Chapter 11

Modeling in Practice

11.1. Modeling software

11.1.1. *Introduction*

The purpose of this chapter is to advise on a number of modeling precautions and to provide a few guidelines to users of modeling software packages.

In an industrial context, the choice of a specific modeling package is most often not made by the engineer or technician who operates it. Whether in a research team or in a consulting company, the selection (or development policy) of the modeling tool proceeds from numerous considerations. The quality and accuracy of the modeling results is only one of the many criteria used in the selection process. This is because numerical modeling is increasingly integrated into multi-criteria decision-making processes. Sophisticated algorithms and numerical techniques are nowadays "encapsulated" in user-friendly modules, served by efficient graphical interfaces with the purpose of facilitating modeling result interpretation and decision-making. To give but one example, commercial river flow modeling packages still use numerical schemes (for example, Preissmann's scheme presented in Chapter 6) developed in the 1960s. For most of them, the development effort has concentrated on graphical interfaces or exchange modules with graphical or decision support tools such as database management systems, geographical information systems, etc.

As indicated in Chapters 6 to 9, numerical techniques provide only approximations to the solutions of the governing equations. Some techniques are more accurate than others, some are more computationally efficient. Sometimes, accuracy and/or computational efficiency are optimal only for a specific type of

Wave Propagation in Fluids: Models and Numerical Techniques, Second Edition Vincent Guinot

situation (e.g. rapid or slow transients, in the presence or absence of solution discontinuities, etc.). There is often little room for the modeling engineer to question the choice of a given software or numerical technique used in his team or company. The engineer's task is rather to be as much aware as possible of the limitations (in terms of robustness, accuracy, validity of model assumptions) of the modeling package used, so as to minimize the risk of misuse and optimize the quality of the modeling results.

The main two questions that must be answered by a modeler are (i) what are the key phenomena involved in the configuration to be modeled, and (ii) what are the basic requirements to be fulfilled by the numerical method so as to guarantee the quality of the numerical solution? Two basic examples of such modeling issues are given in the next sections.

11.1.2. *Conservation*

In many engineering applications related to environmental fluid mechanics, conservation is a key issue. As an example, mass conservation (both in terms of water and solute transport) may seem a natural requirement for a river flow modeling package. Mass conservation implies that the variation in the amount of water (or contaminant) stored within a reach is equal to the difference between the discharge across the upstream and downstream sections. If pollution or water resource allocation studies are to be carried out, conservation appears as an indispensable prerequisite in the model selection process. This simple condition, however, may not be satisfied if the modeling package solves the wrong form of the equations, as shown in the example hereafter.

The continuity equation for transient flow in a channel has been presented in section 1.5.1. The conservation form is given by equation [1.84], recalled here:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$$

where A is the channel cross-sectional area and Q is the volume discharge. In some software packages, however, the dependent variables are not A and Q, but A and $u = Q/A$, where u is the flow velocity. Since $Q = Au$, equation [1.84] can be rewritten as:

$$\frac{\partial A}{\partial t} + A\frac{\partial u}{\partial x} + u\frac{\partial A}{\partial x} = 0 \qquad [11.1]$$

In the case of smooth channel geometry, equations [11.1] and [1.84] can be approximated with a similar order of accuracy by a given numerical technique. If the geometry of the channel is locally discontinuous (see Figure 11.1), A and u are locally discontinuous (while Q is not) and thus locally non-differentiable. Approximating the terms $u\,\partial A/\partial x$ and $A\,\partial u/\partial x$ may lead to mass conservation problems.

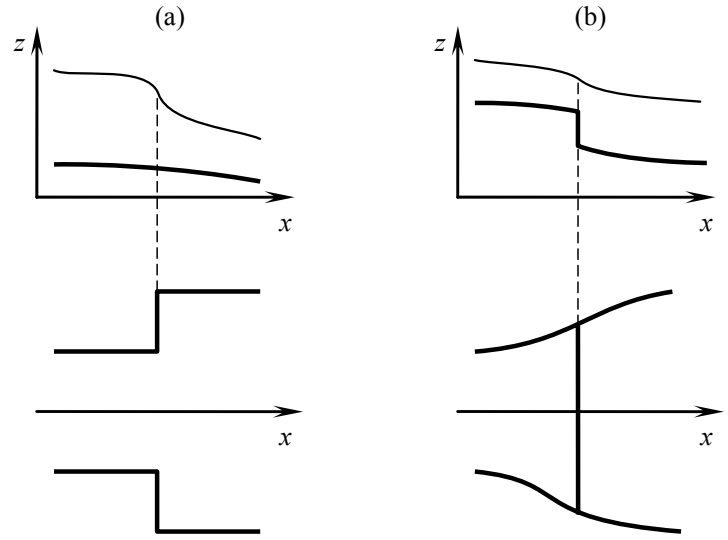

Figure 11.1. *Typical examples of discontinuous geometries in river modeling: sudden widening (a), bottom step (b). Side view (top), plan view (bottom)*

Similar conservation problems may appear if the term $\partial A/\partial t$ is rewritten as:

$$\frac{\partial A}{\partial t} = b\frac{\partial h}{\partial t} = b\frac{\partial \zeta}{\partial t} \qquad [11.2]$$

where b is the top width, h is the water depth and ζ is the free surface elevation (see section 2.5.2 and Figure 2.12 for the notation). If the channel width is a discontinuous function of z (Figure 11.2), b may become discontinuous and the estimate of $b\,\partial h/\partial t$ (or $b\,\partial \zeta/\partial t$) may become incorrect.

In the example of Figure 11.2, the section is piecewise rectangular. The free surface width switches discontinuously from b_1 to b_2 at $z = z_1$. The derivative $b = \partial A/\partial h$ is thus undefined for $z = z_1$. Assume that the free surface elevation ζ is lower than z_1 (thus $b = b_1$) at the beginning of the computational time step and

higher than z_1 (thus $b = b_2$) at the end of the computational time step. Any purely explicit or purely implicit estimate of b in the term $b\,\partial h/\partial t$ yields an incorrect estimate for $\partial A/\partial t$.

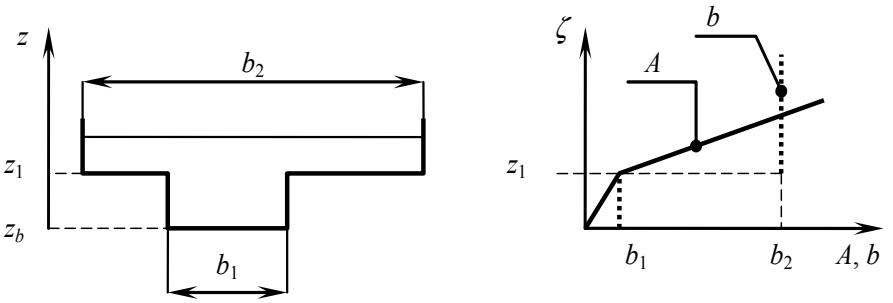

Figure 11.2. *Example of a channel geometry with a discontinuous top width. Left: cross-sectional view of the geometry. Right: variations in A and b with the free surface elevation ζ*

Even if the geometry is smooth, not solving the conservation form of the equations may yield erroneous solutions in the presence of discontinuous solutions (shock waves in the field of gas dynamics; hydraulic jumps or moving bores in free surface hydraulics). This is due to the non-uniqueness of weak solutions (see Chapter 3 for more details). Indeed, the conservation, non-conservation and characteristic forms of the governing equations are not equivalent in the presence of discontinuous solutions because the partial derivatives are locally undefined.

This is illustrated by the dambreak simulation shown in Figure 11.3.

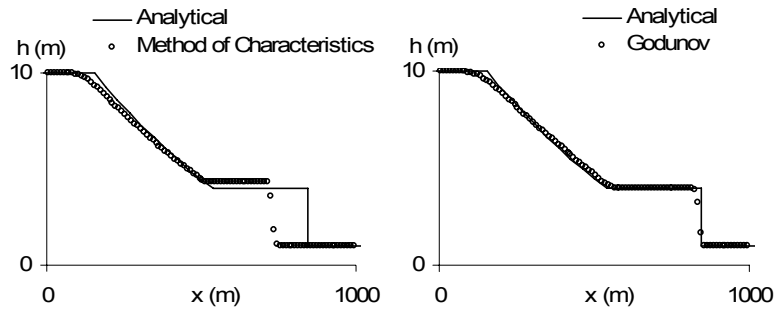

Figure 11.3. *Dambreak problem. Solution at t = 35 s*

The problem and its analytical solution are presented in detail in section 4.3.3. Recall that the solution consists of a region of constant state separated from the upstream and downstream sides of the dam by a rarefaction wave and a shock respectively. The initial water levels on the left- and right-hand sides of the dam are respectively 10 m and 1 m. Two numerical techniques are used to solve this problem. Figure 11.3 (left) shows the results obtained at $t = 35$ seconds with the finite difference-based, first-order method of characteristics (see section 6.2). Figure 11.3 (right) shows the results obtained using the finite volume-based, Godunov scheme (section 7.2), with fluxes calculated by the HLL approximate Riemann solver (see section C.1 in Appendix C for a description of this solver).

The first-order method of characteristics yields an overestimated water depth in the intermediate region of constant state, while the speed of the shock is underestimated. It is clearly visible from the figure that the total volume of water is not conserved with this method. A numerical integration indicates that the volume of water per unit width computed by the method of characteristics is 5,168 m^2, for an initial volume per unit width of 5,520 m^2. Approximately 6% of the total volume is lost artificially over the 35 simulated seconds. In contrast, the volume is preserved exactly in the Godunov simulation.

11.1.3. *Solution monotony*

The solutions of hyperbolic systems of conservation laws are essentially TVD. The practical consequence is that oscillations cannot develop spontaneously in the solutions of hyperbolic systems (unless specific combinations of boundary conditions and/or source terms are used). This is why TVD and upwind schemes are particularly praised by environmental modelers.

As an example, centered schemes are not popular in the field of contaminant transport modeling because they induce artificial oscillations in the computed profiles. For instance, artificial oscillations around a small or zero background concentration may yield locally negative concentration values in the numerical solution of a contaminant transport model. Although such oscillations may be easily justified from a mathematical point of view on the basis of truncation error analyses or from scheme phase and amplitude portraits (see Appendix B), their physically unrealistic character makes them hardly acceptable to decision-makers. The modeler is left with two options:

(1) Using a TVD scheme allows the monotone character of the solution to be preserved, thus eliminating physically unrealistic solutions. Moreover, the numerical diffusion applied by TVD schemes in the neighborhood of steep gradients contributes to make the solution more “realistic” than non-TVD schemes, because they imitate the diffusion and dissipation mechanisms that are present in natural

processes. It must be remembered however that numerical diffusion or dispersion proceed from truncation errors and that they should be seen as a sign that the numerical solution is not optimally accurate.

(2) In contrast, non-TVD schemes such as centered schemes (see section 6.5) minimize numerical dissipation. For this reason, they are most appreciated in modeling fields where numerical diffusion and dissipation are seen as "parasitic" phenomena that jeopardize the quality of the solution. This is the case for instance in the field of turbulence modeling, where upwind schemes are considered too dissipative.

In solving real-world modeling problems, the choice of a numerical technique most often results from a trade-off between the accuracy of the numerical solution and the monotony or positivity properties of the analytical solution that are deemed essential by the modeler.

11.2. Mesh quality

Meshing is an important step in the modeling process. Despite the availability of efficient mesh generation packages, meshing complex geometries still requires human supervision. Optimal result quality is achieved if the mesh is regular and isotropic. These issues are illustrated with the two-dimensional shallow water equations.

Mesh regularity. Numerical methods usually need 3 to 5 grid points (or cells, or elements) to represent steep gradients or discontinuities, as shown by the numerical results presented in Chapters 6 to 8. Strongly diffusive numerical methods may induce front smearing over 10 to 20 cells (see the tests presented in Chapter 8). A strong size contrast between neighboring cells in a grid may amplify the artificial smearing. The consequence may be an artificial damping of transients in some parts of the model.

An example of mesh generation is given in Figure 11.4. The purpose is to generate a mesh for a river main channel and floodplain simulations. The width of the river banks imposes the minimum size for the elements. The banks being narrow compared to the floodplain, the modeler may be tempted to generate a mesh with elements rapidly increasing in size in the direction of the floodplain. The ratio of the area of the smaller to the larger elements is approximately 250. The transition takes place within 4 to 5 elements, which implies an area ratio of 3 to 4 between two neighboring cells. A ratio of 1.5 to 2 is usually advised.

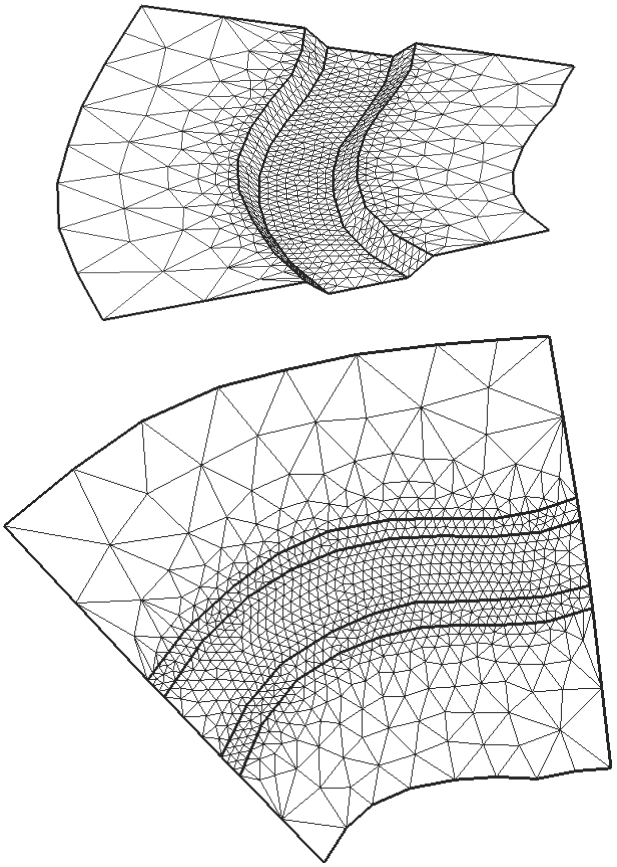

Figure 11.4. *Example of a strongly irregular mesh. Top: perspective view. Bottom: plan view*

Mesh isotropy. Another issue in two-dimensional mesh generation is the meshing of long and narrow geometric features, such as dikes, roads, river embankments, etc. Using long and narrow elements may allow for a substantially reduced computational effort. The modeler is usually inclined to stretch the mesh in the direction of the flow in the main channel and along the river banks (Figure 11.5). The aspect ratio of the cells used to discretize the channel embankments in Figure 11.5 is between 7 and 10.

This meshing approach is traditionally justified with the argument that the Courant number, that is the key parameter to the quality of the numerical solution, should be the same in the longitudinal and transverse direction in order to maximize solution accuracy. If the purpose is to simulate passive scalar advection, the Courant number is determined by the flow velocity vector and stretching the mesh in the longitudinal direction is justified.

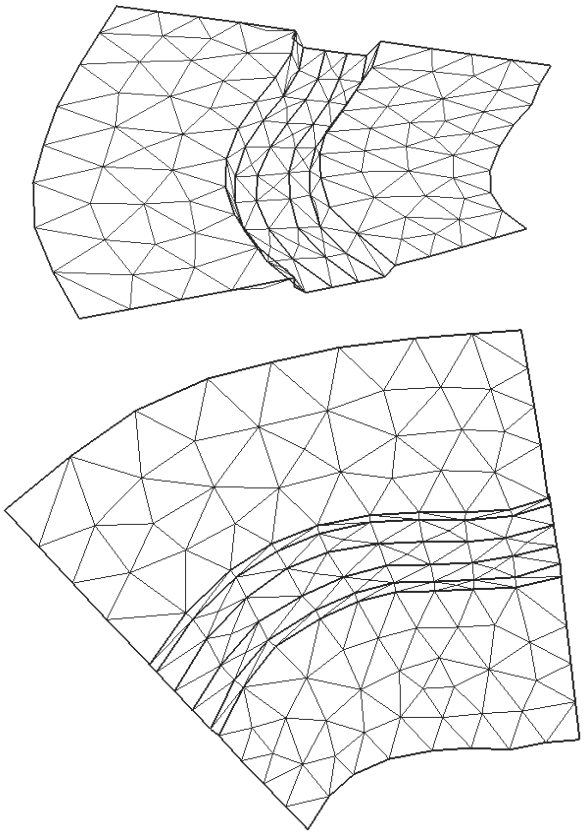

Figure 11.5. *Example of a strongly anisotropic mesh. Top: perspective view. Bottom: plan view*

If the purpose is to simulate floodplain dynamics and two-dimensional free surface transients, however, the Courant number is determined from the propagation speed of the waves. As shown in Chapter 5, the domain of dependence of the solution is made of two surfaces in the phase space (Figure 11.6). The first surface is the curved line with tangent vector (u, v, 1). The second surface is a conical surface,

expanding from the first surface at a speed $c = (gh)^{1/2}$. Two situations may be considered:

– Subcritical flow: the plan view shape of the dependence domain of the solution over a time step Δt is circular (Figure 11.6a). The domain of dependence of the point A in Figure 11.6a is a circle centered around B, that is shifted from A by a distance $(u^2 + v^2)^{1/2}\,\Delta t$. The domain of dependence is isotropic. Stretching the computational cell in the direction of the flow artificially increases the weight of lateral elements in the estimate of the gradients.

– Supercritical flow (Figure 11.6b): the point A is not included in the domain of dependence of the solution. The shape of the domain of dependence becomes narrower as the Froude number (the Mach number in the case of gas dynamics simulations) increases. Figure 11.6b shows the shape of the domain of dependence for a Froude number Fr = 2.5. In this case, the ratio of the longitudinal to transverse dimensions of the domain of dependence is (2.5 + 1)/2 = 1.75. Stretching the cells by a factor larger than 2 in the longitudinal direction is not justified.

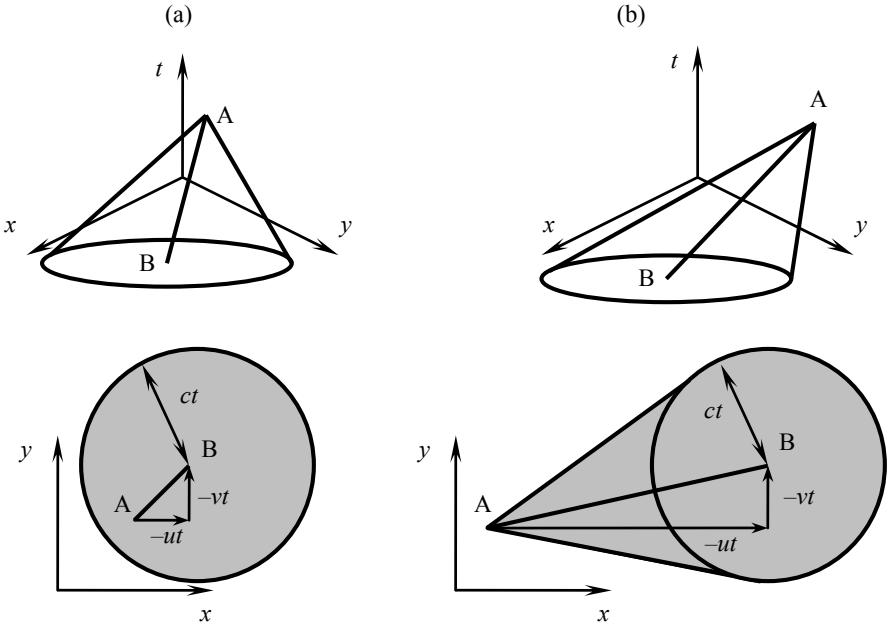

Figure 11.6. *Domain of dependence of a point A. Subsonic/subcritical case (a), supersonic/supercritical case (b). Top: perspective view. Bottom: plan view*

The longitudinal and transverse dimensions of the domain of dependence of point A are given by:

$$\left.\begin{aligned} D_L &= (u^2 + v^2)^{1/2} + c - \min[(u^2 + v^2)^{1/2} - c, 0] \\ D_T &= 2c \end{aligned}\right\} \qquad [11.3]$$

the minimum operator, min( ) allows a single expression to be obtained for both subcritical/subsonic and supercritical/supersonic conditions. The aspect ratio of the domain of dependence is thus obtained as:

$$\frac{D_L}{D_T} = \frac{1}{2}\left[\mathrm{Fr} + 1 - \min(\mathrm{Fr} - 1, 0)\right] = 1 + \frac{\mathrm{Fr} - \min(\mathrm{Fr}, 1)}{2} \qquad [11.4]$$

where $\mathrm{Fr} = (u^2 + v^2)^{1/2} / c$ is the Froude number. In the field of gas dynamics, the Froude number is replaced with the Mach number M, thus yielding the following formula:

$$\frac{D_L}{D_T} = 1 + \frac{M - \min(M, 1)}{2} \qquad [11.5]$$

Equations [11.4] and [11.5] show that stretching the mesh in the longitudinal direction is not justified in the case of subcritical/subsonic flow configurations if the purpose is to solve the hydrodynamic equations. Strong mesh aspect ratios should be used only in the case of scalar transport.

Mesh anisotropy does not have the same consequences on solution accuracy for explicit and implicit schemes:

– if the numerical scheme used is explicit, numerical diffusion is higher when the Courant number is smaller (see the amplitude portraits shown in Appendix B). Consequently, gradient smearing is stronger in the direction the mesh is stretched;

– if an implicit numerical scheme is used, numerical diffusion is usually stronger for larger Courant numbers. Gradient smearing occurs preferentially in the direction perpendicular to mesh stretching. A classical consequence of this is the artificial polarization of the velocity field along the wider dimension of the mesh.

11.3. Boundary conditions

11.3.1. *Number and nature of boundary conditions*

The number of conditions to be prescribed at a model boundary is the number of characteristics entering the computational domain (see Chapters 1 and 2, [CUN 80]). For a one-dimensional configuration, the following holds:

– For all the scalar laws presented in this book, the propagation direction of the characteristics is that of the flow. Consequently, one boundary condition is needed at each inflowing boundary. Typically, the boundary condition is supplied in the form of a prescribed value (prescribed concentration in the case of contaminant transport, water saturation for the Buckley-Leverett model, flow velocity for the inviscid Burgers equation) or a prescribed flux F at the boundary.

– The water hammer equations (section 2.4) are characterized by two constant, opposite wave speeds. There is one incoming characteristic at each domain boundary. One boundary condition is needed at each end of the model. The most classically used conditions are prescribed pressure, prescribed discharge, and pressure-discharge relationships (e.g. pumps).

– In the Saint Venant equations (section 2.5), the number of boundary conditions is a function of the flow regime. Supercritical inflow requires two boundary conditions, supercritical outflow requires none. Both subcritical inflow and outflow require one boundary condition. Typical boundary conditions in free surface hydraulics are prescribed water level, prescribed discharge and stage-discharge relationships.

– In the Euler equations (section 2.6), the number of boundary conditions is also a function of the flow regime. Supersonic inflow requires three boundary conditions, supersonic outflow requires none. Subsonic inflow requires two boundary conditions, subsonic outflow requires only one. Classical types of boundary condition are prescribed pressure, prescribed flow velocity, prescribed density.

Almost all market-available simulation software packages are equipped to deal with all these types of boundary conditions, thus leaving the modeler with an unbounded number of possibilities to model a given situation. The modeler should be aware, however, that some combinations of boundary conditions must be used with extreme care, and that certain boundary condition types may not be used at all boundaries of the model. He should also be aware that the model may not be able to prescribe the desired boundary value in certain situations. These aspects are explored in the following sections.

11.3.2. *Prescribed discharge/flow velocity*

Prescribed discharge conditions are classical in pipe transient and free surface flow simulations. Gas dynamics simulations rather use prescribed flow velocity conditions. Prescribing such conditions at the upstream boundary of a computational domain usually poses no problem. Prescribing an outflowing discharge or velocity at the downstream boundary of a model may jeopardize the simulation. The reasons for this are the following:

(1) There exists a maximum possible value for the prescribed discharge or flow velocity at a downstream boundary. This is true for the water hammer equations, Saint Venant and Euler equations. This point is illustrated with the Saint Venant equations in a frictionless horizontal, rectangular channel. In this case, the characteristic form of the equations (see section 2.5) simplifies to:

$$\left.\begin{array}{ll} u - 2c = \text{Cst} & \text{for } \dfrac{\mathrm{d}x}{\mathrm{d}t} = u - c \\ u + 2c = \text{Cst} & \text{for } \dfrac{\mathrm{d}x}{\mathrm{d}t} = u + c \end{array}\right\} \qquad [11.6]$$

Consider a channel with water depth and flow velocity h_0 and u_0 next to the right-hand boundary. Since the water is flowing out of the channel, the flow velocity u_0 is assumed positive. The second characteristic equation [11.6] leads to:

$$u_0 + 2(gh_0)^{1/2} = u_b + 2(gh_b)^{1/2} \qquad [11.7]$$

where h_b and u_b the water depth and flow velocity at the downstream boundary. Equation [11.7] can be rewritten as:

$$u_b = u_0 + 2(gh_0)^{1/2} - 2(gh_b)^{1/2} \qquad [11.8]$$

The unit discharge $q_b = h_b u_b$ at the downstream boundary is therefore:

$$q_b = u_b h_b = \left[u_0 + 2(gh_0)^{1/2} - 2(gh_b)^{1/2}\right] h_b \qquad [11.9]$$

The function $q_b(h_b)$ is zero for $h_b = 0$ and for a depth $h_{\max}$ defined as:

$$h_{\max} = \frac{\left[u_0 + 2(gh_0)^{1/2}\right]^2}{4g} \qquad [11.10]$$

Figure 11.7 illustrates the behavior of the function $q_b(h_b)$.

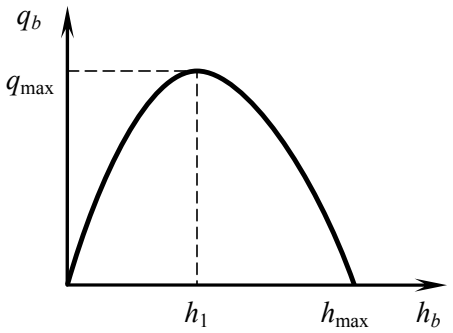

Figure 11.7. *Unit discharge q_b at the downstream boundary of the domain as a function of the downstream water depth h_b*

The function q_b is maximum for $h_b = h_1$ such that $\mathrm{d}q_b/\mathrm{d}h_b = 0$, that is:

$$h_1 = \frac{\left[u_0 + 2(gh_0)^{1/2}\right]^2}{9g} = \frac{4}{9}h_{\max} \qquad [11.11]$$

The maximum possible prescribed discharge $q_{\max}$ at the downstream boundary is given by:

$$q_{\max} = \left[u_0 + 2(gh_0)^{1/2} - 2(gh_1)^{1/2}\right] h_1 = \frac{1}{g}\left[\frac{u_0 + 2(gh_0)^{1/2}}{3}\right]^{3/2} \qquad [11.12]$$

(2) It is not possible to prescribe an outflowing velocity yielding a Froude number (Mach number for gas dynamics simulations) larger than unity at the downstream boundary of the domain.

(3) Prescribing the discharge at both ends of a channel reach is not advised [CUN 80]. Indeed, this is an indirect way of prescribing the amount of fluid stored in the domain. This may lead to simulation problems if the outflowing discharge is larger than the inflowing discharge, because the net mass balance may exceed the quantity of water available in the model at some stage.

11.3.3. *Prescribed pressure/water level*

In free surface hydraulics, prescribed water level conditions are usually met at the downstream end of river models. An exception is the simulation of tidal flows,

where the water level may be prescribed at both ends of the modeled reach. In the field of water hammer and gas dynamics simulations, the pressure may be prescribed at any model boundary.

It is not always possible to prescribe the desired value of the pressure or water level at a domain boundary. This is because prescribing too low a pressure or level may trigger supercritical outflow, a situation where the flow variables at the boundary are entirely determined from the low conditions inside the domain. Since all the characteristics leave the domain, attempting to prescribe a pressure or water level in such situations is meaningless.

The example of the Saint Venant equations in a rectangular, horizontal channel presented in section 11.3.2 is used again. The Froude number at the boundary is obtained from equation [11.8] as:

$$\mathrm{Fr}_b = \frac{u_b}{c_b} = \frac{u_0 + 2(gh_0)^{1/2} - 2(gh_b)^{1/2}}{(gh_b)^{1/2}} \qquad [11.13]$$

It is easy to check that Fr_b is a decreasing function of h_b and that $\mathrm{Fr}_b = 1$ for $h_b = h_1$ as defined in equation [11.11]. The behavior of the function $\mathrm{Fr}_b(h_b)$ is illustrated in Figure 11.8:

– For $h_b > h_1$, Fr_b is smaller than unity. The characteristic $(u - c)$ enters the domain. A boundary condition may be prescribed.

– For $h_b < h_1$, Fr_b is larger than unity. The characteristics $(u - c)$ and $(u + c)$ leave the domain and h_b cannot be prescribed at the boundary.

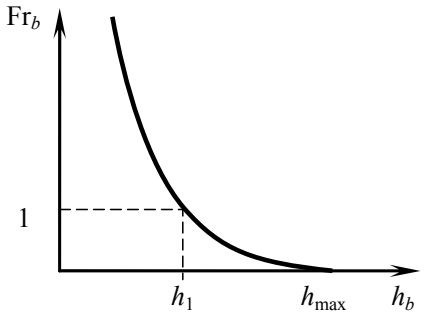

Figure 11.8. *Froude number Fr_b at the right-hand boundary as a function of the downstream water depth h_b*

The above reasoning remains valid for any hyperbolic system with wave speeds that may change sign, such as the Euler equations.

11.3.4. *Stage-discharge and pressure-discharge relationships*

In the field of pipe transient modeling, pressure-discharge relationships are used to represent head losses across a singularity, one side of which is connected to a node with a fixed pressure. They are also used to represent pumps taking water from a source with a known pressure. In the field of free surface flow modeling, stage-discharge relationships are often used to provide a condition at a boundary of the model where no measurement is available for the water level or discharge. In such a case, the stage-discharge relationship is a function of the channel geometry and is derived from specific assumptions on the flow regime: the assumption of a uniform or critical flow allows the discharge to be inferred from the water level. Using the assumption of uniform flow leads to equation [1.83], recalled here:

$$Q = \frac{1}{n_M} \frac{A^{5/3}}{\chi^{2/3}} S_0^{1/2}$$

where A is the cross-sectional area, n_M is Manning's friction coefficient, S_0 is the bed slope and χ is the wetted perimeter. The assumption of critical flow leads to:

$$Q = \left(\frac{g}{b}\right)^{1/2} A^{3/2} \qquad [11.14]$$

where b is the top width of the channel.

Pressure-discharge and stage-discharge relationships must be used with care. In particular:

(1) the inflowing discharge at an upstream boundary should always be a decreasing function of the pressure or water level,

(2) the outflowing discharge at a downstream boundary should always be an increasing function of the pressure or water level.

The computational solution may become unstable if these rules are not satisfied. The reason for this is the following:

– Consider an inflowing (upstream) boundary for a water hammer model. Assume that the discharge is an increasing function of pressure. Any increase in the pressure at the boundary (such an increase may be triggered by a transient

propagating in the pipe) results in an increase in the discharge. From continuity, increasing the discharge at the upstream boundary triggers fluid compression, which triggers an increase in the pressure. This in turn generates an increase in the discharge. This cyclic behavior results in instability. Note that the reasoning also holds for a decrease in the discharge. Similar arguments may be used for the shallow water equations (with the difference that the pressure is replaced with the water level).

– Consider an outflowing boundary. Assume that the discharge is a decreasing function of pressure. Any increase in the pressure resulting from transients generated within the domain yields a decrease in the discharge at the downstream boundary. From continuity, this triggers fluid compression, thus leading to a pressure increase. This triggers a new decrease in the pressure. Repeating the cycle may lead to instability in this case too.

– In contrast, prescribing a decreasing pressure-discharge condition at the upstream boundary or an increasing pressure-discharge relationship at the downstream boundary contributes to stabilizing the solution.

11.4. Numerical parameters

11.4.1. *Computational time step*

Most numerical methods presented in Chapters 6 to 10 achieve optimal accuracy when the absolute value of the Courant number of the waves is close to one. When the flow and geometry are highly variable in space, it is not possible to maintain the same value of the Courant number at all points for all waves. The computational time step often results from a trade-off between solution accuracy and computational efficiency. Recall that:

– explicit schemes are subjected to a stability constraint. This constraint imposes that the absolute value of the Courant (also called CFL) number of the faster wave should not be larger than unity. Simulation packages using explicit methods may reduce the computational time step during the simulation so as to enforce the stability constraint. The computational time step actually used may not always be the computational time step requested by the user;

– implicit schemes are not subjected to stability constraints. Commercially available simulation packages classically use the time step requested by the user, even when this yields very large values of the Courant number. It is advised that the modeler estimate roughly the typical wave speeds to be encountered during the simulation, so as to specify a computational time step that will ensure an average Courant number close to one.

11.4.2. *Scheme centering parameters*

Implicit schemes such as the Preissmann scheme or the finite element schemes presented in Chapter 8 use a time-centering parameter θ. This parameter is used to weight the respective contributions of the time levels n and $n + 1$ in the calculation of the space derivatives. $\theta = 1/2$ give the same weight to the known time level n and the unknown time level $n + 1$. The larger θ, the larger the contribution of the unknown time level $n + 1$. Centered schemes, such as the Crank-Nicholson or Galerkin technique with symmetrical shape functions, give unstable solutions for $\theta < 1/2$. When θ is set to 1/2, oscillations usually appear in the computed solutions when $Cr \neq 1$ because the numerical diffusion in the truncation error is set to zero, thus leaving room for numerical dispersion (see Appendix B for detailed considerations). Increasing θ allows numerical diffusion to be increased, thus leading to solution stabilization, profile smoothing and artificial wave damping.

The Preissmann scheme uses an additional, space centering parameter ψ. Such a parameter is also used or wave speed interpolation by the semi-implicit finite element techniques presented in Chapter 8. When hyperbolic systems are dealt with, with waves traveling in opposite directions, using $\psi = 1/2$ is advised because this allows the waves with positive and negative speeds to be treated in a symmetrical way.

11.4.3. *Iteration control*

Implicit schemes for the solution of nonlinear systems involve iterative procedures. Iterations are also needed in the case of the Alternate Directions Implicit (ADI) technique for the solution of multidimensional systems (see section 6.9.2). The question arises of the criteria used to assess the degree of convergence of the iterative procedure. Four options are available:

– the number of iterations is pre-defined by the user of the numerical technique. The same number of iterations is made at each computational time step, regardless of the degree of convergence of the solution at the end of the iterative loop;

– the user defines iteration stop criteria. Convergence may be checked by computing the residual of the system to be solved, or the difference between two successive values of the solution from one iteration to the next. Iterations are stopped when the residual or the difference between two successive iterations falls below a predefined threshold. Some packages also allow a maximum permissible number of iterations to be defined. The iterative procedure is stopped when this maximum number of iterations is reached, regardless of the state of convergence of the solution;

– the number of iterations and/or the convergence criteria are determined automatically by the software. Although interesting at first sight because this does not require the supervision of an experienced user, this option is questionable because the user loses any control on the degree of accuracy of the solution;

– some packages simply perform no iteration, which allows the computational cost of the solution procedure to be reduced dramatically. This, however, may result in strongly degraded solutions.

If a convergence criterion is to be specified by the user, it may be worth documenting the formula used. In general, two options are available in computational hydraulics packages:

– convergence is checked locally: the residual of the system to be solved (or the difference between two successive iterations) is computed at each point (or cell, or node) of the computational grid. Convergence is achieved if each of the computational points falls below the predefined iteration stop criteria. Convergence implies that the numerical solution is satisfied at all points with a satisfactory degree of accuracy. The drawback of this approach is the large number of iterations often required;

– convergence may be checked on average over the computational domain. For instance, an average value is computed for the residual over the entire domain; or the average value of the difference between two successive iterations is used for comparison with the convergence criteria. This latter approach is faster than the former because there are always areas in the computational domain where convergence is achieved faster than in other areas. The solution may be considered "converged" in an average sense even though there are regions in the domain where convergence is far from being achieved. The resulting error may propagate into the rest of the computational domain at later times. For this reason, specifying strict convergence criteria may prove beneficial in the long term, even though the solution process may be slowed down at early simulation stages.

11.5. Simplifications in the governing equations

11.5.1. *Rationale*

Some of the numerical techniques presented in Chapter 6 are not equipped to deal with transcritical flow configurations. This is the case with the Preissmann scheme presented in Chapter 6. Although transcritical versions of the scheme have been proposed in the literature [JOH 02], they do not seem to have been implemented in industrial packages. To overcome this problem, practical implementations of these schemes in engineering free surface flow modeling software often solve a simplified version of the equations. In these software

packages, the governing equations are simplified so as to guarantee that the waves $(u-c)$ and $(u+c)$ always propagate in opposite directions. Examples of such techniques are the Local Partial Inertia (LPI) and the Reduced Momentum Equation (RME) approaches. Practical consequences of these techniques are detailed in [NOV 10], only the broad lines of the techniques are given here.

11.5.2. *The Local Partial Inertia (LPI) technique*

The LPI approach [JIN 00] consists of multiplying the inertial terms $\partial Q/\partial t$ and $\partial(Q^2/A)/\partial x$ in the Saint Venant equations by a coefficient ε that decreases from one to zero as the absolute value of the Froude number approaches unity:

$$\varepsilon\frac{\partial Q}{\partial t}+\varepsilon\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)+\frac{\partial}{\partial x}\left(\frac{P}{\rho}\right)=(S_0-S_f)gA \qquad [11.15]$$

This amounts to dividing the other terms of the equation by ε:

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)+\frac{1}{\varepsilon}\frac{\partial}{\partial x}\left(\frac{P}{\rho}\right)=\frac{1}{\varepsilon}(S_0-S_f)gA \qquad [11.16]$$

The Jacobian matrix A is modified into:

$$\mathrm{A}=\begin{bmatrix}0 & 1\\ c^2/\varepsilon-u^2 & 2u\end{bmatrix} \qquad [11.17]$$

The eigenvalues of A are:

$$\left.\begin{aligned}\lambda^{(1)}&=u-\varepsilon^{-1/2}c\\ \lambda^{(2)}&=\frac{u}{\varepsilon}+\varepsilon^{-1/2}c\end{aligned}\right\} \qquad [11.18]$$

These eigenvalues are identical to the exact eigenvalues $u-c$ and $u+c$ only when $\varepsilon=1$. When the Froude number tends to one, ε tends to zero and the eigenvalues $\lambda^{(1)}$ and $\lambda^{(2)}$ as given by equation [11.16] tend to infinity. For Froude numbers larger than one, $\varepsilon=0$ and the diffusive wave approximation is obtained. The diffusive wave model is not a hyperbolic model. Figure 11.9 shows the

variations of $\lambda^{(1)}/c$ and $\lambda^{(2)}/c$ with the Froude number with a weighting function ε given by:

$$\varepsilon = \max(0,1 - \mathrm{Fr}^2) \qquad [11.19]$$

The theoretical wave speeds are $\lambda^{(1)}/c = \mathrm{Fr} - 1$ and $\lambda^{(2)}/c = \mathrm{Fr} + 1$. As illustrated in Figure 11.9, the wave speeds in the LPI approach depart from the theoretical values as the absolute value of the Froude number approaches unity. This should be expected because a large Froude number value means that the inertial terms play a significant role in the momentum equation. Neglecting these terms can only lead to incorrect wave speed estimates.

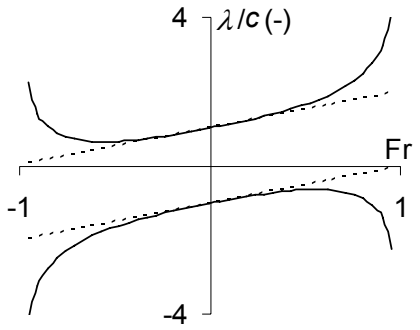

Figure 11.9. *Wave speeds given by the LPI approach. Dashed lines: theoretical. Solid lines: equations [11.18–19]*

11.5.3. *The Reduced Momentum Equation (RME) technique*

The RME approach [DHI 05] is similar in essence to the LPI approach, except that only the derivative $\partial(Q^2/A)/\partial x$ in the Saint Venant equations is multiplied by the weighting coefficient ε in the momentum equation:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\varepsilon \frac{Q^2}{A} + \frac{P}{\rho}\right) = (S_0 - S_f)gA \qquad [11.20]$$

The behavior of ε with the Froude number is the same as in the LPI approach: ε decreases from one to zero when the absolute value of the Froude number

increases from 0 to one. Neglecting the variations in ε with Q and A, the following Jacobian matrix is obtained:

$$\mathrm{A} = \begin{bmatrix} 0 & 1 \\ c^2 - \varepsilon u^2 & 2\varepsilon u \end{bmatrix} \qquad [11.21]$$

The main difference with the LPI approach is that the system remains hyperbolic under supercritical conditions. The eigenvalues of A are:

$$\left.\begin{aligned} \lambda^{(1)} &= \varepsilon u - \left[(\varepsilon - 1)\varepsilon u^2 + c^2\right]^{1/2} \\ \lambda^{(2)} &= \varepsilon u + \left[(\varepsilon - 1)\varepsilon u^2 + c^2\right]^{1/2} \end{aligned}\right\} \qquad [11.22]$$

Figure 11.10 shows the variations of $\lambda^{(1)}/c$ and $\lambda^{(2)}/c$ with the Froude number with ε defined as in equation [11.19]. Note that the modified wave speeds do not change sign with the Froude number. This makes it impossible to reproduce supercritical flow conditions. For large Froude numbers, the wave speeds are equal to the propagation speeds of the waves in still water. As in the LPI technique, the inertial terms are neglected in the range of Froude numbers where they are predominant.

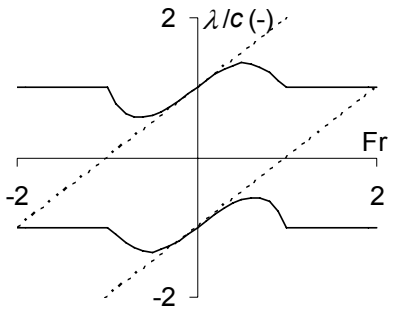

Figure 11.10. *Wave speeds given by the RME approach. Dashed lines: theoretical. Solid lines: equations [11.19–20]*

11.5.4. *Application examples*

11.5.4.1. *Steady flow over a bump*

The RME technique is applied to the steady state shallow water test case presented in Chapter 9. The description and parameters of the test case can be found

in section 9.6. The reference solution is the solution obtained using the Auxiliary Variable-based Balancing (AVB) technique presented in section 9.5, with equations [9.91] and [9.101] for the estimate of Δh. This solution is plotted on the left-hand side of Figure 11.11.

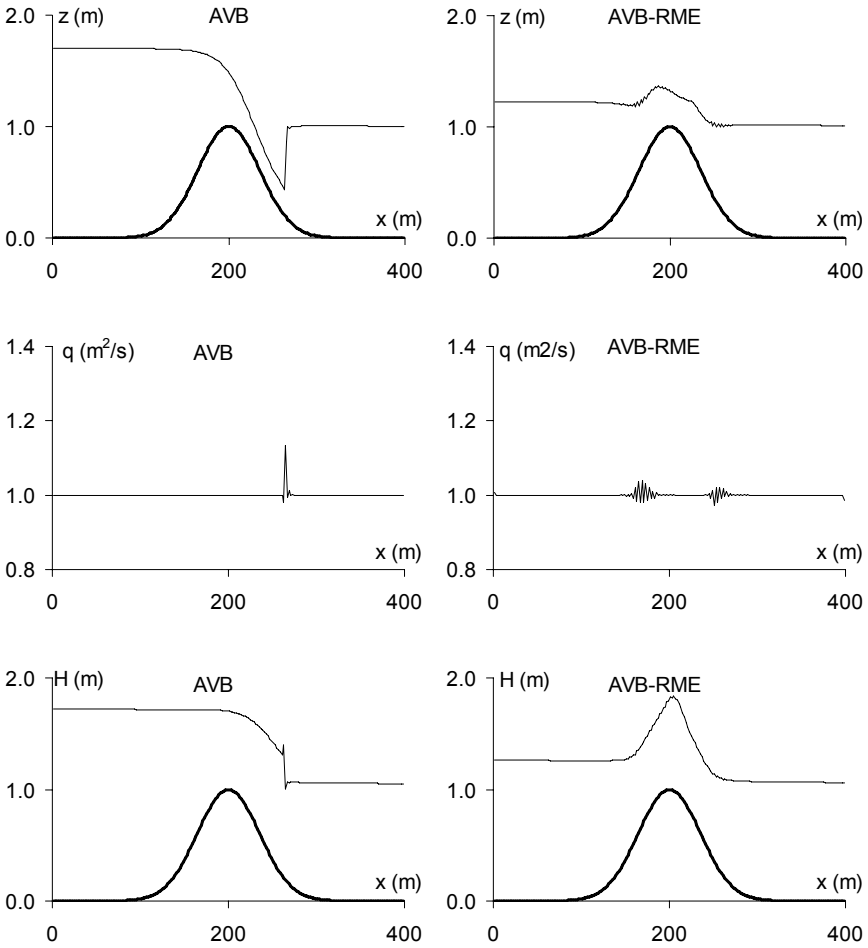

Figure 11.11. *Steady flow over a bump computed using the Auxiliary Variable-based Balancing approach (AVB) and the AVB approach combined with the Reduced Momentum Equation approach (AVB-RME). Top: free surface elevation. Middle: unit discharge. Bottom: hydraulic head*

The profiles on the right-hand side of Figure 11.11 are the water level, unit discharge and hydraulic head obtained by applying the RME approach and the AVB technique to the calculation of the fluxes. Note that the RME approach influences only the estimate of the momentum flux within the computational cells, by multiplying the term q^2/h with the coefficient ε.

The most striking feature is the impossibility for the RME approach to reproduce the hydraulic jump on the downstream side of the bump. This was to be expected because the RME approach does not allow supercritical conditions to be reproduced. The consequence is a local increase in the water level across the bump and a reduced head loss because friction is reduced due to the artificially increased water depth.

The peak discharge observed across the hydraulic jump in the original AVB approach is replaced with small amplitude oscillations on both sides of the bump. Another striking feature is the artificial increase in the hydraulic head induced by the RME approach across the bump. While the original AVB approach classically computes a decreasing head profile from upstream to downstream, applying the RME approach yields an increase on both sides of the bump. The singular head loss at the location of the jump is correctly identified by the AVB solution. Since the hydraulic jump cannot be represented in the RME approach, this singular head loss is not represented in the RME solution.

11.5.4.2. *Dambreak problem*

The RME technique is applied to the dambreak problem presented in section 10.3.4, with the parameters given in Table 1. Remember that the dambreak problem is a Riemann problem, for which an analytical solution is available (see Chapter 4, and more specifically section 4.3.3).

The numerical solution is obtained using a first-order finite volume scheme with a HLL Riemann solver, where the fluxes are computed using the RME approach. The cell size and time step are respectively $\Delta x = 1$ m and $\Delta t = 2\times10^{-2}$ s. Δt is approximately three times as small as the maximum permissible time step allowed by the stability constraint of the scheme. However, experience shows that using larger values for Δt yields sharp oscillations in the computed profiles in the transcritical region of the solution.

The numerical solution at $t = 30$ s is compared to the analytical solution in Figure 11.12. The formula of the flux being modified via the coefficient ε, the speed of the shock is modified in the numerical solution compared to the analytical solution. The solution in the intermediate region of constant state thus differs from the analytical solution.

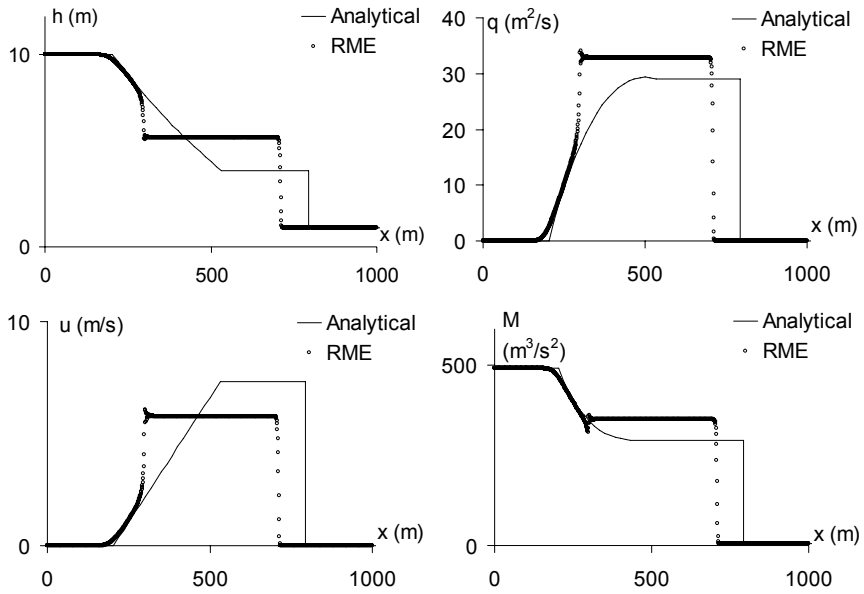

Figure 11.12. *Dambreak problem. Analytical solution and numerical solution obtained using the RME approach*

11.6. Numerical solution assessment

11.6.1. *Software solution accuracy*

The modeler's main concern is that the software package used should actually be able to solve the governing equations with a reasonable degree of accuracy. This leads to the key notion of numerical convergence: the numerical solution is said to converge to the analytical solution if it converges uniformly to it as the time step Δt and cell width Δx tend to zero. Solution convergence should be viewed as a necessary condition to model applicability in that it guarantees that the "actual" solution can be computed numerically with any arbitrary degree of accuracy provided that sufficient computational effort is spent in refining the discretization of space and time.

Convergence is related to the notions of consistency and stability. The purpose of this section is to give an overview of consistency, stability and convergence issues. Consistency and stability analysis techniques are presented in Appendix B.

11.6.2. *Assessing solution convergence*

A widespread way of assessing the convergence of the numerical solution consists of solving numerically a test case for which an analytical solution is available. In the case of a one-dimensional, scalar conservation law solved over a solution domain $[0, L]$, a numerical solution U_i^n and an analytical solution $U(x_i, t^n)$ is assumed to be available at all points x_i of the discretized domain $[0, L]$ for a given time t^n. The pointwise error, defined as the difference between the numerical and analytical solution, is computed as:

$$e_i^n = U_i^n - U(x_i, t^n) \qquad [11.23]$$

The L_p–norm of the error (where p is an integer) is given by the numerical integral:

$$L_p = \left[\frac{1}{L} \sum_i^N \left| e_i^n \right|^p \Delta x_i \right]^{1/p} \qquad [11.24]$$

The most widely used measure of error is the L_2–norm. Sometimes, the L_∞–norm is used. It is obtained as the limit expression of equation [11.24] when p tends to infinity as:

$$L_\infty = \max_i \left| e_i^n \right| \qquad [11.25]$$

Solution convergence is achieved by computing the numerical solution at a given time for different values of Δx. Note that Δt must be decreased proportionally to Δx so as to preserve a constant value for the Courant number and to preserve the stability properties of the solution. The L_p–norm is computed for each of these values of Δx. If L_p tends to zero as Δx tends to zero, the numerical solution is convergent.

The order of convergence of the solution is said to be α if a power law can be fitted to the experimental pairs $(\Delta x, L_p)$:

$$L_p = K \, \Delta x^\alpha \qquad [11.26]$$

where K is a constant. A very simple way of estimating α consists of plotting L_p and Δx along logarithmic axes. α is the slope of the straight line that can be fitted to the set of experimental points $(\log(\Delta x), \log(L_p))$. Figure 11.13 shows an example of such

a graph. The solution given by Scheme 1 (circular dots) can be approximated by a straight line with slope 1 in logarithmic coordinates (the L_2–norm of the error is multiplied by 10^2 when Δx is multiplied by 10^2). The solution given by Scheme 2 (cross-shaped dots) can be approximated with a line with slope 2 (the L_2–norm of the error is multiplied by 10^4 when Δx is multiplied by 10^2). Therefore, the orders of convergence of the solutions obtained by Schemes 1 and 2 are respectively 1 and 2.

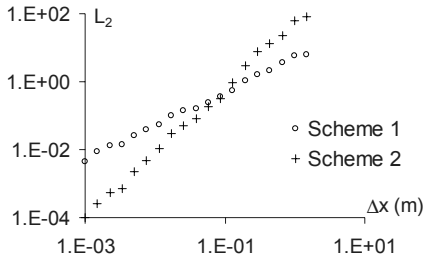

Figure 11.13. *Examples of L_2–norm of the error as a function of Δx for two different numerical schemes*

11.6.3. *Consistency analysis – numerical diffusion and dispersion*

For the sake of simplicity, consider a scalar conservation law in the form [1.1], recalled here:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = S$$

As shown in Appendix B (see section B.1), discretizing equation [1.1] (or its non-conservation, or characteristic form) leads us to solve a different equation that can be written in the form:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = S + \mathrm{TE}(\Delta x, \Delta t) \qquad [11.27]$$

where the difference TE between the discretized equation and the original equation is called the truncation error. While the purpose is to solve equation [1.1], equation [11.27] is solved instead. The truncation error is made of an infinite sum of elementary terms formed by powers of Δx and Δt. These terms also include the higher-order derivatives of U with respect to x and t:

$$\begin{aligned}\mathrm{TE}(\Delta x,\Delta t) &= \sum_{p=1}^{\infty} k_p \Delta x^{a_p} \Delta t^{b_p} \frac{\partial^{c_p} U}{\partial x^{d_p} \partial t^{c_p - d_p}} \\ &= \sum_{p=1}^{\infty} k_p (\lambda \mathrm{Cr})^{a_p} \Delta t^{a_p + b_p} \frac{\partial^{c_p} U}{\partial x^{d_p} \partial t^{c_p - d_p}} \\ &= \sum_{p=1}^{\infty} k_p \frac{\Delta x^{a_p + b_p}}{(\lambda \mathrm{Cr})^{b_p}} \frac{\partial^{c_p} U}{\partial x^{d_p} \partial t^{c_p - d_p}}\end{aligned} \quad [11.28]$$

where a_p, b_p, c_p and d_p are integer powers for the pth term in the truncation error and k_p is a coefficient that usually depends on the Courant number $\mathrm{Cr} = \lambda \Delta t/\Delta x$.

Section B.1 in Appendix B shows how the expression for the truncation error is derived from the numerical scheme. When Δx and Δt tend to zero, the terms with smaller powers of Δx and Δt in equation [11.28] become predominant over the other terms. In the field of numerical methods for hyperbolic conservation laws, two main situations occur:

(1) Numerical diffusion occurs when the truncation error [11.28] can be written in the form:

$$\mathrm{TE}(\Delta x,\Delta t) = D(\Delta x,\Delta t)\frac{\partial^2 U}{\partial x^2} + \mathrm{HOT} \quad [11.29]$$

where HOT represents the sum of the higher-order terms in the sum [11.26]. D is a so-called numerical diffusion coefficient that usually depends on Δt, Δx and possibly the Courant number.

(2) Numerical dispersion is encountered when the truncation error [11.28] can be written in the form:

$$\mathrm{TE}(\Delta x,\Delta t) = k_1(\Delta x,\Delta t)\frac{\partial^3 U}{\partial x^3} + k_2(\Delta x,\Delta t)\frac{\partial^3 U}{\partial x^2 \partial t} + \mathrm{HOT} \quad [11.30]$$

where k_1 and k_2 are so-called dispersion coefficients.

Numerical diffusion and numerical dispersion influence the behavior of the numerical solution in very different ways (Figure 11.14). Numerical diffusion tends to smooth out the computed profiles (Figure 11.14a), thus leading to the damping of transients via dissipation of the energy contained in the solution signal. Numerical dispersion, in contrast, modifies the speed at which the various components of the solution propagate. Shifting these components with respect to each other introduces

oscillations in the computed profiles, especially in the neighborhood of steep gradients (Figure 11.14a).

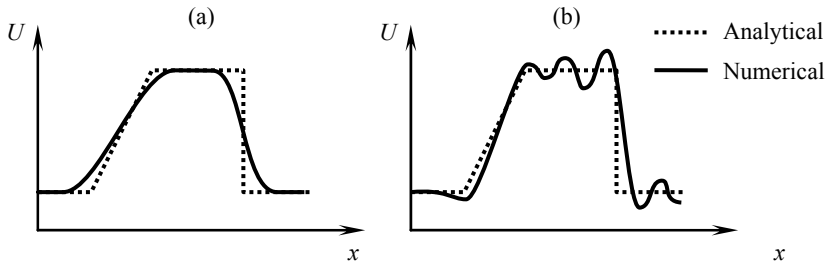

Figure 11.14. *Typical effects of numerical diffusion (a) and numerical dispersion (b) on the numerical solution*

Numerical dispersion destroys the TVD character of the numerical solution. In the field of linear PDEs or hyperbolic systems (passive scalar transport, see section 1.3; or water hammer problems, see section 2.4), numerical dispersion may simply result in non-physical results, such as negative concentrations or pressures. If the PDEs to be solved are nonlinear, numerical dispersion may result in nonlinear instability.

11.6.4. *Stability analysis – phase and amplitude portraits*

Linear (or harmonic) stability analysis provides valuable information on the performance of the numerical technique used. How amplitude and phase portraits should be derived for a given scheme is dealt with in section B.2.5 (Appendix B). The amplitude portrait of a scheme indicates how a harmonic component with wave length M in the solution is amplified from one time step to the next by the numerical scheme. The amplification factor $|A_N|$ is usually given as a function of the so-called wave number M:

$$M = \frac{L}{\Delta x} \qquad [11.31]$$

The minimum possible value for M is 2 (at least two points are needed to represent one period of a harmonic component). Applying the numerical scheme over k computational time steps to a harmonic component of wave length L with a cell width Δx yields an amplification by a factor $|A_N|^k$. A steep front or discontinuity in the solution is represented by half a wavelength that is $L/2$.

Assume for instance that the explicit upwind scheme is used to compute the solution of the linear advection equation over 100 time steps ($k = 100$) with a Courant number $Cr = 0.5$. The amplitude portrait of the scheme is shown in Figure B.2. The numerical values of $|A_N|$ and $|A_N|^k$ ($k = 100$) are shown in Table 11.1 for various values of M.

M	$\lvert A_N\rvert$	$\lvert A_N\rvert^{100}$	M	$\lvert A_N\rvert$	$\lvert A_N\rvert^{100}$
2	0.0	0.0	30	0.995	0.58
10	0.951	6.6×10^{-3}	50	0.990	0.82
16	0.981	0.14	100	0.999	0.95
20	0.988	0.29	200	0.9999	0.99

Table 11.1. *Amplification factor for Cr = 0.5 as a function of the wave number M*

As the table indicates, if 16 cells are used to describe one wavelength of the harmonic component (8 cells for half a wavelength), only 14% of the initial amplitude remains after 100 time steps at $Cr = 0.5$. If 30 cells are used for one period (15 cells for half a wavelength), the harmonic component is damped by more than 40% after 100 computational time steps. The practical consequence of this is that an initially discontinuous profile is spread artificially over 8 to 15 cells within 100 computational time steps. Operating the explicit upwind scheme at $Cr = 0.5$ requires that any steep front in the initial profile be represented by at least 8 cells for the numerical solution to be reasonably accurate after 100 computational time steps. This is why the first-order upwind scheme is classically admitted to require approximately 10 cells to represent a steep front or a discontinuity.

11.7. Getting started with a simulation package

Sections 11.1 to 11.5 only provide examples of problems that may occur when industrial computational hydraulics packages are used. The list of issues raised in these sections is not exhaustive. The modeler's responsibility is to be aware of the limitations and weaknesses of the algorithms implemented in the software package he is using. Experience shows that the user of a modeling package quickly becomes used to this tool and very often makes the algorithms and solution techniques an integral part of his way of thinking and interpreting the modeled reality.

For this reason, the "getting started" phase every modeling software user must go through when learning to use a specific package is extremely important because it largely conditions the future perception of (and way of using) the software by the

user. It is believed that the following recommendations will help any modeler when learning to use a new modeling package:

(1) *Always read the manuals*. Very often, commercially available or public domain computational hydraulics/fluid dynamics packages are provided with a detailed documentation that includes a description of the governing equations, solution algorithms and user options available. Reading the documentation thoroughly most often allows the experienced user to identify the possible weaknesses and limitations of the modeling software package.

(2) *Always run the sample test cases provided with the package*. A complete package documentation should include sample test files, with test cases for which analytical solutions are available (dambreak problem for the open channel equations, sudden valve failure for the water hammer equations, etc.) Solving these test cases often provides a fair idea of the degree of accuracy of the solution techniques.

(3) *Assess the sensitivity of the package to the numerical parameters*. Complying with step (2) is not sufficient. Sample test cases are often chosen to demonstrate the ability of the software to reproduce theoretical solutions, hence proving its accuracy. It may happen, however, that modifying slightly some of the numerical parameters (time centering coefficient, computational time step, iteration convergence criteria, etc.) leads to a strong degradation in the quality of the numerical result. If the package is oversensitive to the numerical parameters used, it is always worth knowing.

(4) *Invent your own test cases*. Since the purpose of a software vendor is to sell the software packages, situations where the package performs poorly are usually not documented, and no sample test cases are provided for them. For this reason, it is advised not to restrict the test phase to step (2) and (3). The user should invent his/her own test cases and test the physical plausibility of the solution, even when no analytical solution is available for comparison.

The main prerequisite in modeling is that the modeler be critical about the numerical results provided by the software. Being critical is all the more difficult as user-friendly modeling tools increasingly put an emphasis on a realistic presentation of the simulation results, thus giving the impression that what is being displayed on the screen is reality. Modeling only provides an approximation of reality and numerical techniques are essentially inaccurate. For this reason, the modeler's critical judgment remains an essential feature of the modeling process. If one piece of advice should be given to a newcomer in the world of modeling, it may be the following: the model is and must remain a tool. Use it, never let it use you.