FUNWAVE

Phase Resolving (Boussinesq-type) Numerical Wave Model
CHL-Coastal & Hydraulics Laboratory
U.S. Army Engineer Research and Development Center
(Originally developed by James Kirby and Fengyan Shi of University of Delaware)

FUNWAVE-1D GUI Instructions

CHL – Coastal Processes Branch
Gabriela Salgado-Dominguez
Dr. Matt Malej

Table of Contents

FUNWAVE-1D GUI Instructions **1**

Introduction **3**

What's it all about 3

Step 1: Input Project Title **4**

Step 2: Bathymetry **5**

Generate Flat Bathymetry 6

Generate Slope Bathymetry 7

Generate Uploaded Bathy 8

Step 3: Input File **10**

Project Intro 10

Initial conditions 11

Wave Maker 12

Sponge Layer 17

Output Options 19

Generate Input 21

Step 4: Run Model **26**

Step 5: Post Process **28**

Introduction

What's it all about

This manual contains a fairly detailed discussion of the One-dimensional FUNWAVE Graphical User Interface (GUI). With the help of this GUI the user will be capable of setting simple 1D simulations for flat, slope or uploaded bathymetries with a variety of wave makers (such as regular, irregular, solitary and JONSWAP). The user will also be able to select other parameters like sponge layer, output variables, initial conditions, etc. All of these features originally were done through the terminal and text editors; but now everything is streamlined since it is done through a browser (**Google Chrome preferably**). This will provide the user a straightforward experience to FUNWAVE.

The description given here is by no means complete. Those who want a more detailed walkthrough can access the FUNWAVE wiki (http://udel.edu/~fyshi/FUNWAVE/definition.html), or contact Gabriela Salgado-Dominguez (gabriela.salgado-dominguez@erdc.dren.mil) or Matt Malej (Matt.Malej@erdc.dren.mil).

Step 1: Input Project Title

The user must input his **project title** *(part a of Figure 1).* Once the **"Generate Project" button** *(part b of Figure 1)* is pressed, a folder with that name will be created. From now on, all the files corresponding to that project (e.g., input.txt, depth.txt, output…) will be saved there. The GUI will warn the user if the name given has been already used, so that the user can input another.

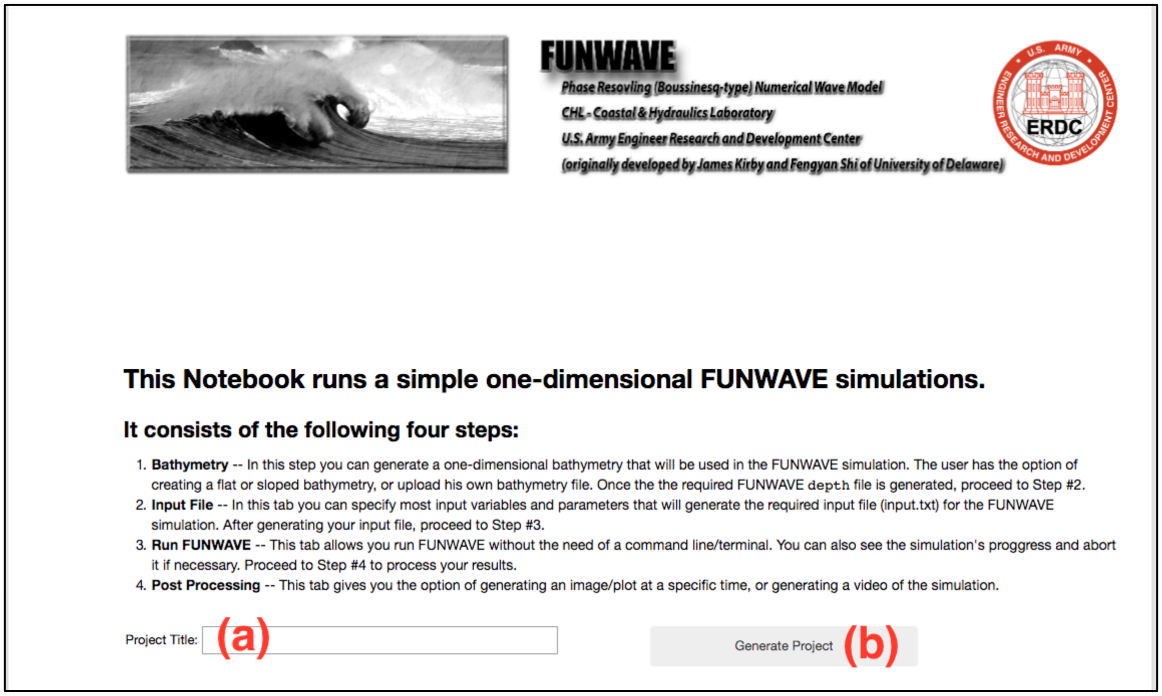

Figure 1: Input Project Title

Step 2: Bathymetry

In this step the user can generate a one-dimensional bathymetry that will be used in the FUNWAVE simulation. The user has the option of creating a flat or sloped bathymetry, or upload a bathymetry file *(Figure 2)*. How these bathymetries are generated will be explained in the following pages.

NOTE: The values are in **metric** units. Also, keep in mind that Depth values are [-] for underwater (below mean water level).

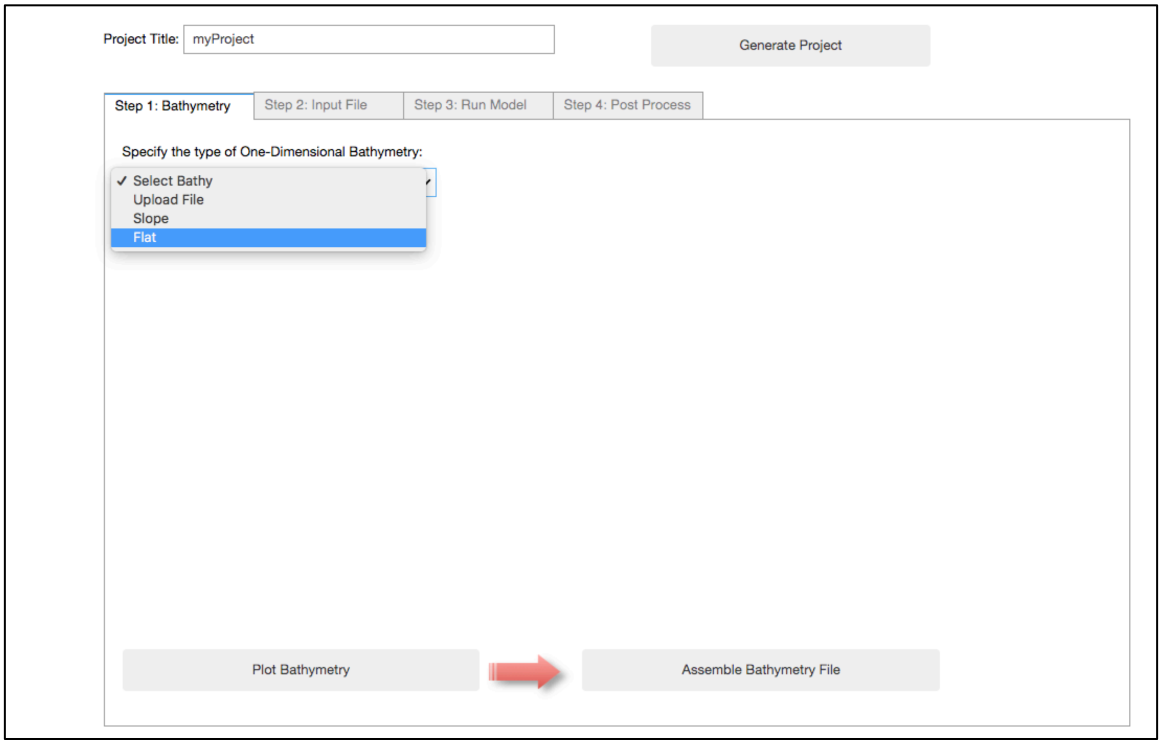

Figure 2: Generate Bathymetry Tab

Generate Flat Bathymetry

This option plots and saves a simple one-dimensional **flat** bathymetry with a constant depth.

- The user can select the **total horizontal length (THL)** of the bathymetry *(part a of Figure 3)*, the **spacing between the points (DX)** *(part b of Figure 3)*, and the Depth *(part c of Figure 3).*
- Once the desired bathymetry is plotted *(part d of Figure 3)* and the user is satisfied, press the **"Assemble Bathymetry File"** button *(part e of Figure 3)* and proceed to next step.

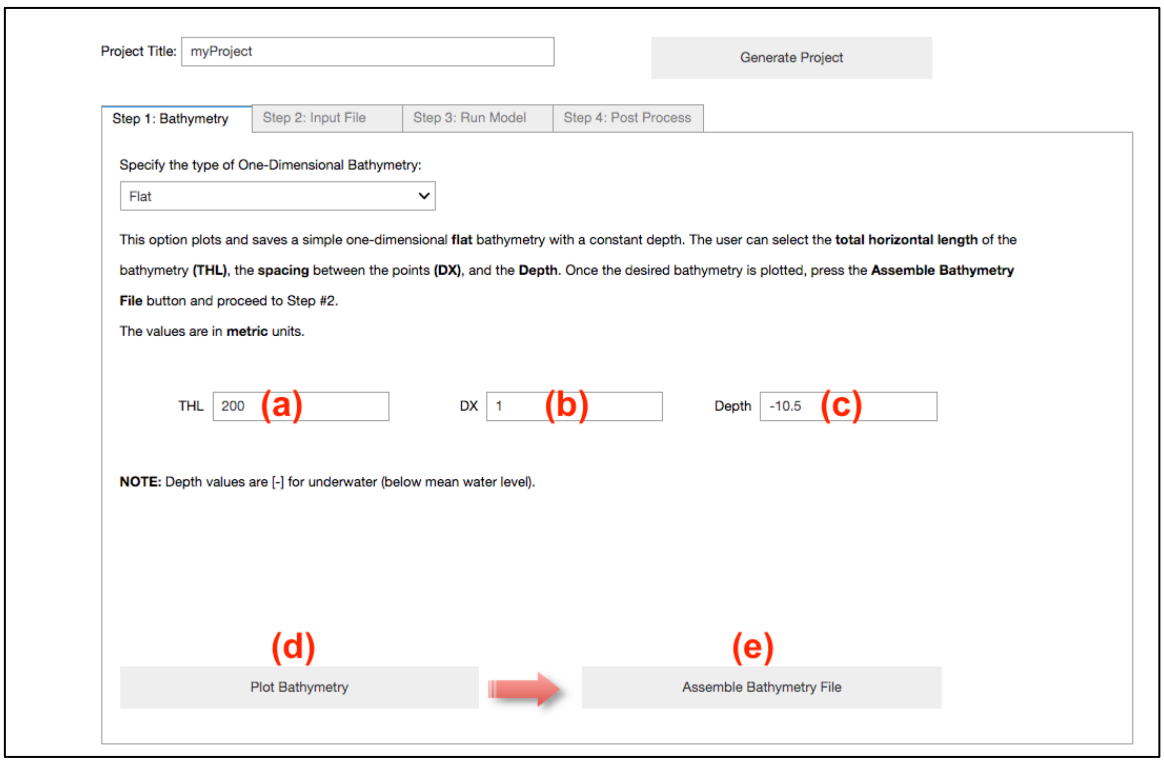

Figure 3: Generate Flat Bathymetry

Generate Slope Bathymetry

This option plots and saves a simple one-dimensional **slope** bathymetry consisting of 2 segments and 3 vertices.

- The user can select the **total horizontal length of the bathymetry (THL)** *(part a of Figure 4)*, the **discretization (horizontal spatial step size) (DX)** *(part b of Figure 4)*, and the elevation with the location of the vertices *(part c of Figure 4)*.
- Once the desired bathymetry is plotted **(part d of Figure 4)** and the user is satisfied with it, press the **"Assemble Bathymetry File"** button **(part e of Figure 4)** and proceed to the next step.

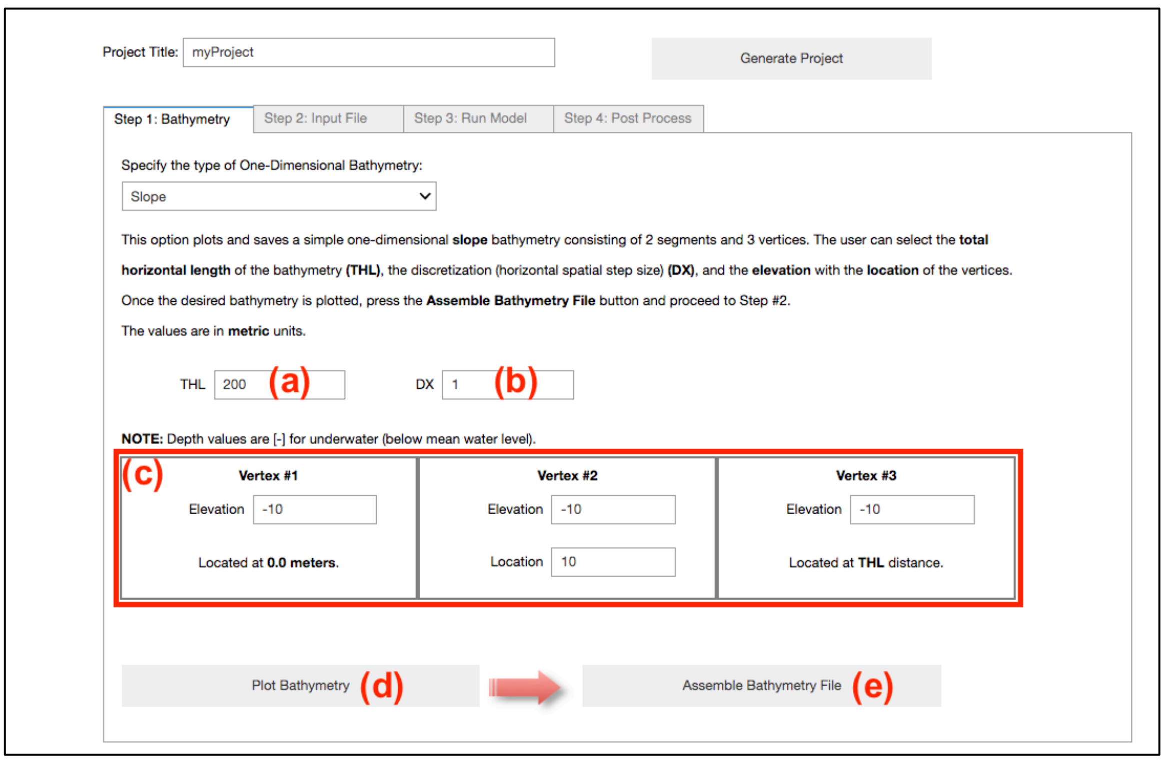

Figure 4: Generate Slope Bathymetry

Generate Uploaded Bathy

Substep 1: Upload the file in the Project Title folder (called "myProject" for this example) through the **Jupyter Notebook home directory** *(Figure 5)*. **Go inside the Project's folder** *(part a in Figure 5)* and upload the file by pressing the **"Upload" button** located at the top right corner of the directory *(part b of Figure 5)*.

> **NOTE:** This file must be a text file (ASCII) of 1 column; with depth values [-] for underwater (below mean water level). Also, the values must be in **metric** units.

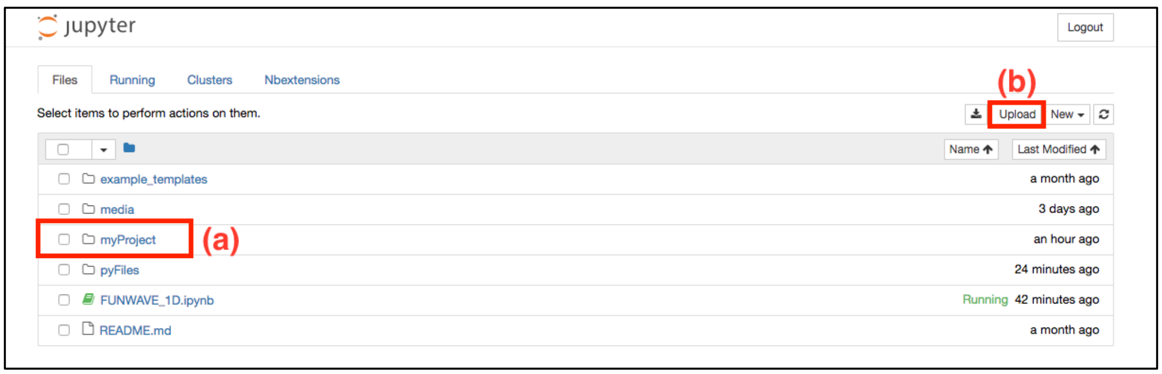

Figure 5: Jupyter Notebook home directory

Substep 2: Once the file is uploaded, identify its name (e.g., myBathy.txt) *(part a of Figure 6),* and either its **Total Horizontal Length (THL)** *(part b1 of Figure 6)* or its **step size (DX)** *(part b2 of Figure 6)* in the dropdown list. Press **"Plot Bathymetry"** *(part c of Figure 6)* to visualize the bathymetry.

Substep 3: If the user is satisfied with the bathymetry, the user must press "**Assemble Bathymetry File"** *(part c of Figure 6)* button before continuing to the next step. This will format the uploaded bathymetry file to the **required FUNWAVE format.**

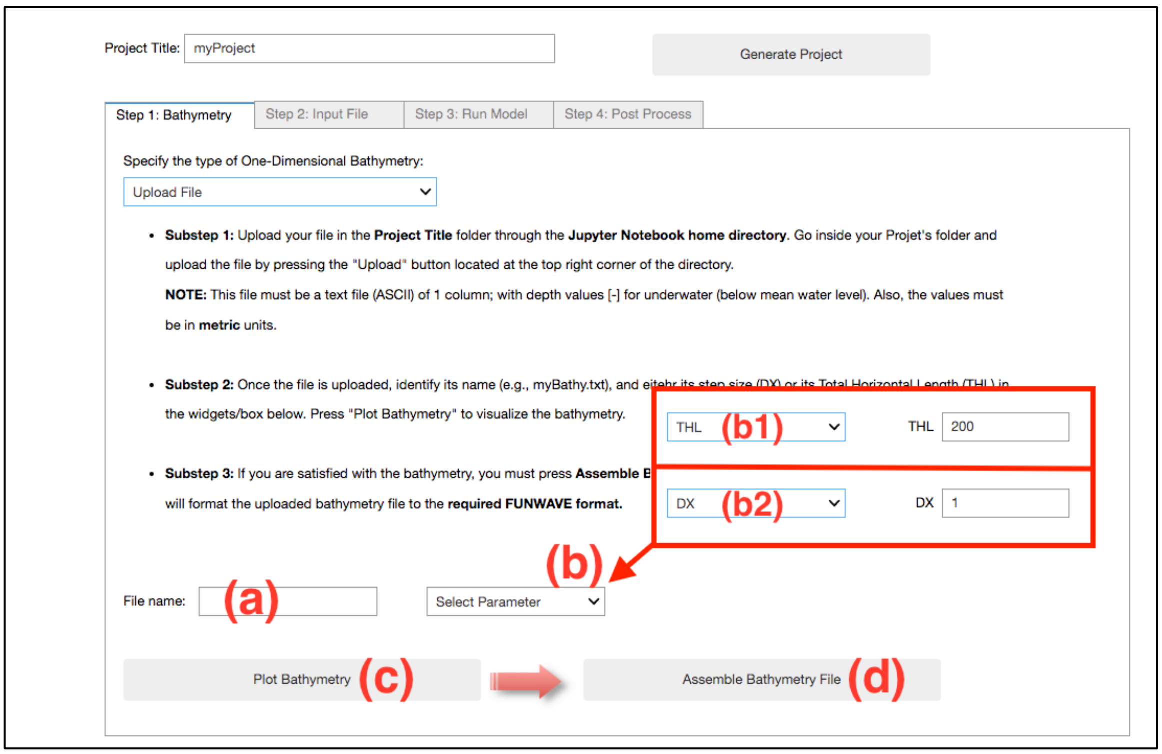

Figure 6: Upload Bathymetry File

Step 3: Input File

In this tab the user can specify most input variables and parameters that will generate the required input file (input.txt) for the FUNWAVE simulation. It consists of the following sections:

Project Intro

In this section the user submits the **Number of Processors** (*part a of Figure 7*), the **simulation's Total Time** (*part b of Figure 7*), and the **Plot Interval** (*part c of Figure 7).*

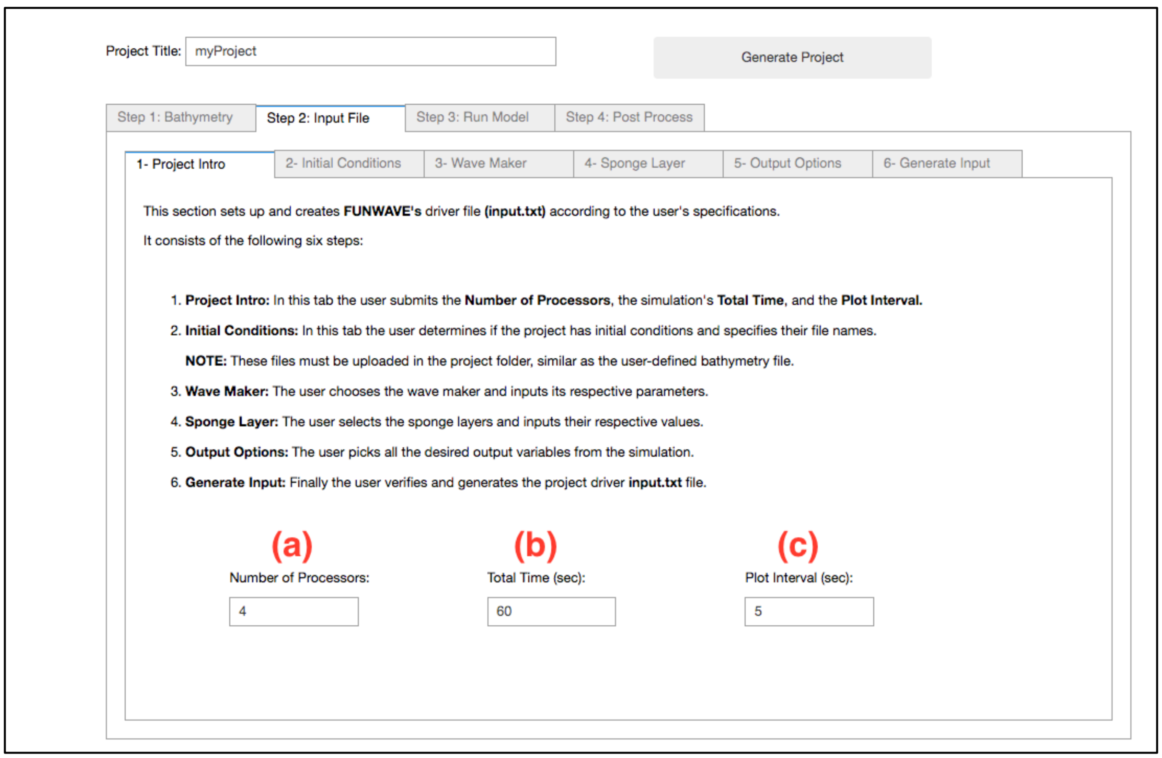

Figure 7: Input File - Project Intro section

Initial conditions

In this tab the user determines if the project has initial conditions and if so, specifies their file names. These files must be uploaded in the project folder, similar as the user-defined bathymetry file (*see "Generate Uploaded Bathymetry" section, where it explains how to do so)*.

Click the **checkbox** *(part a of Figure 8)* if the user wants to initialize the simulation with none-zero elevation and velocity values. The **Initial surface (Z) file** *(part b of Figure 8)* dictates if there is a perturbation on the originally flat-water surface. The **initial U velocity file** *(part c of Figure 8)* specifies the velocities in the x direction.

Once the user has identified their names (e.g. Ini_Z.txt), press **"Generate Initial Condition Files"** *(part d of Figure 8)* to format the uploaded files to the FUNWAVE format.

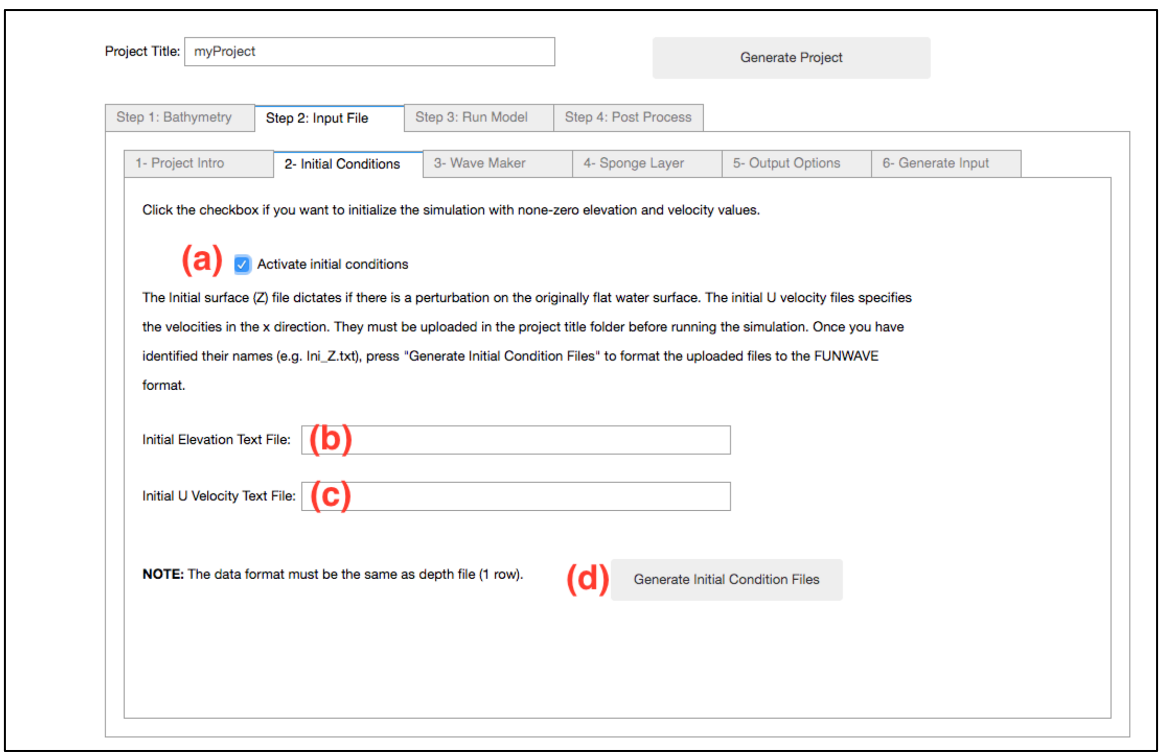

Figure 8: Input File – Generate Initial Condition Files

Wave Maker

In this tab the user determines the wave maker that will be used on his simulation. Following is a summary of all the wave maker options available.

NOTE: These are internal wave makers. Hence, they should NOT be located at the boundary (x=0).

INI_SOL: initial solitary wave *(Figure 9).*

a) **X_wavemaker:** x (m) coordinate for INI_SOL wave maker*.*
b) **DEP:** water depth at wave maker location for INI_SOL wave maker*.* This value will be taken from the depth file, depending on the X coordinate given.
c) **AMP:** amplitude (m) of initial η for INI_SOL.
d) **LagTime:** time lag (s) for the solitary wave generated on the left/west boundary.

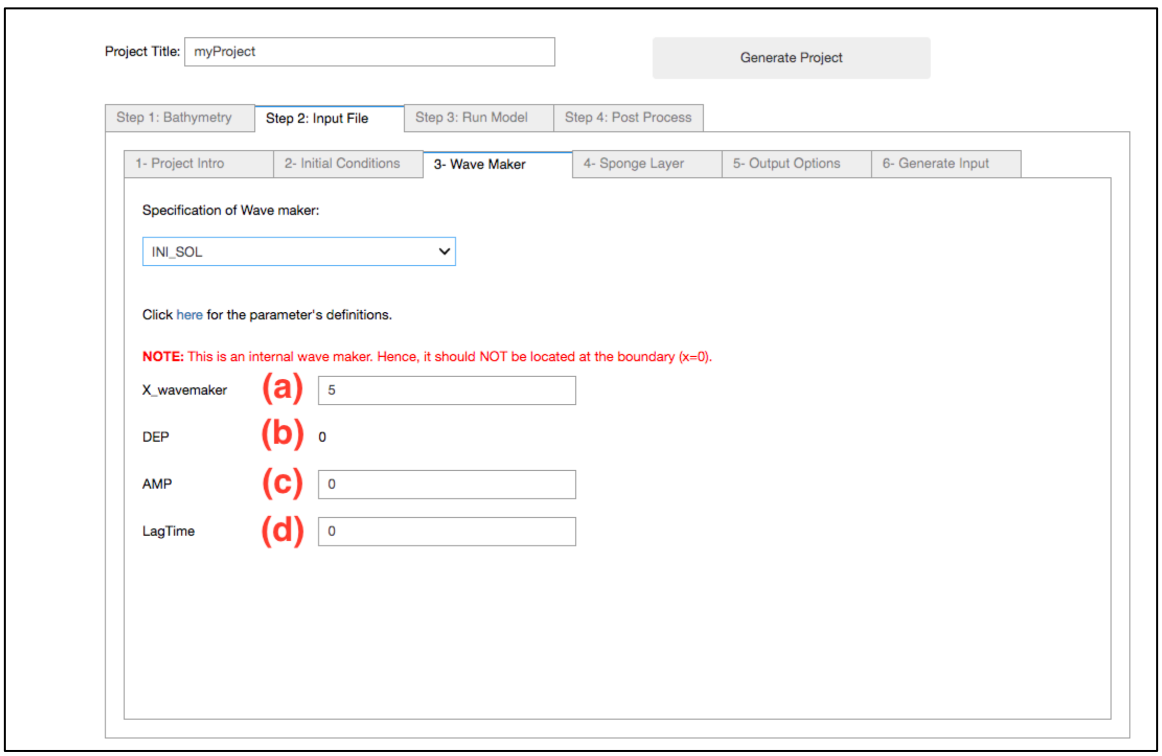

Figure 9: Input File – INI_SOL Wave Maker

INI_REC: initial rectangular hump *(Figure 10).*

a) **Xc:** x (m) coordinate of the center of a rectangular hump for INI_REC wave maker.

b) **AMP:** amplitude (m) of initial η for INI_REC wave maker.

c) **WID:** width (m) of a rectangular hump for INI_REC wave maker.

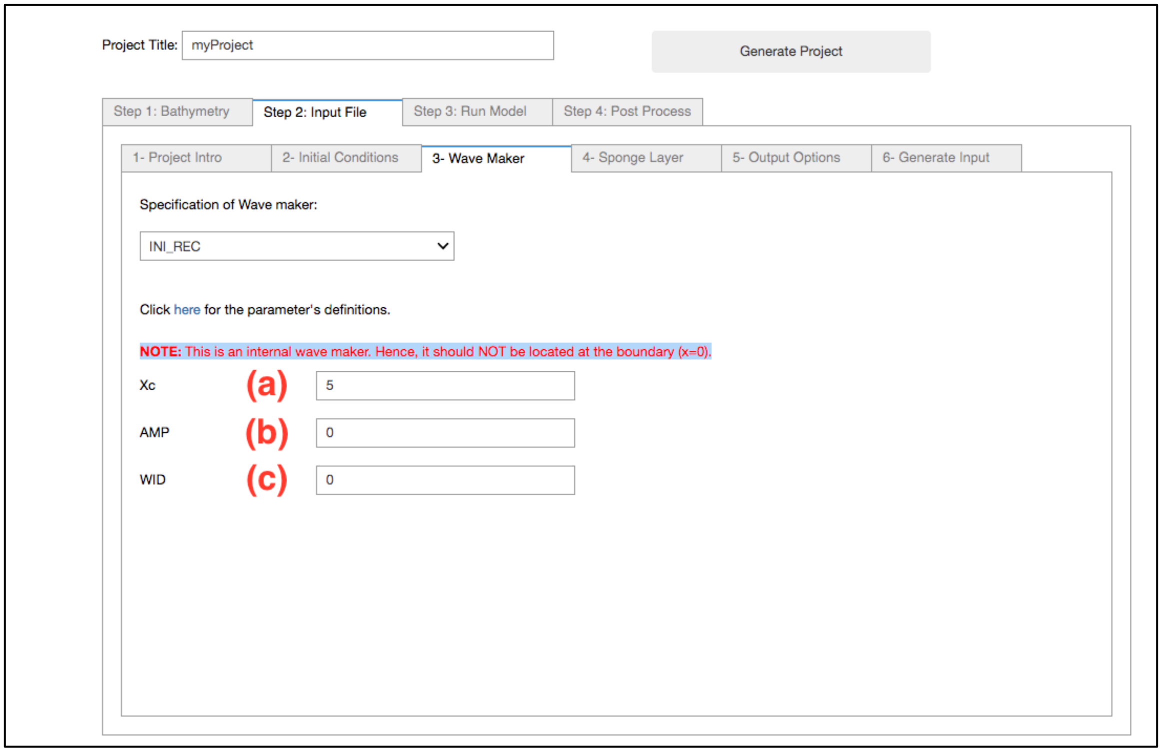

Figure 10: Input File – INI_REC Wave Maker

WK_REG: Wei and Kirby regular internal wave maker *(Figure 11).*

a) **Xc_WK:** x coordinate (m) for WK_REG.

b) **DEP_WK:** water depth at wave maker location for WK_REG wave maker*.* This value will be taken from the depth file, depending on the X coordinate given.

c) **Tperiod:** period (s) of regular wave for WK_REG wave maker*.*

d) **AMP_WK:** amplitude (m) of initial η for WK_REG wave maker.

e) **Theta_WK:** direction (degrees) of regular wave for WK_REG wave maker.

f) **DELTA_WK:** width parameter δ for WK_REG wave maker. Need trial and error, usually, $\delta = 0.3 \sim 0.6$ (not more than 1).

g) **Time_ramp:** time ramp (s) for WK_REG wave maker.

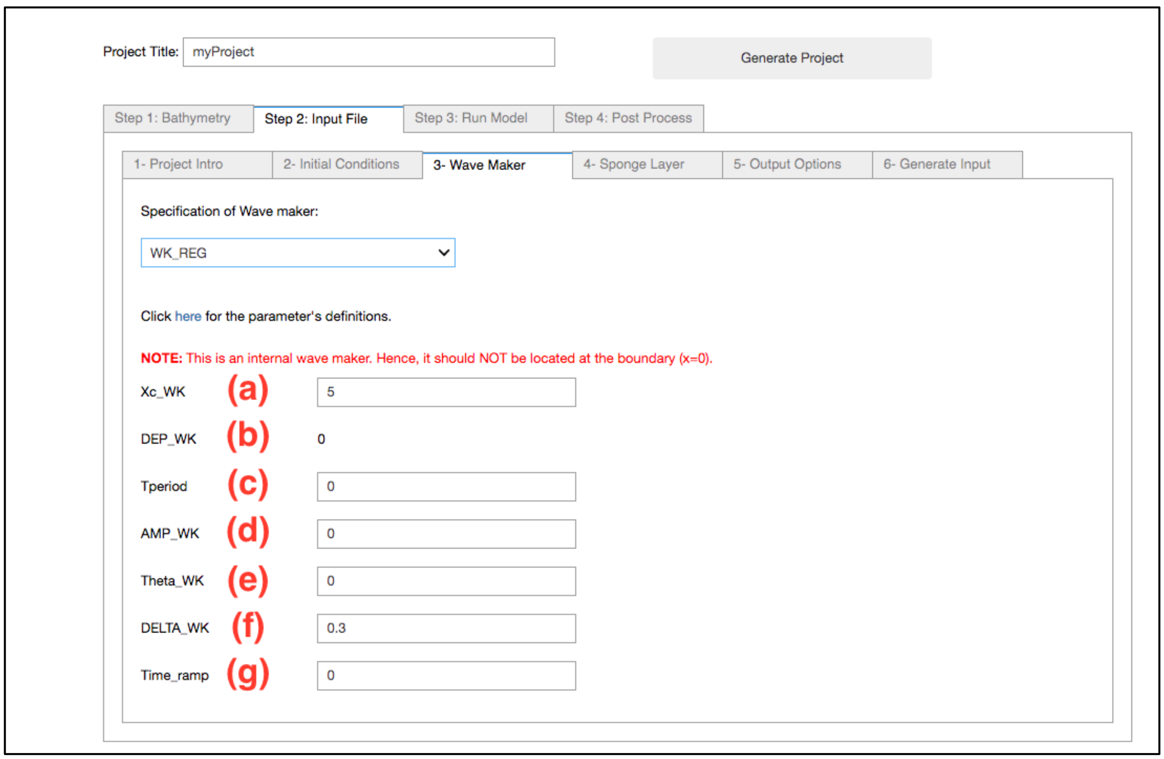

Figure 11: Input File – WK_REG Wave Maker

JON_1D: JONSWAP 1D spectrum wavemaker *(Figure 12)*.

a) **Xc_WK:** x coordinate (m) for JON_1D.
b) **DEP_WK:** water depth at wave maker location for JON_1D wave maker*.* This value will be taken from the depth file, depending on the X coordinate given.
c) **Time_ramp:** time ramp (s) for JON_1D wave maker.
d) **GammaTMA:** TMA parameter γ for JON_1D wave maker. Default 3.3.
e) **Nfreq:** Number of frequency bins. Default 45.
f) **FreqPeak:** peak frequency (1/s) for JON_1D wave.
g) **FreqMin:** low frequency cutoff (1/s) for JON_1D wave.
h) **FreqMax:** high frequency cutoff (1/s) for JON_1D wave.
i) **DELTA_WK:** width parameter δ for JON_1D wave maker. Need trial and error, usually, $\delta = 0.3 \sim 0.6$ (not more than 1).
j) **Hmo:** wave height (m) for JON_1D wave maker.

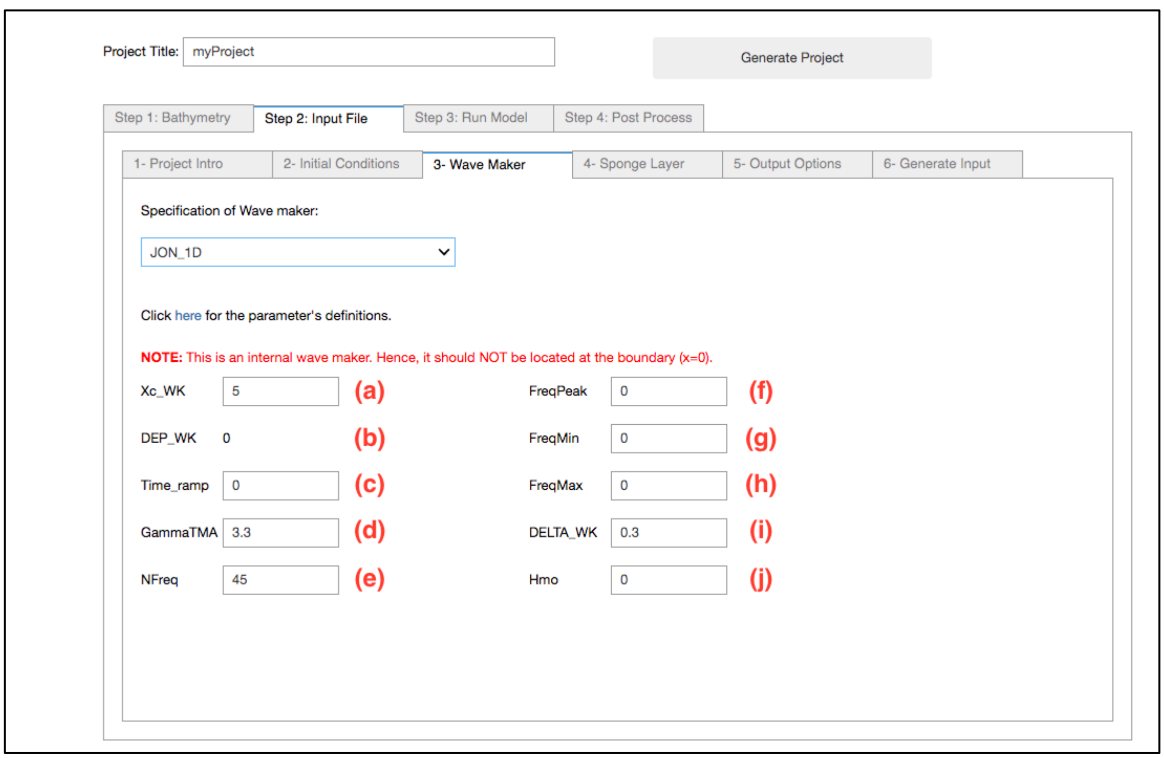

Figure 12: Input File – JON_1D Wave Maker

TMA_1D/IRR_WAVE: irregular wave spectrum wave maker *(Figure 13)*.

a) **Xc_WK:** x coordinate (m) for TMA_1D.
b) **DEP_WK:** water depth at wave maker location for TMA_1D wave maker*.* This value will be taken from the depth file, depending on the X coordinate given.
c) **Time_ramp:** time ramp (s) for TMA_1D wave maker.
d) **GammaTMA:** TMA parameter γ for TMA_1D wave maker. Default 5.
e) **Nfreq:** Number of frequency bins. Default 45.
f) **FreqPeak:** peak frequency (1/s) for TMA_1D wave.
g) **FreqMin:** low frequency cutoff (1/s) for TMA_1D wave.
h) **FreqMax:** high frequency cutoff (1/s) for TMA_1D wave.
i) **DELTA_WK:** width parameter δ for TMA_1D wave maker. Need trial and error, usually, $\delta = 0.3 \sim 0.6$ (not more than 1).
j) **Hmo:** wave height (m) for TMA_1D wave maker.

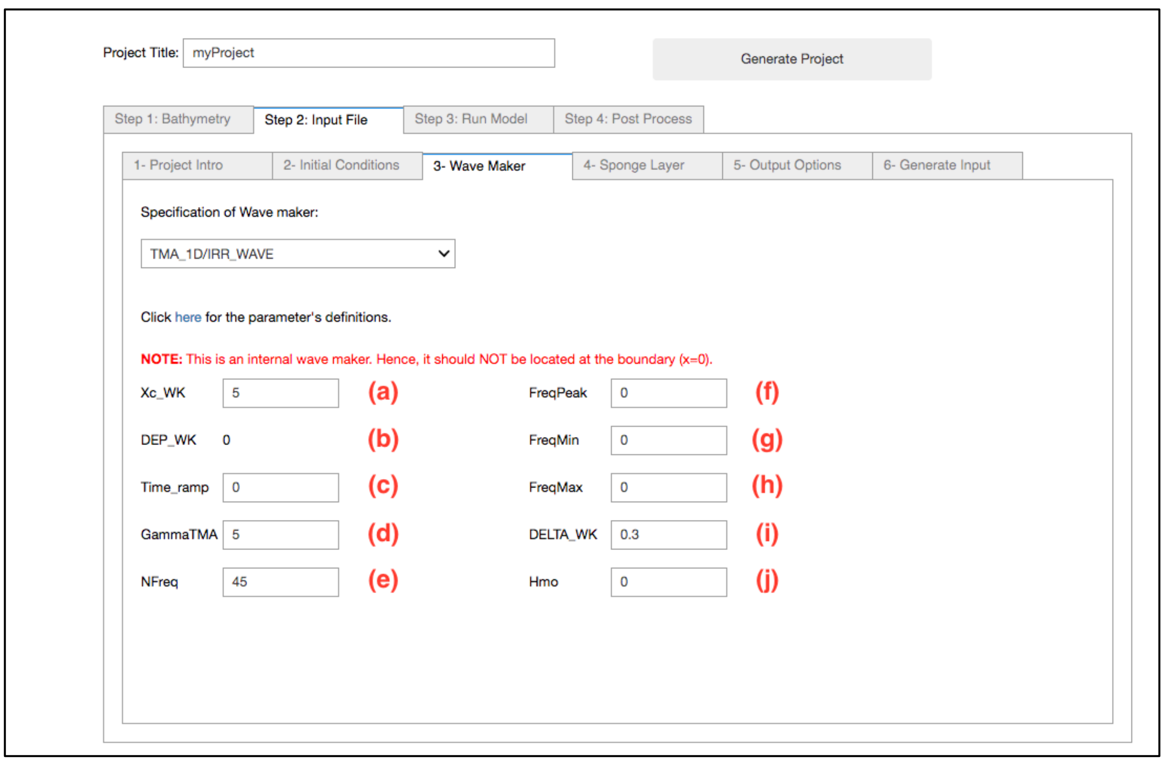

Figure 13: Input File – TMA_1D Wave Maker

Sponge Layer

FUNWAVE possess a DHI type sponge layer. The user needs to specify the widths of two boundaries *(part d of Figure 14)* and parameters.

a) **Diffusion Sponge:** logical parameter for diffusion type sponge. T - sponge layer, F - no sponge layer. If checkbox is on, the user must input the Diffusion Coefficient.

   - **Diffusion Coefficient (CSP):** The maximum diffusion coefficient for diffusion type sponge. The user can use the values 0.5, 1.0, 1.5 for weak, intermediate and strong effects respectively.

b) **Friction Sponge:** logical parameter for friction type sponge. T - sponge layer, F - no sponge layer. If checkbox is on, the user must input the Friction Coefficient.

   - **Friction Coefficient (CDsponge):** The maximum Cd for friction type sponge.

c) **Direct Sponge:** logical parameter for L-D type sponge, T - sponge layer, F - no sponge layer. If checkbox is on, the user must input the Decay Rate and Maximum Decay Rate.

   - **Decay Rate (R_sponge):** decay rate in L-D type sponge layer. Its values are between 0.85 ~ 0.95.
   - **Maximum Decay Rate (A_sponge):** maximum damping magnitude in L-D type sponge. The value is ~ 5.0.

d) Width (m) of sponge layer at west and east boundaries.
   Set width = 0.0 if no sponge.

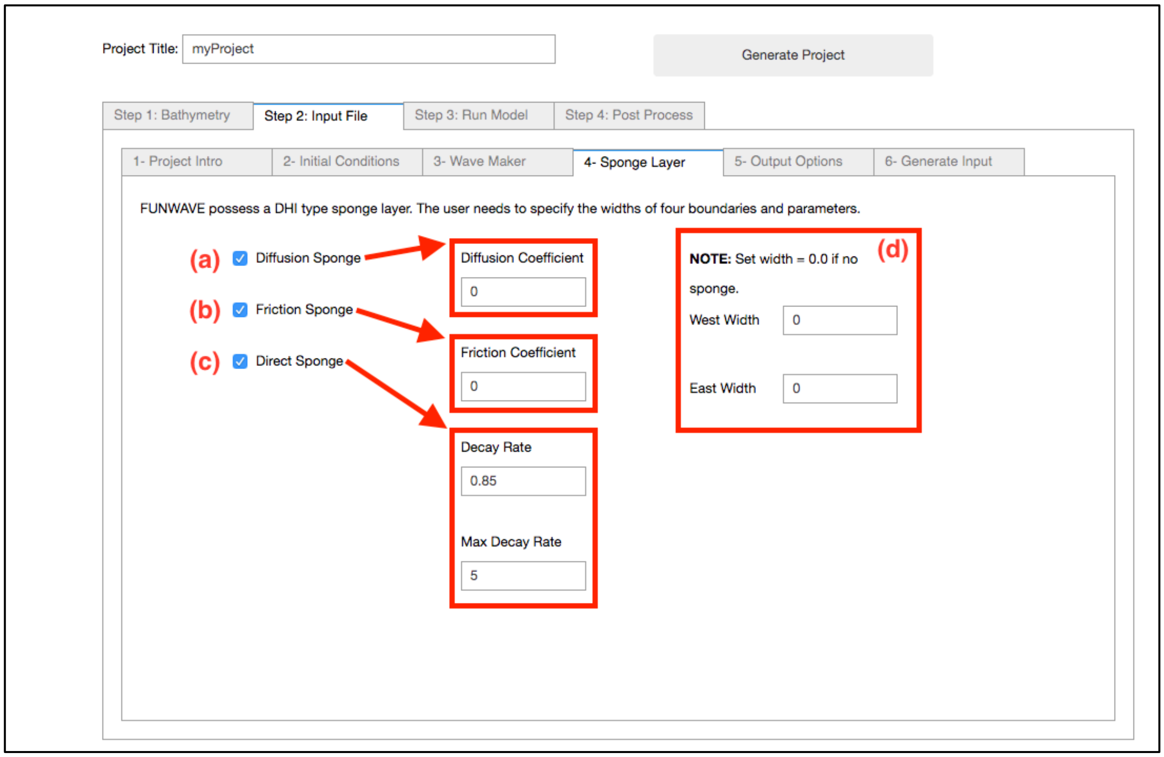

Figure 14: Input File – Sponge Layer

Output Options

The output files are saved in the output folder of the project's directory. For outputs in ASCII, a file name is a combination of variable name and an output series number such eta_00001, eta_00002, etc.… The format and read/write algorithm are consistent with a depth file.

This tabs *(Figure 15)* enables the user to select the following output variables:

- Depth (DEPTH_OUT): logical parameter for output depth.
- U: logical parameter for output u.
- V: logical parameter for output v.
- ETA: logical parameter for output η.
- MASK9: logical parameter for output MASK9 (switch for Boussinesq/NSWE).
- SourceX: logical parameter for output source terms in x direction.
- SourceY: logical parameter for output source terms in y direction.
- P: logical parameter for output of momentum flux in x direction.
- Q: logical parameter for output of momentum flux in y direction.
- Fx: logical parameter for output of numerical flux F in x direction.
- Fy: logical parameter for output of numerical flux F in y direction.
- Gx: logical parameter for output of numerical flux G in x direction.
- Gy: logical parameter for output of numerical flux G in y direction.
- AGE: logical parameter for output of breaking age.
- HMAX: logical parameter for output of recorded maximum surface elevation.
- HMIN: logical parameter for output of recorded minimum surface elevation.
- UMAX: logical parameter for output of recorded maximum velocity.
- VORMAX: logical parameter for output of recorded maximum vorticity.
- MFMAX: logical parameter for output of recorded maximum momentum flux.

- Wave H (WaveHeight): logical parameter for output of wave height (Hsig, Hrms and Havg).
- STEADY_TIME: starting time (t_1) for calculating mean values, significant/RMS wave height (when WaveHeight = T, output parameter below).
- T_INTV_mean: time interval ($t_2 - t_1$) for calculating mean values, significant/RMS wave height (when WaveHeight = T, output parameter below).

NOTE: Input **Starting Time** and **T_INT**values if **output means** (e.g. Umean) and/or **Wave H** are picked.

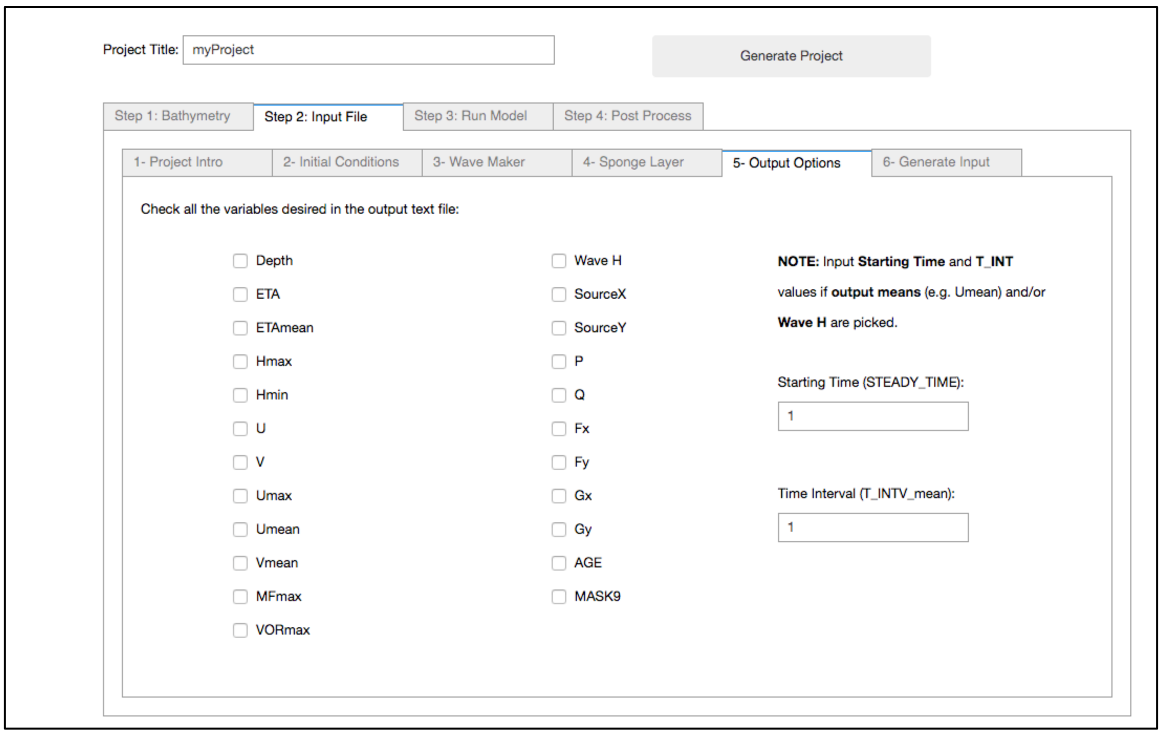

Figure 15: Input File – Output Options

Generate Input

The user verifies and generates the project driver **input.txt** file.

To do so, the user must:

- Press the "Review Input Values" button *(part a of Figure 16).* This will automatically generate the input contents on the box bellow *(part a1 of Figure 16)*.
- If the user is satisfied with the input values, press the "**Generate Input File" button** *(part b of Figure 16).*

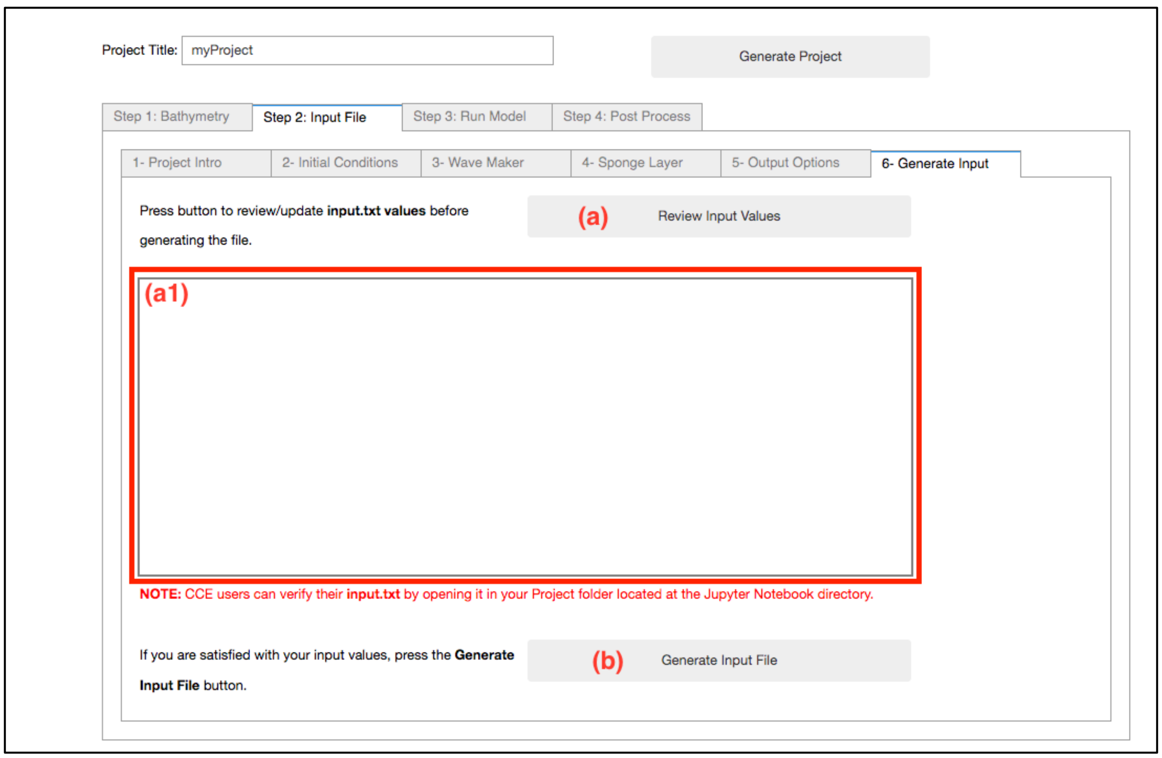

Figure 16: Input File – Generate Input

NOTE: CCE users can verify their **input.txt** by opening it in the Project folder located at the Jupyter Notebook directory.

The following is an example of an input.txt:

```
!INPUT FILE FOR FUNWAVE_TVD
! NOTE: all input parameter are capital sensitive
! --------------------TITLE-------------------------------------
! title only for log file
TITLE = myProject
! -------------------PARALLEL INFO-----------------------------
! PX,PY - processor numbers in X and Y
! NOTE: make sure consistency with mpirun -np n (px*py)
PX = 16
PY = 1
! --------------------DEPTH-------------------------------------
! Depth types, DEPTH_TYPE=DATA: from depth file
! DEPTH_TYPE=FLAT: idealized flat, need depth_flat
! DEPTH_TYPE=SLOPE: idealized slope,
! need slope,SLP starting point, Xslp
! and depth_flat
DEPTH_TYPE = DATA
! Depth file
! depth format NOD: depth at node (M1xN1), ELE: depth at ele (MxN)
! where (M1,N1)=(M+1,N+1)
DEPTH_FILE = depth.txt
DepthFormat = ELE
! -------------------PRINT---------------------------------
! PRINT*,
! result folder
RESULT_FOLDER = output/
! ------------------DIMENSION-----------------------------
! global grid dimension
Mglob = 201
Nglob = 3
! ----------------- TIME----------------------------------
! time: total computational time/ plot time / screen interval
! all in seconds
TOTAL_TIME = 200.00
PLOT_INTV = 2.00
PLOT_INTV_STATION = 0.5
SCREEN_INTV = 10.0
! -----------------GRID----------------------------------
! if use spherical grid, in decimal degrees
! cartesian grid sizes
DX = 1.00
DY = 1.00
! --------------- INITIAL UVZ ---------------------------
! INI_UVZ - initial UVZ e.g., initial deformation
! must provide three (3) files
INI_UVZ = F
! if true, input eta u and v file names
ETA_FILE = ini_z.txt
```

```
U_FILE = ini_u.txt
V_FILE = ini_v.txt
! ----------------WAVEMAKER------------------------------
! wave maker
! LEF_SOL- left boundary solitary, need AMP,DEP, LAGTIME
! INI_SOL- initial solitary wave, WKN B solution,
! need AMP, DEP, XWAVEMAKER
! INI_REC - rectangular hump, need to specify Xc,Yc and WID
! WK_REG - Wei and Kirby 1999 internal wave maker, Xc_WK,Tperiod
! AMP_WK,DEP_WK,Theta_WK, Time_ramp (factor of period)
! WK_IRR - Wei and Kirby 1999 TMA spectrum wavemaker, Xc_WK,
! DEP_WK,Time_ramp, Delta_WK, FreqPeak, FreqMin,FreqMax,
! Hmo,GammaTMA,ThetaPeak
! WK_TIME_SERIES - fft time series to get each wave component
! and then use Wei and Kirby 1999
! need input WaveCompFile (including 3 columns: per,amp,pha)
! NumWaveComp,PeakPeriod,DEP_WK,Xc_WK,Ywidth_WK
WAVEMAKER = INI_SOL
AMP = 1.00
XWAVEMAKER = 15.00
DEP = 10.50
LAGTIME = 0.00
! ---------------- PERIODIC BOUNDARY CONDITION ---------
! South-North periodic boundary condition
PERIODIC = F
! ---------------- SPONGE LAYER ------------------------
! DHI type sponge layer
! need to specify widths of four boundaries and parameters
! set width=0.0 if no sponge
! R_sponge: decay rate
! A_sponge: maximum decay rate
! e.g., sharp: R=0.85
! mild: R=0.90, A=5.0
! very mild, R=0.95, A=5.0
DIFFUSION_SPONGE = F
FRICTION_SPONGE = F
DIRECT_SPONGE = T
Csp = 0.00
CDsponge = 0.00
Sponge_west_width = 10.00
Sponge_east_width = 0.00
Sponge_south_width = 0.00
Sponge_north_width = 0.00
R_sponge = 0.85
A_sponge = 5.00
! ----------------PHYSICS-----------------------------
! parameters to control type of equations
! dispersion: all dispersive terms
! gamma1=1.0,gamma2=1.0: Fully nonlinear equations
! ----------------Friction-----------------------------
Cd = 0.0
```

🡐 bottom friction coefficient

```
!
! ----------------NUMERICS----------------------------
! time scheme: runge_kutta for all types of equations
! predictor-corrector for NSWE
! space scheme: second-order
```

```
! fourth-order
! construction: HLLC
! cfl condition: CFL
! froude number cap: FroudeCap
! CFL
CFL = 0.5
```
← Courant–Friedrichs–Lewy (CFL) number, CFL ~ 0.5.

```
!
! Froude Number Cap (to avoid jumping drop, set 1.5)
FroudeCap = 3.0
```
← cap for Froude number in velocity calculation for efficiency. The value could be 5 ~ 10.0.

```
!
! --------------WET-DRY-------------------------------
! MinDepth for wetting-drying
MinDepth=0.01
```
← minimum water depth (m) for wetting and drying scheme. Suggestion: MinDepth = 0.001 for lab scale and 0.01 for field scale.

```
!---------------breaking-----------------------------
! there are two options for breaking algorithm
! 1: shock-capturing breaking, need SWE_ETA_DEP
! 2: eddy-viscosity breaking, when VISCOSITY_BREAKING = T
! the shock-capturing breaking is invalid
! Cbrk1 and Cbrk2 are parameters defined in Kennedy et al 2000
! suggested in this model Cbrk1=0.65, Cbrk2=0.15
! WAVEMAKER_Cbrk is to avoid breaking inside wavemaker
VISCOSITY_BREAKING = T
```
← logical parameter to calculate breaking index. Note that, if VISCOSITY_BREAKING is not selected, breaking is calculated using shock wave capturing scheme. The index calculated here is based on Kennedy et al. (2000).

```
Cbrk1 = 0.65
```
← parameter C1 in Kennedy et al. (2000)

```
Cbrk2 = 0.35
```
← parameter C2 in Kennedy et al. (2000).

```
! ----------------- MIXING ---------------------------
! if use smagorinsky mixing, have to set -DMIXING in Makefile
! and set averaging time interval, T_INTV_mean, default: 20s
STEADY_TIME = 1.00
T_INTV_mean = 1.00
! -----------------OUTPUT-----------------------------
! stations
! if NumberStations>0, need input i,j in STATION_FILE
NumberStations = 0
STATIONS_FILE = stations.txt
! output variables, T=.TRUE, F = .FALSE.
DEPTH_OUT = F
U = F
V = F
ETA = T
Hmax = F
Hmin = F
MFmax = F
Umax = F
VORmax = F
Umean = F
Vmean = F
ETAmean = F
MASK = T
MASK9 = F
```

```
SourceX = F
SourceY = F
P = F
Q = F
Fx = F
Fy = F
Gx = F
Gy = F
AGE = F
WaveHeight = F
```

The highlighted parts are advance parameters that the user cannot edit in the GUI (advance features will be added in the feature), but can in the input.txt.

Step 4: Run Model

This tab *(Figure 17)* allows the user run FUNWAVE without the need of a command line/terminal. The user can also see the simulation's progress and abort it if necessary.

a) Specify the FUNWAVE Executable *(part a of Figure 17)*. This is a list of all the FUNWAVE executable the GUI can run depending on where it is located.

   NOTE: CCE users must select **FUNWAVE CCE**!!! Choosing otherwise will not work on the CCE system, thus the simulation will FAIL.

b) After selecting the executable, press **"Run FUNWAVE Model"** button *(part b of Figure 17)* to start the simulation.

c) Part c of *Figure 17* is a Model Progress load bar.

NOTE: If it seems like the simulation is stuck press the **"Interrupt Kernel"** button (part a of Figure 18) to kill the simulation. This will force the user to restart the GUI from scratch, with another project name. Contact Gabriela Salgado or Matt Malej to help the user return to the previous folder.

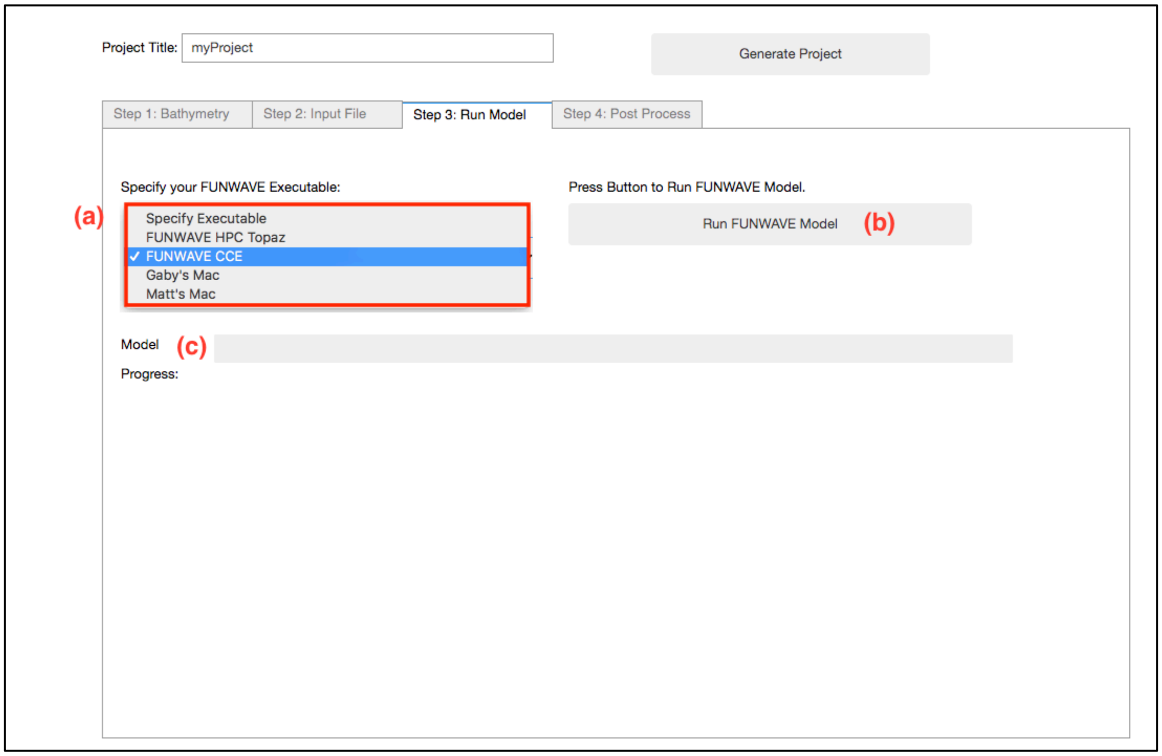

Figure 17: Run Model

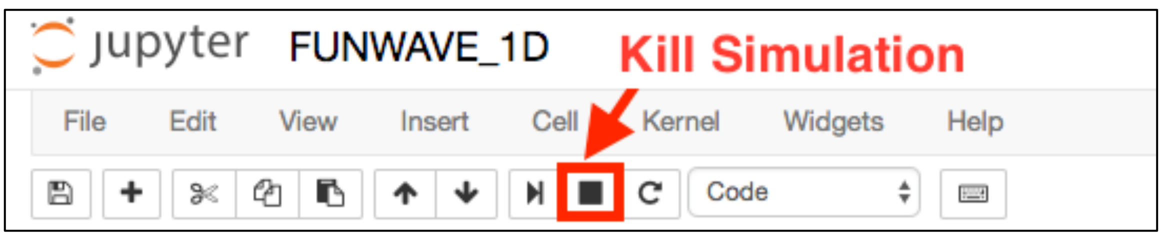

Figure 18: How to kill the simulation if needed

Step 5: Post Process

This tab gives the user the option of generating an image/plot at a specific time, or generating a video of the simulation. It is available for ETA, ETAmean, Hmax and Hmin if their checkboxes are turned on in the Input File tab (see Output Options section).

Figure 19 summarizes its workflow:

Generate Image:

a) Plot surface at a specific time.
b) Choose the plot's axis limits.
c) "Generate Plot" button.
d) If satisfied with that plot, press "Save Plot". It will be saved in the output_plots folder.

Generate Video:

e) Select Video's starting and ending time.
f) Choose the video's axis limits.
g) Choose amount of frames per second.
h) "Generate Video" button. It will be saved in the output_plots folder or the user can download it to the PC by pressing the button seen in *Figure 20*.
i) Video Progress load bar.

NOTE: To download the Project folder with all its files, **go inside the folder** in the jupyter notebook directory and press the download button at the top right (shown in *Figure 21)*. This will create a **TAR (Tape ARchive)** file that will be automatically downloaded to the machine.

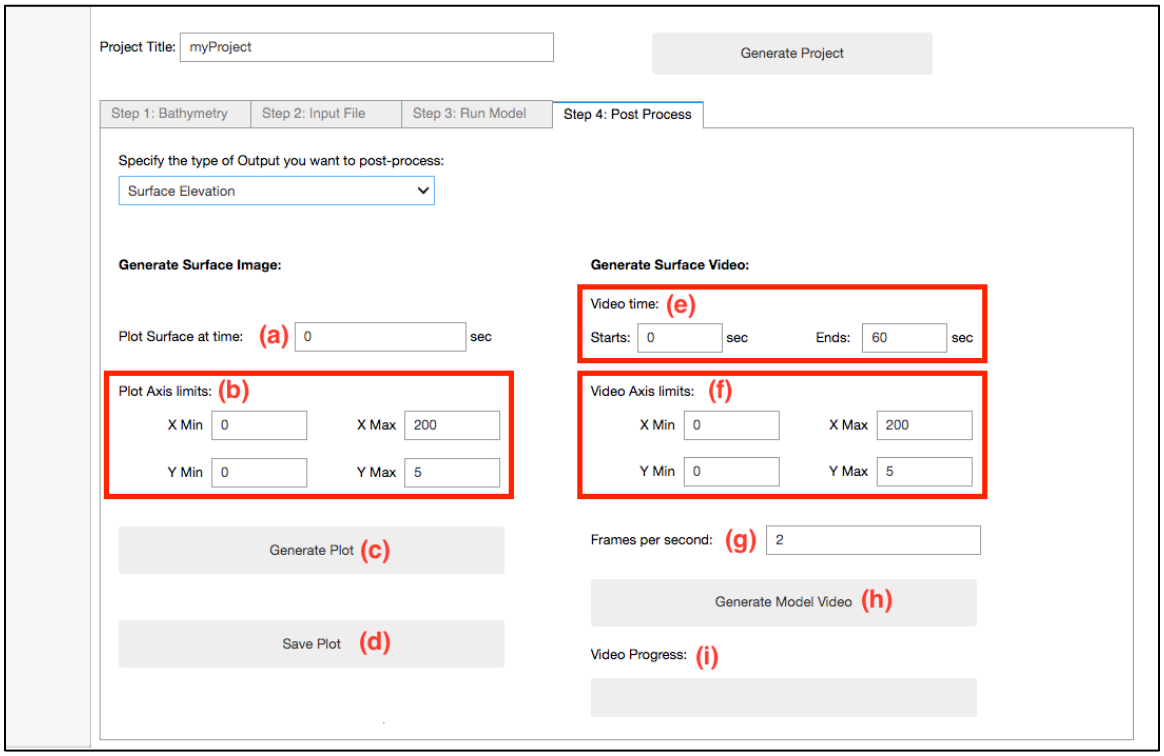

Figure 19: Post Process

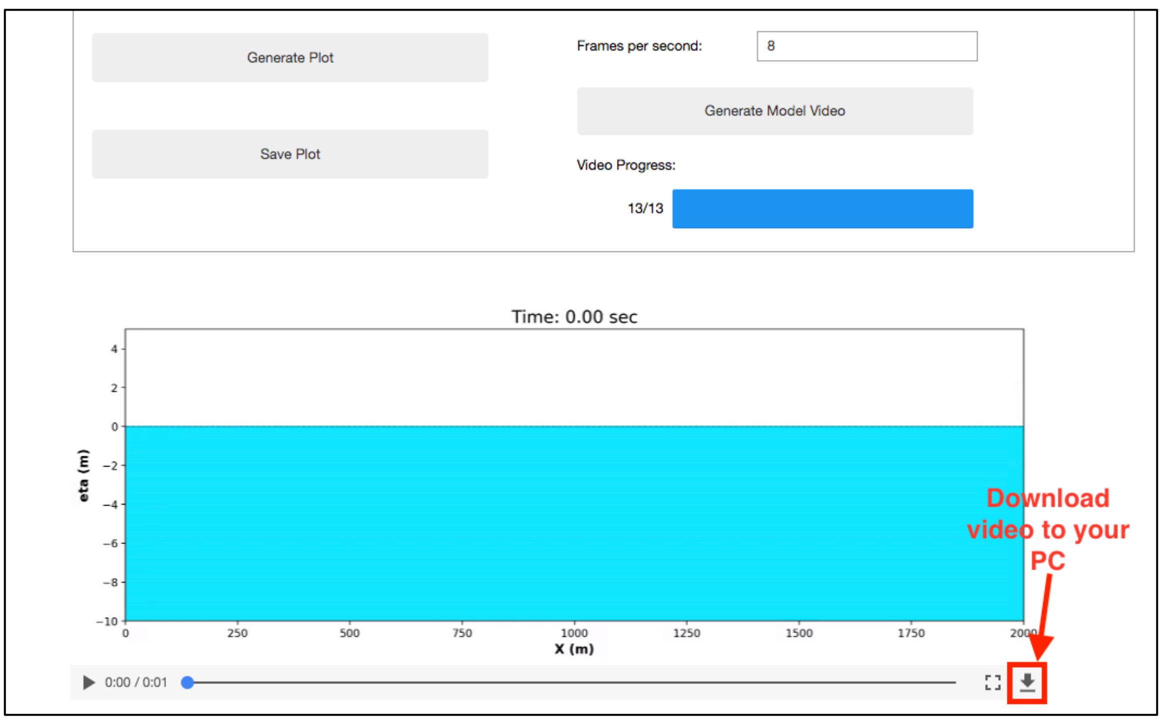

Figure 20: How to download Video to the PC

Figure 21: How to download the Project Folder