

This assignment extends our current implementation – already capable to simulate artificial scenarios – with the most important data for Tsunami simulations: Bathymetry and vertical displacements of the sea bottom generated by earthquakes. Data for the Great Tōhoku Earthquake and Tsunami (11th March 2011) and the February 27, 2010 Chile Tsunami Event is provided as binary netCDF files.

We simulate both Tsunamis in Chapter 4 after we programmed the file handling of netCDF files in Chapter 1, 2 and 3.

Literature

- *NOAA/WDS Global Historical Tsunami Database at NGDC*: http://www.ngdc.noaa.gov/hazard/tsu_db.shtml
- *netCDF Users Guide*: http://www.unidata.ucar.edu/software/netcdf/docs/user_guide.html
- *Conventions for the standardization of netCDF files*: http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html

1 netCDF Output

In this chapter we change our file format from ASCII-VTK to netCDF (Network Common Data Form). You can find an introduction to netCDF as well as a description of the data model in the user guide.

netCDF output has two major advantages over ASCII-VTK. First, netCDF files are binary files, thus fast and smaller than ASCII-VTK. Second netCDF is a single-file and self-describing format: We can store all time steps in one file. This removes the overhead introduced by large number of files.

SWE comes with a netCDF writer. To enable this writer, you first have to install the netCDF library for C (the C++ version is not required). If the library is included in your distribution, you can use the package manager, otherwise you need to install it from source¹. The netCDF library depends on the HDF5 library², hence do not forget to install HDF5 as well. To build SWE with netCDF support use the option `writeNetCDF='yes'`. If you installed netCDF from source, it is very likely that you have to set `netCDFDir` as well to your install directory.

Tasks

1. Recompile SWE with the netCDF writer enabled and rerun a 2D example from the last assignment.

¹<http://www.unidata.ucar.edu/downloads/netcdf/index.jsp>

²<http://www.hdfgroup.org/HDF5/>

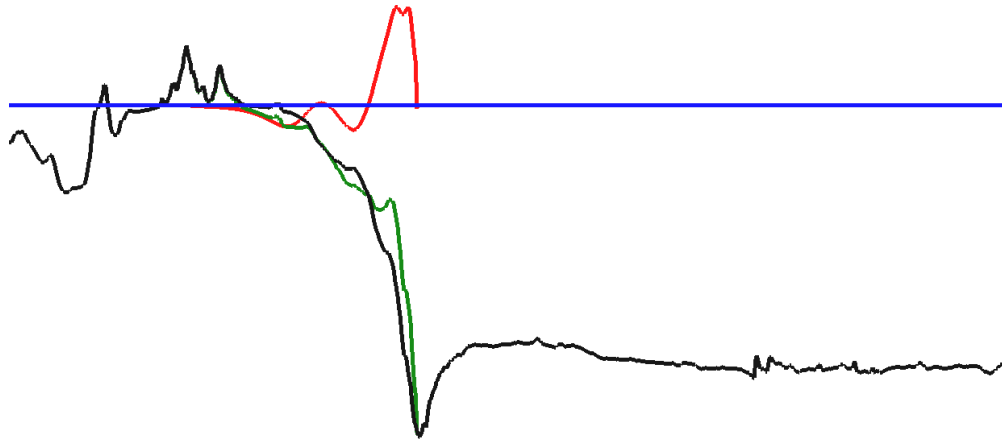


Figure 1: Shown is the scaled bathymetry before an earthquake in black, the displacement in red, bathymetry after an earthquake in green and the sea surface in blue.

2. The netCDF writer implements the COARDS conventions, which allows Paraview to display your newly created output files. Visualize the output files with Paraview.

2 netCDF Input

Up to this point we used hard-coded scenarios for generation of our input data. In this chapter we will add a new scenario to SWE, which is able to load two netCDF files respecting the COARDS conventions: The first file contains space-dependent bathymetry $\mathbf{b}(\mathbf{x}, \mathbf{y})$. Space-dependent vertical displacements, induced by an entire earthquake, are located in the second file. Figure 1 shows a two dimensional cut through bathymetry and scaled displacement data. Each file contains three variables \mathbf{x} , \mathbf{y} and \mathbf{z} . \mathbf{x} and \mathbf{y} are coordinate variables (see the COARDS conventions for details about coordinate variables), \mathbf{z} either bathymetry or displacement.

The computational domain for the scenario matches the extend of the bathymetry file. However, the displacement file covers only regions of non-zero displacement. In other parts of the computational domain, the displacement is zero.

Derive a scenario from `SWE_Scenario`. You have to overwrite at least the two functions `getWaterHeight` and `getBathymetry`. We assume that the sea is at rest before the earthquake. Therefore, those two values can be easily computed from the input files:

$$\text{water height} = -\max(\text{bathymetry before earthquake}, 0) \quad (1)$$

$$\begin{aligned} \text{bathymetry for tsunami simulation} \\ &= \text{bathymetry after earthquake} \\ &= \text{bathymetry before earthquake} + \text{displacement} \end{aligned} \quad (2)$$

Depending on the resolution of your simulation, SWE will request the bathymetry and water height at different coordinates. If the file does not provide a value at the exact coordinate,

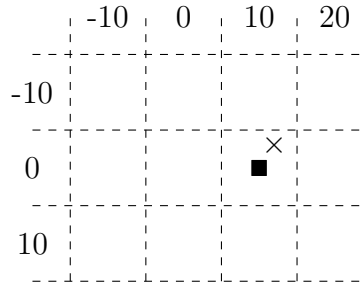


Figure 2: The simulation requires the value at \times (12,-3); the scenario returns \blacksquare (10,0)

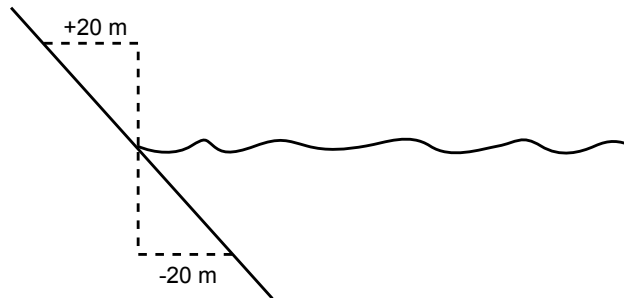


Figure 3: Dashed line: Bathymetry returned by the scenario

use the closest value you have, an example is illustrated in Figure 2. To be able to apply wall boundary conditions in the solver (Assignment 2, Chapter 1), we will not allow bathymetry values between -20 and 20 m (Figure 3). In our scenario we assume a bathymetry of -20 m for any value between -20 and 0 and 20 m for values between 0 and 20 .

To test the netCDF reader, we simulate a “tsunami in a swimming pool”. The swimming pool has a constant bathymetry of -100 m before the “earthquake” and a size of 10×10 km. The “earthquake” generates a displacement in a 1×1 km square in the center of the swimming pool. The displacement (Figure 4) can be described via the function $d(\mathbf{x}, \mathbf{y})$, where $(0, 0)$ is the center of the computational domain:

$$d : [-500\text{m}, +500\text{m}] \times [-500\text{m}, +500\text{m}] \rightarrow \mathbb{R} \quad (3)$$

$$d(\mathbf{x}, \mathbf{y}) = 5\text{m} \cdot d_x(\mathbf{x})d_y(\mathbf{y}) \quad (4)$$

$$d_x(\mathbf{x}) = \sin\left(\left(\frac{\mathbf{x}}{500\text{m}} + 1\right) \cdot \pi\right) \quad (5)$$

$$d_y(\mathbf{y}) = -\left(\frac{\mathbf{y}}{500\text{m}}\right)^2 + 1 \quad (6)$$

Tasks

1. Implement the artificial tsunami in `SWE_ArtificialTsunamiScenario`.

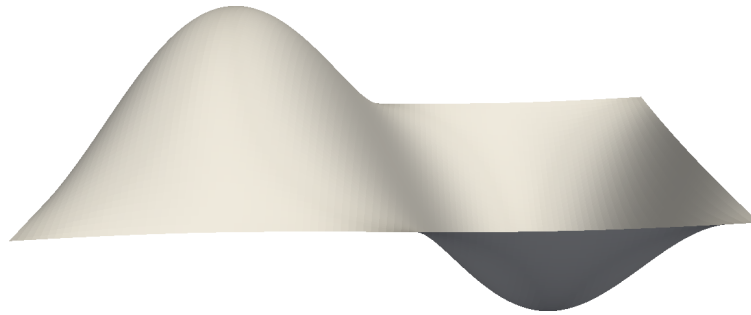


Figure 4: Displacement of the artificial tsunami.

2. Implement the `SWE.TsunamiScenario` that loads netCDF files as described above. You can use `ncgen`³ to generate small netCDF files for your unit tests.
3. Integrate the new scenario into your code. The user should be able to set boundary conditions and the simulated time via command line options.
4. The bathymetry and displacement files for the artificial tsunami are located at http://www5.in.tum.de/lehre/praktika/swe_lab_ss13/artificialtsunami_bathymetry_1000.nc and http://www5.in.tum.de/lehre/praktika/swe_lab_ss13/artificialtsunami_displ_1000.nc. Simulate the tsunami with reflecting/wall (swimming pool) and outflow (ocean) boundaries.
5. Compare the results from `SWE_ArtificialTsunamiScenario` and `SWE_TsunamiScenario`.

3 Checkpointing

Checkpointing mechanisms save the current state of an application allowing the user to continue a paused or crashed application without the need to recompute already simulated time steps. The application has to save the current state in regular intervals (checkpoints). A checkpoint has to contain all information that is required to restart the simulation.

The netCDF output files from Chapter 1 already contain the important data (domain size, `b`, `h` and momentums). However, you need additional information, e.g. boundary conditions.

Tasks

1. Extend the netCDF writer to provide all information required for restarting the simulation.
2. Create a scenario that can load the netCDF checkpoint file. The scenario should use the last available time step in the file.

³<https://www.unidata.ucar.edu/software/netcdf/docs/netcdf/ncgen.html>

3. Test your checkpointing. To simulate a crash, you can kill SWE pressing CTRL-C in the shell. If your netCDF file does not contain any time steps, you might want to take a look at `nc_sync`⁴.
4. (Optional) Extend SWE such that it automatically uses the checkpoint scenario if the output file already exists and contains at least one time step. In all other cases, it should use the scenario from Chapter 2. This allows you to continue the simulation with the same command line parameters you used for the first start.

4 Tsunami Simulation

Bathymetry and displacement files for the 2011 Tohoku Tsunami⁵ and the 2010 Chile Tsunami⁶ events are located at http://www5.in.tum.de/lehre/praktika/swe_lab_ss13/scenarios/. You should be able to load them with the netCDF scenario from Chapter 2. To get the best results, the epicenter of the earthquake is projected to the origin (0,0) of the computational domain.

1. Simulate both Tsunamis and visualize the output! Use 1000 x 1000 cells and outflow boundaries for your simulation.
 - How long (simulated time) do you have to run the simulation until the main wave leaves your domain?
 - What is the size of a cell (in km) in your simulation?
2. Sōma is a town in Japan about 54.6 km north and 127.6 km west of the 2011 epicenter. We are interested in the time between the earthquake and the arrival of the first Tsunami waves in Sōma.
 - Find the observation data for Sōma for the 2011 Tohoku earthquake.
 - A cut (Figure 5) of the bathymetry is located at http://www5.in.tum.de/lehre/praktika/swe_lab_ss13/scenarios/soma_epicenter_tohoku_2011_bathymetry.csv. `Points:0` and `Points:1` are the x and y coordinates and z is the corresponding value for the bathymetry. The cut contains Sōma and the epicenter. Use the following rule of thumb for the wave speed λ to approximate the time a wave requires to travel from the epicenter to Sōma.

$$\lambda \approx \sqrt{gh} \tag{7}$$

- Compare the results with the arrival of the Tsunami in your simulation.

⁴http://www.unidata.ucar.edu/software/netcdf/docs/group__datasets.html#gaf2d184214ce7a55b0a50514749221245

⁵<http://nctr.pmel.noaa.gov/honshu20110311/>

⁶<http://nctr.pmel.noaa.gov/chile20100227/>

⁷The GEBCO_08 Grid, version 20100927, <http://www.gebco.net>

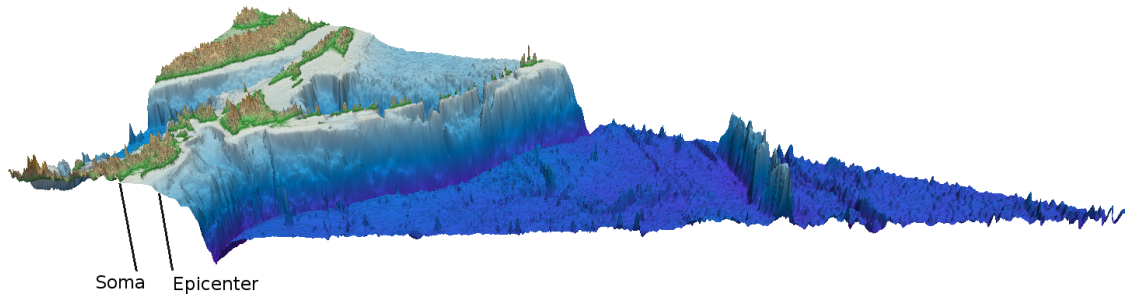


Figure 5: Bathymetry cut through Sōma and the epicenter⁷

Deliverables

The following deliverables have to be handed in no later than 08:00 AM, Monday, 02nd December, 2014. Small files (<1 MB in total) can be send as an attachment directly to *breuera AT in.tum.de*, larger files have to be uploaded at a place of your choice, e.g. <https://github.com/>, <http://home.in.tum.de/>, <https://www.dropbox.com>. In either case inform us about the final state of your solution via e-mail.

- Code in a git-repository. Doxygen documentation and unit tests are mandatory.
- Slides for the presentation during the next meeting; remember to address all questions in the tasks
- Pictures and animations of all runs.
- Documentation how to build and use your code, especially if you extend the SCons-script.
- The netCDF output of the simulations in Chapter 1 and 2.
- A checkpoint file that can be used to restart your application.

Hints

- You can use the command `ncdump -h <file.nc>` to display available variables and dimensions in a netCDF file.