ERDC TN-24-5
June 2024

Getting Started with FUNWAVE-TVD: Troubleshooting Guidance and Recommendations

By Marissa J. Torres, Michael-Angelo Y. Lam, Levi Cass, Matt Malej, and Fengyan Shi

PURPOSE: This technical note reviews some common initialization errors when first getting started with the numerical wave model, FUNWAVE-TVD (Fully Nonlinear Wave model–Total Variation Diminishing), and provides guidance for correcting these errors. Recommendations for troubleshooting the source or cause of instabilities in an application of the model as well as recognizing the difference between physical and numerical instabilities are also outlined and discussed. In addition, a quick start troubleshooting guide is provided in the Appendix. This guidance is particularly useful for novice to intermediate users of FUNWAVE-TVD who are less familiar with the workflow of setting up the model and interpreting error output statements.

From this document, users will gain a fundamental understanding of practical troubleshooting techniques for FUNWAVE-TVD that will improve the problem-solving workflow and enhance the final product of a wave modeling study. Providing coastal planners and engineers with ease of model access and usability guidance facilitates efficient screening of design alternatives for effective decision-making under environmental uncertainty.

BACKGROUND: The FUNWAVE model is a phase-resolving nearshore wave model that solves the fully nonlinear Boussinesq equations in a hybrid manner, where the finite volume scheme is used for terms of the nonlinear shallow-water wave equations and the finite difference scheme for dispersive terms (Shi et al. 2012). The solution is applied over a rectilinear structured grid. The fully nonlinear and weakly dispersive solution allows this model to be best suited for resolving complex coastal processes within the intermediate and shallow water regimes on a local to regional scale and over time scales of minutes to hours (up to 24–48 hours). The model provides the depth-integrated variables, restricting the solution to a single plane (no vertical gaps in bathymetry or free surface). For reference throughout this document, `input file parameters`, log file parameters, and *file names* are displayed as such.

The concept of stability is essential in numerical modeling. Inherent to any finite difference scheme are errors due to numerical approximation of derivatives (i.e., truncation errors). The stability of a finite difference method is defined as when these inherent errors remain bounded in time, that is, the numerical solution converges toward the exact solution when reducing grid size and time step (e.g., consistency). Instability occurs when these errors grow unbounded in time, colloquially called “blow-ups”, and manifest as an unrealistically large value or “jump” in the solution. More importantly, numerical instability is the sole consequence of the numerical scheme and not a phenomenon of the underlying mathematical model. Most often, these instabilities are captured in a failure mode by FUNWAVE, causing the model to quit computation and print the failure condition. Other times, unrealistic values may continue to propagate in FUNWAVE without triggering a failure condition, and thus requiring further investigation.

Towards troubleshooting the cause of a model failure condition, two parameters are available in FUNWAVE to assist users—`CFL` and `FroudeCap`. The `CFL` parameter represents the Courant-Friedrichs-Lewy (CFL) number for discrete formulations of partial differential equations (PDEs). The CFL condition is a constraint on the ratio of grid spacing and time step size, such that the linearized stability condition is satisfied. A typical physical interpretation of the CFL condition is that the time step must be small enough so that at the next time step, the numerical scheme contains all the relevant physical processes at the previous time, that is, the domain of dependence, see Figure 1. For complicated nonlinear models like FUNWAVE, deriving an exact CFL condition is impractical; therefore, the baseline value of 0.5 is chosen, matching the CFL number for the related straightforward linear model. In practice, reducing the value of `CFL` is the most common tactic for stabilizing a simulation. It should be noted that reducing the `CFL` value will increase the computational time of the simulation.

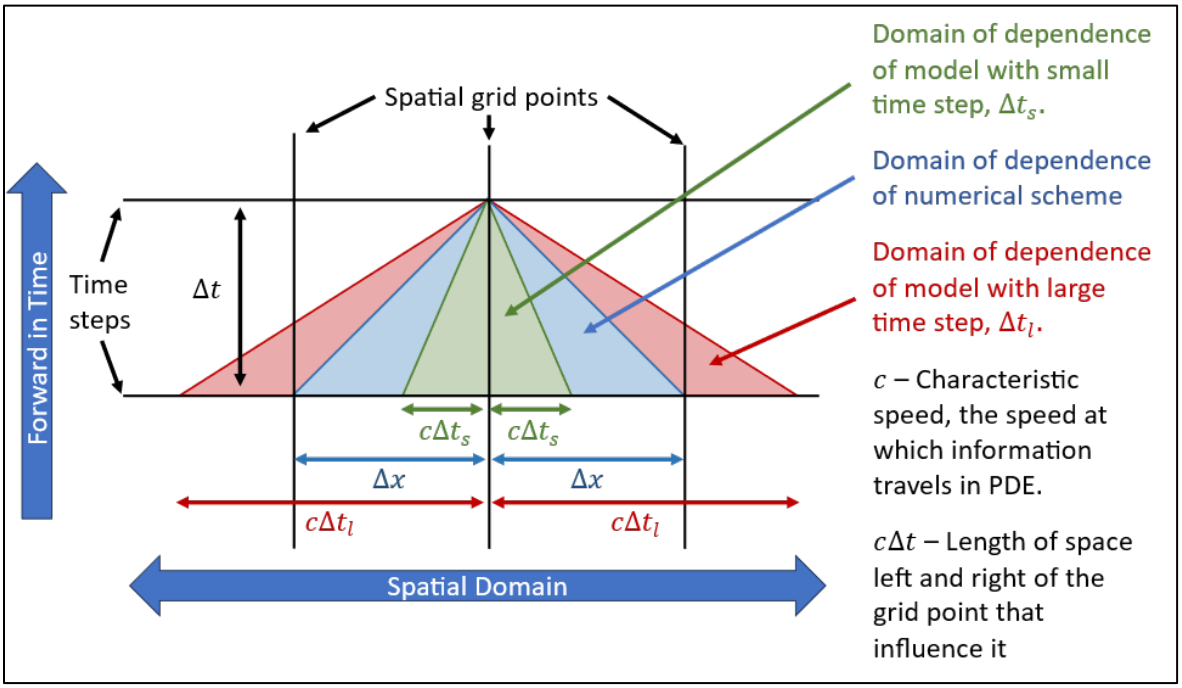

Figure 1. Example of a 1D numerical scheme.

Figure 1 provides a comparison between the domain of dependence of the numerical scheme and the domain of dependence of the mathematical model for different timesteps (Δt). The speed at which "physics" or information travels is given by the "characteristic" speed, c. The quantity is the maximum distance away that points in the past influence the point at the current timestep (i.e., the domain of dependence). The domain of dependence for the numerical scheme is the two adjacent grid points, given by the middle (*blue*) triangle. For smaller timesteps, the model's domain of dependence, the inner (*green*) triangle, is enclosed by the numerical scheme's domain of dependence; therefore, the numerical scheme captures all physical processes. Conversely, for larger timesteps, the domain of dependence, the outer (*red*) triangle, requires physical information outside the domain of dependence of the numerical scheme (i.e., the numerical scheme does not capture all physical information).

The `FroudeCap` parameter can aide in troubleshooting large changes in velocity across grid points, especially in the wetting/drying region (e.g., swash zone). This parameter caps or limits the Froude number, which is the ratio of inertial forces (flow velocity) and gravitational forces over a characteristic length scale. In this case, the flow velocity is the depth-averaged wave orbital velocity, and the characteristic length scale is the ratio of grid spacing and water depth. In the context of waves at an extremely shallow water, the gravitational forces may be considered negligible, leaving the `FroudeCap` to primarily balance the magnitude of velocity changes across grid points (i.e., flux). Reducing the `FroudeCap` value is expected to limit the propagation of large velocities through the domain and stabilize the numerical solution.

It should be noted that Boussinesq formulations are known to be less numerically stable when simulating moving shorelines on steep bathymetric slopes (Løvholt et al. 2013) or over rapid changes in bathymetry. The moving shoreline, or wetting/drying condition, is encompassed in the implementation of the hybrid finite-volume—finite-difference scheme in the latest version of FUNWAVE. The methodology projects a continuous thin layer of fluid across all points in the domain. Areas where this layer is below a threshold value are considered numerically dry. Stability issues can occur when small pockets of water exceed the threshold but fail to propagate further, such as water getting trapped behind a levee. In the solution, this additional fluid mass is not considered dry, nor is it propagating, which can destabilize the volume flux across grid points. Recommendations are provided in this document to recognize and correct instabilities of this type in the domain.

Getting started with a numerical model of any type can be challenging. For FUNWAVE-TVD, additional guidance, background material, and use-case examples are available for new and experienced users on the comprehensive Wiki (https://fengyanshi.github.io/build/html/index.html; Shi et al. 2019). The Wiki is updated continuously as new developments are added to the model and user feedback is received. Recent additions include a checklist when setting up a simulation to reduce the probability of initialization errors and ensure a successful first run.

TROUBLE SHOOTING RECOMMENDATIONS: Knowing what to do when a simulation fails, becomes unstable, or the results are not what was expected, comes from experience in running the model and understanding some of the fundamental physics and limitations of the model. This is true for troubleshooting all types of numerical models. Troubleshooting is an iterative, sometimes tedious, process where new information is learned with each iteration until the source of the error or instability is identifiable. Correcting the instability can be equally repetitive since optimal conditions for a specific simulation will vary by application. The recommendations listed herein will support the initial phases of troubleshooting to identify the source and assist in correcting instability in a FUNWAVE simulation.

The run time status of a FUNWAVE simulation is printed in the log file (*LOG.txt*) that is automatically created when the model initializes. In this file, a summary of the input parameters is followed by a summary of output statistics printed at the model progression frequency (SCREEN_INTV) specified in the control file (*input.txt*). It is recommended to check this file for every simulation to see whether the simulation completed successfully (normal termination), failed to initialize, or became unstable. When the model becomes unstable after the initial print interval at time 0, a BlowUp Time and MaxAbsEta line is printed to the log file and error output files are printed to the working directory (e.g., *eta_99999*, *mask_99999*), if the plot interval time (PLOT_INTV) was

reached. If the model fails to initialize properly, a forrtl: severe line is printed with brief details of the cause. Some common initialization errors are presented later in this document.

A good first step in troubleshooting a failed FUNWAVE simulation is determining whether it is the physics or numerics causing the instability. Fundamental issues in the physics (i.e., model setup) of the simulation may be identified by looking closer at the values in the statistics block in the log file, as well as checking that the input wave and water level conditions are within the valid range of the model (Torres et al. 2022). Sources of numerical instability may be identified and are sometimes corrected by adjusting the `CFL` and `FroudeCap` parameters in the input file.

Interpreting LOG file statistics. An example of the statistics block is shown in Figure 2. The parameters listed in the statistics block can be used to identify potential sources of instability in the model by tracking the magnitude of these values over time. There are two primary failure modes that trigger a stop condition in the model, (1) when the computed wave height exceeds 100 times the water depth (MaxAbsEta > 100 h) resulting in the BlowUp Time error and (2) when a value(s) becomes so large that the number cannot be stored in memory resulting in a NaN in statistics error. This latter error is meant to be a catch-all failure mode.

Of the parameters in the statistics block, the MassVolume, MaxEta and MinEta, Max U and Max V, and Froude values can elucidate possible causes of instability and when they start occurring in time:

(a)

```
----------------- STATISTICS ----------------
 TIME        DT
 0.1001E+02  0.1690E-01
 MassVolume  Energy      MaxEta      MinEta      Max U       Max V
-0.3817E+00  0.3041E+04  0.1122E+00 -0.1571E+00  0.1900E+00  0.0000E+00
 MaxTotalU   PhaseS      Froude      WetDryMass
 0.1900E+00  0.4552E+01  0.4175E-01  0.0000E+00
 TideWest    TideEast      TideSouth    TideNorth
 0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
```

(b)

```
---------------- STATISTICS ---------------
 TIME        DT
 0.4200E+03  0.7390E-03
 MassVolume  Energy      MaxEta      MinEta      Max U       Max V
-0.2461E+02  0.3475E+06  0.1054E+02 -0.1061E+02  0.1409E+02  0.1408E+02
 MaxTotalU   PhaseS      Froude      WetDryMass
 0.1656E+02  0.1104E+02  0.1500E+01  0.0000E+00
  PRINTING FILE NO.   421  TIME/TOTAL:      420.000 /    1800.000
  PRINTING FILE NO.   422  TIME/TOTAL:      421.001 /    1800.000
  PRINTING FILE NO.   423  TIME/TOTAL:      422.000 /    1800.000
 ========================================
 BlowUp Time, MaxAbsEta=   422.144097027892        14432.4840464098
 ========================================
  PRINTING FILE NO. 99999  TIME/TOTAL:      422.144 /    1800.000
MPT ERROR: Rank 34(g:34) is aborting with error code 9.
       Process ID: 13077, Host: r5i3n16, Program: /p/home/mtorres/FUNWAVE-TVD/bin/funwave_central
       MPT Version: HPE MPT 2.21  11/28/19 04:21:40
```

Figure 2. Examples of a statistics block printed to the *LOG.txt* file of a FUNWAVE simulation for a successful (*a*) and an unsuccessful simulation (*b*) that triggered the BlowUp Time failure mode.

MaxEta and MinEta. The total maximum and minimum water level in meters, respectively, over the entire domain. If MinEta values steadily decrease below the input wave height or reach the water

depth, this may indicate or be a precursor to an instability. It should be noted that MaxEta represents water depth and topography due to the wetting/drying formulation in the model. Therefore, if MaxEta seems unusually large for the input wave height, it may be reporting water above an emergent feature (e.g., MaxEta = 5 m with Hmo = 1 m and topographic features reaching 5 m). More information on the wetting/drying condition in FUNWAVE is presented later in this document.

Max U and Max V. Examining peak values in the wave-induced currents or circulation may be helpful in troubleshooting. The Max U and Max V parameters, for maximum orbital velocities (m/s) in *x* and *y*, respectively, are provided in the statistics block. Unrealistic (e.g., very large) values for Max U and Max V can indicate a fundamental error in the setup of the simulation and warrant further investigation into potential problem areas in the domain.

Froude. The Froude parameter is also a good indicator of potential instabilities. Should the Froude value reach or exceed 1.5 or 3 (recommended `FroudeCap` values in *input.txt*) then this indicates likely instabilities. Large values of Max U and Max V may not always trigger a failure mode in FUNWAVE. Therefore, if the global or station outputs from the model seem inaccurate, check the statistics block to determine whether or when unrealistic values are present.

MassVolume. If water appears to be draining from the domain over the course of the simulation, there is likely an imbalance in the mass conservation formulation related to wetting/drying. In this case, check the MassVolume parameter in the statistics block to identify how much mass was lost leading up to failure and when the decrease began. A significant decrease in MassVolume (>10%) warrants closer inspection of the bathymetry, particularly at the boundaries. Gaps between bathymetry/topography and boundary edges may be a source of imbalance or instability. If further troubleshooting is needed, users may consider recompiling FUNWAVE with the `-DCHECK_MASS_CONSERVATION` flag uncommented, turning on the mass conservation formulation specific to wetting/drying. However, turning on this flag will increase the computational time as the model will check the conservation balance every time step.

Steps towards identifying the source. If the potential source of the instability is not identifiable from the log file statistics and input wave conditions, there are a few additional courses of action to take to continue troubleshooting the error. To further distinguish physical versus numerical sources of instability, try lowering the `CFL` or `FroudeCap` values in the input file to 0.25 and 1.5, respectively, and rerunning the model. Be advised that by lowering the `CFL` value the simulation will take longer to run on the order of the ratio of the old `CFL` to new `CFL` number (e.g., halving the CFL doubles the run time). If the simulation fails again around the same point in simulation time (BlowUp Time), try lowering the `CFL` value to its minimum value of 0.1. Should the simulation fail again, this usually means there is a more fundamental issue in the physics of the model. Otherwise, if the simulation is successful with these values lowered, the issue was due to numerical stability.

Another method of troubleshooting is to visualize the surface elevation output leading up to the time of the instability. Plot the *eta_** output files, including *eta_99999*, with bathymetry to learn where in the domain the instability is occurring. Figure 3 shows an example of an *eta_99999* file plotted over the domain bathymetry to identify the location and potential source of the instability that caused this simulation to terminate unsuccessfully. The location could be a single point on a

sloping surface or occur at the wavemaker, for example. Review the FUNWAVE Wiki for assistance reading output files types (ASCII or BINARY) in MATLAB and Python.

If the cause or source of the instability is unclear in the visualizations, consider rerunning the simulation with a smaller plot interval (`PLOT_INTV`) starting near the time of failure (`PLOT_START_TIME`)—to keep the output file storage space from growing too large—and replotting the surface. A smaller time interval between images will help visualize quick changes in the wave dynamics and reduce the need to infer what is happening between images. Creating a GIF or video of the surface elevation or velocity vectors leading up to an instability can also help reveal the dynamics of the problem more clearly. Users can find sample codes (Python and MATLAB) in the *tools/* folder of the FUNWAVE-TVD repository to help with visualization.

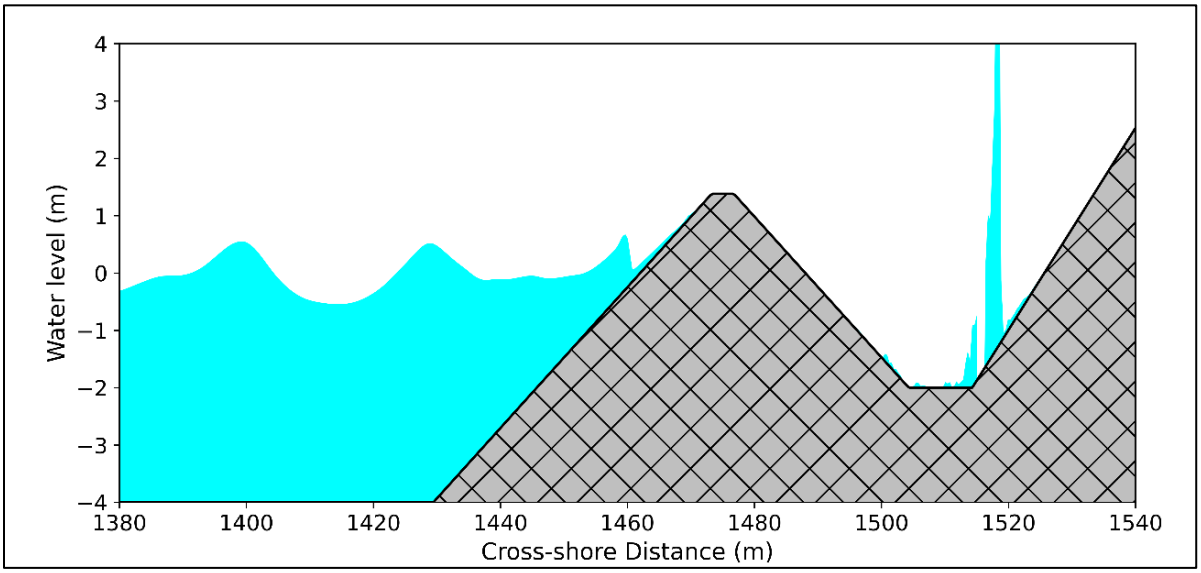

Figure 3. An example image of an *eta_99999* file for an unstable simulation zoomed in to the location of instability. Surface elevation is shown in *teal*, and bathymetry, is shown in *gray*. The source of instability is visible behind the initial levee slope where a column of extreme water level occurred 422 seconds into an 1,800-second simulation. This result is an example of a wetting/drying instability on an emergent slope, where water was building up on a dry surface while not propagating.

Potential sources of instability. Each application of FUNWAVE is unique, and as such, so will be any instabilities in the domain. However, there are a few common sources of instabilities that can occur in the early stages of learning the model and building a stable grid. Again, understanding the cause of these instabilities comes from understanding how the physics and numerics of the model interact. A greater understanding of this process is developed the more time that is spent working with the model. Two of the most common sources of instability are the bathymetry and the wetting/drying process.

Bathymetry. Limiting steep or rapid changes in bathymetry can help reduce instability. FUNWAVE provides a built-in localized bathymetry smoothing function that can be turned on by setting `BATHY_CORRECTION = T` in the input file. The localized smoothing is applied to depths below `SmoothBelowDepth`, with a default value of 0.0; and to slopes greater than `SlopeCap`,

with a default value of 1.0. Note that negative values for `SmoothBelowDepth` denote above still water level (i.e., land). The built-in localized smoother is not guaranteed to smooth the bathymetry and may still leave jumps, at which point a global smoothing technique may be needed. A discussion of global smoothing methods is not within the scope of this document; however, we note that techniques, such as applying a Gaussian filter/smoother to the bathymetry as a preprocessing step, have proved sufficient in practice. If instability still occurs after global smoothing, then either manual modification of the bathymetry is required, or closer scrutiny of the input bathymetry is needed to identify potential issues (e.g., incorrectly merged bathymetry sources, leading to stair-stepping of values). Users can find sample filtering tools in the FUNWAVE-TVD repository.

Wetting and drying. As mentioned previously, the model initializes with a thin layer of fluid everywhere in the domain, including areas above the still water line. The model numerically dries grid points that are below a specified depth threshold of thickness `MinDepth` (meters) in the input file. For any area of the topobathy with elevations below the still water line, these sections will be submerged up to the still water level. It should be noted that once a wave travels over the initial `MinDepth`-thick surface layer, the layer will no longer appear in the surface elevation output file though the `MinDepth` criterion will still be in effect.

A wetting-drying instability can trigger either failure mode (MaxAbsEta or NaN in statistics) since an unsteady balance of momentum flux can generate unrealistically large values at grid points. Sections of emergent topography experiencing intermittent or limited wetting events are susceptible to this instability. For example, the "dry" bucket behind the levee depicted in Figure 3 experienced intermittent inundation due to overtopping, causing a seich in the bucket that accelerated a MaxAbsEta failure mode. Increasing surface friction in problem areas may help dampen wave energy and prevent instability.

In addition to visualizing the surface elevation, users can leverage the *mask_** files to further investigate this specific instability type. The *mask_** files are global output files that distinguish between wet and dry points in the domain with 1.0s and 0.0s, respectively, and are printed at the same frequency as the *eta_** files (`PLOT_INTV`). If there are certain grid points that alternate between wet and dry every time step unexpectedly or uncharacteristically, then the wetting/drying condition is imbalanced.

In this case, the following three courses of action may be taken: (1) adjust the thickness of the fluid layer (`MinDepth`), (2) adjust the elevation of the bathymetry/topography in the problem area, and (3) adjust the elevation of the starting water level in the domain. Users are encouraged to adjust the `MinDepth` parameter in a few simulations, increasing it above and below the default value (0.1), and visualizing the results. This process will help develop an understanding of how the parameter is affecting wave propagation. Increasing the value will decrease the sensitivity of the wetting/drying condition, stabilizing potential momentum imbalances at the edge of the shoreline or on or behind emergent features. It should be noted that if wave runup is of interest, reducing the `MinDepth` yields a more accurate solution but may be less stable, particularly on steep slopes. The recommended minimum value is 0.01 for field-scale applications.

Otherwise, if the problem area is significant to the study and adjusting the `MinDepth` value is not a viable solution, consider applying an initial condition to the water level in the domain using the hot start conditions in FUNWAVE (`INI_UVZ`, `INITIAL_ETA`). The hot start condition allows users to target initial water level thickness in specific areas across the domain separate from the `MinDepth` criterion. Details on how to setup initial conditions are provided on the FUNWAVE Wiki.

NaN in statistics. This type of error is less common than the previous errors, though is worth noting. As mentioned previously, the NaN in statistics error is a catch-all failure mode to prevent the continued propagation of wildly unrealistic values and inaccurate results (Figure 4). If the value of any parameter tracked in the statistics block exceeds internal memory thresholds, a NaN value is returned. This error can occur at any point during a simulation. While there is no clear root cause of this error type for all applications of FUNWAVE, the NaN in statistics error could suggest an instability in the numerics. Users are recommended to lower the `CFL` number (if not already at the minimum value of 0.1) and rerun the model. Should this error continue to occur, consider visualizing the output files to investigate the cause.

```
---------------- STATISTICS ---------------
 TIME        DT
 0.0000E+00  0.0000E+00
 MassVolume  Energy      MaxEta      MinEta      Max U       Max V
      NaN         NaN  0.1000E+01  0.0000E+00  0.1265E+01  0.0000E+00
 MaxTotalU   PhaseS      Froude      WetDryMass
 0.1265E+01  0.7672E+01  0.1649E+00  0.0000E+00
NaN detected in statistics, stopping simulation!
```

Figure 4. An example of an error output in *LOG.txt* where NaNs are detected in the statistics of the simulation. This response suggests a fundamental error in the setup of the model domain or control file.

Vessel Module troubleshooting considerations. Vessel-generated wakes/waves can be modeled in FUNWAVE using the optional vessel module; however, using the vessel module necessitates considering additional troubleshooting techniques. A discussion of the vessel module is not within the scope of this document, and we refer the reader to the FUNWAVE Wiki or to the technical report (Lam et al. 2022) for further details. To troubleshoot vessel simulations, there are two main parts of the vessel file to consider, the vessel path and the model parameters.

Vessel path. The vessel path is time series data of the vessel position with the speed and heading inferred from subsequent positions. If the vessel's path is not smooth enough, this could result in large jumps in the vessel's inferred speed or heading. These instantaneous changes can cause the vessel to behave like a paddle-type wave maker, generating unrealistic waves. It is recommended that a sufficient number of points are used to resolve turns and large changes in speed accurately. A simple technique to employ would be using smooth splines to fit the x and y vessel position data and then interpolation to resolve rapid changes accurately. The interested reader is referred to Lam et al. (2022), where a different fitting procedure was used to generate FUNWAVE vessel paths fitted to Automatic Identification System data.

Another potential issue with vessel paths, particularly for large vessels, is the potential surge wave generated by the instantaneous acceleration of the vessel from rest to its initial speed. Slowly accelerating the vessel from zero speed may eliminate or reduce the size of the surge wave. Note that a vessel path cannot have zero speed, that is, repeated positions in the path, as this would orient the vessel due East and cause an instant turn if the vessel was not already heading due East.

Model parameters. There are four types of vessel wake generation in FUNWAVE and further split into two categories, pressure source and flux source. A detailed description of the four types is not within the scope of this document but note that the pressure source methods are characterized by shape parameters. Instability can occur if these shape parameters result in a vessel hull with a steep slope, particularly a fast-moving vessel with a steep slope at the bow. Hull slope is limited by the resolution of the domain similar to steep sloping bathymetries.

Low under-keel clearance. For large vessels with deep drafts, instability can occur if there is low or no clearance between the bottom of the vessel and bathymetry. FUNWAVE includes a deep draft module to help stabilize this low under keel-clearance instability. The reader is referred to the technical report Malej and Shi (2021) for information on using the deep draft module as well as examples of how this instability manifests in model outputs.

COMMON INITIALIZATION ERRORS: Separate from failure modes due to instability are initialization errors resulting from incorrect definitions of parameters in the input file, or improper setup of other input file types (e.g., depth, friction, obstacles, and initial conditions). While not comprehensive, this section outlines a few common initialization errors, their cause, and recommendations for correction. A more comprehensive list is available on the FUNWAVE Wiki under the "Simulation Checklist".

File not found—The most common mistake made by both novice and experienced users of the model is misnaming one or more input files referenced in the control file *input.txt*. As additional input files, such as bathymetry/topography (*depth.txt*) and bottom friction (*friction.txt*), are added to a simulation, these file names must be referenced in the control file. If the file names are mismatched between the actual file name and what is written in the control file, the model cannot find the specified file and will fail to initialize. Similarly, if the file is not located in the same directory as the control file and a relative path is not provided, the model cannot find it.

The file not found error will appear at the end of the log file with the name of the "missing" file printed. Users can correct this error by updating the name or relative path in the control file and rerunning the simulation. Be advised that the name of the depth, friction, or other global input file can be unique to the simulation (e.g., *friction_testcase_cd002.txt*).

End-of-file—An end-of-file error usually means that an input file is not the correct size (e.g., depth, stations, or friction), meaning it contains less data than was expected. The example in Figure 5 shows an end-of-file return from the model and lists the path and filename of the problem file (*bottom_friction.txt*). A file can be the incorrect size if the grid dimensions are reversed or if there are too few or too many dimensions.

If this error occurs, it is recommended to check that the file is indeed the correct size. For depth, friction, breakwater, or similar files that are meant to be the size of the model domain (global input

files), ensure that the number of rows is equal to `Nglob` (number of grid points in *y*) and the number of columns is equal to `Mglob` (number of grid points in *x*). For station files, ensure that the number of rows is equal to `NumberStations` in the control file *input.txt* and the number of columns is equal to two. As a reminder, if the same end-of-file error occurs after verifying the file is the correct size, verify the file name is correct.

```
forrtl: severe (24): end-of-file during read, unit 1, file /p/work/lcass/projects/friction_sensitivity/
flat_bath_fric0002/input_files_o4939718/bottom_friction.txt
Image              PC                Routine            Line        Source
funwave-central    000000002010DC38  for__io_return        Unknown  Unknown
funwave-central    000000002013780A  for_read_seq_lis      Unknown  Unknown
funwave-central    0000000020096A62  Unknown               Unknown  Unknown
funwave-central    00000000200B3FE4  Unknown               Unknown  Unknown
funwave-central    000000002001CC24  Unknown               Unknown  Unknown
funwave-central    000000002000B792  Unknown               Unknown  Unknown
libc-2.31.so       00001544E135229D  __libc_start_main     Unknown  Unknown
funwave-central    000000002000B6AA  Unknown               Unknown  Unknown
[NID 01695] 2023-06-09 08:54:58 Apid 42695051: initiated application termination
Application 42695051 exit codes: 24
Application 42695051 resources: utime ~0s, stime ~0s, Rss ~14848, inblocks ~1544, outblocks ~24
```

Figure 5. An example output from the standard error/output file from the FUNWAVE model specifying that the "end-of-file" error occurred during read (or initialization) where the path and name of the file is listed.

Number of processors—When working in a high-performance computing (HPC) or cluster environment, the FUNWAVE simulation can be divided across dozens up to hundreds or thousands of processors and solved simultaneously for fast computation. One of the requirements of working in these environments is to choose an appropriate number of processors to use. Each HPC or cluster has different specifications that should be reviewed prior to submitting jobs. For computational efficiency, the general rule of thumb is to assign the number of requested processors to be a multiple of the total number of processors (cores) per node. For example, if the HPC has 44 processors (cores) per node available, the product of `PX` and `PY` in the input file should equal either 44 or a multiple of 44 (Figure 6).

```
lcass@onyx08:~/projects/friction_sensitivity/monochromatic/mono_dx05_flat_bath05_fric0000/input_files> qsub run_test.
pbs
qsub: Error: Job ncpus = 44 and must be divisible by mpiprocs. Your mpiprocs = 8
             Recommended mpiprocs values are: [ 1, 2, 4, 11, 22, 44 ]
```

Figure 6. An example error output from the supercomputer specifying that the number of processors selected is not a multiple of the total number of processors available.

MISCELLANEOUS ERRORS: In some cases, an unexpected error may occur that is unrelated to the setup or execution of the model. One such error is the forrtl: severe segmentation fault failure mode (Figure 7), which may occur at any point in the simulation. Though very rare, it is possible to receive this error when using a beta version of FUNWAVE. Since FUNWAVE is in active development to incorporate new features, expand its existing capabilities, and improve performance and efficiency, there may be times where using a beta version is preferred or required. Beta versions of code may have miscommunications across modules and functions in the foundational code that require further testing and correction. The segmentation fault error is an example of a FORTRAN programming language specific error suggesting that a more fundamental issue is present.

```
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image              PC                Routine            Line        Source
funwave_central    000000002011449A  for__signal_handl     Unknown  Unknown
libpthread-2.31.s  0000151F391F8910  Unknown               Unknown  Unknown
funwave_central    0000000020103E1B  Unknown               Unknown  Unknown
funwave_central    000000002001D790  Unknown               Unknown  Unknown
funwave_central    000000002000B7A2  Unknown               Unknown  Unknown
libc-2.31.so       0000151F3902024D  __libc_start_main     Unknown  Unknown
funwave_central    000000002000B6BA  Unknown               Unknown  Unknown
```

Figure 7. An example error output in the log file (*LOG.txt*) of the FORTRAN-specific "segmentation fault" failure mode.

To work around this error, consider cloning and compiling the latest stable version, or official release, of FUNWAVE from GitHub. Official version releases of the model have been fully tested across a series of simple cases and benchmark applications. Be advised that these version releases may not have the latest features and functionality available. If those features are of interest, consider moving forward cautiously with a latest beta version of FUNWAVE. US Army Corps of Engineers District users are recommended to use the latest official release.

Users of all levels are encouraged to join the FUNWAVE Users Group listserv to seek guidance and recommendations from the community of FUNWAVE modelers, as well as inform developers of foundational errors in the model such as the example shown in Figure 7. More information about the listserv is available on the home page of the FUNWAVE Wiki.

SUMMARY: Troubleshooting is an iterative process to gather new information until the source of the error or instability is identifiable. Correcting an error or instability is equally as repetitive. The guidance and recommendations presented in this document provide users with a foundation for troubleshooting errors and identifying and correcting sources of instability in the Boussinesq-type wave model, FUNWAVE-TVD. For ease of access, a quick start troubleshooting guide is provided in the Appendix of this document. Though the list of error examples is nonexhaustive, the recommendations for troubleshooting are applicable to most sources of error in the model. Users are encouraged to review the FUNWAVE Wiki for additional information related to model development, setup, parameter descriptions, and more.

ADDITIONAL INFORMATION: This technical note is a product of the Inlet Engineering Tools work unit of the Coastal Inlets Research Program being conducted jointly at the US Army Engineer Research and Development Center, Cold Regions Research and Engineering Laboratory and Coastal and Hydraulics Laboratory. Questions about this technical note can be addressed to Ms. Marissa J. Torres (Marissa.J.Torres@usace.army.mil). For information about the Coastal Inlets Research Program, please contact the program manager, Ms. Tanya Beck (Tanya.M.Beck@usace.army.mil). This technical note should be cited as follows:

> Torres, M. J., M.-A. Y. Lam, L. Cass, M. Malej, and F. Shi. 2024. *Getting Started with FUNWAVE-TVD: Troubleshooting Guidance and Recommendations*. ERDC TN 24-5. Hanover, NH: US Army Engineer Research and Development Center. http://dx.doi.org/10.21079/11681/48631.

REFERENCES

Lam, M.-A., M. Malej, and F. Shi. 2022. *Operational Modeling of Large Container-Type Vessel-Generated Waves with Related Erosion and Scour Effects*. ERDC/CHL TR-22-20. Vicksburg, MS: US Army Engineer Research and Development Center. http://dx.doi.org/10.21079/11681/45483.

Løvholt, F., P. Lynett, and G. Pederson 2013. "Simulation Run-Up on Steep Slopes with Operational Boussinesq Models; Capabilities, Spurious Effects and Instabilities." *Nonlinear Processes in Geophysics* 20 (3): 379–395. https://doi.org/10.5194/npg-20-379-2013.

Malej, M., and F. Shi. 2021. *Suppressing the Pressure-Source Instability in Modeling Deep-Draft Vessels with Low Under-Keel Clearance in FUNWAVE-TVD*. ERDC/CHL CHETN-IX-55. Vicksburg, MS: US Army Engineer Research and Development Center. http://dx.doi.org/10.21079/11681/40639.

Shi, F., J. C. Harris, M. Malej, Y.-K. Choi, Y. Yuan, and M. J. Torres. 2019. "FUNWAVE-TVD." *FUNWAVE*. https://fengyanshi.github.io/build/html/index.html.

Shi, F., J. T. Kirby, J. C. Harris, J. D. Geiman, and S. T. Grilli. 2012. "A High-Order Adaptive Time-Stepping TVD Solver for Boussinesq Modeling of Breaking Waves and Coastal Inundation. *Ocean Modelling* 43–44: 36–51. https://doi.org/10.1016/J.OCEMOD.2011.12.004.

Torres, M. J., M.-A. Lam, and M. Malej. 2022. *Practical Guidance for Numerical Modeling in FUNWAVE-TVD*. ERDC TN-22-1. Hanover, NH: US Army Engineer Research and Development Center–Cold Regions Research and Engineering Laboratory. http://dx.doi.org/10.21079/11681/45641.

APPENDIX: QUICK TROUBLESHOOTING GUIDE

General steps. Listed below are a basic order of steps to follow when attempting to stabilize a simulation. Re-run the simulation with the recommended changes at each step. If still unstable, move to the next step, rerun, and repeat as needed:

1. Review input wave, water level, and spatial resolution conditions per Torres et al. (2022)
2. Reduce `CFL` to 0.25 and `FroudeCap` to 1.5
3. Reduce `CFL` to 0.1
4. Use built-in localized topobathy smoothing
    a. `BATHY_CORRECTION = T`
    b. `SmoothBelowDepth = -1.0`
    c. `SlopeCap = 1.0`
5. Manually smooth data using some global technique, for example, Gaussian filter/smoother (see *FUNWAVE-TVD/tools/filter* in GitHub repository)
6. Manually adjust topobathy data in the problem region
7. Submit your problem to the FUNWAVE User Group listserv. Signup is available on the Wiki.

Visualizing problem area. If identifying the instability from visualizing the **_99999* files is challenging, follow the below steps:

1. Get instability time from *LOG.txt* (Figure 2. BlowUp Time).
2. Set `PLOT_START_TIME` to just before instability time. Use last valid outputs as a guide.
3. Reduce `PLOT_INTV`, for example, 10–30 plots between `PLOT_START_TIME` and instability time. Visualize output.
4. Repeat if necessary, further adjusting `PLOT_START_TIME` and `PLOT_INTV`.