CHAPTER

Rotating shallow water dynamics: Dispersion and nonlinearity

6

CONTENTS

6.1 Realism in wave models

From the point of view of a practitioner the previous chapter is quite disheartening. Much of the classical theory seems to be the tail wagging the dog, with mathematical descriptions driving both choices of methodology and the nature of conclusions once can draw. The classical "Ockham's razor" point of view (i.e. build the simplest possible model that is consistent with the data) is thus challenged by the desire to find conclusions as formulae (i.e. in closed form).

On the other hand, the numerical methods featured in the previous chapter are quite advanced, and beyond the set of tools that a typical scientist has experience with. We will thus take a step back and work to close the gap between theory and model. We will do this by examining the physics (and indeed the missing physics) of the shallow water model, and will build more complex simulations from very simple, laboratory inspired, set ups. This will be developed over the space (Chapters 6–8).

In this chapter we develop simple models that we have no intention of solving analytically. We choose these models so that they extend the classical single layer shallow water theory so that the model is more representative of what is already known about real (internal) waves in lakes:

1. Real waves are dispersive (i.e. waves of different length scales travel at different speed).
2. Real waves are finite amplitude (i.e. the nonlinear terms in the governing equations should not be summarily dropped).
3. Real waves are affected by the Earth's rotation.

Physics and Ecology in Fluids. https://doi.org/10.1016/B978-0-32-391244-0.00016-4

4. Real waves are affected by small scale turbulence and viscosity.

For the first three points tidy model formulations are available. The final point remains a somewhat open question and a satisfying answer will not be shown to the reader until Chapters 7 and 8.

The role theory plays in this chapter is to set the stage for numerical methods. In particular, we want all assumptions to be clearly laid out. We of course assume that the numerical solution methods of the models we show are reasonable to derive, code, and run. More often than not, in the modern computing landscape, this is a good assumption and the pathologies of numerical analysis rarely show up in practice.

6.2 The dispersive shallow water model

The classical shallow water equations without rotation are a common feature of both advanced numerical methods and advanced partial differential equations courses. This is because they are an example of a nonlinear hyperbolic system ([4]). They thus carry information exactly along so-called characteristics curves, and for a very wide class of initial conditions they form shocks in finite time.

Shocksare solutions with a jump, or discontinuity, basically a mathematical idealization of the "surfer's" wave shown in Fig. 5.7. Due to their discontinuous nature, shocks appear to violate the idea of an equation involving spatial derivatives, and the mathematics of describing shocks (i.e. weak solutions) is thus quite impressive. Breaking waves do form in lakes, but they do so at scales of description far shorter than those of a typical lake model's single grid box. As such it seems a strange area of focus for a practically minded modeler (an excellent, if sophisticated introduction can be found in [5]).

A different topic of classical applied mathematics proves far more profitable. It has been known since the 19th century that the Korteweg-de Vries (or KdV) allows for so-called solitary wave solutions, or waves that solve a nonlinear equation but do not change shape as they propagate ([8]). The KdV equation is a single-direction advection equation augmented by both nonlinear and dispersion terms,

$$\frac{\partial A}{\partial t} = -c\frac{\partial A}{\partial x} + \alpha A\frac{\partial A}{\partial x} + \beta\frac{\partial^3 A}{\partial x^3}$$

where α and β are physical parameters determined from local conditions (i.e. the stratification in a particular lake). In the second half of the 20th century an amazing amount of applied mathematics was developed for this equation; the so-called inverse scattering, or soliton, theory.

The primary lesson from the KdV equation for our modeling purposes is that if dispersion is present in a system it may balance nonlinearity and preclude shock formation. Both surface water waves and internal waves are well known to exhibit dispersion (with longer waves observed to travel faster), and hence a model with

dispersion is highly desirable. Indeed, the classical shallow water equations without rotation are a very rare example in the physical description of waves that yields nondispersive waves. The classical shallow water waves have many attractive features, since they represent the typically small aspect ratio of natural motions in lakes (i.e. horizontal length scales that are much larger than vertical length scales). Thus an attractive applied mathematical problem is "How can we most economically represent dispersion in extensions of the classical SW system".

The underlying idea for representing wave dispersion in layered model equations is that weak vertical accelerations (which lead to nonhydrostatic pressure) should be parametrized as horizontal dispersion. The route to this parametrization can be quite mathematically laborious, but the key idea is relatively simple. We follow the work of de la Fuente et al. [2], who studied internal waves in a circular basin for a single fluid layer. The governing equations used by these authors read

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0\,, \tag{6.1}$$

$$\frac{\partial (uh)}{\partial t} + \nabla \cdot ((uh)\mathbf{u}) = -gh\frac{\partial \eta}{\partial x} + fvh + \frac{H^2}{6}\frac{\partial}{\partial x}\left(\nabla \cdot \frac{\partial (\mathbf{u}h)}{\partial t}\right), \tag{6.2}$$

$$\frac{\partial (vh)}{\partial t} + \nabla \cdot ((vh)\mathbf{u}) = -gh\frac{\partial \eta}{\partial y} - fuh + \frac{H^2}{6}\frac{\partial}{\partial y}\left(\nabla \cdot \frac{\partial (\mathbf{u}h)}{\partial t}\right), \tag{6.3}$$

where $\mathbf{u} = (u(x, y, t), v(x, y, t))$ is the velocity field, $h(x, y, t) = H(x, y) + \eta(x, y, t)$ is the total depth with H representing the undisturbed depth, and η is the free surface displacement. The constants g and f are the acceleration due to gravity and the Coriolis frequency, respectively. The difference between the set of equations (6.1)–(6.3) and the traditional shallow water model is the inclusion of the dispersive terms $\frac{H^2}{6}\nabla(\nabla \cdot (\mathbf{u}h)_t)$ found in the momentum equations (6.2) and (6.3). Indeed, the above system was first proposed by Brandt et al. [1] in their study of internal waves in the Strait of Messina. As mentioned above, this system is derived by a perturbation expansion in powers of the small dimensionless parameter $\mu = (H/L)$, and therefore is only physically relevant if $\mu \ll 1$ ([2]). As a small aside, note that the variable we refer to as h, is referred to as z by some authors. We found this notation to be too confusing when comparing various references, and perhaps by noting it here we can save the reader some time.

In the above form of the equations, we have neglected bottom and surface stresses in Eqs. (6.1)–(6.3) since their inclusion into numerical schemes is conceptually simple and contributes little to the discussion. We will use the quadratic stress in this chapter which would take the form

$$-c_d|\vec{u}|\vec{u}h \tag{6.4}$$

where the factor of h is included because the equations above are given for $\vec{u}h$ and not $\vec{u}$.

To relate these equations to the classical shallow water equations, consider the case of one dimensional flow without rotation, with a flat bottom. Some simple algebraic manipulations lead to the set of equations,

$$u_t + uu_x = -g\eta_x + \frac{H^2}{6}u_{xxt} \tag{6.5}$$

$$\eta_t + [H+\eta)u]_x = 0. \tag{6.6}$$

Writing the equations this way we can see that the conservation of mass equation is unchanged, but the conservation of momentum has an additional term. This term represents the nonhydrostatic pressure changes, but manifests as a mixed space-time derivative of the velocity u. The challenge for numerical methods is how to integrate this new term into the time-stepping method. Some methods, such as those used later in this chapter have a very easy time with this. For others, like those discussed in Chapters 9–11, it can be a challenge.

The linearized form of Eqs. (6.1)–(6.3) in one space dimension with $f = 0$ take the form

$$\eta_t + (Hu)_x = 0\,, \tag{6.7}$$

$$u_t = -g\eta_x + \frac{H^2}{6}u_{xxt}\,. \tag{6.8}$$

If we again assume a flat bottom ($H =$ constant) and make the wave ansatz, or that η and u are proportional to $e^{i(kx-\sigma t)}$, we can derive the dispersion relation

$$\sigma^2 = \frac{gHk^2}{1+\frac{H^2}{6}k^2}\,, \tag{6.9}$$

with phase speed

$$c_p = \frac{\sigma}{k} = \pm\frac{\sqrt{gH}}{\left(1+\frac{H^2}{6}(k^2+l^2)\right)^{1/2}}\,. \tag{6.10}$$

The shallow water speed is recovered in the long wave limit as $k \to 0$,

$$c_p = \frac{\sigma}{k} \to \pm\sqrt{gH}\,. \tag{6.11}$$

The phase speed decreases from this value and does so significantly when k is large (short waves). It is thus clear what this new term can get us; as wave steepening due to nonlinearity begins to play a role, we effectively activate waves with larger and larger values of k, but the dispersion then causes these waves to lag behind, giving some hope that for many cases shock formation will not play a prominent role in the dynamics. A discussion of the group speed is left to the reader as an exercise.

For the interested reader, the above expressions (6.9) and (6.10) can be compared to the exact dispersion relation for gravity waves in a single layer fluid derived from potential flow theory ([3]):

$$c_p = \frac{\sigma}{k} = \pm\sqrt{\frac{g\tanh(kH)}{k}}\,. \tag{6.12}$$

The phase speeds (6.11), (6.12), and (6.12) are compared in Fig. 6.1. All three dispersion relations agree in the long wave limit $k \to 0$, and the dispersive shallow water model's dispersion relation agrees qualitatively with the behavior of the full dispersion relation for gravity waves (6.12).

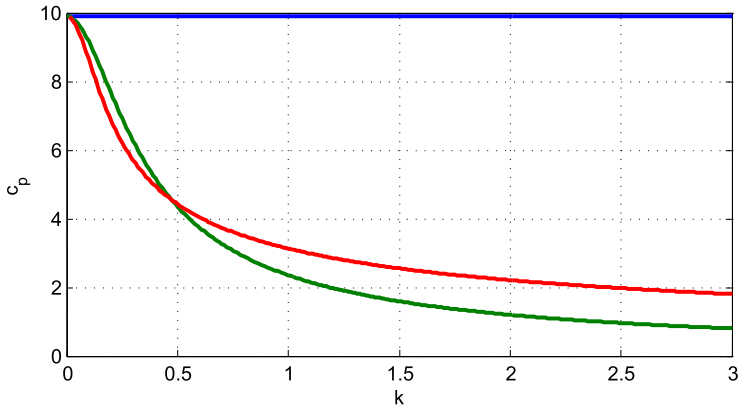

FIGURE 6.1

Comparison of phase speeds from the traditional shallow water mode (blue), the dispersive shallow water model (green), and the full dispersion relation from potential flow theory (red).

The importance of the dispersion is made clear in a standard "adjustment" simulation. The relevant code is given in

```
swnh1d_shortscale.m
```

with parameter values listed in Table 6.1. The domain is periodic in x for simplicity, extending over $-L \leq xleL$. An initial increase in the free surface height occupying $-0.1L \leq xle0.1L$, with flanks smoothed over a region $0.01L$ thick, is allowed to evolve. This yields both rightward and leftward propagating waveforms. For the rightward propagating waveforms, the resulting free surface (scaled by the total depth H) and horizontal velocity (scaled by the wave speed c_0) are shown in Figs. 6.2 and 6.3 for early and late times, respectively. It can be seen that even at early times, the classical SW solution (in red) steepens to form a shock, which is then somewhat smoothed by the numerical method (see the discussion in the final section of this chapter for details). In contrast the equations with the dispersive correction lead to the formation of a wave train and no sign of a shock is observed. Indeed, for later times the train of waves is quite clear, with the largest wave leading (i.e. the wavetrain is rank-ordered).

Table 6.1 Parameters for the Adjustment case.

Label	N	L (m)	H (m)	g (m s^{-1})	η_0 (m)	f (s^{-1})
Short adjustment	2048	1000	10	0.0981	2	0

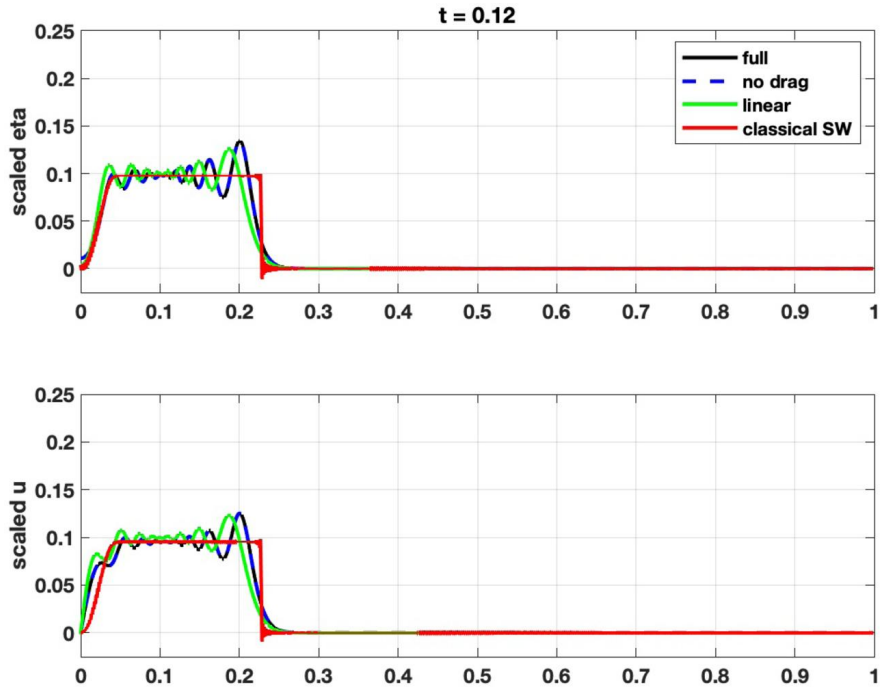

FIGURE 6.2

An adjustment problem for early dimensionless time ($t = 0.12$). Black – full model, blue – model without drag, green – linearized model, red – classical SW (no dispersion). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

For the times shown the quadratic bottom stress damping has little effect. For completeness we show the evolution of the linearized system in green. It can be seen that linearized system leads to a breakdown into an incoherent wave field (due to dispersion). This is due to the fact that steepening cannot occur (because the nonlinear terms have been dropped), and hence dispersion cannot be balanced, as in the classical KdV equation.

6.3 Adjustment problems on the field scale

The sample numerical simulation discussed in the previous section was tailored to illustrate the effects of nonlinearity and dispersion. It was thus a little on the unreal-

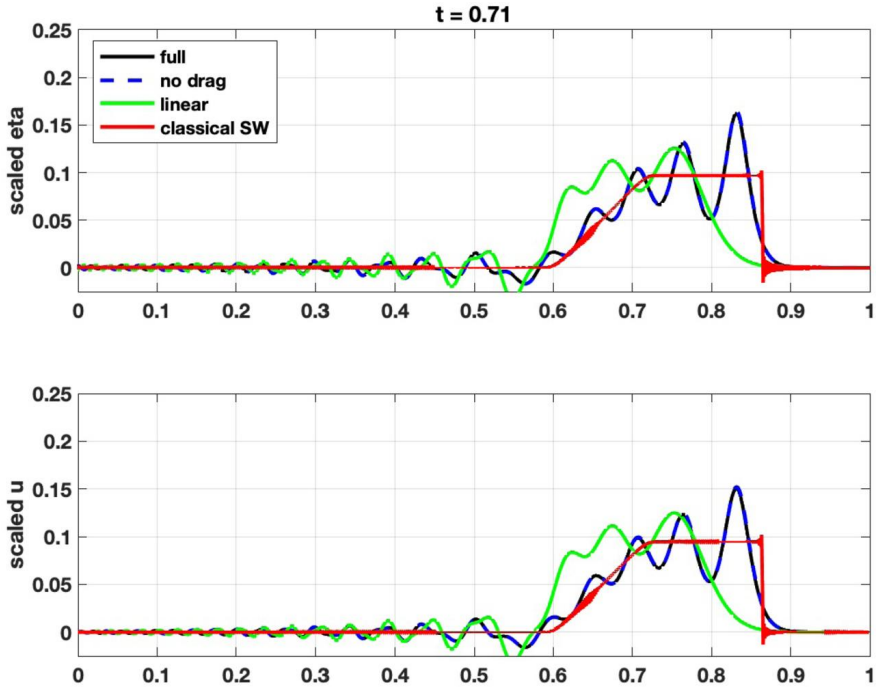

FIGURE 6.3

An adjustment problem for late dimensionless time ($t = 0.71$). Black – full model, blue – model without drag, green – linearized model, red – classical SW (no dispersion). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

istic side. In particular, the initial perturbation was rather large in amplitude (20% of the undisturbed depth) and the simulation was not run for a particularly long time.

Table 6.2 Parameters for the Field Scale Adjustment cases.

Label	N	L (m)	H (m)	g (m s^{-1})	η_0 (m)	f (s^{-1})
Long adjustment	65,536	2×10^5	10	0.0981	1	1×10^{-4}

In this section we discuss a suite of simulations that are tailored to the question "what would we actually expect to happen on the field scale". We repeat the set up from the previous section, but in a much larger domain (200 km for the rightward propagating wavetrain we concentrate on). We also decrease the amplitude of the initial perturbation to 10% of the undisturbed depth, and increase the extent of the smoothed region on the flanks to 2 km. See Table 6.2 for a complete list of parameter values.

We define a characteristic time as the time a linear wave would take to traverse half of the domain. This yields a value of just over 56 hours, implying that rotation may play a significant role in the dynamics.

We use a very large number of grid points, in order to successfully resolve all possible phenomena that our model equations represent: propagation, dispersion, nonlinearity, rotation, and damping by bottom stress. The horizontal grid resolution is just over 6 meters. While this is certainly a much finer resolution than many published lake modeling studies (we will discuss the reasons for this below), it is not a particularly difficult task for even a modern lap top. The relevant code is given in

```
swnh1d_large_adjust.m
```

with paremeters listed in Table 6.2.

Before we turn to the results of the numerical experiments it is useful to think about what information we could imagine gaining by running them. The easiest answer is that we will be able to conduct pairwise comparisons between cases with and without a particular physical mechanism (for example with and without the effect of the Earth's rotation). These differences could be quantitative in nature, but the hope is that if the differences are truly important, they will lead to qualitative differences. Since wave motions are likely part of what we will see, we can imagine comparing propagation speeds of any waves we see, and their shape. We will plot all three fields in the equations, the free surface height, η scaled by the undisturbed depth, and the two components of velocity u and v (the latter only activated in the presence of the Earth's rotation) both scaled by the shallow water wave speed $c_0 = \sqrt{gH}$. We chose to present a case of internal waves where the density jump across the light fluid-heavy fluid interface is one percent of the reference density.

Figs. 6.4, 6.5, and 6.6 show the state of the simulation for early, medium, and late times (measured by how much of the subdomain shown the waves have propagated across).

For early times it is already clear that the dominant qualitative difference is between cases with rotation (all colors but red) and without rotation (red). The second most obvious qualitative distinction is between the linearized equation case (green) and all other cases. All nonlinear cases have what appears to be a region of rapid oscillations. Indeed, the transition is so rapid, it looks like a thick pen had been used to draw the transition between the wave front and the undisturbed fluid ahead of it. With rotation, the front is followed by a long wave form extending from the origin to near $x = 0.34$. The u and v components of the velocity are out of phase for the long, rotation induced wave behind the leading transition. On the time scale of this figure, the case without bottom drag leads to waves with only a slightly larger amplitude.

Fig. 6.5 follows the evolution of the system, and while the qualitative distinctions from the above paragraph remain, there are two particularly noteworthy phenomena due to rotation that merit comment. The first of these is the large amplitude, long length scale free surface deflection near between $x = 0$ and about $x = 0.2$. This deflection is evident in both the nonlinear and linear systems (and on the scale of the image appears unaffected by bottom drag), and has the distinguishing feature that the u component of the velocity field is nearly zero, while the v component is large. This can be understood from the x component of geostrophic balance (5.3), repeated here for the reader's convenience:

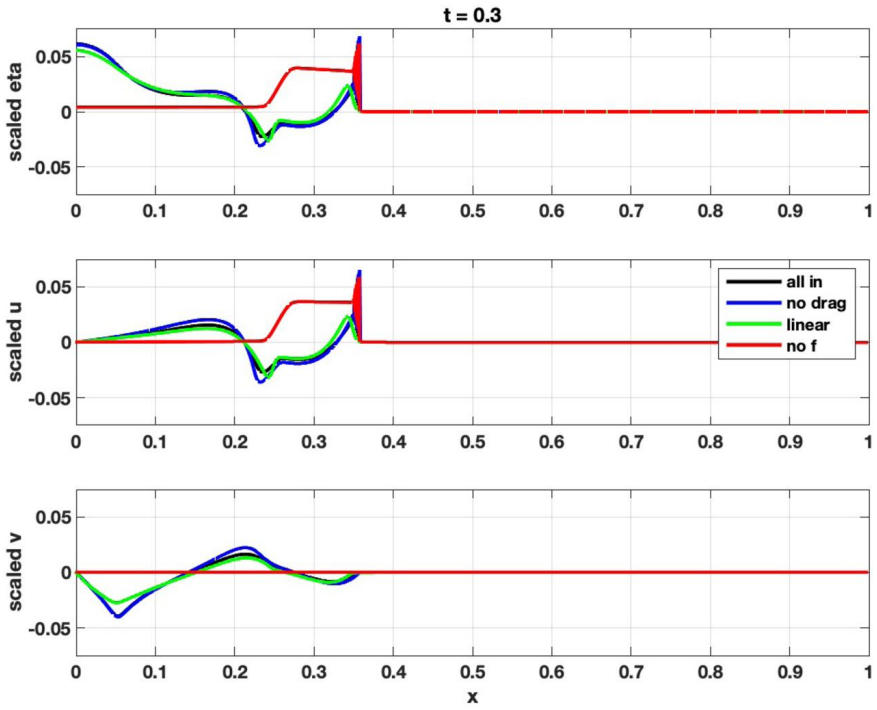

FIGURE 6.4

The Large domain adjustment problem for early dimensionless time ($t = 0.3$). Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

$$-fv = -g\frac{\partial \eta}{\partial x}.$$

It is clear that the v component of velocity is largely set by the rate of change of the free surface height with x. This is a good example how even a limited theory, like geostrophic balance, can provide useful insight.

The second phenomenon that is clear from the figure is that the no bottom drag case has evolved into a situation with a secondary wave train (just to the right of $x = 0.5$). This feature has a clear expression in the u field, but not in the v field. It is also not clearly observed in the case with bottom drag (black curve). This is the first illustration we have of the somewhat dubious effects of bottom drag.

Fig. 6.6 shows the evolving wave trains just as they near the right boundary of the domain. The geostrophic state near $x = 0$ remains largely unchanged for both the linearized and full equations. The leading and secondary wave train are also clearly evident, though at the scale of the half-domain little detail can be made out.

Figs. 6.7 and 6.8 focus on the leading and second wave train, respectively. It can be seen that for the leading wave train the case without bottom drag leads to a wave

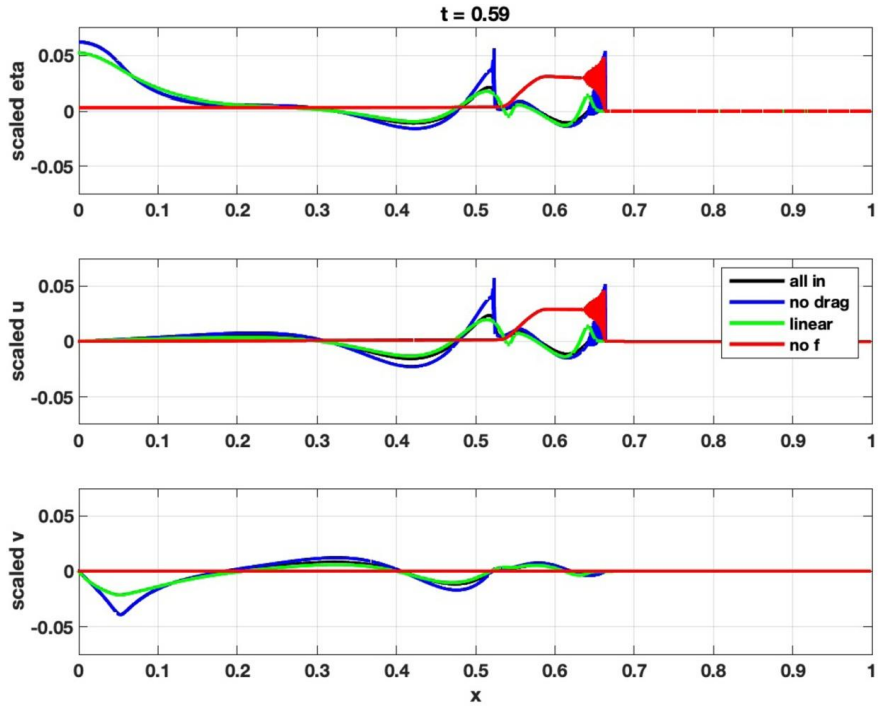

FIGURE 6.5

The Large domain adjustment problem for medium dimensionless time ($t = 0.59$). Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

train that is ahead of that for both the rotating and non-rotating cases with drag. The non-rotating case (red) corresponds to a classical undular bore with the free surface transitioning from its undisturbed value of zero, to a non-zero downstream value after passing through a long wave train. The rotating wave train shows a slightly longer inter-crest spacing and there is a clear modulation due to the long Poincaré waves that are difficult to see on a detailed plot like Fig. 6.7. The secondary wave train shown in Fig. 6.8 is perhaps even clearer in terms of story line. The most obvious point being that there is no secondary wave train in the non-rotating case, implying that its existence is 100% due to rotation. The role of bottom drag is also clear, since the black curve shows a much smaller amplitude wave train, and one that lags some way behind the blue (no bottom drag) wavetrain.

This is an impressive amount of information for a fairly straightforward simulation. In summary, it shows that for large lakes rotation leads both to large length scale motions, as well shorter length scale motions as seen in the secondary wave train. The possible implications for transport are also diverse:

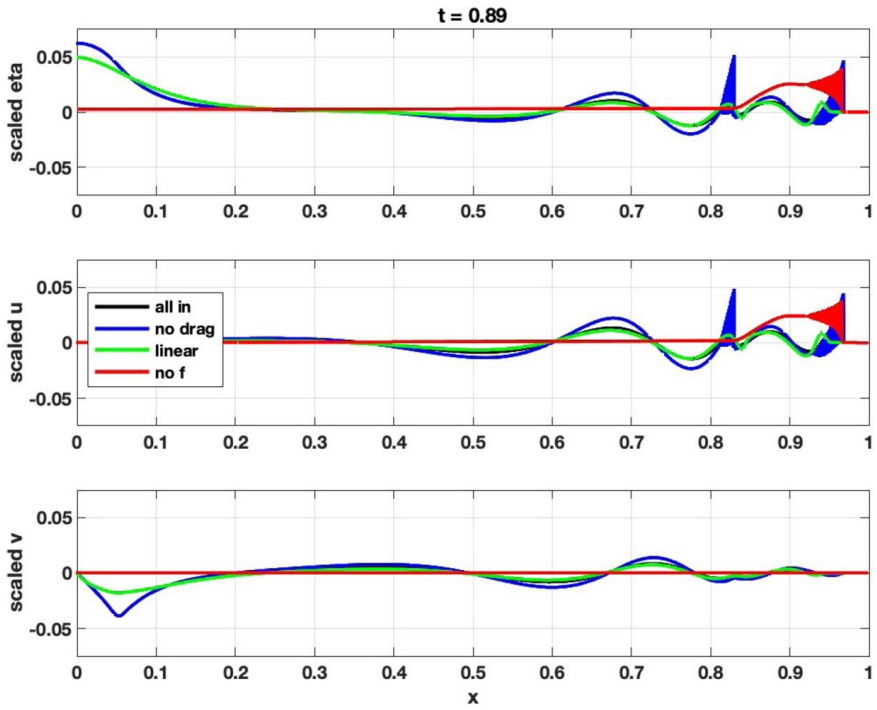

FIGURE 6.6

The Large domain adjustment problem for late dimensionless time ($t = 0.89$). Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

- The long waves, and the geostrophic state near $x = 0$ both induce transverse (i.e. v) currents which could transport tracers in a direction that is perpendicular to the primary direction of wave motion.
- The presence of dispersion modifies the expected behavior on small length scales, leading to long lived wave trains as opposed to dissipative shocks that might disappear before reaching a significant distance in x.
- The secondary wavetrains are affected by bottom drag, but could lead to a secondary "pulse" of transport in the x direction.

We took considerable care in making sure the simulations presented had sufficient resolution, and because the simple choice of a periodic domain allowed us to use FFT-based spectral methods, there was very little numerical dissipation. This is very different from many "off the shelf" lake models and so the primary purpose of this section is to encourage readers to consider what mechanisms may matter for their chosen problem PRIOR to choosing a computational tool. A poor choice may *a priori* filter out the very phenomenon that is responsible for transport!

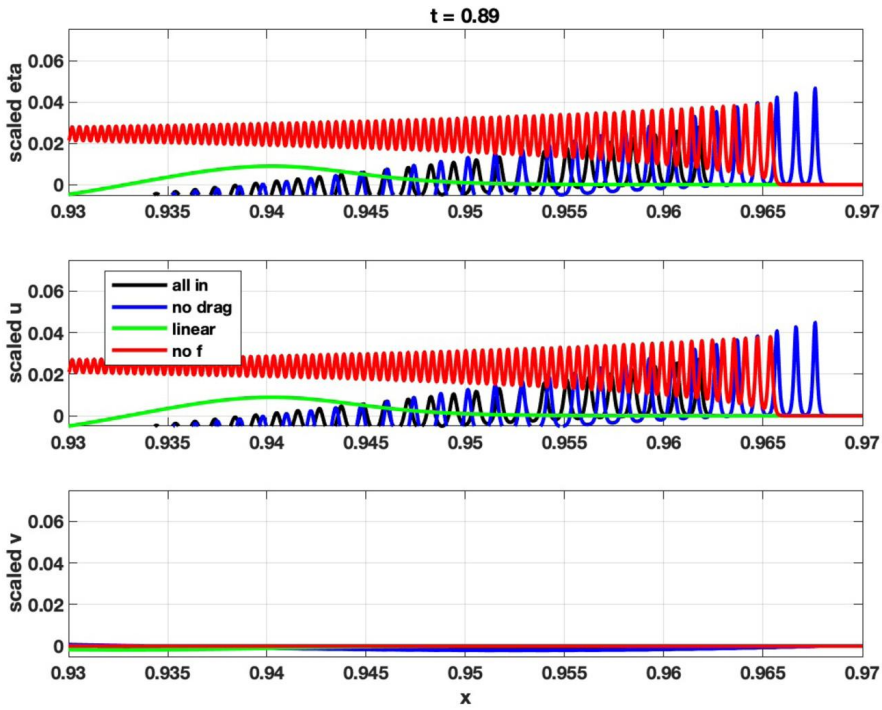

FIGURE 6.7

Detail of the Large domain adjustment problem for late dimensionless time ($t = 0.89$), focusing on the leading wavetrain. Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

6.4 Seiche evolution

While the initial conditions chosen in the last section are a standard tool of numerical methods, outside of engineering applications in which an actual lock in a canal is suddenly opened, they are some ways away from what typically occurs in the natural world.

The question of coming up with a simple, yet applicable set of initial conditions is actually quite difficult. Nevertheless, the observation that mid-to-large natural bodies of water experience oscillations called seiches has led to a considerable history of studying periodic, or nearly periodic, motions in lakes.

Table 6.3 Parameters for the Field Scale Adjustment cases.

Label	N	L (m)	H (m)	g (m s^{-1})	η_0 (m)	f (s^{-1})
Seiche breakdown	8192	1×10^4	10	0.0981	0.5	1×10^{-4}

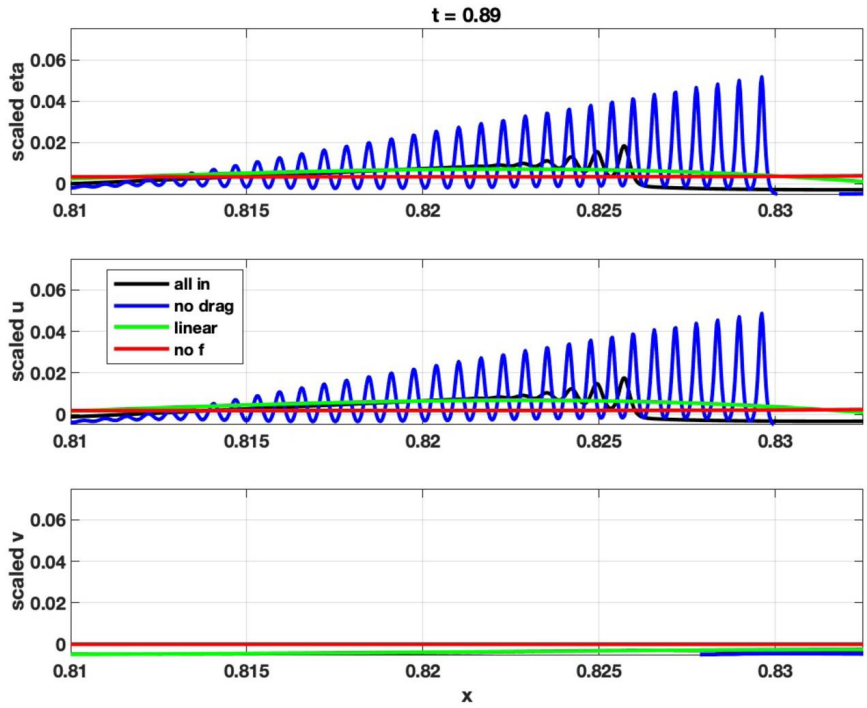

FIGURE 6.8

Detail of the Large domain adjustment problem for late dimensionless time ($t = 0.89$), focusing on the secondary wavetrain. Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

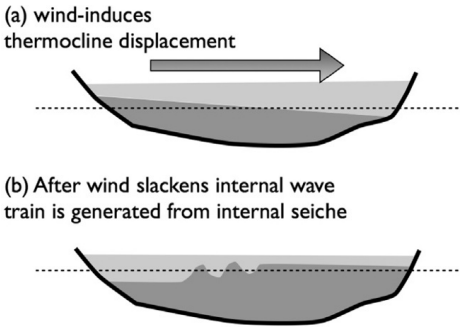

FIGURE 6.9

Schematic of wind-induced changes in a stratified lake, panel (a); and the development of internal wave trains when the wind slackens, panel (b).

In nature, it is believed that seiching behavior most often occurs when a sustained wind pushes water from the upwind to the downwind side of a lake (see the schematic

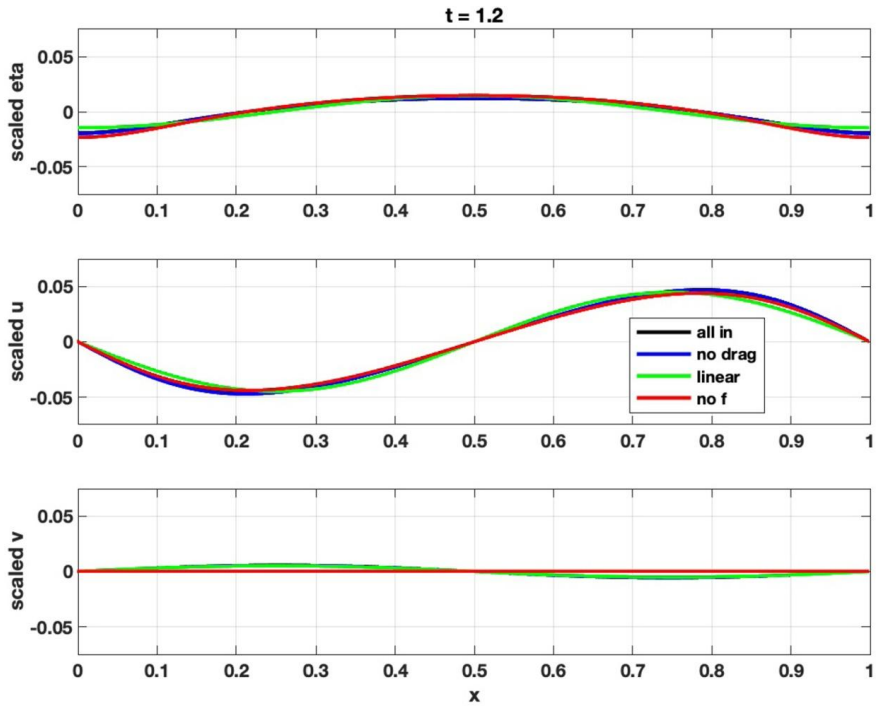

FIGURE 6.10

The seiche development problem for early dimensionless time ($t = 1.2$). Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

diagram in Fig. 6.9). This leads to an increase in water depth on the downwind side, and crucially, pushes the interface between warm and cold water (i.e. the thermocline) down. When the wind slackens a seiche is set up, with the internal seiche in the interior (where the density change across the thermocline is much smaller) far more prominent. Surface, or barotropic seiches can be important as well, and indeed are standard cottage owner lore along the beaches of the Laurentian Great Lakes in Canada. We will concentrate on internal seiches in this section.

We choose a domain that is large enough to be affected by rotation, but not as large as the lock release of the previous section. Our computational domain occupies $-10 < x < 10$ km, and we will concentrate the figures on $0 < x < 10$ km only. The internal seiche is defined to be a simple cosine. More realistic options will be discussed at the end of this section. The amplitude is taken as 5% of the total depth, which is itself taken to be constant (effectively treating the lake as an aquarium; again a mathematical convenience). The relevant code is given in

```
swnh1d_seiche.m
```

with parameters listed in Table 6.3.

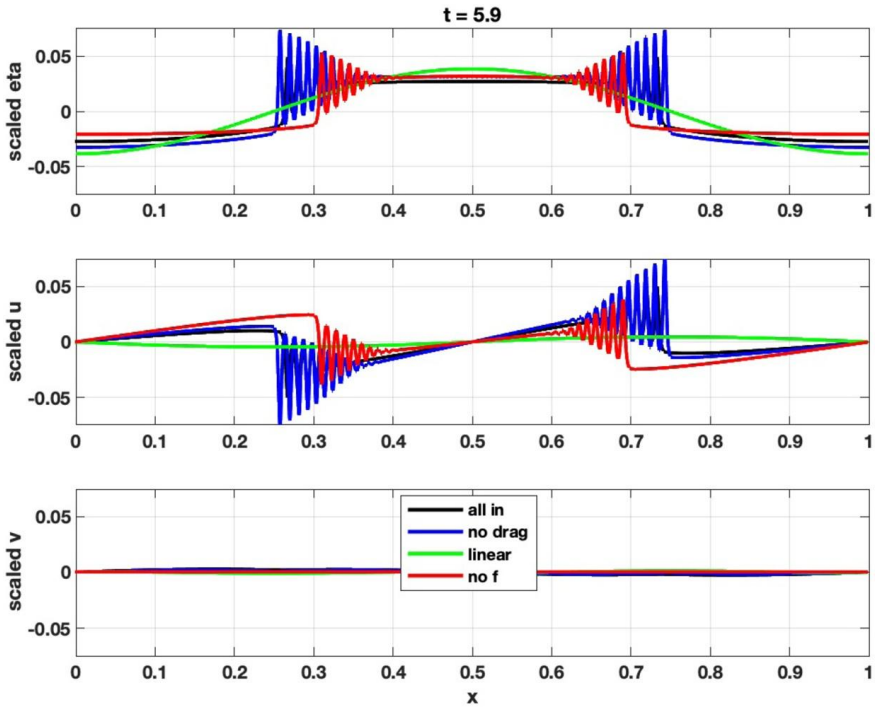

FIGURE 6.11

The seiche development problem for early dimensionless time ($t = 5.9$). Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

Figs. 6.10, 6.11, and 6.12 show the seiche development at three qualitatively different stages of development. The same convention as the previous section is used: the free surface, and both components of velocity are shown, scaled by the domain half-width and linear shallow water speed, respectively. Time is scaled by the advective time scale, or the time it would take a linear shallow water wave to propagate across the domain half width (i.e. $T_{adv} = L/c_0$).

Fig. 6.10 shows that all four numerical experiments behave quite similarly for early times. This means that for a study that only considers short times, it may well be that the bother of nonlinearity in particular, can be safely ignored.

However, Fig. 6.11 serves as an immediate cautionary tale. Here the linearized case (green curve) retains its sinusoidal form, while all the other cases take on a profoundly different shape. This shape is dominated by a wavetrain of rank-ordered "humps" similar to what we saw in the adjustment problem of the previous section. Careful examination shows that the cases with rotation (blue and black curves) lead to a train that is faster than the case without rotation (red curve). Moreover the inclusion of a quadratic bottom drag (black curve) leads to a smaller amplitude wavetrain, and one that lags behind the drag free case (blue curve).

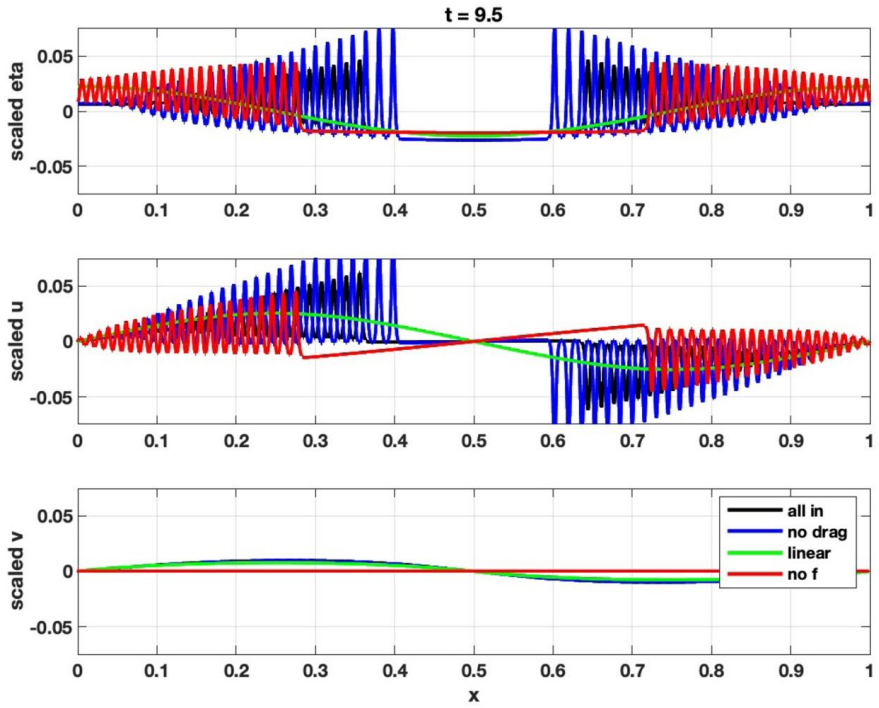

FIGURE 6.12

The seiche development problem for early dimensionless time ($t = 9.5$). Black – full model, blue – model without drag, green - linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

Fig. 6.12 shows that these wave trains fill larger and larger portions of the domain as time increases. The role of bottom drag at keeping amplitudes modest is quite clear in this figure. A reasonable question for the modeler is whether a drag term of the form $-c_d\vec{u}|\vec{u}|$ is sufficient to account for all the possible dissipation mechanisms in a real lake. This is a bit of a strawman argument, with "No" being the obvious answer. The discussion is somewhat irrelevant to most publicly available lake models, since these adopt eddy viscosity and diffusivity values so large that most features shown in our simulations would be smoothed out quickly. We will return to this discussion when we present the finite volume and finite element methods that are typically used by lake models in Chapters 9–11 of this book.

The previously discussed three figure set is useful for getting the broad strokes idea of how the seiche evolves. A few of the details merit some further discussion. In the very early stages, all four of the model types gave very similar results. While it is clear that eventually the four models are quantitatively different, one clear question is whether differences occur before or after the wavetrains form (except for the linear case which does not form wavetrains). Fig. 6.13 shows the four cases at dimensionless time $t = 3.6$. It is clear that wavetrains have not formed at this time, but all three

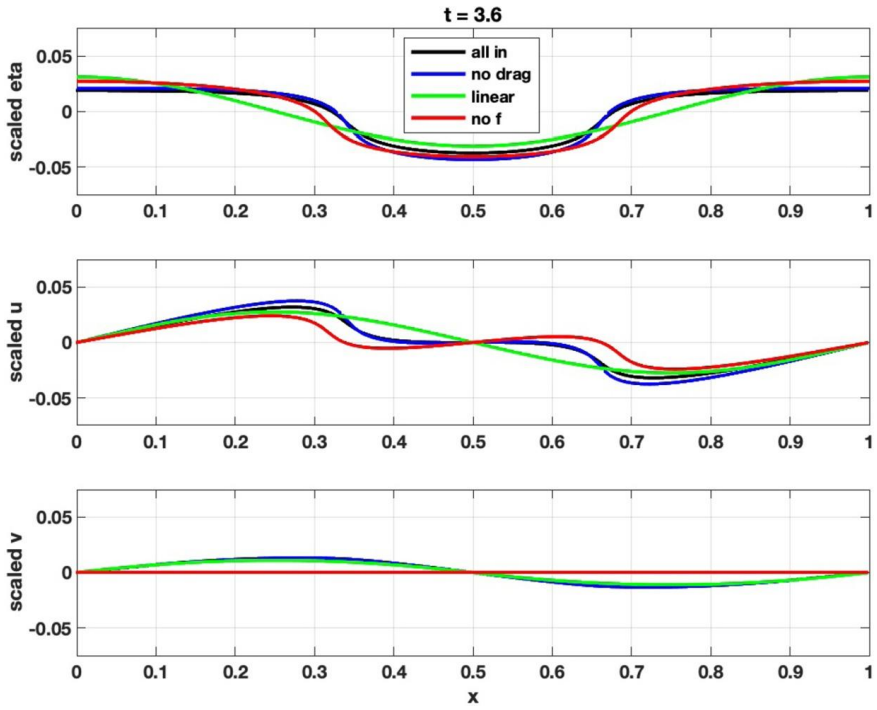

FIGURE 6.13

The seiche development problem showing differences prior to wavetrain formation; at dimensionless time $t = 3.6$. Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

of the nonlinear cases have led to a steepening of the initially sinusoidal waveform. The clearest difference is between the rotating (black and blue) and non-rotating case (red). The quadratic bottom drag (black) has led to a slight decay of the wave amplitude, perhaps clearest in the u field (middle panel).

These differences become considerably clearer when the wavetrains develop. Fig. 6.14 shows a detail of the wavetrains at $t = 5.9$. Again, the difference between the non-rotating (red) and rotating (blue and black) cases is clear. However, the effect of the quadratic damping is now pronounced with the leading crest of the wave train for the black curve reduced in amplitude by about 30% compared to the blue curve.

The seiche breakdown results provide some important takeaways as we build up to more complex cases. First of all, they reaffirm the need for resolution from the previous section. Second of all, they point to the ubiquity of nonlinear wave trains. In the case of seiches these fill the entire domain. This is partly due to the extreme simplicity of the 1D domain, and indeed for a lake with a complex lake shape (like that discussed in Chapter 11) we could expect considerably less focussing. Nevertheless, a form of wavetrains was observed for the round lake case shown in Fig. 5.4.

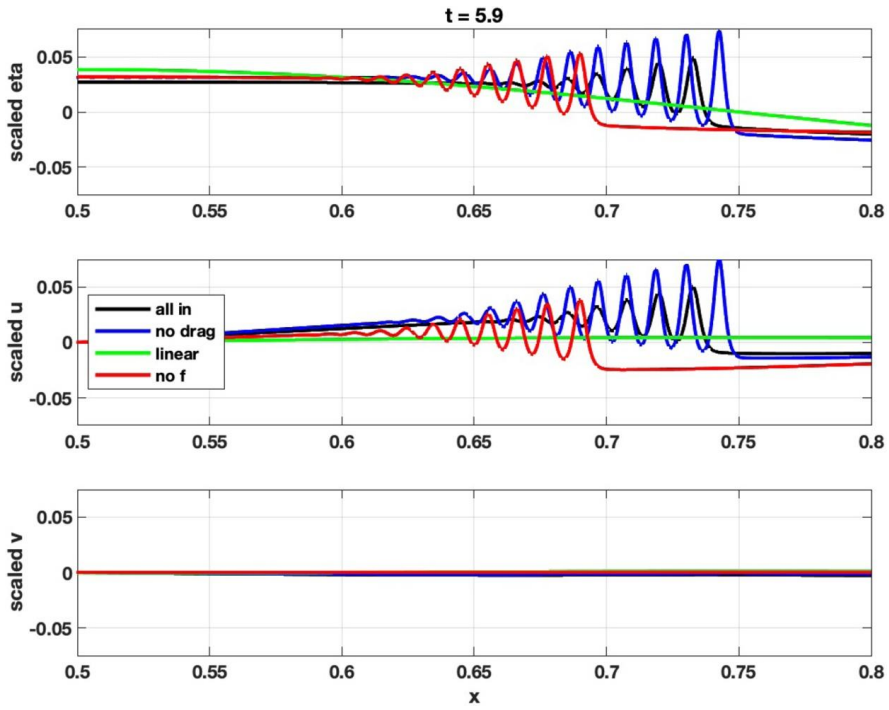

FIGURE 6.14

The detail of the wavetrain development at dimensionless time $t = 5.9$. Black – full model, blue – model without drag, green – linearized model, red – no rotation ($f = 0$). Upper panel – free surface scaled by undisturbed depth, Lower – panel u scaled by c_0.

Essentially all lake scale simulations involve some form of eddy viscosity, or some way to represent diffusion by unresolved motions. Since the models discussed so far have little inherent numerical dissipation, we can examine the issue of what we may lose if we set our eddy viscosity to be too high in detail. To do this we reconsider the seiche case from the previous examples, but now contrast the case with the bottom drag and several cases with eddy viscosity. The x-momentum equation without bottom drag and with eddy viscosity reads

$$u_t + uu_x = -g\eta_x + fv + \frac{H^2}{6}u_{xxt} + \nu_{eddy}u_{xx}.$$

ν_{eddy} is a "modeled" parameter, meaning that there is no first principles theory that specifies it. We present three values, $\nu_{eddy} = (0.1, 1, 10)$, which we label as small, medium, and large. The values do fall into the range used in some lake modeling studies, but the complexity and inherent numerical dissipation of full lake scale models, makes a quantitative comparison impossible. The relevant code including eddy viscosity is given in

```
swnh1d_weddyvisc.m
```

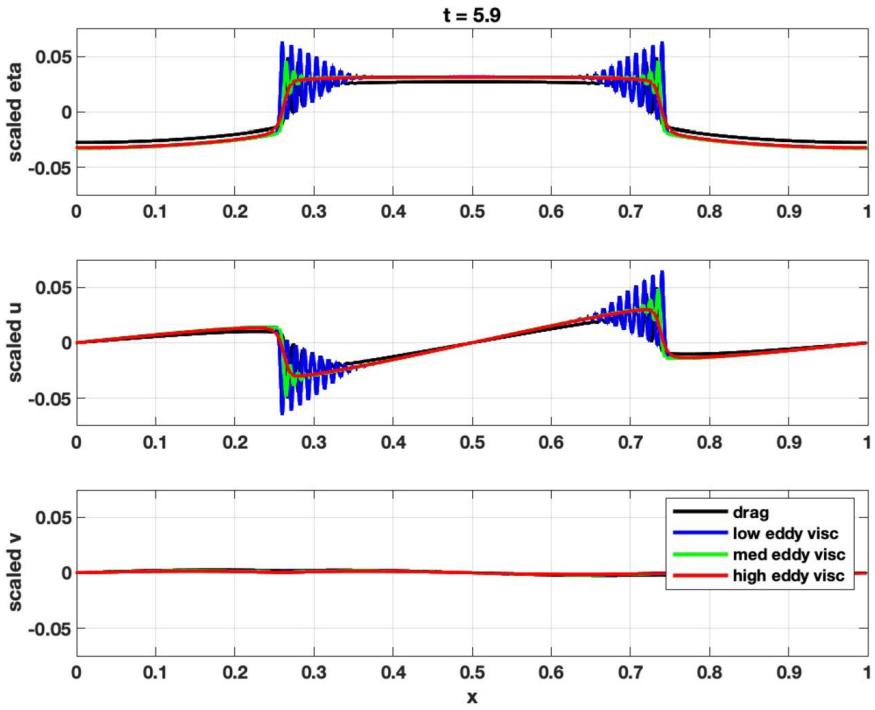

FIGURE 6.15

The seiche development problem for a dimensionless time ($t = 5.9$) at which wavetrains have developed. Black – full model, blue – $\nu_{eddy} = 0.1$, green – $\nu_{eddy} = 1$, red – $\nu_{eddy} = 10$. Upper panel – free surface scaled by undisturbed depth, Lowers – panel u and v scaled by c_0.

Figs. 6.15 and 6.16 show the seiche at dimensionless time $t = 5.9$ (the latter zooming in on the wavetrain in the range $0.65 < x < 0.775$). At this time the wavetrains have developed in most of the cases, though even on the scale of the whole domain, Fig. 6.15, differences are evident. Focusing on wavetrains we can note that changing the value of eddy viscosity qualitatively changes what is observed. The low value of viscosity (blue) has a clear wavetrain, with an amplitude that is actually even larger than the case with the bottom drag (black). The medium value of eddy viscosity (green) leads to a weaker wavetrain, with a much shorter horizontal extent. The high value of eddy viscosity (red) leads to no wavetrains at all.

While it is true that most lake scale models do not even attempt to include nonhydrostatic effects (though to be fair an increasing number of models have some nonhydrostatic capability), we have just shown that even if they did, users accustomed to high values of eddy viscosity may not observe any nonhydrostatic effects in their simulations. This situation is further muddled by the inherent numerical dissipation (essentially an invisible eddy viscosity) of many numerical schemes.

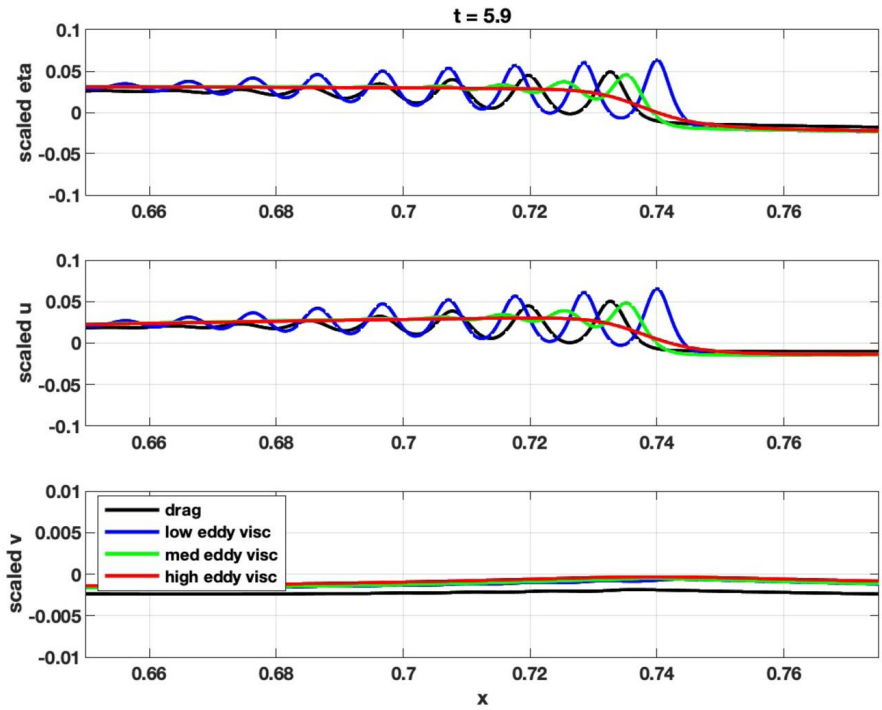

FIGURE 6.16

Detail of the wavetrains at dimensionless time ($t = 5.9$). Black – full model, blue – $\nu_{eddy} = 0.1$, green – $\nu_{eddy} = 1$, red – $\nu_{eddy} = 10$. Upper panel – free surface scaled by undisturbed depth, Lowers – panel u and v scaled by c_0.

6.5 Coding ideas

We have had considerable success with the above equations. This was due to a combination of good modeling choices (i.e. including dispersion) and good quality numerics.

In this section we discuss aspects of the numerical techniques we use. In order to keep the discussion simple we will drop the bottom stress terms, as well as rotation. Both can be included in the numerical method in a very simple way and the interested reader can see the details in the codes provided with this book.

We want to consider the equations (where subscripts are used to denote derivatives)

$$u_t = -uu_x - g\eta_x + \frac{H^2}{6}u_{txx} \tag{6.13}$$

$$\eta_t = -[(H+\eta)u]_x. \tag{6.14}$$

We will assume the bottom is flat (for convenience only) and make use of the Discrete Fourier Transform as implemented in the FFT. Mathematically the Fourier transform

is defined on the whole real line as

$$\mathcal{F}(u(x)) = \bar{u}(k) = \int_{-\infty}^{\infty} u(x)\exp(-ikx)dx$$

and the restriction to a domain $-L \leq x \leq x$ and functions that are periodic on this domain allows us to use the FFT. The mathematical details can be found in a number of excellent applied mathematics texts ([6,7]), though very few are directly relevant to the present, science first setting.

The one mathematical property of the Fourier transform that we do make use of is that it converts derivatives to a multiplication,

$$\mathcal{F}(u_x) = ik\mathcal{F}(u) = ik\bar{u}(k).$$

We can take the Fourier transform of the equations above, and rearrange the momentum equation so that all the time derivative terms are on the left hand side to find

$$\left(1 + k^2\frac{H^2}{6}\right)\bar{u}_t = -\overline{uu_x} - gik\bar{\eta} \tag{6.15}$$

$$\bar{\eta}_t = -Hik\bar{u} + ik\overline{\eta u}. \tag{6.16}$$

Recall that the overbar denotes a Fourier transformed variable. Next, we follow standard notation in numerical methods and write

$$u^{(n)} = u(x, n\Delta t).$$

This allows us to use standard finite differencing to approximate the time derivative. For example the forward Euler time stepping would write

$$u_t = \frac{u^{(n+1)} - u^{(n)}}{\Delta t} + O(\Delta t).$$

This is not a great choice, but a centered second order version, often called the leapfrog method works much better

$$\frac{u^{(n+1)} - u^{(n-1)}}{\Delta t} + O(\Delta t^2).$$

The leap frog method has a bit of a bad reputation in numerical analysis circles, owing to its strange instabilities when used in purely finite difference schemes. Here, however, we are using the FFT to compute spatial derivatives and the technique is very effective. We do one mathematically clever manipulation by noticing that the chain rule of calculus tells us that

$$(u^2/2)_x = uu_x$$

so that

$$\overline{uu_x} = \frac{ik}{2}\overline{u^2}.$$

If we also write the factor due to dispersion as

$$M(k) = \frac{1}{1 + k^2 \frac{H^2}{6}} \tag{6.17}$$

the discretization after Fourier transforming can be written as a simple recipe,

$$u^{(n+1)} = u^{(n-1)} - M(k)\left[\Delta t i k \overline{u^2}^{(n)} + 2\Delta t g i k \bar{\eta}^{(n)}\right] \tag{6.18}$$

$$\bar{\eta}^{(n+1)} = \bar{\eta}^{(n-1)} - Hik\bar{u}^{(n)} + ik\overline{\eta u}^{(n)}. \tag{6.19}$$

In practice after advancing, the $\bar{u}^{(n+1)}$ and $\bar{\eta}^{(n+1)}$ variables are transformed back to physical space using the inverse FFT. It is somewhat amazing that the dispersive correction is so simple to implement (a simple multiplicative a factor that does not change with time). This is partly due to the simplifying assumption of a flat bottom (i.e. Fourier methods are superb at constant coefficient problems), and partly due to the properties of the Fourier transform (that carry over to the FFT). We will see in Chapters 9–11 that other numerical methodologies require more sophisticated methods to implement the dispersive terms.

The perceptive reader will have noticed that the technique requires the u and η fields at the present time step and one time step in the past. This is problematic for the first time step, and usually a single explicit Euler step is taken so that one has the initial condition and the first time step, and can then apply the leapfrog method from that point on.

For readers who prefer a pseudocode listing we provide this in Algorithm 1 below.

Algorithm 1 Theoretical time stepping the dispersive SW equations.

Require: parameters and initial conditions
  COMPUTE the FFT of the ICs for the RHS of the PDEs
  CALCULATE one Euler forward step to get $\bar{u}^{(1)}$ and $\bar{\eta}^{(1)}$
  COMPUTE the inverse FFT to get $u^{(1)}$ and $\eta^{(1)}$
  DEFINE $u_{now} \leftarrow u^{(1)}$, $\eta_{now} \leftarrow \eta^{(1)}$, $u_{past} \leftarrow u^{(0)}$, and $\eta_{past} \leftarrow \eta^{(1)}$
  $n \leftarrow 1$
  while n is less than number of time steps **do**
    COMPUTE the FFT of $u_{now}, u_{past}, \eta_{now}, \eta_{past}$ for the RHS of the PDEs
    CALCULATE one leapfrog forward step to get $\bar{u}^{(n+1)}$ and $\bar{\eta}^{(n+1)}$
    COMPUTE the inverse FFT to get $u^{(n+1)}$ and $\eta^{(n+1)}$
    UPDATE $u_{past} \leftarrow u_{now}$, $\eta_{past} \leftarrow \eta_{now}$, $u_{now} \leftarrow u^{(n+1)}$, and $\eta_{now} \leftarrow \eta^{(n+1)}$
    $n \leftarrow n + 1$
  end while

In practice one typically wants to either store data or produce graphics at intermediate points and the algorithm is modified as in Algorithm 2.

Algorithm 2 Practical time stepping of the dispersive SW equations.

Require: parameters and initial conditions including number of outputs and inner steps
COMPUTE the FFT of the ICs for the RHS of the PDEs
CALCULATE one Euler forward step to get $\bar{u}^{(1)}$ and $\bar{\eta}^{(1)}$
COMPUTE the inverse FFT to get $u^{(1)}$ and $\eta^{(1)}$
DEFINE $u_{now} \leftarrow u^{(1)}$, $\eta_{now} \leftarrow \eta^{(1)}$, $u_{past} \leftarrow u^{(0)}$, and $\eta_{past} \leftarrow \eta^{(1)}$
$n \leftarrow 1$
while i is less than number of outputs **do**
  while j is less than number of inner time steps **do**
    COMPUTE the FFT of unow, upast, etanow, etapast for the RHS of the PDEs
    CALCULATE one leapfrog forward step to get $\bar{u}^{(n+1)}$ and $\bar{\eta}^{(n+1)}$
    COMPUTE the inverse FFT to get $u^{(n+1)}$ and $\eta^{(n+1)}$
    UPDATE $u_{past} \leftarrow u_{now}$, $\eta_{past} \leftarrow \eta_{now}$, $u_{now} \leftarrow u^{(n+1)}$, and $\eta_{now} \leftarrow \eta^{(n+1)}$
    $n \leftarrow n+1$, $j \leftarrow j+1$
  end while
  Store data or create graphical output
  $i \leftarrow i+1$, $j \leftarrow 1$
end while

The primary task of this section is to demonstrate to the reader that the same smart choices that make the mathematical modeling process successful can be applied to numerical methods. The code based on the above algorithms will not solve ALL problems, indeed it has some very clear limitations, but it will solve a subset of problems extremely efficiently. This efficiency then allows the user to concentrate on posing smart thought experiment using the code, as we did in the previous sections.

The code is also quite memory efficient, to the point where the implementation of the algorithms above evolved all the fields for the numerical experiments presented in the above sections at the same time. This is not necessary, but it does cut down on issues associated with data management. If we were presenting large two or even three dimensional simulations a different approach would likely be adopted, but for many "what if" numerical experiments, writing code that does your entire experiment leads to a far more efficient scientific computing process. Put another way, algorithms matter, but usability and user experience matter just as much.

6.6 Mini-projects

This chapter is somewhat exceptional in the sense that rather than pursuing problems beyond what is described, a reader who really wants to understand the material should carefully dissect the codes provided, and run them with their own changes to see how the results are modified.

We suggest starting with the small scale adjustment case, followed by the seiche case (with or without eddy viscosity). This is because the seiche case has a somewhat smaller number of points than the large scale adjustment case, and early explorations tend to go better when there is not a long wait time for results.

Examples of issues worth paying attention to are:

1. How are the nonlinear terms handled?
2. What is done to tame aliasing?
3. How are graphics handled?

Examples of things a beginning reader may change are to move the initial conditions, or change their functional form. A mid-level reader may want to compare quadratic and linear drag, or look at the spectra of the resulting fields. An expert reader may wish to change the method used for time-stepping or to optimize the number of FFTs used at each time step.

References

[1] P. Brandt, et al., Internal waves in the Strait of Messina studied by a numerical model and synthetic aperture radar images from ERS1/2 satellites, Journal of Physical Oceanography 27 (1997) 648–663.
[2] A. de la Fuente, et al., The evolution of internal waves in a rotating, stratified, circular basin and the influence of weakly nonlinear and nonhydrostatic accelerations, Limnology and Oceanography 53 (6) (2008) 2738–2748.
[3] P.K. Kundu, I.M. Cohen, Fluid Mechanics, 4th ed., Elsevier Academic Press, 2008.
[4] Clotilde Le Quiniou, et al., Copepod swimming activity and turbulence intensity: study in the Agiturb turbulence generator system, The European Physical Journal Plus 137 (2) (2022) 1–14.
[5] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[6] Brad G. Osgood, Lectures on the Fourier Transform and Its Applications, vol. 33, American Mathematical Soc., 2019.
[7] L.N. Trefethen, Spectral Methods in MATLAB, Society for Industrial and Applied Mathematics, 2000.
[8] G.B. Whitham, Linear and Nonlinear Waves, Wiley-Interscience, 1999.