JOURNAL OF COMPUTATIONAL PHYSICS **137,** 212–244 (1997)
ARTICLE NO. CP975806

Analysis and Computation with Stratified Fluid Models

Richard Liska* and Burton Wendroff†

**Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Břehová 7, 115 19 Prague 1, Czech Republic; †Los Alamos National Laboratory, Los Alamos, New Mexico*

E-mail: liska@siduri.fjfi.cvut.cz; bbw@lanl.gov

Received August 1, 1996; revised May 30, 1997

Vertical averaging of the three-dimensional incompressible Euler equations leads to several reduced dimension models of flow over topography, including the one-layer and two-layer classic shallow water equations, and the one-layer and two-layer nonhydrostatic Green–Naghdi equations. These equations are derived and their well-posedness is discussed. Several implicit and explicit finite difference approximations of both the shallow water and Green–Naghdi models are presented, but for Green–Naghdi these are obtained using automatic code generation software. Numerical results are given in both well-posed and ill-posed regimes and compared with computations obtained by others. © 1997 Academic Press

1. INTRODUCTION

Vertically averaged models of incompressible flow have an obvious computational advantage over the full three-dimensional Euler equations, provided that important features of the flow are retained. Single layer models such as the hyperbolic shallow water equations and the dispersive Green–Naghdi equations have been shown to be at least qualitatively correct in many situations [1, 2].

The shallow water equations are derived by assuming the flow is hydrostatic, the horizontal velocity is vertically independent, and the continuity equation holds in the weak sense over the vertical range which, together with the vertical boundary conditions, eliminates the vertical velocity.

Dispersive models can be obtained by making assumptions about the form of the variation of velocity with respect to the vertical variable z, typically that the velocity components are polynomials in z. With these assumptions the vertical velocity component can be expressed in terms of derivatives of the horizontal velocity components. Then the vertical momentum equation can be integrated vertically to obtain an expression for the pressure, this being an obvious extension of the classic procedure of using Bernoulli's Law to eliminate the pressure [3]. Then the horizontal momentum equation is integrated with respect to z. The final result is a system of third-order partial differential equations in two space variables and time. This was done for a single layer in [4], taking the horizontal velocity component to be independent of z and the vertical component linear in z. The resulting equations are a generalization of the Boussinesq equation. For a single layer with a flat bottom the equations were found much earlier using the same procedure in [5]. For a variable bottom a very elegant form was found by Bazdenkov, Morozov, and Pogutse (BMP) [6]; see also [7].

In these single layer models the density is assumed constant. It is natural to try to obtain extensions in which the density is a piecewise constant function of z, and then the velocity is a piecewise polynomial. Although the resulting multilayer equations are quite complicated, obtaining them is straightforward. In this way we obtain a generalization of the BMP form of the Green–Naghdi equations. Equations for two-layer shallow water are well known and easily derived [8, 9].

Inviscid incompressible flow in two layers with a velocity jump at the interface is an ill-posed initial value problem. More precisely, it is shown in [3, Section 232] that small oscillations of the interface about a state of steady motion of two fluids of unequal densities and unequal velocities are unstable at sufficiently high wave numbers, at least in the case that the fluids have infinite depth. That is to say, the amplitudes of such disturbances grow exponentially (both in time and in wave number). It should be no surprise then that our vertically averaged two-layer models are also basically ill-posed with respect to small disturbances. In fact, the two-layer Green–Naghdi equations are unconditionally ill-posed for both the case of a free and of a rigid top surface, while the two-layer shallow water equations are of mixed type, that is, there are regions of state space for which the equations are hyperbolic and regions for which they are ill-posed. Nevertheless, there is interest in multilayer flows, for example in the theoretical development of [10], in the experiments reported in [11], in the steady state results of [12], and in the time-dependent work of [13].

However, there is a severe computational difficulty in that the classic stability and convergence theory of Lax and Richtmyer [14] does not apply to ill-posed initial value problems. One should not expect convergence of finite difference schemes as the grid is refined. Nevertheless, some information can be obtained, as in the two-phase flow example in [4]. Some ideas about computation in this case are presented in [15], but a precise theory seems not to be available. Also, it is shown in [16] that the addition of a fourth-order dissipation term regularizes the problem.

Given that there is no theory about just what can be learned from ill-posed models and, also, given that one does not want to simply give up and say such models are useless, we present here selected examples from extensive computations with several finite difference methods, explicit and implicit for shallow water and implicit for Green–Naghdi. The shallow water schemes appear to be new; they are formed by global composition of Lax–Friedrichs, Lax–Wendroff, and Crank–Nicolson methods. As a minimal attempt at validation of these shallow water methods for two layers we show several single layer computations.

In order to obtain numerical results for both the single and double layer Green–Naghdi models we rely heavily on automatic code generation.

2. MULTILAYER MODELS

Here we consider an n-layer model, the layers being indexed by i, with the density in each layer taken as constant ρ_i. The height of the bottom is

$$h_0(x),$$

while the *thickness* of the ith layer is

$$h_i(x, t).$$

The index $i = 1$ refers to the bottom layer, $i = n$ to the top layer.

Setting

$$z_0 = h_0,$$

the interface between the layers is at

$$z_i = z_{i-1} + h_i.$$

The horizontal velocity $u_i(x, t)$ of the ith layer is assumed to be constant in each layer. Vertical velocity is eliminated by vertical averaging.

2.1. *Multilayer Green–Naghdi Equations*

The multilayer generalization of Green–Naghdi equations is derived in Appendix A. The multilayer system includes the mass conservation equations

$$d_i h_i + h_i u_{ix} = 0, \quad i = 1, \ldots, n, \tag{1}$$

where d_i is the total derivation

$$d_i f = f_t + u_i f_x, \tag{2}$$

and the momentum equations, which we write here with the fourth-order dissipation,

$$(z_i - z_{i-1})\, d_i u_i + (z_i - z_{i-1}) g z_{ix} - A_{ix} - B_i z_{(i-1)x} = -(z_i - z_{i-1}) \left(\frac{\mathscr{P}_{ix}}{\rho_i} + \nu u_{ixxxx} \right), \tag{3}$$

where

$$A_i = (z_i - z_{i-1})^2\, d_i[\tfrac{1}{3}(z_i - z_{i-1}) u_{ix} - \tfrac{1}{2}\, d_i z_{i-1}], \tag{4}$$

$$B_i = (z_i - z_{i-1})\, d_i[\tfrac{1}{2}(z_i - z_{i-1}) u_{ix} - d_i z_{i-1}], \tag{5}$$

and $\nu \geq 0$ is the dissipation coefficient. If not explicitly stated otherwise we consider in the following the system without dissipation; i.e., $\nu = 0$.

The interfacial pressures strongly couple the equations together. For example, if the top surface is free, $\mathscr{P}_n = 0$, and

$$\mathscr{P}_i + \rho_i[q_i(z_i) - q_i(z_{i-1}) + g(z_i - z_{i-1})] = \mathscr{P}_{i-1},$$

or

$$\mathscr{P}_i = \sum_{j=i+1}^{n} \rho_j[q_j(z_j) - q_j(z_{j-1}) + g(z_j - z_{j-1})], \tag{6}$$

where $q_i(z)$ is given by (31) in Appendix A. If the top surface is held fixed we can no longer explicitly eliminate the interfacial pressures.

Note that the second time derivatives which appear in (4) and (5) can be eliminated by using (1), which we will do for the finite difference equations.

2.2. *Single Layer Green–Naghdi*

The single layer momentum equations are, after dropping subscripts,

$$(z_1 - z_0)\,du + (z_1 - z_0)gz_{1x} - A_x - Bz_{0x} = 0, \tag{7}$$

where

$$A = (z_1 - z_0)^2\,d[\tfrac{1}{3}(z_1 - z_0)u_x - \tfrac{1}{2}\,dz_0]$$

and

$$B = (z_1 - z_0)\,d[\tfrac{1}{2}(z_1 - z_0)u_x - dz_0].$$

2.3. *Multilayer Shallow Water*

Multilayer shallow water is obtained by setting the A's, B's, and q's equal to zero in (3) and (6) above, so that the equations are

$$d_iu_i + gz_{ix} = -\frac{1}{\rho_i}\mathscr{P}_{ix},$$

together with

$$d_ih_i + h_iu_{ix} = 0, \quad i = 1, ..., n.$$

If the top surface is free, then

$$\mathscr{P}_i = \sum_{j=i+1}^{n} \rho_j q(z_j - z_{j-1}).$$

As above, if the top surface is held fixed we can no longer explicitly eliminate the interfacial pressures.

There is another form for these equations in terms of momenta, which we show for $n = 2$, and a free surface. These are

$$(h_1)_t + (h_1u_1)_x = 0, \tag{8}$$

$$(h_2)_t + (h_2u_2)_x = 0, \tag{9}$$

$$(h_1u_1)_t + (h_1u_1^2 + \tfrac{1}{2}gh_1^2)_x + \frac{\rho_2}{\rho_1}gh_1h_{2x} + gh_1z_{0x} = 0, \tag{10}$$

and

$$(h_2u_2)_t + (h_2u_2^2 + \tfrac{1}{2}gh_2^2)_x + gh_2h_{1x} + gh_2z_{0x} = 0. \tag{11}$$

The total momentum then satisfies

$$\begin{aligned}(\rho_1h_1u_1 + \rho_2h_2u_2)_t &+ (\rho_1h_1u_1^2 + \rho_2h_2u_2^2 + \tfrac{1}{2}g\rho_2h_2^2 + \tfrac{1}{2}g\rho_1h_1^2 + g\rho_2h_1h_2)_x \\ &+ g(h_1 + h_2)z_{0x} = 0.\end{aligned} \tag{12}$$

2.4. *Passive Layer Solution*

Having a solution (Z_1, u) of a single layer (shallow water or Green–Naghdi) system we ask if this might also be a solution (z_1, z_2, u_1, u_2) of corresponding two-layer system with $\rho_1 = \rho_2$ so that the upper surface and velocities are the same $z_2 = Z_1$, $u_1 = u_2 = u$. In other words does there exist the interface z_1 so that (z_1, Z_1, u, u) is the solution of the two-layer system.

For two-layer shallow water system we see that if $\rho_1 = \rho_2$, $u_1 = u_2 = u$, and $h_1 + h_2 = H$ then we get from (8), (9), and (12) the single-layer shallow water equations in inhomogeneous conservation form:

$$H_t + (Hu)_x = 0,$$

$$(Hu)_t + (Hu^2 + \tfrac{1}{2}gH^2)_x + gHz_{0x} = 0.$$

Note that even with a flat bottom the individual momentum equations are not in conservation form. Only the thicknesses and the total momentum are conserved. There is an alternative formulation in conservation form, namely (12) and

$$(u_2 - u_1)_t + \tfrac{1}{2}(u_2^2 - u_1^2)_x + g\left(1 - \frac{\rho_2}{\rho_1}\right)h_{2x} = 0. \tag{13}$$

However, just because this happens to be in conservation form is no guarantee that the implied shock jump conditions are physically correct. This, of course, is a very well known situation even for completely hyperbolic systems such as the compressible Euler equations in one dimension. In that case both energy and entropy are conserved for smooth solutions, but physics disallows entropy conservation in shocks.

It follows from the above that if $\rho_1 = \rho_2$ initially equal velocities will remain equal, as long as there is a unique smooth solution.

For a two-layer Green–Naghdi system the situation is, however, different. If we express from the momentum equations (3) with flat bottom, i.e. $z_{0x} = 0$, the equation for $(u_1 - u_2)_t$, in which we substitute $\rho_1 = \rho_2$, $u_1 = u_2 = u$ and use mass equations (1) for simplification, we obtain

$$u_{txx} + u_{xxx}u - u_{xx}u_x = 0. \tag{14}$$

Substituting the single-layer Green–Naghdi soliton solution (32) [5], described in Appendix B, into this equation we get a nonzero function. Thus the passive layer solution of the two-layer Green–Naghdi with $\rho_1 = \rho_2$, $u_1 = u_2 = u$ coinciding for the upper surface and velocity with the solution of single-layer Green–Naghdi, as it is for the shallow water equations, generally does not exist.

3. DISPERSION RELATIONS

The calculation of the dispersion relations for these systems and others is presented in detail in [16], but for completeness we repeat the basic idea here. All the models are systems of partial differential equations in the (x, t) space. After linearization and Fourier transformation with Fourier variables (ω, k) corresponding to (x, t), the characteristic polynomial of the linearized system $P(\lambda, k)$, where $\lambda = \omega/k$, is calculated.

As we are interested only in the limit $|k| \to \infty$, which allows us to determine if the system is ill-posed, in each coefficient of λ^j we keep only the principal part, that is, the terms with the highest degree in k, and get a new polynomial $P_1(\lambda, k)$. In all cases k can be factored out of the equation $P_1(\lambda, k) = 0$ so that the roots λ_j of this equation do not depend on k. In such a case the original system is (linearly) ill-posed iff there exists a root λ_j with nonzero imaginary part. Note that we cannot say that the problem is well-posed if all roots λ_j are real, not even the linearized problem, since that is not a sufficient condition that the differential operator generate a semi-group. While it seems very likely that the nonlinear problem is ill-posed if the principal part of the linearized constant coefficient problem is ill-posed, we do not know of any theorem to this effect.

All the steps of the outlined dispersion analysis have been implemented in the computer algebra system Reduce [17], including the complete expansion of the condensed form of the differential equations.

3.1. *Shallow Water, Free Surface*

The variable k factors out of the characteristic polynomial $P(\lambda)$, leaving

$$P_1(\lambda) = (\lambda + u_1)^2(\lambda + u_2)^2 - gh_1(\lambda + u_2)^2 - gh_2(\lambda + u_1)^2 + g^2h_1h_2(1 - \rho_2/\rho_1). \tag{15}$$

Without loss of generality we can assume that $u_1 = 0$. Further, we make substitutions $H_i = gh_i, u = u_2, R = \rho_2/\rho_1$. For the case $u = 0$ the odd powers of λ disappear and we have the polynomial

$$P_1(\lambda) = \lambda^4 - \lambda^2(H_1 + H_2) + H_1H_2(1 - R).$$

This is quadratic in λ^2 with roots

$$2\lambda^2 = H_1 + H_2 \pm \sqrt{(H_1 + H_2)^2 - 4H_1H_2(1 - R)}.$$

Then for $R < 1$ there are four distinct real λ. Thus for $R < 1$ there exists $\varepsilon > 0$ such that for $|u_2 - u_1| < \varepsilon$ the problem is well-posed. For $R > 1$ and $u_1 = u_2$ the problem is ill-posed.

The special case $R = 1$ can be treated analytically. In that case

$$0 = 1 - \frac{H_1}{(\lambda + u_1)^2} - \frac{H_2}{(\lambda + u_2)^2}. \tag{16}$$

Differentiating with respect to λ we find that

$$\frac{\lambda + u_1}{\lambda + u_2} = -\left(\frac{H_1}{H_2}\right)^{1/3},$$

and substituting this into (16) gives

$$(u_1 - u_2)^2 = (H_1^{1/3} + H_2^{1/3})^3$$

and the problem is then ill-posed if

$$0 < (u_1 - u_2)^2 < (H_1^{1/3} + H_2^{1/3})^3. \tag{17}$$

The case $H_1 = H_2 = H$ can also be done analytically since then the characteristic polynomial reduces to a quadratic, as observed in [8]. Namely, if

$$2[H(1 - \sqrt{R})]^{1/2} < |u_1 - u_2| < 2[H(1 + \sqrt{R})]^{1/2}$$

the problem is ill-posed.

Using the quantifier elimination method of [18] one can find the condition on the coefficients of the polynomial (15) which is equivalent to the statement that all roots of this polynomial are real. The condition is too long to be presented here; however, we use it in numerical codes for checking ill-posedness.

3.2. *Shallow Water, Rigid Lid*

The variable k factors out of the characteristic polynomial $P(\lambda)$ leaving a polynomial quadratic in λ, which has the discriminant

$$D = -4h_1h_2\left[(u_1 - u_2)^2 - g(\rho_1 - \rho_2)\left(\frac{h_1}{\rho_1} + \frac{h_2}{\rho_2}\right)\right].$$

For the problem to be ill-posed the discriminant has to be negative, which is the case if

$$(u_1 - u_2)^2 > g(\rho_1 - \rho_2)\left(\frac{h_1}{\rho_1} + \frac{h_2}{\rho_2}\right),$$

as found in [19].

3.3. *Green–Naghdi, Rigid Lid*

The dispersion analysis, after factoring out k, gives a polynomial $P_1(\lambda)$ which is quadratic in λ. The discriminant of this polynomial is

$$D = -4(u_1 - u_2)^2 h_1 h_2 \rho_1 \rho_2,$$

the same (except for a positive factor) as in Case 3, Section 3.4. D is negative so that the problem is again unconditionally ill-posed, except in trivial cases.

In [16] it is shown that the two-layer Green–Naghdi rigid lid model with dissipation coefficient $\nu > 0$ is well-posed. Note, however, that the dissipation in this case does not imply decaying Fourier components, but only the bounded ones, so the dissipation regularizes this system boundedly.

3.4. *Green–Naghdi, Free Surface*

The variable k factors out of the characteristic polynomial $P(\lambda)$ leaving

$$P_1(\lambda) = (\lambda + u_2)^2 Q(\lambda),$$

where $Q(\lambda)$ is quadratic in λ and its discriminant is

$$D = -48(u_1 - u_2)^2 h_1 h_2 \rho_1 \rho_2.$$

As D is unconditionally negative, $P(\lambda)$ has roots with a nonzero imaginary part and the problem is unconditionally ill-posed, except in trivial cases.

As the fourth-order dissipation regularizes the rigid lid model, one might expect similar behavior also in the free surface case which is also a third-order system. The first numerical computations have shown that the dissipation regularizes also this model (see Section 5.7.2).

4. FINITE DIFFERENCE EQUATIONS

Our goal is to study finite difference methods for the ill-posed two-layer Green–Naghdi equations and the mixed-type two-layer shallow water equations. In the former there is no reason to expect discontinuities to develop, while just the opposite is true in the latter. A further serious difficulty for shallow water is that, not only are the four two-layer shallow water equations not uniformly hyperbolic, they are not in momentum conservation form; that is, the individual layer thicknesses and total momentum are conserved, but the momenta of each layer are not. This means that there are three shock jump conditions we can have confidence in, but the fourth condition is not known. In fact one can doubt that there is a universally correct fourth condition. We will see an effect of this in Section 5.4.1. Some interesting ideas about this can be found in [20].

The effect of this nonconservation form is that we do not really know how to write difference equations, except that we should honor the three existing conservation laws. The problem is that of representing terms like $h_1 h_{2x}$. We will simply follow [8], using the arithmetic mean times the difference, which has the crucial property that the difference analogue of the derivative of a product holds. It is also true that the corresponding jump conditions are compatible with the single layer jump conditions when the densities are equal and the velocities on each side of the shock are equal.

Two-layer shallow water also differs from the usual hyperbolic system such as gas dynamics in that even in a hyperbolic region of state space the wave speeds, being the roots of a fourth degree polynomial, do not have a simple expression. Together with the possibility that the roots are complex, this eliminates from consideration those methods which use an eigenvector decomposition. This leaves us with explicit schemes such as Lax–Friedrichs and Lax–Wendroff or implicit schemes such as Crank–Nicolson. After considerable experimentation we have found that combinations of these along with anti-diffusion and a filter give the best results.

4.1. *First-Order Balance Laws*

For vector $\mathbf{v}$ with N components the system of partial differential equations for the multilayer shallow-water approximation has the form

$$\mathbf{v}_t + \mathbf{f}_x(\mathbf{v}) + \mathbf{G}(\mathbf{v})\mathbf{v}_x + \mathbf{g}(\mathbf{v}, x) = 0. \tag{18}$$

In the following the index n corresponds to the time variable t and the index j to the space variable x.

4.1.1. *Lax–Friedrichs with anti-diffusion.* We have chosen to use the two-step form of Lax–Friedrichs (LF) with anti-diffusion as proposed in [21].

$$\begin{aligned}
\mathbf{v}_{j+1/2}^{n+1/2} &= \frac{1}{2}\left(\mathbf{v}_j^n + {}_{j+1}^{n}\right) - \frac{\Delta t}{2\,\Delta x}\left[\mathbf{f}(\mathbf{v}_{j+1}^n) - \mathbf{f}(\mathbf{v}_j^n)\right] \\
&\quad - \frac{\Delta t}{2\,\Delta x}\mathbf{G}\left(\frac{1}{2}\left(\mathbf{v}_j^n + \mathbf{v}_{j+1}^n\right)\right)\left[\mathbf{v}_{j+1}^n - \mathbf{v}_j^n\right] - \frac{\Delta t}{2}\mathbf{g}\left(\frac{1}{2}\left(\mathbf{v}_j^n + \mathbf{v}_{j+1}^n\right), x_{j+1/2}\right) \\
\mathbf{v}_j^{n+1} &= \frac{1}{2}\left(\mathbf{v}_{j+1/2}^{n+1/2} + \mathbf{v}_{j-1/2}^{n+1/2}\right) - \frac{\Delta t}{2\,\Delta x}\left[\mathbf{f}(\mathbf{v}_{j+1/2}^{n+1/2}) - \mathbf{f}(\mathbf{v}_{j-1/2}^{n+1/2})\right] \\
&\quad - \frac{\Delta t}{2\,\Delta x}\mathbf{G}\left(\frac{1}{2}\left(\mathbf{v}_{j+1/2}^{n+1/2} + \mathbf{v}_{j-1/2}^{n+1/2}\right)\right)\left[\mathbf{v}_{j+1/2}^{n+1/2} - \mathbf{v}_{j-1/2}^{n+1/2}\right] - \frac{\Delta t}{2}\mathbf{g}\left(\frac{1}{2}\left(\mathbf{v}_{j+1/2}^{n+1/2} + \mathbf{v}_{j-1/2}^{n+1/2}\right), x_j\right).
\end{aligned}$$

The anti-diffusion correction which applies to all components v of $\mathbf{v}$,

$$v_j = v_j - \alpha_j + \alpha_{j-1},$$

is defined in terms of forward and backward differences,

$$d_j^+ = v_{j+1} - v_j, \quad d_j^- = v_j - v_{j-1}, \tag{19}$$

by

$$\alpha_j = \tfrac{1}{2} \max(0, \min(d_{j+1}^+ \operatorname{sign} d_j^+, |d_j^+|/2, d_{j-1}^+ \operatorname{sign} d_j^+)) \operatorname{sign} d_j^+ .$$

Note that the anti-diffusion correction is applied only in regions where the component v is monotone. It serves to cancel the diffusion introduced by the averaging in LF, but it is limited so as to maintain positivity in the scalar case. It is well known that this anti-diffusion counteracts the excessive spreading of shocks associated with LF and that it can distort smooth waves. We found that by applying the anti-diffusion every two to four time steps with a weight, instead of every step, we could maintain steep shocks and also significantly reduce the distortions. We indicate such a scheme as LFAD(W)n, meaning that the anti-diffusion is applied every nth time step with the weight W. For the same scheme with full anti-diffusion, i.e. with $W = 1$, we use the notation LFADn which is the same as LFAD(1)n.

4.1.2. *A composite Lax–Wendroff Lax–Friedrichs method.* The two-step (predictor–corrector) version of the Lax–Wendroff (LW) method without artificial viscosity for the system (18) is given by

$$\begin{aligned}
\mathbf{v}_{j+1/2}^{n+1/2} &= \frac{1}{2}(\mathbf{v}_j^n + {}_{j+1}^n) - \frac{\Delta t}{2\,\Delta x}[\mathbf{f}(\mathbf{v}_{j+1}^n) - \mathbf{f}(\mathbf{v}_j^n)] \\
&\quad - \frac{\Delta t}{2\,\Delta x}\mathbf{G}\left(\frac{1}{2}(\mathbf{v}_j^n + \mathbf{v}_{j+1}^n)\right)[\mathbf{v}_{j+1}^n - \mathbf{v}_j^n] - \frac{\Delta t}{2}\mathbf{g}\left(\frac{1}{2}(\mathbf{v}_j^n + \mathbf{v}_{j+1}^n), x_{j+1/2}\right) \\
\mathbf{v}_j^{n+1} &= \mathbf{v}_j^n - \frac{\Delta t}{\Delta x}[\mathbf{f}(\mathbf{v}_{j+1/2}^{n+1/2}) - \mathbf{f}(\mathbf{v}_{j-1/2}^{n+1/2})] \\
&\quad - \frac{\Delta t}{\Delta x}\mathbf{G}\left(\frac{1}{2}(\mathbf{v}_{j+1/2}^{n+1/2} + \mathbf{v}_{j-1/2}^{n+1/2})\right)[\mathbf{v}_{j+1/2}^{n+1/2} - \mathbf{v}_{j-1/2}^{n+1/2}] - \Delta t\,\mathbf{g}\left(\frac{1}{2}(\mathbf{v}_{j+1/2}^{n+1/2} + \mathbf{v}_{j-1/2}^{n+1/2}), x_j\right).
\end{aligned}$$

Note that the two-step LW and LF are quite similar. They have the same first half-step and they differ in the second correcting half-step. The LW scheme corrects from the previous time level n while the LF scheme corrects from the time half-level $n + \frac{1}{2}$.

To remove the inevitable oscillations appearing close to shocks a filter has been proposed in [22]. This filter should be applied to the characteristic variables. As indicated above, we must avoid an eigenvector expansion and so we had to apply the filter to the basic variables. We found that although it was very effective in removing oscillations there were also large errors left after the shocks had passed.

This led us to consider using Lax–Friedrichs as a form of filter by replacing every *n*th LW step with a LF step, indicated as LWLF*n*. Although the scheme is first-order accurate for any *n*, for *n* between two and four it resolves shocks much better than LF alone and does not have the oscillations of LW alone, the reason being, roughly, that the leading first-order error term is divided by *n*, and the diffusive LF amplification factor is replaced by its *n*th root. We have been also experimenting with combining LW and LF steps with anti-diffusion; however, this method is never used in the test cases presented in this paper.

4.1.3. *A composite Crank–Nicolson scheme.* After spatial discretization the time-centered Crank–Nicolson (CN) scheme has the form

$$\mathbf{v}_j^{n+1} = \mathbf{v}_j^n + \tfrac{1}{2}\Delta t\,[\mathbf{F}(\mathbf{v}^{n+1}) + \mathbf{F}(\mathbf{v}^n)],$$

where the vector $\mathbf{F}$ depends on $\mathbf{v}$ at j, $j+1$, and $j-1$.

This is solved by Newton's method,

$$\mathbf{v}_j^{(k)} = \mathbf{v}_j^n + \tfrac{1}{2}\Delta t\,[\mathbf{F}(\mathbf{v}^{(k-1)}) + \mathbf{J}(\mathbf{v}^{(k-1)})(\mathbf{v}^{(k)} - \mathbf{v}^{(k-1)}) + \mathbf{F}(\mathbf{v}^n)],$$

where $\mathbf{J}$ is the Jacobian matrix of $\mathbf{F}$ and $\mathbf{v}^{(0)} = \mathbf{v}^n$.

Here we found that a combination of CN steps, LF steps, and the filter in [22] was very effective. This is indicated as CNLF*n*F*m*, that is, every *n*th step CN is replaced by LF, and the filter is applied every *m*th step.

The algebra involved in the derivation of the linearized scheme is quite tedious and hand coding of Newton's method while not difficult is error-prone. So the discretization has been done by the computer algebra system Reduce [17] with the package FIDE [23] and the code for solving the linear implicit scheme has been automatically generated by the module Linband [24] which is designed for generation of code for solving linear systems with a band matrix. During the code generation the computer algebra program creates the numerical source code in Fortran which solves the given implicit scheme. For the one-layer shallow water model about 50% (i.e., about 8 kB or 250 lines) of the numerical code has been automatically generated and for the two-layer shallow water model about 67% (i.e., about 25 kB or 720 lines) of the code has been generated automatically. The remainder of the code involving such things as input–output operations or initial conditions has been done manually.

For completeness we describe the filter which is the simplest one from [22]. It is applied to every component v of $\mathbf{v}$ in each grid point in which v_j is a local extremum, that is when $d_j^+ d_j^- < 0$, where d_j^+, d_j^- are differences (19). So if $d_j^+ d_j^- < 0$ then v_j is corrected by

$$v_j = v_j + \delta_j \operatorname{sign} d_j^+,$$

where

$$\delta_j = \min(\min(|d_j^+|, |d_j^-|), \tfrac{1}{2}\max(|d_j^+|, |d_j^-|)).$$

Further, to retain conservation one of v_{j-1}, v_{j+1} must be corrected in the opposite sense according to

$$v_J = v_J - \delta_j \operatorname{sign} d_j^+ ,$$

where $J = j + 1$ for $|d_j^+| > |d_j^-|$ and $J = j - 1$ otherwise.

4.2. *Green–Naghdi*

4.2.1. *One-layer Green–Naghdi.* The single layer Green–Naghdi equations (1) (with $n = 1$) and (7) contain mixed time-space derivatives and are most naturally approximated by implicit schemes.

For the single layer Green–Naghdi we have used a backward Euler difference scheme resulting in a system of nonlinear difference equations which is then solved by Newton's method. We call this scheme backward Euler Newton (BEN) scheme. The scheme construction proceeds in the following way. First, all derivatives of products are expanded and all the time derivatives are replaced by

$$v_t \approx \frac{v^{n+1} - v^n}{\Delta t}. \tag{20}$$

All remaining terms, i.e. terms which are not inside a time derivative, are discretized at the implicit time level $n + 1$. So we are using a fully implicit discretization. The space derivatives are approximated by

$$\begin{aligned}
v_x &\approx \frac{-v_{j+2} + 8v_{j+1} - 8v_{j-1} + v_{j-2}}{12\,\Delta x}, \\
v_{xx} &\approx \frac{-v_{j+2} + 16v_{j+1} - 30v_j + 16v_{j-1} - v_{j-2}}{12\,\Delta^2 x}, \\
v_{xxx} &\approx \frac{v_{j+2} - 2v_{j+1} + 2v_{j-1} - v_{j-2}}{2\,\Delta^3 x}, \\
v_{xxxx} &\approx \frac{v_{j+2} - 4v_{j+1} + 6v_j - 4v_{j-1} + v_{j-2}}{\Delta^4 x},
\end{aligned} \tag{21}$$

which are fourth-order approximations for v_x, v_{xx} and a second-order approximation for v_{xxx}, v_{xxxx} (fourth-order derivative appears in the dissipation term of the two-layer Green–Naghdi (3)). Such discretization defines the nonlinear finite difference scheme. In most cases we found that a single iteration of Newton's method was sufficient.

The FIDE package, Reduce, and Linband were also used here. All the derivation, including derivation of the one-layer Green–Naghdi system of PDEs and their full discretization has been performed by a computer. The proposed space discretizations have been also obtained by a computer. The final difference scheme is checked for correctness by doing the truncation error analysis. Having derived the final difference scheme, the Fortran code calculating the band matrix and calling the numerical solution routine, which occupies more than 84% of the whole code, has been generated by a few lines (10 lines) of high level code. The full Fortran source code implementing this solution procedure occupies about 90 kB (about 1600 lines), most of which has been generated completely automatically. To improve the efficiency of the code and decrease the size of Fortran source code the finite differences (21) for all quantities at the explicit and intermediate levels have been stored in independent arrays and all the derivations of the difference scheme (after expanding derivatives of products in the original PDEs) and code generation have been done without expanding products of sums and/or differences.

4.2.2. *Two-layer Green–Naghdi.* The equations in (1), (3), using (4), (5), are expanded so that there are no derivatives of products. The second time derivative that appears, z_{1tt}, is eliminated using

$$z_{1tt} = -(u_1 h_1)_{xt}.$$

At first we tried the same approach, i.e. the backward Euler Newton scheme, as described in the previous section for single-layer Green–Naghdi, here with $\mathbf{w} = (z_1, z_2, u_1, u_2)$. However, this leads to an algorithm that produces over 550 kB (more than 8000 lines) of automatically generated code. And, of course, being so big, this code is too slow. The complexity of this algorithm is caused by the large Jacobian of the two-layer Green–Naghdi equations. So for this case precise Newton linearization is not practical and we have applied ad-hoc linearization where in the discretization of products only some factors (usually one) of each product has been kept on the implicit time level.

The first step, i.e. discretization of time derivatives, proceeds in the same manner as for BEN by using (20). Now for the other terms which do not include the time derivative we have to decide what will be taken explicit at time n and what will be taken implicit at time $n + 1$. In those terms with no time derivative, if there is a unique highest order spatial derivative, it is taken at time $n + 1$ and the other factors at time n. The remaining terms are done with some splitting, for example, a term of the form ab might be defined as $(a^n b^{n+1} + a^{n+1} b^n)/2$.

The space discretization is defined again as for BEN by (21); however, the ad-hoc linearization here guarantees that the resulting difference scheme is linear in the implicit $\mathbf{v}^{n+1}$ terms.

Again we have used the computer algebra system Reduce with the package FIDE and module Linband for discretization and code generation. For the ad-hoc linearization more than 90% of the code, i.e. about 130 kB or 2500 lines, has been generated automatically by computer. This code is about four times faster than the BEN code.

For experimenting with difference schemes for two-layer Green–Naghdi the usage of code generation facilities has been essential. It is hard to imagine that one writes and debugs several thousands of source code lines just for testing a particular difference scheme. One might think that hand coding would produce more efficient code; however, efficiency was not our aim and we also believe that with the current state of the art in compiler optimization the difference in speed between hand written and generated code would not be high. Note that almost the only task of automatically generated code is to calculate the matrix and right-hand side of a system of linear algebraic equations. Also the probability of bugs in generated code is much less than in hand written code.

5. NUMERICAL RESULTS

Before proceeding to two-layer, possibly ill-posed models, one needs to have robust numerical methods working well for similar well-posed one-layer models. So for both cases of shallow water and Green–Naghdi models we start the presentation with comparisons of our one-layer codes with earlier results of others.

5.1. *Initial and Boundary Conditions*

The initial conditions for n-layer shallow water and Green–Naghdi equations for the test problems treated in this section are for heights $z_i(x, t)$ and velocities $u_i(x, t)$ given by

$$z_i(x,0) = z_i^0, \quad u_i(x,0) = u_i^0, \quad i = 1, ..., n. \tag{22}$$

Note that the vertical averaging procedure remains valid only for positive thicknesses of each layer. So if we have a problem for which the bottom profile is higher than the lowest layer, i.e. $z_1^0 < b_c$, where b_c is the height of bottom bump (see, e.g., Fig. 5), then we have to set up the initial conditions so that the thickness of the lowest layer is greater than some small positive number; i.e., it is equal to this number in the area where $z_1^0 < z_0(x)$. The same applies also to other layers if their upper constant surface would be below the height of the bottom.

We solve these problems on the space interval $x \in (A, B)$, $A < 0$, $B > 0$. For all problems we use free boundary conditions at both ends of the interval,

$$z_{ix}(A,t) = 0, \quad u_{ix}(A,t) = 0, \quad z_{ix}(B,t) = 0, \quad u_{ix}(B,t) = 0, \quad i = 1, ..., n. \tag{23}$$

In the different problems we use three different bottom profiles. All profiles have a bump with maximum height b_c at $x = 0$. The profiles differ in the shape of the bump. For shallow water problems we use the profile from [1, 8],

$$z_0(x) = \begin{cases} b_c\left(1 - \dfrac{x^2}{a^2}\right) & \text{for } -a \le x \le a \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

For comparison of one-layer Green–Naghdi with the steady solution of [25] we use the profile

$$z_0(x) = b_c \begin{cases} \dfrac{1}{2}\left(1 + \cos\dfrac{\pi x}{2}\right) & \text{for } -2 \le x \le 2 \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

For comparison of the Green–Naghdi codes with the results of [2] we take as the profile

$$z_0(x) = \frac{b_c}{(1+x^2)^{3/2}}. \tag{26}$$

In all cases the normalization with the gravitational constant $g = 1$ is used. Since $g = 1$ and in most cases $z_n^0 = 1$, the velocity u_n^0 coincides with the Froude number $F_n = u_n^0/\sqrt{g z_n^0}$.

5.2. *Adaptive Time Step*

In all codes for one- and two-layer shallow water and Green–Naghdi models we have used the following procedure for adaptive time step control. Before each time step we have calculated the maximum "pseudo-eigenvalue"

$$\lambda_{\max} = \max_{i=1,\ldots,n, x\in(A,B)} \left(u_i(x) \pm \sqrt{h_i(x)}\right)$$

over n layers and the current time step is then given by

$$\Delta t = \frac{C_{\max}}{\lambda_{\max}} \Delta x,$$

where $C_{\max}$ is the maximum Courant–Friedrichs–Levy (CFL) number which is used as an input parameter.

This procedure is completely valid for the one-layer shallow water equations. For the two-layer problems and for Green–Naghdi we are assuming that the gravity wave speeds in each layer provide a reasonable limit on the time step.

5.3. *One-Layer Shallow Water*

As a preliminary to and a means of validating two-layer computations we have done many runs, including convergence tests, for different one-layer shallow water problems for different LFAD(W)n, LWLFn, CNLFnFm methods. In all shallow water problems we have used the bottom profile (24) as used in [1, 8] with $a = 2$.

Figure 1 shows the comparison of results of LFAD2, LWLF2, and CNLF1F8 methods (the numerical results of these three methods differ only negligibly) with analytical estimates from [1] for the problem with $b_c = 0.65$, $u_0 = 0.7$ (which is the case 1 of [2] which belongs to region IIb of [1]) at time $t = 20$ which has a shock running upstream, a shock running downstream, and a rarefaction wave running downstream. We have used 2000 points on the interval $x \in (-40, 40)$ with $C_{\max} = 0.7$ for LF and LW time steps and $C_{\max} = 2$ for the CN time steps. The plots show a good agreement of our three numerical methods with analytical estimates of the speeds and heights of the shocks and the rarefaction wave.

As from the estimates we know only heights and speeds and not positions, we have taken positions from numerical results at time $t = 10$ and from these positions

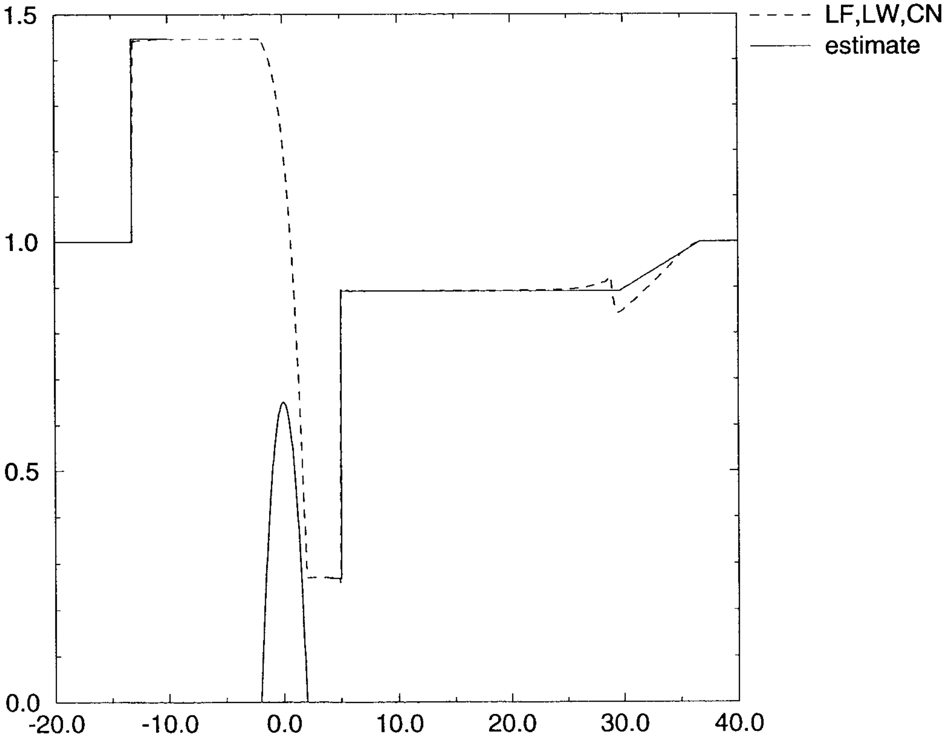

FIG. 1. Comparison of heights of LFAD2, LWLF2, and CNLF1F8 methods (gave same results) with analytical estimates for the problem with $b_c = 0.65$, $u_0 = 0.7$.

and velocity estimates we have calculated estimated positions at time $t = 20$ which are plotted in the figure. As concerns the downstream propagating rarefaction wave, we have analytical estimates for speeds of its right- and left-hand sides, but we do not know its profile. So in the plots we have included the flat profile before and after the rarefaction wave and a linear profile for the rarefaction wave. As the right-hand side of the rarefaction wave we have taken the point of local minimum from the numerical results.

5.4. *Two-Layer Shallow Water*

For two-layer shallow water problems we have used the same profile (24) with $a = 2$ as for one-layer problems.

5.4.1. *Comparison with one-layer shallow water with constant density.* First we compare the two-layer shallow water model with the same densities $\rho_1 = \rho_2$, i.e. $R = 1$, in both layers with the one-layer shallow water model. As has been shown in Section 2.3 if the velocities of both layers remain the same, $u_1 = u_2$, then the two-layer model has to produce the same results as the one-layer model. Note, however, that this two-layer shallow water model is ill-posed wherever there are small nonzero velocity differences, as has been shown in Section 3.1; see (17).

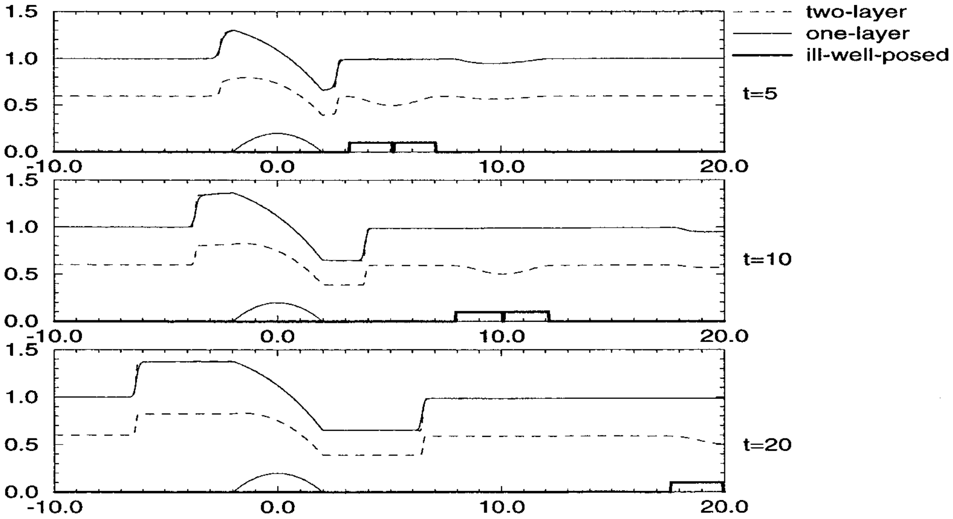

FIG. 2. Comparison of heights of two-layer (LFAD2, LWLF2, CNLF1F8 methods) and one-layer shallow water models for the problem with $b_c = 0.2$, $u_0 = 1$, $R = 1$ at times $t = 5, 10, 20$. The bold line distinguishes the well-posed (value 0) and ill-posed (value 0.1) regions.

We use here the initial conditions (22) with $n = 2$, the initial velocities $u_1^0 = u_2^0 = u_0$ and the initial heights $z_2^0 = 1$ and $z_1^0 = (1 + b_c)/2$.

In Fig. 2 we show the results of this comparison of two-layer and one-layer shallow water models for the problem with $b_c = 0.2$, $u_0 = 1$ (this corresponds to case 3 from [2]) at times $t = 5, 10, 20$. Both the two-layer and one-layer models were solved by the methods LFAD2, LWLF2, and CNLF1F8 which gave results with negligible differences. The figure shows good agreement for the upper surface obtained by the one-layer and two-layer models. In this calculation we have used 1000 cells on the interval $x \in (-20, 20)$. The CFL number $C_{\max}$ was again 0.7 for LF and LW time steps and 2 for CN time steps. In this problem there appear both ill-posed and well-posed regions. These regions are distinguished in the figure by the bold line which has the value 0 in the well-posed region and the value 0.1 in the ill-posed region. The ill-posed regions around the rarefaction wave on the layers interface reflect the areas where the velocities of upper and lower layers differ; for more explanation check the comments to Fig. 3 below or (17).

In Fig. 3 we show a problem for which we have not obtained agreement between the one-layer and two-layer models. The problem has the initial data $b_c = 0.65$, $u_0 = 0.7$ (this corresponds to case 1 from [2]) and the figure is at time $t = 20$. In this problem we have used 1000 points on the interval $x \in (-20, 20)$ with the CFL number $C_{\max}$ 0.35 for LFAD2 method, 0.2 for LWLF2 method, and 0.5 for CNLF1F8 method with 0.2 for LF substeps. All three methods have produced results which are very close and which are shown by dashed lines in the figure. There is a big disagreement between the upper surface z_2 of the two-layer model and the surface z_1 of the one-layer model on the downstream side. Actually the downstream shock of the one-layer model is bracketed by two downstream shocks of the two-layer model.

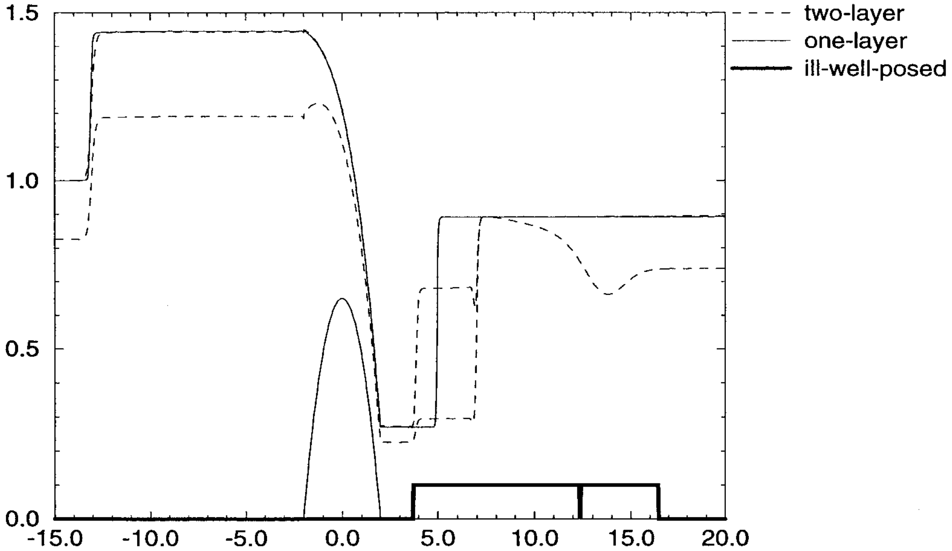

FIG. 3. Comparison of heights of two-layer and one-layer shallow water models for the problem with $b_c = 0.65$, $u_0 = 0.7$ at time $t = 20$ showing the disagreement of the two-layer model with the one-layer model. The bold line distinguishes the well-posed (value 0) and ill-posed (value 0.1) regions.

The reason for this is that while the differential equations guarantee that initially equal velocities will remain equal (if $R = 1$) if there is unique dependence on the data for smooth solutions, as shown in Section 2.3, the difference equations using the individual momenta as variables do not have this property. So we quickly get into a true two-layer ill-posed situation, with an evolution that does not agree with the single layer solution. The ill-posedness is caused by different velocities of both layers; compare (17). There are two intervals of ill-posedness, in the first one $u_1 > u_2$, in the second one $u_1 < u_2$ and $u_1 \approx u_2$ around $x \approx 12.5$.

Instead of using the two-layer model with (10)–(11) we tried the alternative formulation (12) and (13) (mass equations (8)–(9) remain the same). This model produced the single layer solution for very long times without any sign of instability. However, in a genuinely two-layer problem such as that in Fig. 4 below, a different, albeit similar, structure is obtained. As indicated earlier, we do not at the moment have any reason to choose one over the other, although surely some kind of viscosity argument will eventually settle the issue. We will stick to the individual layer momenta formulation for now. However, if one could be certain that no shocks will develop then the alternative formulation might be a better choice.

5.4.2. *Different density in each layer.* In this section we present a comparison with results of [8]. We use again the initial conditions (22), here with $n = 2$ and the initial heights $z_1^0 = 1$, $z_2^0 = 2$ and velocities $u_1^0 = u_2^0 = u_0$ as in [8]. The density ratio $R = \rho_2/\rho_1$ is 0.8 also as in [8].

Figure 4 shows the time evolution (at times $t = 15, 60$) for the problem with $b_c = 0.8$, $u_0 = 0.4$, which is the problem 15 from [8], calculated by the LFAD2, LWLF4, and CNLF1F8 methods. This problem is well-posed everywhere. The

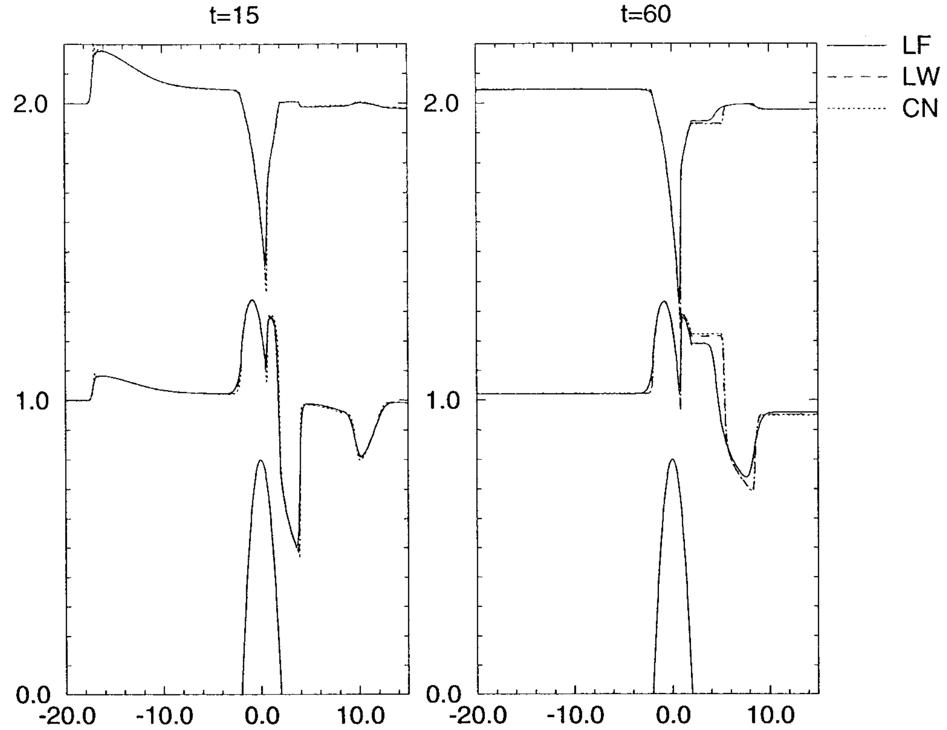

FIG. 4. Comparison of heights for two-layer shallow water model of LFAD2, LWLF4, and CNLF1F8 methods for the problem with $b_c = 0.8$, $u_0 = 0.4$, $R = 0.8$ at times $t = 15, 60$.

computations have been done on the interval $x \in (-20, 20)$ with 1000 cells and we have again used $C_{\max}$ 0.7 for LF and LW time steps and 2 for CN time steps. The structure of the shocks and rarefaction waves correspond to the Fig. 10 of [8]. Note that the downstream moving shock on the upper surface and on the interface (at $t = 60$) is better resolved by the LWLF4 or CNLF1F8 method than by the LFAD2 method.

Figure 5 shows the time evolution (at times $t = 15, 30, 100$) for the problem with $b_c = 1.2$, $u_0 = 0.4$, which is the problem 36 from [8], calculated by the LFAD2, LWLF4, and CNLF1F8 methods. In this problem there appear both ill-posed and well-posed regions. These regions are distinguished in the figure by the bold line which has the value 0 in the well-posed region and the value 0.1 in the ill-posed region. The ill-posed region lies between the shock standing on the downstream side of the bump and the shock travelling downstream. The computations have been done on the interval $x \in (-20, 20)$ with 2000 cells and we have used $C_{\max}$ 0.35 for LF method, 0.2 for LW method, and 0.5 for CN method with 0.2 for LF time substeps. More cells are needed to resolve the small dip on the interface standing around $x = 1.2$. The structure of the shocks and rarefaction waves corresponds to Fig. 8 of [8]. Note that the value of the interface height at the left-hand end of the bump $z_1(-2)$ is around 1.43, as predicted by analytical estimates in Table 1 of [8] while the Lax–Wendroff method with artificial viscosity gave in [8] this

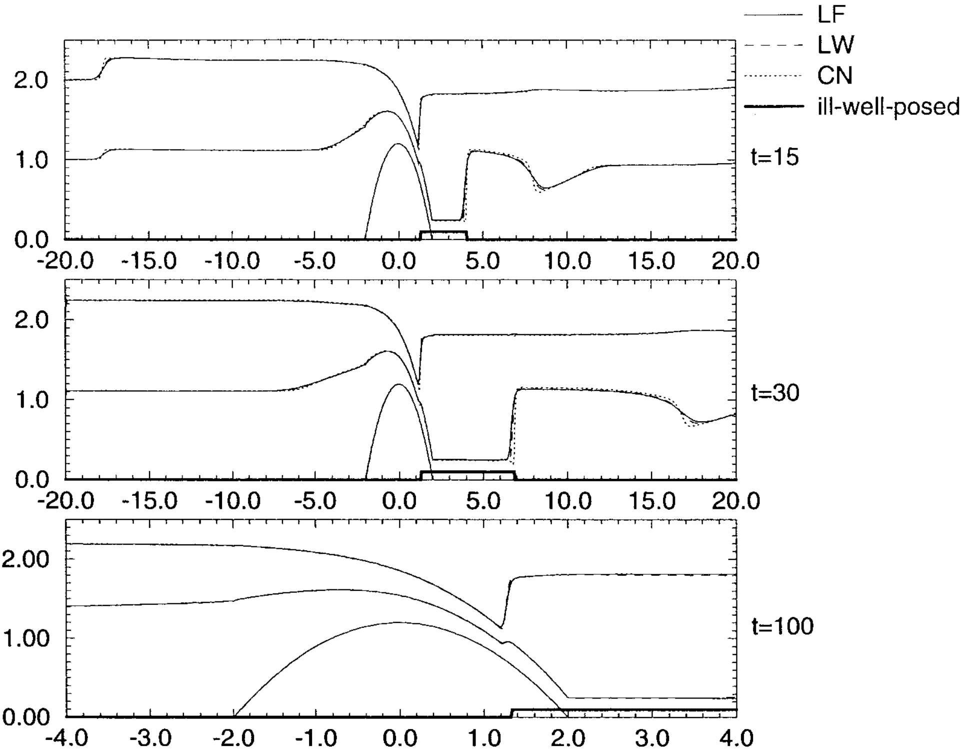

FIG. 5. Comparison of heights for two-layer shallow water model of LFAD2, LWLF4, and CNLF1F8 methods for the problem with $b_c = 1.2$, $u_0 = 0.4$, $R = 0.8$ at times $t = 15, 30, 100$. The bold line distinguishes the well-posed (value 0) and ill-posed (value 0.1) regions.

value 1.22. This applies also to other cases. We have to note that in some cases our results in some regions do not correspond to schematic drawings presented in [8] which, however, were not claimed to be precise but rather representative.

5.5. *Comparison of Schemes for Shallow Water*

We have used three families of difference schemes LFAD(W)n, LWLFn, and CNLFnFm with varying parameters W, n, m (see Section 4.1) and varying maximal CFL number $C_{\max}$ (see Section 5.2). From our experience the reasonable values of these parameters for the LFAD(W)n methods are $W \in (0.5, 1)$, $1 \leq n \leq 4$, $C_{\max} \in (0.2, 0.9)$, for LWLFn methods $2 \leq n \leq 8$, $C_{\max} \in (0.2, 0.9)$, and for CNLFnFm methods $1 \leq n \leq 8$, $4 \leq m \leq 16$, $C_{\max} \in (0.5, 3)$ (note that this $C_{\max}$ is for the CN step, for the LF step it has to be less than one). For most problems good results are obtained with LFAD2 ($C_{\max} \approx 0.7$), LWLF4 ($C_{\max} \approx 0.7$), and CNLF1F8 ($C_{\max} \approx 2$ for CN step and $C_{\max} \approx 0.5$ for LF step) methods. As might be expected LWLF and CNLF schemes resolve shocks better than LFAD schemes (see Fig. 4). As is well known the anti-diffusion might introduce staircase behavior in some regions of smooth solution. The same applies to the filter, especially if it is used too often. Being implicit the CNLF schemes are generally slower, even with

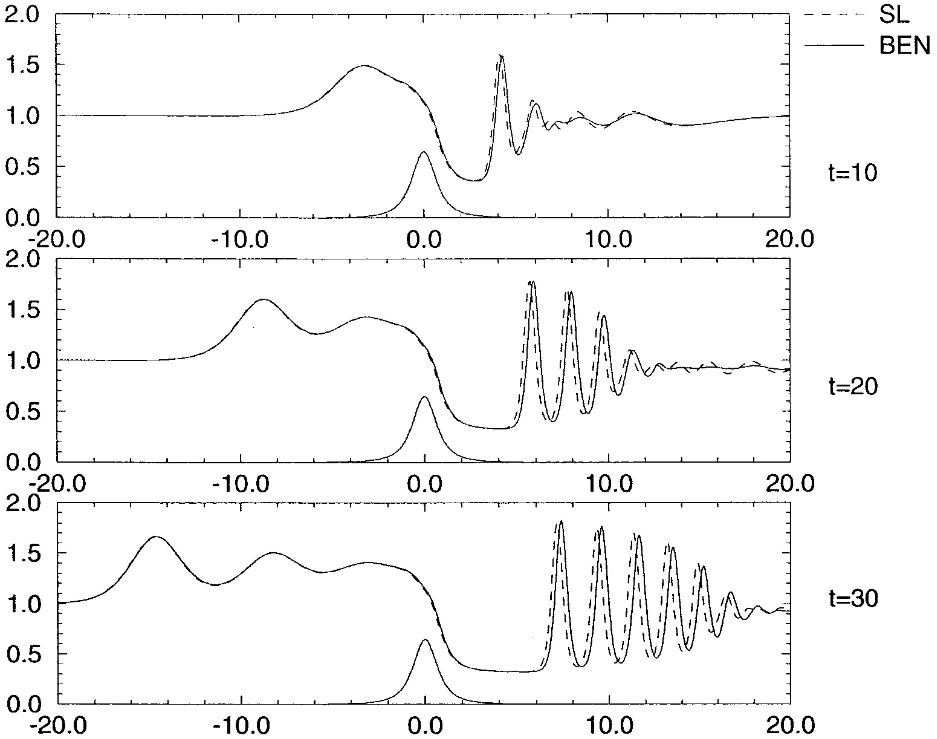

FIG. 6. Comparison of heights z_1 for one-layer GN semi-Lagrangian code with our one-layer GN code.

greater CFL limit, than explicit LFAD and LWLF schemes. A generalization of the LWLF scheme to two dimensions has proved to work well [26].

5.6. *One-Layer Green–Naghdi*

As the first check of the code we have used a solitary wave solution (32) [5] of one-layer Green–Naghdi. We have obtained very good agreement between the calculated numerical solution and the exact solitary solution.

5.6.1. *Comparison with a semi-Lagrangian code.* Here we compare our code with the semi-Lagrangian (SL) code for the Green–Naghdi equations [2], where six cases are studied. The bottom profile is given by (26). The initial conditions (22) with $n = 1$ are given by $z_1^0 = 1$, $u_1^0 = u_0$.

We have obtained reasonably good agreement in all six cases presented in [2]. As an example we present in Fig. 6 the comparison for case 1 from [2] for which $u_0 = 0.7$, $b_c = 0.65$. This computation has been done on the interval $x \in (-40, 40)$ with 2000 cells and $C_{\max} = 0.8$. As can be seen in the figure there is a slight disagreement in the speed of the downstream high frequency, high amplitude waves.

5.6.2. *Comparison with a steady solution of the Euler equations.* In [25] the steady state solutions of the Euler equations are obtained by a boundary integral formulation. The bottom profile is given by (25). The initial conditions (22) and

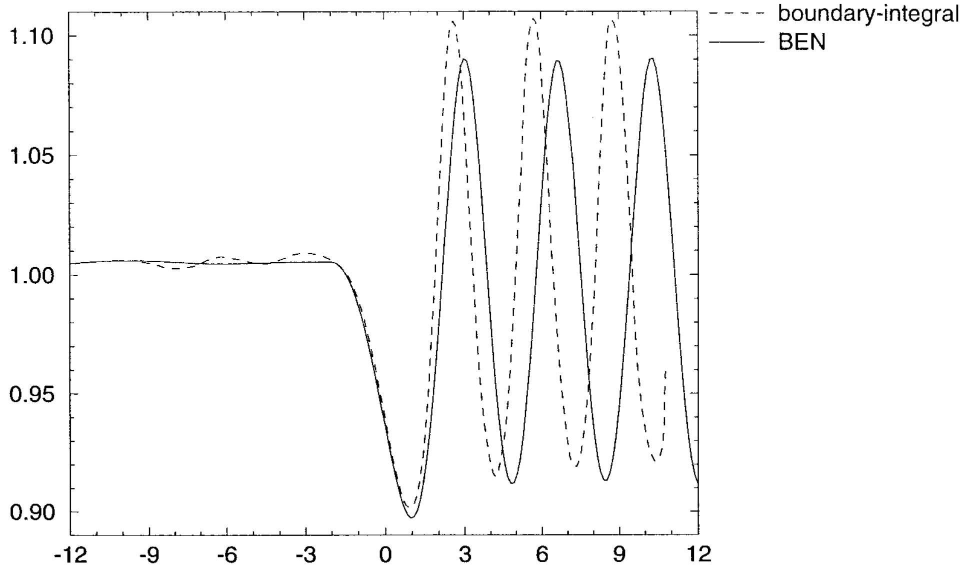

FIG. 7. Comparison of heights for steady state solution obtained by the boundary integral formulation and our one-layer GN code.

the boundary conditions (23) with $n = 1$ are used again. In Fig. 7 we present a comparison of the study state solution obtained from our Green–Naghdi code (BEN) and the one from Fig. 3 in [25] for the case $z_1^0 = 1$, $u_1^0 = 1/\sqrt{2}$, $b_c = 0.056$. This computation has been done on the interval $x \in (-80, 80)$ with 2000 cells and $C_{\max} = 1.6$. The wavelength of GN downstream standing waves is greater than that of the boundary integral formulation and their amplitude is smaller. We also do not have steady upstream waves, our upstream profile is flat. However, we do not know to what extent this discrepancy is due to the boundary conditions or the Green–Naghdi approximation itself.

5.7. *Two-Layer Green–Naghdi*

In all the two-layer Green–Naghdi tests we use the bottom profile (26). The initial conditions are given by (22) and boundary conditions by (23) with $n = 2$.

When we set $R = 0$ and keep the interface constant and flat, then the soliton solution of single-layer Green–Naghdi (32) described in Appendix B is the solution of the two-layer Green–Naghdi for the upper layer. For $R = 0$ the height of the interface $z_1(x, t)$ is independent of space and time when either the bottom is flat or the initial velocity of lower layer is zero. Of course, $R = 0$ means that the density of the upper layer is zero which does not make real physical sense. However, this analytical solution has served as a starting point for checking numerical codes for this complicated system.

5.7.1. *Comparison with one-layer Green–Naghdi.* Both single-layer and two-layer Green–Naghdi models are approximations of the full-dimensional Euler equa-

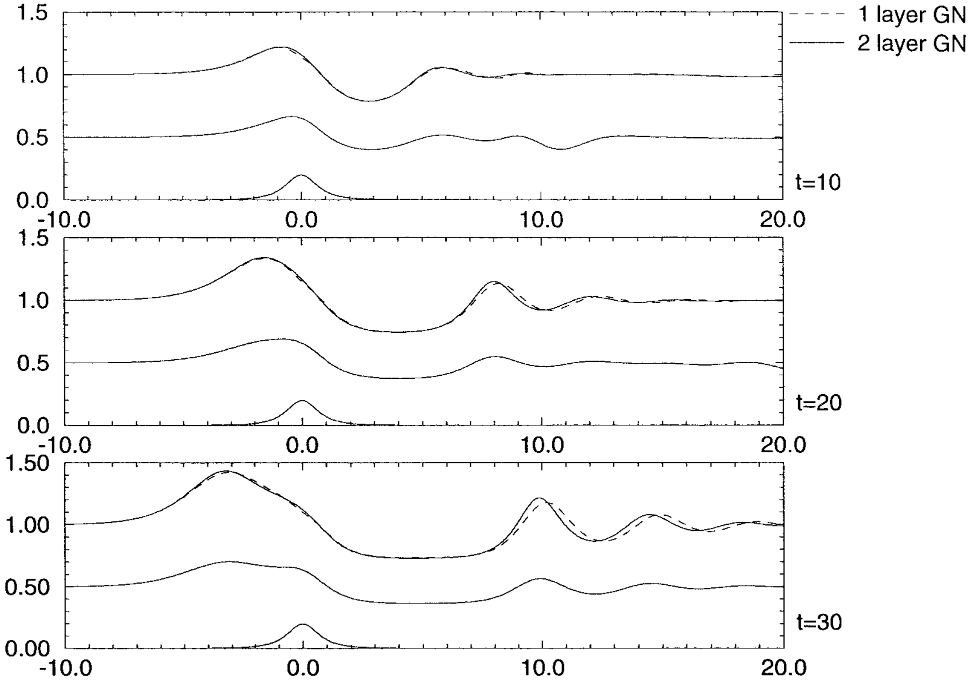

FIG. 8. Comparison of heights for GN one- and two-layer codes.

tions. Here we compare these two models on passive layer solutions with constant density which have been discussed in Section 2.4.

The comparison of our one- and two-layer Green–Naghdi codes for case 3 from [2] with

$$z_1^0 = 0.5, \quad z_2^0 = 1, \quad u_1^0 = u_2^0 = 1, \quad R = 1, \quad b_c = 0.2$$

is shown in Fig. 8. This computation has been done on the interval $x \in (-40, 40)$ with 2000 cells and $C_{\max} = 5$. Initially equal velocities of the layers remain close and the results agree nicely for the height of the upper surface.

However, for some other configurations we are unable to calculate with two-layer GN codes long enough; the instability breaking the computation appears quite early. As an example we present in Fig. 9 the computation for the configuration used in Fig. 6, with

$$z_1^0 = 0.825, \quad z_2^0 = 1, \quad u_1^0 = u_2^0 = 0.7, \quad R = 1, \quad b_c = 0.65. \tag{27}$$

The downstream wave in the two-layer solution stays at around $x = 3$ and its maximum becomes very sharp, which breaks further computation. We present results of both ad-hoc and BEN codes. As can be seen the instability in the BEN results is just starting at $t = 6$. Maximums of the downstream waves which are sharper than those of single-layer GN results have appeared also in other cases.

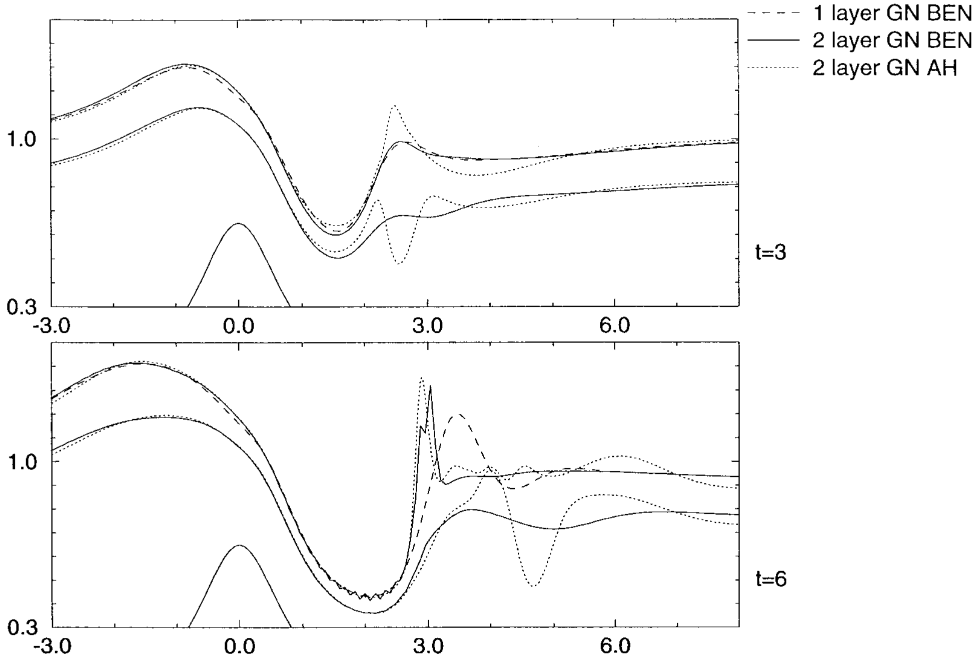

FIG. 9. Comparison of heights for GN one- and two-layer codes with instability.

5.7.2. *Regularization by dissipation.* The instability in the example with parameters (27) presented in Fig. 9 can be regularized by fourth-order dissipation. Of course, the question is how much dissipation should be added, i.e. how to choose ν. In Fig. 10 we present the results of the ad-hoc scheme with dissipation coefficients $\nu = 0.01, 0.0025$ (for $\nu = 0.001$ the computation is broken by instability around $t = 10$) and results of the BEN scheme with $\nu = 0.1, 0.025$ (for $\nu = 0.025$ the computation is broken by instability around $t = 17$ and for $\nu = 0.01$ around $t = 7$). Note that for the BEN scheme more dissipation is needed which causes deformation of the solution around the top of bump.

5.7.3. *Comparison with an Euler code.* The comparison of our 2-layer Green–Naghdi code with an Euler code for case 3 from [2] with

$$z_1^0 = 0.5, \quad z_2^0 = 1, \quad u_1^0 = u_2^0 = 1, \quad R = 1, \quad b_c = 0.2,$$

is shown in Fig. 11. The interface between the layers is a streamline starting at height 0.5 at the left boundary, obtained for us from the Euler code by Piotr Smolarkiewicz and Balu Nadiga. Although the GN upstream waves are faster than Euler ones we can again say that the results qualitatively agree.

5.7.4. *Convergence test.* We have performed several convergence tests by computing with decreasing time and space grid intervals while keeping their ratio constant. The convergence tests show in some cases reasonably good agreement of

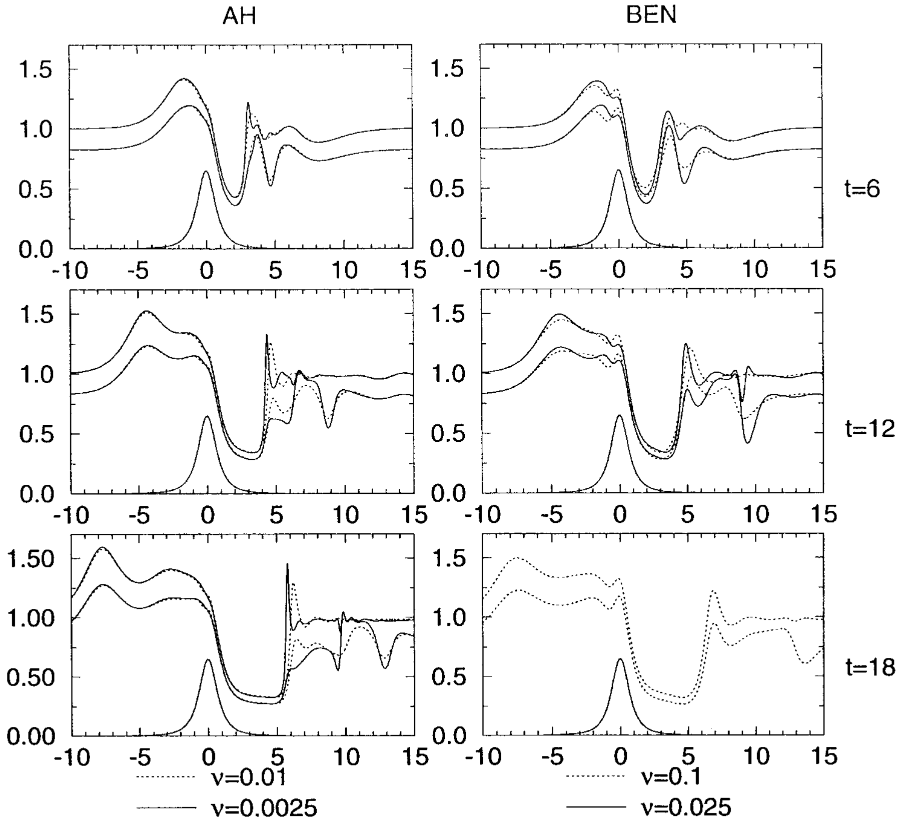

FIG. 10. Regularization of the instability of two-layer GN presented in Fig. 9 by fourth-order dissipation done by ad-hoc (AH) and backward Euler Newton (BEN) schemes with different dissipation coefficients ν (BEN with $\nu = 0.025$ becomes unstable around $t = 17$, so it does not appear in the plot at $t = 18$).

the inner and upper surface heights as the grids are refined while in other cases the convergence is not achieved.

In Fig. 12 we present one such convergence test for initial/boundary data

$$z_1^0 = 0.65, \quad z_2^0 = 1, \quad u_1^0 = u_2^0 = 1, \quad R = 1, \quad b_c = 0.3.$$

The test has been performed for three sucessive grid refinements with 1000, 2000, and 10000 points on the interval $x \in (-50, 50)$ with the CFL number $C_{\max} = 2.5$.

For finer grids a kind of wavelet is generated on the layers interface on the downstream side of the bump. The wavelet speed is higher than the speed of downstream waves. The amplitude of the wavelet grows until the interface touches the upper surface which violates the model assumption of positive thicknesses of both layers and causes an instability later. The presented data are from ad-hoc code. We have tried the same problem with BEN code which produced a similar wavelet and becomes unstable much sooner, as soon as the interface touches the upper surface, e.g. with 2000 cells and $C_{\max} = 5$ around $t = 10$.

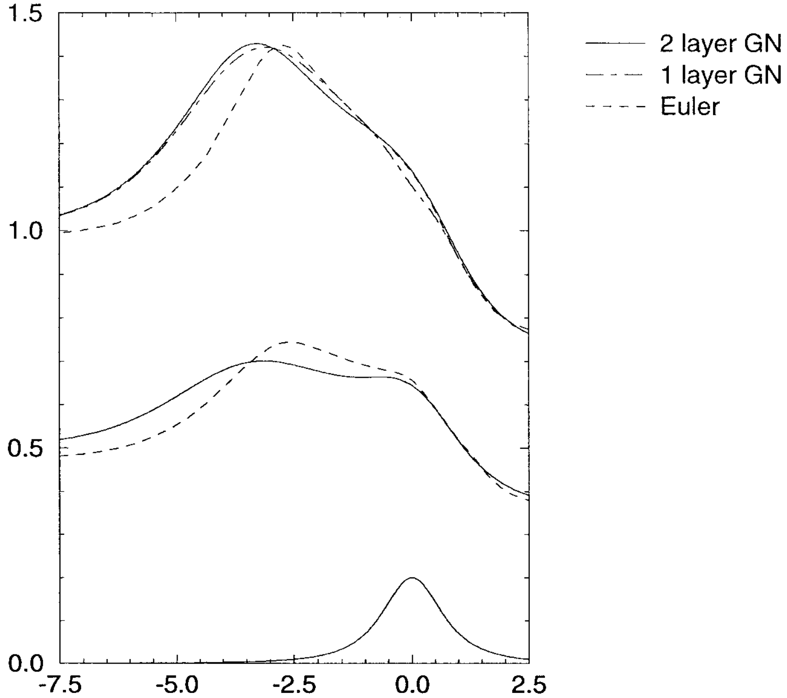

FIG. 11. Comparison of heights z_1, z_2 of the two-layer Green–Naghdi (GN) code with an Euler code for layers (0, 0.5) and (0.5, 1).

This instability is surely related to the ill-posedness of the two-layer Green–Naghdi model. Similar instabilities have appeared also in some other two-layer tests. The dissipation does not remove this wavelet behavior, it only decreases its amplitude.

5.7.5. *Numerical properties of codes.* As shown in Section 3.4 the two-layer Green–Naghdi model is ill-posed. The numerical examples show that in some cases we do not obtain convergence as the grid is refined (see Fig. 12) and in others the instability breaks down the computation rather early (see Fig. 9). Even for some parameters the computation breaks just in its beginning in the first time steps.

A common feature of these problematic numerical results (without dissipation) is their dependence on space and time steps Δx, Δt. Usually there exists an optimal space step Δx and an optimal CFL number $C_{\max}$ (which determines the adaptive time step Δt as described in the Section 5.2) for which the result looks most promising, meaning in most cases that the computation can proceed without instability up to the biggest time. For finer/coarser space grid or greater/less CFL number the instability usually breaks the computation earlier. Figure 9 presents an example of such optimally chosen steps (see the Table I).

The instabilities, for cases where they appear, can be avoided by regularizing the model with a fourth-order dissipation, although the system of equations is then not dissipative, only boundedly well-posed. At the suggestion of G. Browning we have included a computation with these dissipative terms included (see Fig. 10). However,

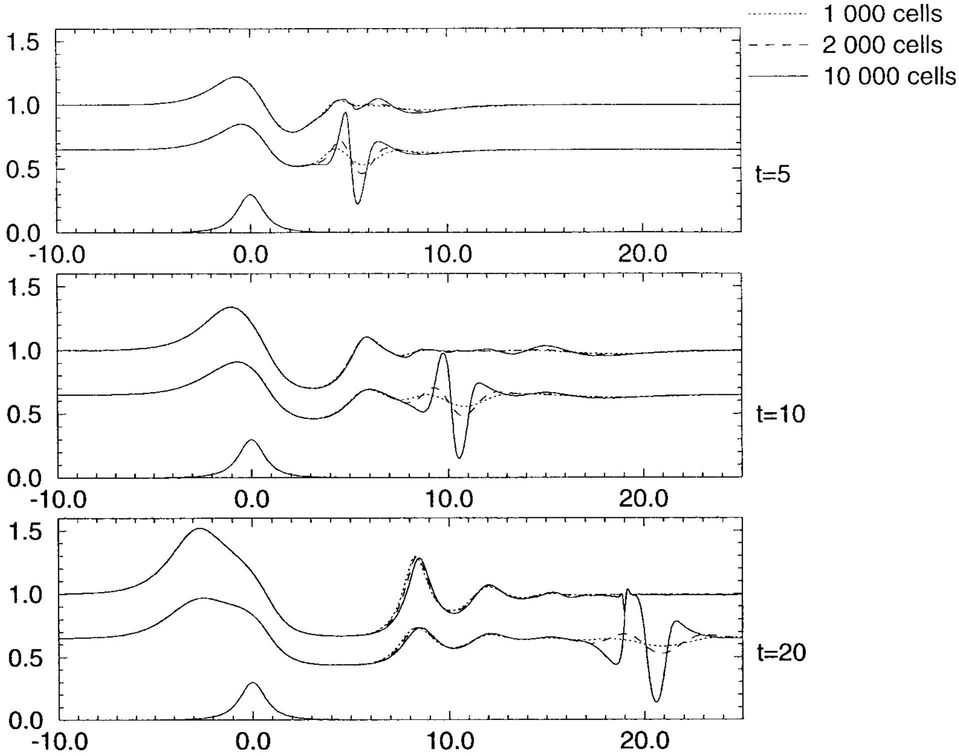

FIG. 12. Convergence test for two-layer Green–Naghdi with $b_c = 0.3$ and interface at $z_1^0 = 0.65$ at times $t = 5, 10, 20$.

even with the dissipation a problem with the layer thickness approaching zero and, thus, violating the averaged model the assumption remains. What is said in the previous paragraph is now extended by the dependence on the dissipation coefficient ν. If ν is too small, instability is not avoided; if it is too large, it might deform the solution.

5.8. *Summary of Examples*

The previous sections include a lot of examples using different continuous models, different numerical methods, and different data. In Table I we present a summary of all examples presented in all figures in this paper. The table includes the figure number, model (e.g., SW1 is single-layer shallow water, GN2 is two-layer Green–Naghdi), initial conditions (z_1^0, u_1^0 for single layer problems and $z_1^0, z_2^0, u_1^0, u_2^0$ for two-layer ones), height of the bump b_c, ratio of densities $R = \rho_2/\rho_1$, half-length of the interval on which the problem has been solved A (solved for $x \in (-A, A)$), number of cells, used numerical method (as described in Section 4), CFL limit $C_{\max}$ used for the adaptive time step calculation (see Section 5.2) and a note, which includes an abbreviation of cited paper (NMS is Nadiga–Margolin–Smolarkiewicz [2], HI is Houghton–Isaacson [8], and BF is Belward–Forbes [25]) and the number of the example from that paper solving the same problem.

TABLE I
Summary of Presented Example Computations

		z_1^0	u_1^0							
Fig.	Model	z_1^0, z_2^0	u_1^0, u_2^0	b_c	R	A	Cells	Method	$C_{\max}$	Note
1	SW1	1	0.7	0.65	—	40	2000	LFAD2	0.7	NMS 1
								LWLF2	0.7	
								CNLF1F8	2/0.7	
2	SW1	1	1	0.2	—	20	1000	LFAD2	0.7	NMS 3
	SW2	0.5, 1	1, 1		1			LWLF2	0.7	
								CNLF1F8	2/0.7	
3	SW1	1	0.7	0.65	—	20	1000	LFAD2	0.35	NMS 1
	SW2	0.825, 1	0.7, 0.7		1			LWLF2	0.2	
								CNLF1F8	0.5/0.2	
4	SW2	1, 2	0.4, 0.4	0.8	0.8	20	1000	LFAD2	0.7	HI 15
								LWLF4	0.7	
								CNLF1F8	2/0.7	
5	SW2	1, 2	0.4, 0.4	1.2	0.8	20	2000	LFAD2	0.35	HI 36
								LWLF4	0.2	
								CNLF1F8	0.5/0.2	
6	GN1	1	0.7	0.65	—	40	2000	BEN	0.8	NMS 1
7	GN1	1	$1/\sqrt{2}$	0.056	—	80	2000	BEN	1.6	BF 3
8, 11	GN1	1	1	0.2	—	40	2000	BEN	5	NMS 3
	GN2	0.5, 1	1, 1		1			AH	5	
9	GN1	1	0.7	0.65	—	40	2000	BEN	0.8	NMS 1
	GN2	0.825, 1	0.7, 0.7		1		1000	BEN	6	
						20	2000	AH	3	
10	GN2	0.825, 1	0.7, 0.7		1	20	2000	AH, BEN	3	NMS 1
12	GN2	0.65, 1	1, 1	0.3	1	50	1000–10000	AH	2.5	

6. CONCLUSION

There are two very well known vertically averaged single-layer approximations to the incompressible Euler equations: hyperbolic shallow water and dispersive Green–Naghdi. We have presented multilayer extensions of these models, posedness analysis for two-layer models, several finite difference schemes for their solution and a selected set of numerical examples of one-layer and two-layer problems with free surface.

The two-layer Green–Naghdi equations are ill-posed in both the rigid lid and free surface cases. The more familiar two-layer shallow water equations are of mixed type; that is, they are hyperbolic in some regions of state space but ill-posed in others. Both of these facts were shown using a computer algebra system.

The two-layer Green–Naghdi equations are ill-posed so basically we use for them the backward Euler finite difference scheme which in the presented examples proved to be stable enough to allow solution of this ill-posed problem, at least for some typical parameters. We have used two modifications of the nonlinear backward Euler, namely one with full Newton linearization and the other with ad-hoc linearization. The situation is further complicated by the extreme complexity of the equations, but this part of the difficulty was overcome by the use of automatic code generation software. Fortran code solving the implicit difference equations was created in this way and applied to several problems of flow over a bump. As far as we know this is the first attempt to solve two-layer Green–Naghdi equations. We showed that in one case in which there is a purely passive inner surface the one-layer and two-layer results agree on the top surface height, while the inner surface height agrees qualitatively with the corresponding streamline from an Euler equation calculation. On several examples we have shown difficulties arising during numerical solution of this model. Pure instabilities originating in the ill-posedness of the model can be avoided by introducing the fourth-order dissipation into the system. However, dissipation regularizes the problem only boundedly and even with dissipation one cannot achieve convergence in all cases. Further, the thickness of a fluid layer can, during the computation, approach zero which violates the assumption of the vertically averaged model.

Two-layer shallow water offers a different challenge that was first addressed in the 1970 paper [8]. Apart from the possible ill-posed region of state space there is also the difficulty of the appearance of shock waves in a system that is not in conservation form. Less serious but still troublesome is the fact that in a hyperbolic region the wave speeds are not known in a simple form, making it difficult to use numerical methods relying on an eigenvector expansion. We have proposed some simple explicit and implicit composite schemes that we tested on single-layer problems where they showed excellent resolution of the discontinuous and smooth parts of the solution. Here, as with Green–Naghdi, we included in our examples two problems with a passive second layer. In one case we found agreement for the top surface height, but in another this was not so for there it appeared that a shock in the single-layer model split into two shocks in the two-layer model. We suggest as the reason for this the fact that the two-layer difference equations in momentum variables do not maintain initially equal velocities and thus quickly get into the ill-posed region. It remains an open question as to which formulation of two-layer shallow water equations should be used.

APPENDIX A: DERIVATION OF THE GREEN–NAGHDI EQUATIONS

The derivation starts with the Euler equations in each layer i,

$$\begin{aligned} u_{ix} + w_{iz} &= 0, \\ \rho_i(u_{it} + u_i u_{ix} + w_i u_{iz}) &= -p_{ix} + F_i, \\ \rho_i(w_{it} + u_i w_{ix} + w_i w_{iz}) &= -p_{iz} - g\rho_i, \end{aligned} \tag{28}$$

for

$$z_{i-1} \le z \le z_i, \quad i = 1, ..., n,$$

where $u_i(x, z, t)$, $w_i(x, z, t)$ are horizontal and vertical velocities in the ith layer. The quantities F_i include various forcing terms, but are taken to be zero.

The boundary conditions (in z) involve total derivatives (2) of the heights z_i. The boundary conditions are

$$\mathrm{d}_i z_i = w_i(z = z_i) \tag{29}$$

and

$$\mathrm{d}_i z_{i-1} = w_i(z = z_{i-1}). \tag{30}$$

At the top surface $z = z_n$ we will pose either the rigid lid condition

$$z_n = \text{const}, \quad w_n = 0,$$

or the free surface condition

$$\mathrm{d}_n z_n = w_n, \quad p_n = 0.$$

Finally, the pressure is assumed continuous at the interfaces $z = z_i$, $i = 1, ..., n - 1$.

Now we suppose that the horizontal velocities u_i are independent of z. For the vertical velocities we set

$$w_i = \mathrm{d}_i z_{i-1} - (z - z_{i-1}) u_{ix}.$$

Then the incompressibility condition (28) and boundary condition (30) are satisfied. The other boundary condition (29) is equivalent to the mass conservation equations,

$$\mathrm{d}_i h_i + h_i u_{ix} = 0, \quad i = 1, ..., n.$$

Note that the vertical component of velocity is also discontinuous. The condition on the jump in velocity across the interface, $\langle\cdot\rangle$, is

$$\langle u \rangle z_{ix} = \langle w \rangle.$$

Now consider the layer $z_{i-1} \le z \le z_i$. Let

$$q_i(s) = \int^s \bar{\mathrm{d}}_i w_i dz,$$

where

$$\bar{\mathrm{d}}_i f = \mathrm{d}_i f + w_i f_z.$$

Now, using the mass conservation equation $\bar{\mathrm{d}}_i w_i$ can be rewritten:

$$\bar{\mathrm{d}}_i w_i = \mathrm{d}_i^2 z_{i-1} - \frac{z - z_{i-1}}{z_i - z_{i-1}} \mathrm{d}_i \left[(z_i - z_{i-1}) u_{ix}\right].$$

Then,

$$q_i(s) = s\mathrm{d}_i^2 z_{i-1} - \frac{\dfrac{s^2}{2} - s z_{i-1}}{z_i - z_{i-1}} \mathrm{d}_i\,[(z_i - z_{i-1})u_{ix}]. \tag{31}$$

Then integrating the vertical momentum equation from a point in the *i*th layer to the upper surface of that layer gives

$$p_i(z) = \mathscr{P}_i(x, y, t) + \rho_i[q_i(z_i) - q_i(z) + g(z_i - z)],$$

where $\mathscr{P}_i$ is the pressure at the interface. This is well defined by the assumption that the pressure is continuous at the interfaces.

The horizontal momentum equations are then obtained by integration across each layer, that is,

$$(z_i - z_{i-1})\mathrm{d}_i u_i + \int_{z_{i-1}}^{z_i} (q_i(z_i) - q_i(z))_x dz + (z_i - z_{i-1})g z_{ix} = -\frac{1}{\rho_i}(z_i - z_{i-1})\mathscr{P}_{ix},$$

evaluating the integral results in the multiple layer momentum equations (3).

APPENDIX B: SOLITON SOLUTION OF SINGLE-LAYER GREEN–NAGHDI

The equations (1) (with $n = 1$) and (7) with flat bottom, i.e. $h_0(x) = z_0(x) = 0$, have the solitary wave solution [5]

$$\begin{aligned} z_{1s}(x,t) &= 1 + (z_m - 1)\operatorname{sech}^2\left[\sqrt{\frac{3}{4}\left(1 - \frac{1}{z_m}\right)}\,(x - \sqrt{z_m}t)\right], \\ u_{1s}(x,t) &= \sqrt{z_m}\left(1 - \frac{1}{z_{1s}(x,t)}\right), \end{aligned} \tag{32}$$

where z_m is the maximal height of solitary wave. The unperturbed height of this upper surface is $z_1 = 1$.

ACKNOWLEDGMENTS

This work was supported in part by the CHAMMP Program of the U.S. Department of Energy. R. Liska was supported in part by the National Science Foundation International Programs Grant INT-9212433, the Czech Grant Agency grant 201/94/1209, Czech Ministry of Education Grants KONTAKT ES 016 (1996) and ME 050 (1997), and he thanks the Institute for Geophysics and Planetary Physics of the Los Alamos National Laboratory for hosting his visits at Los Alamos. We also thank Balu Nadiga and Piotr Smolarkiewicz for their help in obtaining Figs. 6 and 11 and Heinz Kreiss and Gerald Browning for pointing out the importance of the dissipation for two-layer Green–Naghdi. B. Wendroff also thanks both Professor Kevin Burrage for providing the opportunity to visit the University of Queensland and Professor Lawrence Forbes for informing him about his work in the area of flow over obstacles. We also thank the reviewers for many helpful comments which we have incorporated in the paper.

REFERENCES

1. D. D. Houghton and A. Kasahara, Nonlinear shallow fluid flow over an isolated ridge, *Comm. Pure Appl. Math.* **21,** 1 (1968).
2. B. T. Nadiga, L. G. Margolin, and P. K. Smolarkiewicz, Different approximations of shallow fluid flow over an obstacle, *Phys. Fluids* **8**(8), 2066 (1996).
3. H. Lamb, *Hydrodynamics,* 6th ed. (Dover, New York, 1945).
4. B. Wendroff, Composite flows, *Physica D* **60,** 208 (1992).
5. A. E. Green, N. Laws, and P. M. Naghdi, On the theory of water waves, *Proc. R. Soc. Lond. A* **338,** 43 (1974).
6. S. V. Bazdenkov, N. N. Morozov, and O. P. Pogutse, Dispersive effects in two-dimensional hydrodynamics, *Sov. Phys. Dokl.* **32** (1987).
7. D. D. Holm, Hamiltonian structure for two-dimensional hydrodynamics with nonlinear dispersion, *Phys. Fluids* **31,** 2371 (1988).
8. D. Houghton and E. Isaacson, Mountain winds, in *Studies in Numerical Analysis* (J. M. Ortega and W. C. Rheinboldt, Eds.) (SIAM, Philadelphia, 1970), p. 21.
9. D. Houghton, An example of interaction between finite-amplitude gravity waves, *J. Atmos. Sci.* **21,** 493 (1964).
10. J. L. Bona and R. L. Sachs, The existence of internal solitary waves in a two-fluid system near the kdv limit, *Geo. Astr. Fl. Dyn.* **48,** 25 (1989).
11. P. D. Weidman and M. Johnson, Experiments on leapfrogging internal solitary waves, *J. Fluid Mech.* **122,** 195 (1982).
12. L. K. Forbes, Two-layer critical flow over a semi-circular obstruction, *J. Eng. Math.* **23,** 325 (1989).
13. S. Diebels, Numerical study of barotropic and baroclinic behavior of a nonlinear two-layer model, *J. Comput. Phys.* **117,** 114 (1995).
14. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problems,* 2nd ed. (Wiley, New York, 1967).
15. H. B. Stewart and B. Wendroff, Two-phase flow: Models and methods, *J. Comput. Phys.* **56,** 363 (1984).
16. R. Liska, L. Margolin, and B. Wendroff, Nonhydrostatic two-layer models of incompressible flow, *Comput. Math. Appl.* **29**(9), 25 (1995).
17. A. C. Hearn, *REDUCE User's Manual, Version 3.6,* Technical Report CP 78 (Rev. 7/95) (RAND, Santa Monica, CA, 1995).
18. L. González-Vega, A combinatorial algorithm solving some quantifier elimination problems, in *Quantifier Elimination and Cylindrical Algebraic Decomposition* (B. F. Caviness and J. R. Johnson, Eds.), (Springer-Verlag, Vienna, 1996).
19. M. Watson, Nonlinear waves in pipeline two-phase flow, in *Third International Conference on Hyperbolic Problems, 1990* (B. Engquist and B. Gustafson, Eds.).
20. L. Sainsaulieu, Finite volume approximation of two phase-fluid flows based on an approximate roe-type riemann solver, *J. Comp. Phys.* **121,** 1 (1995).
21. T. Boukadida and A.-Y. LeRoux. A new version of the two-dimensional Lax-Friedrichs scheme. *Math. Comp.* **63,** 541 (1994).
22. B. Engquist, P. Lötstedt, and B. Sjögreen, Nonlinear filters for efficient shock computation, *Math. Comp.* **186,** 509 (1989).
23. R. Liska and L. Drska, FIDE: A REDUCE package for automation of FInite difference method for solving pDE, in *ISSAC '90, Proceedings of the International Symposium on Symbolic and Algebraic Computation, Tokyo, August 20–24, 1990* (S. Watanabe and M. Nagata, Eds.), (ACM Press/Addison–Wesley, New York, 1990), p. 169.
24. R. Liska, Numerical code generation for finite difference schemes solving, in *Computational and Applied Mathematics I—Algorithms and Theory,* (C. Brezinski and U. Kulisch, Eds.) (Elsevier,

Amsterdam, 1992), p. 295. [IMACS, Selected and revised papers from the 13th IMACS World Congress, Dublin, Ireland, July 1991]

25. S. R. Belward and L. K. Forbes, Interfacial waves and hydraulic falls: Some applications to atmospheric flows in the lee of mountains, *J. Eng. Math.* **29,** 161 (1995).

26. R. Liska and B. Wendroff, *Composite Schemes for Conservation Laws,* Technical Report LA-UR 96-3589, LANL, Los Alamos, 1996. [*SIAM J. Numer. Anal.*, to appear.]