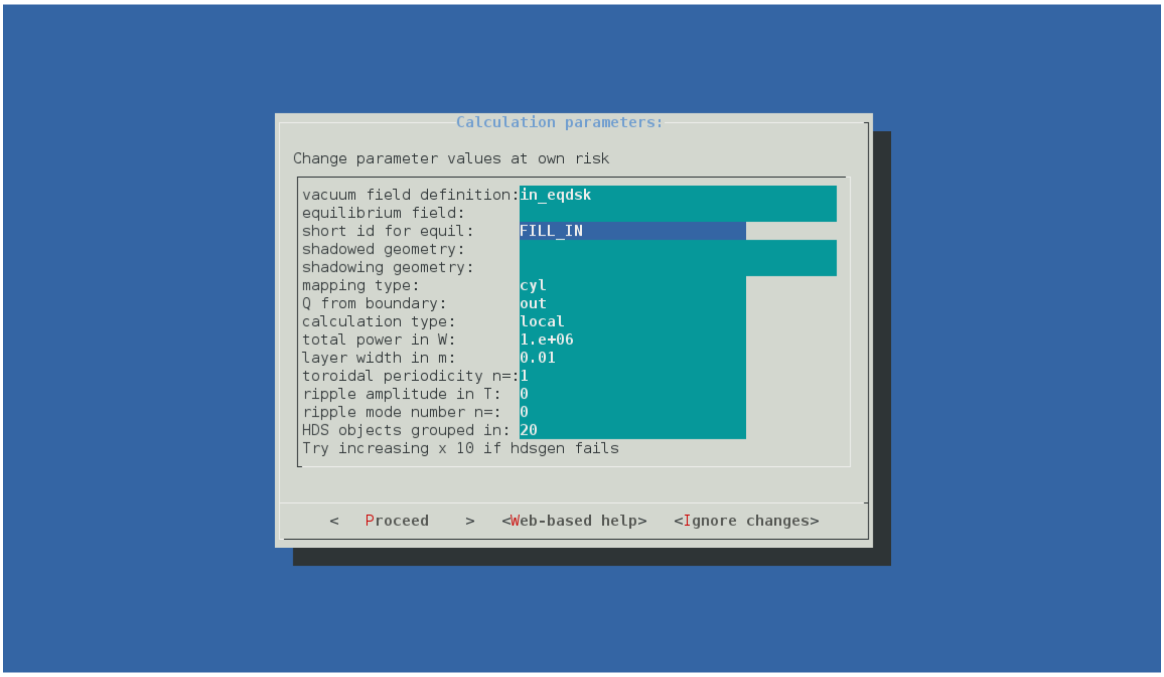

Figure 8: Calculation parameters in SMITER TUI