SPECFEM2D_on_docker_with_boat Documentation

masaru nagaso

May 16, 2019

CONTENTS:

1 Introduction 1

1.1 SPECFEM2D on Docker with analytics and visualization tools. 1

2 Installation and configuration 3

2.1 How to setup this library on your computer. 3

2.2 0. Initial configration for docker. (only for Windows) 3

2.3 1. Install Docker 3

2.4 2. Install Gmsh 3

2.5 3. Import docker image or build it from initial 3

3 Tutorial 5

3.1 1. Check if the docker daemon is running. 5

3.2 2. Start docker container 5

3.3 3. Open jupyter lab on your internet browser (chrome etc.) 5

3.4 4. Explanation about Jupyter lab 6

4 Meshing tutorial 7

4.1 Example 1: homogeneous medium with a hole 7

4.2 Example 2: heterogeneous medium (e.g. hole filled with oil) 10

4.3 Example 3: 3D mesh 11

5 Indices and tables 13

CHAPTER

ONE

INTRODUCTION

1.1 SPECFEM2D on Docker with analytics and visualization tools.

This library is an utility tool for SPECFEM2D (official repository). The original code of SPECFEM2D requires to compile and use it on only linux based operating system. This library allow users to use SPECFEM2D on any kind of operating systems including Windows, by using Docker.

Thanks to a docker script, users may compile and install SPECFEM2D with only one single command, instead of checking/installing several dependency.

This library includes an user interface which allow to configure the calculation settings, instead of modifying several plane text files. Some simple post-processing functions, e.g. visualization of signals, making movies etc., is provided as well. As using Jupyterlab as the base of the user interface, the user may additionally implement the pre/post processing functions for their purposes.

CHAPTER

TWO

INSTALLATION AND CONFIGURATION

2.1 How to setup this library on your computer.

This library is developed for a use with Docker system, thus firstly a few configuration of the system are required only for Windows 10 users.Users of other operating systems e.g. may skip the step 0 below and may start from step 1.

2.2 0. Initial configration for docker. (only for Windows)

Windows need to be configured as following the official reference. Here is a summary.

- At first *visualization technology (VTx)* need to be enabled from BIOS setup menu.
- Then after Hyper-V needs to be enabled also.

2.3 1. Install Docker

Download and install Docker (or Docker-ce for Windows) in your computer. Download link and instructions can be found here depending on your OS type.It is necessary to create an user account of Docker site.During the installation, please *DO NOT* check the option to use windows container.

2.4 2. Install Gmsh

From here download the gmsh executable. After extraction, place the gmsh executable in somewhere you prefere.

2.5 3. Import docker image or build it from initial

In this step, user prepare the docker image, which is the pre-composed linux environment with all dependencies of linux libraries and python modules.

For this way, users need to have two files,

```
wasi_2D_v1.0.tar
SPECFEM2D_on_windows_with_boat-master.zip
```

Please contact to Masaru NAGASO mnsaru22@gmail.com to obtain these files.

After obtaining these files, user needs to start docker engine (refering *the first section of this page*).

Locate the file path `wasi_2D_v1.0.tar` and `wasi_3D_v1.0.tar`.

Next, extract `SPECFEM2D_on_docker_with_boat-master.zip` and `SPECFEM3D_on_docker_with_boat-master.zip` need to be extracted in somewhere.

For example, this document continues assuming the positions of two files are in `ThisPC\Documents\wasi` as,

Open `power shell` window and navigate to the place where `wasi_2D_v1.0.tar` and `wasi_3D_v1.0.tar` are placed with the command below:

`cd Documents/wasi`

Run the docker command below on a `powershell` to load the docker image into user's docker environment.

```
docker load -i wasi_2D_v1.0.tar
```

```
docker load -i wasi_3D_v1.0.tar
```

Now all the environment for making a simulation with toolbox-wasi is set.

Explanation on how to start WASI is described in *Tutorial* page.

CHAPTER

THREE

TUTORIAL

In this page, an example of entire calculation with this library is indicated with a small geometry.

3.1 1. Check if the docker daemon is running.

On linux, it is ok if the command `docker ps` (in the terminal) indicates a list of docker images. On OSX and Windows, the state of the daemon can be seen on the menu bar as this image:

3.2 2. Start docker container

Initialization of the colculation environment for use of this library is quite simple.At first, please open the terminal or Power shell (windows) and navigate it to the directory of this library by running the command (file location is assumed to be at `Documents/wasi`)

```
cd ~/Documents/wasi/SPECFEM2D_with_boat-master
```

Then start the docker container (the environment packaged with docker) by the command

```
docker-compose up
```

This command will be finished instantly.After finishing it, you may see the indication like below:

When starting 3D wasi, we can do the same thing with just modify the file path as

```
cd ~/Documents/wasi/SPECFEM3D_with_boat-master
```

then

```
docker-compose up
```

3.3 3. Open jupyter lab on your internet browser (chrome etc.)

After starting the docker container, user may open Jupyter lab screen on a web browser by accessing the url below:

Wasi 2D

```
127.0.0.1:8002
```

and Wasi 3D

```
127.0.0.1:8001
```

3.4 4. Explanation about Jupyter lab

The appearance of Jupyter lab is below,

A file browser is on the left side. At the initial time, you may find only one single example directory on the list. You can add any files or directory from the buttons above or by drag and drop directly on this browser.Some buttons on the right side are for creating a new jupyter notebook file or a text file, or starting python console.A terminal window may be openned from the button below, which is directly opened form the docker environment thus it is useful to e.g. installing additional libraries or erasing files etc.

`Example_small` directory includes the files for making a small simulation as an example. After entering this folder by double clicking in the file browser

Detailed explanation on how to use the user interface etc. are described in the notebook file `example_small.ipynb` in this directory.

CHAPTER

FOUR

MESHING TUTORIAL

In this page, reader may find a simple example of meshing method with Gmsh. As the mesh generation function is separated from the function of Toolbox wasi, here only a minimum amount of examples are provided. Users may go to the official documents of Gmsh to find further functions of this great software Gmsh.

4.1 Example 1: homogeneous medium with a hole

At first, we will create a mesh for modeling a homogeneous medium with a hole as the image below:

This model has a rectangular shape and a hole in the rectangular. this hole will be a void (no material), thus this circular boundary will reflect 100% of energy. This kind of meshing may be used for modeling 2 materials which has greatly different impedance e.g. metal and air.

4.1.1 1-1: setting of characteristic length

After openning Gmsh application, we starts from creating a new geometry file from “New” at the menu bar above the application window:

For the place to place the geometry file, it is useful to put it in the same directry with the Jupyter notebook file (.ipynb) of your simulation. A geometry file in .geo extension will be generated at the selected place.

When the new geometry file is opened, Gmsh ask us which type of geometry kernel is used for the geometry definition (`OpenCASCADE` or `Built-in`).

We can use both type of geometry kernels for meshing. For this example, we select `OpenCASCADE` as this kernel may do boolean operation easily than the other kernel.

Then we define parameters which will used to decide the mesh size or some part of geometry. In this example, we will make 2 parameters:

```
lc :  element size
rc :  radial of a hole
```

For this step, you may put any name for any parameter as you want.

Parameters may be defined from the left function tree, Modules->Geometry->Elementary entities->Add->Parameter.

In this image, the parameter `lc` is defined with the initial value `0.0005`.In `Name` box, you can put the name of this parameter.`Label` and `Path` may be used for grouping the parameters which is useful for handling many parameters. We don’t use them for this example as we use only 2 parameters.

4.1.2 1-2: Definition of nodes

Here we define four corner points composing the rectangular shape at:

```
point 1 = 0.0, 0.11
point 2 = 0.0, 0.0
point 3 = 0.15, 0.0
point 4 = 0.15, 0.11
```

For define a node, we use the function Modules->Geometry->Elementary entities->Add->Point:

The image above is the window defining the first node. In the forth box for defining a mesh size around the defining node, we can put the parameter names defined in the former step. After filling these boxes, pushing `Add` button creates a node.

This image above shows the 4 nodes defined in this step. Initially the names of nodes are not indicated. To do so, configure Tools on the upper menu bar->Tools->Options->Geometry->Visibility.

4.1.3 1-3: Definition of lines

After defining the node points, we will define lines by connecting two nodes for one line. We define 4 lines:

```
line 1 = point 1, 2
line 2 = point 2, 3
line 3 = point 3, 4
line 4 = point 4, 1
```

Definition of lines is done from Modules->Geometry->Elementary entities->Add->Lines then select 2 points for each line by clicking directly on the GUI window. The image below is the appearance after line definition.

Then we will also define a circle which will be a hole later. A circle may be generated from Modules->Geometry->Elementary entities->Add->Circle

4.1.4 1-4: Definition of line groups and surface

At this moment, the geometry includes only some nodes and lines. By using them, we define surfaces where mesh will be generated and physical characteristics may be assigned.Definition of a surface may be done from clicking Modules->Geometry->Elementary entities->Add->plane Surface then chose lines which will be boundaries of this surface. At first, we make only the surface of the rectangular area without hole by selecting like:

then press `e` to finish the definition of one single surface. After this, the circle area is also converted to be another surface with the same way:

4.1.5 1-5: boolean operation

Now we have 2 plane surfaces, one is the rectangular area without hole and one circular area which will be a hole. In order to make rectangular geometry, we carry out a boolean difference from Modules->Geometry->Elementary entities->Boolean->Difference to cut the circle out from the rectangle. Please make it sure that the check box status for erasing the original objects is as:

then select the plane surface that will be the actual surface but not a hole part. `plane 1` in this case. Looks of the Gmsh window after this selection will be:

Press e to finish the selection of the first plane surface. Next selection is for the part which will be a hole. Please select the hole part `Plane 2` as:

then finish the boolean difference with tapping the key `e` then `q`. This boolean difference creates a new plane surface with a hole, and erased the original 2 plane surfaces i.e. rectangle and circle.

4.1.6 1-6: Definition boundary conditions and material flags

We need to define the physical names of outer boundary of the meshing region, which SPECFEM requires to recognize the ends of the simulation domain. We define the 4 physical names for each 4 bounds with exactly the same name as:

```
(tag name: explain)
Left   : left edge of the domain.
Bottom : lower edge of the domain.
Right  : right edge of the domain.
Top    : upper edge of the domain
```

This process can be done from Modules->Geometry->Physical Groups->Add->Curve as this image below:

by putting the name of the tags in the box and selecting the target boundary line, then pressing `e` to finish this process. Please define all 4 defintions with the same way.

You can verify the defined names of physical curves on the GUI:

Next, we will define the physical tags for the plane surfaces with the rule of:

```
M1 : for the first material
M2 : for second ...
...
Mn : for n-th material
```

Please pay an attention that the number of material need to be matched with the material id when you configured in WASI user interface.

Definition of a physical surface may be done from Modules->Geometry->Physical Groups->Add->Surface like:

4.1.7 1-7: Meshing

Before generating a mesh, we need to setup 2 things,

```
1. define the element size.
2. specify the type of mesh as quad
```

The element sizes may be defined from Modules->Mesh->Size at Points

In the image, the element size at all nodes are defined to be equal the number of the parameter `lc` which we defined at the first step of this example.

In the next step, we specify the finite element type which Gmsh generates. Because SPECFEM supports only Quad elements, we need to change the setting of Gmsh as so. This configuration may be modified from the top menu bar Tools->Options->Mesh->General then set the parameters as the image below:

Then user may generate the all-Quad mesh from Modules->Mesh->2D The result of meshing is indicated as:

In the case you find partially small elements, you may try to smooth them from Modules->Mesh->Smooth 2D.

As we prepared 2 parameters i.e. the radius of the hole and mesh element size, they are easily modified as:

In this image, the radius is set to larger and mesh size is changed to be smaller.

After finishing the generation of the mesh, mesh file may be saved from Modules->Mesh->Save then Gmsh creates a `.msh` file. In order to use this mesh file, it is necessary to move this file into the simulation directory (i.e. where your jupyter notebook, for simulation configuration, is placed.), and verify the file name before .msh part is exactly the same with the `file name` which you put in the simulation configuration step.

4.1.8 1-8: The way to mesh a large model (without GUI)

Depending on the amount of RAM and graphics, sometimes meshing of a model with the great number of meshes is not possible to be visualized by GUI of Gmsh. In that case, we can use a terminal command of Gmsh.

Before using gmsh cui command, we need to add only 2 lines at the end of the `.geo` file. The lines to be added is here:

```
Mesh.SubdivisionAlgorithm=1;
Recombine Surface "*";
```

then we open the terminal on Jupyter lab, then run the command below:

```
gmsh -2 yourgeofile.geo -o yourmeshfile.msh
```

This command will generate the mesh file.

4.2 Example 2: heterogeneous medium (e.g. hole filled with oil)

In this example, we prepare a mesh for a heterogeneous model i.e. a simulation domain composed with multiple material. We will use the same shape of geometry thus the step the explanation for this example starts from just after the section 1-4 Definition of line groups and surface.

4.2.1 2-5: Boolean operation

In stead of just cutting off the circle area, this example fills in this circle with another material. Instead of boolean difference, we use here boolean fragment for keeping the plane surface inside of the circle.

The initial preparation process is same with Example 1 thus please follow the instruction from 1-1 to 1-4.

4.2.2 2-6: Definition boundary conditions and material flags

In order to keep the surface of the circle, we use the boolean fragments, from Modules->Geometry->Elementary entities->Boolean->Fragments. Firstly please select the rectangle and press `e`, then select the circle and `e`. After selecting the circle, the appearance of Gmsh window is like :

4.2.3 2-7: Meshing

The setting of mesh element size may be done with exactly the same way with 1-7 thus you may go back to Example 1 then finish meshing. Please pay attention that you need to define the 2 physical surface, namely not only `M1` for the rectangle but also `M2` for the circle.

The generated mesh will be like this image:

4.3 Example 3: 3D mesh

In this example, we will make a 3D mesh model by extruding the 2D surface which we have made in Example 1.Thus this example starts after the section 1-5 (Boolean opperation).However for 3D geometry, boundary condition flags (i.e. top, bottom, left, right) are not necessary.Thus only material flags (M1, M2) are necessary.

4.3.1 3-1: Extrusion

For extrusion, select Geometry->Extrude->Transpose.Then select a plane which will be extruded.On the newly opened window, configure the amount of extrusion for x,y,z direction and select the number of layer (elements). At this point, we can again use the self-defined parameters e.g. `lc` for the target element size.

The image below is the result of this step.

4.3.2 3-2: Material flags

We will do the same process with the section 1-6.However for 3D geometry, boundary condition flags (i.e. top, bottom, left, right) are not necessary.Thus only material flags (M1, M2) are necessary.

4.3.3 3-3: Meshing

Finally by selecting Mesh->3D, meshing will be finished as,

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search