Cliffs User Manual

Elena Tolkova *
NorthWest Research Associates, Bellevue, WA 98009-3027, USA

October 2, 2014

1 Overview

Cliffs is an open-source relative of MOST (Method Of Splitting Tsunamis) numerical model. Cliffs solves the non-linear shallow-water equations using the 1D finite-difference approximation VTCS-2 in open water (Titov and Synolakis, 1995), combined with the dimensional splitting (Strang, 1968; Titov and Synolakis, 1998; Titov and Gonzalez, 1997), and an original solution on the land-water boundary (vertical wall, moving shoreline) (Tolkova, 2014). The flow of the computations and input/output data types are similar to that in MOST version 4 (not documented; benchmarked for NTHMP in 2011 (Tolkova, 2012)), which was developed as an adaptation of a curvilinear version of the MOST model (Tolkova, 2008) to spherical coordinate systems arbitrary rotated on the Globe. For this Cliffs distribution, the computational flow of MOST-4 has been optimized to focus specifically on geophysical and Cartesian coordinate systems. Cliffs performs computations in a single grid (further referred to as Master grid) given initial deformation of the free surface or the sea floor, and/or initial velocity field, and/or boundary forcing. It also computes boundary time-series input into any number of enclosed grids, to allow further refinement of the solution. Cliffs operates in either geographical (longitude, latitude) spherical or Cartesian coordinates, in 2D or 1D domains.

All input and output data files, including bathymetry, should be in netcdf format.

Cliffs description is given in three parts. The first part, Running Options, presents a quick user manual. It explains command line options to run the executable, input parameters, and the format of input/output data. The second part, Examples, describes the sample modeling problems included with the code distribution. The last part, Numerics, lists numerical techniques utilized by Cliffs, and the original sources for those techniques. It concludes with discussing a specific numerical instability of the VTCS-2 algorithm introduced by abrupt changes in depth, and the ways to prevent the development of this instability.

2 Running Options

Cliffs is coded in Fortran-95 and parallelized using OpenMP. Makefile included with the code distribution is used to compile the code on Mac with gfortran. Netcdf libraries should be installed.

The command line to run the executable has five parameter fields delimited by $< \cdots >$ as follows:
./Cliffs <OuputDir/CaseTitle> <InputDataDir> <BoundaryInputTitle or 0 if no boundary input>
<InitialConditionsTitle or 0 if no initial conditions> <ParameterDir/ParameterFile>

*e.tolkova@gmail.com; elena@nwra.com

The program output is written in directory OutputDir. CaseTitle is assigned to all output files, which are:

- Netcdf binaries of velocities and elevation screenshots in Master grid saved to files named CaseTitle_sea_(u/v/h).nc, in the format specified below;
- Netcdf binary of maximum water surface elevation in Master grid in a file named CaseTitle_maxwave.nc;
- Time histories of the water surface elevation at virtual gages in a file named CaseTitle_gages.nc;
- Netcdf binaries of boundary input time-series for the enclosed grids saved to files named CaseTitle_BathyName_(west/east/south/north).nc, one file for each boundary as implied by the name ending, four files per a grid named BathyName.
- Text log file documenting the program execution named CaseTitle_log.txt

Netcdf structure of Cliffs input/output and the naming convention is kept consistent with that of the MOST model, to facilitate Cliffs use by MOST users. The screenshot output is defined against two spatial dimensions lon or xxx of length nx, lat or yyy of length ny, and one temporal dimension $time$ of length nt. During the computations, the $time$ dimension is defined as unlimited. The screenshot output is written in three netcdf files named CaseTitle_sea_(u/v/h).nc. Each file contains a 3D float-type variable $ua(time, lat/yyy, lon/xxx)$, or $va(time, lat/yyy, lon/xxx)$, or $ha(time, lat/yyy, lon/xxx)$ of size $nt \times ny \times nx$, representing wave zonal (x-axis) current u, or meridional (y-axis) current v, or the free surface displacement η, respectively. Each screenshot file also contains three 1D double-type variables of the coordinates lon/xxx and lat/yyy, and $time$, aligned with the same-name dimension.

Netcdf output of maximal water elevation contains two vectors (type double) of spatial coordinates and a 2D data set (type float) against the spatial dimensions lon or xxx, and lat or yyy.

Netcdf output of virtual gage records is written against two dimensions named *point* (which contains a number of the observation point) and *time*. It contains two vectors of the gages coordinates (type double) along the dimension *point*, time vector (type double) along the dimension *time*, and a 2D data set (type float) of the gage records along dimensions *point* and *time*.

Each boundary time-series data file contains 3D double-type variable $vals(tim, uvq, pnt)$ of size $nt \times 3 \times np$ and 1D variable $time(tim)$ of length nt. Variable $vals$ (Netcdf id = 1) consists of $nt \times np$ trios of instant values of zonal (x) current, meridional (y) current, and surface elevation aligned with dimension uvq of length 3, at time moments given by variable $time$ (Netcdf id = 2) aligned with dimension tim of length nt, in the consequent nodes of the corresponding grid boundary aligned with dimension pnt of length np (either nx or ny).

Variables and dimensions in the screenshot data and initial conditions are sought by their names (case-sensitive); those in the boundary input are sought by their number (Netcdf id).

Bathymetry data on a structured rectangular grid are represented by a double or float vector of the longitude (or x-coordinate) values at the grid nodes, the same-type vector of the latitudes (or y-coordinates), and a double or float 2D variable of water depth or land elevation. The bathymetric data are sought by the variable Netcdf id, with the lon/x node coordinates being the first variable, lat/y being the second, and the bathy/topo data being the third. Bathymetry data give the land elevation relative to the still sea surface, with the positive direction being down, that is, negative bathymetry values correspond to dry areas, while positive values provide an undisturbed water depth at each node. A 2D grid must have no less than 3 nodes in each direction. A 1D grid must still be defined against two netcdf dimensions, with the size of one dimension being equal 1. Nesting is not enabled for 1D grids, but varying spacing can be used instead.

The Units in all data sets are m for length, s for time, and decimal degrees for longitude (positive toward East) and latitude (positive toward North).

The input to the computations is sought in the directory InputDataDir. The input should be some of the following:

- boundary time-series files named BoundaryIputTitle_MasterGridName_(west/east/south/north).nc, one file for each boundary; and/or
- initial deformation of the free surface or the sea floor InitialConditionsTitle_h.nc; and/or
- either of both initial velocities of the water column InitialConditionsTitle_[u/v].nc.

The files providing initial velocities and surface elevation should have the same structure as the screenshot output. In particular, the initial condition data set should contain time dimension. Initial conditions containing a single screenshot must have time dimension of length one. In the absence of the boundary input, the start time of the computations will be the time of the screenshot. Initial conditions can also be read from the data set containing multiple screenshots (such as a data set created by the previous Cliffs run). This option is intended for the use in combination with the boundary input. The screenshot data set will be scanned for the frame taken at the time closest to the start of the boundary input. This frame will be used to initialize the wave state in the domain. The start time of the computations will be the start time of the boundary input.

A grid on which the initial conditions are defined does not have to encompass the Master grid, or coincide with its nodes. The initial conditions will be interpolated onto a part of Master grid within the source grid, and set to zero outside it.

In the absence of the initial conditions, the computations are run under the boundary input into the still domain. If the boundary input is provided, then all four files must be present in a 2D case, and two files (east&west, or south&north) must be present in a 1D case. In a 1D case, the boundary input must still have the same structure as in a 2D case, with the length of the dimension *pnt* equal 1, and the length of dimension *uvq* equal 3, though only one velocity component will be used (which is u if a 1-D domain occupies x-axis, and v if it occupies y-axis).

When computations are done in a nested grid configuration, the input is provided by the output from the previous simulation in the parent grid. Hence in this sequence of the simulations, $OutputDir$ and $InputDataDir$ refer to the same directory.

The input Parameter File is a text file with a list of computational parameters, which are:

1. an integer indicating the type of a coordinate system: 1 - cartesian, otherwise - geophysical;
2. name of Master grid bathymetry file
3. *nests* - number of grids enclosed in Master grid [1];
4. names of bathymetry files for the enclosed grids, one name per line;
5. *cuke* - still sea threshold in the boundary input, to detect wave arrival on the boundary. The start time of the computations will be the moment when the surface displacement exceeded the still sea threshold. Set this parameter less or equal zero - to start the computations when the boundary record starts. This parameter is not used when the computations are initiated with initial conditions only.
6. *ground* - minimal flow depth. At this depth and below it, a node is considered dry;
7. *crough* - friction coefficient (drag or Manning n^2, depending on an expression for the bottom friction with which the code is compiled);
8. *itopo* - an integer indicating a type of the land-water boundary: 0 - vertical wall at depth *dwall*, otherwise - land inundation enabled;

[1] Cliffs performs computations in a single grid (Master grid) and generates boundary input for each enclosed grid

9. *dwall* - the depth to place a vertical wall; not used when $itopo \neq 0$;
10. *dt* - time increment for computations in Master grid;
11. *steps_total* - total number of time steps to solve for;
12. *quake* - an integer indicating whether an initial surface deformation should be applied to the sea floor ($quake = 1$) or to the free surface (otherwise).
13. an integer indicating whether the computations should stop if the boundary input stops (0), or continue for the requested number of time steps (otherwise); not used with the initial conditions input only, regardless of its value in the parameter file;
14. *seaout* - number of time steps between screenshots. Setting $seaout > steps\_total$ will suppress screenshot output, and the file trio CaseTitle_sea_* will not be generated;
15. *lonsub* - node number increment in lon/x direction to sub-sample screenshots, resulting in the output at $1 : lonsub : ma$ rows;
16. *latsub* - node number increment in lat/y direction to sub-sample screenshots, resulting in the output at $1 : latsub : na$ columns;
17. *bndout* - number of time steps between saving wave variables along the boundaries of the enclosed grids; not used when no grids are enclosed;
18. *maxout* - number of time steps between updating the maximum water surface elevation. When $maxout > steps\_total$, the maximal surface height will be saved only at the end of the computations;
19. $Ngages$ - number of virtual gages / observation points, to output the surface height time histories;
20. *gout* - number of time steps between outputs to the gage time histories;
21. two-column list of the observation points indexes (lon/x node number, lat/y node number).

Some parameters might not be used, but should still be present (except enclosed grids in field 4, or gage parameters in fields 20 and 21, if the respective number in field 3 or 19 is zero).

Bathymetry files (grids) must be in the directory ParameterDir.

Sample parameter files are provided with the Examples.

3 Examples

More elaborated description of the next two canonical benchmark tests and their simulation with Cliffs can be found in (Tolkova, 2014).

3.1 Runup on a sloping beach

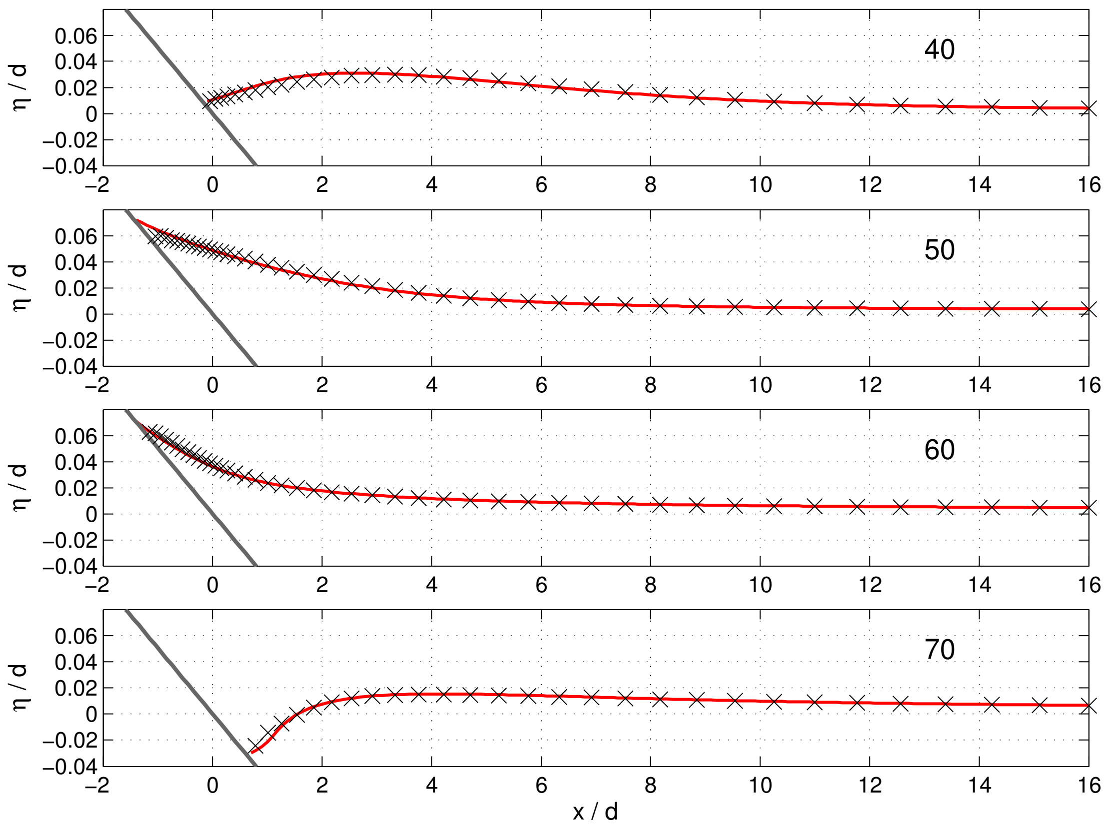

Figure 1: Water surface profiles for an initial solitary wave $0.0185d$ high climbing up a 1:19.85 beach at $t(g/d)^{1/2} = 40, 50, 60, 70$ (shown in a plot), black x - Cliffs numerical solution, red line - analytical solution. Plotted with Matlab script *plotprofiles0185_cliffs_vs_analyt.m*

This example simulates a canonical problem of a solitary wave runup onto a plane beach with a 1:19.85 slope introduced by Synolakis (1987); and illustrates Cliffs operation in 1D configuration in Cartesian coordinates. The geometry of the beach, the lab experiment, and the wave-profile are described in many articles (Synolakis, 1987; Titov and Synolakis, 1995; Synolakis et al., 2007). The cases with $H/d = 0.0185$ and $H/d = 0.3$, representing respectively a non-breaking and a severely breaking wave of initial height H over depth d, are most commonly used for model verification (Li and Raichlen, 2002; Nicolsky et al., 2011; NTHMP, 2011) against laboratory and analytical data of Synolakis. Provided Cliffs set-ups to simulate these two cases include:

- parameter file *slpbeach_params.txt*, shown in Table 1;
- bathymetry file *slpbeach1D.nc*, with the grid spacing gradually varying from d on the deep end to $0.1d$ near the beach, as described in (Tolkova, 2012, 2014);
- initial conditions files *solitary.0185_h.nc* and *solitary.0185_u.nc*, to simulate a non-breaking wave case;
- initial conditions files *solitary.30_h.nc* and *solitary.30_u.nc*, to simulate a breaking wave case.

Table 1: Parameter file for simulating runup onto a sloping beach

1	1 - cartesian, otherwise-spherical
slpbeach1D.nc	
0	Number of grids enclosed in Master
0	Still sea threshold on the boundary
0.003	Minimal flow depth (m)
0	friction coefficient (Manning n**2)
1	topo flag: 0-wall, otherwise - land inundation
0.5	Vertical wall, if any, at depth (m)
0.03	time step (sec)
2000	total amount of steps
0	1 - to deform bottom
1	0 - to stop when boundary forcing stops
3	steps between screenshots
1	subsample screenshots in x
1	subsample screenshots in y
1	save feed into nested grids every n steps
500	steps between maxwave updates
0	N gages

Should all input/output files and the executable be in the same directory, a command line to run the simulation can read:

./Cliffs ./sol0185 ./ 0 solitary.0185 ./slpbeach_params.txt OR
./Cliffs ./sol30 ./ 0 solitary.30 ./slpbeach_params.txt

Given the particular parameter file, Cliffs will generate four files (screenshots of velocity and elevation, maximal elevation, and a log file) with each run, for example:

sol0185_sea_h.nc
sol0185_sea_u.nc
sol0185_maxwave.nc
sol0185_log.txt

Figure 1 shows surface profiles extracted from Cliffs output *sol0185_sea_h.nc* with a provided Matlab script (*plotprofiles0185_cliffs_vs_analyt.m*) atop the analytical solution (read from file *canonical_profiles.txt* provided by Dr. Dmitry Nicolsky, UAF for the 2011 NTHMP Model Benchmarking Workshop).

3.2 Simulation of the 1993 Hokkaido-Nansei-Oki (Okushiri Island) tsunami

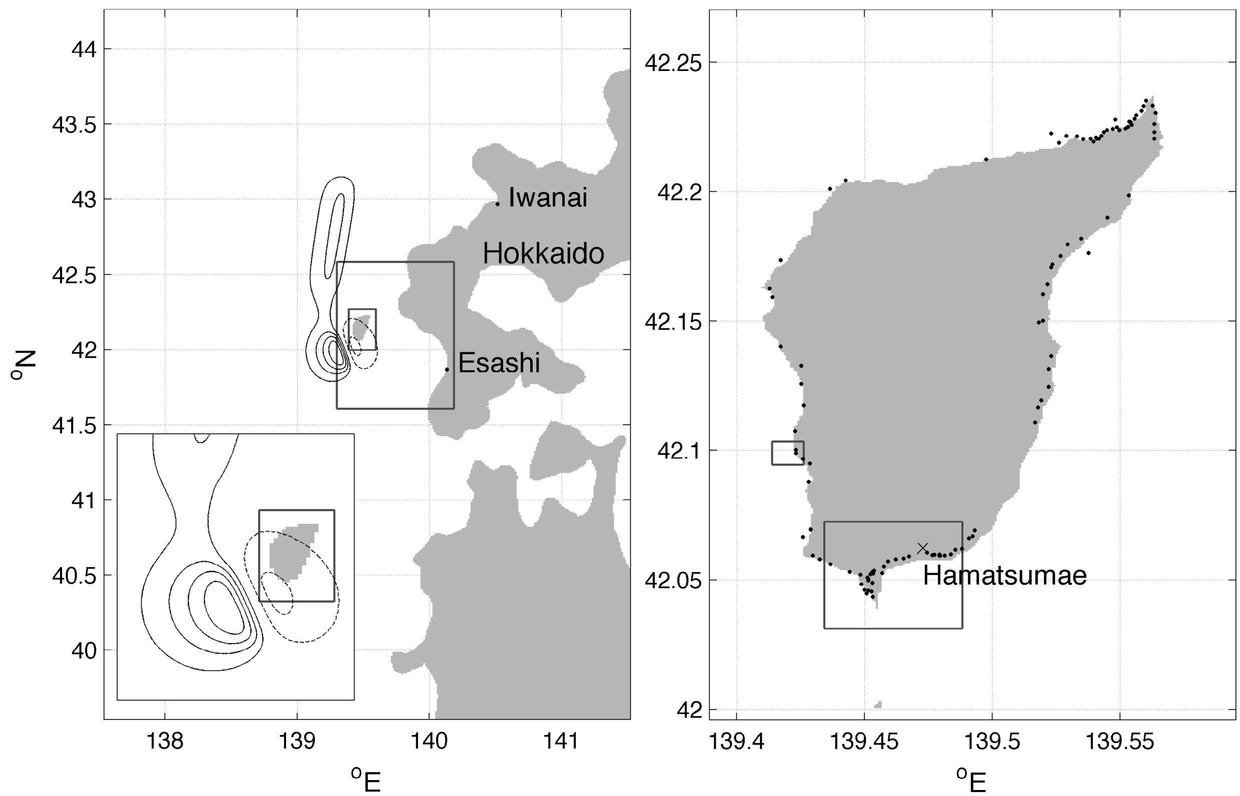

Figure 2: Left: computational domain used to simulate the 1993 Hokkaido tsunami with contours of the two nested grids; initial sea surface deformation with contour lines at -1, -0.5, 1, 2, 3, 4 m levels, subsidence contours are shown with dashed lines, uplift with solid lines. The deformation area near Okushiri is zoomed-in in the bottom left corner. Right: 3-d nesting level grid around Okushiri island with contours of the 4-th level grids around Monai and Aonae; locations of field measurements.

On July 12, 1993, the Mw 7.8 earthquake west of Okushiri island, Japan, generated a tsunami that has become a test case for tsunami modeling efforts (Takahashi, 1996), (Synolakis et al., 2007), (NTHMP, 2011). The complete set-up to simulate this event with Cliffs is provided. This example illustrates Cliffs operation in 2D configuration in geophysical coordinates with the use of nested grids, for simulating tsunami propagation from the source earthquake, and the consequent runup onto land.

The event is simulated with the use of five grids at four levels of nesting, with resolution varying from 30 sec of Great arc (930 m) to 6 m. The outer grid (OK30s_SSL2.1.nc) used to simulate the 1993 Okushiri tsunami and the initial sea surface deformation are shown in the left pane in Figure 2. The computations at a resolution of 30 sec of the Great arc were sequentially refined with nested grids spaced at 10 arc-sec (OK10s_SSL2.1.nc, enclosed in OK30s_SSL2.1.nc), 2 arc-sec around Okushiri island (OK02s_SSL2.1.nc, enclosed in OK10s_SSL2.1.nc), 15 m around Aonae peninsular (AO15m_SSL2.1.nc, enclosed in OK02s_SSL2.1.nc), and 6 m around Monai valley (MB06m_SSL2.1.nc, enclosed in OK02s_SSL2.1.nc as well). The grids contours are pictured in Figure 2. Computations in the first nesting level grid (OK30s_SSL2.1.nc) are initiated with the initial deformation read from the file bottomdefBP9_h.nc and applied to the sea floor. Computations in all other grids are run under the boundary input computed in the parent grid. The initial deformation of the sea floor is applied in each grid as well.

To run the simulation, copy Cliffs executable to directory OkushiriTsunami. The directory also contains a subdir OkushiriGrids with all the grids and the corresponding parameter files, and a subdir Simulation with the bottom deformation file. The sequence of commands to perform the entire simulation (also contained in

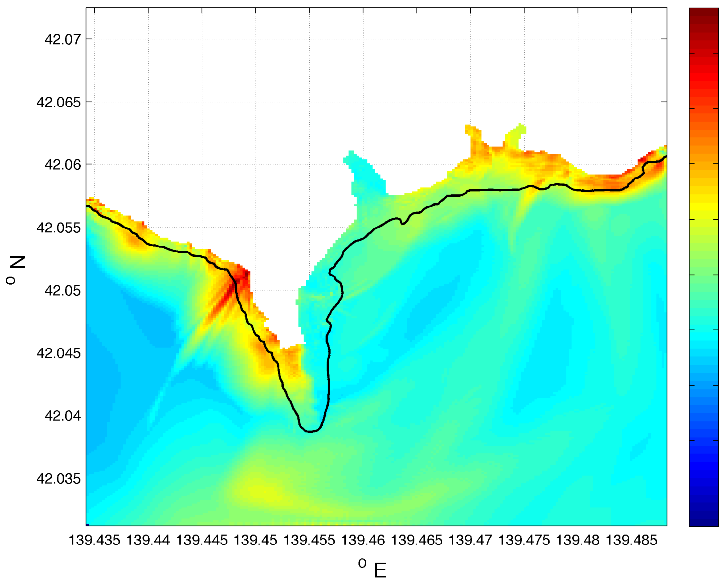

Figure 3: Maximal wave height in Aonae grid; colorbar - meters, black line - original shoreline before subsidence. Plotted with Matlab script *aonae_inundation.m*

a script runcliffs2Okushiri) reads:

```
./Cliffs  Simulation/OK30  Simulation/  0     bottomdefBP9  OkushiriGrids/paramsOK30s.txt
./Cliffs  Simulation/OK10  Simulation/  OK30  bottomdefBP9  OkushiriGrids/paramsOK10s.txt
./Cliffs  Simulation/OK02  Simulation/  OK10  bottomdefBP9  OkushiriGrids/paramsOK02s.txt
./Cliffs  Simulation/AO    Simulation/  OK02  bottomdefBP9  OkushiriGrids/paramsAO15m.txt
./Cliffs  Simulation/MB    Simulation/  OK02  bottomdefBP9  OkushiriGrids/paramsMB06m.txt
```

Each run will result in populating directory Simulation with the results of the computations in a respective grid, and generated boundary input into the enclosed grid(s) if any. As seen in the parameter files (tables 2-6), screenshot output interval *seaout* is set greater than the total number of steps *steps_total*, to suppress the screenshot output. Thus the computations in every grid result in saving the maximal wave height and in generating the boundary input into the next level grid. Computations in the first and second level grids also generate gage outputs with water level time histories near Iwanai and Esashi. The simulation time is 4 hrs, which is set by the simulation time in the first (30-sec) grid. The simulation time in other grids is set to longer according to the requested total number of time steps, but will be stopped earlier due to termination of the boundary input into that grid. A vertical wall is placed at 5 m depth in the 30-sec grid, and at 1 m depth in the 10-sec grid, while land inundation is permitted in the next three grids.

Table 2: Parameter file for simulating the Hokkaido tsunami, 1-st level grid

0	1 - cartesian, otherwise-spherical
OK30s_SSL2.1.nc	
1	Number of grids enclosed in Master
OK10s_SSL2.1.nc	
0.001	Still sea threshold on the boundary
1	Minimal flow depth (m)
0.0009	friction coefficient (Manning n**2)
0	topo flag: 0-wall, otherwise - land inundation
5	Vertical wall, if any, at depth (m)
4.0	time step (sec)
3600	total amount of steps
1	1 - to deform bottom
1	0 - to stop when boundary forcing stops
5000	steps between screenshots
1	subsample screenshots in x
1	subsample screenshots in y
1	save feed into nested grids every n steps
500	steps between maxwave updates
1	N gages
5	steps between outputs to gages
265 416	gage indexes

Table 3: Parameter file for simulating the Hokkaido tsunami, 2-nd level grid

0	1 - cartesian, otherwise-spherical
OK10s_SSL2.1.nc	
1	Number of grids enclosed in Master
OK02s_SSL2.1.nc	
0.001	Still sea threshold on the boundary
0.5	Minimal flow depth (m)
0.0009	friction coefficient (Manning n**2)
0	topo flag: 0-wall, otherwise - land inundation
1.0	Vertical wall, if any, at depth (m)
1.5	time step (sec)
9600	total amount of steps
1	1 - to deform bottom
0	0 - to stop when boundary forcing stops
10000	steps between screenshots
1	subsample screenshots in x
1	subsample screenshots in y
1	save feed into nested grids every n steps
500	steps between maxwave updates
1	N gages
12	steps between outputs to gages
223 94	gage indexes

Table 4: Parameter file for simulating the Hokkaido tsunami, 3-d level grid

0	1 - cartesian, otherwise-spherical
OK02s_SSL2.1.nc	
2	Number of grids enclosed in Master
AO15m_SSL2.1.nc	
MB06m_SSL2.1.nc	
0.001	Still sea threshold on the boundary
0.5	Minimal flow depth (m)
0.0009	friction coefficient (Manning n**2)
1	topo flag: 0-wall, otherwise - land inundation
1	Vertical wall, if any, at depth (m)
0.4	time step (sec)
36000	total amount of steps
1	1 - to deform bottom
0	0 - to stop when boundary forcing stops
50000	steps between screenshots
1	subsample screenshots in x
1	subsample screenshots in y
1	save feed into nested grids every n steps
500	steps between maxwave updates
0	N gages

Table 5: Parameter file for simulating the Hokkaido tsunami, 4-th level grid, Aonae

0	1 - cartesian, otherwise-spherical
AO15m_SSL2.1.nc	
0	Number of grids enclosed in Master
0.001	Still sea threshold on the boundary
0.1	Minimal flow depth (m)
0.0009	friction coefficient (Manning n**2)
1	topo flag: 0-wall, otherwise - land inundation
1	Vertical wall, if any, at depth (m)
0.5	time step (sec)
30000	total amount of steps
1	1 - to deform bottom
0	0 - to stop when boundary forcing stops
50000	steps between screenshots
1	subsample screenshots in x
1	subsample screenshots in y
1	save feed into nested grids every n steps
500	steps between maxwave updates
0	N gages

Table 6: Parameter file for simulating the Hokkaido tsunami, 4-th level grid, Monai

0	1 - cartesian, otherwise-spherical
MB06m_SSL2.1.nc	
0	Number of grids enclosed in Master
0.001	Still sea threshold on the boundary
0.1	Minimal flow depth (m)
0.0009	friction coefficient (Manning n**2)
1	topo flag: 0-wall, otherwise - land inundation
1	Vertical wall, if any, at depth (m)
0.2	time step (sec)
50000	total amount of steps
1	1 - to deform bottom
0	0 - to stop when boundary forcing stops
60000	steps between screenshots
1	subsample screenshots in x
1	subsample screenshots in y
1	save feed into nested grids every n steps
5000	steps between maxwave updates
0	N gages

4 Numerics

VTCS-2 numerical approximation was introduced by VT and CS in 1995 to model the propagation and runup of 1-D long waves under the framework of the non-linear shallow-water theory (Titov and Synolakis, 1995). VTCS-2 finite-difference scheme is described in detail in (Titov and Synolakis, 1995, 1998). Burwell et al (2007) analyzed diffusive and dispersive properties of the VTCS-2 solutions.

A 1-D algorithm can be efficiently applied to solving 2-D shallow-water equations using well-known dimensional splitting method (Strang, 1968; Yanenko, 1971; LeVeque, 2002). In this way, the VTCS-2 scheme was extended to handle 2-D problems in Cartesian coordinates (Titov and Synolakis, 1998), geophysical spherical coordinates (Titov and Gonzalez, 1997) (the first mentioning of the MOST model), and in an arbitrary orthogonal curvilinear coordinates (Tolkova, 2008).

However, the dimensional splitting might result in underestimating an amplitude of a reflected wave in the MOST model, unless the reflecting boundary is aligned with either x or y coordinate axis. Slight modification to the reflective boundary conditions in MOST, equivalent to a half-node re-positioning of the reflecting wall within a cell, caused an appreciable difference in the results in some situations. The modification was extended to include runup computations, to complement the known propagation algorithm (note that the MOST inundation algorithm contains elements which are not public knowledge) (Tolkova, 2014).

Hence the resulting model, Cliffs, is a close relative of MOST, built on the numerical approximations well described in the foregoing literature (to where a reader is referred to for details). Common with the MOST model, Cliffs utilizes the VTCS-2 difference scheme in open water combined with the dimensional splitting, and the same treatment of open boundaries. An original solution is used on the land-water boundary which computes both reflection from a solid wall and land inundation (Tolkova, 2014). Cliffs inundation algorithm is more compact and approximately 10% more efficient computationally than the present MOST algorithm.

4.1 Numerical stability

In the presence of sharp changes in depth, the VTCS-2 solutions sometimes develop a specific, slowly growing instability. This instability is not described in the literature, and therefore it is discussed in details here. The VTCS-2 difference scheme is formulated below as in (Tolkova, 2014).

The VTCS-2 scheme applies to Riemann invariants of the 1D shallow-water equations (SWE)

$$p = u + 2\sqrt{gh}, \quad q = u - 2\sqrt{gh},$$

where u is depth-averaged velocity, and h is the water column height. The characteristic form of the 1D SWE, without forcing terms other than gravity, reads:

$$p_t = -(\lambda \cdot p_x - gd_x) \tag{1a}$$

$$q_t = -(\tilde{\lambda} \cdot q_x - gd_x) \tag{1b}$$

where $\lambda = u + \sqrt{gh}$, $\tilde{\lambda} = u - \sqrt{gh}$, subscript denotes a respective partial derivative. The finite-difference scheme for non-uniform grid spacing is given below for p-invariant:

$$\begin{aligned} p_j^{n+1} = p_j^n - \frac{\Delta t}{\Delta x_j + \Delta x_{j-1}} \Big[& Q(j+1, j-1) + \\ & + \frac{\lambda_j \Delta t}{\Delta x_{j-1}} Q(j, j-1) - \frac{\lambda_j \Delta t}{\Delta x_j} Q(j+1, j) \Big] \end{aligned} \tag{2}$$

where

$$Q(k, j) = \frac{1}{2}(\lambda_k + \lambda_j)(p_k - p_j) - g(d_k - d_j) \tag{3}$$

Δt is time increment; $\Delta x_j = x_{j+1} - x_j$ is space increment; all the right hand side variables are evaluated at a time step n. Term Q can be recognized as a counterpart of the expression in brackets on the right hand side of (1). In the steady state, $Q = 0$, which ensures an automatic preservation of steady state.

Once values p and q have been updated, the corresponding state variables are recovered as

$$u = (p+q)/2 \tag{4a}$$

$$h = (p-q)^2/16g \tag{4b}$$

In a basin with constant depth, the scheme is stable under the known limit on the Courant number: $|\lambda|\Delta t/\Delta x \leq 1$.

To investigate the stability of the scheme (2) in a basin with varying depth, let us compute one-time-step evolution of an infinitesimally small initial pulse with height (surface displacement) η_j and velocity u_j applied in a single node j in a still basin. For simplicity, assume uniform spacing Δx. Then

$$Q(j+1,j-1) = 0,$$
$$Q(j,j-1) = \frac{1}{2}u_j\left(3\sqrt{gd_j} - \sqrt{gd_{j-1}}\right) + g\eta_j + O^2,$$
$$Q(j+1,j) = -\frac{1}{2}u_j\left(3\sqrt{gd_j} - \sqrt{gd_{j+1}}\right) - g\eta_j + O^2,$$

where O^2 refers to the second order quantities in u and η. Substituting the above into (2) yields

$$p_j^{+1} = p_j - \sqrt{gd_j}\frac{\Delta t^2}{\Delta x^2}\left[\frac{u_j}{4}\left(6\sqrt{gd_j} - \sqrt{gd_{j-1}} - \sqrt{gd_{j+1}}\right) + g\eta_j\right] + O^2. \tag{5}$$

Solution for q_j^{+1} can be obtained from (5) by reverting signs of radicals. Substituting p_j^{+1} and q_j^{+1} into (4) yields the velocity and water surface displacement after one time step:

$$u_j^{+1} = \left(1 - \beta_j^2 f_j\right) u_j + O^2 \tag{6a}$$

$$\eta_j^{+1} = \left(1 - \beta_j^2\right) \eta_j + O^2 \tag{6b}$$

where $\beta_j = \Delta t\sqrt{gd_j}/\Delta x$ is the Courant number associated with j-th node; factor f_j is equal to

$$f_j = \frac{6\sqrt{d_j} - \sqrt{d_{j-1}} - \sqrt{d_{j+1}}}{4\sqrt{d_j}} \tag{7}$$

As seen from (6a), the VTCS-2 solution over varying bottom is likely to become unstable ($|u_j^{+1}| > |u_j|$) whenever factor f_j is negative. Thus a condition $f > 0$ restricting depth variations as

$$6\sqrt{d_j} > \sqrt{d_{j-1}} + \sqrt{d_{j+1}} \tag{8}$$

is necessary to ensure stability of the VTCS-2 solver, in addition to CFL condition.

Being limited to a very particular case, (8) does not provide a sufficient stability condition. To strengthen the condition, one might want to restrict depth variations on the left and the right of j-th node independently. Let us expand factor f_j as

$$f_j = \frac{1}{2}(1+r_l) + \frac{1}{2}(1+r_r) \tag{9}$$

where

$$r_l = \frac{\sqrt{d_j} - \sqrt{d_{j-1}}}{2\sqrt{d_j}}, \quad r_r = \frac{\sqrt{d_j} - \sqrt{d_{i+j}}}{2\sqrt{d_j}} \tag{10}$$

An empirical stability condition which ensures positive f follows:

$$1 + r_r \geq \epsilon > 0 \quad AND \quad 1 + r_l \geq \epsilon > 0, \tag{11}$$

or

$$\max\{d_{i-1}/d_i, \quad d_{i+1}/d_i\} \leq \alpha^2, \tag{12}$$

where $\alpha = 3 - 2\epsilon$, $0 < \epsilon < 1$, $1 < \alpha < 3$. Typically, $\alpha = 2$ ($\epsilon = 0.5$) is sufficient. $\alpha = 1$ can only be met in a basin with constant depth.

It is recommended to pre-process a digital map of a sea floor to impose restrictions (11)/(12) both row-wise and column-wise, as follows: given α, at any node j where (12) does not hold, the depth value is to be replaced with

$$d_j = (\sqrt{d_{j-1}} + \sqrt{d_{j+1}})^2/4 \tag{13}$$

in an open water, and with

$$d_j = d_w/\alpha^2 \tag{14}$$

next to the coastline, where d_w is the depth at the wet neighbor of the j-th node. Substituting updated depth (13) into (10), and the later ones into condition (11) results in

$$(1-\epsilon)\sqrt{d_{i-1}} + (3-\epsilon)\sqrt{d_{i+1}} > 0, \quad (1-\epsilon)\sqrt{d_{i+1}} + (3-\epsilon)\sqrt{d_{i-1}} > 0 \tag{15}$$

which satisfies (11) for any $\epsilon < 1$.

Program *depth_ssl* (supplied with Cliffs distribution) can be used to perform the suggested pre-processing of a depth map. A command line to run the program should read:

./depth_ssl <InputDir/InputBathyName.nc > <OutputDir/> $< n >$ [<OutputBathyName.nc>]

The third argument is an integer indicating that the program should inquire about the desired limit α (between 1.5 and 2.5), and the minimal water depth (m) in wet nodes. If this argument is set to 0, the default $\alpha = 2$ and *min_depth* $= 0.1$ m will be used. If the last argument is omitted, the output bathy file will be created, named InputBathyName_ssl.nc, in the directory OutputDir.

Bathymetry files should/will be in the bathy/topo netcdf format described in section 2.

References

Burwell, D., Tolkova, E., and Chawla, A. (2007), Diffusion and Dispersion Characterization of a Numerical Tsunami Model. Ocean Modelling, 19 (1-2), 10-30. doi:10.1016/j.ocemod.2007.05.003

LeVeque, R.J., Finite volume methods for hyperbolic problems. (Cambridge University Press, UK 2002).

Li, Y., and Raichlen, F. (2002), Non-breaking and breaking solitary wave run-up. J. Fluid Mech., 456, 295318.

Nicolsky, D.J., Suleimani, E.N., and Hansen, R.A. (2011), Validation and verification of a numerical model for tsunami propagation and runup. Pure Appl. Geophys. 168, 1199-1222.

[NTHMP] National Tsunami Hazard Mitigation Program, July 2012. Proceedings and Results of the 2011 NTHMP Model Benchmarking Workshop. Boulder: U.S. Department of Commerce/NOAA/NTHMP; NOAA Special Report. 436 p.

Shi, F., Kirby, J.T., Harris, J.C., Geiman, J.D., Grilli, S.T. (2012), A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation. Ocean Modelling, 43-44, 36-51.

Strang, G. (1968), On the construction and comparison of difference schemes. SIAM Journal on Numerical Analysis, 5(3), 506-517.

Synolakis, C.E. (1987), The runup of solitary waves. J. Fluid Mech., 185, 523-545.

Synolakis, C.E., Bernard, E.N., Titov, V.V, Kanoglu, U., and Gonzalez, F.I. (2007), Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models. NOAA Tech. Memo. OAR PMEL-135, NTIS: PB2007-109601, NOAA/Pacific Marine Environmental Laboratory, Seattle, WA, 55 pp.

Takahashi, T. (1996), Benchmark problem 4: the 1993 Okushiri tsunami - Data, conditions and phenomena. In Long-Wave Runup Models, World Scientific, 384-403.

Titov, V. V., and Synolakis, C. E. (1995), Modeling of breaking and nonbreaking long-wave evolution and runup using VTCS-2, J. Waterw., Port, Coastal, Ocean Eng., 121(6), 308316.

Titov, V., and Gonzalez F.I. (1997), Implementation and testing of the Method of Splitting Tsunami (MOST) model. NOAA Tech. Memo. ERL PMEL-112 (PB98-122773), NOAA/Pacific Marine Environmental Laboratory, Seattle, WA, 11 pp.

Titov, V. V., and Synolakis, C. E. (1998), Numerical modeling of tidal wave runup, J. Waterw., Port, Coastal, Ocean Eng., 124(4), 157171.

Titov V. V., Gonzalez, F. I., Bernard, E. N., Eble, M. C., Mofjeld, H. O., Newman, J. C., and Venturato, A. J. (2005), Real - Time Tsunami Forecasting: Challenges and Solutions, Natural Hazards, 35: 41 - 58.

Tolkova, E. (2008) Curvilinear MOST and its first application: Regional Forecast version 2. In: Burwell, D., Tolkova, E. Curvilinear version of the MOST model with application to the coast-wide tsunami forecast, Part II. NOAA Tech. Memo. OAR PMEL-142, 28 pp.

Tolkova, E. (2012), MOST (Method of Splitting Tsunamis) Numerical Model. In: [NTHMP] National Tsunami Hazard Mitigation Program. Proceedings and Results of the 2011 NTHMP Model Benchmarking Workshop. Boulder: U.S. Department of Commerce/NOAA/NTHMP; NOAA Special Report. 436 p.

Tolkova, E. (2014), LandWater Boundary Treatment for a Tsunami Model With Dimensional Splitting. Pure and Applied Geophysics, 171(9), 2289-2314. DOI 10.1007/s00024-014-0825-8

Yanenko N.N. (1971). The method of Fractional Steps. Translated from Russian by M.Holt. Springer, New York, Berlin, Heidelberg.