Contents

Chapter 1

Examples

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

1.1 Prerequisites

1.1.1 Gnuplot

Gnuplot is called inside the Fortran program in order to plot the convergence of the solution in case of steady flows. If you activated the parameter *Have_ Gnuplot* in the CMake configuration file CMake.config, then Gnuplot mush be callable from the Terminal and must therefore be mentioned in your 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*. When Gnuplot is executed, it reads its parameters and data and displays a new window with the convergence of the error.

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

1.1.2 Sigwatch

Fortran-sigwatch is an external library of routines to provide simple signal watching for Fortran programs. This allows a minimal level of control of a running program from outside it, for example to tell it to checkpoint itself on receipt of a signal. If you compiled it beforehand then you can link the sigwatch library to the Fortran program shallow in the CMake configuration file CMake.config (parameter *SIGWATCH_ LIBRARIES*)

1.1.3 OpenMP

If you compiler supports OpenMP parallelism, then you can activate the parameter *Have_ OpenMP* in the CMake configuration file CMake.config. The number of threads on which OpenMP will run is given by the environment variable *OMP_ NUM_ THREADS*.

Chapter 2

Straight channel with slope and bottom friction

This example computes the super-critical flow inside a straight channel of width $B_0 = 2m$, length $L = 1000m$ and of rectangular cross-section. A unit discharge $q = Q/B_0 = 4m^2/s$ is imposed at the inlet, the geometrical slope is $S_0 = 0.002$ and the height imposed at the inlet is $h_i = 0.7m$. This test case corresponds to the steep-slope profile S3 in Section 2.1 of the documentation file *shallow_water.pdf*.

2.1 Generation of the mesh

The geometry of the flow is provided in the file *channel_slope.geo*. This Gmsh (see [1]) contains the geometry and the data required to build an uniform mesh. The length and width of the channel can be changed by the user (parameters L and H). The number of nodes along the $x-$ and $y-$directions are specified by the parameters Lp and Hp respectively.

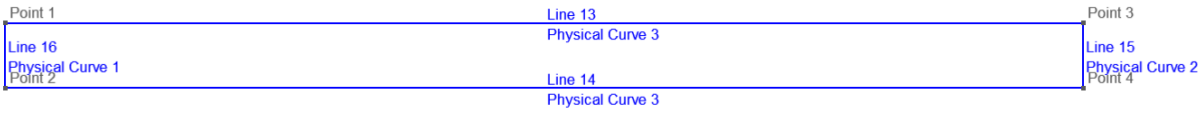

Figure 2.1: Straight channel with slope and bottom friction - Geometry of the flow.

2.2 Initial and boundary conditions

In the previous step, physical entities were created in order to join lines having the same boundary condition. The inlet has the physical tag 1, the outlet has the physical tag 2 and the horizontal walls have the physical tag 3. 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_channel_slope* in order to create a link between the physical tags and the actual boundary type.

```
1 channel_slope.msh                          : name of mesh file
2 3                                : number of boundary types, the next 3 lines
3 1                                : Inlet, physical tag in gmsh
4 2                                : Outlet, physical tag in gmsh
5 3                                : 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. A unit discharge $q = Q/B_0 = 4m^2/s$ is imposed at the inlet, the geometrical slope is $S_0 = 0.002$ and the height imposed at the inlet is $h_{inlet} = 0.7m$. Thus the imposed velocity at the inlet is $u_{inlet} = q/h_i = 5.7143m/s$.

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_channel_slope channel_slope_init.msh
```

2.3 Computation of the flow

The parameter *name of mesh file* in the file *parameters_ channel_ slope* must be changed to the file that contains the initial values and boundary conditions.

```
1  channel_slope_init.msh                          : name of mesh file
2  3                               : number of boundary types, the next 3 lines
3  1                               : Inlet, physical tag in gmsh
4  2                               : Outlet, physical tag in gmsh
5  3                               : Wall, physical tag in gmsh
6  channel_slope_sol.msh                           : name of the output, gmsh format
7  channel_slope_sol.dat                           : name of the output, dat format
8  channel_slope_sol.dat                           : name of the restart file, dat format
9  5000                            : total number of temporal steps
10 500                             : statistics are printed every X steps
11 500                             : solution is saved every X steps
12 9.81                            : g : the gravity constant
13 0.01                            : Manning roughness coefficient (bed)
14 0.0                             : Manning roughness coefficient (wall)
15 1                               : [1] steady or [0] unsteady flow
16 2.4                             : CFL number (used only for steady flows)
17 0.002                           : time step [s] (used only for unsteady flows)
18 0                               : restart from previous simulation [0] no or [1] yes
```

The computation is then launched by the command

```
EXE\shallow.exe parameters_channel_slope
```

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*. The normal depth $y_n = 1m$ is reached asymptotically at the outlet as indicated in the documentation file *shallow_ water.pdf*, Section 2.1.

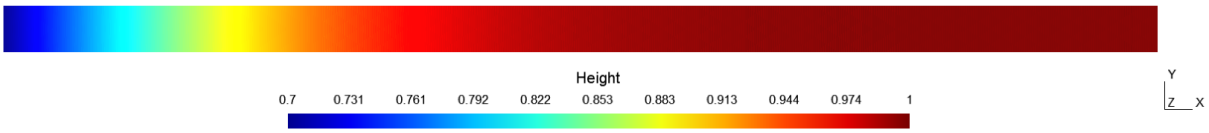

Figure 2.2: Straight channel with slope and bottom friction - Solution after convergence.

Chapter 3

Oblique hydraulic jump

A supercritical flow is deflected inwards by a vertical boundary (angle θ). The user is referred to Section 2.2 of the documentation file *shallow_ water.pdf*

3.1 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. The angle of the deflection is $\theta = 10\,\text{deg}$ and can be changed in this file (parameter *angle*). 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*. The generated mesh looks like

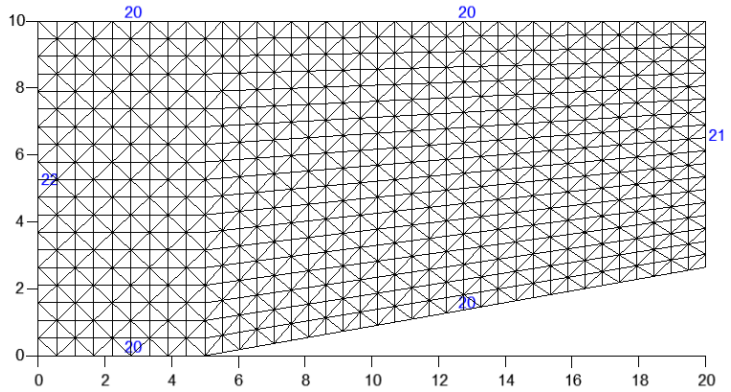

Figure 3.1: Oblique hydraulic jump - Uniform mesh as generated by the software Gmsh.

3.2 Initial and boundary conditions

As shown in Fig. 3.1, 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.

```
1  oblique_jump_uni.msh                          : name of mesh file
2  3                                : number of boundary types, the next 3 lines
3  22                               : Inlet, physical tag in gmsh
4  21                               : Outlet, physical tag in gmsh
5  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. In the present case, the super-critical boundary conditions at the inlet are $h_{inlet} = 1m$ and $U_{inlet} = (9.0, 0.0)m/s$ respectively. These values are also used to initialize the solution field. The Froude number at the inlet is $Fr = 2.8735$.

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
```

3.3 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.

```
1  oblique_jump_uni_init.msh                  : name of mesh file
2  3                                : number of boundary types, the next 3 lines
3  22                               : Inlet, physical tag in gmsh
4  21                               : Outlet, physical tag in gmsh
5  20                               : Wall, physical tag in gmsh
6  oblique_jump_uni_sol.msh                   : name of the output, gmsh format
7  oblique_jump_uni_sol.dat                   : name of the output, dat format
8  oblique_jump_uni_sol.dat                   : name of the restart file, dat format
9  2000                             : total number of temporal steps
10 200                              : statistics are printed every X steps
11 500                              : solution is saved every X steps
12 9.81                             : g : the gravity constant
13 0.0                              : Manning roughness coefficient (bed)
14 0.0                              : Manning roughness coefficient (wall)
15 1                                : [1] steady or [0] unsteady flow
16 2.4                              : CFL number (used only for steady flows)
17 0.001                            : time step [s] (used only for unsteady flows)
18 0                                : restart from previous simulation [0] no or [1] yes
```

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*. The theoretical height after the jump is equal to $h_2 = 1.589m$ as indicated in Section 2.2 of the documentation file *shallow_ water.pdf*.

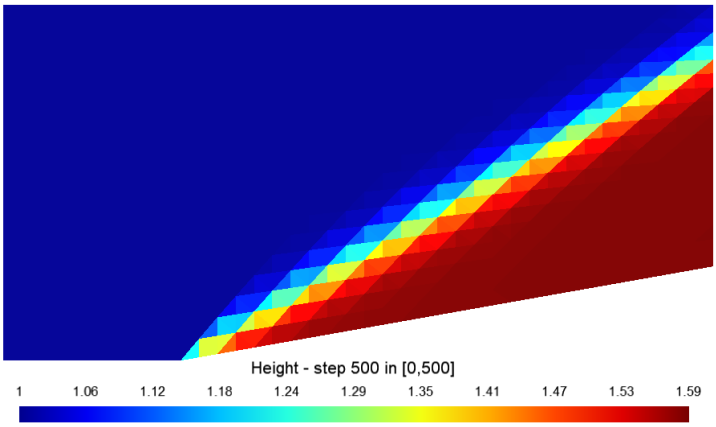

Figure 3.2: Oblique hydraulic jump - Solution after convergence.

Chapter 4

Supercritical flow in symmetrical contraction

This example computes the supercritical flow inside a symmetrical contraction of a channel of rectangular cross-section. This test case corresponds to the supercritical flow described in Section 2.3.1 of the documentation file *shallow_ water.pdf*. The geometry of the channel matches the one described in Lai and Chan [2].

4.1 Generation of the mesh

The geometry of the flow is provided in the file *channel_ contraction_ Lai.geo*. This Gmsh (see [1]) contains the geometry and the data required to build an (non)uniform mesh. The parameter *uniform* allows to switch between an uniform and a nonuniform mesh. Let's keep the parameter *uniform=1* for the moment.

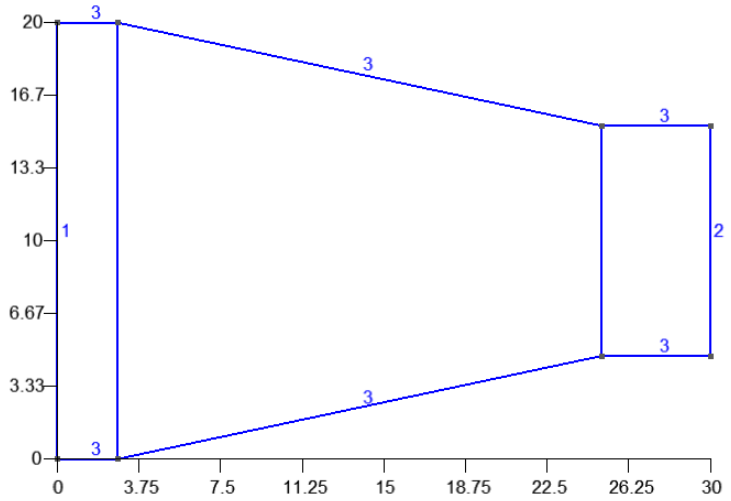

Figure 4.1: Supercritical flow in symmetrical contraction - Geometry of the flow.

The parameters *Vp*, *Hp* and *Hp2* contain the numbers of nodes along the vertical, horizontal middle and horizontal left+right lines respectively. The generated mesh looks like

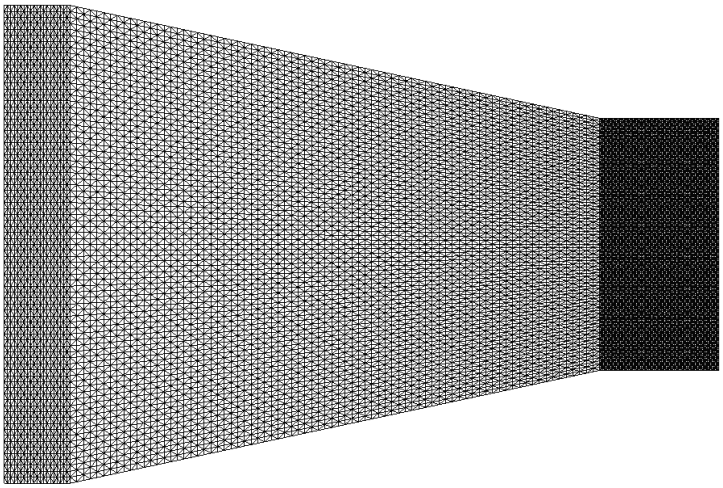

Figure 4.2: Supercritical flow in symmetrical contraction - Uniform mesh.

4.2 Initial and boundary conditions

As shown in Fig. 4.1, physical entities were created in order to join lines having the same boundary condition. The inlet has the physical tag 1, the outlet has the physical tag 2 and the horizontal walls have the physical tag 3. 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_ channel_ contraction* in order to create a link between the physical tags and the actual boundary type.

```
1 channel_contraction_Lai_uni.msh             : name of mesh file
2 3                             : number of boundary types, the next 3 lines
3 1                             : Inlet, physical tag in gmsh
4 2                             : Outlet, physical tag in gmsh
5 3                             : 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. In the present case, the super-critical boundary conditions at the inlet are $h_{inlet} = 1m$ and $Fr_{inlet} = 2.7$ respectively. These values are also used to initialize the solution field. The velocity at the inlet is thus equal to $u_{inlet} = Fr_{inlet}\sqrt{9.81 h_{inlet}} = 8.4567 m/s$.

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_channel_contraction
                                          channel_contraction_Lai_uni_init.msh
```

4.3 Computation of the flow on the uniform mesh

The parameter *name of mesh file* in the file *parameters_ channel_ contraction* must be changed to the file that contains the initial values and boundary conditions.

```
1 channel_contraction_Lai_uni_init.msh             : name of mesh file
2 3                             : number of boundary types, the next 3 lines
3 1                             : Inlet, physical tag in gmsh
4 2                             : Outlet, physical tag in gmsh
5 3                             : Wall, physical tag in gmsh
6 channel_contraction_Lai_uni_sol.msh              : name of the output, gmsh format
```

```
channel_contraction_Lai_uni_sol.dat              : name of the output, dat format
channel_contraction_Lai_uni_sol.dat              : name of the restart file, dat format
4000                            : total number of temporal steps
100                             : statistics are printed every X steps
500                             : solution is saved every X steps
9.81                            : g : the gravity constant
0.0                             : Manning roughness coefficient (bed)
0.0                             : Manning roughness coefficient (wall)
1                               : [1] steady or [0] unsteady flow
2.4                             : CFL number (used only for steady flows)
0.002                           : time step [s] (used only for unsteady flows)
0                               : restart from previous simulation [0] no or [1] yes
```

The computation is then launched by the command

```
EXE\shallow.exe parameters_channel_contraction
```

A Gnuplot window appears to show the convergence of the solution and the solution is written in the Gmsh file *channel_contraction_Lai_uni_sol.msh*. As indicated in Section 2.3.1 of the documentation file *shallow_water.pdf*, the uniform mesh does not capture the correct location of the reflection of the hydraulic jump on the corner. As a consequence, diamond-shaped cross waves are generated in the downstream section of the channel.

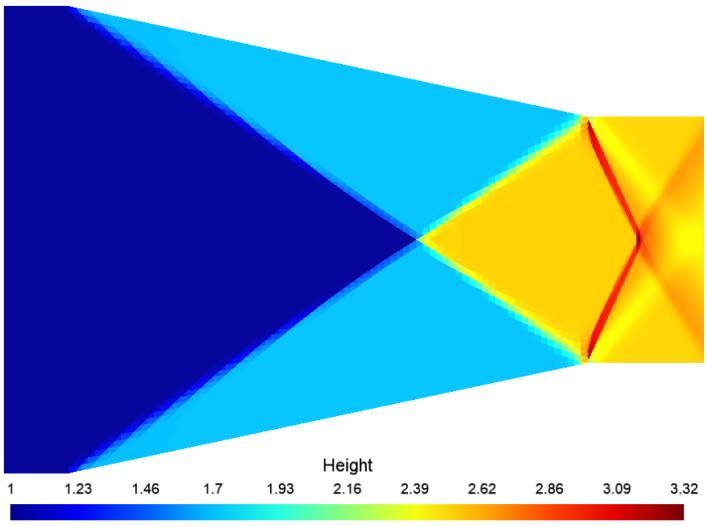

Figure 4.3: Supercritical flow in symmetrical contraction - Solution after convergence on the uniform mesh.

4.4 Computation of the flow on the non-uniform mesh

A new non-uniform mesh can be generated by changing the parameter *uniform* to 1 in the file *channel_contraction_Lai.geo*. A post-processing file .pos is loaded and contains the regions where strong gradients in the solution are present. This view is used to imposed the mesh size in the whole two-dimensional domain. Regions with slow variations will have coarse mesh sizes and regions of strong spatial variation will have small mesh sizes. The generated non-uniform mesh looks is shown in Fig. 4.4.

The converged solution on the non-uniform mesh (Fig. 4.5) captures the correct location of the incoming hydraulic jump on the corner and no diamond-shaped cross waves are generated downstream.

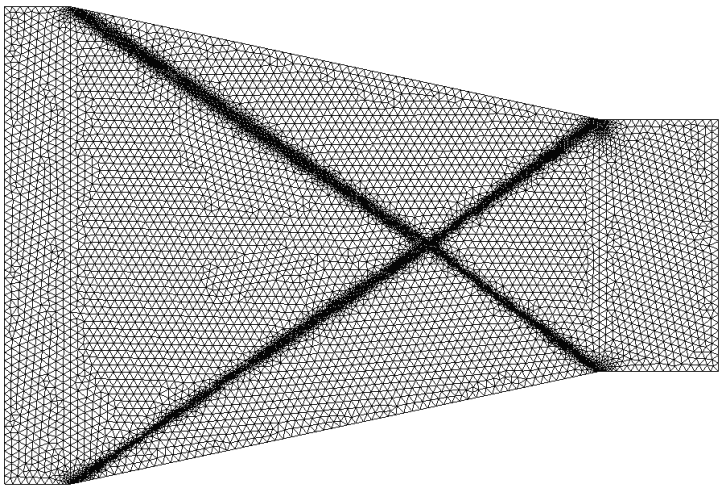

Figure 4.4: Supercritical flow in symmetrical contraction - Non-uniform mesh.

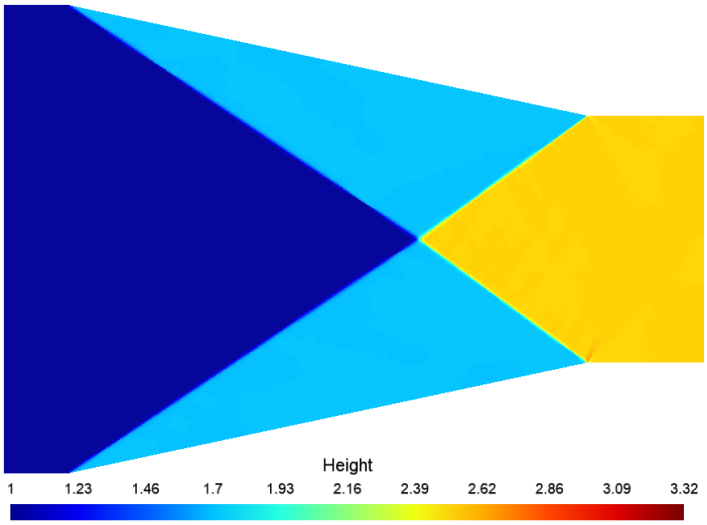

Figure 4.5: Supercritical flow in symmetrical contraction - Solution after convergence on the non-uniform mesh.

Chapter 5

Transcritical flow in long channels

This example computes the transcritical flow inside a symmetrical mild contraction of a channel of rectangular cross-section. This test case corresponds to the supercritical flow described in Section 2.4 of the documentation file *shallow_water.pdf*.

5.1 Generation of the mesh

The geometry of the flow is provided in the file *transcritical_long_channel_Ld500.geo*. This Gmsh (see [1]) contains the geometry and the data required to build a non-uniform mesh. A post view .pos is loaded and aims at positioning the location of the refinement areas where the solution shows fast spatial variations.

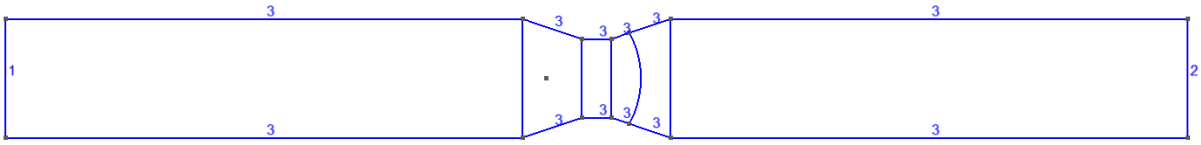

Figure 5.1: Transcritical flow in long channels - Geometry of the flow.

The user can play with the different parameters available in the .geo file in order to modify the geometry or the mesh sizes. The generated mesh looks like

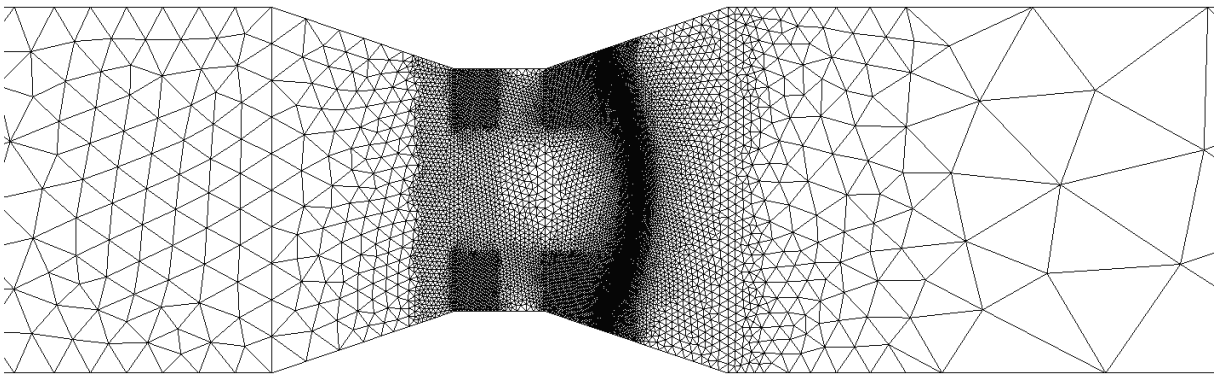

Figure 5.2: Transcritical flow in long channels - Mesh.

5.2 Initial and boundary conditions

As shown in Fig. 5.1, physical entities were created in order to join lines having the same boundary condition. The inlet has the physical tag 1, the outlet has the physical tag 2 and the horizontal walls have the physical tag 3. 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_ transc_ long_ channel* in order to create a link between the physical tags and the actual boundary type.

```
1  transcritical_long_channel_Ld500.msh             : name of mesh file
2  3                               : number of boundary types, the next 3 lines
3  1                               : Inlet, physical tag in gmsh
4  2                               : Outlet, physical tag in gmsh
5  3                               : 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. Subcritical boundary conditions are imposed at the inlet ($h_{inlet} = y_n = 4.543m$, $u_{inlet} = Q/b_1 h_{inlet} = 2,751m/s$) and at the outlet ($h_{outlet} = y_c = 2.516m$).

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_channel_contraction
                                          channel_contraction_Lai_uni_init.msh
```

5.3 Computation of the flow

The parameter *name of mesh file* in the file *parameters_ transc_ long_ channel* must be changed to the file that contains the initial values and boundary conditions.

```
1  transcritical_long_channel_Ld500_i.msh              : name of mesh file
2  3                               : number of boundary types, the next 3 lines
3  1                               : Inlet, physical tag in gmsh
4  2                               : Outlet, physical tag in gmsh
5  3                               : Wall, physical tag in gmsh
6  transcritical_long_channel_Ld500_s.msh               : name of the output, gmsh format
7  transcritical_long_channel_Ld500_s.dat               : name of the output, dat format
8  transcritical_long_channel_Ld500_s.dat               : name of the restart file, dat format
9  10000                           : total number of temporal steps
10 200                             : statistics are printed every X steps
11 1000                            : solution is saved every X steps
12 9.81                            : g : the gravity constant
13 0.04                            : Manning roughness coefficient (bed)
14 0.0                             : Manning roughness coefficient (wall)
15 1                               : [1] steady or [0] unsteady flow
16 2.4                             : CFL number (used only for steady flows)
17 0.002                           : time step [s] (used only for unsteady flows)
18 0                               : restart from previous simulation [0] no or [1] yes
```

The computation is then launched by the command

```
EXE\shallow.exe parameters_transc_long_channel
```

A Gnuplot window appears to show the convergence of the solution and the solution is written in the Gmsh file *transcritical_ long_ channel_ Ld500_ s.msh*. A Matlab/Octave script is also provided (*theory_ channel_ varying_ width_ long_ channel.m*) in order to compute the theoretical curves used as references.

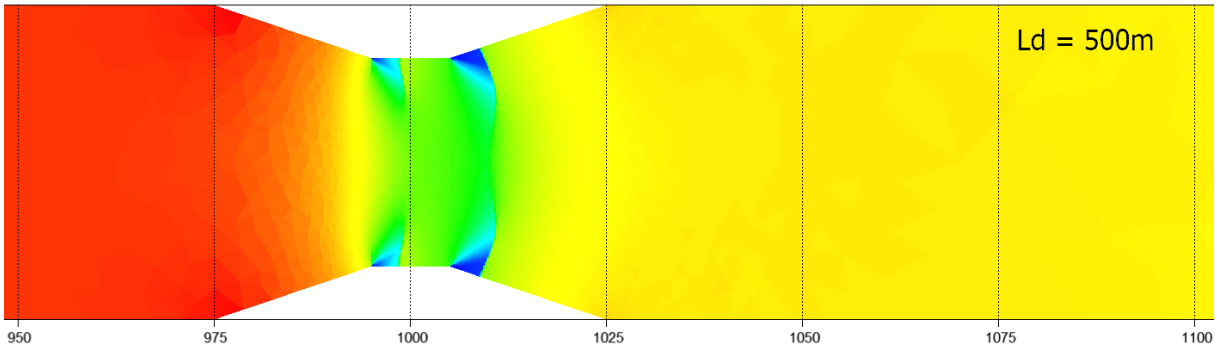

Figure 5.3: Transcritical flow in long channels - Solution after convergence for $L_d = 500m$.

Chapter 6

Partial breach of a dam

This example computes the unsteady flow resulting from the instantaneous partial breach of a dam. This test case corresponds to Section 2.5 of the documentation file *shallow_water.pdf*.

6.1 Generation of the mesh

The geometry of the flow is provided in the file *dam_ref.geo*. This Gmsh (see [1]) contains the geometry and the data required to build an uniform mesh. The user can play with the different parameters available in the .geo file in order to modify the geometry or the mesh sizes. The generated mesh looks like

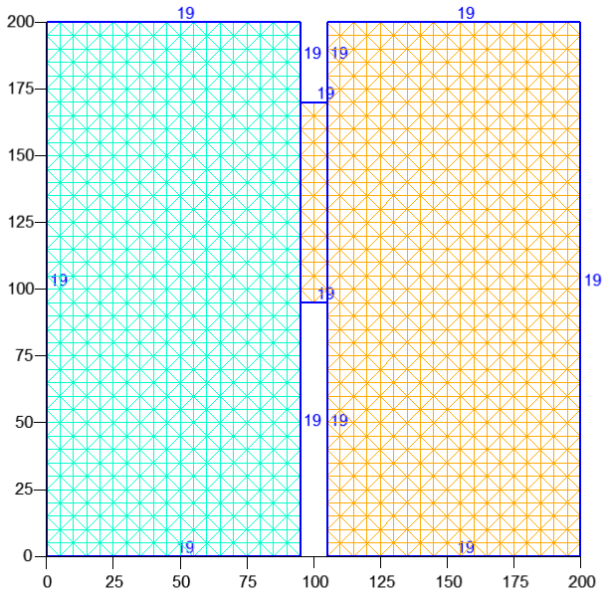

Figure 6.1: Partial breach of a dam - Geometry and mesh of the flow.

6.2 Initial and boundary conditions

The flow is initially at rest and is contained inside the domain without any inflow or outflow boundary. The height is set to $10m$ in the left half of the domain and to $5m$ in the central and right parts of the domain. As shown in Fig. 6.1, physical entities were created in order to join lines having the same boundary condition. The walls have the physical tag 19. This tag

was 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_ dam* in order to create a link between the physical tags and the actual boundary type.

```
1  dam_ref.msh                              : name of mesh file
2  3                                : number of boundary types, the next 3 lines
3  0                                : Inlet, physical tag in gmsh
4  0                                : Outlet, physical tag in gmsh
5  19                               : 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_dam dam_ref_init.msh
```

6.3 Computation of the flow

The parameter *name of mesh file* in the file *parameters_ dam* must be changed to the file that contains the initial values and boundary conditions. Moreover the parameter *[1] steady or [0] unsteady flow* must be set to 0 (unsteady flows do not use local time steps). In the present case, no friction is taken into account (Manning friction coefficient set to 0). The computation is run until the physical time $7.2s$ is reached.

```
1  dam_ref_init.msh                         : name of mesh file
2  3                                : number of boundary types, the next 3 lines
3  0                                : Inlet, physical tag in gmsh
4  0                                : Outlet, physical tag in gmsh
5  19                               : Wall, physical tag in gmsh
6  dam_ref_sol.msh                          : name of the output, gmsh format
7  dam_ref_sol.dat                          : name of the output, dat format
8  dam_ref_sol.dat                          : name of the restart file, dat format
9  720                              : total number of temporal steps
10 20                               : statistics are printed every X steps
11 20                               : solution is saved every X steps
12 9.81                             : g : the gravity constant
13 0.0                              : Manning roughness coefficient (bed)
14 0.0                              : Manning roughness coefficient (wall)
15 0                                : [1] steady or [0] unsteady flow
16 2.4                              : CFL number (used only for steady flows)
17 0.01                             : time step [s] (used only for unsteady flows)
18 0                                : restart from previous simulation [0] no or [1] yes
```

The computation is then launched by the command

```
EXE\shallow.exe parameters_transc_long_channel
```

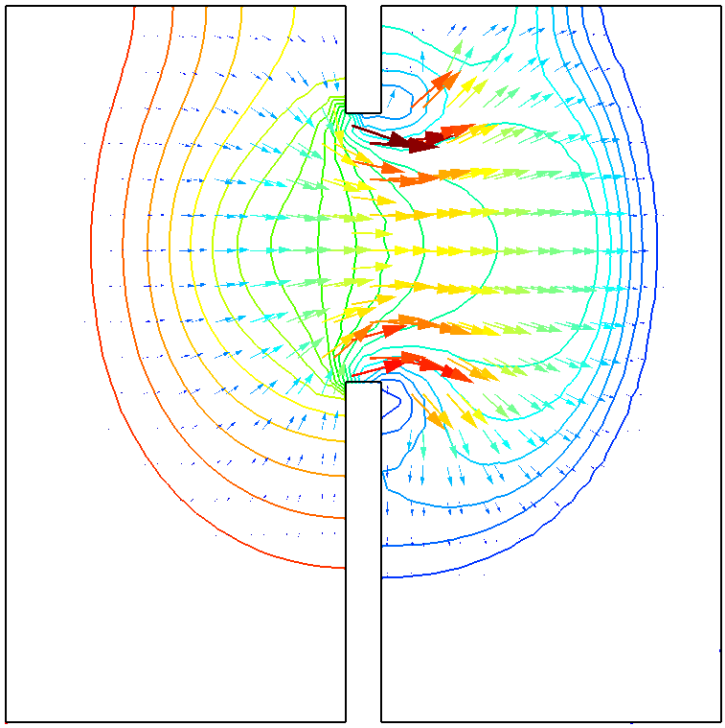

Figure 6.2: Partial breach of a dam - Iso-value lines of the water height and velocity arrays at $t = 7.2s$.

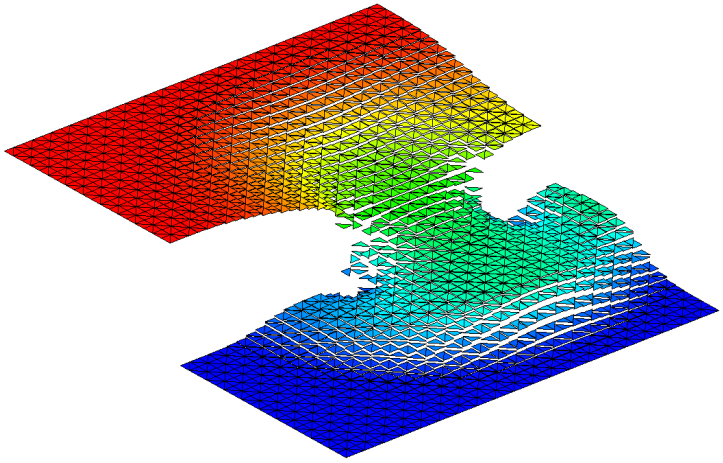

Figure 6.3: Partial breach of a dam - Water surface at $t = 7.2s$.

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.

[2] W. Lai and A.A. Khan. A discontinuous Galerkin method for two-dimensional shock wave modeling. *Modelling and Simulation in Engineering*, Article ID 782832, 2011.