Numerical implementation of (absorbing) boundary conditions in 1D (and in 2D)

Mail 14 december 2023 (19:17), Mart Borsboom

Hoi Frank,

It is becoming important to figure out and write down all details with respect to the boundary conditions. I was working on that in SMART numerics context until about a year ago, when it was put on hold after the decision to keep the numerical implementation of boundary conditions ‘simple’ for the time being by using only 2 grid points normal to the boundary. Although still not fully analyzed, it is clear since 2009 that the use of 3 points gives substantial improvements. Recent results obtained by Jan indicate that 3 points are actually required to avoid possible problems due to wiggles (didn’t see that one before, oops). Since Jan is using these 3 points now (to avoid wiggles when zero physical and numerical/artificial viscosity/diffusion is present inside the domain for the suppression of wiggles), it is high time to write down in some detail how that works. I did that briefly a month ago in an e-mail to Jan (in Dutch). In this e-mail an extended and more complete yet still concise description in English. Many details and explanations still to be added.

Warning: text below may not be 100 % error-free.

When using a collocated scheme, we have at boundaries the same number of unknowns per grid point/cell as inside the domain, hence at boundaries the same number of equations per grid point/cell is required as inside the domain. Boundary conditions provide only a part of these equations. The remaining equations are obtained from (a combination of) the equations applied inside the domain, describing how information propagates from inside the domain outward (the genuine boundaries describe how information is reflected back into the domain and how information from outside the domain travels inward). Jan started to refer to these two groups of equations by the names used in the FE world: essential boundary conditions and natural boundary conditions. Although convenient names, I do not like the term ‘natural boundary condition’. These are not conditions, but procedures applied at or near boundaries (i.e., not necessarily at the genuine boundary) that follow from the physical model and its numerical implementation inside the domain. Let’s go for convenience for the moment and the use of the terms essential and natural boundary condition, pending a better nomenclature.

Consider the 1D shallow water equations formulated in the unknowns h ($\ln h$ is p.m.) and q, i.e., eqs (5.1) or (5.2) in the report “SMART numerics (1D)” by Jan, from here on referred to as Mooiman (2023) (version of 7 December 2023; this is work in progress, so equation numbering may change). For convenience, omit the convection term (easy to include) $\Rightarrow$ Froude number is ‘zero’ $\Rightarrow$ solution inside domain consists of a left-going signal and a right-going signal $\Rightarrow$ solution variations dq and dh can be written as sums of left-going variations and right-going variations: $dh = dh^- + dh^+$ and $dq = dq^- + dq^+$. This is different from (5.93) and (5.94) in Mooiman (2023), which expressions assume linear behavior. It is straightforward to show that the left-going and the right-going characteristic information are (compare with (5.95) in Mooiman (2023)) $\sqrt{gh}\,dh^- - dq^-$ and $\sqrt{gh}\,dh^+ + dq^+$. Since left-going information does not carry information moving from left to right and vice versa, we have by definition $\sqrt{gh}\,dh^- + dq^- = 0$ and $\sqrt{gh}\,dh^+ - dq^+ = 0$ (compare with (5.96)) $\Rightarrow$ $\sqrt{gh}\,dh - dq$ and $\sqrt{gh}\,dh + dq$ are the left-going and right-going variations. Neglecting the effect of a bed slope (and bed friction, . . .), these variations are zero along characteristics (Riemann invariants). NB, we may not need that splitting in left-going and right-going components per variable. It is something that I have used and found useful in the construction of non-reflecting and partially reflection boundary conditions for a Boussinesq-type wave model, see Borsboom et al. (2001).

The natural condition at the left boundary (similar procedure at the right boundary) is the equation that describes the dynamic behavior of outgoing $\sqrt{gh}\,dh - dq$ information at or near that boundary. That equation is $\sqrt{gh}$ times continuity equation minus momentum equation, cf. (5.114) in Mooiman (2023) (bed-slope term etc. to be added). NB, the equations (5.110) to (5.112) are not convenient and assume linear behavior. For the software implementation of (5.114) I would start by discretizing all model equations separately at the boundary (including linearization etc., i.e., altogether similar to the procedure used inside the domain), followed by combining them to whatever combination(s) is/are required (we have been discussing this several times in KISS). In case of supercritical outflow all model equations are to be applied at a boundary while no conditions are to be imposed. In all other cases, the construction of the combination(s) to be applied will free the memory space required for the application/discretization of the genuine/essential boundary conditions. In case of supercritical inflow, no (combination of) model equations are applied and all variables are imposed by boundary conditions. I found (and still find) this procedure very convenient for the handling of changing flow regimes at boundaries (from sub- to supercritical out- or inflow and back again), cf. SUPSUB tests of (nearly?) 25 years ago that I must have somewhere.

The discretization of natural conditions is always at the outer boundary of the collection of inner volumes, cf. the orange line in the Figure below. In 2D (not relevant here) we have discretizations along the boundary at edge volumes, in convex corners at corner volumes, and along the boundary around concave corners at compound edge volumes (not in the Figure). The genuine boundary always lies somewhere inside the outermost layer of finite volumes (e.g., the blue line in the Figure), i.e., the grid must be such that that is the case. This generalizes to arbitrarily shaped boundaries (provided they are sufficiently smooth) and the discretization of genuine/essential boundary conditions by means of a cut-cell technique (see below in the Ps's (Figure 1).

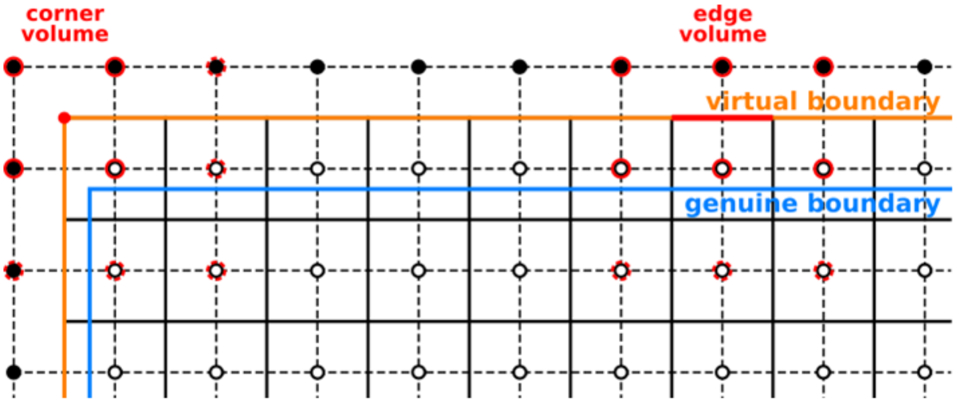

Figure 1: *Different type of boundary location and control volumes.*

Dashed lines indicate grid lines; solid lines indicate the inner finite volumes (dual grid). Open black circles indicate inner grid points for which we have a full set of FVE discretizations. Black dots indicate virtual grid points for which we (must) have a full set of discretizations of essential and natural boundary conditions. The full red circles indicate the grid points that are used in a computational stencil at boundaries when only 2 grid points normal to the boundary are used. When 3 points normal to the boundary are used in the boundary stencils also the dashed red circles are used.

In 1D we have the situation as shown in Fig. 5.1 of Mooiman (2023), which we have copied below. For convenience, I am going to use a computational coordinate system with unit grid size $\Delta\xi = 1$ and grid point coordinates $\xi_0 = 0$, $\xi_1 = 1$, etc. The natural boundary condition at the left side is imposed/applied on FV face $\xi = 1/2$ using the 3 grid points ξ_0, ξ_1 and ξ_2.

In SUPSUB I have used 2-point stencils at boundaries, using locally the same linear approximation of variables per grid cell as used inside the domain, which seemed like an obvious approach. I later realized that this leads to non-negligible errors because of the mismatch between the scheme used inside the domain (integration over inner FVs) and that used at boundaries (evaluation at boundary volumes). That error can be reduced significantly by using an additional grid point, extending the computational molecule used at boundaries to 3 (by 3). The same computational-stencil size as used inside the domain, hence acceptable.

I learned the Fourier-mode analysis for the optimization of the discretization of natural boundary conditions from the 44-page article Vichnevetsky (1987), one of the few papers on numerical analyses and techniques that I consider must-read and must-know. It is not an easy paper to read, especially when one is not familiar with the concepts phase speed and group speed. It helps to first read the fairly easy paper Trefethen (1982). Pending full documentation and full analysis, sufficient for now is to know that the discretization of the natural boundary condition that best fits to the FVE scheme applied inside the domain is obtained by evaluating that condition at $\xi = 1/2$ using the parabolic interpolation:

$$\begin{aligned} c_{GP}(\xi) &= c_0\,(1-\xi) + c_1\,\xi + (c_0 - 2\,c_1 + c_2)/2\,(\xi - 1)\,\xi = \\ &= c_0\,(1-\xi)\,(1-\xi/2) + c_1\,\xi\,(2-\xi) + c_2\,(\xi-1)\,\xi/2 \end{aligned} \tag{1}$$

In the first expression the interpolation is written as a quadratic correction to the ‘standard’ linear interpolation. From the second expression we immediately obtain $c_{GP}(\xi = 0) = c_0$, $c_{GP}(\xi = 1) = c_1$ and $c_{GP}(\xi = 2) = c_2$., i.e., this is a parabolic interpolation through the grid-point values (whence the subscript GP: Grid Point). Note that that is not a design choice, but what appears to be optimal for the space discretization of natural boundary conditions (pending full analysis and full testing with full documentation) when the FVE scheme with piecewise linear function approximations is used inside the domain. Using (1) that discretization is obtained straightforwardly from the transformation of the equations to computational space followed by evaluation at $\xi = 1/2$ (which is why we propose to work with (1), since that gives us a simple recipe to handle any model equation at boundaries not containing derivatives higher than 2). The continuity equation as example:

$$\begin{aligned} &\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0 \Rightarrow \frac{\partial x}{\partial \xi}\frac{\partial h}{\partial t} + \frac{\partial q}{\partial \xi} = 0 \Rightarrow \\ &(x_1 - x_0)\frac{\partial}{\partial t}(\frac{3}{8}h_0 + \frac{3}{4}h_1 - \frac{1}{8}h_2) + (q_1 - q_0)/2 = 0 \end{aligned} \tag{2}$$

In Section 2.1.1.2 of Mooiman (2023) the equations as used in the analysis/optimization of the space discretization of an outflow/natural boundary condition. The analysis (in Maple; full documentation pending) shows that $a_1 = 0$ and $a_2 = -1/4$ is optimal $\Rightarrow$ Equation (2).

Discretization/integration in time is as always: Crank-Nicolson with $\theta = 0.5$ (or slightly higher for some dissipation to suppress spurious behavior, which should not be required). Important detail: also the coefficient $\sqrt{gh}$ in the combination “$\sqrt{gh}$ times continuity equation minus momentum equation" is to be evaluated at $\xi = 1/2$, using Equation (1) to get h at that location and discretized in time using Crank-Nicolson. I would not include linearization of $\sqrt{gh}$ in the Newton iterative solver. That does not fit in the procedure to first discretize and linearize the model equations separately at boundaries, followed by combining them to equation(s) describing how information propagates outward. That procedure requires explicit evaluation of the coefficients involved in getting the appropriate combinations. Two options: either we evaluate $\sqrt{gh}$ at the previous time level n (probably often good enough), or we evaluate $\sqrt{gh}$ at time level $n + 1/2$ using the previous iteration, with some underrelaxation for stability if necessary. Underrelaxation is what I apply in SUPSUB for extremely nonlinear discretization-error-dependent artificial viscosity coefficient that is impossible to linearize. Works very well.

As for the essential condition at the left boundary (similar procedure at the right boundary) we consider imposing a non-reflecting condition (other conditions p.m.). This means imposing $\sqrt{gh}\, dh + dq$ (see the third paragraph above), i.e.:

$$\sqrt{gh}\, dh + dq = df_{in} - eps_{corr}\, (\sqrt{gh}\, h + q - f_{in}) \tag{3}$$

with $df_{in} = \sqrt{gh}\, dh_{in} + dq_{in}$ the information to be moved from outside to inside the domain, e.g., $df_{in} = 0$ (no incoming signal). We will consider the d's to be solution variations in time at certain locations, i.e., at boundaries. The second term in the right-hand side, with $eps_{corr} << 1$ and $f_{in} = \sqrt{gh}\, h_{in} + q_{in}$ (e.g., $f_{in} = \sqrt{gh}\, h_{still}$ if no incoming signal), is a correction term to avoid that imposing only variations in time leads to a drifting away of values at boundaries due to discretization, convergence and round-off errors. A similar correction term has been used in Borsboom et al. (2001). Note that for $eps_{corr} = 1$ expression Equation (3) becomes $\sqrt{gh}\, h + q = f_{in}$. This boundary correction is non-reflecting when linear behavior can be assumed, but the larger the nonlinear effects the larger the reflection that it will introduce. NB, using the correction term in Equation (3) (i.e., an $eps_{corr} > 0$) will also introduce reflections due to nonlinear behavior, but at a much lower level provided that $eps_{corr} << 1.$

First the time discretization of Equation (3):

$$\begin{aligned}&\sqrt{g\, h^{n+1/2}}\, (h^{n+1} - h^n) + (q^{n+1} - q^n) = \\ &(f_{in}^{n+1} - f_{in}^n) - eps_{corr}\, (\sqrt{g\, h^{n+1/2}}\, h^n + q^n - f_{in})\end{aligned} \tag{4}$$

Depending on the size of the time step, using first-order accurate discretization $\sqrt{g\, h^n}$ instead of second-order accurate discretization $\sqrt{g\, h^{n+1/2}}$ (i.e., evaluation of $\sqrt{gh}$ at previous time level) may be a reasonable approximation when nonlinear effects are moderate (waves of limited steepness at boundaries). As for applying approximation $\sqrt{gh^{n+1/2}}$ and solving Equation (4) iteratively, explicit updates with underrelaxation of $\sqrt{gh^{n+1/2}}$ (same procedure as before) will often work, although in highly nonlinear situations full linearization is expected to be beneficial to fast convergence. NB, for consistency it is probably important to use the same discretization/evaluation of coefficient $\sqrt{gh}$ in all terms of Equation (4) where it appears.

In view of the future extension to a cut-cell technique and the possibility of having the location of the boundary anywhere with respect to grid lines, we need a space discretization of (3) that allows for that. With reference to the 1D figure above, this means that it should be possible to have the left boundary anywhere in the interval $[1/2, 3/2)$, i.e., somewhere in the outermost left finite volume. (If the boundary is elsewhere, then grid points and volumes should be added or removed to arrive at that situation.) The simplest way to realize that would be to interpolate grid point values to the location of boundaries by means of the same piecewise linear function approximations used inside the domain. For a boundary at $\xi = 1$ we would then impose values like h_1, while for a boundary at $\xi = 1/2$ this would lead to interpolations like $(h_0 + h_1)/2$. It has recently become clear from computations by Jan that the latter is 'blind' for wiggle modes, i.e., with a boundary at $\xi = 1/2$ certain wiggle modes are then not prescribed and free to develop unstable behavior if not damped by dissipative mechanisms elsewhere. Another drawback of using piecewise linear interpolations in the specification of boundary conditions is the amplitude error. With a boundary at $\xi = 1$ we prescribe grid point values. Since discrete solutions take their extreme values at grid points (with linear interpolations in between), this means underspecification of the solution at that boundary. Likewise, we have overspecification of the solution when a boundary is at $\xi = 1/2$ and the average of grid point values is prescribed. A preliminary Fourier-mode analysis indicates that the difference between the two is 2 times larger than the error of the scheme applied inside the domain, which we consider unacceptable. Conclusion: a 3-point quadratic interpolation needs to be applied. (The use of more than 3 points would extend boundary discretizations to beyond 3(-by-3) point molecules.)

One possible quadratic interpolation is parabolic fit (2) through the grid points, giving an underspecification of the solution at the boundary regardless of its position. Another possible quadratic fit is the one through Cell-Centered values:

$$c_{CC}(\xi) = c_0\,(1 - \xi) + c_1\,\xi + (c_0 - 2c_1 + c_2)/2\,(\xi - 1/2)^2 \tag{5}$$

This gives $c_{CC}(1/2) = (c_0 + c_1)/2$, $(dc_{CC}/d\xi)_{\xi=1/2} = c_1 - c_0$, $c_{CC}(3/2) = (c_1 + c_2)/2$ and $(dc_{CC}/d\xi)_{\xi=3/2} = c_2 - c_1$, i.e., this is the (only) 3-point parabolic fit that is continuous and continuously differentiable. Despite this desirable property, it cannot be used to specify boundary conditions at arbitrary locations because that would give overspecifications.

Obviously, the parabolic fit that gives neither underspecification nor overspecification of imposed values at boundaries is a combination of Equation (1) and (5). It is easy to show that that combination consists of 1/3 times Equation (1) plus 2/3 times Equation (5):

$$c_{OPT}(\xi) = c_0\,(1 - \xi) + c_1\,\xi + (c_0 - 2\,c_1 + c_2)/2\,((\xi - 1)\,\xi/3 + (\xi - 1/2)^2\,2/3) \tag{6}$$

$$= c_0\,(13/12 - \xi\,3/2 + \xi^2/2) + c_1\,(-1/6 + \xi\,2 - \xi^2) + c_2\,(1/12 - \xi/2 + \xi^2/2) \tag{7}$$

Verification of its correctness by means of integration over the left outermost finite volume:

$$\int_{\xi_{1/2}}^{\xi_{3/2}} c_{OPT}(\xi)\,d\xi = \frac{1}{8}c_0 + \frac{3}{4}c_1 + \frac{1}{8}c_2 \tag{8}$$

In the right-hand side we see the weights of the mass matrix of the piecewise linear FVE method, i.e., averaged over the left outermost finite volume the quadratic function c_{OPT} equals the piecewise linear function used in the FVE scheme.

Parabolic interpolation Equation (7) is discontinuous at FV interfaces, i.e., when a boundary would move across an FV edge (e.g., from $\xi < 3/2$ to $\xi > 3/2$ or vice versa) not only the next or previous FV becomes the left outermost FV, but also the way an essential boundary condition is imposed changes slightly. A preliminary analysis shows that the latter introduces an error about 2 times smaller than the discretization error inside the domain, which for the moment we deem acceptable.

Nice to know: the whole procedure shows similarities with the error analysis of the FVE scheme. This analysis requires smooth and sufficiently differentiable close fits to the piecewise linear numerical function approximations to allow construction/derivation of the equivalent differential problem, see Section 5.1, Error analysis in space, starting on page 19 of Borsboom et al. (2001). In Figure 3 of that section an illustration of a smooth fit through the Grid Points ($\gamma = 0$) and through the Cell Centers ($\gamma = 1$). The best fit is for a gamma somewhere in between 0 and 1. From equivalent equation (34) we obtain that overall the smallest second-order error terms are obtained for $\gamma = 1/24$, i.e., for $\gamma = 2/3$., cf. bottom of page 24. This is the same value that we have used to obtain Equation (7).

Space discretization of Equation (3) is now straightforward: evaluate everything at the genuine left boundary using Equation (7). So for a boundary at $\xi = 1/2$ or at $\xi = 1$ we use things like:

$$dh(\xi = \frac{1}{2}) = \frac{11}{24}dh_0 + \frac{7}{12}dh_1 - \frac{1}{24}dh_2 \tag{9}$$

or

$$dh(\xi = 1) = \frac{1}{12}dh_0 + \frac{5}{6}dh_1 + \frac{1}{12}dh_2 \tag{10}$$

That’s all folks! (For now.)

Mart. ——-

Ps. A few words about all this in 2D. Natural boundary conditions are imposed at the outer FV virtual boundary and discretized at (compound) edge volumes and corner volumes. Along (compound) edge volumes (cf., the first figure above) we can integrate the equations as inside the domain. Normal to the boundary we use interpolation Equation (1) to evaluate terms with normal derivatives. At corner volumes we use interpolation Equation (1) in both directions. Essential boundary conditions are imposed at the genuine boundary that (somewhere in the future) can be anywhere with respect to the grid cells, so here we always need per outermost FV parabolic interpolation Equation (5) in both directions, followed by evaluation of the essential boundary condition along the boundary section per outermost FV using some (Gauss quadrature) integration rule of sufficient accuracy. Note that for the genuine boundary shown in the first figure above we could use cellwise linear interpolations along the boundary because it is aligned with the grid. It is easy to see that the full set of natural boundary condition discretizations equals the number of virtual grid points, however, this does not always seem to be the case for the set of essential boundary conditions. A solution would be to add an auxiliary equation where necessary, e.g., an equation like $\partial^4 c/(\partial\xi^2\partial\eta^2) = 0$.

Ps2. We use different parabolic approximations in natural and in essential boundary conditions. I also would have preferred them to be the same. Keeping them the same may work but is bound to reduce the simulation quality at boundaries.

Ps3. Frank, it may be a nice idea to implement the FE way of constructing natural and essential boundary conditions (i.e., the approach that Maliska applies) for comparison. Analysis confirms what I observed >30 years ago in a number of tests: spurious reflections in the form of wiggles. Not very large, but clearly visible (except when introducing well-designed programming errors, hahaha!).

Ps4. The proposed boundary implementation is an improvement over the one that we developed for Boussinesq-type wave model TRITON, cf. the 2-page abstract M. and I. (2009). In that cut-cell implementation, the virtual staircase FV boundary is just inside the genuine boundary, imposing genuine boundary conditions by means of extrapolation, which for several reasons is not accurate at all. Funny fact: the method proposed here has been ‘invented’ 4 years ago while having a beer with Peter Wellens in Utrecht, thinking of ways to improve the cut-cell methods used in fluid-structure interaction simulations (moving ships and offshore structures in waves and currents). A serious problem with those methods are the large spurious pressure spikes that are typically generated when structures move across grid cells. The proposed method has been designed for maximum smooth and regular behavior in combination with accuracy, flexibility and feasibility. It has never been implemented yet, so whether or not it solves the pressure-spike problem remains to be demonstrated.

References

Borsboom, Mart, Jacco Groeneweg, Neelke Doorn, and Marcel van Gent (2001). *Flexible boundary conditions for a Boussinesq-type wave model*. File: waves2001.pdf.

M., Borsboom and Wenneker I. (2009). “Implementation of Boundary Conditions on an Irregular Geometryby Extrapolation to the Staircase Boundary of a Cartesian Grid”. In: *Immersed Boundary Methods: Current Status and Future ResearchDirections”, 15—17 June 2009*. Amsterdam.

Mooiman, J. (2023). *SMARTnumerics (1D)*. Tech. rep. (print 2023-12-07). Deltares.

Trefethen, L. N. (1982). “Group velocity in finfite difference schemes”. In: *Society for Industrial and Applied Mathematics*.

Vichnevetsky (1987). “Wave Propagation Analysis Of Difference Schemes For Hyperbolic Equations: A Review”. In: *International Journal For Numerical Methods In Fluids* Vol. 7, pp. 409–452.