"MISSING" BOUNDARY CONDITIONS? DISCRETIZE FIRST, SUBSTITUTE NEXT, AND COMBINE LATER*

ARTHUR E. P. VELDMAN†

Abstract. A simple approach exists to prevent the need for constructing boundary conditions in situations where they are not explicitly supplied by the original analytical formulation of the problem. An example is the Poisson equation for the pressure in calculations of incompressible flow. Other examples are the streamfunction-vorticity formulation where no condition for the vorticity is present, and ADI methods where boundary conditions for the intermediate timesteps must be provided. In short, this approach can be described as follows: first discretize the equations of motion, next substitute the original boundary conditions (for the velocity), and finally combine the discrete equations (e.g., to a modified Poisson equation).

Key words. boundary conditions, discretization methods, incompressible Navier–Stokes equations

AMS(MOS) subject classifications. 65N05, 76D05

1. Introduction. When the incompressible Navier–Stokes equations are solved numerically, often boundary conditions seem to be “missing.” One example is formed by the boundary conditions for the pressure when a Poisson equation is employed. Another example is the boundary condition for the vorticity in case a streamfunction-vorticity formulation is used. Uncertainty exists about the choice of these conditions; when Neumann conditions are selected, the corresponding compatibility relation poses an additional difficulty.

Gresho and Sani [1] give an extensive discussion of the former example. They discuss a number of approaches used to solve the above problem. Their favorite approach is what they call the “direct attack.” This is a simple method that has been known for at least two decades (see the references in Chapter 6.3.1 of [2]). Gresho and Sani show that this approach circumvents the problem of the “missing” boundary conditions in a natural way.

From discussions with colleagues it became clear that this “direct attack” is applicable to many more situations where boundary conditions are “missing.” Therefore in this paper we want to highlight this approach and show some applications. It will not be surprising that the methods being obtained in this way are familiar ones. However, the way in which they have been derived ensures there is no need to distrust them, whereas other derivations of the same formulas might leave some room for distrust.

The starting point is an analytical set of equations, including boundary conditions, that is well posed and for which a unique solution exists. For the unsteady incompressible Navier–Stokes equations we may use its formulation in primitive variables (velocity and pressure). At solid walls only boundary conditions for the velocity are required to make the solution unique [3, Chap. 3, § 3]. No conditions on the pressure have to be prescribed in the continuum case. The Navier–Stokes equations will be discretized and its boundary conditions substituted. Hereafter the discrete set of equations may be combined in any way that is found convenient, e.g., to a discrete Poisson equation or to a discrete streamfunction-vorticity formulation. This shuffling of the equations does not change the solution, and hence is harmless. In short, this approach can be described as: discretize first, substitute next, and combine later.

* Received by the editors June 17, 1988; accepted for publication (in revised form) November 28, 1988.
† Faculty of Technical Mathematics and Informatics, Delft University of Technology, P.O. Box 356, 2600 AJ Delft, the Netherlands.
For those who are unfamiliar with this approach we will present it in detail for the pressure conditions. Moreover, a number of other applications will be given. Next to the streamfunction-vorticity formulation for the Navier-Stokes equations, we apply it to the shallow-water equations and to ADI methods. In Appendix A it also will be shown useful for the treatment of the constraints in differential-algebraic equations.

2. Problem. Consider a rectangular domain $\Omega$ with boundary $\Gamma$ on which the incompressible Navier-Stokes equations have to be solved:

\begin{align}
\text{(2.1a)} \quad & \text{div } q = 0, \\
\text{(2.1b)} \quad & \frac{\partial q}{\partial t} + (q \cdot \text{grad})q = -\text{grad } p + \nu \text{ div grad } q,
\end{align}

with boundary conditions

\begin{align}
\text{(2.2)} \quad & q = q^\Gamma \text{ on } \Gamma \\
\text{(often } q^\Gamma = 0). \text{ Here } q = (u, v) \text{ is the velocity vector, } p \text{ is the kinematic pressure, and } \nu \text{ is the kinematic viscosity.}
\end{align}

As the treatment of the convective terms and diffusive terms is irrelevant for the discussion that follows, the equations of motion (2.1) will be abbreviated as

\begin{align}
\text{(2.3a)} \quad & \text{div } q = 0, \\
\text{(2.3b)} \quad & \frac{\partial q}{\partial t} + \text{grad } p = \mathbf{R}.
\end{align}

The above equations can be combined to obtain a Poisson equation for the pressure $p$:

\begin{align}
\text{(2.4)} \quad & \text{div grad } p = \text{div } \mathbf{R} - \frac{\partial}{\partial t} \text{ div } q.
\end{align}

From the analytical point of view, the second term in the right-hand side of (2.4) vanishes, but it has been retained to stress that its discrete numerical treatment is nontrivial: accumulation of errors is possible. This will be clarified in Appendix A.

Usually after this stage the equations of motion, (2.3b) and (2.4), are discretized. The latter, elliptic, equation obviously requires boundary conditions for $p$. These are not immediately available, since in (2.2) only the velocity appears. It is possible to derive boundary conditions for the pressure from the momentum equation (2.1b)—usually the normal component is used—but their evaluation requires values of the velocity components in points located one full mesh outside the boundary $\Gamma$. These values are not available. Various methods have been proposed as a remedy. This has led to confusion, and some controversy has arisen, as referred to above. A more complete discussion of this point is given by Gresho and Sani [1] and Peyret and Taylor [2, Chap. 6].

3. Solution. The above problem can be circumvented by first discretizing the original equations (2.1). In these discrete equations the boundary condition (2.2) is substituted. Only hereafter we will perform in a discrete sense the above reformulation. This leads to a discrete version of (2.4), but with a modification near the boundary, such that no boundary conditions are required. This process will be worked out in more detail for a discretization using the well-known staggered grid from the MAC-method [4]. The time-integration will be performed with an explicit two-level scheme,
but the discussion below applies to any time-integration method. The discrete time-evolution can be written as

\begin{align}
\text{(3.1a)} & \quad \text{div} \, q^{n+1} = 0, \\
\text{(3.1b)} & \quad \frac{q^{n+1} - q^n}{\delta t} + \text{grad} \, p^{n+1} = R^n
\end{align}

where \( n \) indicates the time level. The term \( \text{grad} \, p \) is written with an index \( n+1 \) to stress that its value has to be such that \( \text{div} \, q^{n+1} = 0 \); [2, Chap. 6] uses the same convention. Equation (3.1b) can be reformulated as

\[ q^{n+1} = q^n + \delta t R^n - \delta t \text{grad} \, p^{n+1}. \]

At this moment we do \textit{not} substitute (3.2) into (3.1a) to create the Poisson equation. Instead, we discretize first. Let the equations in discrete form be given by

\begin{align}
\text{(3.3a)} & \quad D_h q_h^{n+1} = 0, \\
\text{(3.3b)} & \quad \frac{q_h^{n+1} - q_h^n}{\delta t} + G_h p_h^{n+1} = R_h^n
\end{align}

where \( D_h \) and \( G_h \) are the discrete div and grad operator, respectively. Further \( q_h, p_h, \) and \( R_h \) are the discrete grid functions corresponding with \( q, p, \) and \( R \). Equations (3.3) are essentially the equations that are being solved. The way in which they are solved only uses some "shuffling" of these equations.

The treatment of the continuity equation in a cell adjacent to the boundary is the only thing that matters. Consider the cell given in Fig. 1. The discrete continuity equation (3.3a) reads

\[ \frac{1}{\delta x} (u_e^{n+1} - u_w^{n+1}) + \frac{1}{\delta y} (v_n^{n+1} - v_s^{n+1}) = 0. \]

Next the boundary condition (2.2) is applied, which states that \( u_e^{n+1} = u_e^r \). Only thereafter is the discrete version of (3.2) substituted. We end up with an equation for

\[ \text{FIG. 1. Stencil for (modified) Poisson equation near a boundary.} \]
the pressure, in which the pressure defined in the grid cell \( E \), lying outside the domain, does not occur. For completeness, the pressure equation reads

\[
\left\{ \begin{array}{l}
\frac{1}{(\delta x)^2} \left( \frac{1}{(\delta y)^2} \right) p_{E}^{n+1} + \frac{1}{(\delta x)^2} p_{N}^{n+1} + \frac{1}{(\delta y)^2} \left( p_{N}^{n+1} + p_{S}^{n+1} \right) \\
= \frac{1}{\delta t} \left[ \frac{1}{\delta x} \left( u_{x}^{n} - u_{x}^{n-1} \right) + \frac{1}{\delta y} \left( v_{y}^{n} - v_{y}^{n-1} \right) \right] + \left\{ \frac{1}{\delta x} R_{N}^{n} + \frac{1}{\delta y} \left( R_{N}^{n} - R_{S}^{n} \right) \right\}.
\end{array} \right.
\]  

(3.5)

In a more compact notation, the above can be formulated as follows. Split the discrete divergence operator as defined in (3.3a) in two parts

\[
D_{h} = D_{h}^{0} + D_{h}^{r}
\]

(3.6a)

where \( D_{h}^{0} \) corresponds with velocity components defined in interior points, and \( D_{h}^{r} \) corresponds with velocity components defined on the boundary \( \Gamma \). Then (3.3a) can be written as

\[
D_{h}^{0} q_{h}^{n+1} = -D_{h}^{r} q_{h}^{n+1}.
\]

(3.6b)

The right-hand side is known from the boundary condition (2.2). Now substitute the discrete version of (3.2), i.e., (3.3b):

\[
q_{h}^{n+1} = q_{h}^{n} + \delta t R_{h}^{n} - \delta t G_{h} p_{h}^{n+1},
\]

into (3.5). The result is

\[
D_{h}^{0} G_{h} p_{h}^{n+1} = D_{h}^{0} \left( \frac{1}{\delta t} q_{h}^{n} + R_{h}^{n} \right) + \frac{1}{\delta t} D_{h}^{r} q_{h}^{n+1}.
\]

(3.7)

We have a system of \( N_{x} \times N_{y} \) equations (where \( N_{x} \) and \( N_{y} \) are the number of cells in \( x \)- and \( y \)-direction, respectively) for an equal number of unknowns \( p \). It can be solved straightforwardly.

**Remark 1.** The system (3.7) is singular, since a constant pressure satisfies \( G_{h} p = 0 \). However, due to the staggered grid, \( p = Ct \) is the only solution of the homogeneous system. The right-hand side of (3.7) has to satisfy a compatibility relation: it must be perpendicular to the nullspace of \( (D_{h}^{0} G_{h})^{T} \). Here, this amounts to

\[
\sum_{i=1}^{N_{x}} \sum_{j=1}^{N_{y}} r_{ij} = 0,
\]

(3.8)

where \( r_{ij} \) is an abbreviation for the right-hand side of (3.7) in the cell \((i, j)\). It is easily verified that

\[
\sum_{i=1}^{N_{x}} \sum_{j=1}^{N_{y}} (D_{h}^{0} \phi)_{ij} = 0
\]

for any \( \phi \). Hence (3.8) reduces to

\[
\frac{1}{\delta y} \sum_{i=1}^{N_{x}} (v_{i,N_{y}+1/2} - v_{i,1/2}) + \frac{1}{\delta x} \sum_{j=1}^{N_{y}} (u_{N_{x}+1/2,j} - u_{1/2,j}) = 0.
\]

(3.9)

After multiplication with \( \delta x \delta y \), (3.9) equals a discrete version of

\[
\int_{\Gamma} \mathbf{q} \cdot \mathbf{n} \, ds = 0,
\]

which is a relation that must hold analytically. Hence any reasonable choice of the discrete boundary condition (2.2) will satisfy the compatibility condition.
**Remark 2.** Frequently, the full Poisson equation

\[ D_h G_h p^{n+1} = D_h \left( \frac{1}{\delta t} q_h^n + R_h^n \right) \]  

is solved with the Neumann boundary condition

\[ n \cdot G_h p^{n+1} = n \cdot \left( R_h^n - \frac{1}{\delta t} (q_h^{n+1} - q_h^n) \right) \]  

on \( \Gamma \).

The problem is that \( R_h^n \) cannot be computed since it requires a velocity component in a point one mesh outside the domain. This gave rise to approximations such as \( R_h^n = 0 \) (hence \( \partial / \partial n \) \( p = 0 \)), resulting in ambiguities and confusion. However, when the terms \( R_h^n \) that appear in (3.10) and in (3.11) are treated consistently, then the ambiguity cancels (see [2 Chap. 6.3]). Thus (3.5) can also be considered as obtained from (3.10) in which (3.11) is substituted.

**Remark 3.** The above technique works equally well for other time-integration methods. To see this consider the semidiscretized version of (2.3), i.e.,

\[ D_h q_h = 0, \quad \frac{d}{dt} q_h + G_h p_h = R_h. \]

By proceeding as in (3.6a), the pressure follows from

\[ D_h^0 G_h p_h = D_h^0 R_h + D_h^\Gamma \frac{\partial}{\partial t} q_h. \]

(To prevent error accumulation a term \( (1/\delta t) D_h q_h^n \) should be added to the right-hand side of (3.12).) No matter which time-integration method is used, the pressure can be computed from (3.12) without needing boundary conditions. When an ADI method is used, § 4 shows how to deal with the boundary velocities required by \( R_h \) at intermediate time levels.

**Remark 4.** In (3.3) the discrete divergence \( D_h \) and discrete gradient \( G_h \) are not yet specified. Higher-order discretizations are allowed, as long as the total number of equations in (3.3) plus the velocity boundary conditions equals the total number of unknowns \( q_h \) and \( p_h \).

4. Other applications. The philosophy presented above can be described as follows:

   Step (1). Discretize the equations of motion in their original (velocity-pressure) formulation.

   Step (2). Substitute the boundary conditions.

   Step (3). Combine the discrete equations into the desired form.

The preceding section shows how this philosophy can be applied to prevent the need for a boundary condition for the pressure in incompressible flow computations. There are more situations where this philosophy can be applied. We will briefly describe a few of them:

- The \( \psi-\omega \) formulation in incompressible flow;
- The shallow-water equations;
- ADI methods.

In Appendix A another, generalized, application is presented:

- Differential equations with algebraic constraints.
\textit{\psi-\omega} formulation.

\begin{align}
\nu \Delta \omega &= -\frac{\partial (\psi, \omega)}{\partial (x, y)}, \\
\Delta \psi &= -\omega,
\end{align}

with homogeneous boundary conditions

\begin{equation}
\psi = \frac{\partial \psi}{\partial n} = 0 \quad \text{on } \Gamma.
\end{equation}

Thus we have two boundary conditions for the streamfunction \( \psi \), whereas one for the vorticity \( \omega \) is "missing." Usually, the Dirichlet condition in (4.2) is added to (4.1b), while the Neumann condition is manipulated into a condition for \( \omega \) which is added to (4.1a). Many ways exist to do this; (see [2, Chap. 6.5]).

As an alternative, the above philosophy can be applied.

Step (1) means that we should start with a velocity-pressure formulation that is discretized, e.g., the steady version of (3.3) from the previous section. Then a discrete streamfunction \( \psi_h \) and vorticity \( \omega_h \) are defined by

\begin{align}
\frac{1}{\Delta y} \left[ (\psi_h)_{i+1/2,j+1/2} - (\psi_h)_{i+1/2,j-1/2} \right] &= u_{i+1/2,j}, \\
\frac{1}{\Delta x} \left[ (\psi_h)_{i+1/2,j+1/2} - (\psi_h)_{i-1/2,j+1/2} \right] &= -v_{i,j+1/2},
\end{align}

\begin{equation}
(\omega_h)_{i+1/2,j+1/2} = \frac{1}{\Delta x} \left[ v_{i+1,j+1/2} - v_{h,j+1/2} \right] - \frac{1}{\Delta y} \left[ u_{i+1/2,j+1} - u_{i+1/2,j} \right].
\end{equation}

Note that \( \psi_h \) and \( \omega_h \) are located in the vertices of the grid cells: their familiar location. The discrete continuity equation (3.3a) is identically satisfied by the choice (4.3). Also we have \( \psi_h = 0 \) on \( \Gamma \). Furthermore, it is easily verified that

\begin{equation}
\Delta_h \psi_h = -\omega_h
\end{equation}

where \( \Delta_h \) is the usual five-point formula. Thus far, there is nothing new.

Step (2) implies substituting the boundary conditions for the velocity \( q_h \) into (3.3b).

Step (3) means that we should take the discrete rotation (curl) of the discrete momentum equation (3.3b). Hence we perform

\begin{equation}
\frac{1}{\Delta x} \left[ y\text{-equation at } \left( i+1, \frac{j+1}{2} \right) - y\text{-equation at } \left( i, \frac{j+1}{2} \right) \right] \\
- \frac{1}{\Delta y} \left[ x\text{-equation at } \left( \frac{i+1}{2}, j+1 \right) - x\text{-equation at } \left( i+\frac{1}{2}, j \right) \right].
\end{equation}

This need only be done in the interior vertices. Substituting (4.3) into (4.4), we obtain a discrete version of (4.1a). The resulting discrete Laplacian becomes the usual five-point formula; the form of the discrete convective terms depends on the discretization performed in (3.3). Important is that no boundary values of \( \omega \) at \( \Gamma \) are required any more.

\textit{Remark 1.} This approach does not lead to discretizations that were unknown thus far. We leave it to the reader to verify that the resulting equations can also be obtained when the vorticity boundary condition is chosen according to Thom's formula [2, eq. (6.5.10)]. The latter is usually derived from a Taylor expansion using (4.1b) at the boundary.
Remark 2. Equation (4.6) is an algebraic combination of the discrete equations (3.3). Thus, when the solution of (4.5) and (4.6) is expressed in u and v (using (4.3)) it is identical to the steady solution of (3.3).

Shallow-water equations. Another application of the above philosophy is formed by the shallow-water equations. In a linearized primitive-variable form they read

\[
\begin{align*}
\frac{\partial \mathbf{q}}{\partial t} + (\mathbf{q} \cdot \text{grad}) \mathbf{q} + g \text{ grad } \zeta &= 0, \\
\frac{\partial \zeta}{\partial t} + (\mathbf{q} \cdot \text{grad}) \zeta + H \text{ div } \mathbf{q} &= 0.
\end{align*}
\]

Here \( \mathbf{q} \) is the depth-integrated velocity vector, \( \zeta \) is the surface elevation, \( H \) is the linearized water height, and \( g \) is the gravitational acceleration. Boundary conditions are often formulated in terms of the velocity components; no condition for \( \zeta \) exists then.

These equations are discretized in a staggered arrangement, as in the MAC-method. The elevation \( \zeta \) is defined in cell centers, as is the pressure \( p \). There is more similarity between \( \zeta \) and \( p \): the momentum equation (4.7) contains grad \( \zeta \), while the continuity equation (4.8) contains div \( \mathbf{q} \). As above, a Poisson-type equation can be derived. It is an unsteady wave equation that reads

\[
\frac{\partial^2 \zeta}{\partial t^2} - gH \text{ div grad } \zeta = \text{RHS}
\]

where RHS contains all convective terms.

When this equation is used in the computation, a boundary condition for \( \zeta \) is required. This one is "missing" however, but the above philosophy can be used to circumvent the problems. Due to the similarity between \( p \) and \( \zeta \), this proceeds in a way similar to that described in § 3.

ADI-methods. Consider a semidiscretized equation

\[
\frac{d \phi}{dt} = D \phi = (D_x + D_y) \phi
\]

where the spatial differential operator \( D \) can be split into one in \( x \)-direction (\( D_x \)) and one in \( y \)-direction (\( D_y \)). When (4.9) is solved by an ADI-method, or more generally a splitting-up (fractional step) method [5], conditions are required for the boundary values of \( \phi \) at intermediate time levels. Consider, for instance, the Peaceman–Rachford method (in Douglas–Gunn notation) in two dimensions [2, Chap. 2.8]:

\[
(I - \frac{1}{2} \delta t D_x)(\phi^* - \phi^n) = \delta t D \phi^n, \quad (I - \frac{1}{2} \delta t D_y)(\phi^{n+1} - \phi^n) = \phi^* - \phi^n.
\]

The term \( D_x \phi^* \) will, in general, require values of \( \phi^* \) on the boundary that are not immediately available. When the boundary conditions for \( \phi \) are independent of \( t \) there is no problem, as the above splitting is time-consistent. But in other cases the boundary values of \( \phi^* \) have to follow from nontrivial computations (see, e.g., Mitchell and Griffiths [6]).

The above philosophy suggests first to substitute the boundary conditions in (4.9). Hereafter the splitting is performed. This will lead to slightly modified \( D_x \) and \( D_y \) for which no boundary values are required. Hence the problem of the missing boundary values for the intermediate time levels is solved. Again, this strategy is not new: we have recovered Marchuk’s approach for treating the intermediate time levels [5].
Marchuk writes his motivation in words that exactly describe the philosophy presented in this paper. Concluding this section, we can do no better than cite him [5, p. 410]:

... it is much simpler first to put the original problem of mathematical physics into correspondence with a system of difference equations (with respect to the spatial variables) and then to eliminate the boundary conditions using the difference analogs of the boundary conditions, the accuracy of which matches that of the difference equations. Having done this, we can next proceed by approximating the equations in time using the splitting-up method or another algorithm. This approach allows us to sidestep the compatibility problem for the boundary conditions...

5. Conclusion. The paper describes a philosophy that can be used when boundary conditions are "missing." It can be formulated in short as: discretize first, substitute next, and combine later. The philosophy is not new, and neither are the resulting methods. But apparently its power is not yet generally appreciated, as the discussions that pop up now and then in the literature reveal. Four applications of the philosophy have been presented. It is hoped that these will help to enlarge the acquaintance with this solution to the problem of the "missing" boundary conditions.

Appendix A. The treatment of the constraint \( \text{div} q = 0 \) requires some care in the time-integration method. Accumulation of errors is possible. In essence this is due to the difference that exists numerically between

(A1) \[ \text{div} q = 0 \]

and

(A2) \[ \frac{\partial}{\partial t} \text{div} q = 0 \]

with a homogeneous initial condition. We will point out this difference using a formulation in which only a time-discretization is used. It equally applies to the space-discretized version, but the latter features more complicated formula which only distract attention from the essential point.

We start with (2.4) and substitute the constraint (in this case (A2)). This results in

(A3) \[ \text{div} \text{grad} p = \text{div} R. \]

Next, the equations of motion are solved with a numerical time-integration method. When (3.1) is used, this gives the following time discretization for (A3)

(A4) \[ \text{div} \text{grad} p^{n+1} = \text{div} R^n. \]

Having solved this equation, the velocity at time level \( n+1 \) is obtained from (3.2), repeated here

(A5) \[ q^{n+1} = q^n + \delta t (R^n - \text{grad} p^{n+1}). \]

However, the solution of (A4) cannot be obtained exactly: machine accuracy can be reached at most, and often this equation is solved iteratively until (only) a few figures have converged. Suppose we solve it with an error \( e^{n+1} \), i.e.,

\[ \text{div} \text{grad} p^{n+1} = \text{div} R^n + e^{n+1}. \]

Substitution in (A5) leads to a \( q^{n+1} \) whose divergence satisfies

(A6) \[ \text{div} q^{n+1} = \text{div} q^n - \delta t e^{n+1}. \]
This is an approximation of the discrete version of (A2) that for the chosen time-integration method would read

$$\frac{1}{\delta t} (\text{div } q^{n+1} - \text{div } q^n) = 0.$$  

Hence (A2) is satisfied, in a discrete sense, with an error which approaches zero as $\delta t \to 0$.

The same is not true with respect to the original constraint (A1). Error amplification is possible. Suppose a systematic error $\varepsilon$ is made each timestep. Then at a fixed time $t = n \delta t$ we have

$$(A7) \quad \text{div } q^n = -n \delta t \varepsilon = -te.$$ 

Note that it makes no sense letting $\delta t \to 0$. Moreover, when $t \to \infty$, e.g., because we are interested in an equilibrium solution, then we even have $\text{div } q \to \infty$.

The situation changes when we start with the discretization of the equations of motion. This results in (3.1). A rearrangement of (3.1b) is given in (A5). Only now we demand that the constraint (3.1a) is satisfied, repeated here

$$(A8) \quad \text{div } q^{n+1} = 0.$$ 

This is a discrete version of the "original" constraint (A1). Combining this with the discrete equations of motion leads to

$$(A9) \quad \text{div } \text{grad } p^{n+1} = \frac{1}{\delta t} \text{div } q^n + \text{div } R^n.$$ 

This equation has to be compared with (A4). Now when an error $\varepsilon^{n+1}$ is made in solving the equation, after substitution in (A5) it results in

$$\text{div } q^{n+1} = -\delta t \varepsilon^{n+1}.$$ 

In contrast with (A6), here no accumulation of errors is possible. Furthermore, letting $\delta t \to 0$ results in $\text{div } q^n \to 0$.

Remarks. (1) The possibility of error accumulation in solving differential equations with algebraic constraints such as (2.1) was noted two decades ago by Hirt and Harlow [7], and by Gear [8]. The above simple demonstration hereof is not yet widely known.

(2) The situation described in this Appendix is yet another, generalized, application of the philosophy discussed in this paper: the term "boundary condition" has to be replaced by "constraint." The method that is prone to error accumulation combines the analytical equations first, leading to (A3); only thereafter is the time-discretization performed. The other, stable, method first discretizes the equations and combines them with the constraint later.

Acknowledgments. The author is indebted to Mr. J. J. I. M. van Kan of the Delft University of Technology for pointing out the application of the described philosophy to the $\psi$-\omega formulation in § 4, and to Dr. P. Wilders of the Delft University of Technology for pointing out the shallow-water equations application in § 4.

REFERENCES

"MISSING" BOUNDARY CONDITIONS


