Contents

Chapter 1

Examples

This documents aims at explaining how to use the shallow water program on the provided simple examples.

1.1 Prerequisites

On Windows the program NirCmd is required to handle the Gnuplot windows and can be downloaded from the Web.

Gnuplot is called inside the Fortran program in order to plot the convergence of the solution in case of steady flows. Gnuplot mush be callable from the Terminal and mush therefore be mentioned in you environment variable PATH. The data to plot and the Gnuplot commands are written in local files through subroutines located in the file *SRC/gnufor.f90*. Then Gnuplot is executed, it reads its parameters and data and displays a new window with the convergence of the error. If you encounter any difficulty with Gnuplot, you can still comment the code in *SRC/runge_kutta.f90* that calls *write_xyy_data*, *write_xyy_plots* and *run_gnuplot*.

1.2 Oblique hydraulic jump

1.2.1 Generation of the mesh

The geometry of the flow is provided in the file *oblique_jump_geom.geo*. This Gmsh (see [1]) contains the geometry and the data required to build an uniform mesh.

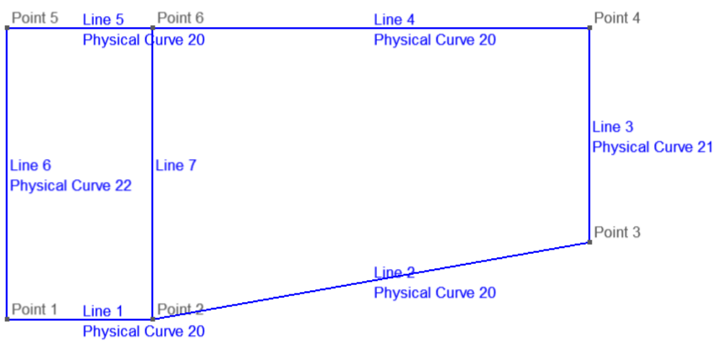

Figure 1.1: Oblique hydraulic jump - Geometry as displayed in the software Gmsh.

The generated mesh looks like

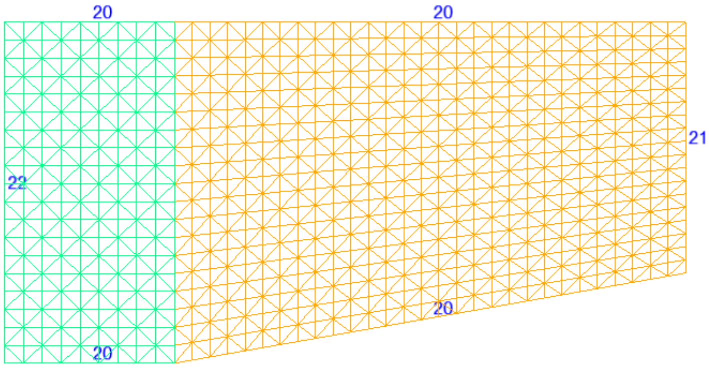

Figure 1.2: Oblique hydraulic jump - Uniform mesh as generated by the software Gmsh.

The user can change the number of elements by modifying the value of the parameter *Vp* (which corresponds to the number of nodes along the vertical direction). The number of nodes along the horizontal direction is automatically adjusted by the parameters *Hp1* and *Hp2*.

1.2.2 Initial and boundary conditions

As shown in Fig. 1.4, physical entities were created in order to join lines having the same boundary condition. The inlet has the physical tag 22, the outlet has the physical tag 21 and the horizontal walls have the physical tag 20. These tags were referenced in the source file *SRC/Build_initial_condition.f90* to create the initial height, velocity, bathymetric depth, inlet depth and velocity. Feel free to modify this file to your desired values. The physical tags are also referenced in the parameter file *parameters_oblique_shock* in order to create a link between the physical tags and the actual boundary type.

```
3                               : number of boundary types, the next 3 lines
22                              : Inlet, physical tag in gmsh
21                              : Outlet, physical tag in gmsh
20                              : Wall, physical tag in gmsh
```

Take care that during the generation of the initial conditions, the parameter *name of mesh file* must be set to the mesh file.

The program *EXE/build_initial_solution* is then launched to read the mesh and create a new Gmsh .msh file that contains the initial values and boundary conditions.

```
EXE\build_initial_solution.exe parameters_oblique_shock oblique_jump_uni_init.msh
```

The resulting Gmsh file that contains the initial values and boundary conditions looks like

1.2.3 Computation of the flow

The parameter *name of mesh file* in the file *parameters_oblique_shock* must be changed to the file that contains the initial values and boundary conditions. The computation is then launched by the command

```
EXE\shallow.exe parameters_oblique_shock
```

A Gnuplot window appears to show the convergence of the solution and the solution is written in the Gmsh file *oblique_jump_uni_sol.msh*

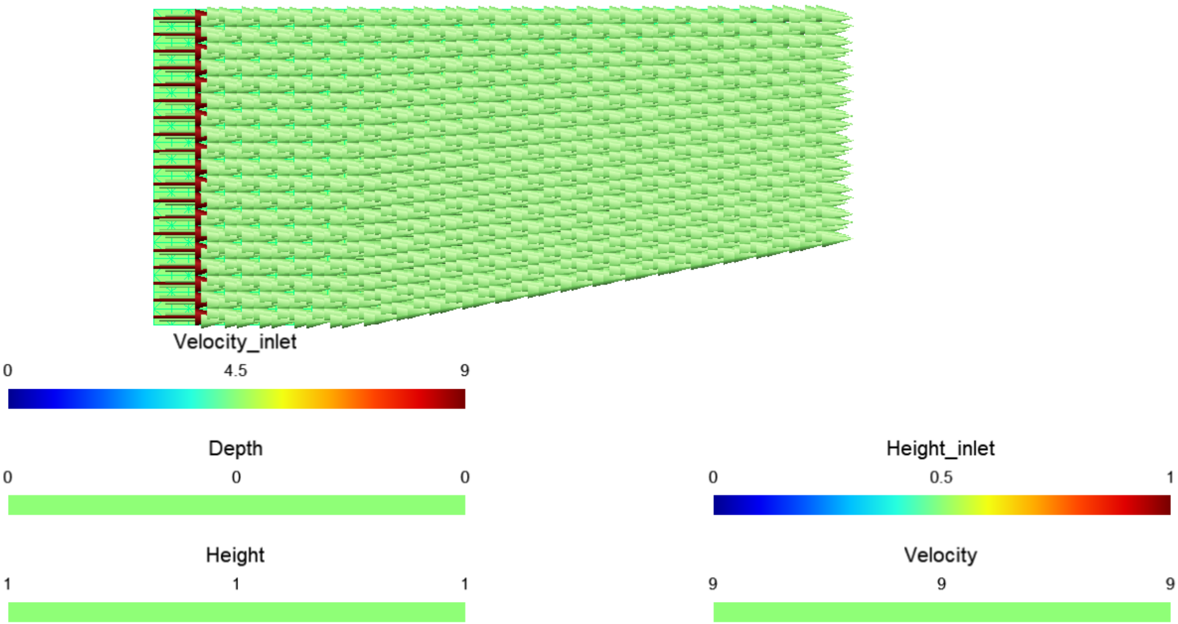

Figure 1.3: Oblique hydraulic jump - Initial values and boundary conditions.

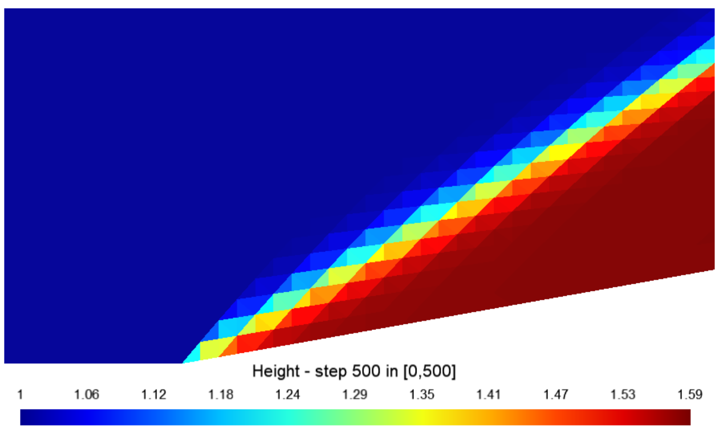

Figure 1.4: Oblique hydraulic jump - Solution after convergence.

Bibliography

[1] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.