ERDC/CHL CHETN-II-59
August 2020

US Army Corps of Engineers®

On the Use of CSHORE for Beach-fx

By Bradley Johnson and Dylan Sanderson

PURPOSE: This Coastal and Hydraulics Engineering Technical Note (CHETN) presents and documents a series of MATLAB and Python scripts that prepare, run, and process CSHORE files for use in Beach-fx. CSHORE (Johnson et al. 2012) is a one-dimensional cross-shore profile evolution model that predicts storm-induced beach profile change. Beach-fx (Gravens et al. 2007) is an engineering-economic model that computes the evolution and economic benefits associated with beach renourishment projects and requires profile erosion estimates. Historically, the cross-shore profile response model SBEACH (Larson et al. 1990) has been implemented in the creation of Beach-fx studies although the Beach-fx model was designed to allow alternative profile response models, such as CSHORE, to be used.

INTRODUCTION: Beach-fx is an engineering-economic planning model that computes the evolution and economic benefits associated with beach renourishment projects. Beach-fx studies are divided into project reaches (Figure 1) that are morphologically similar and are represented by idealized profiles composed of key morphologic features (dune height, dune width, berm width, etc.). The representative profiles evolve across time due to randomly selected storm events, background erosion, and planned renourishment. Although profile responses to storm events are a key component of Beach-fx, no response computations are actually performed at runtime. Rather, a relational database (Storm Response Database [SRD]) is accessed that contains a matrix of pre and post-profile dimensions for each storm event. The SRD acts as a *lookup table* for the Beach-fx computational kernel and is populated from an external cross-shore response model. Because no profile response computations are performed at runtime, all of the expected profiles that will be encountered over a project's lifecycle must be pre-computed to populate the SRD. This profile space is created by varying the dune height, dune width, and berm width between a maximum profile (typically the largest renourishment template that will be tested), and a minimum profile (typically dune height, dune width, and berm width dimensions of 0). An example of a typical Beach-fx profile parameter space is shown in Figure 2.

Figure 1. Example of Beach-fx Study

Historically, the SRD has been populated using the profile response model SBEACH. The SBEACH Data Generator

Tool was developed to assist a Beach-fx user in setting up and running all of the SBEACH computations that are necessary to populate the SRD. Additionally, features have been implemented within the SBEACH graphical user interface to allow model results to be exported to the appropriate format for Beach-fx.

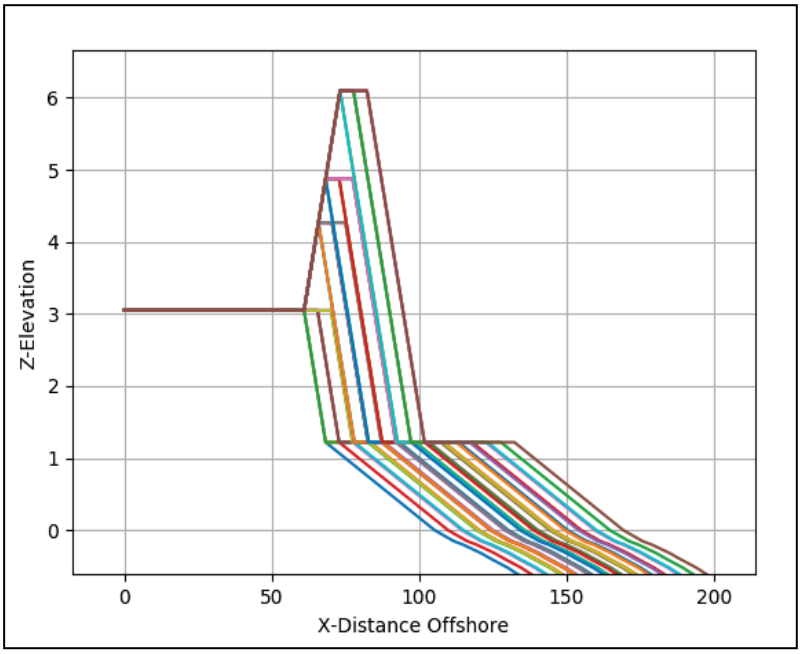

Figure 2. Example of Beach-fx profile parameter space. Variations in dune height, dune width, and berm width.

The computational architecture of Beach-fx allows alternative shoreline response models to be used, provided that the model output is properly formatted in ASCII text files. Because the number of profile response computations associated with a Beach-fx study may be large, manually preparing, running, and processing all of the necessary files can be a tedious and cumbersome experience. While this process could be done manually, a series of tools to assist in preparing the necessary files that populate the SRD is required in most cases.

As the general understanding of coastal processes and beach profile responses to storm events increases, new models, and tools become available. One such model, CSHORE, a robust and accurate code that predicts nearshore hydrodynamics and beach evolution, has been under development for the past several years. Designed for efficient application to cross-shore transects, the computation time to predict the impact of a storm of several days duration is less than 30 sec[*]. The combined wave and current model operates under the assumption of longshore uniformity and includes the effects of a wave roller and quadratic bottom shear stress. The numerical integration of the depth-averaged energy, momentum, and continuity equations manifests with predictions of wave height, water level, wave-induced steady currents, and wave runup excursion. The development of new and physically defensible sediment transport algorithms for a nearshore breaking wave environment has been the focus of the most recent research efforts. The model accounts for wave and current interaction, bedload, suspended load, and wave-related sediment transport. In a departure from conventional models that directly or indirectly relate transport to bottom shear, the CSHORE model relates suspended sediment volume with energy dissipation due to breaking and bottom boundary shear. Likewise, the bedload expression has an inherent dependence on the energy dissipation for a wave-dominated nearshore environment and includes the important wave-generated flux that is directed with the wave propagation vector. A full treatment of the theoretical underpinnings of CSHORE is available in Johnson et al. 2012.

[*] For a full list of the spelled-out forms of the units of measure used in this document, please refer to *US Government Publishing Office Style Manual*, 31st ed. (Washington, DC: US Government Publishing Office 2016), 248-52, https://www.govinfo.gov/content/pkg/GPO-STYLEMANUAL-2016/pdf/GPO-STYLEMANUAL-2016.pdf.

To integrate CSHORE with Beach-fx, two separate sets of scripts have been developed in MATLAB and Python. Both sets independently generate the necessary CSHORE infiles, run all created CSHORE infiles, and process the results to be populated in the SRD. Alternative sets were developed to provide a choice of language. While it is not necessary for a Beach-fx user to be proficient in MATLAB or Python, both sets of scripts allow flexibility if additional features are desired.

METHODS: Regardless of the language of choice, the two sets of scripts utilize the same inputs and workflow. The directory structure of each distribution is shown in Figure 3, where each rectangular shape represents a directory and each circle represents a file, series of files, or executable. The main directory, "Dist," contains three sub-directories (executables, mfiles/pyfiles, work), three MATLAB/Python files, and a README file (not shown). The executables directory contains the CSHORE executable for both Windows and Linux operating systems. The mfiles/pyfiles directory contains various scripts that are utilized at runtime, although they are intended to run in the background and do not need to be modified.

Of the directories and files shown in Figure 1, the user needs to be aware of the work directory as well as each of the three primary scripts: (1) *make_cshore_infiles*, (2) *run_cshore*, and (3) *make_dat_file*. The scripts are intended to be run sequentially, with minimal user input required in *make_cshore_infiles*. The general workflow is comprised of three steps:

1. Prepare input files.
2. Modify input values in *make_cshore_infiles*.
3. Run MATLAB/Python scripts sequentially.

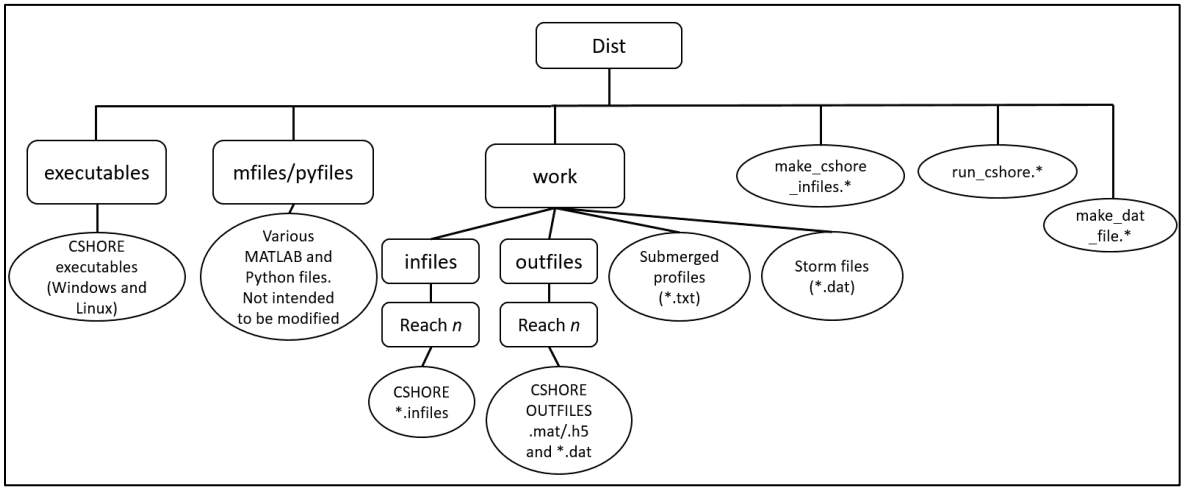

Figure 3. Directory structure of CSHORE for Beach-fx scripts.

Preparing Input. Two types of input files are required prior to running the set of scripts: submerged profile files (.txt) and storm information files (.dat). Examples of the required files are provided with each distribution.

Submerged profile files are in x-z format and tab delimited. The submerged profile files must follow a specific naming convention represented by the following:

*ReachName*_ submerged_profile.txt

where *ReachName* is variable and corresponds to each specific reach. Note that the *ReachName* must be identical to that specified within the *make_cshore_infiles* script. Additionally, each submerged profile file name must be succeeded by "submerged_profile.txt." An example of data contained within a submerged profile file is shown in Figure 4. The first column is composed of the x-values, or distance offshore (ft)*, whereas the second column contains the z-values (ft), positive up.

0.33	-0.033
10.331	-0.266
20.33	-0.5
30.33	-0.654
40.331	-0.849
50.331	-1.04
60.33	-1.272
70.33	-1.489

Figure 4. Example submerged profile file.

Storm information files are space delimited and contain data pertaining to storm surge and associated wave time series. Each storm file must be followed by the file extension ".dat." Similar to the submerged profile files, the storm .dat files must also follow a specific file format. An example storm .dat file is shown in Figure 5. The header row is the storm identifier. Each of the remaining columns correspond to the date (yyyymmddHHMM), zero moment wave height (m), mean wave period (sec), and storm surge water elevation (m). All files located in the *work* directory that end with the file extension ".dat" will be read when generating storm events.

```
204
200007120600 0.082157 2.911990 0.134639
200007120700 0.083399 2.906840 0.138414
200007120800 0.084491 2.911167 0.140420
200007120900 0.086214 2.909698 0.140830
200007121000 0.067202 2.947738 0.143338
200007121100 0.067616 2.924059 0.144676
200007121200 0.070739 2.921044 0.148033
200007121300 0.070864 2.923879 0.150680
```

Figure 5. Example storm .dat file.

* For a full list of the unit conversions used in this document, please refer to *US Government Publishing Office Style Manual*, 31st ed. (Washington, DC: US Government Publishing Office 2016), 345-7, https://www.govinfo.gov/content/pkg/GPO-STYLEMANUAL-2016/pdf/GPO-STYLEMANUAL-2016.pdf.

Modify Input Values in *make_cshore_infiles*. Following the preparation of necessary input files, the user can then modify values in the first script, *make_cshore_infiles*. The input required by this script can be decomposed into one of three components: profile information, CSHORE information, and tide information. In total, there are 16 required input attributes (Table 1 and Figure 6).

Table 1. *make_cshore_infiles* input requirements.

Attribute	Function (units)	Component
names	Reach names	Profile
height_dune	Array of dune heights (ft.)	
width_dune	Array of dune widths (ft.)	
width_berm	Array of berm widths (ft.)	
width_upland	Array of upland width (ft.)	
height_upland	Array of upland heights (ft.)	
slope_dune	Array of dune slopes	
height_berm	Array of berm heights (ft.)	
slope_foreshore	Array of foreshore slopes	
dx	CSHORE grid size (m.)	CSHORE
gamma	Shallow water ratio of wave height to water depth	
d50	D_{50} grain size (mm.)	
effb	Suspension efficiency due to breaking	
amp	Array of tide amplitudes (m.)	Tide
T	Tide period (hrs.)	
phases	Array of tidal phase shifts combined with peak storm surge	

```
# ~~~~~~~~~~~ BEGIN USER INPUT ~~~~~~~~~~~
profile_dict['names']              = ['Reach1', 'Reach3']
profile_dict['height_dune']        = [[10, 11, 12], [9.9, 10.5]]
profile_dict['width_dune']         = [[10, 12] , [12]]
profile_dict['width_berm']         = [[100, 150], [120, 140, 160]]
profile_dict['width_upland']       = [200, 220]
profile_dict['height_upland']      = [6, 7]
profile_dict['slope_dune']         = [0.25, 0.25]
profile_dict['height_berm']        = [4, 5]
profile_dict['slope_foreshore']    = [0.2, 0.2]

meta_dict['dx']         = 1
meta_dict['gamma']      = 0.7
meta_dict['d50']        = 0.3
meta_dict['effb']       = 0.002

tide_dict['amp']        = [.15, .2, .25]
tide_dict['T']          = 12.5
tide_dict['phases']     = [1, 2, 3, 4]
# ~~~~~~~~~~~ END USER INPUT ~~~~~~~~~~~
```

Figure 6. Example of python *make_cshore_infiles*.

The profile component of the input attributes defines both the project reaches and profile parameter space for each reach. The *names* attribute corresponds to the reach names and must match exactly with the *ReachName* specified in the submerged profile text file names. The length of the arrays for each profile component following the *names* attribute must have the same length as that of *names*.

Because the profile parameter space in Beach-fx is populated by varying the dune height, dune width, and berm width, these values are optionally composed of a list of two or more entries for each reach. The result is a *list of lists* as shown in Figure 6. The remaining profile items (*width_upland, height_upland, slope_dune, height_berm*, and *slope_foreshore*) are constant for each Beach-fx reach and are therefore specified as single values within a list. Upon running *make_cshore_infiles*, the profile space is populated with all combinations of the specified dune height, dune width, and berm width attributes. For example, three dune height values, four dune width values, and two berm width values results in 24 profiles.

There are four required CSHORE attributes: *dx*, *gamma*, *d50*, and *effb*. The attribute *dx* defines the spacing of the CSHORE grid in meters, and a typical value is 1.0. The attribute *gamma* defines the shallow water ratio of wave height to water depth. The attribute *d50* defines the median grain size (mm) and is additionally used to calculate the sediment fall velocity (Soulsby 1997). The attribute *effb* defines the suspension efficiency due to breaking, typically .002 to .005.

The final three attributes are related to defining tides that are combined with each storm surge hydrograph to create a total water elevation. Historically in Beach-fx, each storm surge hydrograph has been combined at peak surge with three statistically defined tidal amplitudes (low, medium, and high) at four variations in tidal phase shifts (high tide, mid-tide falling, low tide, and mid-tide rising). Given a storm surge hydrograph, the script *make_cshore_infiles* internally computes the tide time series and combines it with the storm surge hydrograph time series. There are three required tide attributes: *amp, T, phases.* The attribute *amp* defines the tidal amplitudes (m), and is specified as a list. The attribute *T* is the tidal period and is specified as a single value in hours

(typically 12.5 for diurnal tides). The attribute *phases* defines the phase shift used in creating each tide time series. The *phases* attribute can include 1: high tide; 2: mid-tide falling; 3: low tide; and 4: mid-tide rising. An example of the storm surge hydrograph, variations in tide amplitudes and resulting combined total water elevation is shown in Figure 7.

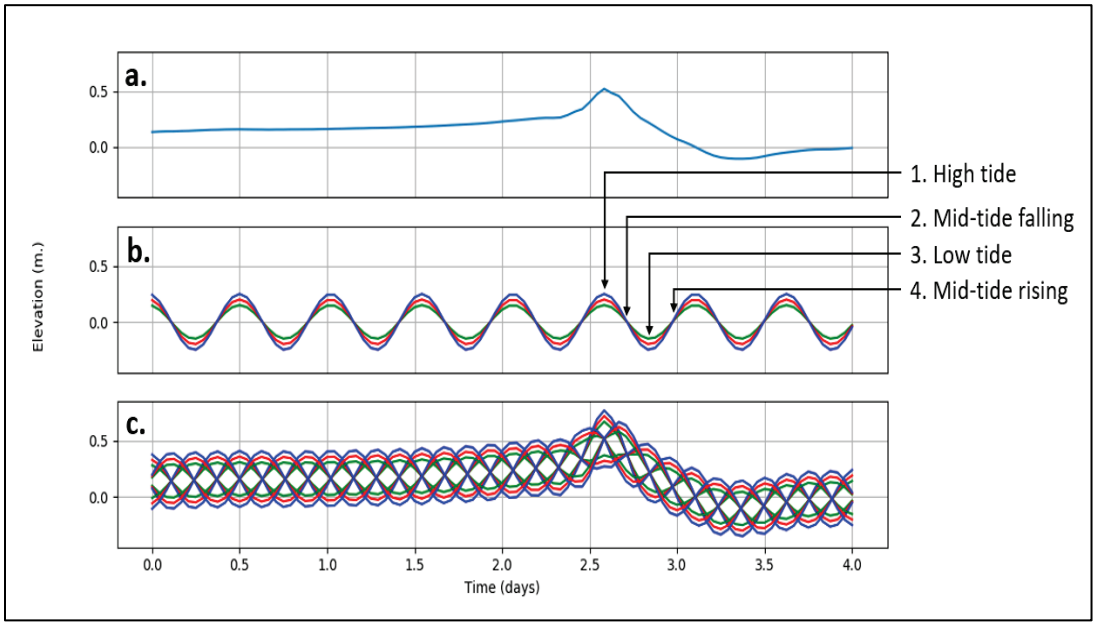

Figure 7. Example of storm surge hydrograph combined with variations in tide time series (a) storm surge, (b) tides with variation in amplitude, (c) combined tides and surge.

Run MATLAB/Python Scripts. Following the preparation of all input data and defining the profile space, CSHORE information, and variations in tide that will be utilized to create the total water elevation time series, the three scripts can be run sequentially (*make_cshore_infiles, run_cshore, make_dat_file*). Upon running *make_cshore_infiles*, CSHORE infiles are created for each reach/profile/storm combination. Example input and the resulting number of infiles are shown in Table 2.

Table 2. Combination of CSHORE infiles created by *make_cshore_infiles.*

Attribute	**Number of items**
Reaches	3
Profiles	24 (per reach)
Storm surge hydrographs	4
Variations in tide	12 (4 phase shifts; 3 variations of amplitude)
Total infiles created	**3456**

Each of the infiles are written to sub-directories within the *work* directory (.../work/infiles/*reach_n*). Note that the names and locations of these directories should not be modified as the subsequent scripts rely on the created directory structure.

Both *run_chore* and *make_dat_file* require no user input. The script, *run_cshore*, executes all of the created infiles sequentially. A new directory (../work/outfiles) will be created that stores output from the CSHORE runs that is necessary for Beach-fx. The MATLAB scripts write to .mat files whereas the Python scripts write to .h5 files.

The script *make_dat_file* converts the created .mat/.h5 CSHORE result files to the appropriate .dat file format for import to Beach-fx. A single .dat file will be created for each reach containing all of the profile/storm combinations within that reach. The .dat files are written to "../work/outfiles/*reach_n,*" and are created to be directly imported to Beach-fx.

CONCLUSION: This CHETN presents and documents a series of scripts that prepare, run, and process CSHORE files for use in Beach-fx. Scripts were written in both MATLAB and Python such that the user has a choice of language. Although two sets of scripts were developed, they both have the same input requirements, run order, and use the same directory structure. Three primary scripts are provided, *make_cshore_infiles, run_cshore, make_dat_file*, that are intended to be run sequentially to create the necessary .dat files for Beach-fx. The .dat files that are written can be directly imported to Beach-fx to build the storm response database. The series of MATLAB and Python scripts were written by Bradley Johnson and Dylan Sanderson, respectively, and are available at *https://github.com/erdc/BeachFX_CSHORE_link*.

ADDITIONAL INFORMATION: This CHETN was prepared by Bradley D. Johnson (*Bradley.D.Johnson@usace.army.mil*) and Dylan R. Sanderson, US Army Engineer Research and Development Center. The study is funded by the USACE Flood and Coastal Systems Research and Development Program. This CHETN should be cited as follows:

> Johnson, B. D., and D. R. Sanderson. 2020. *On the Use of CSHORE for Beach-fx*. ERDC/CHL CHETN-II-59. Vicksburg, MS: US Army Engineer Research and Development Center. *http://dx.doi.org/10.21079/11681/37949*

REFERENCES

Gravens, M. B., R. M. Males, and D. A. Moser. 2007. "Beach-fx: Monte Carlo Life-Cycle Simulation Model for Estimating Shore Protection Project Evolution and Cost Benefit Analyses." *Shore and Beach* 75(1): 12–19.

Johnson, B. D., N, Kobayashi and M. B. Gravens. 2012. *Cross-Shore Numerical Model CSHORE for Waves, Currents, Sediment Transport and Beach Profile Evolution*. ERDC/CHL TR-12-22. Vicksburg, MS: US Army Engineer Research and Development Center.

Larson, M., N. C. Kraus, and M. R. Byrnes. 1990. *SBEACH: Numerical Model for Simulating Storm-Induced Beach Change, Report 2 – Numerical Formulation and Model Tests.* Technical Report CERC-89-9. Vicksburg, MS: US Army Engineer Waterways Experiment Station. *http://hdl.handle.net/11681/12475*

Soulsby, R. L. 1997. *Dynamics of Marine Sands*. London: Thomas Telford.