Mesh Generation and Adaptation for M3D-C[1]

Fan Zhang, Mark S. Shephard, and E. Seegyoung Seol
September 2, 2015

1. Introduction

This document describes how to generate a mesh with an enclosed vacuum vessel domain and how to perform mesh adaptation in M3D-C[1]. All the examples shown in this document are made in PPPL *portalr6* except for converting the initial Simmetrix mesh (.sms) into .vtk and .smb files used for Paraview and PUMI, respectively.

Section 2 describes how to set up a connection and environments. Section 3 presents how to generate a model and mesh files used in M3D-C[1]. Section 4 describes how to run mesh adaptation in M3D-C[1].

2. Setting-up

2.1. Login

For remote users, follow http://w3.pppl.gov/~jardin/M3DC1/NEWDOC-latest.pdf (Section "Accessing computers at PPPL"), and create NX connection.

Please open a terminal for each of *portalr5 and portalr6.*

2.2. Setting up environmental variables

[portalr5]

```
ssh –X portalr5
module load intel/11.0.081
module load openmpi/1.4.3-qlogic
```

[portalr6]

```
ssh –X portalr6
module load intel/2015.u1
module load openmpi/1.8.4
setenv RLM_LICENSE "2800@v-cluster01.pppl.gov:/usr/pppl/Simmetrix/simmodsuite.lic"
module load paraview
```

3. Mesh Generation

3.1. Programs

- /p/tsc/m3dc1/lib/SCORECLib/rhel6/utilities/create_mesh/create_smd

    - purpose: generating simmetrix model (.smd), PUMI model (.txt) and simmetrix mesh (.sms)
    - run in portalr6

- /p/tsc/m3dc1/lib/SCORECLib/rhel5/utilities/convert_sms/convert_sms
    - purpose: converting simmetrix mesh (.sms) to Paraview (.vtk) for visualization
    - run in portalr5

- /p/tsc/m3dc1/lib/SCORECLib/rhel5/utilities/split_sms/split_sms
    - purpose: converting simmetrix mesh (.sms) to PUMI (.smb) and split to N parts
    - run in portalr5

- /usr/pppl/Simmetrix/SimModeler4.0-150404/simmodeler
    - purpose: visualizing simmetrix model (.smd) and mesh (.sms)
    - run in portalr6

- paraview
    - purpose: visualizing .vtk files

The following subsections describe the examples to generate the mesh on the domain with an enclosed vacuum vessel. Example files and readme are available in */p/tsc/m3dc1/lib/develop.petsc3.Fan/MeshDemo.* See *readme* in each subfolder.

For more information on meshing, see http://redmine.scorec.rpi.edu/projects/epsi/wiki/Mesh_generation_for_M3D-C1_and_XGC

3.2. Example 1: NSTX-1

- Location: */p/tsc/m3dc1/lib/develop.petsc3.Fan/MeshDemo/NSTX-1*

- To run:

```
cp /p/tsc/m3dc1/lib/SCORECLib/rhel6/utilities/create_mesh/create_smd your_folder
cd your_folder
./create_smd
```

- Input:
    - The file *"input"* with the meshing control parameters
        - *modelType*: 0 for interpolate analytic model, 1 for piece-wise, 2 for three region model, 3 for piece-wise polynomial
        - *modelName*

        - *pointFile*: a file to describe the geometry
        - *meshSizes*: mesh size for the plasma (used by all types), wall (used by type 2) and vacuum (used by type 2) areas
        - *meshGradationRate*
        - *numInterPts*: parameter for type 0
        - *thickness*: wall thickness, only for type 2
        - *height/width*, *offsetX, offsetY*: for vacuum vessel in type 2

    - *NSTX*: the ordered set of points on the wall boundary.
        - In the file *"input"*, parameter *"pointFile"* is set to "NSTX".

- Output
    - *NSTX0.02.smd*: Simmetrix model file.
    - *NSTX0.02.txt*: geometric model file used by M3D-C[1]
    - *NSTX0.02.sms*: mesh in PUMI .sms format
- Converting .sms mesh to .vtk and visualizing in portalr5

```
cp /p/tsc/m3dc1/lib/SCORECLib/rhel5/utilities/convert_sms/convert_sms your_folder
cd your_folder
./convert_sms NSTX0.02.sms mesh.vtk
module load paraview
paraview&
```

- Converting .sms mesh to .smb mesh partitioned into N parts in portalr5

```
mpirun -np N /p/tsc/m3dc1/lib/SCORECLib/rhel5/utilities/split_sms/split_sms NSTX0.02.sms
mesh.smb
```

To use the geometry and the mesh by M3D-C[1], specify model/mesh file in C1input

- *Set the parameter "mesh_model" to "NSTX0.02.txt"*
- *Set the parameter "mesh_filename" to "mesh.smb"*

Alternatively, an initial mesh can be generated through simmodeler. Launch simmodeler in portalr6. The model file, *NSTX0.02.smd,* can be opened through simmodeler. The mesh generated is saved as Simmetrix .sms file and can be converted to .smb file by the following steps.

- Converting a serial Simmetrix mesh (.sms) to PUMI mesh (.smb) in portalr6

```
cp /p/tsc/m3dc1/lib/SCORECLib/rhel6/utilities/create_mesh/convert_sim_sms your_folder
cd your_folder
./convert_sim_sms NSTX0.02.smd NSTX0.02.sms in-mesh.smb
```

- Splitting a serial PUMI mesh (.smb) to N parts in portalr6

```
cp /p/tsc/m3dc1/lib/SCORECLib/rhel6/utilities/split_smb/make_model your_folder
cp /p/tsc/m3dc1/lib/SCORECLib/rhel6/utilities/split_smb/split_smb your_folder
cd your_folder
mpirun –np 1 ./make_model in-mesh.smb model.dmg
mpirun –np N ./split_smb model.dmg in-mesh.smb mesh.smb N
```

3.3. Example 2: NSTX-2

Change "*meshSizes*" in *input* from "0.05 0.1 0.1" to "0.05 0.1 0.2" and see coarser mesh in the vacuum area.

3.4. Example 3: NSTX-3

Change parameters "*width*", "*height*", and "*offsetX*", to adjust the size and position of the vacuum area.

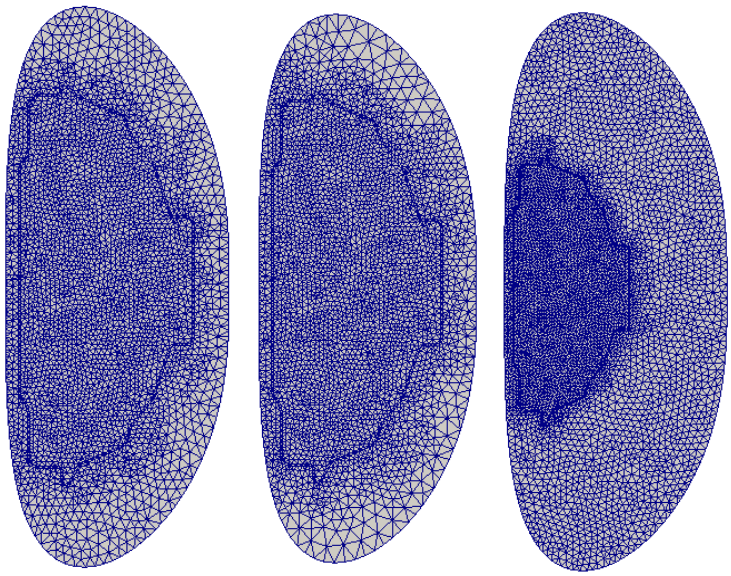

Figure 1 Mesh under NSTX-1, NSTX-2, NSTX-3

3.5. Example 4: NSTX-4

Due to improper parameters in *input*, the geometry will be invalid. Therefore, you will see the error message.

"Error: Code: 1103 String: Unable to mesh face 3"

Visualize the model with simmodeler and change the parameters to shift the vacuum curve.

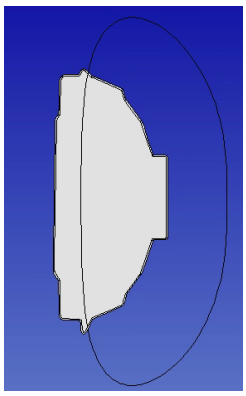

Figure 2 Invalid geometry

4. Mesh adaptation

The mesh is adapted to match the mesh size field defined either by post-processed magnetic flux field in the equilibrium or the estimated error [1].

The examples presented in this document are generated in *portalr6* and available in

/p/tsc/m3dc1/lib/develop.petsc3.Fan/Aug26/DATA/adapt/

The examples are also available under M3D-C^1 repository/DATA/adapt.

4.1. Control parameters for adaptation in *C1input*

- ***iadapt***
  - 0: no adaptation
  - 1: adapt mesh from the magnetic flux field in the equilibrium
  - 2: adapt mesh from the estimated error in the solution field.
- ***adapt_hmin, adapt_hmax***
  - maximum and minimum sizes of the mesh elements in the adapted mesh.
- ***adapt_hmin_rel, adapt_hmax_rel***
  - bounds of a mesh element that can be changed from its original size in the adapted mesh (rel=relative).
- ***adapt_target_error***
  - target discretization error on the adapted mesh.
- ***iadapt_order_p***
  - target mesh size of a mesh element is proportional to the original mesh size as $(\tau/h_{org})^{-p-1}$ [2], where τ is the estimated error contributed by the mesh element. The value is no larger than 3 in H^2 space for M3D-C^1 [3].
- ***iadapt_max_node***
  - maximum node number in the adapted mesh. If the estimated mesh node number from *adapt_target_error* exceeds *iadapt_max_node,* the target mesh size in the adapted mesh is scaled such that the mesh node number is below *iadapt_max_node.*
- ***adapt_ke***
  - adapt the mesh when kinetic energy exceeds the value*.* If *adapt_ke* is set to 0, this adaptation criterion is ignored.
- ***iadapt_ntime***
  - adapt the mesh every *ntime* steps*. .* If *iadapt_ntime* is set to zero, this adaptation criterion is ignored.
- ***adapt_control:***
  - 0: *adapt_target_error* is global (integral over the domain) [2]
  - 1: *adapt_target_error* is local (integral over the element)
- ***iadapt_useH1:***
  - set value to 1 if fluid viscosity and electrical resistivity << 1

- ***iadapt_removeEquiv:***
  - set value to 1 to remove the terms containing the equilibrium solution in the estimated error
- ***iadapt_writevtk/iadapt_writesmb***
  - set value to 1 to output the adapted mesh as the vtk/smb file.

4.2. Anisotropic mesh by magnetic flux field in the equilibrium

This section describes the mesh adaptation with the post-processed magnetic flux field in the equilibrium.

- The mesh size field is defined as:

$$\tilde{\psi} \equiv \frac{\psi - \psi_0}{\psi_l - \psi_0}, \qquad \begin{array}{l} \psi_0 = \text{ value at magnetic axis} \\ \psi_l = \text{value at plasma boundary} \end{array}$$

inside plasma: $\tilde{\psi} < a_1$ exterior to plasma: $\tilde{\psi} > a_1$

$$\tilde{h}_1 = a_{4P}\left[1 - e^{-\left|\frac{\tilde{\psi}}{a_1}-1\right|^{a_2}}\right] + a_7 \qquad \tilde{h}_1 = a_{4V}\left[1 - e^{-\left|\frac{\tilde{\psi}}{a_1}-1\right|^{a_3}}\right] + a_7$$

$$\tilde{h}_2 = a_{5P}\left[1 - e^{-\left|\frac{\tilde{\psi}}{a_1}-1\right|^{a_2}}\right] + a_6 \qquad \tilde{h}_2 = a_{5V}\left[1 - e^{-\left|\frac{\tilde{\psi}}{a_1}-1\right|^{a_3}}\right] + a_6$$

$$h_i^{-1} = \tilde{h}_i^{-1} + \frac{1}{l_{ci}}\left[\frac{1}{1 + \left(\frac{\tilde{\psi} - \psi_c}{W_c}\right)^2}\right]; \qquad i = 1,2$$

Note that h_1 is the length normal to the surfaces and h_2 the length tangential.

In your work directory should specify 13 size field parameters in the file "*sizefieldParam*" in the following order (single line, space delimited)

a_1 a_2 a_3 a_{4p} a_{4v} a_{5p} a_{5v} a_6 a_7 l_{c1} l_{c2} W_c ψ_c

- After adaptation, the initial equilibrium is re-calculated on adapted mesh so the analysis can continue
- The adapted mesh is saved in "adapted*.vtu" and/or "adapted*.smb. The * represents the global part ID starting from 0.
- Parallel anisotropic mesh size smoothing is not supported. (Will be available later)

- Change parameters in *sizefieldParam* to obtain different meshes. For example, a_6/a_7 affects the aspect ratio of the element in the adapted mesh at the flux surface with the normalized psi value equaling to a_1.

4.2.1. Example in adapt/anisotropic (4 processes)

Plasma equilibrium is obtained by Grad-Shafranov solver in M3D-C[1].

[C1input]
iadapt=1
iadapt_writevtk=1

[sizefieldParam]
0.9 2 1 .05 .5 .05 .5 .1 .01 5. 5. 0.3 0.148

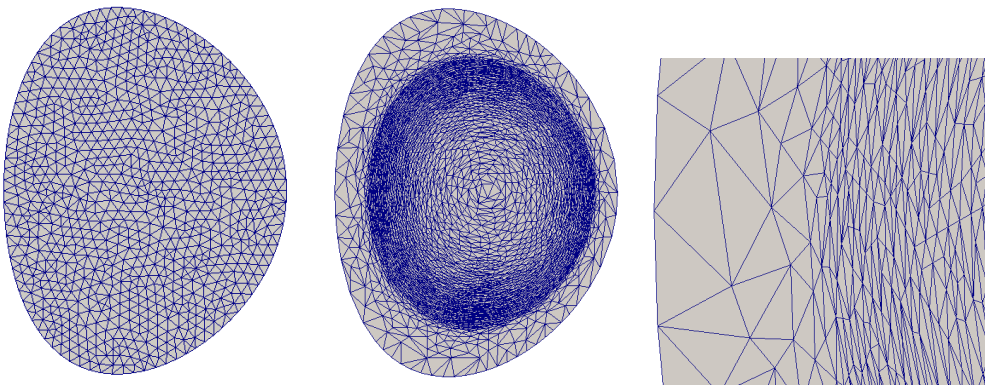

Figure 3 Initial meshes, adapted mesh and its close-up under folder adapt/anisotropic

4.2.2 Example in adapt/anisotropic2 (4 processes)

[C1input]
iadapt=1
iadapt_writevtk=1

[sizefieldParam]
0.8 2 1 .05 .5 .05 .5 .1 .02 5. 5. 0.3 0.148

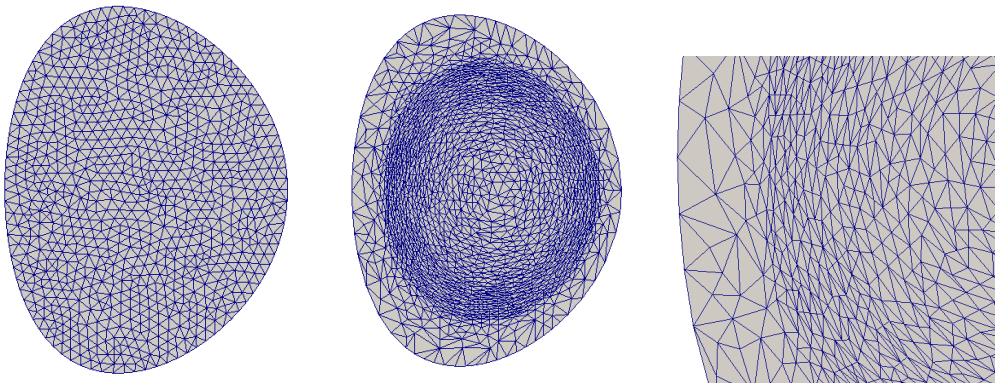

Figure 4 Initial meshes, adapted mesh and its close-up under folder adapt/anisotropic2

4.3. Time-dependent adaptive mesh in adapt/tilt

This section describes the mesh adaptation example with the error estimation using the tilt mode. The solution is transferred to the new mesh in the non-linear simulation.

```
[C1input]
mesh_filename = tilt.smb
mesh_model = tilt.txt
iadapt = 2
iadapt_ntime = 4
adapt_target_error = 0.02
adapt_control = 0
iadapt_max_node = 600
iadapt_writevtk = 1
iadapt_order_p = 1.5
adapt_hmin = 0.03
adapt_hmax = 0.4
ntimemax=200
```

To visualize the result, launch IDL and enter the following

```
plot_field, 'jphi',
 file='/p/tsc/m3dc1/lib/develop.petsc3.Fan/Aug26/DATA/adapt/tilt/C1.h5',
00, /mesh
```

In order to visualize *jphi* field on the adapted mesh in different time slice, change 00 to 30, 60, 90, or 200.

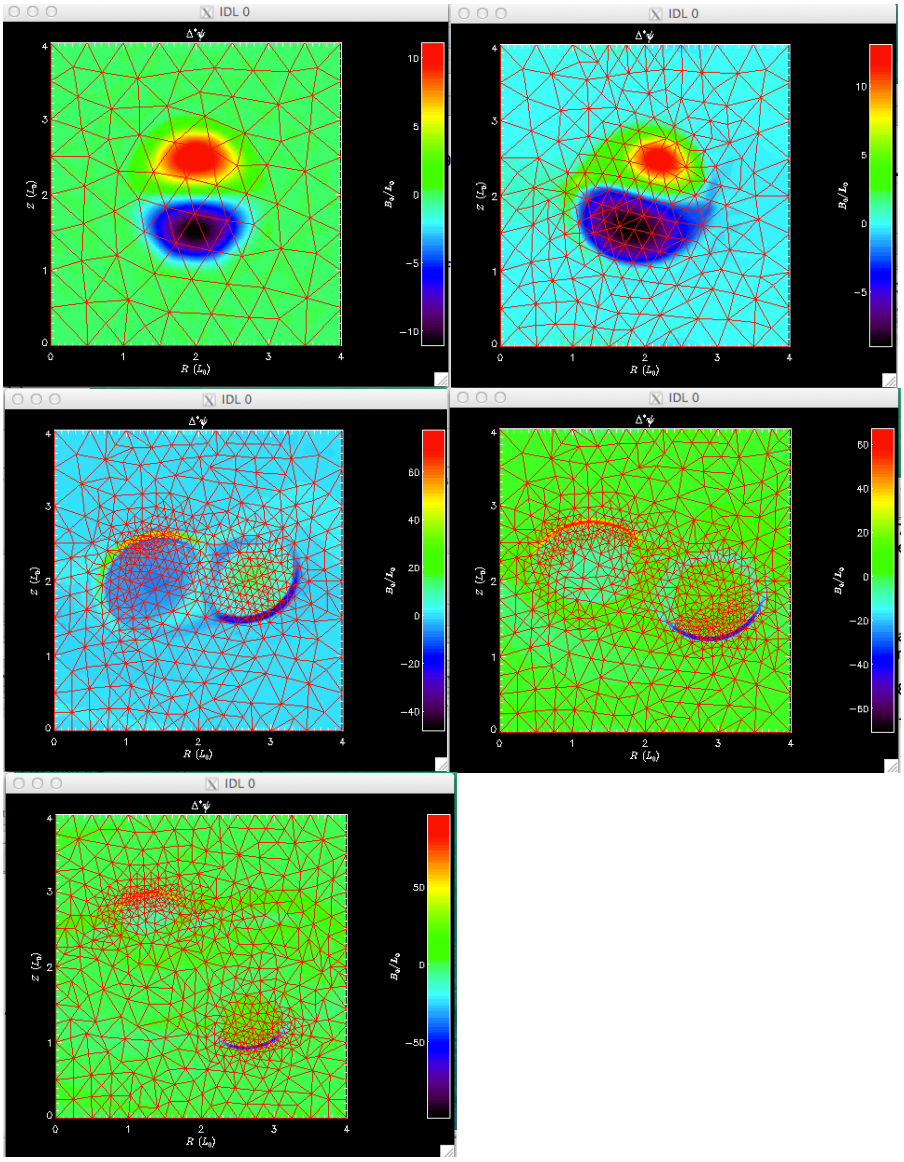

Figure 5 *jphi* field on the adapted mesh in time step 00, 30, 60, 90 and 200

4.4. Adapted mesh from eigen mode

The mesh can be adapted from the estimated error in the eigen mode. Two examples are available.

4.4.1 Example in adapt/ELM (8 processes)

[C1input]
mesh_filename = Analytic.smb
mesh_model = AnalyticModel
iadapt = 2
iadapt_max_node = 15000
iadapt_useH1=1
iadapt_removeEquiv =1
adapt_target_error = 0.005
adapt_hmin = 0.005
adapt_hmax = 0.1
adapt_ke = 5e-2

iadapt_order_p = 2
iadapt_writevtk = 1

In order to visualize the growth rate on the adapted mesh, launch IDL and enter the following

```
plot_scalar, 'ke',
file='/p/tsc/m3dc1/lib/develop.petsc3.Fan/Aug26/DATA/adapt/ELM/C1.h5',
/growth, yrange=[0.,0.15]
```

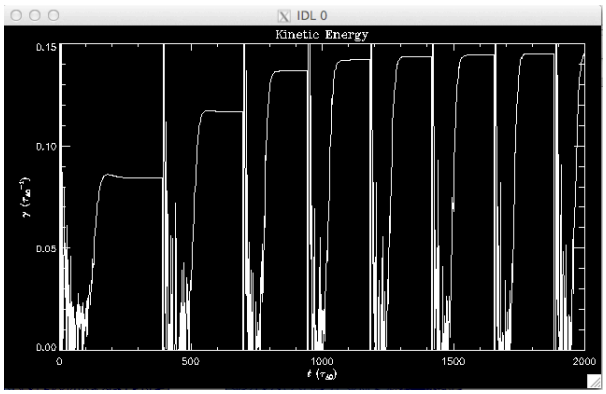

Figure 6 Growth rate of kinetic energy on the adapted mesh (the mesh is adapted 7 times)

To start a new simulation with the adapted mesh,

```
cd adapt
mkdir ELM2
cd ELM2
cp  ../ELM/adapt943*smb .
./change_name.sh adapt943 adapt 8
Copy the input files from adapt/ELM
Modify C1input to set mesh_filename to "adapt.smb"
Run the simulation
```

Enter the following command in IDL to view the growth rate.

```
plot_scalar, 'ke',
file='/p/tsc/m3dc1/lib/develop.petsc3.Fan/Aug26/DATA/adapt/ELM2/bk/C1.h5', /growth,
yrange=[0.145,0.15]
```

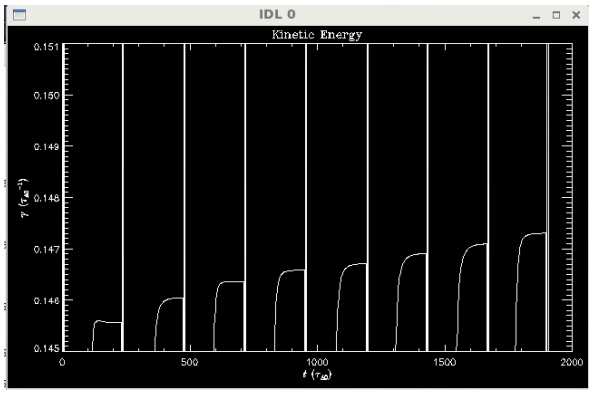

Figure 7 Growth rate of kinetic energy starting from the previously adapted mesh

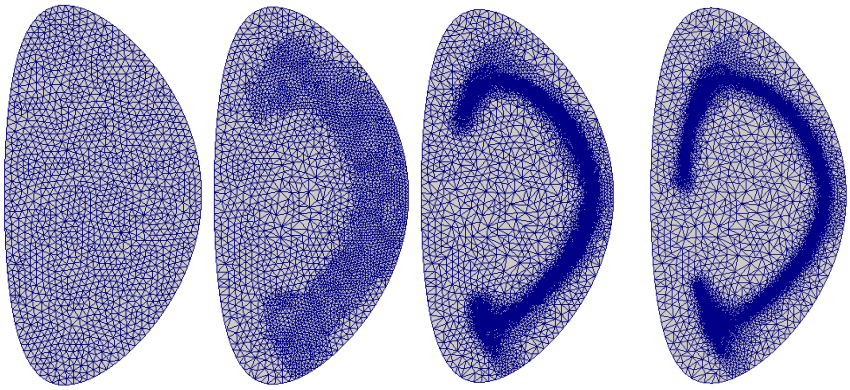

Figure 8 Adapted meshes for ELM after 0, 1, 6, 13 times adaptation

(Visualized with Paraview, note that the current IDL will only show mesh at time 0 for linear runs)

4.4.2 Example in adapt/*doubleTearing* (8 processes)

mesh_filename = Analytic.smb
mesh_model = AnalyticModel
iadapt = 2
iadapt_order_p = 2
adapt_target_error = 1e-16
adapt_hmin = 0.01
adapt_hmax = 0.1
adapt_hmin_rel = 0.3
adapt_hmax_rel = 3.
iadapt_max_node = 8000
iadapt_useH1=1
iadapt_removeEquiv =1
iadapt_writevtk = 1
iadapt_ntime = 300

Exercise: start a new doubleTearing simulation with the adapted mesh

References

1. M. Ainsworth and J. T. Oden, "A posteriori error estimation in finite element analysis," Comput. Methods Appl. Mechanics Eng., vol. 142, no. 1, pp. 1–88, Mar. 1997
2. E. Onate and G. Bugeda, "A study of mesh optimality criteria in adaptive finite element analysis," Eng. Comput., vol. 10, no. 4, pp. 307–321, Dec. 1993.
3. T. J. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Mineola, NY, USA: Dover Publications, 2012.