Comparison between Dommermuth/Yue and Dirichlet-Neumann Operator Approaches to Solving the Wate-Wave Problem

Introduction

The following report compares the performance of two approaches to solving for freely propagating surface water waves. These are the method of Dommermuth and Yue (DY) [1, 2], and the Dirichlet-to-Neumann Operator (DNO) method [3, 4, 5, 6, 7]. However, building off the work of Ablowitz, Fokas, and Mussilimani (AFM) [8], we derive our DNO expansions in this report using the AFM formalism, and thus we will refer to the overall method as the AFM/DNO method. The AFM formalism has proven to be a versatile means by which to represent the free surface problem in a variety of contexts such as stratified flows [9], flows with shear currents [10], flows with varying bathymetry [11, 12], flows over point vortices [13], and combinations thereof [14]. This formalism makes the derivation of the expansions associated with the DNO method efficient and straightforward while also providing a ready means for developing approximate models [8, 11, 14], or even serving as a framework for direct numerical simulation [15, 16, 17].

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 [18], both the DY and AFM/DNO methods are 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

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

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 two different approaches 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

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

While similar, as we will see, the AFM/DNO method provides conceptual advantages over the DY method, and it does provide a slight performance improvement over the DY method as well.

Method of DY

we can then readily show that

$$q_x = \phi_x + \epsilon\eta_x \left.\phi_z\right|_{z=\epsilon\eta}$$

Using this identity, we can then rewrite the two nonlinear free surface equations above as

$$\eta_t + \epsilon\eta_x q_x - \frac{1}{\mu^2}\left(1 + \epsilon^2\mu^2\eta_x^2\right) \left.\phi_z\right|_{z=\epsilon\eta} = 0, \tag{1}$$

$$q_t + \eta + \frac{\epsilon}{2}q_x^2 - \frac{\epsilon}{2\mu^2}\left(1 + \epsilon^2\mu^2\eta_x^2\right) \left.\phi_z^2\right|_{z=\epsilon\eta} = 0. \tag{2}$$

Supposing we have periodic boundaries in x at $x = \pm L$, in order to find the potential $\phi(x, z, t)$, we begin with the ansatz

$$\phi(x, z, t) = \sum_{m=0}^{M} \epsilon^m \phi^{(m)}(x, z, t) + \mathcal{O}(\epsilon^{M+1}),$$

where we take

$$\phi^{(m)}(x, z, t) = \sum_{l=-\infty}^{\infty} \hat{\phi}_l^{(m)}(t) \frac{\cosh(\mu\tilde{l}(z+1))}{\cosh(\mu\tilde{l})} e^{-i\tilde{l}x}, \ \tilde{l} = \frac{\pi l}{L}.$$

Note, at $z = 0$, we have that

$$\phi^{(m)}(x, 0, t) = \sum_{l=-\infty}^{\infty} \hat{\phi}_l^{(m)}(t) e^{-i\tilde{l}x},$$

making the coefficients $\hat{\phi}_l^{(m)}$ the Fourier transform of the function $\phi^{(m)}(x, 0, t)$, thereby motivating the use of the $\hat{\cdot}$ notation.

Taylor expanding then, we have that

$$\phi_z|_{z=\epsilon\eta} = \sum_{j=0}^{M} \epsilon^j \sum_{m=0}^{j} \frac{\eta^{j-m}}{(j-m)!} \sum_{l=-\infty}^{\infty} (\mu\tilde{l})^{j-m+1} L_{j-m+1}(\tilde{l}) \hat{\phi}_l^{(m)} e^{-i\tilde{l}x} \tag{3}$$

where

$$L_j(\tilde{l}) = \left\{ \begin{array}{rl} 1, & j \text{ is even}, \\ \tanh(\mu\tilde{l}), & j \text{ is odd}. \end{array} \right.$$

We clearly then need to have a means of finding $\hat{\phi}_l^{(m)}$. Following [1], this is readily furnished by the recursive formulas

$$\hat{\phi}_k^{(j)} = -\widehat{\left(\sum_{m=0}^{j-1} \frac{\eta^{(j-m)}}{(j-m)!} \sum_{l=-\infty}^{\infty} (\mu\tilde{l})^{j-m} L_{j-m}(\tilde{l}) \hat{\phi}_l^{(m)} e^{-i\tilde{l}x} \right)}_k, \ j \geq 1, \tag{4}$$

where

$$\hat{\phi}_k^{(0)} = \hat{q}_k,$$

and where the $\widehat{\cdot}$ denotes a Fourier transform. By computing $\phi_z(x, \epsilon\eta(x, t), t)$ at each time step, which as we have shown is now effectively given in terms of the surface potential q, thereby closing the free boundary value problem, we can then solve the coupled nonlinear equations (1) and (2) using psuedo-spectral collocation schemes.

AFM/DNO Method

The AFM/DNO method begins from an observation first made in [19] 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 [19], 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

$$\eta_t = G(\eta)q \tag{5}$$

$$q_t = -\eta - \frac{\epsilon}{2}q_x^2 + \frac{\epsilon\mu^2}{2}\frac{\left(G(\eta)q + \epsilon\eta_x q_x\right)^2}{1+\epsilon^2\mu^2\eta_x^2} \tag{6}$$

Of course, to make this approach useful, one must determine a computable form of the DNO. To do so, following arugments in [3], 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 [8], 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 $j \geq 1$,

$$\widehat{(G_j q)}_k = -\sum_{m=0}^{j-1}\frac{(\mu\tilde{k})^{(j-m)}}{(j-m)!}L_{j-m}\widehat{\left(\eta^{(j-m)}G_m q\right)}_k - \frac{i}{\mu}\tilde{k}^j L_{j+1}\widehat{\left(\eta^j q_x\right)}_k, \tag{7}$$

with

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

Comparison of the Methods

The only significant difference between the DY and AFM/DNO approaches is in how efficient it is to compute ϕ_z versus computing $G(\eta)q$. By comparing the recursive formulas (4), (3) and (7) used to compute ϕ, ϕ_z and $G(\eta)q$ in the DY and AFM/DNO methods respectively, it is not immediately clear which is the more efficient method.

One of the issues in using the DY method is making efficient use of computing the terms $\hat{\phi}_l^{(m)}$ via (4) so as to minimize redundancy when using these terms to compute ϕ_z in (3). However, by writing the first few terms down, a ready strategy can be found. To wit, using (4), we see that

$$\begin{aligned}\phi(x,0,t) = \phi^{(0)}(x,0,t) - \epsilon\left(\eta\phi_z^{(0)}(x,0,t) + \phi^{(1)}(x,0,t)\right)\\ - \epsilon^2\left(\frac{\eta^2}{2}\phi_{zz}^{(0)}(x,0,t) + \eta\phi_z^{(1)}(x,0,t) + \phi^{(2)}(x,0,t)\right) + \cdots\end{aligned}$$

Likewise, using (3), we have that

$$\begin{aligned}\phi_z(x,\epsilon\eta,t) = \phi_z^{(0)}(x,0,t) + \epsilon(\eta\phi_{zz}^{(0)}(x,0,t) + \phi_z^{(1)}(x,0,t))\\ + \epsilon^2\left(\frac{\eta^2}{2}\phi_{zzz}^{(0)}(x,0,t) + \eta\phi_{zz}^{(1)}(x,0,t) + \phi_z^{(2)}(x,0,t)\right) + \cdots\end{aligned}$$

Thus we see that computing a term at a given order of ϵ in ϕ_z gives the next order terms for finding ϕ after appropriately multiplying by η. Thus, by staggering the order of computation, we are able to minimize redundant uses of Fourier transformas and vector multiplication. However, we also note that in order to find ϕ_z up to a given order of ϵ, we must compute ϕ to one higher order. We also note incidentally that

$$\phi_z^{(0)}(x,0,t) = \mu^2 G_0 q,$$

which in effect shows how the DY and AFM/DNO methods are tied together, and likewise shows how by precomputing this term, we can readily start the recursion scheme for finding ϕ_z.

In contrast, the AFM/DNO method requires only the direct use of (7). However, as can be seen, unlike in the DY method, the final term in (7), i.e. the term

$$\frac{i}{\mu}\tilde{k}^j L_{j+1}\widehat{\left(\eta^j q_x\right)}_k,$$

does not fit within the recursive scheme. Ultimately, this requires one more Fourier transform per computation of the term $G_j q$ than would be needed for finding the equivalent term in ϕ_z. However, as noted above, we must go to one higher order in finding ϕ via (4) to find ϕ_z to the same order of ϵ as $G(\eta)q$.

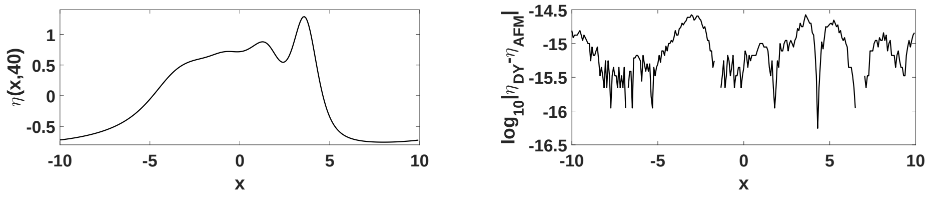
Given these trade-offs then between the two schemes, the easiest means of assessing performance is to time the different algorithms. To do this, we use a pseudo-spectral collocation scheme for the spatial variable and an implicit/explicit (IMEX) scheme AM2*/AB4 [20] for the time stepping. While the time stepping scheme is only second order accurate, it allows for relatively large time steps for dispersive nonlinear problems, and it thus provides an efficient means of comparing the DY and AFM/DNO methods. Throughout our simulations, we take a step size of $\delta t = 1e-2$. We work on a computational domain $-L \leq x \leq L$ where $L = 10$. We note that this is an arbitrary choice of a non-dimensional domain size, though it should have little effect with regards to measuring the performance of the two methods. Again, we choose $\epsilon = .1$ and $\mu = \sqrt{\mu}$. Throughout, we choose as initial conditions

$$\eta(x,0) = \cos\left(\frac{\pi x}{L}\right), \; q(x,0) = \sin\left(\frac{\pi x}{L}\right).$$

All of the code is implemented in Matlab.

Letting the total number of modes in our pseudo-spectral scheme be $K_M = 256$, corresponding to a step size of $\delta x = 5/64 \approx .08$, and letting the simulation run to non-dimesional time $t_f = 40$, and taking $M = 16$ terms in both expansions, we are able to produce the results seen in Figure 1. As seen by comparing Figures 1 (a) and (b), the two methods produce results differing only at or close to machine precision for $M = 16$. Thus, since we have two different methods that produce the same answer, we can be quite confident the implementation of both is correct, and that the results reported are meaningful. Given that we expect the difference between the results of the two method to be $\mathcal{O}(\epsilon^{17})$, that we have differences on the order of machine precision for $\epsilon = .1$ is consistent with our prediction for the magnitude of the differences. Further, given the decidely more complicated nature of the profile seen in Figure 1 (a), and the long time the simulation has run ($t_f = 4/\epsilon$), we are seeing significant impacts of nonlinearity on these time scales, and thus that we still achieve machine precision differences between the two schemes means we are not seeing the effects of our particular choice of initial conditions.

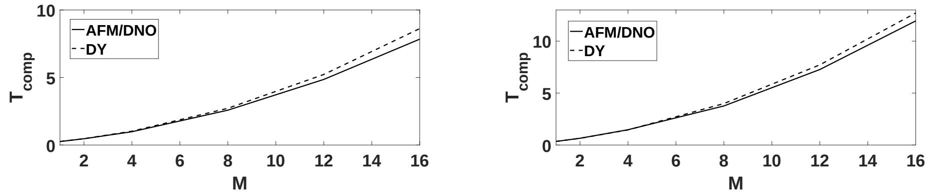
Given that we have some confidence that the two implementations are correct, using the Matlab code profiler, we can assess the difference in times between computing ϕ_z in the DY method and $G(\eta)q$ in the AFM/DNO method. We do this for $1 \leq M \leq 16$, using $K_M = 256$ modes and running simulations to non-dimensional time $t_f = 40$. The results of the profiling are found in Figure 2. Both ϕ_z and $G(\eta)q$ are computed 4001 times over the course of the simulation for $K_M = 256$ modes, see Figure 2 (a), and $K_M = 512$ modes, see Figure 2 (b). As can be seen, ultimately as M is increased, the AFM/DNO method has slightly better performance, though it is hard to argue that there is a large improvement. Further, as seen by comparing Figures 2 (a) anb (b), the difference in performance is not



(a) (b)

Figure 1: (a) Plot of the surface profile at $t_f = 40$ produced by both methods. (b) Difference of the two methods using $M = 16$ terms at $t_f = 40$. As can be seen, the difference is at or close to machine precision.

significantly altered by changing the problem size from $K_M = 256$ modes to $K_M = 512$ modes. Note, Matlab code for computing both $\phi_z(x, \epsilon\eta(x,t), t)$ and $G(\eta)q$ are provided in the Appendix.



(a) (b)

Figure 2: Time, T_{comp}, in seconds taken to compute $\phi_z(x, \epsilon\eta(x,t), t)$ (dashed line) and $G(\eta)q$ (solid line) 4001 times for $K_M = 256$ modes (a) and $K_M = 512$ modes (b). The AFM/DNO method outperforms the DY method, though only slightly, and only for larger choices of M. The performance difference is not significantly affected by problem size though.

Conclusion

As we have shown, the AFM/DNO method provides a conceptual advantage in terms of ease of implementation and derivation while also providing, an albeit modest, performance improvement over the DY method. However, given the number of different modeling environments in which both methods have been used, it is a non-trivial question to ask if this performance improvement of the AFM/DNO method would be enhanced in more complicated physical problems. For example, in a stratified flow where multiple surface potentials would need to be evaluated, this would exacerbate the computational time difference in favor of the AFM/DNO method by a factor of the number of layers present in the simulation. Thus, there is certainly further need for study of this interesting issue.

Appendix: Code

Computation of $\phi_z(x, \epsilon\eta(x,t), t)$

```
function phiz = phiz_comp(KT,eta,q,G0,ep,mu,Kmesh,Mval)
    % KT is number of modes used in pseudo-spectral scheme
    % eta is surface height in physical space
    % q is surface potential in frequency space
    % ep is epsilon
    % mu is mu
    % Kmesh = pi/Llx*[0:K -K+1:-1]
    % Mval+1 is number of terms used in phiz expansion

    mDk = mu*Kmesh;
    tnh = tanh(mDk);
    phis = zeros(KT,Mval+1);
    phiz = mu^2*G0;
    phis(:,1) = q;
    phis(:,2) = -fft(eta.*phiz);
    epp = ep;

    for jj=3:Mval+2
        phic = zeros(KT,1);
        etap = ones(KT,1);
        Dkp = ones(KT,1);
        for ll=1:jj-1
            Dkp = Dkp.*mDk;
            if mod(ll,2)==0
                tvec = real(ifft(Dkp.*phis(:,jj-ll)));
            else
                tvec = real(ifft(Dkp.*tnh.*phis(:,jj-ll)));
            end
            phiz = phiz + epp*etap.*tvec;
            if(ll<Mval+1)
                etap = eta.*etap/ll;
                phic = phic + etap.*tvec;
            end
        end
        if(jj<Mval+2)
            phis(:,jj) = -fft(phic);
            epp = epp*ep;
        end
    end
end
```

Computation of $G(\eta)q$

```
function dnohot = dno_maker(KT,eta,qx,G0,ep,mu,Kmesh,Mval)
    % KT is number of modes used in pseudo-spectral scheme
    % eta is surface height in physical space
    % q is surface potential in physical space
    % G0 is first term of DNO in physical space
    % ep is epsilon
    % mu is mu
    % Kmesh = pi/Llx*[0:K -K+1:-1]
    % Mval+1 is number of terms used in DNO expansion

    mDk = mu*Kmesh;
    tnh = tanh(mDk);
    phis = zeros(KT,Mval+1);
    dnohot = zeros(KT,1);
    phis(:,1) = G0;
    epp = 1;

    for jj=2:Mval+1
        phic = zeros(KT,1);
        Dkp = ones(KT,1);
        etap = ones(KT,1);
        for ll=1:jj-2
            Dkp = mDk.*Dkp;
            etap = eta.*etap/ll;
            if mod(ll,2) == 0
               tvec = Dkp.*fft(etap.*phis(:,jj-ll));
            else
               tvec = Dkp.*tnh.*fft(etap.*phis(:,jj-ll));
            end
            phic = phic + tvec;
        end
        Dkp = mDk.*Dkp;
        etap = eta.*etap/(jj-1);
        if mod(jj,2) == 0
            fvec = Dkp.*(tnh.*fft(etap.*G0) + ...
                1i/mu*fft(etap.*qx));
        else
            fvec = Dkp.*(fft(etap.*G0) + ...
                1i/mu*tnh.*fft(etap.*qx));
        end
        phic = -real(ifft(phic + fvec));
        phis(:,jj) = phic;
        epp = epp*ep;
        dnohot = dnohot + epp*phic;
    end
```

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] M.R. Alam. Nonlinear analysis of an actuated seafloor-mounted carpet for a high-performance wave energy extraction. *Proc. Roy. Soc. A*, 468:3153–3171, 2012.

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

[4] W. Craig, P. Guyenne, and H. Kalisch. Hamiltonian long-wave expansions for free surfaces and interfaces. *Comm. Pure Appl. Math.*, 58:1587–1641, 2005.

[5] W. Craig, P. Guyenne, D. Nicholls, and C. Sulem. Hamiltonian long-wave expansions for water waves over a rough bottom. *Proc. R. Soc. A*, 461:839–873, 2005.

[6] W. Craig, P. Guyenne, and C. Sulem. Water waves over a random bottom. *J. Fluid Mech.*, 640:79–107, 2009.

[7] P. Guyenne and D.P. Nicholls. A high-order spectral method for nonlinear water waves over moving bottom topography. *SIAM J. Sci. Comput.*, 30:81–101, 2007.

[8] 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.

[9] T.S. Haut and M.J. Ablowitz. A reformulation and applications of interfacial fluids with a free surface. *J. Fluid Mech.*, 631:375–396, 2009.

[10] A.C. Ashton and A. S. Fokas. A nonlocal formulation of rotational water waves. *J. Fluid Mech.*, 689:129–148, 2011.

[11] C.W. Curtis and S.S.P. Shen. Three-dimensional surface water waves governed by the forced Benney–Luke equation. *Stud. Appl. Math.*, 135:447–465, 2015.

[12] V. Vasan and B. Deconinck. The inverse water wave problem of bathymetry detection. *J. Fluid Mech.*, 714:562–590, 2013.

[13] C.W. Curtis and H. Kalisch. Vortex dynamics in nonlinear free surface flows. *Physics of Fluids*, 29:032101, 2017.

[14] C.W. Curtis, K.L. Oliveras, and T. Morrison. Shallow waves in density stratified shear currents. *Eur. J. Mech. B-Fluid*, 61:100–111, 2017.

[15] B. Deconinck and K. Oliveras. The instability of periodic surface gravity waves. *J. Fluid Mech.*, 675:141–167, 2011.

[16] K. L. Oliveras, V. Vasan, B. Deconinck, and D. Henderson. Recovering the water-wave profile from pressure data. *SIAM J. Appl. Math.*, 72:897–918, 2012.

[17] K.L. Oliveras and V. Vasan. A new equation describing travelling water waves. *J. Fluid Mech.*, 717:514–522, 2013.

[18] 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.

[19] 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.

[20] B. Fornberg and T.A. Driscoll. A fast spectral algorithm for nonlinear wave equations with linear dispersion. *J. Comp. Phys.*, 155:456–467, 1999.