

Quick examples of generating 1D Greens function by using FK

Liang Ding, Qinya Liu, Gang Yang

Contents

- Install and use the MTUQ docker image
- Running FK examples
- Use your own velocity model and data

Install and use the MTUQ docker image

- Download the MTUQ image

[docker pull ghcr.io/sei/sscoped/mtuq:main](https://github.com/sei/sscoped/mtuq:main)

```
dingl@olivine:~$ docker pull ghcr.io/sei/sscoped/mtuq:main
main: Pulling from sei/sscoped/mtuq
e0b25ef51634: Already exists
fd0442249241: Pull complete
69d2d869ecd7: Pull complete
4f4fb700ef54: Pull complete
df63c0d8e7d8: Pull complete
36ed403724f5: Pull complete
e98483073e02: Pull complete
f09cd92522a8: Pull complete
6b80f2796c4b: Pull complete
52d565985003: Pull complete
adef19292676: Pull complete
af2df34c81cc: Pull complete
239187b792fb: Pull complete
769884be511d: Pull complete
93ea4ccf9e96: Pull complete
1ebcbd376123: Pull complete
6bf006f9bb5e: Pull complete
efd2eda9455e: Pull complete
4b1dbdee6a77: Pull complete
85862795f89b: Pull complete
287d98057869: Pull complete
Digest: sha256:316474d035494918713330cc74c80913a20df73b53826748824cb2b3db8248f6
Status: Downloaded newer image for ghcr.io/sei/sscoped/mtuq:main
ghcr.io/sei/sscoped/mtuq:main
dingl@olivine:~$
```

- Tip: Check docker images

[docker images](#)

```
dingl@olivine:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ghcr.io/sei/sscoped/mtuq	main	8b9b5faa426d	12 hours ago	3.56GB

Install and use the MTUQ docker image

- Run the MTUQ image

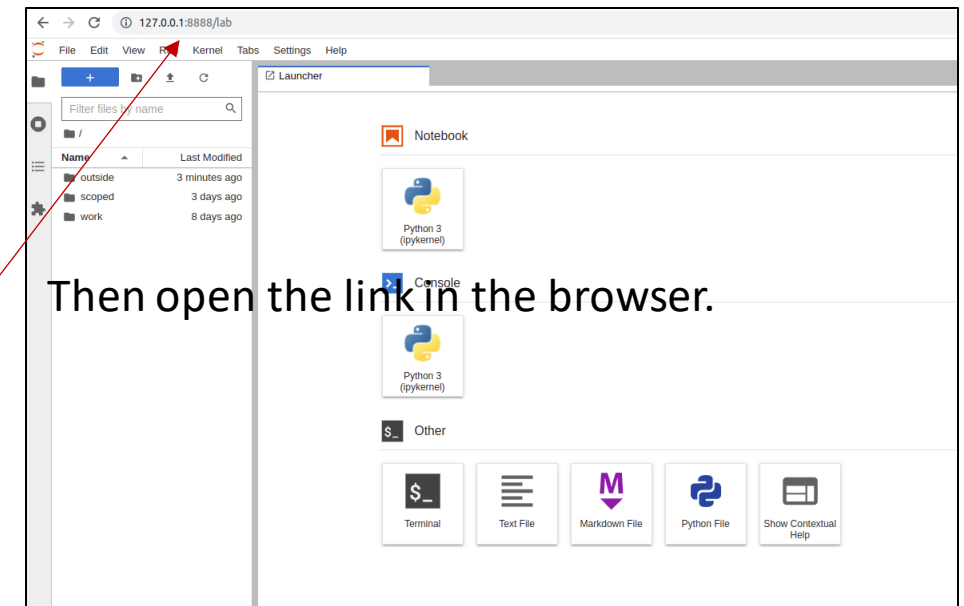
[docker run -p 8888:8888 ghcr.io/seisscoped/mtuq:main](#)

Or binding with a local folder (eg. current directory)

[docker run --mount src=`pwd`,target=/home/jovyan/outside,type=bind -p 8888:8888 ghcr.io/seisscoped/mtuq:main](#)

```
dtngl@oltvine:~/Docker_Workspace/MTUQ_docker$ docker run --mount src=`pwd`,target=/home/jovyan/outside,type=bind -p 8888:8888 ghcr.io/seisscoped/mtuq:main
Entered start.sh with args: jupyter lab
Executing the command: jupyter lab
[I 2022-04-18 18:47:14.814 ServerApp] jupyterlab | extension was successfully linked.
[W 2022-04-18 18:47:14.840 NotebookApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2022-04-18 18:47:14.840 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2022-04-18 18:47:14.840 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2022-04-18 18:47:14.848 ServerApp] nbclassic | extension was successfully linked.
[I 2022-04-18 18:47:14.851 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2022-04-18 18:47:15.867 ServerApp] notebook_shim | extension was successfully linked.
[I 2022-04-18 18:47:16.002 ServerApp] notebook_shim | extension was successfully loaded.
[I 2022-04-18 18:47:16.003 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.9/site-packages/jupyterlab
[I 2022-04-18 18:47:16.003 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 2022-04-18 18:47:16.008 ServerApp] jupyterlab | extension was successfully loaded.
[I 2022-04-18 18:47:16.022 ServerApp] nbclassic | extension was successfully loaded.
[I 2022-04-18 18:47:16.023 ServerApp] Serving notebooks from local directory: /home/jovyan
[I 2022-04-18 18:47:16.023 ServerApp] Jupyter Server 1.16.0 is running at:
[I 2022-04-18 18:47:16.023 ServerApp] http://04948765bed7:8888/lab?token=c440b0120e5befe09fb001877a395962c8d67bb385ccd337
[I 2022-04-18 18:47:16.023 ServerApp] or http://127.0.0.1:8888/lab?token=c440b0120e5befe09fb001877a395962c8d67bb385ccd337
[I 2022-04-18 18:47:16.023 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2022-04-18 18:47:16.031 ServerApp]

To access the server, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/jpserver-7-open.html
Or copy and paste one of these URLs:
http://04948765bed7:8888/lab?token=c440b0120e5befe09fb001877a395962c8d67bb385ccd337
or http://127.0.0.1:8888/lab?token=c440b0120e5befe09fb001877a395962c8d67bb385ccd337
```

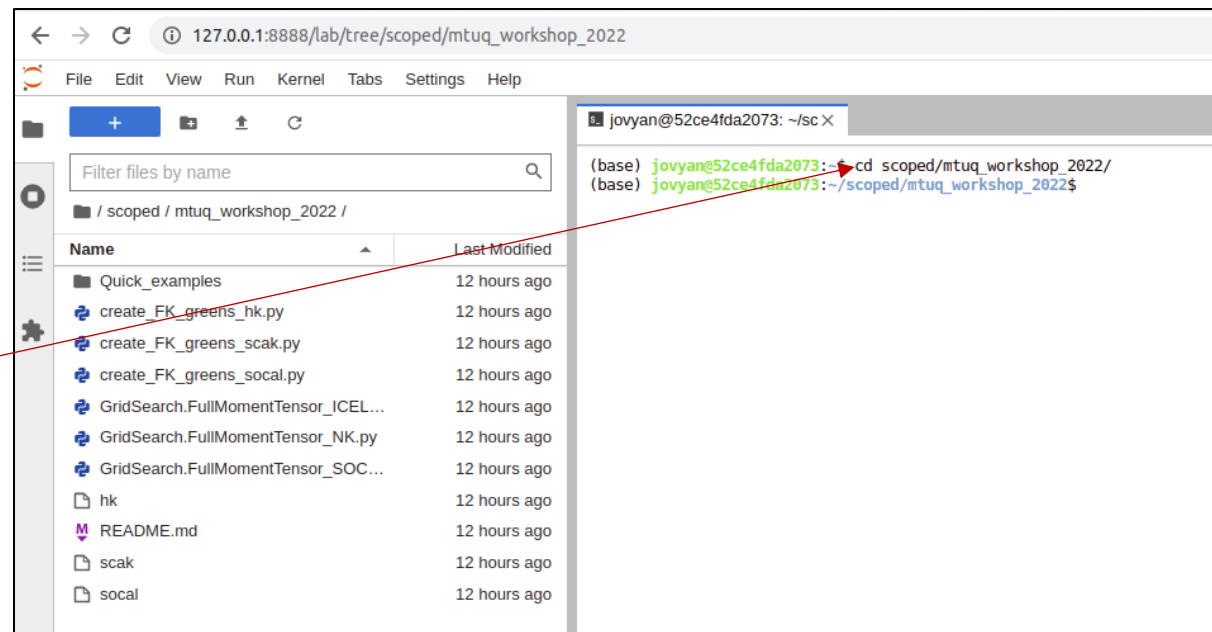
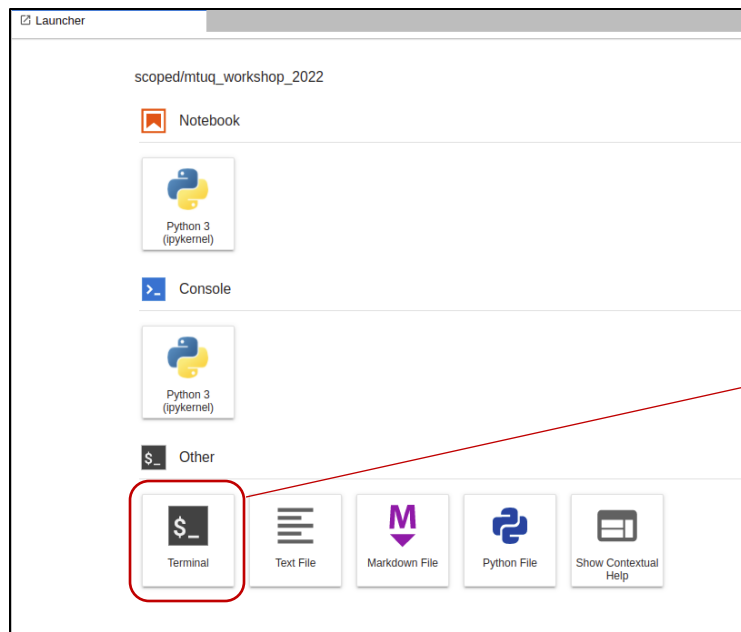


Running FK examples

Attention: Examples should be run in the MTUQ docker image

- Prepare the working space and python scripts

open a terminal and go to the folder `scoped/mtuq_workshop_2022/` by:
`cd scoped/mtuq_workshop_2022/`



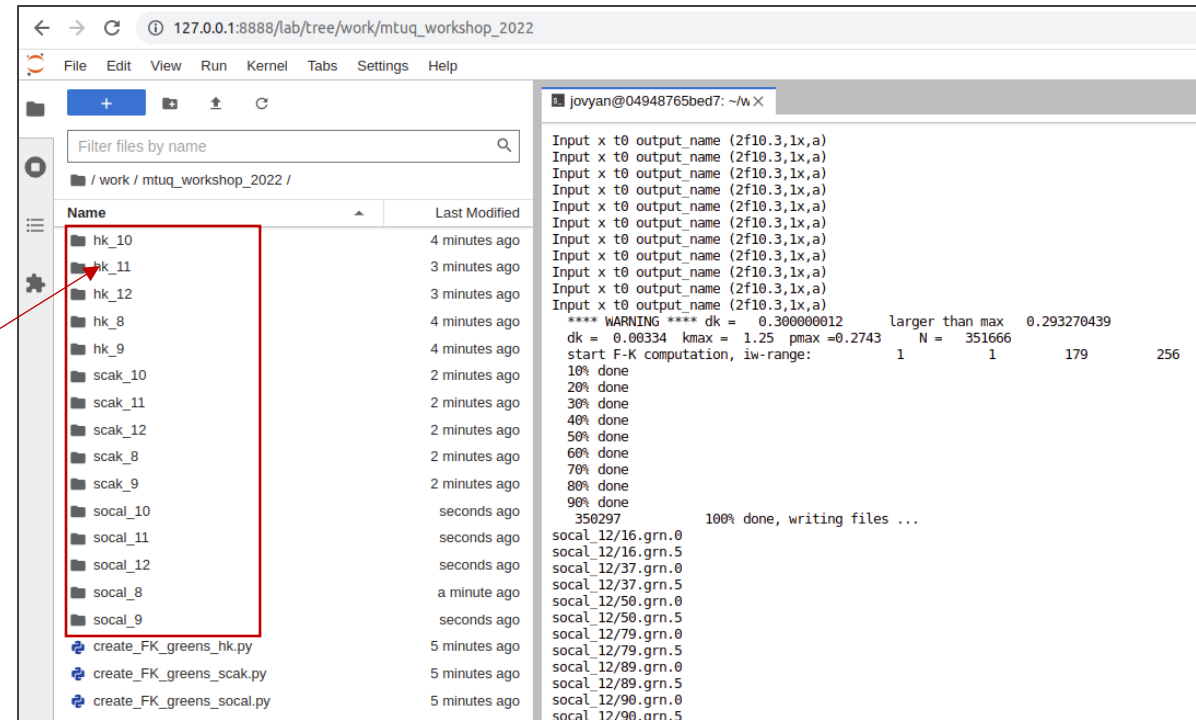
Running FK examples

Attention: Examples should be run
in the MTUQ docker image

- Create Greens function

```
python create_FK_greens_hk.py  
python create_FK_greens_scak.py  
python create_FK_greens_socal.py
```

Then, you will see the newly
created folders storing the GF.
Folder name: modelName_depth(km)



Use your own velocity model and data

- Velocity model (in units of km, km/s, g/cm³)

Model type	model_type = 'f'					
Format	thickness	vs	vp	rho (default: 0.77 + 0.32*vp)	Qs (default: 500)	Qp (default: 2*Qs)

Model type	model_type = 'k'					
Format	thickness	vs	vp/vs	rho (default: 0.77 + 0.32*vp)	Qs (default: 500)	Qp (default: 2*Qs)

- In python script (lines 23-24):

model_name = 'filename of your model'
model_type = 'your model type'

```
22 tk_command = 'tk.plt'
23 model_name = 'hk'
24 model_type = 'k'
25 npts = 512 # must
```

Use your own velocity model and data

- Use your data

Update the following parameters:

path_data = "path to your data"

path_weights = "path to the weight file"

event_id = "your event id"

Searching_depths = "the depth of source generating GF"

```
1 #!/usr/bin/env python
2
3 import os
4 import numpy as np
5 from mtuq import read
6 from mtuq.util import fullpath
7 from mtuq.util.cap import parse_station_codes
8
9
10 def create_FK_greens():
11     '''Create Greens' function associated with data by using FK. '''
12
13     path_data = fullpath('data/examples/20090407201255351/*.zrt')
14     path_weights = fullpath('data/examples/20090407201255351/weights.dat')
15     event_id = '20090407201255351'
16
17     # user specified searching depth in km
18     # searching_depths = np.array([5, 11, 18]) # eg: at 5, 11, 18 km.
19     searching_depths = np.arange(8, 13, 1) # eg: from 8 to 13 km with interval of 1 km.
20
```

- Other paras:

You may change the following parameters:

npts = "number of points of the GF" (must be 2ⁿ)

dt = "time interval of the GF"

```
25 npts = 512 # must be 2^n
26 dt = 0.1
27 # ...
```