Data-Assimilation in Shallow Water Flows

Introduction

Throughout the remainder of this paper, we focus on solving the free boundary value problem

$$\begin{aligned}
\Delta\phi &= 0, \quad -h < z < \eta(x,t)\\
\phi_z &= 0, \quad z = -h\\
\eta_t + \eta_x\phi_x - \phi_z &= 0, \quad z = \eta(x,t)\\
\phi_t + \frac{1}{2}\left|\nabla\phi\right|^2 + g\eta &= 0, \quad z = \eta(x,t),
\end{aligned}$$

where $\phi(x,z,t)$ is the velocity potential and the free fluid surface is given by $z = \eta(x,t)$. As pointed out in [1] and [2], the AFM/DNO method is best suited to shallow water environments. We therefore introduce the following non-dimensionalizations

$$\tilde{x} = \frac{x}{\lambda},\ \tilde{z} = \frac{z}{h},\ \tilde{t} = \frac{\sqrt{gh}}{\lambda}t,\ \eta = a\tilde{\eta},\ \phi = \frac{ag\lambda}{\sqrt{gh}}\tilde{\phi},$$

and non-dimensional parameters

$$\epsilon = \frac{a}{h},\ \mu = \frac{h}{\lambda},$$

whereby we can, after dropping tildes, rewrite the free surface problem in the non-dimesnional form

$$\begin{aligned}
\phi_{xx} + \frac{1}{\mu^2}\phi_{zz} &= 0, \quad -1 < z < \epsilon\eta(x,t),\\
\phi_z &= 0, \quad z = -1,\\
\eta_t + \epsilon\eta_x\phi_x - \frac{1}{\mu^2}\phi_z &= 0, \quad z = \epsilon\eta(x,t),\\
\phi_t + \frac{\epsilon}{2}\left(\phi_x^2 + \frac{1}{\mu^2}\phi_z^2\right) + \eta &= 0, \quad z = \epsilon\eta(x,t).
\end{aligned}$$

Here, ϵ is a measure of the characteristic wave amplitude, a, to the quiescent fluid depth, h, while μ is a measure of the quiescent fluid depth to the characteristic wavelength, λ, of the free surface waves. We will describe the fluid as being 'shallow-water' when ϵ and μ are both small. For the purposes of this report, we will choose $\epsilon = .1$ and $\mu = \sqrt{\epsilon}$, which corresponds to looking at meter high waves over ten meters of fluid, with characteristic wavelengths then on the order of thirty meters. Our choice of $\mu = \sqrt{\epsilon}$ is the one used to derive the Korteweg–de Vries equation, which is a classic shallow water model. In the following, we look at an approach whereby the free boundary value problem above becomes a closed system in terms of the surface variables $\eta(x,t)$ and $q(x,t)$ where the surface potential $q(x,t)$ is defined to be

$$q(x,t) = \phi(x, \epsilon\eta(x,t), t),$$

AFM/DNO Method

The AFM/DNO method begins from an observation first made in [3] that one can readily show that

$$\eta_t = \phi_z - \eta_x\phi_x = \sqrt{1+\epsilon^2\mu^2\eta_x^2}\,\partial_{\hat{\mathbf{n}}}\phi,$$

where $\partial_{\hat{\mathbf{n}}}\phi$ denotes the normal derivative of the velocity potential at the surface $z = \epsilon\eta(x,t)$. Then, following [3], we can define a Dirichlet-to-Neumann operator (DNO), say $G(\eta)$, whereby we can write

$$\sqrt{1+\epsilon^2\mu^2\eta_x^2}\,\partial_{\hat{\mathbf{n}}}\phi = G(\eta)q.$$

We can then derive the closed system of evolution equations for η and q of the form

$$\eta_t = G(\eta)q \tag{1}$$

$$q_t = -\eta - \frac{\epsilon}{2}q_x^2 + \frac{\epsilon\mu^2}{2}\frac{\left(G(\eta)q + \epsilon\eta_x q_x\right)^2}{1+\epsilon^2\mu^2\eta_x^2} \tag{2}$$

Of course, to make this approach useful, one must determine a computable form of the DNO. To do so, following arugments in [4], we use the fact that G is an analytic function of η so that

$$G(\eta)q = \sum_{j=0}^{M}\epsilon^j G_j(\eta)q + \mathcal{O}(\epsilon^{M+1}).$$

While there are several ways to determine the terms in this expansion, we argue that using the AFM formulation of the free surface wave problem makes this process more straightforward than other approaches. Following then the results in [5], we derive the following integro-differential equation

$$\int_{-L}^{L} e^{-ikx}\left(\cosh(\mu\tilde{k}(1+\epsilon\eta))\eta_t + \frac{iq_x}{\mu}\sinh(\mu\tilde{k}(1+\epsilon\eta))\right)dx = 0,\ \tilde{k} = \frac{\pi k}{L}.$$

Inserting the DNO expansion from above, expanding the hyperbolic trignometric terms, and then matching powers of ϵ provides the following recursive formulas for the coefficients of the DNO, where for $j \geq 1$,

$$\widehat{(G_j q)}_k = -\sum_{m=0}^{j-1} \frac{(\mu\tilde{k})^{(j-m)}}{(j-m)!} L_{j-m} \widehat{\left(\eta^{(j-m)} G_m q\right)}_k - \frac{i}{\mu}\tilde{k}^j L_{j+1} \widehat{\left(\eta^j q_x\right)}_k, \tag{3}$$

with

$$\widehat{(G_0 q)}_k = \frac{\tilde{k}}{\mu} \tanh(\mu\tilde{k})\hat{q}_k,$$

and

$$L_j(k) = \begin{cases} 1, & j \text{ even}, \\ \tanh(\mu\tilde{k}), & j \text{ odd}. \end{cases}$$

The Ensemble Kalman Filter

The Linear Kalman Filter

We briefly review the now classic and widely used linear Kalman filter before discussing its nonlinear variant, the ensemble Kalman filter. The linear Kalman filter supposes that some true state of a system, described by the time dependent N_m-dimensional vector at time t_n, say $x_n^{(tr)}$, is given via the model

$$x_n^{(tr)} = Mx_{n-1}^{(tr)} + \epsilon_{n-1}^{(m)},$$

where M is a model matrix which attempts to transform the prior true state into the next true state. However, we suppose that the model is not complete to describe the true state, and that model deficencies manifest themselves via the error vector ϵ_{n-1}. The error vectors are assumed to consist of zero-mean Gaussian random variables with the further assumption that

$$\mathbb{E}\left[\epsilon_j^{(m)} \left(\epsilon_k^{(m)}\right)^T\right] = \delta_{jk}Q_k,$$

which is to say that the errors are uncorrelated in time, while at a given time step there is one covariance matrix to typefy the deviation among the model errors. Likewise, we suppose that at time t_n, we have an N_d-dimensional vector of measurements given by d_n which corresponds to the true state via the equation

$$d_n = Hx_n^{(tr)} + \epsilon_n^{(d)},$$

where H is the measurment matrix and the errors in measurement are incorporated into the zero-mean, Gaussian vector $\epsilon_n^{(d)}$, which we further suppose satisfies the identity

$$\mathbb{E}\left[\epsilon_j^{(d)} \left(\epsilon_k^{(d)}\right)^T\right] = \delta_{jk}R_k, \ \mathbb{E}\left[\epsilon_j^{(m)} \left(\epsilon_k^{(d)}\right)^T\right] = 0.$$

We now suppose that, in effect, we have no access to the true state. However, we suppose that at time t_n, through some mechanism, we have a N_m-dimensional *forecast*, say $x_n^{(f)}$. We can then define the error covariance matrix

$$P_n^{(f)} = \mathbb{E}\left[\left(x_n^{(f)} - x_n^{(tr)}\right)\left(x_n^{(f)} - x_n^{(tr)}\right)^T\right].$$

We then ask, given a forecast, can we incorporate the data in such a way to create a better estimate to the true state? To do this, we suppose that this better *analysis* state, say $x_n^{(a)}$ is given by

$$x_n^{(a)} = x_n^{(f)} + K_n(d_n - Hx_n^{(f)}),$$

where the gain matrix K_n is chosen so as to minimize the trace of the affiliated error covariance matrix $P_n^{(a)}$ where

$$P_n^{(a)} = \mathbb{E}\left[\left(x_n^{(a)} - x_n^{(tr)}\right)\left(x_n^{(a)} - x_n^{(tr)}\right)^T\right].$$

Thus, the analysis vector represents the best estimate to the true state using the given forecast and measurements of the system. As can be shown, K_n is then given by

$$K_n = P_n^{(f)}H^T\left(HP_n^{(f)}H^T + R_n\right)^{-1}.$$

This optimization procedure likewise gives us the affiliated analysis error covariance matrix $P_n^{(a)}$ where

$$P_n^{(a)} = (I - K_nH)P_n^{(f)}.$$

At this point, we note that with a forecast estimate $x_n^{(f)}$ and error covariance matrix $P_n^{(f)}$, we were able to create a best estimate incorporating the data obtained via measurement to produce the analysis estimate $x_n^{(a)}$ with the corresponding error covariance matrix $P_n^{(a)}$. To go to the next time step then, we let

$$x_{n+1}^{(f)} = Mx_n^{(a)}, \; P_{n+1}^{(f)} = MP_n^{(a)}M^T + Q_n.$$

Note the recursive structure of the Kalman filter, whereby information from only the immediately previous time-step is necessary to progress to the next time step.

The Ensemble Kalman Filter

As noted, the above process is well-defined when modeling linear systems. In the case though that the model is nonlinear, i.e.

$$x_n^{(tr)} = f\left(x_{n-1}^{(tr)}\right) + \epsilon_{n-1}^{(m)},$$

then more thought is necessary in order to perform the filtration. In particular, we see that the update step associated with generating the forecast error covariance matrix is no longer applicable. While one could replace M with linearizations around the computed analysis estimate, as in the Extended Kalman Filter, this has proved to be both computationally costly and unstable with respect to error.

Instead, we introduce the idea of an ensemble of say N_e forecasts in which now $x_n^{(f)}$ denotes an $N_m \times N_e$ matrix which generates the ensemble average, $\bar{x}_n^{(f)}$ where

$$\bar{x}_n^{(f)} = \frac{1}{N_e} \sum_{j=1}^{N_e} x_{n,j}^{(f)}.$$

Using this average as an approximation to the true state, we can likewise generate an estimate to the error covariance matrix $P_n^{(f)}$ via the formula

$$P_n^{(f)} \approx \frac{1}{N_e - 1} \sum_{j=1}^{N_e} \left(x_{n,j}^{(f)} - \bar{x}_n^{(f)} \right) \left(x_{n,j}^{(f)} - \bar{x}_n^{(f)} \right)^T. \tag{4}$$

Correspondingly, given the data vector d_n, we build a corresponding ensemble represented by the data matrix $d_n^{(e)}$ whose j^{th}-column is given by

$$d_{n,l}^{(e)} = d_n + \tilde{\epsilon}_{n,l}, \ l = 1 \cdots N_e,$$

where the vectors $\tilde{\epsilon}_{n,j}$ are zero-mean Gaussian random vectors such that

$$\lim_{N_e \to \infty} \frac{1}{N_e - 1} \sum_{l=1}^{N_e} \tilde{\epsilon}_{j,l} \tilde{\epsilon}_{k,l}^T = \delta_{jk} R_k.$$

In the Ensemble Kalman Filter, the gain matrix K_n is still given by the formula above, but with the error covariance matrices given via Equation (4). This then allows for an analysis state, say $x_n^{(a)}$ ensemble-matrix approximation to be computed which is consistent with the classic Kalman filter in the linear, infinite ensemble number limit. To complete the recursive cycle then, we find the next ensemble-forecast matrix via

$$x_{n+1}^{(f)} = f\left(x_n^{(a)} \right).$$

Averages and approximate error covariance matrices are computed as above, and the Ensemble Kalman Filter cycle is continued.

Eularian Data Assimilation

In contrast to the Lagrangian approach, we focus on using Eularian measurements as the source of the data used in the Kalman filter. Given that most wave height measurements are obtained via pressure plates, which corresponds to an Eularian measurement, this makes an Eularian data assimilation approach more natural than a Lagrangian one. To do this, we suppose that we have at some set of fixed locations, say $x_j^{(p)}$, $j = 1 \cdots N_f$, pressure plates over which we obtain surface height measurements $\eta(x_j^{(p)}, t) + \epsilon_j^{(d)}(t)$. We likewise, using standard pseudo-spectral methods for solving the water-wave problem, have at any given time a numerically computed approximation to the surface, say $\tilde{\eta}(x, t)$, where

$$\tilde{\eta}(x,t) = -\frac{1}{2K}\sum_{j=-K+1}^{K} \hat{\eta}_j e^{i\pi jx/L_x}, \ x \in [-L_x, L_x].$$

Note, the scaling and sign choice are commensurate with the standard implementation of the Fast Fourier Transform in Matlab, NumPy, and so forth. Given the requirement that $\tilde{\eta}(x,t)$ is real, we have the symmetry constraint $\hat{\eta}_{-j} = \hat{\eta}_j^*$. For ease then, we set $\hat{\eta}_K = 0$, and note that the above representation for $\tilde{\eta}(x,t)$ can be rewritten as

$$\tilde{\eta}(x,t) = -\frac{1}{2K}\left(\hat{\eta}_0 + 2\sum_{j=1}^{K-1}\left(\mathrm{Re}(\hat{\eta}_j)\cos\left(\frac{\pi jx}{L_x}\right) - \mathrm{Im}(\hat{\eta}_j)\sin\left(\frac{\pi jx}{L_x}\right)\right)\right),$$

We therefore see that when comparing data to model forecasts, the observation operator $H = -\tilde{H}/K$, where $\tilde{H}$ is given by the Vandermonde like matrix

$$\tilde{H} = \begin{pmatrix} 1/2 & c_{1,1} & \cdots & c_{1,(K-1)} & s_{1,1} & \cdots & s_{1,(K-1)} \\ 1/2 & c_{2,1} & \cdots & c_{2,(K-1)} & s_{2,1} & \cdots & s_{2,(K-1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1/2 & c_{N_f,1} & \cdots & c_{N_f,(K-1)} & s_{N_f,1} & \cdots & s_{N_f,(K-1)} \end{pmatrix}$$

where

$$c_{j,l} = \cos(l\tilde{x}_j), \ s_{j,l} = \sin(l\tilde{x}_j)$$

with $\tilde{x}_j = \pi x_j^{(p)}/L_x$. Throughout the remainder of this section, we take the covariance matrices Q_n and R_n to be $\sigma^2 I$, where I denotes the identity matrix appropriate to the corresponding dimensionality of the random variable in question, and σ corresponds to the variance of the associated Gaussian random variables.

In a similar vein, we also have to decide how to initialize the filter. In real-world conditions, we would generally be obliged to generate real world forecasts from existing buoy data as opposed to being able to capture a sea state via say acoustic imaging done from a boat. Thus, we suppose that at time $t = 0$, we have some set of pressure plate measurements giving us the initial surface heights $\eta(x_j^p, 0) + \epsilon_j^{(d)}$. By interpolating and transforming this data, we can initialize the filter with respect to the surface height. However, this leaves the determination of the initial surface velocity potential $q(x,0)$ in question. It is worth noting that at best, one might hope to find a derivative of this quantity using the identity

$$q_x = \phi_x + \epsilon\eta_x\phi_z = \dot{x} + \epsilon\eta_x\dot{z},$$

so this is something that would still only be determined up to a constant. Perhaps since the buoy is tethered one can possibly estimate some of the relevant quantities, but suffice to say, it is not immediately clear how one would determine an initial estimate of this variable. Thus, without any other information, we initialize the surface velocity potential with zero-mean, σ-variance Gaussian noise. Thus, we effectively look at a wide range of flow fields, limited only by the requirements on the variance and mean. Note, in order then to get meaningful results, this inidicates that we should let the number of ensembles $N_e \gg 1$, which is already in accord with best practice.

Lagrangian Data Assimilation

While similar to the framework described above, following the framework advanced by [6, 7], in order to incorporate sea height measurments provided by freely floating buoys, we augment the equations describing the free surface flow derived above by the differential equations describing particle paths at the free surface, which are given by

$$\frac{dx}{dt} = \frac{\epsilon\left(q_x - \epsilon\mu^2\eta_x\eta_t\right)}{1+\epsilon^2\mu^2\eta_x^2}, \ \frac{dz}{dt} = \frac{\epsilon\left(\eta_t + \epsilon\eta_x q_x\right)}{1+\epsilon^2\mu^2\eta_x^2}.$$

If we then suppose that we have, at time $t = 0$, N_f buoys with initial positions say $(x_j^{(f)}(0), \epsilon\eta(x_j^{(f)}(0), 0) + \epsilon_j^{(d)}(0)$, $j = 1 \cdots N_f$. As before, we use an interpolation of this data as an initial condition for the filter. In cotrast though to the Eularian framework, in which it is more natural to treat the ensemble forecast and analsyis ensembles in frequency space, the Lagrangian framework lends itself to having the forecast and analysis ensembles be given in physical space. This greatly simplifies the observation operator which becomes

$$H = \left(0_{2N_f\times 2K} \ I_{2N_f\times 2N_f}\right),$$

where, equivalent to the role of K above, we think instead of sampling η and q at the points

$$x_j = -L_x + \frac{j}{K}L_x, \ j = 0, \cdots, 2K-1.$$

Data Assimilation

We now present results. For the true flow, we take $L_x = 10$

$$\eta(x,0) = \cos(\tilde{x}),\ q(x,0) = \sin(\tilde{x}),\ \tilde{x} = \frac{\pi}{L_x}x,$$

and use a pseudo-spectral scheme with $K_T = 256$ modes and a 4th-order Runge–Kutta scheme with an integrating factor and time step $dt = 10^{-2}$. We take $\epsilon = .1$ and $\mu = \sqrt{\epsilon}$. The number of terms used in the DNO expansion is $M = 14$, which generally ensures machine precision accuracy for relatively long time evolutions. As for how we assimilate data, we set the variance $\sigma = .1$, and the number of ensemble members $N_e = 3200$. We truncate the DNO expansions at both $M = 1$ and $M = 14$ to examine the impact of including higher order nonlinearity in the models and how that affects the data analysis process. We take a sampling rate in time of $dt_s = .5$, which is 50 times that of the time step of the full numerical solver. Likewise, we use a 4th-order Runge–Kutta scheme with an integrating factor in order to perform the analysis to forecast update. We then run the simulation to $t_f = 20$, which given the choice of ϵ, is a long enough period of time for nonlinear effects to create significant distortions of the original wave profile.

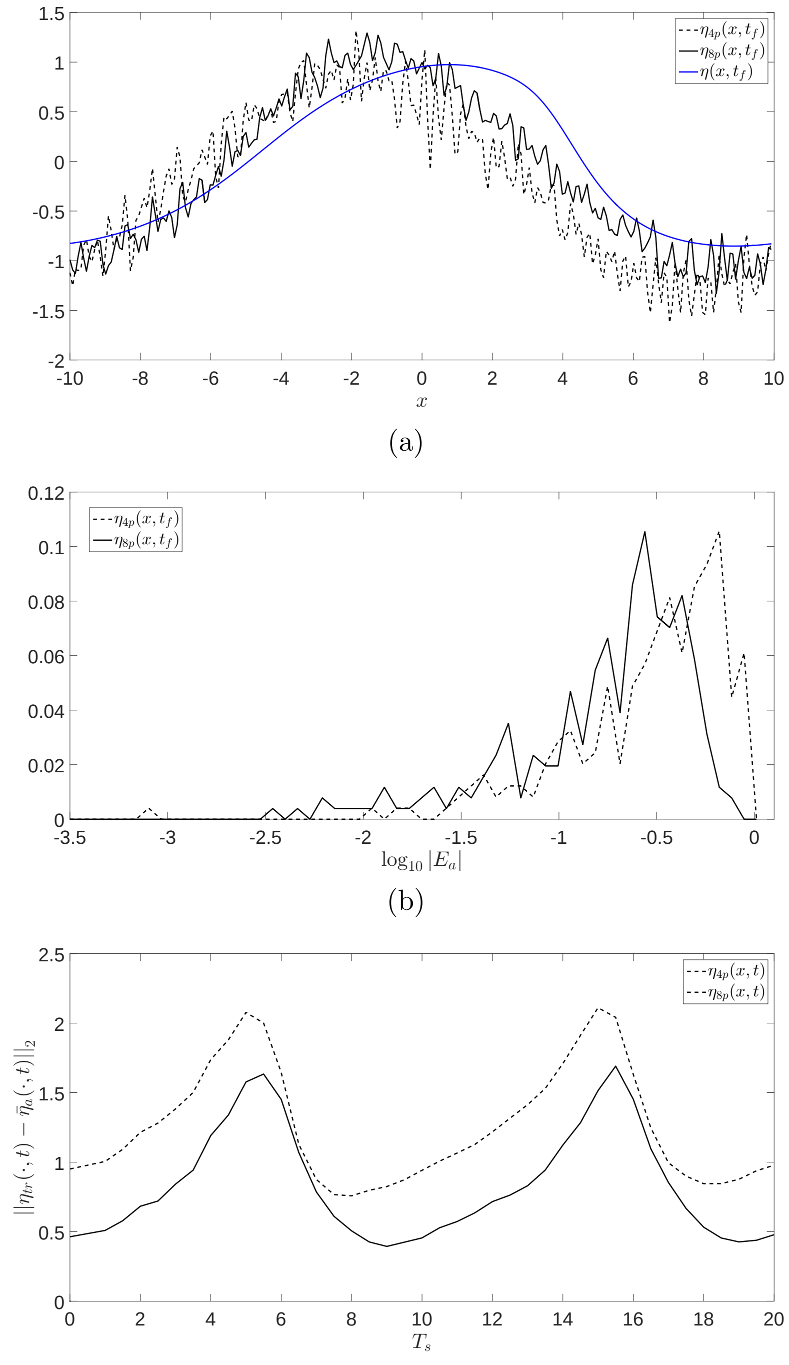
Eularian Data Assimilation Results

In order to provide comparisons, we then look at two simulations, one in which we have four equally spaced pressure plates in the domain $[-10, 10]$, and one in which we have eight. Choosing $M = 1$ first, we see the results in Figure 4. In Figure 4 (a) we compare the pointwise approximations generated by using four and eight pressure plate measurements compared to the true profile, $\eta_{tr}(x, t_f)$. In Figure 4 (b), we plot a histogram of $\log_{10} |\eta_a(x,t) - \eta_{tr}(x,t)|$, where $\eta_a(x,t)$ comes from either the four plate or eight plate measurements. Finally, we look at the relative error,

$$\left(\frac{\sum_{j=0}^{2K-1} \left(\eta_{tr}(x_j,t) - \eta_a(x_j,t)\right)^2}{\sum_{j=0}^{2K-1} \eta_{tr}^2(x_j,t)}\right)^{1/2},\ x_j = -L_x + j\frac{L_x}{K},$$

at each sampling time in Figure 4 (c). As one would expect, we get markedly better estimates by using twice the number of pressure plate based height measurements.

We then look at the impact of including higher-order nonlinearity by setting $M = 14$. We see the results of this in Figure 2. As can be seen, while there is some tendency to reduce error; see Figures 2 (b) and (c), including this number of terms does not materially improve the filtration process in any significant way. This is an intriguing result hinting that while higher order effects may be critical for detailed, pointwise understandings of a process, they may essentially be so overwhelmed by noise as to be essentially



(a)

(b)

(c)

Figure 1: For $M = 1$, $dt_s = .5$, with four plates (–) compared to eight (-), we have the wave profiles at $t_f = 20$ (a), a histogram of the log of the pointwise error (b), and the root-square error at every sampling time (c).

useless in data assimilation algorithms. Of course, another possibility is that the ensemble Kalman filter is so poorly adapted to the degree of nonlinearity present that it is incapable of providing significant improvements via the incorporation of data. Finally, we may also just be paying a fundamental price for not providing any data about the velocity field at the surface.

Lagrangian Data Assimilation Results

References

[1] D.G. Dommermuth and D.K.P. Yue. A high-order spectral method for the study of nonlinear gravity waves. *J. Fluid Mech.*, 184:267–288, 1987.

[2] J. Wilkening and V. Vasan. Comparison of five methods of computing the Dirichlet–Neumann operator for the water wave problem. In *Nonlinear Wave Equations: Analytic and Computational Techniques.* AMS, 2015.

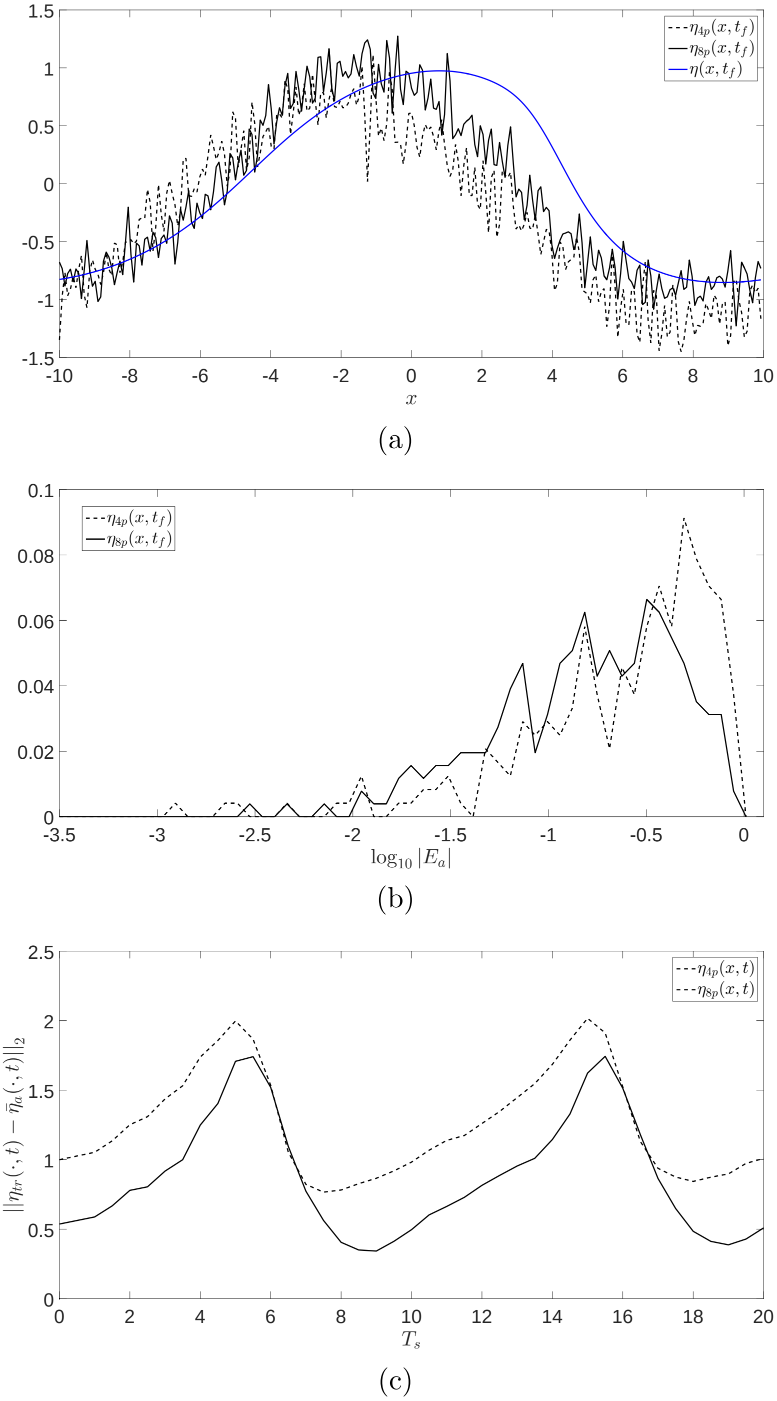
[3] V.E. Zakharov. Stability of periodic waves of finite amplitude on the surface of a deep fluid. *Zhurnal Prikladnoi Mechaniki i Teckhnicheskoi Fiziki*, 8:86–94, 1968.

[4] W. Craig and C. Sulem. Numerical simulation of gravity waves. *J. Comput. Phys.*, 108:73–83, 1993.

[5] M.J. Ablowitz, A.S. Fokas, and Z.H. Musslimani. On a new non-local formulation of water waves. *J. Fluid Mech.*, 562:313–343, 2006.

[6] L. Kuznetsov, K. Ide, and C.K.R.T. Jones. A method for assimilation of Lagrangian data. *Mon. Wea. Rev.*, 131:2247–2260, 2003.

[7] H. Salman, L. Kuznetsov, and C.K.R.T. Jones. A method for assimilating Lagrangian data into a shallow-water-equation ocean model. *Mon. Wea. Rev.*, 134:1081–1101, 2006.

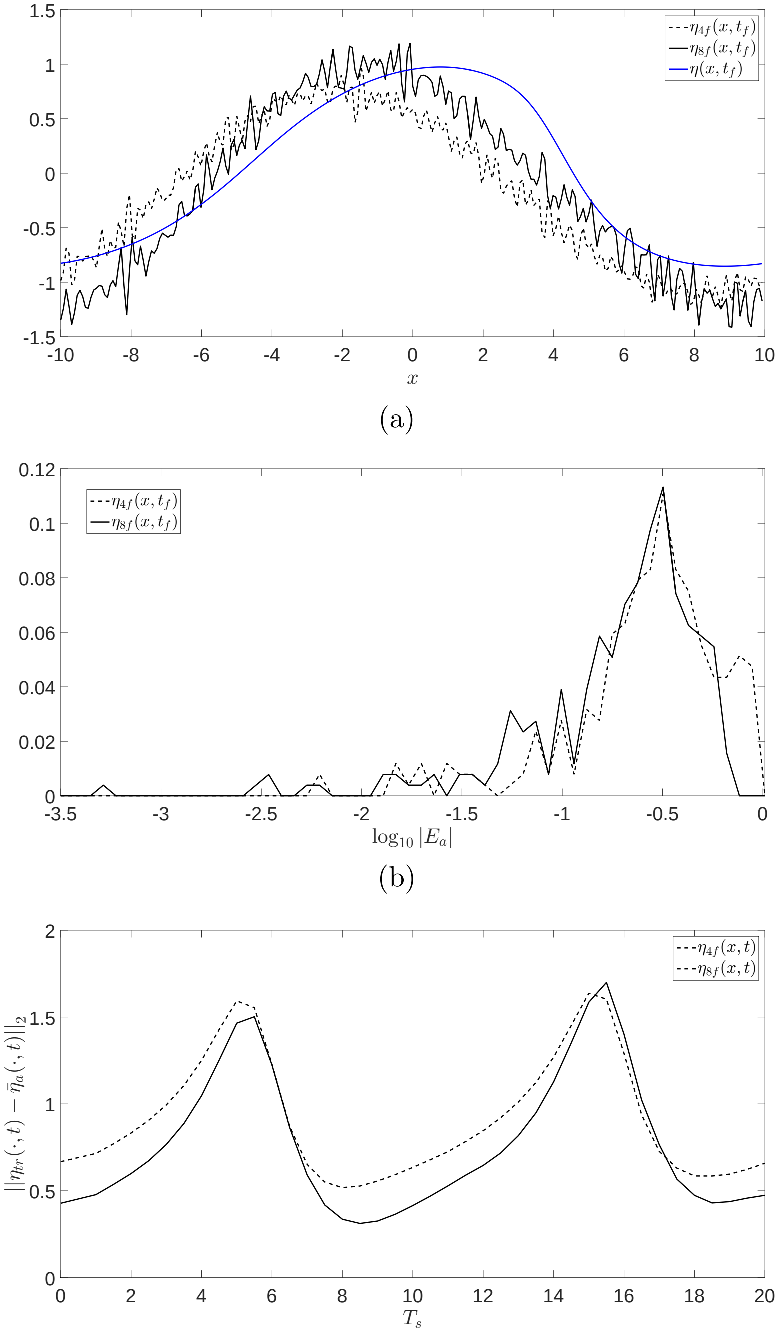


(a)

(b)

(c)

Figure 2: For $M = 14$, $dt_s = .5$, with four plates (–) compared to eight (-), we have the wave profiles at $t_f = 20$ (a), a histogram of the log of the pointwise error (b), and the root-square error at every sampling time (c).

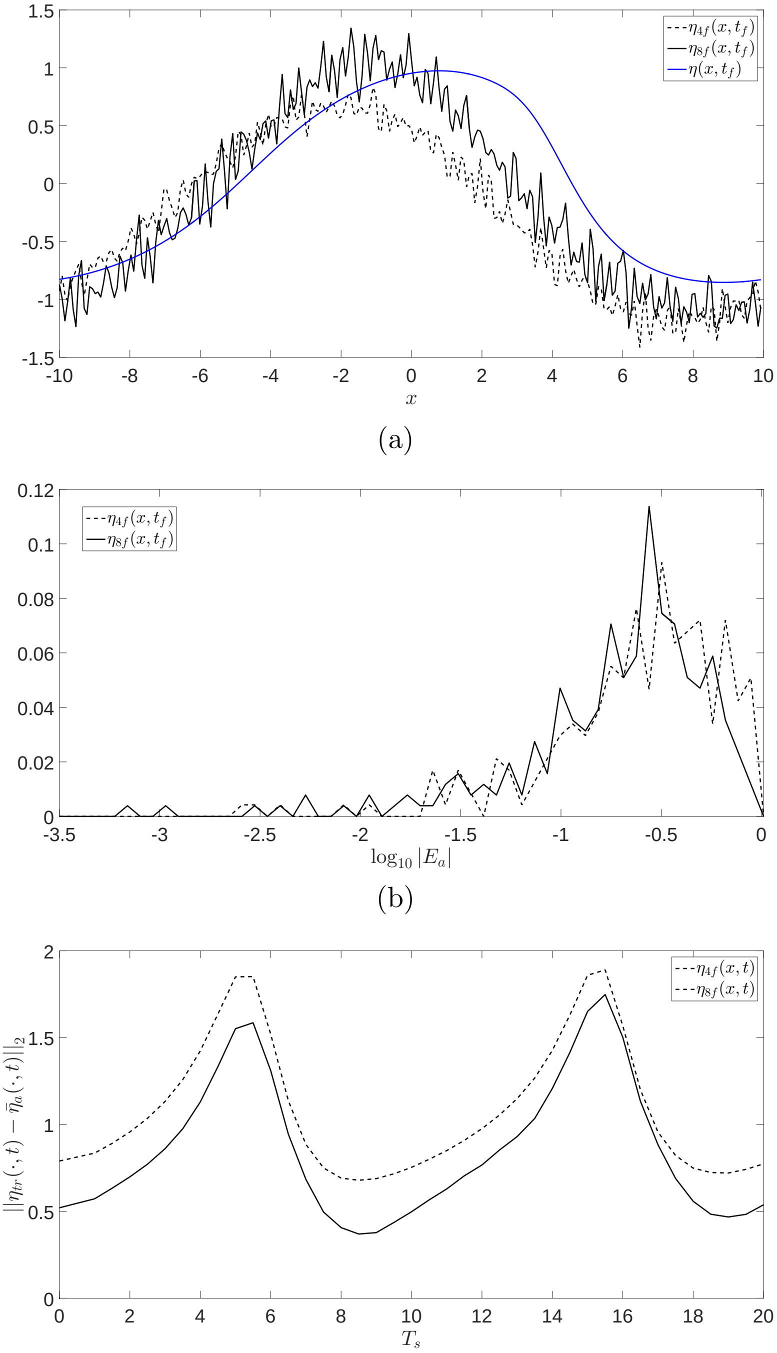


(a)

(b)

(c)

Figure 3: For $M = 1$, $dt_s = .5$, with four floats (–) compared to eight (-), we have the wave profiles at $t_f = 20$ (a), a histogram of the log of the pointwise error (b), and the root-mean square error at every sampling time (c).



(a)

(b)

(c)

Figure 4: For $M = 1$, $dt_s = .5$, with four floats (–) compared to eight (-), we have the wave profiles at $t_f = 20$ (a), a histogram of the log of the pointwise error (b), and the root-square error at every sampling time (c).