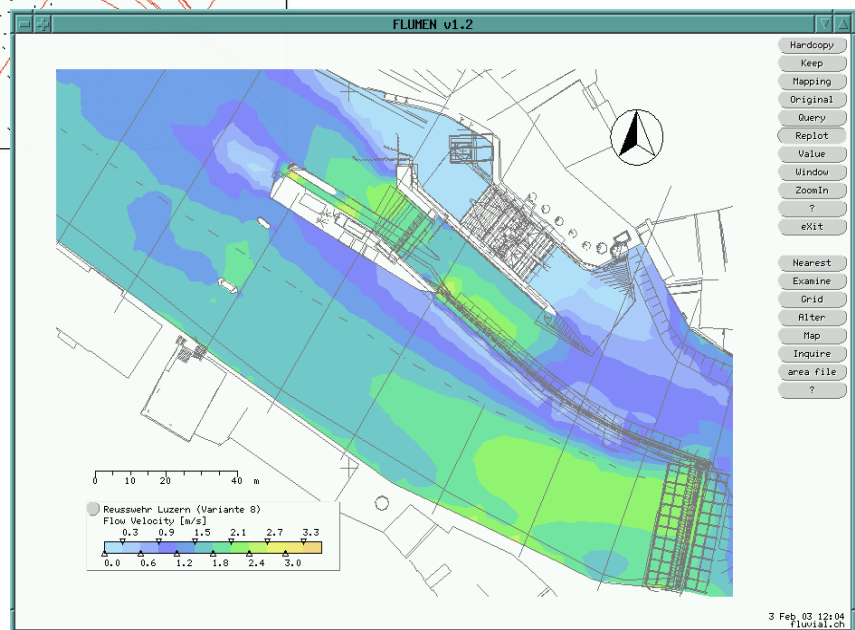
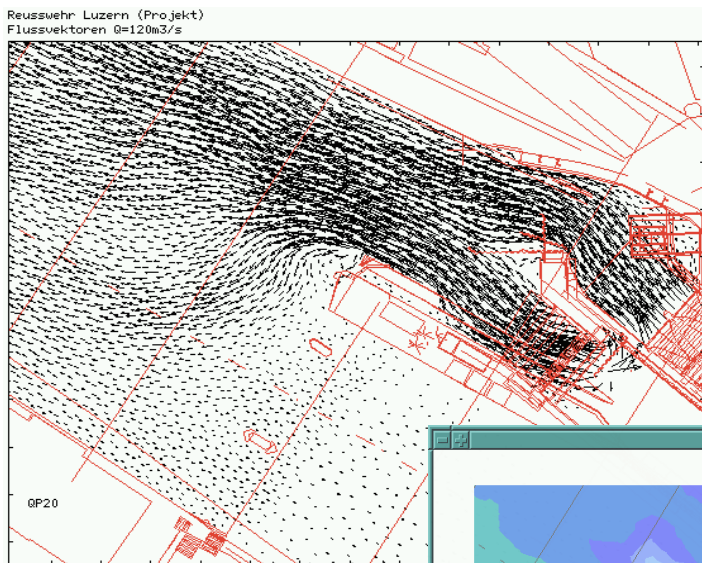


FLUMEN

User Manual



Content

1 Introduction.....	3
1.1 Overview.....	3
1.2 Applications.....	3
2 Basics.....	4
2.1 Governing Equations.....	4
2.1.1 Flow Equations.....	4
2.1.2 Friction Factors.....	4
2.1.3 Initial conditions.....	5
2.1.4 Boundary Conditions.....	5
2.2 Numerical solution.....	5
2.2.1 Finite Volume Method on Triangles.....	5
2.2.2 Numerical Fluxes.....	6
2.2.3 Time Discretization.....	7
3 Using FLUMEN.....	8
3.1 Input Syntax.....	8
3.2 Includes.....	8
3.3 Options.....	9
3.4 Alternatives.....	10
3.5 Running FLUMEN in Batch Mode.....	10
3.6 Generating Quality Meshes.....	10
4 Literature.....	12

1 Introduction

1.1 Overview

FLUMEN (FLUvial Modelling ENgine) is a program for solving the depth-averaged shallow water equations on a cell-centred unstructured mesh. It allows for wet and dry domains, sub- and supercritical flow conditions and the specification of variable bed topography.

The main features of FLUMEN are:

- explicit time integration (1st order)
- numerical fluxes with flux-difference-splitting according to Roe (1st order)
- bed friction with Manning-Strickler formula, logarithmic friction law or full log-law
- mobile bed module to account for deposition/erosion of sediments and bank failure
- allows embedded modelling with 1D river branches
- adaptive meshes

Thanks to the finite volume scheme FLUMEN is extremely robust for mixed sub- and supercritical flow regimes. Conservation of volume is observed.

FLUMEN reads meshes as nodes and elements. For generating high-quality meshes (with no small angles) the preprocessor FLUVIZ together with the program TRIANGLE (Shewchuk 1995) can be used. For a detailed description of the input syntax see the Reference document.

1.2 Applications

FLUMEN can be used for almost all flow modelling purposes where the shallow water equation holds.

- Flood modelling and hazard assessment
- Dam Break Analysis
- Habitat assessment
- Flow analysis
- Design of hydraulic structures

2 Basics

2.1 Governing Equations

2.1.1 Flow Equations

FLUMEN is based on the depth-averaged shallow water equations that can be written in conservation form as

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial G}{\partial y} + S = 0 \quad (1)$$

with x, y = horizontal dimensions, t = time and $U = (h, q, r)^T$ = variable vector with the flow depth h and the components of the specific flow q and r . The flux vectors in x and y are

$$E = \begin{pmatrix} q \\ \frac{q^2}{h} + \frac{g}{2} h^2 \\ \frac{qr}{h} \end{pmatrix}; \quad G = \begin{pmatrix} r \\ \frac{qr}{h} \\ \frac{r^2}{h} + \frac{g}{2} h^2 \end{pmatrix} \quad (2)$$

and the source vector is

$$S = \begin{pmatrix} 0 \\ g h \frac{\partial z_b}{\partial x} + \frac{\tau_{bx}}{\rho} \\ g h \frac{\partial z_b}{\partial y} + \frac{\tau_{by}}{\rho} \end{pmatrix} \quad (3)$$

with g = acceleration of gravity, ρ = density of the fluid, z_b = bed level, and τ_b = bed shear. These equations can be obtained from the Reynolds equations if hydrostatic pressure distribution is assumed. The conservative formulation of the equations remains valid in the presence of discontinuous variations of the flow variables such as hydraulic jumps.

2.1.2 Friction Factors

Bed shear stresses are related to the depth-averaged flow velocities as

$$\frac{\tau_{bx}}{\rho} = \frac{u \sqrt{u^2 + v^2}}{c_f^2}; \quad \frac{\tau_{by}}{\rho} = \frac{v \sqrt{u^2 + v^2}}{c_f^2} \quad (4)$$

with $u = q/h$ and $v = r/h$. The friction factor can be estimated by empirical friction laws:

- Manning-Strickler $c_f = \frac{h^{1/6}}{n \sqrt{g}} = \frac{k_{st} h^{1/6}}{\sqrt{g}}$ with n = Manning's value or k_{st} = k-Strickler value

- logarithmic friction law $c_f = 2.5 \ln \left(\frac{h}{k_s} \right) + 6.0$ with k_s = equivalent sand roughness

2.1.3 Initial conditions

The shallow water equations need initial conditions at the beginning of each simulation run. There are mainly three different types of initial conditions that can be used in FLUMEN:

<i>Type</i>	<i>Description</i>
„Dry Start“	The flow variables (flow depth and velocity) are assumed to be zero over the whole model domain.
„Hot Start“	FLUMEN reads the flow variables from data stored by a previous run.
„Polygon“	The flow variables (flow depth or water level) are defined by closed polygons.

Table 1 - Types of initial conditions in FLUMEN

2.1.4 Boundary Conditions

Boundary conditions influence the solution if the flow reaches a model boundary. If no boundary conditions are defined, FLUMEN assumes non-permeable boundaries, that is, the flow over the boundaries is assumed to be zero. In all other cases boundary conditions must be defined by the modeller. The number of necessary boundary conditions depend on the flow regime (see table below). FLUMEN accepts one boundary condition at each boundary. At an inflow boundary FLUMEN assumes that the inflow is normal to the model boundary and that the inflow is always subcritical.

<i>Flow Regime</i>	<i>inflow boundary</i>	<i>outflow boundary</i>
subcritical	2	1
supercritical	3	0

Table 2 - Number of boundary conditions

At an inflow boundary the discharge must be known (keyword: inflow). FLUMEN distributes the inflow over the boundary edges assuming uniform flow conditions. Experience shows that defining the waterlevel at an inflow boundary often leads to an ill-conditioned system that does not converge. At an outflow boundary it is best to define the slope of the energy head (keyword: slope) or a known waterlevel (keyword: waterlevel). It is also possible to define the outflow discharge if this is physically sound. Time dependent boundary conditions can be defined using time tables (see the Reference).

2.2 Numerical solution

2.2.1 Finite Volume Method on Triangles

FLUMEN uses an unstructured mesh of triangles to discretise the model domain. For a cell centred finite volume

method equation (1) can be integrated over the i th triangle as

$$\int_{A_i} \frac{\partial U}{\partial t} d\Omega = - \int_{A_i} (E_x + G_y) d\Omega + \int_{A_i} S(U) d\Omega \tag{5}$$

where A_i = area of the triangle. Using the divergence theorem the area integral can be replaced by a line integral around the bounding volume

$$A_i \frac{\partial U}{\partial t} = - \oint_{\Gamma_i} F_n(U) ds + A_i S(U) \tag{6}$$

with Γ = circumference of the triangle. The line integral is approximated by the midpoint quadrature rule, that is, the numerical flux is calculated at the midpoint of the edges (Sleigh et al. 1995)

$$\frac{\partial U}{\partial t} = - \frac{1}{A_i} [F_n(U)_{ik} l_{ik} + F_n(U)_{ij} l_{ij} + F_n(U)_{il} l_{il}] + S(U_i) \tag{7}$$

with cell edges as defined in the figure below.

Side ij is the common edge to the triangles associated with U_i and U_j , l_{ij} = length of this side and $F_n(U)_{ij}$ = flux in the outward normal direction evaluated at the midpoint of this edge. As the shallow water equations are rotationally invariant the solution of the two-dimensional equations can be found in applying locally one-dimensional equations with the local coordinate system (\bar{x}, \bar{y}) . If n_x and n_y are the unit normals in the x and y directions respectively such that $n_x^2 + n_y^2 = 1$, the condition for rotational invariance is

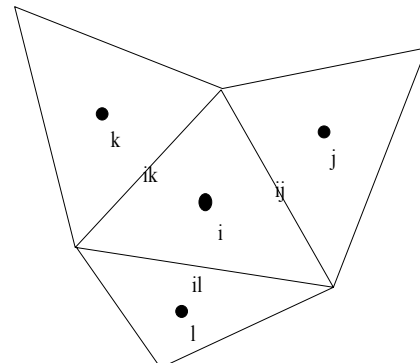


Figure 1 - Triangular Mesh

$$T F_n(U) = E(TU) = E(\bar{U}) \tag{8}$$

where

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_x & n_y \\ 0 & -n_y & n_x \end{pmatrix}$$

The velocities in the local coordinate system are $\bar{u} = u_n = u n_x + v n_y$, and $\bar{v} = u_t = -u n_y + v n_x$ respectively. The flux $F_n(U)$ over each cell face is found from the solution of the one-dimensional equation

$$\frac{\partial \bar{U}}{\partial t} + \frac{\partial E(\bar{U})}{\partial \bar{x}} = 0 \tag{9}$$

2.2.2 Numerical Fluxes

For a central scheme the fluxes over the cell faces are estimated by the values in the cell midpoints. Taking the

arithmetic mean of the adjacent points would lead to spurious oscillations in the solution caused by the decoupling of odd and even grid points (Hirsch 1988). Therefore, different methods have been developed in order to dampen these oscillations. Roe (1981) proposed a flux difference splitting scheme (FDS) which introduces upstream weighting for the flux evaluation. Originally developed for the Euler equations, the flux difference splitting scheme has been shown to have superior properties for the calculation of viscous flows (van Leer et al. 1987). The scheme can also be applied to the system of shallow-water equations, which are essentially hyperbolic (Glaister 1988). Using Roe's scheme the numerical flux across a cell face can be written as

$$F_{ij} = \frac{1}{2} [F(U_L) + F(U_R)] - \frac{1}{2} |A_{ij}| (U_R - U_L) \quad (10)$$

with U_L and U_R = values at the left- and right-hand side of the cell face and A = Jacobian of the flux vector $A = \partial F / \partial U$. Equation (10) differs from the arithmetic mean by an additional flux that is proportional to the difference $\Delta U = U_L - U_R$. The values U_L and U_R can be considered as the states right and left from a discontinuity that leads to the propagation of waves across the cell face. Since the solutions are obtained for the linearized equations, Roe's method defines an approximate Riemann solver (Hirsch 1990). It exactly recognises discontinuities and leads to a high resolution of shocks. The simplest choice for the variables at the cell faces is setting $U_L = U_i$ and $U_R = U_j$ which leads to a scheme whose spatial accuracy is of first order.

2.2.3 Time Discretization

FLUMEN uses an explicit scheme for solving the differential equation

$$\frac{\partial U}{\partial t} = H(U, t) \quad (11)$$

where the vector H can be considered as the differential space operator prior to discretisation (see Equation 1). The simplest choice is the explicit Euler method

$$U^{n+1} = U^n + H^n \Delta t \quad (12)$$

where U^n and U^{n+1} denote the values on the old and new time level, respectively, and $H^n = H(U^n)$. The accuracy of the Euler method is 1st order in time. The explicit Euler scheme has to consider the Courant-Friedrichs-Lewy (CFL) condition that limits the size of the time step. With the CFL numbers in x- and y-direction

$$\sigma_x = (|u| + \sqrt{gh}) \frac{\Delta t}{\Delta x} \quad \sigma_y = (|v| + \sqrt{gh}) \frac{\Delta t}{\Delta y} \quad (13)$$

where Δt = time step, and Δx , Δy = grid size, the CFL condition can be written for TVD schemes as (Hirsch 1988)

$$\sqrt{\sigma_x^2 + \sigma_y^2} < \frac{1}{\sqrt{2}} \quad (14)$$

3 Using FLUMEN

3.1 Input Syntax

The model input is given in free format using predefined keywords. Keywords starting with double arrow allow to structure the input. The input following a >>keyword is related to this >>keyword until the next >>keyword is specified, e.g.

```
>>source
  discharge 10.
  node      9823
>>compute
  start 0.5
  end   1.5
>>
```

A double-arrow without a keyword ends the input reading, i.e. all the input that follows is ignored. Lines starting with a slash (/) in the first column are considered as comments and are ignored. The keywords are given in the same order as they should appear in the input file. If a number has to be specified, *r* denotes a real number and *i* denotes an integer number. String lengths are limited to 32 characters unless explicitly specified.

3.2 Includes

The syntax allows for include-files to divide the model input into separate files.

Example: If you have stored polygon information for the bedlevels on a file called 'bedlevel.inc':

```
bedlevel 452.5
 129412. 457967.
 129445. 457932.
 129452. 458112.
 129423. 458097.
bedlevel 445.75
 130412. 458561.
 130601. 458467.
 130592. 458512.
```

you can refer to this file in your input file with

```
include 'bedlevel.inc'
```

The program then reads from the file 'bedlevel.inc' until an end_of_file marker or the keyword '>>' appears. Then the program continues to read from the input file.

Important: Include-files must always start with a keyword. This means that is not allowed to split a time table into different pieces.

3.3 Options

Options allow to specify different parameters in the input file. Thus, a single input file can be used to describe many different scenarios. Options are invoked with curly brackets {}

```
{option_name
  input_statement 1
  input_statement 2
  ..
}
```

When calling the program the option must be written following the @ sign, e.g.

```
flumen input_file_name @option_name
```

or if there are several options

```
flumen input_file_name @option1,option2,option3
```

Example: We want to run the same model once for the 100 year flood (hq100) or the 300 year flood (hq300). In the input file we write

```
>>boundary
{hq100 inflow 150. }
{hq300 inflow 220. }
```

and start the program with the 100year flood with

```
flumen input_file_name @hq100
```

or for the 300 year flood with

```
flumen input_file_name @hq300
```

It is possible to have options within an other option statement, e.g.

```
{hq100
  {steady
    inflow 150.
  }
  {unsteady
    inflow **
      0.  10.
      1.5 150.
      3.0 10.
  }
}
```

and start the program for the 100 year unsteady flood with

```
flumen input_file_name @hq100,unsteady
```

Important: The option_name must follow directly the open curly bracket without any space in between.

3.4 Alternatives

Alternatives are an extension of the options described above. They are invoked with the character | that replaces the logical “or”. Example:

```
>>boundary
{hq100 inflow 150.
|hq300 inflow 220.
| inflow 50.
}
```

The inflow in the model is either 150. (option @hq100) or 220. (option @hq300) or 50. (no option specified).

3.5 Running FLUMEN in Batch Mode

FLUMEN can be run in batch mode using script files that allows to start different FLUMEN runs one after the other, e.g. if long jobs has to be done overnight or over the weekend. Such a script file could look like this:

```
#script to run Flumen
~/Flumen/flumen in @option1
~/Flumen/flumen in @option2
~/Flumen/flumen in @option3,option4
```

Script files must be executable and their first character is always #. The script above does three FLUMEN jobs with the same input file “in” and different options. To avoid that FLUMEN waits for user input to start the simulation it is important that you tell FLUMEN to run in batch mode, that is, the input file must contain the lines

```
>>compute
batch_mode
```

3.6 Generating Quality Meshes

There are many different ways to create calculation meshes depending on data quality, spatial resolution and structure of the terrain. Often it is useful to distinguish between the original terrain model (DTM) and the calculation mesh that should contain only cells with no small angles. The procedure can be divided into three different steps:

<i>Step</i>	<i>To Do</i>	<i>Tools</i>
1	Creating the Digital Terrain Model (DTM) from given data (e.g. with TRIANGLE)	FLUVIZ for data editing and storing data in .node and .poly formation. TRIANGLE for constrained Delaunay triangulation
2	Creating the calculation mesh	FLUVIZ for generating the model boundary and important break lines. TRIANGLE for quality refinement of the model domain.
3	Mapping of the calculation mesh on the DTM	FLUVIZ for reading in the DTM and mapping the nodes of the calculation mesh.

Table 3 - Generating a quality mesh in three steps

1. Create a TIN of the digital terrain model from given data (points and break lines)
 - a) Use [Fluviz](#) to load and edit the given data. Export data in .node and .poly format for input into [TRIANGLE](#).
 - b) Do the meshing in TRIANGLE with the -cp option, e.g.


```
triangle -cp dtm
```

 if the DTM data is stored in the files dhm.node and dhm.poly (see the manual of TRIANGLE for details).
2. Create the calculation mesh
 - a) Define the boundary of the domain using [Fluviz](#). Export the boundary and the break lines in .node and .poly format
 - b) Create a quality refined mesh using TRIANGLE with the -q option. e.g.


```
triangle -pq mesh
```

 if the boundary and break lines are stored in mesh.node and mesh.poly (see the manual of TRIANGLE for details)
 - c) For local refinement use .area files that describe the maximum area for the triangles.
 - d) To create an area file load the existing mesh into Flumen and edit the values (e.g. of the bedlevels) to define the maximum area for each triangle (negative values define areas that are not refined). Write the nodal values as an area file and give the file the appropriate name (e.g. mesh.1.area). Then start TRIANGLE with the -ra option, e.g.


```
triangle -pqra mesh.1
```

 if the previous mesh is stored in mesh.1.node, mesh.1.poly, and mesh.1.area.
3. Map the mesh points with the DTM data
 - a) Load the TIN nodes and elements into [Fluviz](#), e.g. `fluviz dtm.1.node`
 - b) Use the Map option to map the mesh nodes on the TIN surface, i.e. load the node file (e.g. mesh.1.node) that needs an update of the z-values.

4 Literature

- Beffa, C. (1994). „Praktische Lösung der tiefengemittelten Flachwassergleichungen“. Communication 133, Laboratory of Hydraulics, ETH Zürich.
- Hirsch, C. (1988). „Numerical computation of internal and external flows“. Volume 1: Fundamentals of Numerical Discretization. Wiley, Chichester.
- Hirsch, C. (1990). „Numerical computation of internal and external flows“. Volume 2: Computational Methods for Inviscid and Viscous Flows. Wiley, Chichester.
- Marburger, M. (1999). „Entwicklung eines tiefengemittelten morphodynamischen Fließgewässermodells“. Mitteilungen Heft 8. Wasserbau Universität Kassel.
- Roe, P. L. (1981). „Approximate Riemann solvers, parameter vectors and difference schemes.“ *J. Comp. Phys*, 43, 357-372.
- Shewchuk, J. R. (1995). „Triangle: Engineering a 2D quality mesh generator and Delaunay Triangulator“. Carnegie Mellon University, Pittsburgh, Pennsylvania.
- Sleigh, P. A., Gaskell, P. H., Berzins, M., Ware, J. M., Wright, N. G. (1995). „A reliable and accurate technique for the modelling of practically occurring open channel flows“. Proc. 9th Int. Conf. on Numerical Methods in laminar and turbulent flow, p 881-892. Pineridge Press, Swansea.