FUNWAVE-TVD WORKSHOP TUTORIAL

July 25-27, 2018
Embassy Suites, Newark DE

Prerequisites for the FUNWAVE Workshop

- Your laptop is able to use Wi-Fi to access to the internet
- You can use SSH Secure Shell Client (like: Putty or Cygwin on Windows)
- If your laptop uses Windows OS, it is recommended that you download the latest version of cygwin (http://www.cygwin.com). Cygwin is a bash shell/unix emulation program and contains many of the tools such as tar, gzip/gunzip, and cpp, which will be useful for installation and compilation of FUNWAVE-TVD.
- Having a postprocessing toolbox on your machine (laptop). Both Matlab and Python post-processing script are provided with practice examples. Many participants already use Matlab, but if you do not, it is recommended that you install a Python package (information below).

Downloading and Installing a Python Package (Anaconda)

The best and most comprehensive FREE package for the Python language, along with most tools and modules (e.g., NumPy, Matplotlib, etc.) is distributed by the Continuum Analytics under the Anaconda package. It is available for Linux, Mac OS X, and Windows machines. You do **NOT** need administrator privileges to install the Anaconda package, you can do so as a standard user on all three platforms listed above.

For the Anaconda package go to: https://www.continuum.io/downloads

Pick the **appropriate platform** (Linux, Mac OS X, Windows) by clicking the correct tab and get the Anaconda distribution that comes with **Python 3.6** (not 2.7). You can either download the **Graphical Installer** (recommended), of if you are comfortable with the terminal in the Linux/Mac OS X environment, you can also download it through the command line.

For the complete list of packages/modules included in the Anaconda Python distribution see: https://docs.continuum.io/anaconda/packages/pkg-docs

Backup Plan (if UDel HPC has issues): Use Amazon AWS Cloud Computing

- Amazon AWS EC2 provides 12-month free tier, 760 hours/month
- If you don't have a HPC cluster, you can build your cluster in EC2
- A simple example can be found in **Appendix** in this tutorial

Training Session # 1 (Wednesday, lead: **Matt Malej)**

Topics

- FUNWAVE-TVD and Parallel Computing (MPI) - Documentation Wiki

  Wiki ⇒ https://fengyanshi.github.io/build/html/index.html

- Where do I get the code? - Version Control (Github)

  Full Repository ⇒ https://github.com/fengyanshi/FUNWAVE-TVD

  Latest Code Release (July 2018) is version 3.3 ⇒ https://github.com/fengyanshi/FUNWAVE-TVD/releases/tag/Version_3.3

- Sandbox for USACE and DoD members with GUI **[internal -- private on US Army side]**

  Link (need access) ⇒ https://funwave.erdc.dren.mil

- USACE/DoD HPC Portal Deployment of FUNWAVE **[internal to DoD with ACL]**

- How to build (compile/link) and install FUNWAVE-TVD on different machines for parallel computation?

  https://fengyanshi.github.io/build/html/setup.html#compile-and-setup

Practice

1) Log into ==> ***mills***

```
> ssh your_user_id@mills.hpc.udel.edu
   ... input your password
> mkdir your_funwave_folder    (optional, otherwise all will be in $HOME)
```

NOTE: $HOME will be /home/funwave_tvd_workshop/tvdguest##

We suggest using **scp** on Mac/Linux **(or FileZilla for USACE on ACE-IT laptop)** to transfer data to/from ***mills***

```
> sftp your_user_id@mills.hpc.udel.edu
```

or

```
> scp your_user_id@mills.hpc.udel.edu:/home/your_user_id/test.txt  ./
  input your password
```

2) **Clone** FUNWAVE-TVD Package

> **cd your_funwave_folder** (*if you created in step 1*)
> **git clone https://github.com/fengyanshi/FUNWAVE-TVD.git**

3) Compile the source code (DIFFERENT EXECUTABLES)

```
> cd FUNWAVE-TVD/src
> emacs Makefile-Mills    (if you want to modify the source file)
```

NOTE: Modify the Makefile if needed. Emacs/vi/gedit are text editors. You can use any other editor you are familiar with on the Unix/Linux system. To exit emacs editor press (Ctrl X then Ctrl C).

EXEC = funwave_surface_wave (for example)
FLAG_1 = -DDOUBLE_PRECISION
FLAG_3 = -DCARTESIAN
FLAG_2 = -DPARALLEL *(if you want to run with parallel mode)*
FC = mpif90 *(on **mills and amazon clooud**)*

*The compiled/linked executable file will be **funwave_surface_wave** inside the **src** directory*

- Compile the code for surface wave applications

```
> vpkg_require openmpi  (loads MPI libraries for parallel execution)
> make clean (or 'make clobber' if you want to remove the executable)
> make -f Makefile-Mills
```

- Compile the code for ship-wake applications **[different executable]**

  Besides FLAGs used for surface wave applications, add/change

FLAG_8 = -DVESSEL
EXEC = funwave_ship_wake

```
> vpkg_require openmpi  (loads MPI libraries)
> make clean (or 'make clobber' if you want to remove the executable)
> make -f Makefile-Mills-Vessel
```

- Compile the code for sediment transport **[different executable]**

  FLAG_11 = -DSEDIMENT
  EXEC = funwave_wave_sed (for example)

  ```
  > vpkg_require openmpi  (loads MPI libraries for parallel simulations)
  > make clean
  > make -f Makefile-Mills-Sediment
  ```

- Compile the code for spherical coordinates (e.g., tsunami simulations)

  Use FLAGs for surface wave applications, but **remove**

  FLAG_3 = -DCARTESIAN

  ```
  > vpkg_require openmpi  (loads MPI libraries for parallel simulations)
  > make clean
  > make -f Makefile-Mills-Spherical
  ```

Training Session # 2 (Wednesday, lead: **Matt Malej)**

Topics

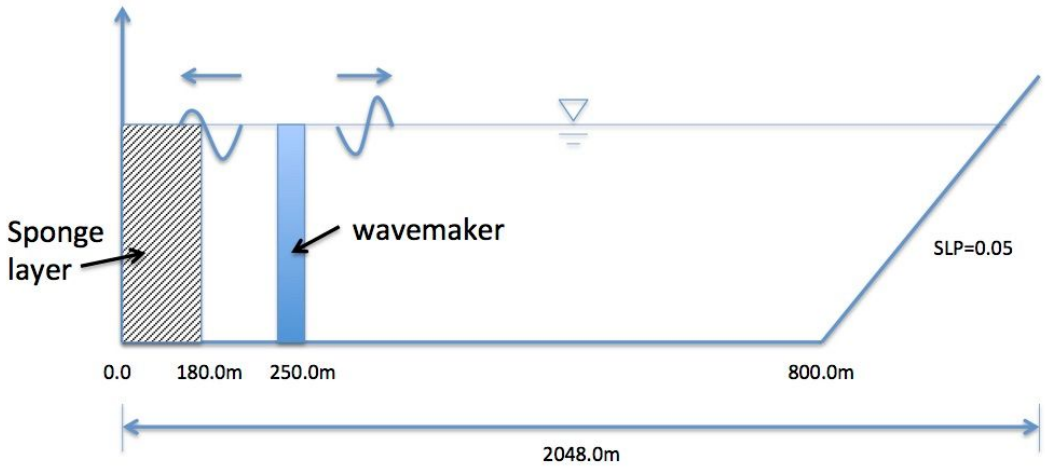
- How to run FUNWAVE-TVD? Navigating the basic sections within the INPUT file for different simulations (numerics, physics, input, output, etc.).

  Wiki Direct Link ⇒ https://fengyanshi.github.io/build/html/definition.html

- Setting up (Linux/Mac OS X and HPC machines with PBS scheduler), running, and post-processing your first FUNWAVE-TVD simulation (1D beach runup or levee overtopping with shoaling and wetting/drying).

Practice

1) Surface Waves on 1D Sloped Beach



Grid Dimensions: *1024X3. Grid sizes: DX=DY=2m. Depth at flat bottom: 10 m, Beach slope: 1/20.*

- Go to the directory of the 1D case

```
> cd FUNWAVE-TVD/simple_cases/surface_wave_1d
```

- Create a **work** directory

```
> mkdir work
```

- Copy and rename one of input the files into a work folder (regular wave as an example)

```
> cd work
> cp .. / input_files/input_reg.txt  input.txt
```

- Copy compiled executable into the **work** directory

```
> cp ../../../src/funwave_mills   ./
```

- Check and Modify **input.txt**

The following statements are necessary in the **input.txt**

Parallel (if applicable)
PX = 4
PY = 1

Depth
DEPTH_TYPE = SLOPE
DEPTH_FLAT = 10.0
SLP = 0.05
Xslp = 800.0

Dimensions
Mglob = 1024
Nglob = 3

Time
TOTAL_TIME = 200.0
PLOT_INTV = 10.0
SCREEN_INTV = 10.0

Grid sizes
DX = 1.0
DY = 1.0

Add wavemaker
WAVEMAKER = WK_REG
DEP_WK = 10.0
Xc_WK = 250.0
Yc_WK = 0.0
Tperiod = 8.0
AMP_WK = 0.5
Delta_WK = 3.0 ! the default is 0.5, set a larger number for long waves

Add sponge layer
FRICTION_SPONGE = T
DIRECT_SPONGE = T
Sponge_west_width = 180.0
Sponge_east_width = 0.0
Sponge_south_width = 0.0
Sponge_north_width = 0.0

Breaking scheme (default: SWE breaker)
VISCOSITY_BREAKING = T
Cbrk1 = 0.65
Cbrk2 = 0.35

Wetting and Drying
MinDepth=0.01

Output

RESULT_FOLDER = output/
ETA = T
MASK = T

- **Run the Model**

```
> cp ~/FUNWAVE-TVD/simple_cases/YourNamePBS.qs ./run_script.qs
```

open the `run_script.qs` and examine/change desired content

```
> qsub run_script.qs
```

NOTE: make sure run_script.qs is in the current work directory. Computational time: 100 sec across a total of 4 cores.

- **Post-Process Your Results**

Download your results to your laptop **(eta_#####, mask_#####, dep.out)** and use the provided Matlab or Python scripts (using either **sftp** or **scp**).

```
> cd directory_where_you_want_to_download_your_results
> scp -r
your_user_id@mills.hpc.udel.edu:/home/your_user_id/FUNWAVE-TVD/simple
_cases/surface_wave_1d/work  ./

...enter your password
```

NOTE: You will need to modify the output folder name in Matlab/Python scripts to load your results.

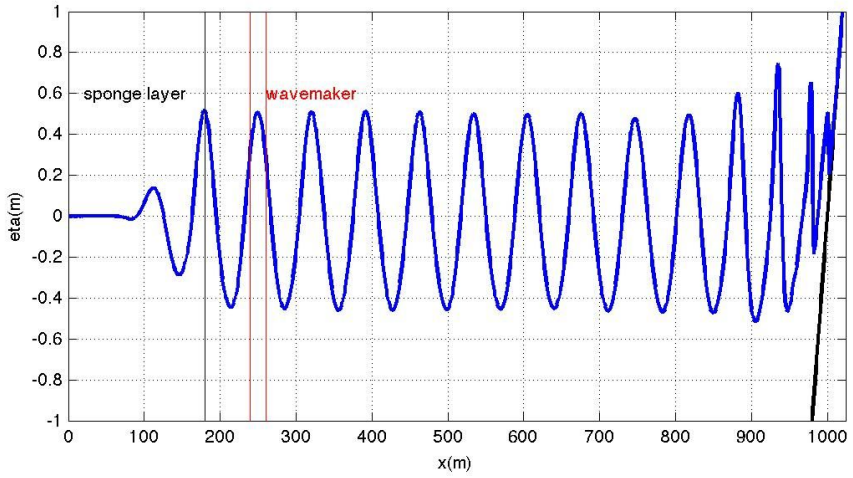
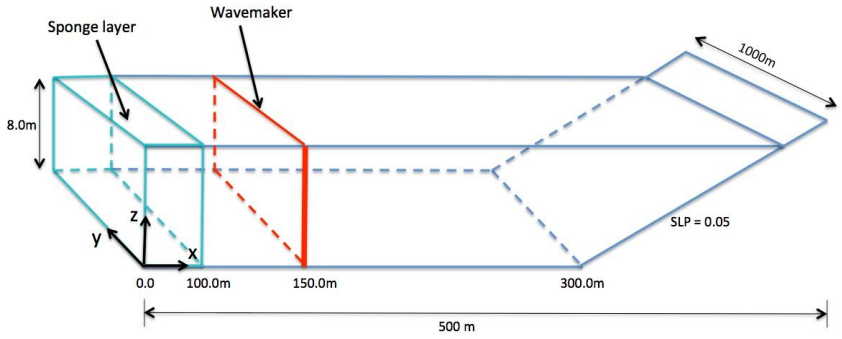


Figure: an example plot using Matlab **plot_wave.m**

2) Waves on 2D plane beach



Model Configuration: *Grid dimensions: 250X500. Grid sizes: DX=DY=2m. Depth at flat bottom: 8 m, Beach slope: 1/20.*

- Go to directory **FUNWAVE-TVD/simple_cases/beach_2d**
- Create a work directory

```
> mkdir work
```

- Copy and rename one of input files into the work directory (regular wave as an example)

```
> cd work
> cp ../input_files/input_reg.txt  input.txt
```

- Check and Modify **input.txt**

  The following statements are necessary in input.txt

Slope bed
DEPTH_TYPE = SLOPE
DEPTH_FLAT = 8.0
SLP = 0.05
Xslp = 300.0

Output folder
RESULT_FOLDER = output/

Dimensions
Mglob = 250
Nglob = 500

Grid sizes
DX = 2.0
DY = 2.0

Add wavemaker
WAVEMAKER = WK_REG
DEP_WK = 8.0
Xc_WK = 150.0
Yc_WK = 0.0
Tperiod = 8.0
AMP_WK = 0.5
Theta_WK = 30.0
Delta_WK = 3.0

Add periodic boundary condition
PERIODIC = T

Sponge layer
DIFFUSION_SPONGE = F
FRICTION_SPONGE = T
DIRECT_SPONGE = T
Csp = 0.0
CDsponge = 1.0
Sponge_west_width = 100.0
Sponge_east_width = 0.0
Sponge_south_width = 0.0
Sponge_north_width = 0.0

Wave breaking
VISCOSITY_BREAKING = T
Cbrk1 = 0.65
Cbrk2 = 0.35

Wave average property
T_INTV_mean = 100.0
STEADY_TIME=100.0

Output
ETA = T
Umean = T
Vmean = T
ETAmean = T
MASK = T
WaveHeight = T

NOTE: the required model computational time will be set to TOTAL_TIME=950 sec, and ran on 8 cores

- Use the same procedures as in the previous sessions to run the model and post-process results

- Try different **wavemakers**, **sponge layers**, **wave breaking scheme**, **non periodic boundary condition**, etc.

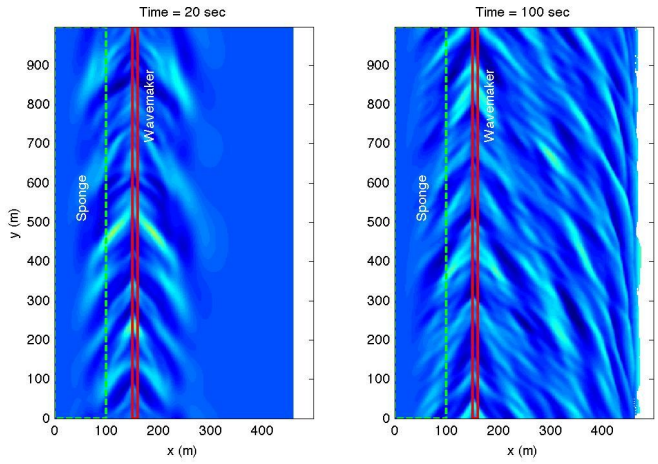
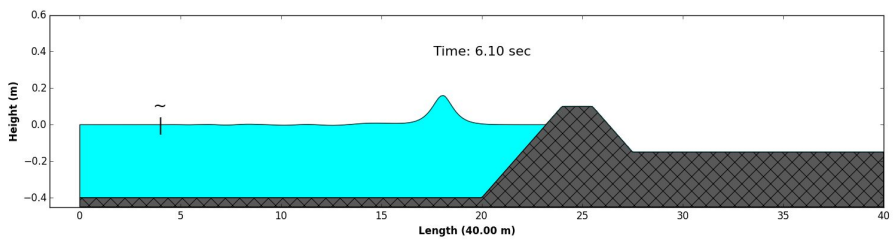


***Figure**: an example of plot using **plot_wave.m**, irregular obliquely incident waves.*

3) EXTRA CHALLENGE - PROGRESSION

For those participants who were able to to complete the above task and want an additional challenge, try to set up the following (in `simple_cases/levee_1d`)

a) Modify the **input.txt** to force the wavemakers with a **Solitary Wave** as input (**Hint:** WAVEMAKER=INIT_SOL), of **0.16 meter** amplitude.
b) Use the supplied bathymetry (**depth_levee.txt**) as the **'DATA'** depth type with dimensions of [500 x 3] points.
c) The depth at the wavemaker is **0.4** meters and it is located **4.0** meters from the left boundary.
d) Set the **TOTAL_TIME** to **30.0** seconds with a **PLOT_INTERVAL** of **0.1** seconds.
e) Activate the sponges (**DIRECT_SPONGE** only) on the **west** (2.0m) and **east** (1.0m) side.
f) Set the spatial discretization to **dx=dy=0.08** meters, wetting/drying to 1 millimeter (**MinDepth=0.001**), **FroudeCap=2.0**, and **CFL** condition to **0.1**.
g) Output the **depth**, surface elevation (**eta**), and **mask**.



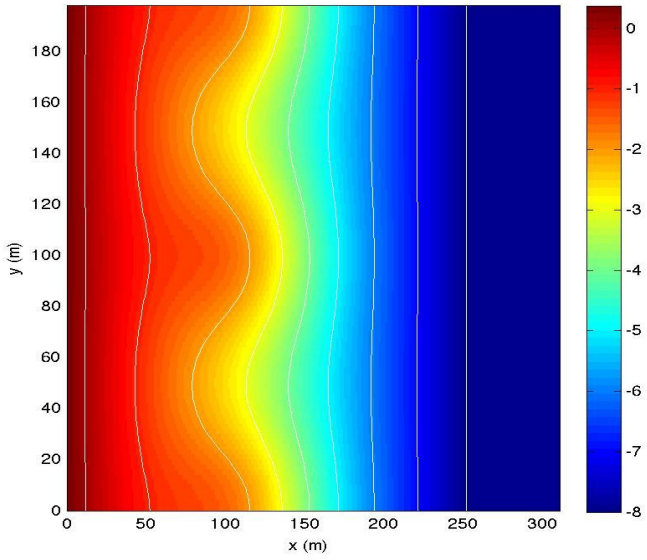
Training Session # 3 (Thursday, lead: **Fengyan Shi**)

Topics

1) Waves and wave-induced circulation
2) Sediment transport

Practice

1) Rip channels



Model Configuration: *The figure shows bathymetry (/simple_cases/sediment_rip/bathy/depth_a15.txt). Waves generated at x = 280 m on flat bottom: 8 m. Grid dimensions: 312X100. Grid sizes: DX=1.0m, DY=2.0m. Depth file is in* `/simple_cases/sediment_rip/bathy/depth_z15.txt`.

- Compile the code with -DSEDIMENT flag on
- Go to directory **FUNWAVE-TVD/simple_cases/sediment_rip**

```
> cd work
> cp ../../../src/funwave_wave_sed ./
```

- Check and modify **input.txt**

  The following statements are necessary in **input.txt**

```
Bathymetry
  DEPTH_TYPE = DATA
  DEPTH_FILE= ../bathy/depth_a15.txt
```

Output folder
RESULT_FOLDER = output/

Dimensions
Mglob = 312
Nglob = 100

Grid sizes
DX = 1.0
DY = 2.0

Time
TOTAL_TIME = 1000
PLOT_INTV = 100.0
PLOT_INTV_STATION = 0.5
SCREEN_INTV = 100.0

Add wavemaker
WAVEMAKER = WK_REG
DEP_WK = 8.0
Xc_WK = 280.0
Tperiod = 8.0
AMP_WK = 0.5
Theta_WK = 0.0
Delta_WK = 3.0

Add periodic boundary condition
PERIODIC = T

Sponge layer
DIFFUSION_SPONGE = F
FRICTION_SPONGE = T
DIRECT_SPONGE = T
Csp = 0.0
CDsponge = 1.0
Sponge_west_width = 0.0
Sponge_east_width = 60.0
Sponge_south_width = 0.0
Sponge_north_width = 0.0

Wetting and drying
MinDepth = 0.01

Wave breaking
VISCOSITY_BREAKING = F

Wave average property
T_INTV_mean = 50.0
STEADY_TIME=100.0

Sediment parameters
Bed_Change = T
BedLoad = T

D50 = 0.0005
Sdensity = 2.68
n_porosity = 0.47
WS = 0.0125
Shields_cr = 0.055
Shields_cr_bedload = 0.047
Tan_phi = 0.7
Kappa1 = 0.3333
Kappa2 = 1.0
MinDepthPickup = 0.1

Output

ETA = T
Umean = T
Vmean = T
ETAmean = T
MASK = T
WaveHeight = T

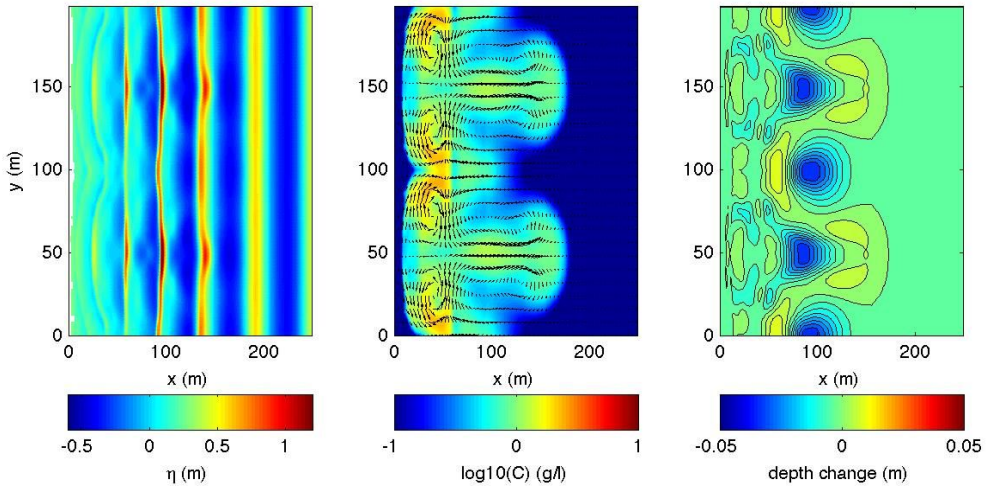


Figure: *Case of sediment transport in rip channels. (left) surface elevation, (middle) sediment concentration, (right) morphological change*

2) EXTRA CHALLENGE - PROGRESSION:

- analysis of wave-averaged properties such as the significant wave height (**Hsig**) and wave-induced currents.
- rip current case in /simple_cases/rip_2d/

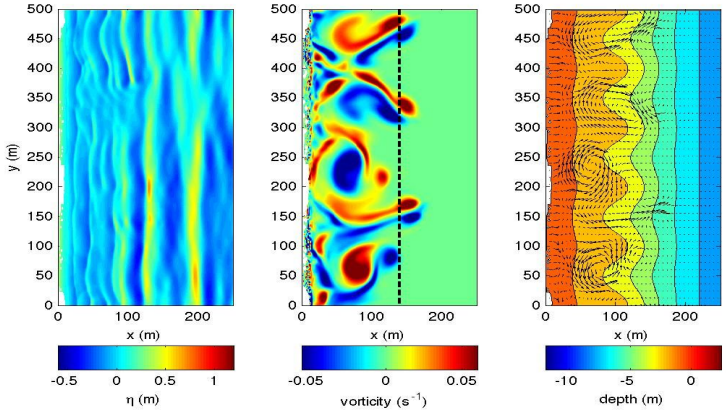


Figure: *rip current case. (Left) surface elevation, (middle) vorticity field, (right) nearshore circulation.*

Training Session # 4 (Thursday, lead: **Fengyan Shi)**

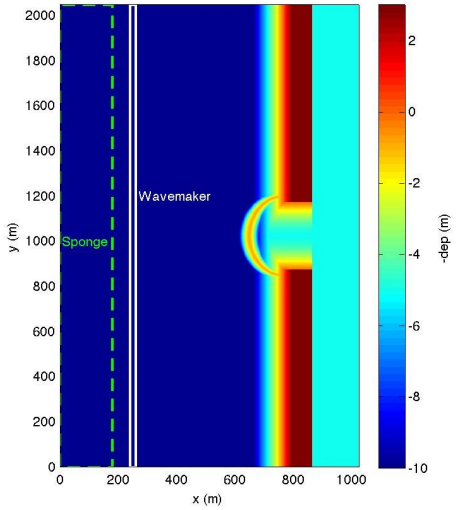
Topics

- Wave simulation on 2D random bathymetry with complex shoreline geometries

    1) Inlet Problem (to include post-processing analysis of Harbor Resonance)
    2) Obstacles and Breakwaters (partially absorbing and reflecting inner boundaries)

Practice

1) Inlet Problem



Model Configuration: *Grid dimensions: 512X1024. Grid sizes: DX=DY=2m. The bathymetry / topography include flat bottom, barrier beaches, shoal, inlet and shallow basin.*

- Go to **FUNWAVE-TVD/simple_cases/inlet_shoal**
- Create a work directory

```
> mkdir work
```

- Copy and rename one of input files into the work directory (regular wave as an example)

```
> cd work
> cp ../input_files/input_reg.txt   input.txt
```

- Check and Modify **input.txt**

The following statements are necessary in input.txt

```
Parallel (if applicable) … running here on 16 CPU's (or cores/ranks)
        PX = 4
        PY = 4
Depth
        DEPTH_TYPE = DATA
        DEPTH_FILE = ../bathy/dep_shoal_inlet.txt
Output folder
        RESULT_FOLDER = output/
Dimensions
                Mglob = 512
                Nglob = 1024
Time
                TOTAL_TIME = 1200.0
                PLOT_INTV = 30.0
                PLOT_INTV_STATION = 0.5
                SCREEN_INTV = 30.0
Grid sizes
                DX = 2.0
                DY = 2.0
Wavemaker
                WAVEMAKER = WK_REG
                DEP_WK = 10.0
                Xc_WK = 250.0
                Yc_WK = 0.0
                Tperiod = 12.0
                AMP_WK = 1.0
                Theta_WK = 0.0
Sponge layer
                FRICTION_SPONGE = T
                DIRECT_SPONGE = T
                Csp = 0.0
                CDsponge = 1.0
                Sponge_west_width = 180.0
                Sponge_east_width = 0.0
                Sponge_south_width = 0.0
                Sponge_north_width = 0.0
Add periodic boundary condition:
                PERIODIC = T
Wetting and drying
                MinDepth=0.01
Breaking scheme
                VISCOSITY_BREAKING = T
                Cbrk1 = 0.65
                Cbrk2 = 0.35
Wave averaging property
                T_INTV_mean = 240.0
                STEADY_TIME=480.0
Output
                ETA = T
                MASK = T
                WaveHeight = T
```

Use the same procedures as in the previous sessions to run the model and post-process results

EXTRA CHALLENGE - PROGRESSION:

Try different options, for example, with **periodic boundary condition for obliquely incident waves**, **different wavemaker and wave parameters**.

2) Obstacles and Breakwaters (partially absorbing and reflecting inner boundaries)

For obstacles: Add an OBSTACLE FILE

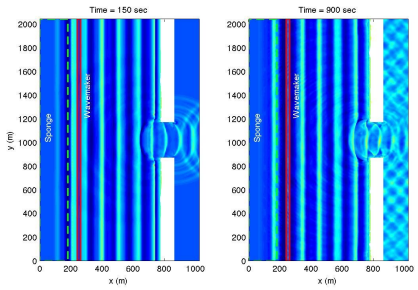
```
OBSTACLE_FILE = ../bathy/obs_shoal_inlet.txt
```

For breakwaters with full reflection: Add a breakwater structure in the bathymetry file

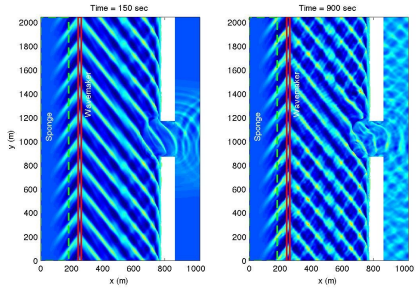
```
DEPTH_FILE = ../bathy/dep_shoal_inlet_brk.txt
```

For breakwaters with partially absorbing and reflecting inner boundary conditions: In addition to the bathymetry with the breakwater structure, **add BREAKWATER FILE for partial reflection**

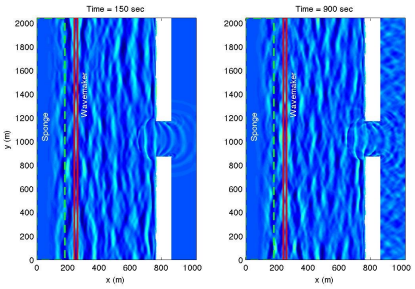
```
BREAKWATER_FILE = brk_shoal_inlet.txt
```



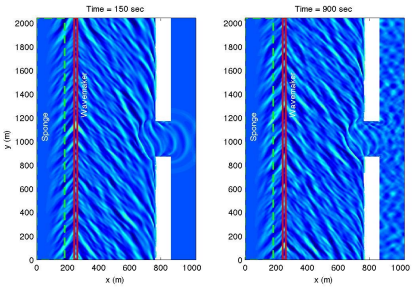
Case: *monochromatic, normal incidence*



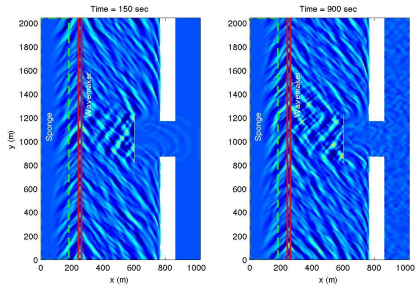
Case: *monochromatic, oblique incidence*



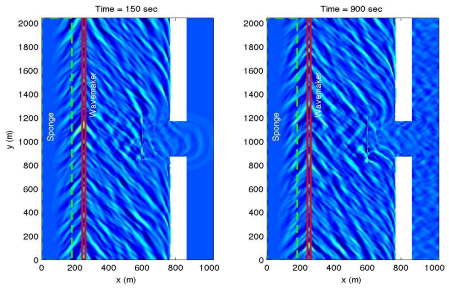
Case: *irregular, normal incidence*



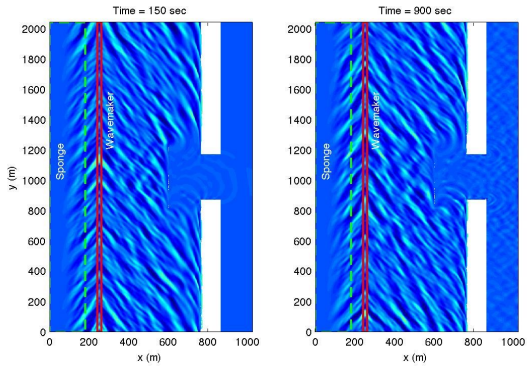
Case: *irregular, oblique incidence*



Case: *breakwater using obstacle option*



Case: *breakwater by modifying depth file*



Case: *partially reflecting/absorbing breakwater*

EXTRA CHALLENGE - PROGRESSION:

1) Analysis of harbor resonance.
2) Set up your own surface wave case

Training Session # 5 (Friday, lead: **Matt Malej)**

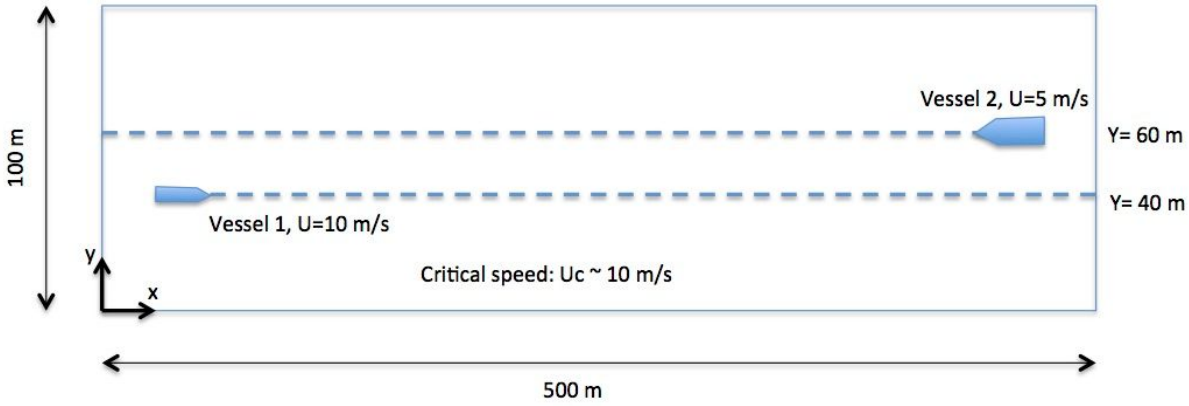
Topics

- Ship-wakes

    1) Setup with multiple vessels (**paths, size, velocities**)
    2) Ship wake-induced sediment transport
    3) A vessel moving on random bathymetry

Practice

1) Setup with Multiple Vessels



Model Configuration: *Grid dimensions 500x100. Grid sizes DX=DY=1.0m. Vessel 1, length=10m, width=5m, draft=2.0m, speed=10m/s, starts from x=50m. Vessel 2, length=20m, width=8m, draft=3m, speed=5m/s, starts from 450m toward -x direction.*

- Compile the code with **-DVESSEL** flag in the Makefile
- Go to **FUNWAVE-TVD/simple_cases/vessel_flat_bottom**
- Copy the executable to the current working directory
- Check and Modify **input.txt**

  The following statements are necessary in **input.txt**

Parallel Info (if use parallel)

PX = 4
PY = 1

Depth

DEPTH_TYPE = FLAT
DEPTH_FLAT = 10.0

Dimensions

Mglob = 500
Nglob = 100

Grid sizes

DX = 1.0
DY = 1.0

Set time

TOTAL_TIME = 50.0
PLOT_INTV = 1.0
PLOT_INTV_STATION = 50000.0
SCREEN_INTV = 1.0

Add vessels

VESSEL_FOLDER = ./
NumVessel = 2

NOTE: You will need two vessel files: **vessel_00001** and **vessel_00002** in the current folder.

In **vessel_00001**, specify:

Title: Vessel # 1
Blue_Star_I

Length(m), Width(m), Alpha(0.5), Beta(0.5), P(draft,m)
10.0 5.0, 0.5, 0.5, 2.0

Time, X(m), Y(m) (relative to the origin of the coordinates)
0.0 50.0 40.0
100.0 1050.0 40.0

In **vessel_00002**, specify:

Title: Vessel # 2
Blue_Star_II

Length(m), Width(m), Alpha(0.5), Beta(0.5), P(draft,m)
20.0 8.0, 0.5, 0.5, 3.0

Time, X(m), Y(m) (relative to the origin of the coordinates)
0.0 450.0 60.0
100.0 -50.0 60.0

Output

RESULT_FOLDER = output/
ETA = T

- Use the same procedures as in the previous sessions to run the model and post-process results

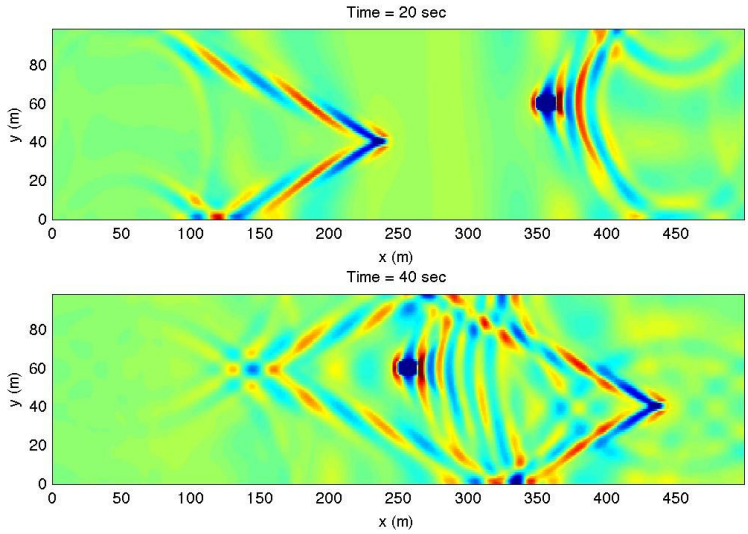


Figure**: an example of plot using matlab code **plot_wave_vessel.m

2) Ship wake-induced sediment transport

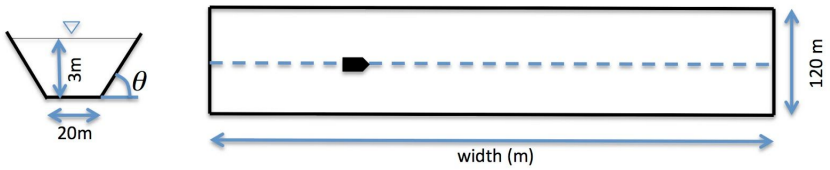


***Figure:** Simulation is performed in a half domain y=0-60m. Grid dimensions 400x60. Grid sizes DX=DY=1.0m. Vessel is specified in vessel_00001. Vessel length=20m, width=10.0, draft=2m.*

***NOTE:** SIMULATION WILL BE PERFORMED IN A HALF DOMAIN.*

Parallel Info (if use parallel)

PX = 4
PY = 1

Depth

DEPTH_TYPE = DATA
DEPTH_FILE = ../bathy/depth.txt

Dimensions

Mglob = 400
Nglob = 60

Grid sizes

DX = 1.0
DY = 1.0

Set time

TOTAL_TIME = 60.0
PLOT_INTV = 2.0
PLOT_INTV_STATION = 0.1
SCREEN_INTV = 2.0

Sediment parameters

Bed_Change = T
BedLoad = T
D50 = 0.0005
Sdensity = 2.68
n_porosity = 0.47
WS = 0.0125
Shields_cr = 0.055
Shields_cr_bedload = 0.047
Tan_phi = 0.7
Kappa1 = 0.3333
Kappa2 = 1.0
MinDepthPickup = 0.1

Add vessels

VESSEL_FOLDER = ./
NumVessel = 1

Output

RESULT_FOLDER = output/
ETA = T

NOTE: You will need a vessel file: **vessel_00001** in the current folder.

In **vessel_00001**, specify:

Title: Vessel # 1
Blue_Star_I

Length(m), Width(m), Alpha(0.5), Beta(0.5), P(draft,m)
20.0 10.0, 0.5, 0.5, 2.0

Time, X(m), Y(m) (relative to the origin of the coordinates)
0.0 40.0 60.0
25523.0 180040 60.0

NOTE: this is based on Froude number = 1.3 == U_vessel / Cp = 7.05 / sqrt(9.81*3.0) = 1.3

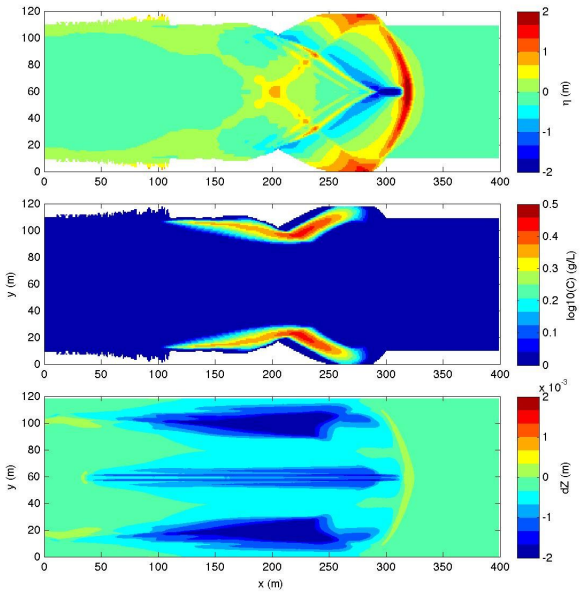
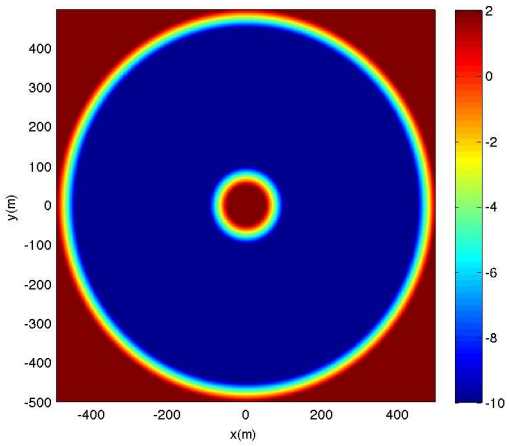


Figure: *(top) surface elevation, (middle) sediment concentration, (bottom) bed change. Use* `/simple_examples/vessel_short_channel/postprocessing/plot_vessel_morpho.m`

3) A vessel moving in a random bathymetry



Model Configuration: *Grid dimensions 500x500. Grid sizes DX=DY=2.0m. Vessel is specified in vessel_00001. Vessel length=20m, width=10.0, draft=1m.*

- Go to **FUNWAVE-TVD/simple_cases/vessel_island_beach**
- Check and Modify **input.txt**

  The following statements are necessary in **input.txt**

Parallel info (if use parallel) … 16 CPU's used for computation here

PX = 4

PY = 4

Depth

DEPTH_TYPE = DATA

DEPTH_FILE = depth.txt

Set up dimensions

Mglob = 500

Nglob = 500

Discretization

DX = 2.0

DY = 2.0

Time

TOTAL_TIME = 300.0

PLOT_INTV = 1.0

SCREEN_INTV = 1.0

Add a vessel

VESSEL_FOLDER = ./

NumVessel = 1

Output

RESULT_FOLDER = output/

ETA = T

In **vessel_00001**, specify:

Title: Vessel # 1
Blue_Star_I
Length(m), Width(m), Alpha(m), Beta(m), P(unit)
20.0 10.0, 0.5, 0.5, 1.0
0.0000000e+00, 5.6000000e+02, 5.0000000e+02
1.0000000e+00, 5.6374897e+02, 5.0255132e+02 … … etc.

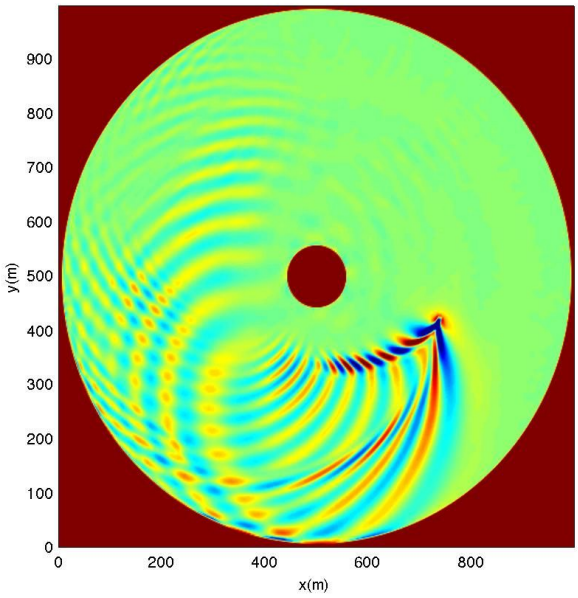


*Figure: example of plot using Matlab code **plot_wave_vessel_island.m***

EXTRA CHALLENGE - PROGRESSION:

Design vessel paths in the inlet-shoal domain. Consider the interaction between wind waves and ship-waves.

- Compile the code with **-DVESSEL**
- Go back to one of the inlet-shoal cases, for example, **input_irr_30deg_ship.txt** (rename it to input.txt when running the case) is in the folder **`/simple_cases/inlet_shoal/input_files/`**

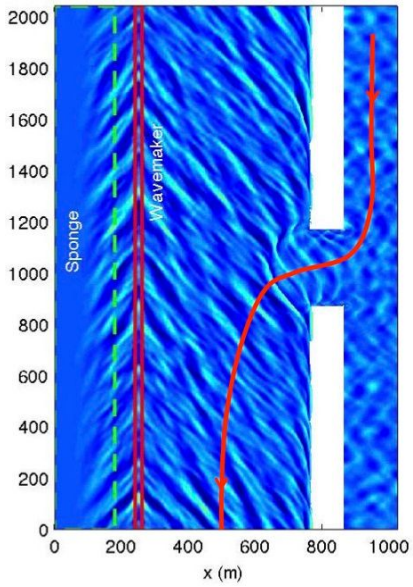
- Modify input.txt

```
VESSEL_FOLDER = ./
NumVessel = 1
```

- Create a vessel file: vessel_00001

  In vessel_00001, specify:

```
Title: Vessel # 1
Blue_Star_I
Length(m), Width(m), Alpha(m), Beta(m), P(unit)
10.0 5.0, 0.5, 0.5, 2.0
Time, X(m), Y(m) (relative to the origin of the coordinates)
0.0 900.0 0.0
150.0 900.0 0.0
250.0 900.0 1000.0
1000.0 -6600 1000.0
```



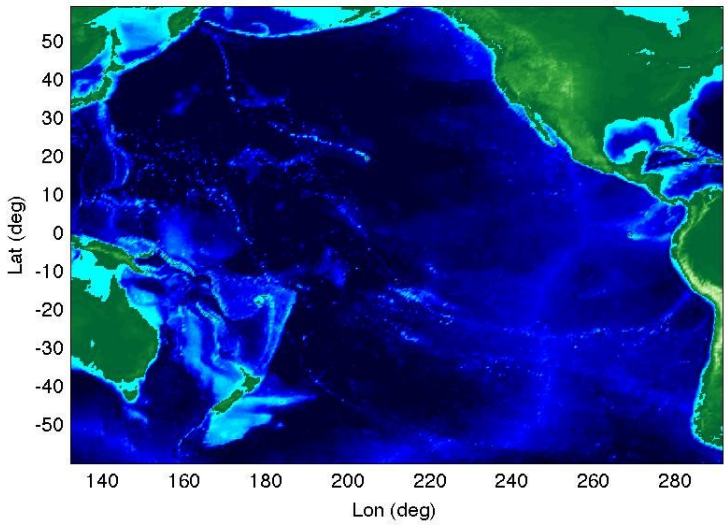
An example of vessel path in the inlet-shoal domain.

Training Session # 6 (Thursday, lead: **Fengyan Shi)**

Topic: Tsunami sources and tsunami simulation using the **spherical coordinates**

Practice

1) Tohoku Tsunami



Model configuration: *Computational domain covers a region of the Pacific Ocean from 60°S to 60°N in the south-north direction, and from 132°E to 68°W in the west-east direction. The example is a 30min x 30min resolution case. Grid dimensions: 320x240, Grid sizes Dphi=Dtheta=0.5 deg.*

- Recompile the source code **without** the flag **-DCARTESIAN** in the Makefile-Mills
- Go to **FUNWAVE-TVD/simple_cases/tohoku_tsunami**
- Copy the new executable into the current working directory
- Check and Modify **input.txt**

The following statements are necessary in **input.txt**

Parallel (if applicable)
PX = 2
PY = 2

Specify bathymetry
DEPTH_TYPE = DATA
DEPTH_FILE = ../external_files/depth_30min.txt

Dimensions
Mglob = 320
Nglob = 240

Grid
Lon_West = 132.0
Lat_South = -60.0
Dphi = 0.5
Dtheta = 0.5

Time

TOTAL_TIME = 86400.0
PLOT_INTV = 3600.0
PLOT_INTV_STATION = 1.0
SCREEN_INTV = 3600.0

Add initial conditions

INI_UVZ = T
ETA_FILE = ../external_files/ETA_30min.txt
U_FILE = ../external_files/U_30min.txt
V_FILE = ../external_files/V_30min.txt

Add Sponge layers

DIRECT_SPONGE = T
FRICTION_SPONGE = T
Sponge_west_width = 100000.0
Sponge_east_width = 100000.0
Sponge_south_width = 100000.0
Sponge_north_width = 100000.0

Add friction

Cd = 0.001

Avoid inundation in the basin scale (specify a large minimum depth)

MinDepth= 10.0

Stations/Wave Gauges

NumberStations = 78
STATIONS_FILE = stations-pacific.txt

Output

RESULT_FOLDER = output/
ETA = T
Hmax = T

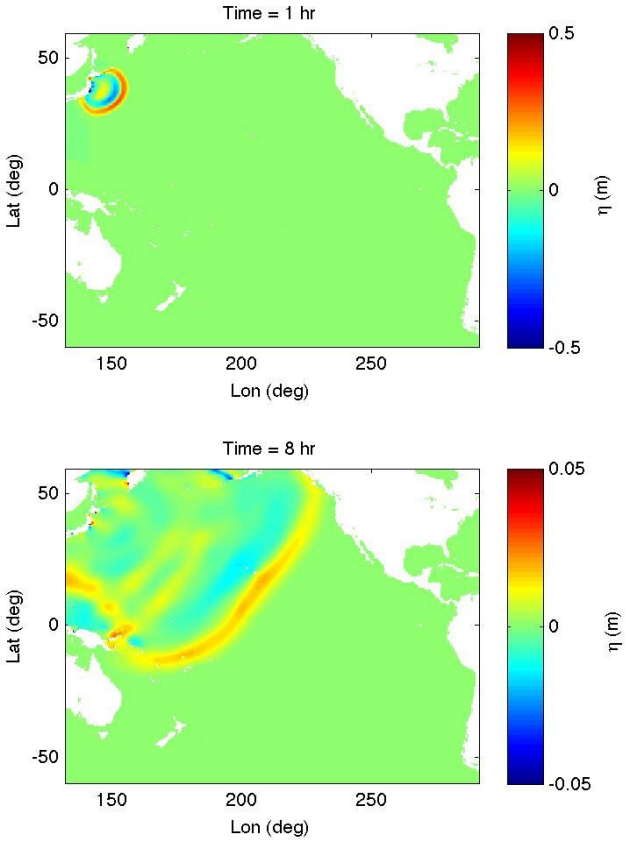
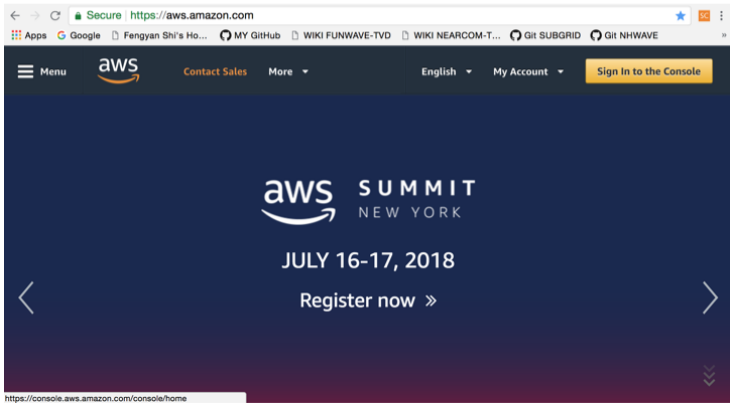


Figure: *an example of plot using Matlab code* ***plot_surface.m***

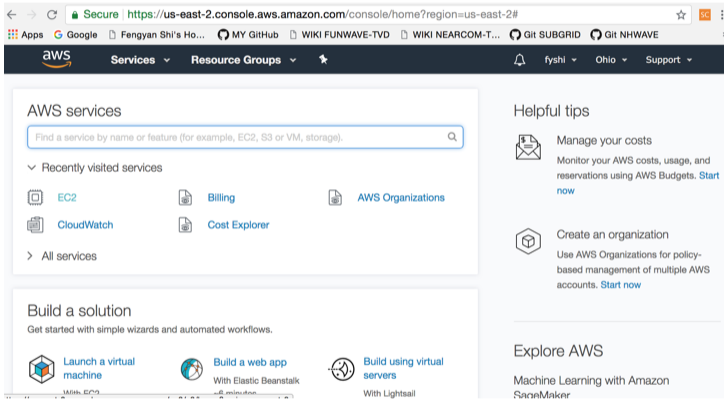
APPENDIX: *Amazon AWS Cloud Computing (EC2)*

Amazon AWS provides 12-month free tier https://aws.amazon.com/free/

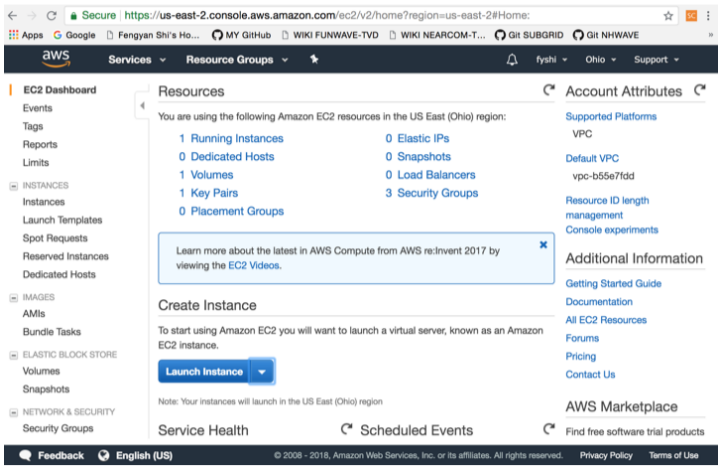
1. Sign in AWS or 'Register now' if you don't have an account



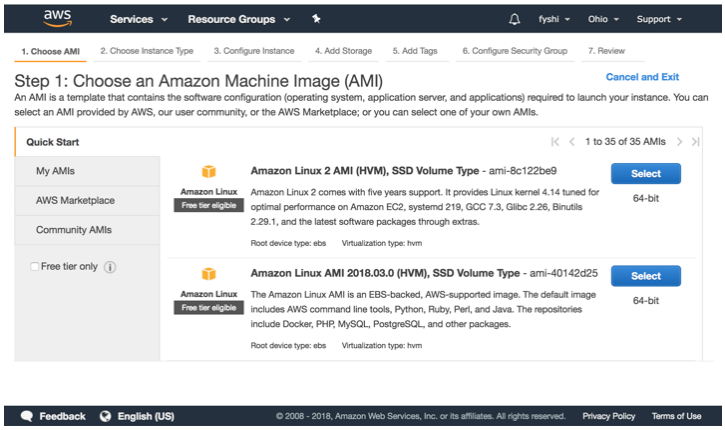
2. Click EC2 to choose EC2 service



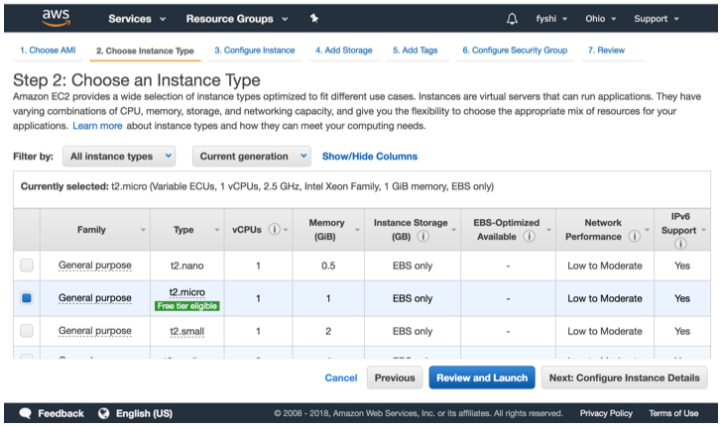
3. Create instance (build a machine for you)



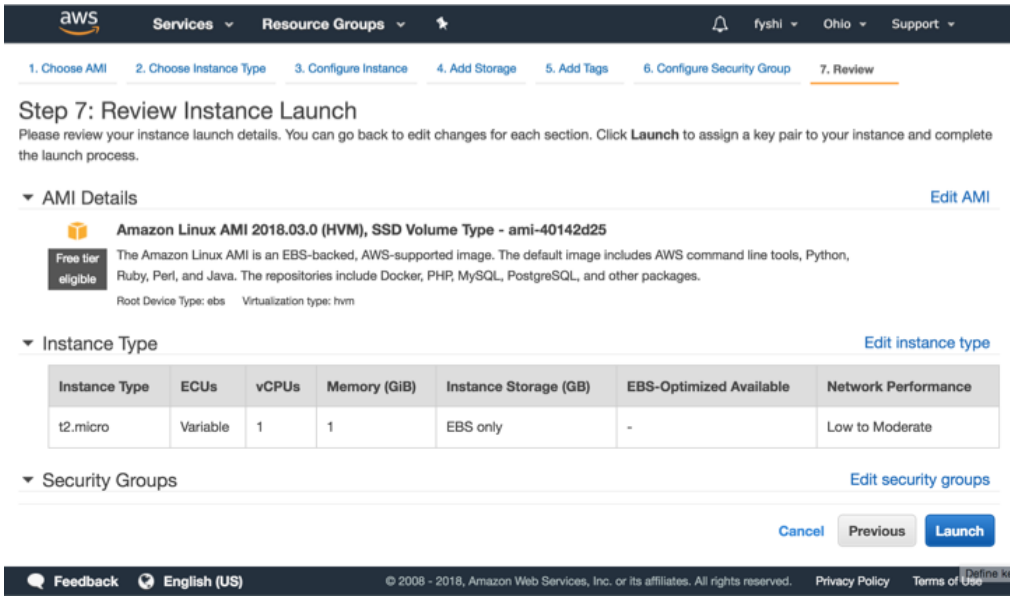
4. Select a machine (I usually choose Linux AMI 2018)



5. Select instance type (any free one)

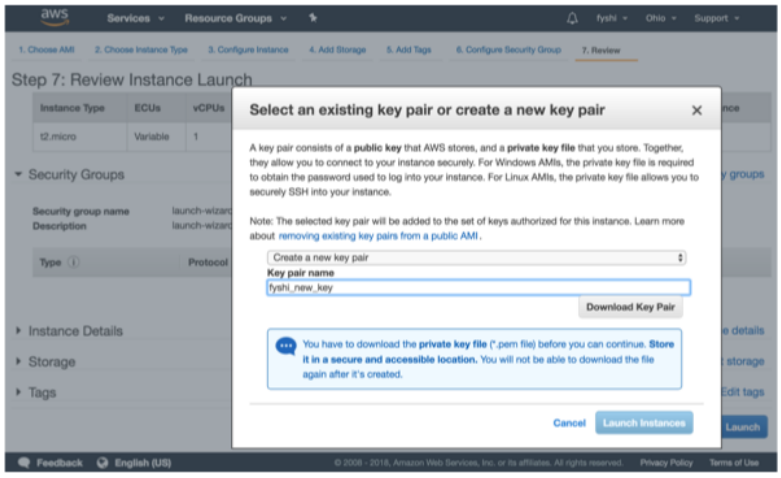


6. Click Launch

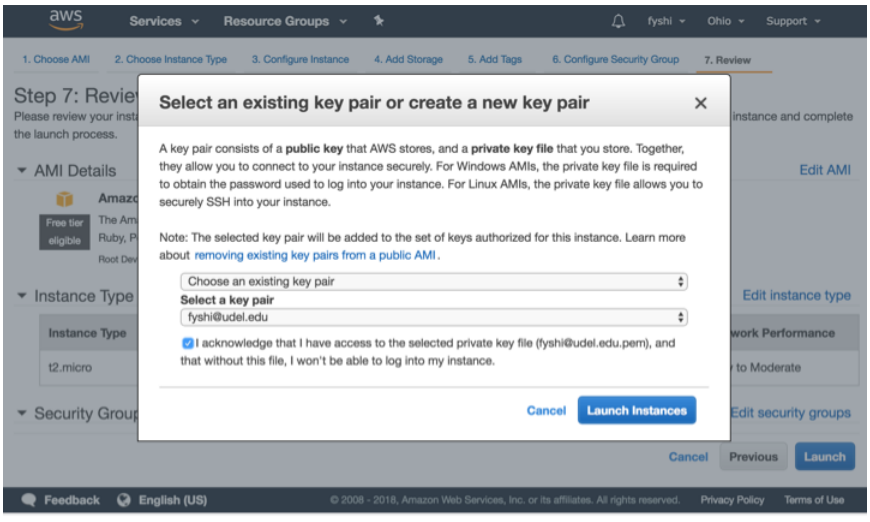


7. If you don’t have a key pair, create a new one

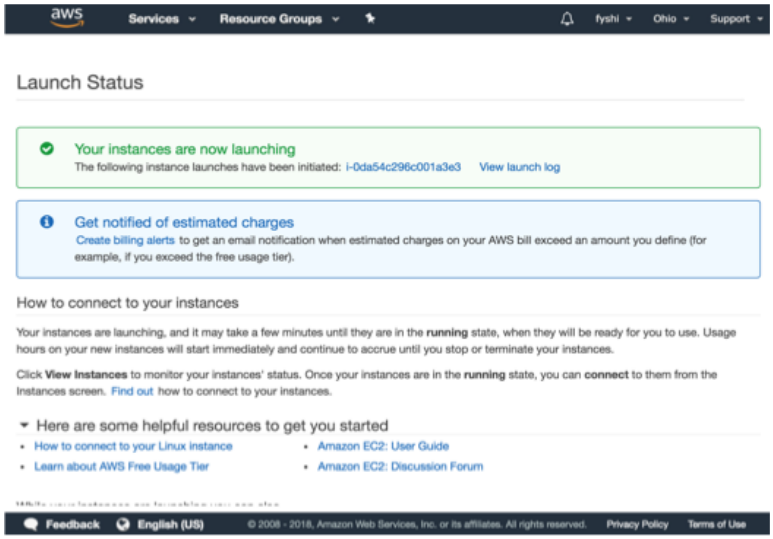
- Choose **‘create a new key pair’**
- Name it in the second line **(here I type fyshi_new_key)**
- Download **Key Pair**
- Store the downloaded file **(fyshi_new_key.pem)** into a directory you can find later
- Go to the directory**,** in command line**, type: chmod 400 fyshi_new_key.pem**
- You can re-use the **Key Pair** in AWS **(next page)**



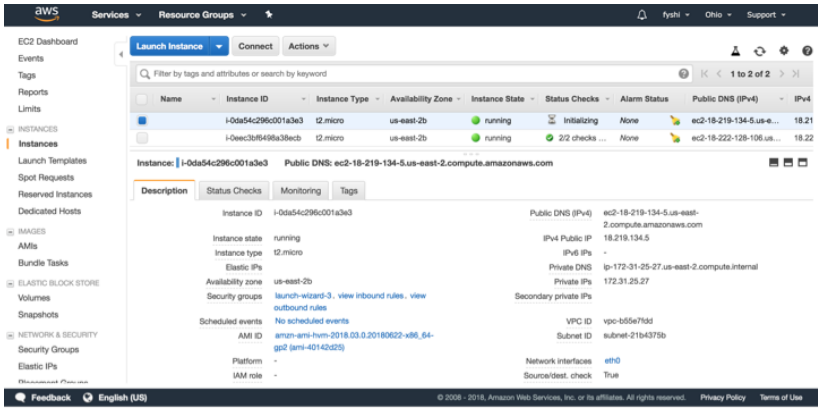
7-A. You can re-use the Key Pair if you already had one



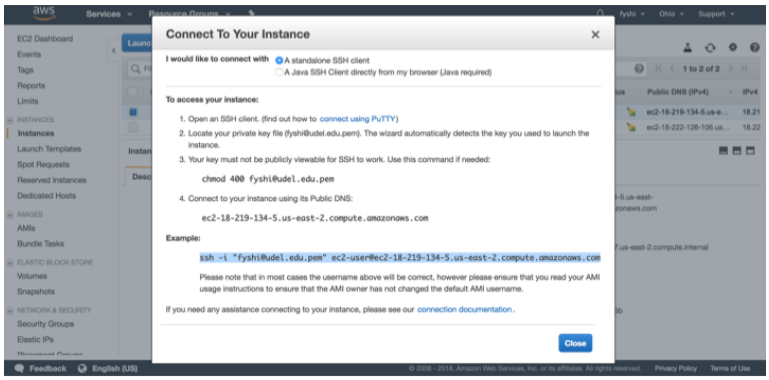
8. After successful launch, you will see the following page



9. Click Instances on left panel, you will see instances (here I launched two) running
   Choose one (blue square) you want to access by a terminal, Click Connect



10. Copy the line highlighted (paste in the terminal later)



11. Access to the virtual machine using ssh

A. Go to the directory you stored your Key pair file fyshiudeledu.pem:

```
$ cd directory_you_store_key_pair
```

B. ssh (or sftp), paste the ssh link you did in step 10:

```
$ ssh -i "fyshiudeledu.pem"
ec2-user@ec2-18-219-134-5.us-east-2.compute.amazonaws.com
```

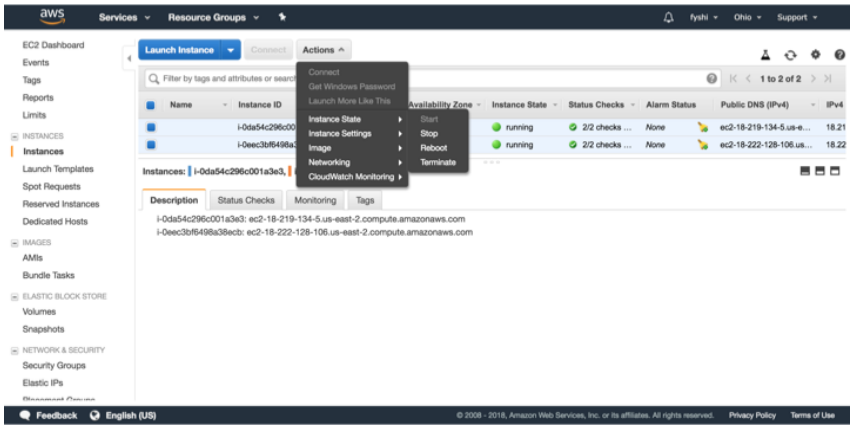
```
The authenticity of host 'ec2-18-219-134-5.us-east-2.compute.amazonaws.com
(18.219.134.5)' can't be established.
RSA key fingerprint is 72:49:9d:11:cd:4c:70:79:8c:06:3e:6c:66:aa:59:6c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added
'ec2-18-219-134-5.us-east-2.compute.amazonaws.com,18.219.134.5' (RSA) to the
list of known hosts.
       __|  __|_  )
       _|  (     /   Amazon Linux AMI
       ___|\___|___|
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
```

12. Start to use the virtual machine

13. After using it, you should stop your instances.

The “bill” will be based on the time used by instances after the free period

- To stop or terminate your instances, click Actions ->Instance state-> stop or terminate
- If you terminate your instance, everything associated with this instance will be removed, including software you installed.



13. Install **Git, gfortran, MPICH,** and **FUNWAVE-TVD**

Using the terminal that you created in step 11, go and

A) Install Git by

```
$ sudo yum install git
```

B) Install gfortran

```
$ sudo yum install gcc-gfortran
```

C) Install MPICH by

- download MPICH at http://www.mpich.org/downloads/
- use sftp to put the downloaded file `mpich-3.2.1.tar.gz` into the virtual machine
- `$ tar -xzf mpich-3.2.1.tar.gz`
- `$ cd mpich-3.2.1/`
- `$ ./configure --disable-cxx`
- `$ make`
- `$ sudo make install`

D) Get FUNWAVE-TVD

```
$ git clone https://github.com/fengyanshi/FUNWAVE-TVD.git
```

E) Compile the code and test it! And ….. Voila!