

Using SToRM—A Short Primer*

Francisco J.M. Simões
USGS Geomorphology and Sediment Transport Laboratory
4620 Technology Drive, Suite 400
Golden, CO 80403
Phone 303.278.7956 (Fax 303.279.4165)
E-mail frsimoes@usgs.gov
<http://wwwbrr.cr.usgs.gov/gstl>

1. Introduction

The study of natural river changes and the interference of man in natural water bodies is a difficult but important activity, as increasing and shifting populations place more demands on the natural sources of fresh water. Although the basic mechanical principles for these studies are well established, a complete analytical solution is not known but for the most basic cases. The complexity required to describe flow movement and its interaction with its boundaries, which are themselves deformable, have precluded the development of closed form solutions to the governing equations that describe the mechanical behavior of fluid and solid-fluid mixtures. As a result, alternative techniques have been developed to provide quantitative predictions of these phenomena as an aide to engineering projects and river restoration efforts. Numerical modeling is one of such techniques.

Numerical modeling has become very popular in the past few decades, mainly due to the increasing availability of more powerful and affordable computing platforms. Much progress has been made, particularly in the fields of sediment transport, water quality, and multi-dimensional fluid flow and turbulence. SToRM (System for Transport and River Modeling) is a numerical model that uses the two-dimensional depth-averaged equations, also referred to as the shallow water equations (SWEs), to simulate environmental surface waters. Depth-averaged modeling using the SWEs is widely used in rivers, lakes, estuaries, and coastal flows, and has been shown to predict many flow features well. With the advent of fast desktop computers it has become computationally efficient, and the availability of large-scale, high-resolution data collection techniques (such as LiDAR) has matched the models' data requirements. Within this framework, SToRM was developed to simulate flows in natural rivers, with emphasis on wetting and drying effects in rapidly-varying, unsteady flows.

This document provides a short synopsis of basic concepts concerning numerical modeling of surface environmental flow that may be useful when working with SToRM. The first section presents some general aspects about modeling, but the focus of this primer is on practical aspects of numerical modeling germane to SToRM. The presentation is very succinct and contains the very minimum of mathematical equations and jargon. This primer is not meant to substitute for basic texts in hydraulics and numerical methods, and the reader is discouraged from using any computer model, including

* Version 2, 19 May 2013

STORM, without having the proper background in general fluid dynamics, free surface hydraulics, and at least some understanding of the basic principles of calculus and numerical methods.

2. Basic Concepts

STORMm is a computer model. To understand what a computer model is, we must look at the whole discipline of modeling. A model is a representation of a portion of the physical world. In other words, it is an abstraction of it. This is the top level in modeling: it is the most general use of the term “model” which, in this sense, means the knowledge we have of the physical phenomena of interest, which is represented by the laws of physics and by data. In fluvial hydraulics, a model is often used to represent a segment of a river, a lake, or an estuary, and the basic laws of Newtonian fluid dynamics are employed, which describe the behavior of fluids, flow turbulence, sediment transport mechanisms, mixing processes, etc. The data consists of boundary conditions, such as bathymetry, water discharges, sediment particle size distributions, vegetation types, sizes and locations, etc.

The next level of modeling involves the conceptualization of the physical phenomena into its mathematical formulation, which results in the mathematical model. Stepping from the physical phenomena to the mathematical model involves a simplification, whereby the complexity of natural environments is reduced to a limited set of relations between the dependent and the independent variables*. This is accomplished by a process of successive elimination, where the relative importance of the different phenomena is compared and those with the smallest influence are eliminated—or alternatively by starting with an oversimplified view and adding those with the largest influence. Examples of simplifying approximations are steady versus unsteady and one- versus two- versus three-dimensional formulations; simplifying descriptions of turbulence; etc. This process depends on the modeling objectives, on our knowledge of the physical system, on the availability and quality of data, and even on the perception of the modeler(s). The mathematical model is, therefore, composed by a number of mathematical formulae (governing equations) that must be solved under particular sets of initial and/or boundary conditions. In water resources, one usually (but not always) arrives to the set-up of a boundary-value problem whose governing equations contain partial differential equations with nonlinear terms.

The processes employed to solve the governing equations are called the numerical model, which is the next level of modeling. The numerical model can be very simple—the direct solution of the Manning’s equation on the back of an envelope, for example—or very complex. The latter often arises from the need to solve partial differential equations that have no known closed-form analytic solutions, and which require complex numerical techniques and algorithms to solve—such as finite differences, finite volumes, or finite elements. These techniques and algorithms are approximate and do not provide the exact solution to the equations. The quality of the approximation depends on several factors, such as order of discretization, approaches to deal with nonlinear terms, discretization refinement (more about

* The independent variables are the inputs and the forcing quantities, such as boundary conditions and spatial coordinates; the dependent variables are the outputs and effects, such as flow velocity and water surface elevation.

this later), choice of numerical parameters, boundary conditions, and even the experience of the modeler.

The next level of modeling involves the implementation of the numerical model in a computer, and it is called the computer model. The computer model is the result of coding the numerical model in a computer language and is, therefore, software. STORM is a computer model. Often, the terms numerical model and computer model are used interchangeably, but that is not correct. In fact, the same numerical model can be coded differently by different programmers and, therefore, yield different results. For example, this may be due to the order in which algebraic operations are implemented, resulting in different truncation errors. Or it may be due to the use of numerical constants of different resolutions (say, hard-coding $\pi = 1.1415962$ instead of using a function such as $4 \times \arctan 1$). Compilers may use different libraries and/or different optimization options. Different operating systems and different hardware may also be the cause for numerical differences among software. And there are coding errors, something that is easy to happen in modern computer models containing thousands of lines of code. The different steps in modeling are summarized in Fig. 1.1.

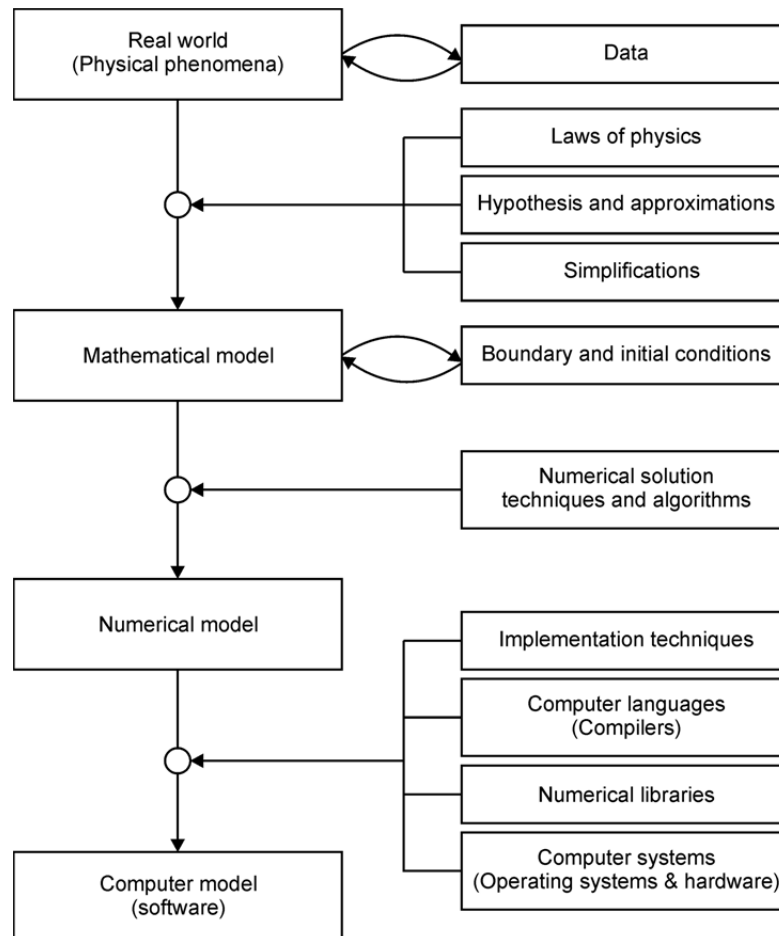


Figure 1.1 The different modeling levels and the factors contributing to each level change.

The final step in modeling involves the use of the computer model by an operator—such as applying SToRM to a particular site of interest. Further approximations and simplifications are introduced in this stage, generally resulting from incomplete knowledge of the physical system, such as imprecise bathymetry, unknown water and/or sediment discharges, friction factors, etc. Furthermore, this step introduces an additional human operator in the system, raising the issues of training (i.e., the results of the simulation project depend on operator skill) and increasing the likelihood of human error.

3. Mathematical Modeling

The flow phenomena in natural rivers are three-dimensional, especially those at or near a meander bend, local expansion and contraction, or a hydraulic structure. Turbulence is an essentially three-dimensional phenomenon, and three-dimensional models are particularly useful for the simulation of turbulent heat and mass transport. When shallow water flows are nearly horizontal, however, the three-dimensional (3D) effects are not essential, and in the long wave approximation (i.e., waves with long characteristic lengths compared to the water depth, such as tidal waves in seas and flood waves in rivers), considerable simplifications can be made to the governing equations and efficient mathematical models can be built, where depth-averaged quantities replace their fully three-dimensional counterparts. These models are useful in many practical applications, where three-dimensional detail is not needed or when the extent of the problem is too large for the available computing power. On the other hand, two-dimensional (2D) modeling constitutes a step up from the much simpler one-dimensional (1D) approach, albeit at a cost of significant additional data requirements. Table 3.1 shows the scales of applicability of the different types of modeling in view of the commonly used current-day computing facilities. In Table 3.1, the number of partial differential equations of the mathematical model is given, with additional differential equations for turbulence modeling in parenthesis.

Table 3.1 Typical scales for the application of surface hydraulics models. The number of nodes provides a measure of the numerical burden associated with discretizing the governing equations, and the number of equations relates to multiplicative computational costs.

| Dimension | Number of nodes | Spatial scale (m) | Time scale | Number of equations |
|-----------|-----------------|-------------------|------------------|-----------------------|
| 1D | $\sim 10^2$ | 10^3 – 10^5 | Up to decades | 2 |
| 2D | 10^4 – 10^5 | 10^3 | Days | 3 (+2 for turbulence) |
| 3D | 10^4 – 10^6 | 10 – 10^2 | Minutes to hours | 4 (+8 for turbulence) |

SToRM uses a 2D depth-averaged mathematical model, which results from averaging the 3D flow equations along the vertical direction and replacing the point velocities and other dependent variables with their depth-averaged counterparts. The depth-averaged velocity components are defined as

$$U = \frac{1}{h} \int_{z_b}^{\eta} u dz \quad \text{and} \quad V = \frac{1}{h} \int_{z_b}^{\eta} v dz \quad (1)$$

where z_b denotes the bed elevation, η is the free surface elevation, h is the water depth, and u and v are the point velocities (see Fig. 3.1 for a sketch of the coordinate system and representation of the variables).

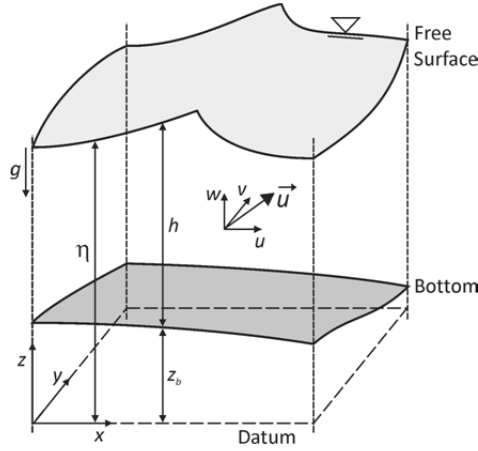


Figure 3.1 Sketch showing the coordinate system used and the definition of some of the variables.

Integrating the continuity equation (conservation of mass), making use of the kinematic condition at the free surface and the fact that the normal component of the velocity must vanish at the solid bed, one obtains

$$\frac{\partial h}{\partial t} + \frac{\partial(hU)}{\partial x} + \frac{\partial(hV)}{\partial y} = 0 \quad (2)$$

The momentum equations can be averaged in the same way. Including the Coriolis and pressure terms, and after further simplification, the depth-averaged momentum equations are

$$\frac{\partial(hU)}{\partial t} + \frac{\partial(hU^2)}{\partial x} + \frac{\partial(hUV)}{\partial y} = F_x + fhV - gh \frac{\partial \eta}{\partial x} - \frac{\tau_{bx}}{\rho_0} + \frac{\partial(h\tau_{xx})}{\partial x} + \frac{\partial(h\tau_{xy})}{\partial y} \quad (3)$$

$$\frac{\partial(hV)}{\partial t} + \frac{\partial(hUV)}{\partial x} + \frac{\partial(hV^2)}{\partial y} = F_y - fhU - gh \frac{\partial \eta}{\partial y} - \frac{\tau_{by}}{\rho_0} + \frac{\partial(h\tau_{xy})}{\partial x} + \frac{\partial(h\tau_{yy})}{\partial y} \quad (4)$$

In the equations above, g is the acceleration due to gravity, f is the Coriolis parameter ($= 2\Omega \sin\phi$, where Ω is the angular rate of Earth's revolution and ϕ is the geographic latitude), F_i are driving forces ($i = x, y$), ρ_0 is the density of water, and τ_{bi} are the bottom stresses ($i = x, y$). They are sometimes called the shallow-water equations (SWE) or the depth-averaged Navier-Stokes equations. The cross-stresses τ_{ij} include viscous friction, turbulent friction, and the nonlinear terms resulting from the vertical averaging process, which are usually called the radiation stresses:

$$\tau_{ij} = \frac{1}{h} \int_{z_b}^{\eta} \left[\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \overline{u'_i u'_j} + (u_i - U_i)(u_j - U_j) \right] dz \quad (5)$$

Where ν is the molecular viscosity and the term $-\overline{u'_i u'_j}$ represents the 3D turbulent effects. In most natural bodies of water, the molecular viscosity terms can be safely neglected in comparison with the turbulence terms. The radiation stresses are often neglected, but they represent important physical phenomena. For example, in bends they are at least partly responsible for shifting the high velocity part of the flow profile from the inner bank at the upstream region to the outer bank at the downstream region of the bend. In general, however, these terms are collapsed in the form of diffusion coefficients and eq. (5) is written as

$$\tau_{ij} = h_i D_i \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{1}{2} \delta_{ij} \frac{\partial U_k}{\partial x_k} \right) \quad (6)$$

where δ_{ij} is the Kronecker delta (= 1 if $i = j$, 0 otherwise), and D_i is a diffusion coefficient in the i^{th} direction (in general, $D_1 = D_2 = D_H$). For a detailed derivation of the SWE see, for example, D.W. Pritchard, "Hydrodynamic models, in Estuarine Models: An Assessment," Tracor, Inc, for the Water Quality Office of the Environmental Protection Agency, 1971, 33 p.

In turbulent flow, the diffusion coefficients can be prescribed or computed from any of the many existing turbulence models, and the bottom shear stresses are assumed to have the same direction of the depth-mean velocity and proportional to the square of its magnitude:

$$\frac{\tau_{bx}}{\rho_0} = c_f U \sqrt{U^2 + V^2} \quad \text{and} \quad \frac{\tau_{by}}{\rho_0} = c_f V \sqrt{U^2 + V^2} \quad (7)$$

where c_f is the standard friction (drag) coefficient ($c_f \approx 0.003$). Note that the bed shear stresses can be also written in terms of the Manning's roughness coefficient, n , or in terms of Chézy's roughness coefficient, C :

$$\frac{\tau_{bx}}{\rho_0} = \frac{gn^2}{h^{1/3}} U \sqrt{U^2 + V^2} = \frac{g}{C^2} U \sqrt{U^2 + V^2} \quad \text{and} \quad \frac{\tau_{by}}{\rho_0} = \frac{gn^2}{h^{1/3}} V \sqrt{U^2 + V^2} = \frac{g}{C^2} V \sqrt{U^2 + V^2} \quad (8)$$

The driving forces remaining in the SWEs include such effects as atmospheric pressure gradients, wind stresses, density gradients, and tidal stresses. The SWEs can be written in many possible forms. Those forms may include different terms than the ones considered above (corresponding to other physical effects), or may be written in terms of curvilinear coordinates, for example. There are many other aspects interest, but that is outside the scope of this monograph.

There are some limitations to this mathematical model. One of the leading assumptions used in the averaging procedure above is that of hydrostatic pressure, which reflects the fact that vertical accelerations (but not the vertical velocity) and shear stresses are neglected in the derivation of the SWE. Some of the vertical motions and accelerations are related to small scale features (e.g., channel

boundary, local topography, and large roughness elements), while others are related to large geomorphologic features (e.g., planform channel curvature and longitudinal bed slope).

In 2D modeling, the vertically non-uniform velocity profiles are replaced by constant velocity profiles. Some examples of different vertical velocity profiles with the same depth-averaged value are shown in Fig 3.2. Because of this limitation, depth-averaged models are not suitable for flows where vertical variations are important, such as in density flows. The inability of representing the vertical non-uniformity of the velocity field is also responsible for weakening the performance of 2D models in predicting flow in channel bends, where flow curvature is important, unless special secondary flow correction models are applied. Without secondary flow correction, 2D models tend to predict the maximum flow velocities to remain at the inside of the bend, instead of the gradual shift from the inside to the outside of the bend that is observed in natural channels. However, bed friction tends to weaken secondary flows and to diminish this effect.

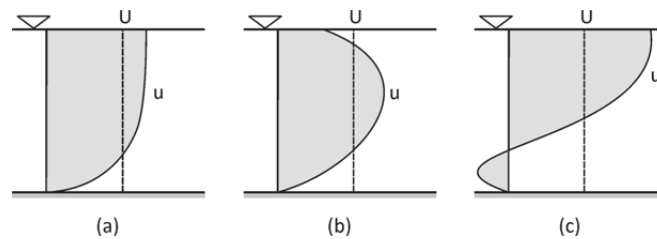


Figure 3.2 Different velocity profiles with the same depth averaged velocity U : (a) logarithmic profile, (b) parabolic profile, and (c) profile with a negative local velocity u , but with positive depth averaged velocity.

Eq. (7) is a Chézy-type equation that is based on an assumption of uniform flow. This is not an unreasonable assumption, because its application is done at the grid cell level and not to the channel as a whole. Unfortunately, although c_f in Eq. (7) is related to its 1D and 3D cousins, it does not have the same meaning and cannot be entirely determined in the same way (i.e., via longitudinal free surface slope in the 1D case, or via the size of the subgrid-scale roughness elements in 3D). A clear and detailed analysis of this matter can be found in Morvan et al. [H. Morvan, D. Knight, N. Wright, X. Tang, and A. Crossley, “The concept of roughness in fluvial hydraulics and its formulation in 1D, 2D, and 3D numerical simulation models,” *Journal of Hydraulic Research*, 2008, 46(2):191–208]. Here, it suffices to say that c_f (or any alternative representation, such as Manning’s n or Chézy’s C) is a quantity that cannot be effectively and unambiguously measured in the field, therefore requiring indirect observation and model calibration using ancillary data, such as free surface water levels at known discharges. Furthermore, note that the friction coefficient for unsteady flow is generally higher than the corresponding steady state friction coefficient, which is due to the fact that the thickness of the boundary layer for unsteady flow is smaller than for steady flow.

In spite of the approximations and limitations described above, 2D river modeling has been shown to provide accurate representations of the flow field for practical applications, especially for mild slopes and large width to depth ratios, which are the conditions prevalent in natural bodies of water.

3.1 Numerical Modeling

This section describes briefly how the equations are solved numerically in STORM. The information presented here is not essential and can be bypassed by the reader that is not interested in details.

STORM solves a subset of eqs. (2)–(4) which, written in matrix form for simplification, are

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{S}(\mathbf{U}) \quad (9)$$

where \mathbf{U} is the vector containing the conservative variables, \mathbf{F} and \mathbf{G} are the inviscid fluxes, and \mathbf{S} is the vector containing the forcing (source) terms. They are the continuity (mass conservation) equation and the horizontal momentum equations (conservation of momentum), with all the terms defined as

$$\mathbf{U} = \begin{bmatrix} h \\ hU \\ hV \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} hU \\ hU^2 + gh^2/2 \\ hUV \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} hV \\ hUV \\ hV^2 + gh^2/2 \end{bmatrix}, \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} 0 \\ F_{wx} + gh(S_{0x} - S_{fx}) \\ F_{wy} + gh(S_{0y} - S_{fy}) \end{bmatrix}$$

In the above definitions S_0 is the bed slope, S_f is the bottom friction, and F_w represents the wind forcing. The bed slope terms are defined by

$$S_{0x} = -\frac{\partial z_b}{\partial x} \quad \text{and} \quad S_{0y} = -\frac{\partial z_b}{\partial y},$$

the bottom friction terms by

$$ghS_{fx} = c_f U \sqrt{U^2 + V^2} \quad \text{and} \quad ghS_{fy} = c_f V \sqrt{U^2 + V^2},$$

and the wind friction terms by

$$F_{wx} = \frac{\rho_a}{\rho_0} c_{fw} U_w \sqrt{U_w^2 + V_w^2} \quad \text{and} \quad F_{wy} = \frac{\rho_a}{\rho_0} c_{fw} V_w \sqrt{U_w^2 + V_w^2}, \quad (10)$$

where U_w and V_w are the components of the wind velocity vector, ρ_a is the air density, and c_{fw} is the wind friction (drag) coefficient at the water surface.

Eq. (9) constitutes a nonlinear, time-dependent system that, for convection-dominated flows, is a hyperbolic system, therefore it can be treated with a wide range of techniques developed for such systems. The system's conservative formulation*, which remains valid across discontinuities in the flow variables such as hydraulic jumps, provides an ideal basis for use by traditional control volume methods, which is the technique used in STORM.

* In conservative form, the equations are expressed in terms of a local time rate of change plus the divergence of a flux, which constitute the terms in the left-hand side of eq. (9). Its advantage is that the equations are easily cast in an integral form that allows discontinuous solutions.

The first step in solving eq. (9) involves integrating it over a control volume. Defining $\mathbf{E} = (\mathbf{F}, \mathbf{G})^T$ as the flux terms and applying Gauss' theorem to the flux integral yields

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\partial\Omega} (\mathbf{E} \cdot \mathbf{n}) ds - \nu_t \oint_{\partial\Omega} (\mathbf{R} \cdot \mathbf{n}) ds = \int_{\Omega} \mathbf{S} d\Omega \quad (11)$$

where Ω is the area of the control volume, $\partial\Omega$ its perimeter, and \mathbf{n} is the outward-pointing unit vector normal to the control volume boundary. In SToRM, eq. (11) is integrated algebraically over finite-sized, polygonal control volumes:

$$\frac{\partial \Omega_i \mathbf{U}_i}{\partial t} = \sum_{k=1}^{n_c} \mathbf{E}_{ik} \Delta l_{ik} + \mathbf{S}_i \Omega_i \quad (12)$$

where \mathbf{U}_i are the average values of the conserved variables over cell i , \mathbf{E}_{ik} are the inviscid fluxes through edge k , \mathbf{S}_i contains the source terms, Ω_i is the cell's area, and n_c is the number of sides of the polygonal control volume. Currently, the scheme adopted in SToRM discretizes the domain in non-overlapping triangles ($n_c = 3$) and the values of the conserved variables are cell-centered. The linear reconstruction operators used, which must satisfy the cell averaging requirements that guarantee cell-wise discrete conservation properties, exploit the fact that the cell average is also a pointwise value of any permitted linear reconstruction evaluated at the centroid of the triangle. The bed elevation above datum, however, is defined at the triangle vertices. Each triangle defines a control volume, which is used to solve eq. (9). The general configuration of the discretization apparatus is shown in Fig. 3.3.

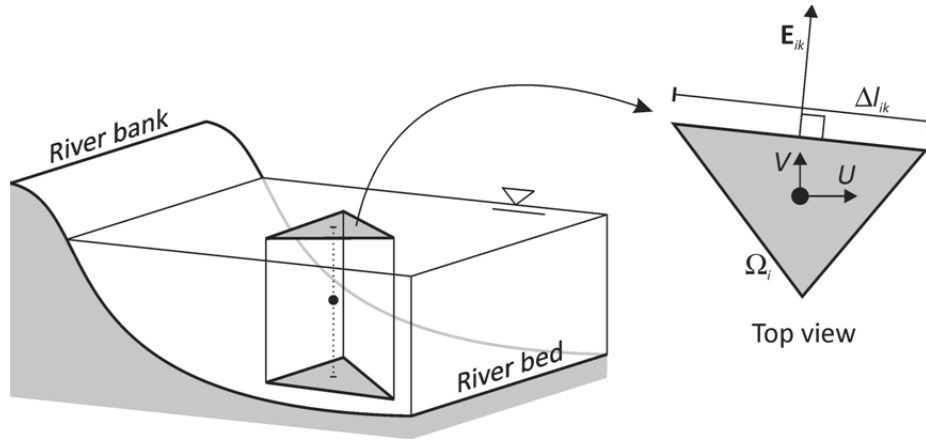


Figure 3.3 General configuration of the triangular control volume formulation used in the numerical model of SToRM.

To avoid long and tedious detail, here it suffices to say that \mathbf{E}_{ik} , which are the numerical fluxes of mass and momentum across the triangles' edges, are computed using approximate Riemann solvers with a MUSCL reconstruction*. Time stepping—left-hand side of eq. (12)—is computed explicitly using

* These are standard terms in numerical modeling of shallow water flows. Details can be readily found on the internet with the help of a search engine.

nonlinear Strong Stability Preserving Runge-Kutta (SSPRK) schemes, also known as Total Variation Diminishing (TVD) Runge-Kutta schemes. There are three different SSPRK schemes implemented in STORM with first, second, and third order in accuracy, respectively. Only the first and second order schemes are recommended for practical use.

To conclude this section, it may be useful to point out that, like the techniques discussed above, the methods most commonly used to solve the fluid flow equations are based on some type of gridded discretization of the problem at hand. In this computational paradigm, the continuous variables for which the solution is sought are solved only at specific discrete locations of the physical domain. The algebraic equations that form the numerical model are functions of those discrete quantities. Depending on the discretization approaches used, however, it is possible to obtain very distinct sets of algebraic numerical equations for exactly the same problem (i.e., for the same set of differential governing equations and boundary conditions). In practice, there are many techniques available to numerically solve the fluid flow equations, such as finite differences, finite elements, and finite volumes, each category itself having many different variations.

This translation from continuous to discrete replaces one problem (the continuous formulation) by another (the discrete formulation). The latter should provide a solution that should converge to the solution of the former. In this context, convergence is a term that denotes a relationship between the numerical and the analytical solutions. Convergence should be obtained as the grid spacing (Δx in one-dimensional problems) and time step (Δt in unsteady problems) are refined, i.e., as $\Delta x, \Delta t \rightarrow 0$.

Convergence is ensured by Lax's theorem—which has been proven for linear problems only, but which is nonetheless at the foundation of computational fluid dynamics—which states that consistence and stability are sufficient to ensure the convergence of a numerical scheme. Consistence is a term applied to the algebraic equations: a numerical scheme is said to be consistent with the original differential equation if its truncation error disappears in the limit $\Delta x, \Delta t \rightarrow 0$. Stability is a term that applies to the numerical solution itself: a solution is stable if it remains bounded at all times during the course of the computations.

4. Data Requirements

In general, the basic data requirements for loose boundary hydraulic models can be grouped in three broad categories: geometric data, hydraulic data, and sediment data. These data sets establish the boundary conditions necessary to solve the governing equations and are an integral part of a model. The term model thus refers to the ensemble of the set of governing equations, their numeric solution technique, its implementation in a computer program, and the data that defines the prototype. In practice, data gathering is the responsibility of the user of the computer program, and often originates from sources unconnected to numerical modeling. However, data collection and preparation is no less important than the computer model itself, and often plays the dominant role in determining the accuracy and applicability of the final numerical solutions generated by the computer.

The geometry data defines the topography of the reach to be simulated, i.e., the channel bed, banks, and flood plains. In two- and three-dimensional models, the data is most often presented as a set of

points given by its x , y , and z coordinates. The data is then interpolated to the locations of the grid nodes used in the discretization of the problem. Interpolation may be done using standard triangulation techniques, or with more sophisticated techniques, such as kriging. In river channels, special techniques are often used that, in the interpolation process, emphasize the contribution of the data points along the flow streamlines, as they have the potential of better capturing the longitudinal characteristics of the river bed (i.e., the meandering of the thalweg).

In one-dimensional models, the geometry is usually defined by cross sections. Each cross section is a line representing a particular section of the modeled reach and is given by a set of points, each defined by a lateral distance and a bed elevation above a common datum. This line provides the information about the section shape and the locations of the subchannels, and should be taken between locations well above the highest stage levels. It should also be taken perpendicularly to the flow streamlines. Additionally, the distance between the cross sections needs to be given, and this distance must be collected along the flow streamlines. Under bankfull conditions, this distance can be taken along the channel's thalweg, but this may vary under flooding conditions. Multi-dimensional models generally use the same type of data, but require much more dense surveys.

Examples of typical data sets collected to match different types of model requirements are shown in Fig. 4.1. In one-dimensional modeling, one is usually interested in having a cross-sectional description of the river reach. The data shown at the left of Fig 4.1 is a typical example of such an approach. The black dotted line was added to indicate the approximate location of the shoreline, delineating the boundary between points with low elevation (in blue) and high elevation (in red). Note the presence of an island. The cross sections somewhat extend past these limits and over the dry island to allow the possibility of modeling flows with higher discharge, where those areas may become submerged.

The survey data shown at the right of Fig. 4.1 is much denser and the points are randomly distributed. The denser data is necessary for two- and three-dimensional modeling, where a more detailed description of the topography is needed. The blue points show the actual wet area of the reach. This part of the survey was done using a boat with sounding equipment, and the path followed by the vessel is reflected in the location of the blue survey points. As before, additional survey data was collected above the water line, as it might be needed for model runs with higher water discharge.

Hydraulic data encompasses the necessary upstream and downstream flow conditions, as well as friction factors and local head losses. Subcritical flows require the flow discharge at the upstream boundary and the stage at the downstream end, while supercritical flows require both the discharge and the stage at the upstream boundary (the analysis leading to this result can be found in any standard hydraulics textbook). In two- and three-dimensional flows this is equivalent to prescribing flow velocity field at the upstream, and a stage at the downstream boundary. Because usually only the upstream discharge is known, a vector velocity field must be synthesized, which is usually done by distributing the discharge proportionally to the conveyance along each node of the computational boundary, and then fitting a logarithmic profile along the vertical direction. Thus, any number of hydrographs can be discretized on an arbitrarily complex computational domain.

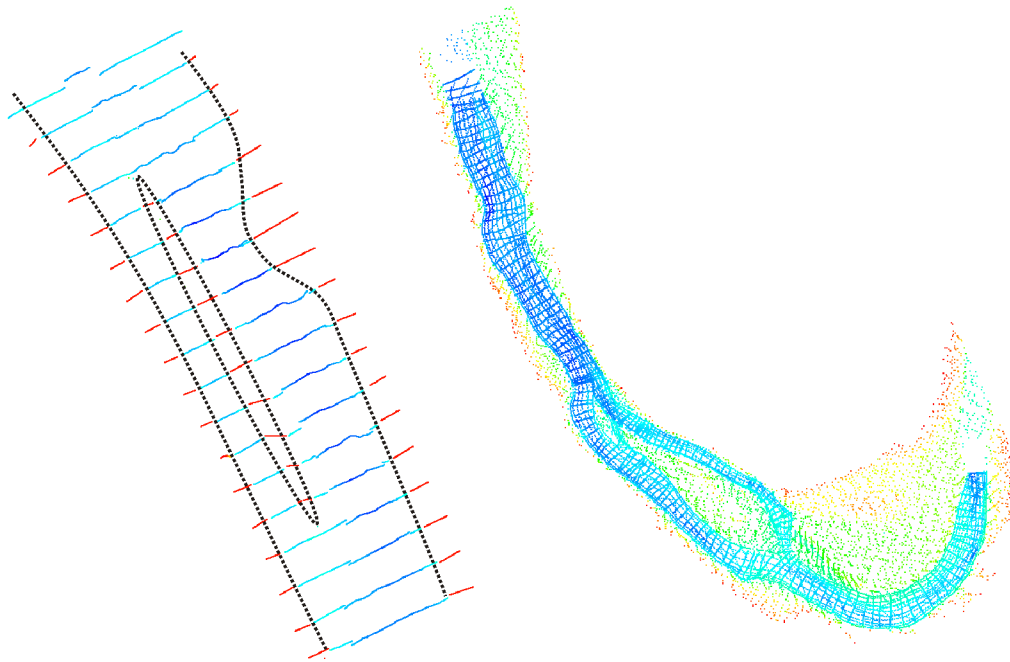


Figure 4.1 Example of river survey approaches. Left: cross section survey typically used in one-dimensional models; right: survey with unordered data with high density of survey data points. Color indicates bed elevation, lower elevations in blue and higher elevations in red.

The stage can be defined in a number of ways. Stage-discharge rating curves, an elevation hydrograph, or a water-surface-slope hydrograph are common ways to achieve this purpose. More complex conditions may be defined when the downstream boundary is a dam. In such cases, the reservoir operational scheme may be used. When the dam outlet works are used, relations for the gates and spillways may have to be employed.

Friction factors play an important role in determining stages and flow velocities, and must be given in the form of numerical values associated with particular regions of the bed, or in the form of relations that allow their representation as a function of other parameters—usually using hydraulic and/or sediment quantities. Local energy loss coefficients must also be given, such as those due to channel bends, natural or man-made obstacles to the flow, bridge piers and abutments, etc. Once again, these may be prescribed, or they may be calculated, and commonly require iterative calibration procedures.

Sediment data encompasses all the necessary information for sediment transport computations. The sediment inflow hydrograph must be given at the inflow boundaries. For two- and three-dimensional models it is also necessary to specify the sediment concentration distribution along those boundaries, as well as the separation between bed load and suspended load. The sediment particle-size distributions are also needed. Bed size distributions need also to be determined for each computational grid node (or for each cross section, in the case of one-dimensional models). Additionally, especially in the case of scour computations and bank widening, it is also necessary to provide the underlying bed material size

distribution. Water temperature variations must also be prescribed or modeled, because they have an indirect impact on sediment transport via the sediment particles' fall velocities.

In practice, it is difficult to determine a priori some of the hydraulic parameters necessary for a successful simulation. For that reason, usually there is a model calibration stage in which stage and discharge observations along the study reach are used to adjust the values of those parameters, such as bed roughness, discharge coefficients, or other parameters particular to the model employed. Similarly, there should be a calibration stage for the sediment transport calculations. Observations of the sediment outflow quantities, of variations in channel width and bed elevation, and of changes in sediment particle-size distributions can be used to properly adjust model parameters.

5. Application of Models in River Hydraulics

Just as in many other areas of application, the complexity and variety of the factors involved in river phenomena requires a careful and methodical approach to their modeling, with large support of field observations and prototype data. A step-by-step approach is recommended to deal with these issues. A schematic view of the entire process is presented in Fig. 5.1.

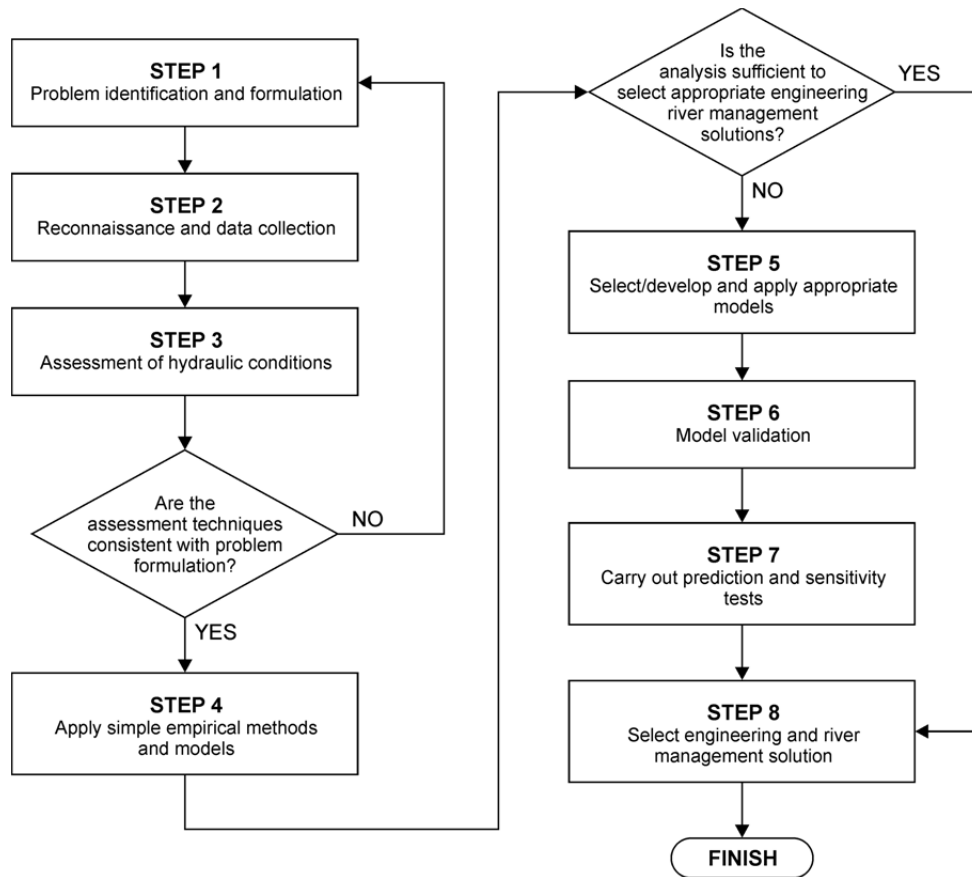


Figure 5.1 Procedure for the identification, analysis, and modeling of river hydraulics problems.

Step #1, problem identification and formulation, involves determining the factors at play, which may be associated with river engineering factors, with social activities, and may relate to existing conditions or future activities. It establishes the level of analysis required and defines the appropriate level of response. It is followed by step #2, field reconnaissance and data collection, in order to identify the nature and extent of problem of interest. In this stage, channel characteristics are identified, bank conditions and materials, hydrologic conditions, vegetation, any engineering works, and any other parameters that are considered relevant for the case at hand. During this stage there is an assessment of the existing data available and, if necessary, it is the time to design and mount an adequate data gathering program.

At this stage, a simple assessment of the conditions should be made using basic techniques of fluvial hydraulics and geomorphology. The results of this analysis are compared with existing conditions to provide an indication of the present morphological status of the prototype. This allows the determination of the impact of the proposed engineering works.

In step #4, simple empirical channel hydraulics models are applied. This step may help to interpret existing and proposed conditions, and in identifying the dominant processes and trends at play. The information gathered at this stage forms a framework for the more detailed modeling work that follows (if appropriate). In step #5, more advanced models are developed and used, as required, to provide the more detailed information. These models should be validated with existing prototype data (step #6), and applied to current conditions and also to assess the impacts of the proposed engineering works (step #7). At this stage, a sensitivity analysis involving all the pertinent parameters should be carried out, with particular emphasis in the parameters that are difficult to determine or that have significant spatial and/or temporal variability. In step #8, all the information gathered in the previous steps is used to formulate and implement the appropriate plan of action.

6. Using SToRM in iRIC 2.1

The practical use of a computer model based on the SWEs involves understanding and acceptance of the limitations of the governing equations—such as the assumption of hydrostatic pressure and the implicit acceptance that the error resulting from neglecting vertical accelerations is small for phenomena having characteristic length scales much larger than the flow depth. But it also involves dealing with the numerical techniques used to solve the SWEs. The numerical solution of the SWEs is a difficult problem and involves its own level of assumptions and approximations that have direct impact on data requirements and on the accuracy of the solution. This brief primer is written to provide some assistance with the latter aspects of SToRM and assumes that the reader is already familiar enough with the former—for example, from the previous use of a model, such as FASTMECH.

In iRIC, the numerical parameters used by SToRM's numerical engine are intermingled with a number of physical parameters in a series of parameter entry screens that must be properly filled before the model can run. Here, an overview of those screens is given, and guidelines to help their proper use are provided. These screens are displayed by going to the main pull-down menu in iRIC and choosing

*Calculation Condition > Setting**. There are four different entry screens: *Initial Conditions*, *Cell Wetting and Drying*, *Solution Parameters*, and *General Options*. They are selected at the left-hand side panel of the *Calculation Condition* window by left-clicking the mouse on the corresponding labels.

Note: both the program STORM and the user interface iRIC are subject to periodic improvements and updates, at which time the particular forms and contents of each entry screen may change. The information contained herein may, therefore, not be in perfect synchronism with the latest version of the software. The current version of this primer is displayed in the first page, and reflects the last date it was updated. The date of STORM's last update can be viewed in the *Select Solver* window that appears when iRIC is started, as shown in Fig. 6.1.

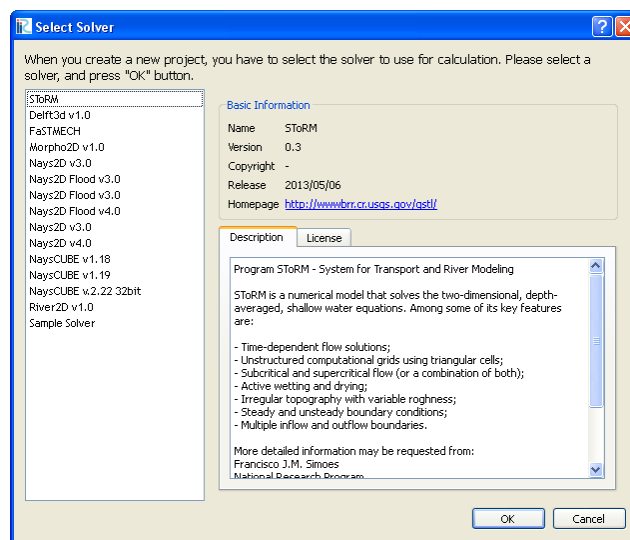


Figure 6.1 *Select Solver* input window with model information.

6.1 Initial Conditions

This screen (Fig. 6.2) is used to specify the initial state of the dependent variables, i.e., the values of the flow velocity and stage at start of a numerical simulation. In the present implementation of iRIC, there are three different ways to specify the initial flow field: using multiple coverage polygons defined interactively by the user; using a uniform flow field for the entire computational domain; and using a flow solution computed in a previous model run. The choice is made using the pull down menu in *Set-up initial conditions*.

When using multiple coverage polygons (*From coverage polygons* in the selection pull down menu), the user is asked to interactively define a polygon covering the area corresponding to each of the velocity components and free surface elevation. Multiple polygons may be defined for each variable. The use of uniform initial conditions (*Uniform initial conds.* in the pull down menu) requires the user to enter the

* Henceforth in this document, Italicized words are used to represent entities in iRIC, such as menu entries, titles of windows, individual window elements, etc.

quantities for the starting values of the flow velocity and free surface elevation in the dialogs *Starting value for U (m/s)*, *Starting value for V (m/s)*, and *Water surface elevation (m)*. Finally, when using an existing solution for a “warm” start-up (*Use existing solution* in the pull down menu), the user must choose the file where the desired solution is stored using the *Solution file name* entry dialog. Solution files reside inside project directories and are called *Case1.cgn*. When using coverage polygons, grid nodes located outside polygons have an assigned value of zero for the corresponding variable. Similarly, when using uniform initial conditions, grid nodes having an elevation higher than the specified water surface elevation are set to the dry state and the flow there is zero.

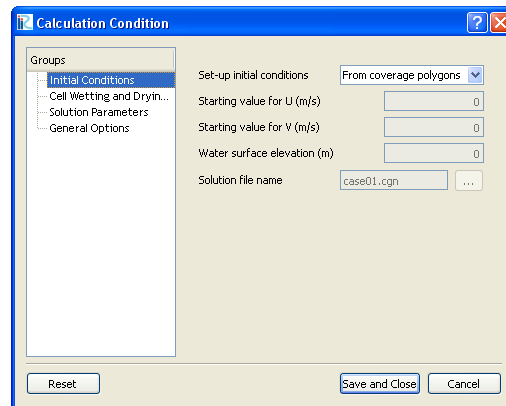


Figure 6.2 The *Initial Conditions* entry window.

The multiple choices offered are designed to give the user different possibilities to specify starting conditions for a model run. The closer the initial conditions are to the solution, the fewer iterations it will take the model to reach steady state. Use good common sense when defining these values.

6.2 Cell Wetting and Drying

The parameters defined in this entry screen (Fig. 6.3) determine how the wet-dry cell boundaries are treated in SToRM. In SToRM, cells with a water depth below a certain threshold depth are considered dry. Conservation of mass is still computed for these cells, but the velocity there is set to zero and the momentum equations are not used. This allows to track wetting and drying events, i.e., to determine when a particular cell is dry and becomes wet, and vice-versa. It also allows excluding dry cells from the computation, improving computational performance and numerical stability. Dry cells may become wet when there is a rise in the downstream stage, or as the result of a catastrophic collapse of a dam or levy. They become dry when a flood recedes. SToRM can deal with all these cases.

The depth threshold for the drying of a cell (i.e., water depth for when a wet cell is considered dry) is specified in the *Threshold for drying cells (m)* parameter. This may be changed from case to case, but 1 mm is a good starting point (Fig. 6.3 shows 3 mm). A higher threshold value improves stability and computational efficiency, but lower values promote accuracy. Note that all input parameters must be given in the SI system, therefore to specify a threshold of 1 mm, a value of 0.001 must be entered.

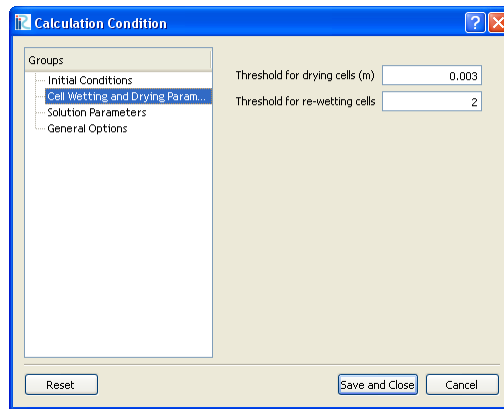


Figure 6.3 The *Cell Wetting and Drying Parameters* input window.

The water depth threshold for when a dry cell becomes wet is defined through the *Threshold for re-wetting cells* parameter. This re-wetting threshold parameter is a multiplier to the threshold for dry cells parameter: the threshold for wet cells is obtained from threshold for wet cells times the threshold for dry cells. For example, if the threshold for dry cells is set to 2 mm and the threshold for re-wetting cells to 3, the value of the water depth above which a dry cell becomes wet again is 6 mm. A value greater or equal to 1 is allowed, but a value greater than 1 is recommended (2 is a good value). This provides hysteresis and prevents intermittent wetting and drying of cells in consecutive iterations, which adversely impacts the convergence of the solution.

6.3 Solution Parameters

The *Solution Parameters* entry screen (Fig. 6.4) contains parameters related to the main solution algorithms, including the numerical methods, iterative procedures, and the time stepping. There are three time marching procedures* in STORM: *Order 1*, *Order 2*, and *Order 3*. These can be selected under the *Time stepping method* pull-down menu. Order 1 is the least accurate and the fastest; Order 3 is the most accurate and the slowest. In general, Order 1 is accurate enough for most applications. It can also be used in the first stages of a run to march faster to a converged solution, providing a better point to start a more accurate solution process. Order 2 is more accurate than Order 1, but it also is computationally more expensive (i.e., it uses more computer resources to run). The most accurate Order 3 can also be used, but its computational requirements make it less useful for use in practical applications.

The time step size is the time interval for which numerical solutions are computed. It is entered in the *Time step size (s)* entry dialog. If the time step is denoted by Δt , solutions are computed at time Δt , $2\Delta t$, $3\Delta t$, ..., $n\Delta t$, ... in which n is the time step number. The initial conditions are the solution at time zero. The time step Δt must be small enough to ensure numerical stability at all times during the calculations. To determine the length of the run, the total number of time steps N must be entered in the *Number of time steps* entry dialog. The total run will represent an interval of $N\Delta t$ seconds of simulation time.

* These methods were mentioned in Section 3.1.

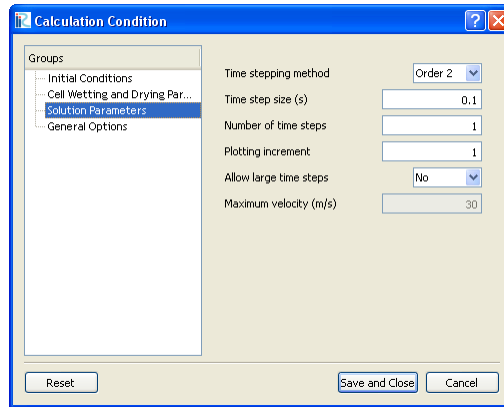


Figure 6.4 The *Solution Parameters* input window.

It may be desirable to plot solutions at specific time intervals. The plotting interval N_p , in time steps, is defined in the *Plotting increment* entry dialog. As a result, iRIC will store and display the solutions at the times $N_p\Delta t$, $2N_p\Delta t$, $3N_p\Delta t$, etc., until the final time $N\Delta t$ is reached. Note that N_p must be lower or equal to N .

STORM uses explicit time stepping techniques that are subject to time step size constraints. When these constraints are met (by using a small enough time step), the solution proceeds normally and remains bounded at all times, with no *Infinitys* or *Division by zeros* reported by the solver. If, however, instability occurs, the solution ceases to be bound and the computations “blow up,” which is a way to say that the variables become uncontrollably large (or small), cause divisions by zero, the water depth may become negative, the run stops prematurely, and its results are meaningless. Usually, in convection-dominated flows, stability is related to flow celerity, size of the time step, and triangle dimension by a dimensionless parameter:

$$C_r = \frac{\Delta t (U_0 + \sqrt{gh})}{P}$$

where U_0 is the magnitude of the flow velocity at the center of the triangle ($=\sqrt{U^2 + V^2}$) and P is a length scale (for example, the triangle’s perimeter). The dimensionless parameter C_r is called the Courant number and is fixed for a certain time stepping method, but usually different for different methods. The above relation translates into the flowing statements:

1. For a triangle of a certain size, larger flow velocity requires smaller time steps.
2. In the same situation, larger depths also require smaller time steps, but this constraint is less severe than the one for flow velocity.
3. Larger time steps can be used for larger triangles. For smaller triangles the time step size must be reduced accordingly.

These are important facts to keep in mind when selecting the parameters that go into the Solution entry screen. Unfortunately, things are not that simple. For example, for shallow depths the friction terms

become dominant, the flow is no longer convection-dominated, and new restrictions are imposed to the time step to keep the computations stable: small time steps are needed for very shallow depths. Therefore, a trial-and-error cycle is usually necessary to determine the largest time step that retains the stability of the computations. This is case dependent and varies with bed geometry and other boundary conditions. Each situation is different. Regardless of the particular model used, most parameters are case-dependent and require their own tuning. Through use, this process becomes somewhat more intuitive.

There are instances where the stability requirements are very severe for only a few triangles. This usually happens at the beginning of a simulation run, when the variables are changing very quickly (perhaps due to poor initial conditions). One way to address this issue is to choose a very small time step (one that ensures stability), march the solution in time for a few time steps, stop the simulation, and then restart it at the same point using a larger time step. This is a somewhat cumbersome and time consuming process for which STORM offers an alternative: simply set the program to forcefully bound the solution at all times. This is done by setting the *Allow large time steps* option in the *Solution Parameters* window to *Yes*. When this is done, flow velocity is constrained to the maximum value defined in the *Maximum velocity (m/s)* entry dialog.

Note that this procedure is designed to overcome the initial time steps that pose a greater challenge to the solver; it is not meant to force a solution with an incorrectly large time step (i.e., larger than that required for numerical stability). When this option is active, it is up to the user to ensure that each solution obtained is accurate and that it does not contain the results of unstable calculations. This can be verified by setting a *maximum velocity* (set to 30 m/s in Fig. 6.4) that is much larger than any flow velocity expected in the course of the calculations. If the computation results show large velocities (i.e., velocities close to the *maximum velocity*), then the solution is contaminated by areas of instability and the time step must be subject to further reductions.

6.4 General Options

The final entry window, called *General Options* (Fig. 6.5), allows the use of a few additional options in STORM, which are described herein.

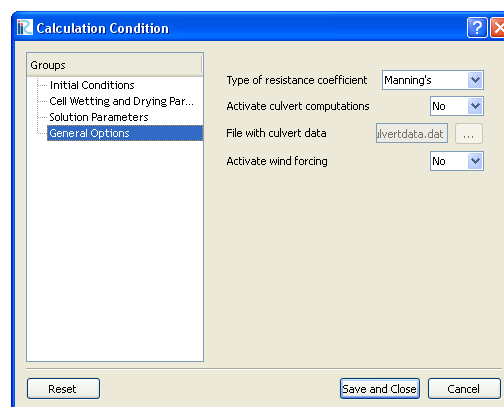


Figure 6.5 The *General Options* entry window.

6.4.1 Roughness coefficient

SToRM allows the use of three types of friction coefficient: Manning's n , the Chézy friction coefficient C , and the traditional skin friction coefficient c_f . These coefficients were presented in eqs. (7) and (8). The user can select which type of roughness formulation to use by means of the pull down menu *Type of resistance coefficient*.

6.4.2 Culverts and other types of hydraulic structures

SToRM allows the inclusion of hydraulic structures defined by performance rating curves. A performance rating curve is the depth versus discharge relation for the flow through the structure. The depth is defined at the invert of the structure's head. A structure is defined by a rating curve, by the (pointwise) location of its inlet and outlet, and by the invert of the structure at the inlet. SToRM transfers water from the inflow to the outflow, removing the appropriate amount of water from the triangle where the inflow is located and transferring it to the triangle where the outlet is located. This transfer is instantaneous and enforces conservation of mass.

Data is entered via a flat file, i.e., a readable ASCII text file. The computations must be activated by selecting *Yes* in the *Activate culvert computations* pull down menu, and the file must be located using the *File with culvert data* dialog (see Fig. 6.5). The performance curve must be discretized in a table with head, discharge pairs. The file format is the following:

1. The first entry must define the number of structures. Any number of structures is allowed.
2. Each structure is defined in the same manner: the first line must contain the x and y coordinates of the inlet, plus its invert; the second line must contain the x and y coordinates of the outlet; the third line must contain the number of entries in the performance curve table; finally, there must be as many lines of head-discharge pairs as were specified in the previous line.
3. Data must be typed separated by commas or blank spaces. Leading spaces are allowed, but the data must be contained within the first 40 columns of the file.
4. Blank lines are allowed (they are ignored by SToRM).
5. Lines whose first character is '#' are comment lines and may contain any text (they are ignored by SToRM).
6. Any text may be included in the lines (rows) containing data, but it must start in column 41 or after. This text does not need the '#' character.

An example of input data file with 2 structures is given in Fig. 6.6. The first one has a head-discharge table with 6 entries, while the second one only has two (there is a required minimum of one entry per table). The user must ensure that all the inlet and outlet coordinates refer to locations inside the computational domain, otherwise SToRM will terminate with an error message.

| | | |
|------------------|---|-----------------------|
| | #2345678901234567890123456789012345678901234567890123456789012345 | |
| | # Data file containing two hydraulic structures. | |
| | # Data is for demonstration purposes only. | |
| | 2 | Number of culverts |
| First structure | # Culvert number 1. | |
| | # Location of inlet and outlet. | |
| | 646086.5 4473217.5 1445.0 | Inlet: x,y,elevation |
| | 646139.0 4473237.5 | Outlet: x,y |
| | # Performance curve. | |
| | 6 | Number of entries |
| | 0.1 0.0001 | Head, discharge pairs |
| | 0.2 0.001 | |
| | 0.7 0.005 | |
| | 0.8 0.01 | |
| 1 0.0105 | | |
| 1.1 0.011 | | |
| | | |
| Second structure | # Culvert number 2. | |
| | 646179.9 4472958.5 1446.0 | Data for the |
| | 646238.8 4472986.5 | second culvert |
| | 2 | |
| | 0.1 0.0015 | |
| | 0.5 0.0030 | |
| | | |

Figure 6.6 Input data file containing two hydraulic structures. The first row is included for facilitating reading column number, as are the shaded vertical bars.

The performance curve must contain only the flow that passes through the structure. In some cases, however, there may be flow passing through the structure as well as above it. This may be the case in overflowing highway culverts, when the culvert is at maximum capacity and there is also flow passing over the roadway above it. In this case, the culvert flow rate is calculated from the performance curve, and the free surface flow over the roadway is calculated normally by STORM. Therefore, performance curves must not contain the structure's overflow. This is illustrated in Fig. 6.7, where a typical culvert performance curve is shown. Only the inlet and outlet control parts of the performance curve are used in the STORM input data, and the roadway overtopping contribution must not be included. Similar considerations must be made for other hydraulic structures, such as for submerged gates and drain pipes.

6.4.3 Wind forcing

Wind circulation is included when the wind forcing terms in eq. (10) are added to the governing equations. This is done by selecting *Yes* in the *Activate wind forcing* pull down menu. Wind friction and velocity must be specified using coverage polygons defined interactively by the user. Wind velocity data is specified by magnitude (m/s) and azimuth (degrees from northing, clockwise direction). The wind friction data needed is the product $\rho_a c_{fw}$ in eq. (10). The density of air, ρ_a , is 1.225 kg/m³ at 15 degrees centigrade (approximately 60 degrees Fahrenheit). The wind friction coefficient at the water surface varies, ranging from 0.005 over smooth water to 0.015 over grassland.

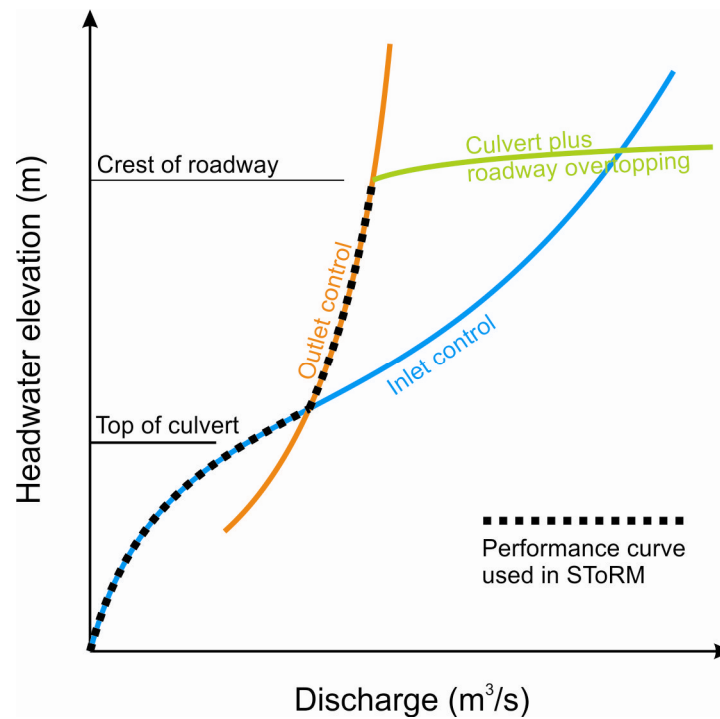


Figure 6.7 Performance curve used for flow through culverts in STORM.

7. Further Considerations

Experience and good engineering judgment are the most important elements when using any computer model. Together, they provide a roadmap of what to expect from a particular model run and how to brainstorm unexpected results. That cannot be taught in a short monograph such as this one, but the next remarks may be useful when using STORM.

STORM uses a numerical discretization based on triangles of constant slope. It uses a control volume formulation in which the control volumes are triangles and the dependent variables (flow velocity and depth) are defined at the triangles' centers. Triangular unstructured grids may be less intuitive to use than the more usual structured grids, such as the one used in FASTMECH. An example on how a wet-dry boundary could be represented by the two types of discretization is shown in Fig. 7.1. Both have advantages and disadvantages: models that use structured grids are usually computationally more efficient (i.e., they require shorter computational run time for grids of comparable size), but unstructured grids can make a better use of local refinement and provide a more accurate representation of the topography.

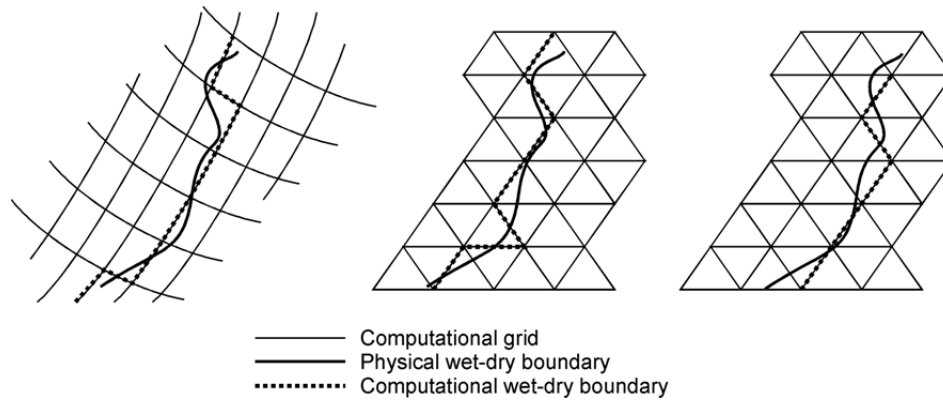


Figure 7.1 Representation of a physical boundary (a river bank) by a structured rectangular curvilinear grid (left) and by unstructured triangular grids with approximately the same cell size, in area, but placed in different locations (center and right).

A particularly difficult issue is the generation of computational grids for the arbitrarily shaped channels and irregular topographies usually encountered in natural rivers. The river channel and adjacent flood plains have a continuous nature that becomes discontinuous when discretized. The features captured in any discretization are determined by the size and placement of the computational cells: smaller triangles capture more features at the expense of more computational effort in computer memory and run time. Thus, the grid size defines a scale that ultimately controls the nature of the model predictions and the results need to be interpreted accordingly. But there must be a proper balance: at coarse grids the model may be unable to resolve the flow features or even produce converged solutions, but at finer grids the computational requirements may be too demanding. Note also that when the computational grid is finer than the topographic resolution, the model predictions become hostage to the nature of the topographic interpolation without representing any real topographic variation. Therefore, the data collection program should be guided by the modeling objectives, and the modeling efforts must be consistent with the data collected.

iRIC provides a number of tools to help in grid generation and feature representation. Those should be used to their full extent, but in a manner consistent with the topographic data. A starting rule of thumb is to use 8 to 10 triangle edges across any feature that needs to be resolved—for example, across a river’s cross-section, or across a recirculation area, or behind an obstacle. Local refinement may be useful to accomplish this. Improve computational efficiency by removing triangles that do not contribute to the solution (SToRM accepts holes and any type of irregular boundaries). iRIC allows defining sharp breaks in bed slope: use those tools to place triangle edges where they can provide the most resolution at the smallest cost. Finally, note that SToRM treats all outer edges (other than at inflow or outflow boundaries) as solid walls, therefore water will not spill out of the computational domain in those areas. Use them to define features like levees, abutments, vertical walls, etc.