# Delft3D Open Source Workshop

***Delft, November 02, 2017***

## Exercise I: Download the open source code and compile it

Be sure that you have already registered yourself at oss.deltares.nl and download the latest tagged version of the source code (using TortoiseSVN) from
**https://svn.oss.deltares.nl/repos/delft3d/tags**

Software needed for download and compiling, see
http://oss.deltares.nl/web/delft3d/source-code#prerequisites

For compiling (Windows and Linux), see "...\src\README"

After compiling, you will have a new folder "…\bin", containing the compiled binaries.

For more information about compiling, see the webminar given by Adri Mourits on 9 February 2012 and Fedor Bart on 11 January 2012.
http://oss.deltares.nl/web/delft3d/webinars

# Exercise II: Run the preset examples in directory "...\examples"

There are 11 examples in the folder:
**01_standard**: a 3D FLOW model, optionally with parallel computation.
**02_domaindecomposition**: a 2D domain decomposition FLOW model containing 3 subdomains, connected by a ddbound file named "zzz.ddb".
**03_flow-wave**: a 2D FLOW model with morphology and sediment transport, running online with a WAVE computation, optionally with parallel computation
**04_fluidmud**: a 3D FLOW model of the water phase, running online with a 2D FLOW model of the mud phase.
**05_mormerge**: a 2D domain decomposition FLOW model containing 2 subdomains with morphology and sediment transport, running online with a WAVE computation. This model is run several times in parallel with different boundary conditions. Each half time step, the bedlevel changes of all conditions are averaged. All condition runs use these averaged bedlevel changes.
**06_delwaq**
**07_wave**
**08_part-tracer**
**09_part-oil**
**10_delwaq-part-tracer**
**11_standard_netcdf**: same model as 01_standard, producing output in NetCDF format

You can work on the following assignments:

**01_standard**
1. Go to folder: "...\examples\01_standard\" and execute the file "run_flow2d3d.bat"(Linux: "run_flow2d3d.sh"). The simulation will start (and will run for a few seconds).
2. Open the result files using QuickPlot. Try to make an animation for velocity vector.
3. Parallel calculation:
Windows only: Open file "run_flow2d3d_parallel.bat" in a text editor. Follow the instructions about "smpd" as described in the comments at the top of the file.
Both Windows and Linux: Execute the file "run_flow2d3d_parallel.bat" (Linux:" run_flow2d3d_parallel.sh"). The parallel simulation will start (and will run for a few seconds). On Windows, the number of partitions is choosen to be the same as the number of cores in your machine. But the maximum number of partitions for this example is limited to 5, because the model is too small to split into 6 or more partitions. Check the NPROC related lines at the end of file "run_flow2d3d_parallel.bat".
4. You may check if the parallel run provides the same output as the sequential run.
5. Check the diagnosis file(s).

**02_domaindecomposition**
1. Go to folder: "...\examples\02_domaindecomposition\" and execute the file "run_flow2d3d.bat" (Linux: "run_flow2d3d.sh"). The simulations will start (and will run for a few seconds).
2. Open the result files using QuickPlot. Try to make an animation for velocity vectors.
3. Try to run each single domain separately.
4. Check the diagnosis file(s).

**03_flow-wave**
1. Go to folder: "...\examples\03_flow-wave\" and execute the file "run_flow2d3d.bat" (Linux: "run_flow2d3d.sh"). The simulation will start (and will run for a few minutes).
2. Open the result files using QuickPlot. Try to make animations for velocity vector, wave height, wave vectors, morphology change and so on.
3. You may check how the model performs without wind and/or wave.
4. How does the morphology looks like when using a morphological factor of 70?
5. Check the diagnosis file(s).
6. Try to run "run_flow2d3d_parallel.bat".

**04_fluidmud**
1. Go to folder: "...\examples\04_fluidmud\ " and execute the file "run_flow2d3d.bat" (Linux: "run_flow2d3d_flm.sh"). The simulations will start (and will run for a few minutes).
2. Open the result files using QuickPlot. Try to make animations for velocity vector and so on.
3. Check the diagnosis file(s).

**05_mormerge**
1. Go to folder "…\examples\05_mormerge\merge" and execute the file "run_flow2d3d_wave_mormerge.bat" (Linux: "run_flow2d3d_wave_mormerge.sh"). Have a look in "…\examples\05_mormerge\merge\sync" and "…\examples\05_mormerge\0deg" during the calculation.
2. Compare the bedlevel in "…\examples\05_mormerge\0deg" and "…\examples\05_mormerge\45deg"; they should be identical.
3. Only the two conditions "0deg" and "45deg" are enabled. Open file "…\examples\05_mormerge\merge\basin_windows.mm" in a text editor and try to enable the other 4 conditions.

**06_delwaq**
This testcase is handled during the "Water Quality and Ecological modeling" symposium.

**07_wave**
1. Go to folder "…\examples\07_wave" and execute the file "run_wave.bat" (Linux: "run_wave.sh").
2. Investigate the result files using QuickPlot.
3. During the calculation, the following line appears on the screen:
```
Number of threads during execution of parallel region =  4
```
This means that SWAN splits the calculation into 4 partitions, using all cores on the machine. Open file "…\bin\win32\swan\scripts\swan.bat" (Linux: "…\bin\lnx\swan\scripts\swan.sh") in a text editor and remove "rem " in front of line 8: rem set OMP_NUM_THREADS=1
(Linux: remove "# " in front of line 70:  # export OMP_NUM_THREADS=1).
What is the effect?

**08_part-tracer**
This testcase is handled during the "Water Quality and Ecological modeling" symposium.

**09_part-oil**
This testcase is handled during the "Water Quality and Ecological modeling" symposium.

**10_delwaq-part-tracer**
This testcase is handled during the "Water Quality and Ecological modeling" symposium.

**11_standard_netcdf**
1. Go to folder: "...\examples\11_standard_netcdf\ " and execute the file "run_flow2d3d.bat" (Linux: "run_flow2d3d.sh").  The simulations will start (and will run for a few minutes).
2. Open the result NetCDF files using QuickPlot.  Try to view the water level. It might be necessary to specify the clipping value zero (0) for X and Y.

**Additional exercise**
Run (one of) the examples via the Delft3D-GUI, using the freshly compiled binaries. See:
http://oss.deltares.nl/web/delft3d/source-code#Run a calculation
section 5.2.
Check in the tri-diag file that the correct binaries are used.

# Exercise III: Add a new sediment transport formula

As an example, we choose the Nino&Garcia (1998) formula:

$$q^* = \frac{12}{\mu_d}(\tau_* - \tau_{*c})(\tau_*{}^{0.5} - 0.7 * \tau_{*c}{}^{0.5})$$

which is obtained by fitting to uniform size sand and gravel bedload data by saltation particles.

$\mu_d$ : Dynamic friction coef is 0.23 (3 times smaller than the value proposed by Bagnold);

$q^*$ : Dimensionless volumetric sediment transport rate, defined as: $q^* = \frac{q}{d\sqrt{g\Delta d}}$;

$\tau_*$ : Dimensionless shear stress (shields parameter), defined as: $\frac{\tau_b}{(\rho_s - \rho)gd}$;

$\tau_{*,c}$ : critical shear stress, defined as: $\frac{\tau_{b,cr}}{(\rho_s - \rho)gd}$;

You can work on the following assignments:

1. Go to the folder "…\src\utils_gpl\morphology\packages\morphology_kernel\src", copy file *tranb7.f90* to the new file *tranb21.f90*, since there are already 20 other formulas in Delft3D.

2. Add the new tranb21.f90 to the project "morphology_kernel" inside VisualStudio: right click the "Source Files" folder in the solution explorer and add an existing item: *tranb21.f90*.

3. Edit *tranb21.f90*. Firstly, change the subroutine name to *tranb21* and the last row, end of subroutine name to *tranb21*. Secondly, check the input/output parameters. Since here we are aiming to a new sediment transport formula, thus the output parameter is *sbot*. Thirdly, add the formula in the code. Fourthly, check if all the variables used in the formula are defined or being passed through the parameter list.

4. Edit "…\src\utils_gpl\morphology\packages\morphology_kernel\src\eqtran.f90", where the *tranb21* is called. Add a line of code like the following:

```
elseif (iform == 21) then
   call tranb21(utot, di50, d90, h1, par, sbot, ssus)
```

5. Edit "…\src\utils_gpl\morphology\packages\morphology_io\src\rdtrafrm.f90" for input. In subroutine traparams, add code like the following:

```
elseif (iform==21) then
   name       = 'Nino & Garcia (1998)'
   nparreq    = 2
   parkeyw(1) = 'rksc'
   pardef(1)  = 0.01_fp
   parkeyw(2) = 'mud_d'
   pardef(2)  = 0.46_fp
```

… The parameters are read from the addition parameter file: flume_new.tra, which is specified in the mdf file with keyword: TraFrm = #flume_new.tra#.

6. Compile your code.

7. Prepare the model files (flu.mdf and tra files).

8. Run the model and check the model results using QuickPlot.

Use the QuickPlot tutorial and try to make an animation in which you plot bed level at water level points and water level as function of time.

Change the value of $\mu_d$ in the tra file. Check how the model behaves with different values.

Change the transport formula into the Engelund Hansen formula. Check the model results.