European Journal of Mechanics B/Fluids 28 (2009) 224–233

Contents lists available at ScienceDirect

European Journal of Mechanics B/Fluids

www.elsevier.com/locate/ejmflu

Detection of ocean bathymetry from surface wave measurements

David P. Nicholls *, Mark Taber

Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 South Morgan Street (MC 249), Chicago, IL 60607, USA

ARTICLE INFO

Article history:
Received 7 August 2007
Received in revised form 9 June 2008
Accepted 10 June 2008
Available online 18 June 2008

Keywords:
Inverse problems
Free-surface potential flows
Gravity water waves
Spectral methods

ABSTRACT

The detection of ocean bathymetry is one of the most important and difficult problems in oceanography. On the theoretical side it is a classical inverse problem which features severe ill-posedness found in similar problems from a wide array of applied sciences. From a practical standpoint, standard procedures based upon "Underwater Acoustics" are quite dangerous and expensive as the desired surface is separated from measuring devices by (at least) the entire ocean layer providing a very hostile and unpredictable sampling environment. In this research we take a rather different approach to this inverse problem as we rely upon *nonlinear* features of the governing fluid mechanical equations to detect information about the ocean bathymetry. This is also in contrast to similar methods in the literature which rely solely upon the variations in the dispersion relation. Using a formulation of the water wave problem due to Zakharov, and Craig and Sulem, and the analyticity of the "Dirichlet–Neumann operator" we find surprisingly convenient formulas involving the ocean bathymetry. Of course, these formulas are ill-conditioned and nonlinear, however, we have found that application of standard techniques from the theory of inverse problems allow us to predict the shape of bottom topography with excellent precision.

1. Introduction

The detection of ocean bathymetry is one of the most important problems in oceanography for a number of reasons with, perhaps, the most obvious being the safe navigation of watercraft in nearshore regions. Furthermore, this problem is one of oceanography's most difficult for both theoretical and practical reasons. On the theoretical side it is a classical inverse problem which features all of the ill-posedness found in similar problems from a wide array of applied sciences (see, e.g., [6]). From a practical standpoint, the oftentimes easy step of simply making measurements is quite dangerous and expensive as the desired surface is separated from measuring devices by (at least) the entire ocean layer providing a very hostile and unpredictable sampling environment.

Of course, for a problem of such fundamental importance there is a vast literature on approaches to finding approximations of the ocean floor. The most mature and widely used technology is based upon the generation and subsequent detection of acoustic waves which propagate down to the ocean floor and reflect back up to the surface. For a brief sample of recent progress within this enormous field of "Underwater Acoustics" see [5,22] and the references cited therein.

In this research we take a rather different approach to this inverse problem as we rely upon *nonlinear* features of the governing fluid mechanical equations to detect information about the ocean bathymetry. This is very much in the spirit of [12,18] who use the nonlinear properties of ocean waves to find the bottom shape. Piotrowski and Dugan's [18] method is one among many in the literature which use linear variations in the dispersion relation for shoaling gravity waves as a function of depth to deduce information about the shape of the ocean floor. Grilli [12] has expanded upon these types of methods by taking into account nonlinear contributions to the dispersion relation and achieved remarkable success.

Unlike the methods outlined above we do not rely solely upon the dispersion relation for our method, rather, we take as our starting point the *entire* dynamic water wave problem and retain fully nonlinear dependence upon the bottom variation. We consider a surface-variable formulation of the governing Euler equations for free-surface water waves due to Zakharov [23] and Craig and Sulem [10] which features, quite explicitly, the unknown bottom topography. Making an ansatz of standing-wave input data and a judicious (and rigorously justified [17]) truncation of a crucial nonlinear operator (the "Dirichlet–Neumann operator" or DNO), we find surprisingly convenient formulas involving the ocean bathymetry. Of course, these formulas are ill-conditioned and nonlinear, however, we have found that application of rather standard techniques from the theory of inverse problems allow us to predict the shape of bottom topography with excellent precision.

The organization of the paper is as follows: In Section 2 we recall the governing equations of the free-surface motion of an ideal fluid under the effects of gravity (the "water wave problem"). In

* Corresponding author.
E-mail address: nicholls@math.uic.edu (D.P. Nicholls).

0997-7546/$ – see front matter © 2008 Elsevier Masson SAS. All rights reserved.
doi:10.1016/j.euromechflu.2008.06.001

Section 2.1 we outline the role of the Dirichlet–Neumann operator (DNO) in a surface formulation of the water wave problem, and in Section 2.2 we describe the depth-inversion formula due to W. Craig [7] which forms the basis for our family of techniques. In Section 2.3 we outline the "Field Expansions" method for approximating DNO which is utilized in the later design of our algorithms. In Section 3 we devise higher-order generalizations of Craig's formula which we test with a battery of numerical simulations in Section 4. In Section 5 we give some concluding remarks.

2. Governing equations

As we mentioned above, we will use as our starting point the Euler equations of ideal fluid flow [16] to model the motion of our body of water above the unknown topography. We define the domain of our problem to be

$$S_{h,\zeta,\eta} := \{-h + \zeta(x) < y < \eta(x,t)\},$$

where h is the reference depth, ζ is the variation of the bottom from this value, and η measures the perturbation of the free surface of the fluid from its quiescent state at $y = 0$. With these assumptions our data must come from the near-shore zone with an identifiable reference depth (which we assume can be found by, e.g., appealing to the dispersion relation) but *not* in the region of shoaling waves; we seek to identify $\zeta(x)$.

Within this domain the well-known equations of motion for an ideal fluid under the effects of gravity [16] are

$$\Delta\varphi = 0 \quad \text{in } S_{h,\zeta,\eta}, \tag{1a}$$

$$\partial_y\varphi - \nabla_x\zeta \cdot \nabla_x\varphi = 0 \quad \text{at } y = -h + \zeta, \tag{1b}$$

$$\partial_t\eta - \partial_y\varphi + \nabla_x\eta \cdot \nabla_x\varphi = 0 \quad \text{at } y = \eta, \tag{1c}$$

$$\partial_t\varphi + \frac{1}{2}|\nabla\varphi|^2 + g\eta = 0 \quad \text{at } y = \eta, \tag{1d}$$

where φ is the velocity potential, the velocity is given by $u = \nabla\varphi$, and g is the gravitational constant.

In an influential paper on stability of water waves, V. Zakharov [23] pointed out that the water wave problem (1a)–(1d) can be restated as a Hamiltonian system in terms of surface quantities, $\eta(x,t)$ and $\xi(x,t) := \varphi(x, \eta(x,t), t)$. While this formulation is somewhat implicit in its dependence upon η and ξ, Craig and Sulem [10] restated it in a completely *explicit* way by using the Dirichlet–Neumann operator (DNO). We define this DNO by considering the elliptic problem motivated by (1a)–(1d)

$$\Delta v = 0 \quad \text{in } S_{h,\zeta,\eta}, \tag{2a}$$

$$v(x, \eta(x)) = \xi(x), \tag{2b}$$

$$\partial_y v - \nabla_x\zeta \cdot \nabla_x v = 0 \quad \text{at } y = -h + \zeta. \tag{2c}$$

If η and ζ are sufficiently smooth then (2a)–(2c) admits a unique solution and we can compute the normal derivative of the solution at the surface $y = \eta$. The DNO carries out this procedure by mapping the Dirichlet data, ξ, to the Neumann data

$$G(\eta,\zeta)[\xi] := [\nabla v]_{y=\eta} \cdot N = \partial_y v\bigl(x, \eta(x)\bigr) - \nabla_x\eta \cdot \nabla_x v\bigl(x, \eta(x)\bigr).$$

In terms of this operator the Euler equations can now be written [10]

$$\partial_t\eta = G(\eta,\zeta)[\xi], \tag{3a}$$

$$\partial_t\xi = -g\eta - \frac{1}{2(1 + |\nabla_x\eta|^2)}\bigl[|\nabla_x\xi|^2 - \bigl(G(\eta,\zeta)[\xi]\bigr)^2 - 2G(\eta,\zeta)[\xi]\nabla_x\xi \cdot \nabla_x\eta + |\nabla_x\xi|^2|\nabla_x\eta|^2 - (\nabla_x\xi \cdot \nabla_x\eta)^2\bigr]. \tag{3b}$$

2.1. The Dirichlet–Neumann operator

Before proceeding, we review some of the key properties of the DNO. First, its dependence upon the Dirichlet data, ξ, is linear while the η and ζ variations are genuinely nonlinear. Of particular relevance to us is the analyticity result of Nicholls and Taber [17] (see Theorem 2.1) which implies the strong convergence of the following expansion:

$$G(\eta,\zeta)[\xi] = G(\varepsilon f, \delta b)[\xi] = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} G_{n,m}(f,b)[\xi]\varepsilon^n\delta^m \tag{4}$$

for $\eta(x) = \varepsilon f(x)$ and $\zeta(x) = \delta b(x)$ sufficiently small.

Theorem 2.1. *Given any integer $s \geqslant 0$, if $f, b \in C^{s+2}$ and $\xi \in H^{s+3/2}$ then $G_{n,m}[\xi] \in H^{s+1/2}$ and*

$$\bigl\|G_{n,m}[\xi]\bigr\|_{H^{s+1/2}} \leqslant K\|\xi\|_{H^{s+3/2}} D^n E^m$$

for constants $K, D, E > 0$.

We note that the $G_{n,m}[\cdot]$ inherit the property of linearity in ξ, while being homogeneous of order n and m in η and ζ, respectively. This is by no means the only result on analyticity of DNO with respect to boundary variations and we refer the interested reader to [17] for a complete literature review.

2.2. The Craig formula

In theory, given full information about $\eta(x,t)$ and $\xi(x,t)$ one should be able to use (3a)–(3b) to determine $\zeta(x)$. Clearly this is an ominous prospect so we make a number of simplifying assumptions to reduce our problem to one which is more amenable to analysis. To start, we consider only linear *surface* waves, i.e. if $\eta = \mathcal{O}(\varepsilon)$ and $\xi = \mathcal{O}(\varepsilon)$ then we truncate $\mathcal{O}(\varepsilon^2)$ terms in (3a)–(3b) to realize:

$$\partial_t\eta = G(0,\zeta)[\xi], \tag{5a}$$

$$\partial_t\xi = -g\eta. \tag{5b}$$

Given this simplification, these equations can be written in terms of a single equation for η:

$$\partial_t^2\eta = -gG(0,\zeta)[\eta]. \tag{6}$$

Notice that while surface quantities have been linearized, we retain, for the moment, fully *nonlinear* dependence on the bottom deformation ζ.

Next, to provide some workable data for our method we assume that we can identify *standing, periodic* waves on the surface of the fluid, i.e.

$$\eta(x,t) = e^{i\omega t}\bar{\eta}(x) \tag{7}$$

with frequency ω and envelope $\bar{\eta}(x)$ which satisfies

$$\bar{\eta}(x + \gamma) = \bar{\eta}(x), \quad \forall\gamma \in \Gamma,$$

for some period lattice Γ. This is by no means the only "data" that one could collect, however, it gives rise to a straightforward inverse problem and, furthermore, is the type of data which can be recovered from wave–tank measurements (e.g., the facility of D. Henderson at Penn State University [15]). Insertion of (7) into (6) results in the eigenvalue problem

$$-\omega^2\bar{\eta}(x) = -gG(0,\zeta)\bigl[\bar{\eta}(x)\bigr].$$

Using the expansion (4), we rewrite this as

$$-\omega^2\bar{\eta}(x) = -gG_{0,0}\bigl[\bar{\eta}(x)\bigr] - gG_{0,1}(\zeta)\bigl[\bar{\eta}(x)\bigr] - gG^{(2)}(0,\zeta)\bigl[\bar{\eta}(x)\bigr], \tag{8}$$

where

$$G^{(2)}(0,\zeta)[\xi] := \sum_{m=2}^{\infty} G_{0,m}(0,\zeta)[\xi]$$

and, as we shall see in Section 2.3 (Lemma 2.4),

$$G_{0,1}(\zeta) = -A(D)\zeta A(D)$$

where $A(D)$ is a pseudodifferential operator which will be specified in (21). By ignoring terms of order $\mathcal{O}(\zeta^2)$ we can approximate (8) by

$$-\omega^2\bar{\eta}(x) = -gG_{0,0}\big[\bar{\eta}(x)\big] + gA(D)\big[\zeta A(D)\big[\bar{\eta}(x)\big]\big];$$

upon solving for ζ we recover the first-order formula of Craig [7]:

$$\zeta = \frac{A(D)^{-1}[(G_{0,0} - \omega^2/g)\bar{\eta}]}{A(D)[\bar{\eta}(x)]}. \tag{9}$$

Remark 2.2. In this, and subsequent formulas, we specify in (9) a division which turns out to be one of several ill-conditioned operations which must be performed. To see this we point out that $A(D)[\bar{\eta}]$ may, of course, be exactly or nearly zero at one or many points. Naturally, the numerator will also be zero or nearly zero at corresponding points, however, the amplification of errors in the numerator at such points is clearly undesirable. To ameliorate these errors we implement a "careful divide" which we now describe for the generic division:

$$q(x) = n(x)/d(x). \tag{10}$$

To begin, we interpret (10) as

$$d(x)q(x) = n(x), \tag{11}$$

which is equivalent as long as $x \neq 0$, and, if $d(\bar{x}) = 0$ we will naturally need $n(\bar{x}) = 0$ leaving $q(\bar{x})$ not uniquely defined. As our method will be a Fourier collocation scheme we are interested in solving:

$$d(x_j)q(x_j) = n(x_j), \tag{12}$$

at all of the N_x collocation points x_j. We rewrite (12) as

$$D\vec{q} = \vec{n}, \quad D \in \mathbf{R}^{N_x \times N_x}, \quad \vec{q} \in \mathbf{R}^{N_x}, \quad \vec{n} \in \mathbf{R}^{N_x},$$

where $D = \operatorname{diag}\{d_j\}$. If the determinant of D is non-zero then we recover the classical division (10) using the inverse of D. However, since $\det(D)$ may be zero (or very small) we find $\vec{q}$ using the Singular Value Decomposition (SVD) of D coupled with a spectral cut-off regularization [6]. Using this method, at points of singularity the (approximate) solution vector $\vec{q}$ is simply set to zero which may or may not give good accuracy.

We implemented this scheme and had some success but found that a slight variant worked better. Assuming that d, q, and n all have one derivative, we differentiate (11) to realize:

$$\left[d'(x) + d(x)\frac{d}{dx}\right]q(x) = n'(x). \tag{13}$$

We now evaluate (13) at the collocation points x_j and perform the "safe divide" outlined above with the SVD. To completely specify this method we approximate the derivative with a second–order accurate finite difference scheme which also provides desirable numerical smoothing to our methods [6].

Remark 2.3. We point out that formulas (8) and (9) do not depend upon the periodicity of the envelope $\bar{\eta}(x)$ (though the operators which appear are easier to apply and invert in this case). If other boundary conditions are relevant to the application at hand, then these formulas can still be utilized, however, one must devise numerical methods to handle the resulting operators.

2.3. Field expansions

Our goal in this research is to test the utility of (9) and its generalizations. For this, it is clear that we need convenient formulas for the Taylor coefficients of the DNO, $G_{0,m}$ in (4). Several methods have been developed for this purpose including those based upon "Operator Expansions" (OE) due to Smith [21], and Craig, Guyenne, Nicholls, and Sulem [8,13,14]. In this section we present an alternative, and we believe more straightforward, approach based upon the "Field Expansions" (FE) approach of Rayleigh [19], Rice [20], and Bruno and Reitich [1–3].

To begin we set $\zeta(x) = \delta b(x)$ and, since we do not consider top variations for the moment, $\eta(x) \equiv 0$. The results of Nicholls and Taber [17] imply that, if $b(x)$ is sufficiently smooth, then

$$v = v(x, y; \delta) = \sum_{m=0}^{\infty} v_m(x, y)\delta^m,$$

which, upon insertion into (2a)–(2c), gives rise to the sequence of problems

$$\Delta v_m = 0 \quad \text{in } S_{h,0,0}, \tag{14a}$$

$$v_m(x, 0) = \delta_{m,0}\xi(x), \tag{14b}$$

$$\partial_y v_m(x, -h) = H_m(x), \tag{14c}$$

where

$$H_m(x) = -\sum_{l=0}^{m-1} B_{m-l}\partial_y^{m-l+1} v_l(x, -h) + (\nabla_x b)\cdot \sum_{l=0}^{m-1} B_{m-l-1}\nabla_x\partial_y^{m-l-1} v_l(x, -h), \tag{14d}$$

and

$$B_l(x) := \frac{b(x)^l}{l!}.$$

If these v_m can be recovered, then computing the terms $G_{0,m}$ in the expansion of the DNO is straightforward as (for $\eta \equiv 0$)

$$G(0, \zeta)[\xi] = \partial_y v(x, 0),$$

so that

$$G_{0,m}(b)[\xi] = \partial_y v_m(x, 0), \tag{15}$$

and we suppress the η dependence for simplicity.

2.3.1. Order zero

At order $m = 0$ it is easy to show that

$$v_0(x, y) = \sum_{p \in \Gamma'} \frac{\cosh(|p|(y + h))}{\cosh(|p|h)}\hat{\xi}_p e^{ip\cdot x}, \tag{16}$$

where Γ' is the conjugate lattice of wavenumbers, and $\hat{\xi}_p$ are the Fourier coefficients of $\xi(x)$. From this and (15) we recover the classical result:

$$G_{0,0}[\xi] = \partial_y v_{0,0}(x, 0) = \sum_{p \in \Gamma'} |p|\tanh\big(h|p|\big)\hat{\xi}_p e^{ip\cdot x} =: |D|\tanh\big(h|D|\big)\xi \tag{17}$$

which defines the Fourier multiplier $G_{0,0} = |D|\tanh(h|D|)$.

2.3.2. Higher orders

For orders $m \geqslant 1$, the solution of (14a) is

$$v_m(x,y) = \sum_{p \in \Gamma'} \left(\alpha_p^{(m)} \cosh\left(|p|y\right) + \beta_p^{(m)} \sinh\left(|p|y\right)\right) e^{ip\cdot x},$$

while (14b) implies that $\alpha_p^{(1)} = 0$, so that

$$v_m(x,y) = \sum_{p \in \Gamma'} \beta_p^{(m)} \sinh\left(|p|y\right) e^{ip\cdot x}. \tag{18}$$

Finally, (14c) mandates that

$$\begin{aligned}
&\sum_{p \in \Gamma'} |p| \cosh\left(h|p|\right) \beta_p^{(m)} e^{ip\cdot x} \\
&\quad = \partial_y v_m(x,-h) = H_m(x) \\
&\quad = -\sum_{l=0}^{m-1} B_{m-l} \partial_y^{m-l+1} v_l(x,-h) \\
&\qquad + (\nabla_x b) \cdot \sum_{l=0}^{m-1} B_{m-l-1} \nabla_x \partial_y^{m-l-1} v_l(x,-h) \\
&\quad = -B_m \partial_y^{m+1} v_0(x,-h) + (\nabla_x b) \cdot B_{m-1} \nabla_x v_0(x,-h) \\
&\qquad - \sum_{l=1}^{m-1} B_{m-l} \partial_y^{m-l+1} v_l(x,-h) \\
&\qquad + (\nabla_x b) \cdot \sum_{l=1}^{m-1} B_{m-l-1} \nabla_x \partial_y^{m-l-1} v_l(x,-h),
\end{aligned} \tag{19}$$

where we have separated the v_0 terms as they are fundamentally different from the v_m $(m \geqslant 1)$. In fact, from (16),

$$\begin{aligned}
\partial_y^l v_0(x,-h) &= \left[\sum_{p \in \Gamma'} |p|^l \left\{ \frac{(d_z^l \cosh)(|p|(y+h))}{\cosh(h|p|)} \right\} \hat{\xi}_p e^{ip\cdot x} \right]_{y=-h} \\
&=: \Phi_l(D)\xi
\end{aligned}$$

where

$$\Phi_l(D) = \begin{cases} |D|^l \operatorname{sech}(h|D|), & l \text{ even}, \\ 0, & l \text{ odd}, \end{cases}$$

while (for $m \geqslant 1$), from (18),

$$\begin{aligned}
\partial_y^l v_m(x,-h) &= \left[\sum_{p \in \Gamma'} |p|^l \left\{ \left(d_z^l \sinh\right)\left(|p|y\right) \right\} \beta_p^{(m)} e^{ip\cdot x} \right]_{y=-h} \\
&=: \Psi_l(D)\beta^{(m)}
\end{aligned}$$

where

$$\Psi_l(D) = \begin{cases} -|D|^l \sinh(h|D|), & l \text{ even}, \\ |D|^l \cosh(h|D|), & l \text{ odd}. \end{cases}$$

Returning to (19) we find that

$$\begin{aligned}
|D| \cosh\left(h|D|\right) \beta^{(m)} &= -B_m \Phi_{m+1}(D)\xi + (\nabla_x b) \cdot B_{m-1} \nabla_x \Phi_{m-1}(D)\xi \\
&\quad - \sum_{l=1}^{m-1} B_{m-l} \Psi_{m-l+1}(D) \beta^{(l)} \\
&\quad + (\nabla_x b) \cdot \sum_{l=1}^{m-1} B_{m-l-1} \nabla_x \Psi_{m-l-1}(D) \beta^{(l)}.
\end{aligned}$$

Recalling that $\nabla_x = -iD$ and noting that

$$\Phi_{j+2}(D) = |D|^2 \Phi_j(D), \quad \Psi_{j+2}(D) = |D|^2 \Psi_j(D), \quad j \geqslant 0,$$

we have that

$$\begin{aligned}
|D| \cosh\left(h|D|\right) \beta^{(m)} &= -\left[B_m |D|^2 + (Db) \cdot B_{m-1} D\right] \Phi_{m-1}(D)\xi \\
&\quad - \sum_{l=1}^{m-1} \left[B_{m-l} |D|^2 + (Db) \cdot B_{m-l-1} D\right] \Psi_{m-l-1}(D) \beta^{(l)}.
\end{aligned}$$

Finally, with $|D|^2 = D^2$ and the product rule, which implies that

$$B_k |D|^2 f + (Db) \cdot B_{k-1} Df = D \cdot B_k Df,$$

we have

$$\begin{aligned}
|D| \cosh\left(h|D|\right) \beta^{(m)} &= -D \cdot B_m D \Phi_{m-1}(D)\xi \\
&\quad - \sum_{l=1}^{m-1} D \cdot B_{m-l} D \Psi_{m-l-1}(D) \beta^{(l)}.
\end{aligned} \tag{20}$$

Using (15) we compute the m-th term in the DNO to be:

$$G_{0,m}[\xi] = \partial_y v_{0,m}(x,0) = |D| \beta^{(m)}.$$

Lemma 2.4. *If we define the Fourier multiplier*

$$A(D) := \operatorname{sech}\left(h|D|\right) D, \tag{21}$$

then the FE recursions give, at order one,

$$\begin{aligned}
|D| \cosh\left(h|D|\right) \beta^{(1)} &= -D \cdot bD \operatorname{sech}\left(h|D|\right)\xi \\
&= -D \cdot bA(D)\xi
\end{aligned}$$

and

$$G_{0,1}[\xi] = -\operatorname{sech}\left(h|D|\right) D \cdot bD \operatorname{sech}\left(h|D|\right)\xi = -A(D) \cdot bA(D)\xi. \tag{22}$$

Lemma 2.5. *The FE recursions give, at order two,*

$$\begin{aligned}
|D| \cosh\left(h|D|\right) \beta^{(2)} &= -D \cdot bD\left(-\sinh\left(h|D|\right)\right) \beta^{(1)} \\
&= -D \cdot bD \sinh\left(h|D|\right) |D|^{-1} \operatorname{sech}\left(h|D|\right) D \cdot bA(D)\xi \\
&= -D \cdot b \operatorname{sgn}(D) D \cdot \tanh\left(h|D|\right) bA(D)\xi,
\end{aligned}$$

and

$$G_{0,2}[\xi] = -A(D) \cdot b \operatorname{sgn}(D) D \cdot \tanh\left(h|D|\right) bA(D)\xi. \tag{23}$$

In this final lemma we point out that at every order the DNO has a particular form.

Lemma 2.6. *The FE recursions give, at order m,*

$$\begin{aligned}
|D| \cosh\left(h|D|\right) \beta^{(m)} &= -D \cdot B_m D \Phi_{m-1}(D)\xi \\
&\quad - \sum_{l=1}^{m-1} D \cdot B_{m-l} D \Psi_{m-l-1}(D) \beta^{(l)} \\
&= -D \cdot b Q_m,
\end{aligned}$$

which defines the operator Q_m, and thus

$$G_{0,m}[\xi] = -\operatorname{sech}\left(h|D|\right) D \cdot b Q_m = -A(D) \cdot b Q_m. \tag{24}$$

For every m one can factor the operator $A(D)$ and the multiplication by b to the left of the m-th term in the expansion of the DNO. This fact allows us to generalize one of the methods of Section 3 to arbitrarily high order. Before proceeding we note that the formulas given in Lemmas 2.4, 2.5, and 2.6 were also derived by Craig, Guyenne, Nicholls, and Sulem [8] via the Operator Expansions formalism.

3. Higher order algorithms

As we shall see in Section 4, the Craig formula presented in Section 2.2, see (9), provides a fast and easy-to-implement method for recovering bottom topography from surface wave measurements which makes no assumptions of long wavelength or shallow depth. However, as the method is based upon the truncation of the DNO after the linear term there is an implicit assumption of smallness in the bathymetry amplitude. In an effort to relax this restriction we now derive a variety of higher-order (in bottom perturbation magnitude) schemes to improve upon the formula of Craig. At this point we specialize to the case of two-dimensional water waves (one-dimensional bottom topography) as this makes our algorithms somewhat easier to implement. This can already be seen with the Craig formula, (9), containing (somewhat implicitly) the inversion of a divergence operator, $A(D)$, which requires the development of a certain amount of standard technology; we will save the generalization to three dimensions for future work.

3.1. Second order methods

Returning to (8), the natural way to include higher order information about the bottom perturbation is to truncate the DNO after the *second* order term, $G_{0,2}$, and make the approximation:

$$-\omega^2\bar{\eta}(x) = -gG_{0,0}\big[\bar{\eta}(x)\big] - gG_{0,1}(\zeta)\big[\bar{\eta}(x)\big] - gG_{0,2}(\zeta)\big[\bar{\eta}(x)\big].$$

Using our formulas for $G_{0,1}$, (22), and $G_{0,2}$, (23), we find

$$-\omega^2\bar{\eta}(x) = -gG_{0,0}\bar{\eta}(x) + gA\zeta A\bar{\eta}(x) + gA\zeta\big(\operatorname{sgn}(D)\tanh\big(h|D|\big)D\big)\zeta A\bar{\eta}(x).$$

Note that in two dimensions

$$\operatorname{sgn}(D)\tanh\big(h|D|\big)D = |D|\tanh\big(h|D|\big) = G_{0,0},$$

so that, simplifying,

$$A\zeta A\bar{\eta}(x) + A\zeta G_{0,0}\zeta A\bar{\eta}(x) = \left(G_{0,0} - \frac{\omega^2}{g}\right)\bar{\eta}(x) =: R(\bar{\eta};\omega,g). \quad (25)$$

Naturally, with the complicated quadratic dependence upon ζ there is no convenient, closed-form solution for our unknown. However, there are several *iterative* approaches which immediately present themselves, supplemented with the solution of the Craig formula as an initial guess.

3.1.1. Left factorization

Beginning with (25), one could factor the operator A and the multiplication by ζ on the left-hand side so that

$$A\zeta[A\bar{\eta} + G_{0,0}\zeta A\bar{\eta}] = R.$$

Defining the linear (in ζ) function

$$\psi(\zeta) := A\bar{\eta} + G_{0,0}\zeta A\bar{\eta} \quad (26)$$

we can now solve

$$\zeta = \frac{A^{-1}R}{\psi(\zeta)}.$$

Since ζ appears on both sides we set up an iteration scheme

$$\zeta_{n+1} = \frac{A^{-1}R}{\psi(\zeta_n)} \quad (27)$$

which we denote the "Second Order Left Factorization" (2L) method.

3.1.2. Right factorization

In a similar fashion, starting with (25), we factor the operator A and the multiplication by ζ on the right-hand side so that

$$[A + A\zeta G_{0,0}]\zeta A\bar{\eta} = R.$$

Defining the operator

$$\tilde{A}(\zeta) := A + A\zeta G_{0,0},$$

we can now solve

$$\zeta = \frac{\tilde{A}^{-1}(\zeta)R}{A\bar{\eta}}.$$

Again, we set up an iteration scheme

$$\zeta_{n+1} = \frac{\tilde{A}^{-1}(\zeta_n)R}{A\bar{\eta}}, \quad (28)$$

which we denote the "Second Order Right Factorization" (2R) method.

3.1.3. Picard iteration

Finally, starting with (25) we can set up a Picard iteration in the following way:

$$A\zeta A\bar{\eta} = R - A\zeta G_{0,0}\zeta A\bar{\eta},$$

and

$$\zeta = \frac{A^{-1}(R - A\zeta G_{0,0}\zeta A\bar{\eta})}{A\bar{\eta}}.$$

We now have the iteration scheme

$$\zeta_{n+1} = \frac{A^{-1}(R - A\zeta_n G_{0,0}\zeta_n A\bar{\eta})}{A\bar{\eta}}, \quad (29)$$

which we call the "Second Order Picard" (2P) method.

3.2. Higher order methods

Due to the form of the DNO (see Lemma 2.6), the Left Factorization method, (27), can easily be generalized to arbitrary order. Beginning with (8), we truncate after M perturbation orders:

$$-\omega^2\bar{\eta}(x) = -gG_{0,0}\big[\bar{\eta}(x)\big] - g\sum_{m=1}^{M} G_{0,m}(\zeta)\big[\bar{\eta}(x)\big],$$

or, using the notation of Section 3.1.1,

$$R = -\sum_{m=1}^{M} G_{0,m}(\zeta)\big[\bar{\eta}(x)\big].$$

We now use Lemma 2.6 to realize

$$R = -\sum_{m=1}^{M} A(D)\zeta Q_m(\zeta)\big[\bar{\eta}(x)\big],$$

and

$$A^{-1}R = -\zeta\left(\sum_{m=1}^{M} Q_m(\zeta)\big[\bar{\eta}(x)\big]\right).$$

In analogy with (26) we define

$$\psi_M(\zeta) := -\left(\sum_{m=1}^{M} Q_m(\zeta)\big[\bar{\eta}(x)\big]\right)$$

which gives rise to the equation

$$\zeta = \frac{A^{-1}R}{\psi_M(\zeta)}.$$

We set up the iteration

$$\zeta_{n+1} = \frac{A^{-1}R}{\psi_M(\zeta_n)} \tag{30}$$

which we denote the "M-th Order Left Factorization" (M L) method.

The design philosophy of the Picard method is also amenable to higher order implementation. In fact, using the M-th order approximation of the DNO in (8), we can write:

$$A\zeta A\bar{\eta} = \left(G_{0,0} - \frac{\omega^2}{g}\right)\bar{\eta} + \sum_{m=2}^{M} G_{0,m}(0,\zeta)[\bar{\eta}].$$

Setting up the natural iteration

$$\zeta_{n+1} = \frac{A^{-1}[(G_{0,0} - \omega^2/g)\bar{\eta} + \sum_{m=2}^{M} G_{0,m}(0,\zeta_n)[\bar{\eta}]]}{A\bar{\eta}} \tag{31}$$

we have the "M-th Order Picard" (M P) method.

We remark that it is not immediately obvious how to generalize the Right Factorization scheme to arbitrary order. In fact a preliminary study of the terms $G_{0,3}$ and $G_{0,4}$, seems to indicate that such a method may be *impossible* to conveniently formulate.

4. Numerical results

To completely specify any of the algorithms we have proposed (Craig's formula, ML, 2R, MP) we choose a Fourier collocation method [4,11] for discretization at N_x collocation points with products evaluated via Fast Fourier transforms (FFTs). This method is particularly appropriate not only due to the periodic boundary conditions and the rapid rate of convergence of these schemes, but also because these techniques are so amenable to the approximation of Fourier multipliers like $A(D)$ and $G_{0,m}$ (see, e.g., [9,10,13, 14]).

To test the algorithms that we have outlined in Section 3 we select a *known* profile for the bottom topography, insert this as an input to the eigenvalue problem (8) (evaluated up to a high order in expansion of the DNO), generate an eigenvalue/eigenfunction pair $(\omega, \bar{\eta})$, and insert this as data into each of our algorithms (ML, 2R, MP). Clearly, the fact that we have an *exact* solution for this data allows us to make very precise measurements of our error.

Regarding the profiles, we select two which we feel are representative of reasonable bottom topography:

$$\zeta^{(1)}(x) = a\,\operatorname{sech}(bx), \tag{32a}$$

$$\zeta^{(2)}(x) = a\big[\tanh\big(b(x+c)\big) - \tanh\big(b(x-c)\big)\big]. \tag{32b}$$

The first, $\zeta^{(1)}$ (depicted in Fig. 1), is a simple Gaussian while the second, $\zeta^{(2)}$ (pictured in Fig. 2), is meant to resemble a sandbar. Both profiles are extended periodically, with period 2π, outside the displayed interval $[-\pi,\pi]$. The parameters a and b (for the Gaussian profile) and a, b, and c (for the sandbar profile) can be adjusted to alter the shape of the bottom topography and we will test our methods with respect to change of these parameters. One note before proceeding, the parameter a for $\zeta^{(1)}$ does measure the amplitude of the Gaussian topography, however, for $\zeta^{(2)}$ the value of a gives only about one half of the amplitude of the sandbar profile.

To begin, we consider the Gaussian profile, (32a), with fixed parameters $a = 0.07$, $b = 2$, and $h = 0.6$, and study the convergence of five algorithms as the iteration order is increased for fixed $N_x = 64$. The five algorithms we consider are:

(1) Second-order Picard (2P),
(2) Second-order Left-Factored (2L),
(3) Second-order Right-Factored (2R),

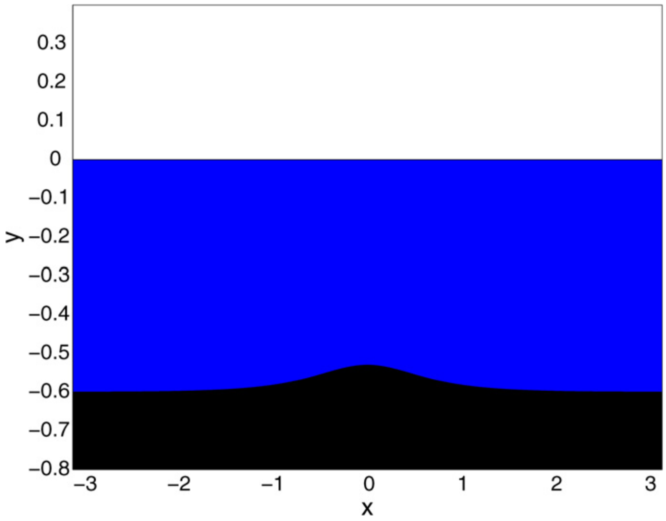

Fig. 1. Plot of Gaussian profile, (32a), with $a = 0.07$, $b = 2$, and $h = 0.6$.

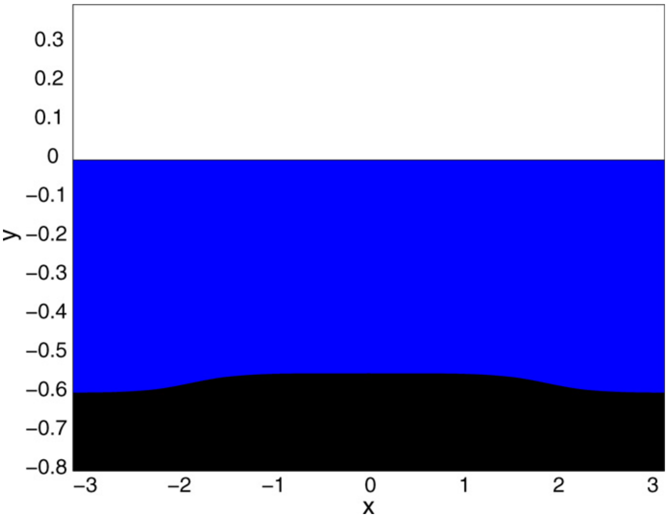

Fig. 2. Plot of sandbar profile, (32b), with $a = 0.025$, $b = 2$, $c = 3\pi/5$, and $h = 0.6$.

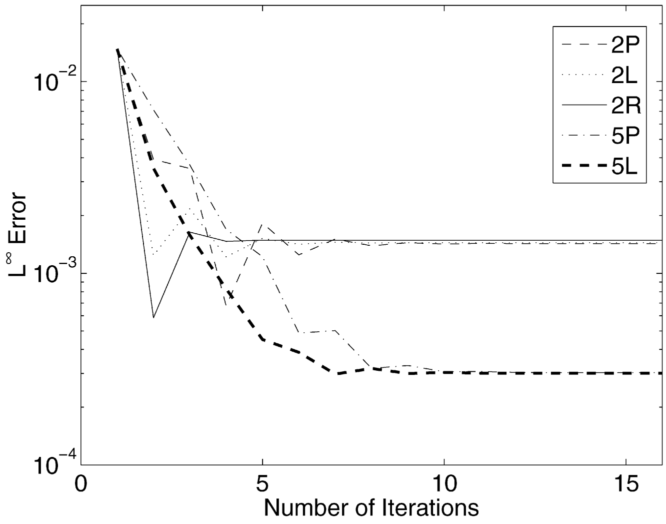

Fig. 3. Plot of convergence of five algorithms to the exact bottom topography given by the Gaussian profile (32a). L^∞-error is plotted versus iteration number. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

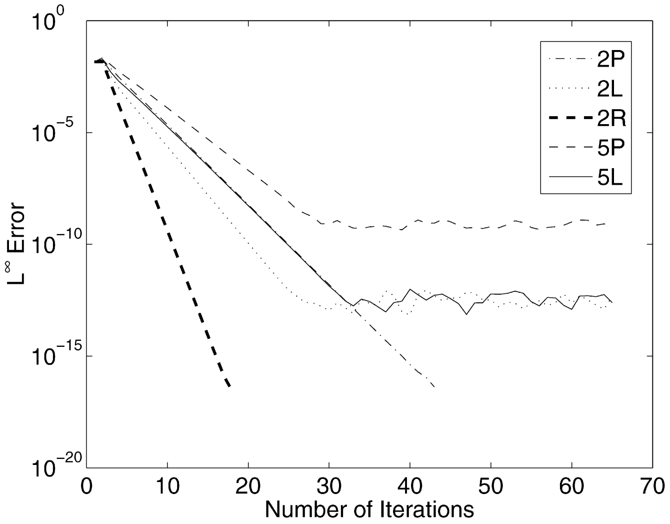

Fig. 4. Plot of self-convergence of five algorithms for bottom topography given by the Gaussian profile (32a). L^∞-difference is plotted versus iteration number. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

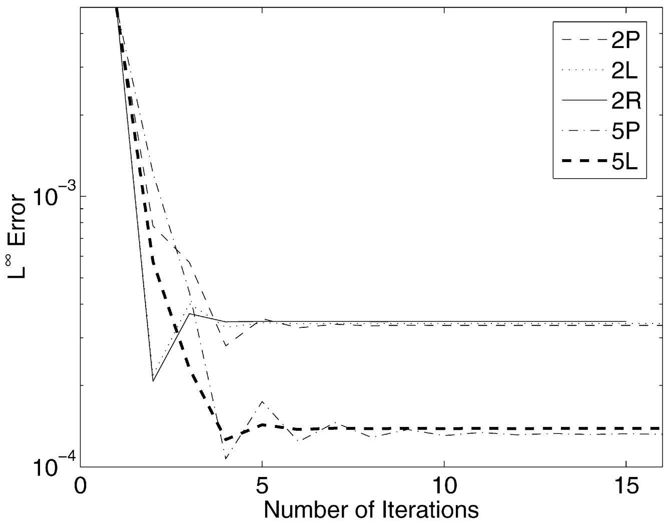

Fig. 6. Plot of convergence of five algorithms to the exact bottom topography given by the sandbar profile (32b). L^∞-error is plotted versus iteration number. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

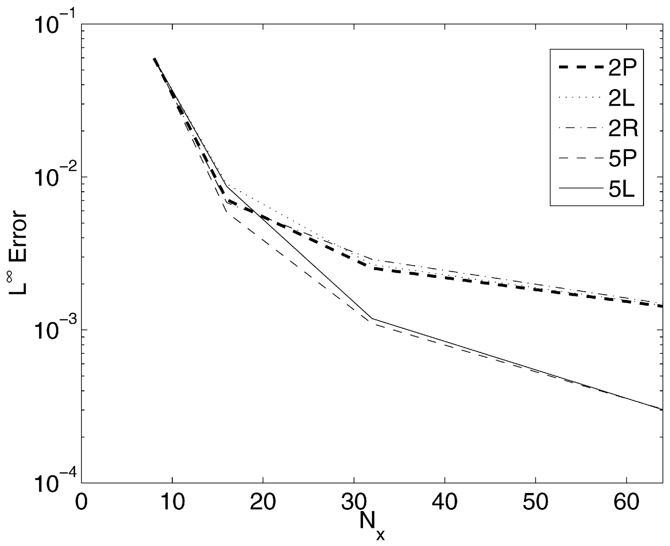

Fig. 5. Plot of convergence of five algorithms to the exact bottom topography given by the Gaussian profile (32a). L^∞-error is plotted versus N_x. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

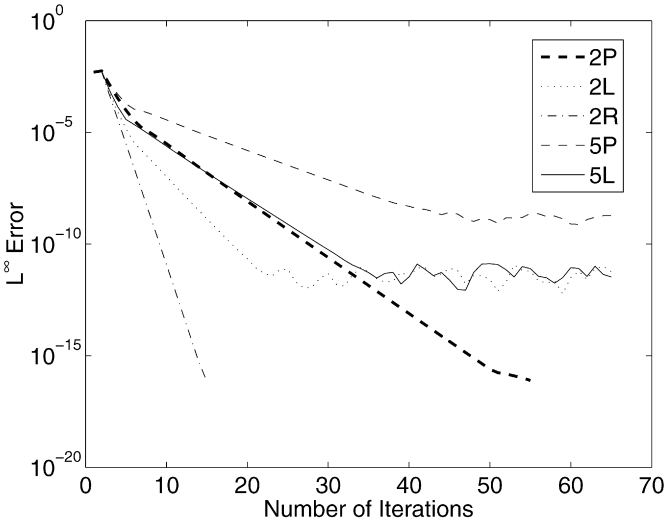

Fig. 7. Plot of self-convergence of five algorithms for bottom topography given by the sandbar profile (32b). L^∞-difference is plotted versus iteration number. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

(4) Fifth-order Picard (5P),
(5) Fifth-order Left-Factored (5L).

Before considering the details of the behavior of these five methods, we point out that all five do converge quite well with errors of 0.1% or even 0.01% once they are properly converged. Furthermore, they all outperform the Craig formula since this, as we mentioned earlier, is always used as the initial guess, and all methods improve upon the first iteration step. It should be pointed out that none of the methods converge beyond errors of about 10^{-4} as these are all approximate algorithms. Additionally, the division algorithm we discussed in Remark 2.2 is effectively a smoothing technique which mollifies the recovered approximation thereby introducing extra errors. Despite this, we find the performance of all algorithms to be excellent while being extremely easy to implement.

Considering the Gaussian profile, in Fig. 3 we study the convergence in iteration order of these five methods to the exact solution, while in Fig. 4 we display their self-convergence. As expected, Fig. 3 shows that the fifth-order algorithms outperform the second-order methods, while algorithms of the same order perform similarly. We note also that nearly an order of magnitude of additional accuracy can be realized by moving to one of the fifth-order schemes. In Fig. 4 we notice that all methods self-converge quite quickly, however, the left–factored methods and the fifth-order Picard schemes seem to saturate with errors around 10^{-10} to 10^{-12}; we do point out that this is well below the precision any of these algorithms can hope to deliver. Finally, in Fig. 5 we see that there is quite rapid convergence for all methods as N_x is refined. In this plot the iteration order is chosen so that each of the methods is properly converged (error approximately 10^{-10} or less). Not surprisingly, the fifth-order methods converge more

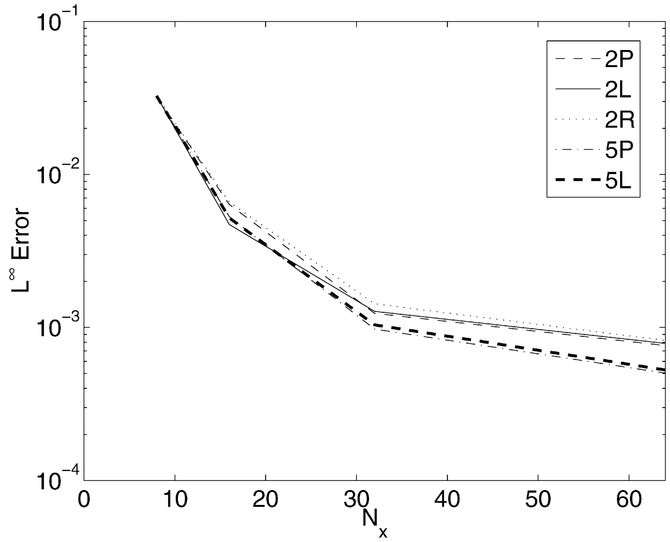

Fig. 8. Plot of convergence of five algorithms to the exact bottom topography given by the sandbar profile (32b). L^∞-error is plotted versus N_x. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

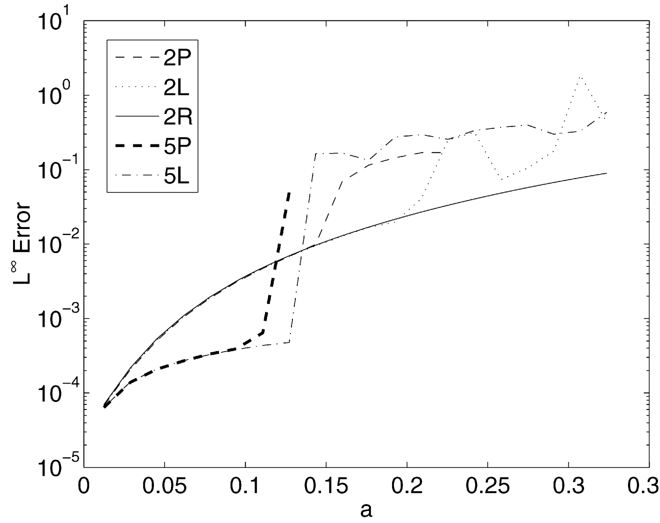

Fig. 9. Plot of convergence of five algorithms to the exact bottom topography given by the Gaussian profile (32a) with varying a. L^∞-error is plotted versus a. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

quickly than the second-order while, again, methods of the same order perform approximately the same.

These computations were repeated for the sandbar profile, (32b), (with $a = 0.025$, $b = 2$, $c = 3\pi/5$, and $h = 0.6$) and the results are shown in Figs. 6, 7 and 8. The conclusions that we reached above are, for the most part, realized once again with only slight variations in the particular details of the convergence. Generally, the higher order methods perform better while the difference in capabilities among methods of the same order is negligible.

Having convinced ourselves that our new methods work extremely well for the topography parameter choices given above, we set about testing the capabilities of our methods as these constants are varied, even toward extreme values. In all of these experiments we use, for the parameters *not* being varied, the following baseline values:

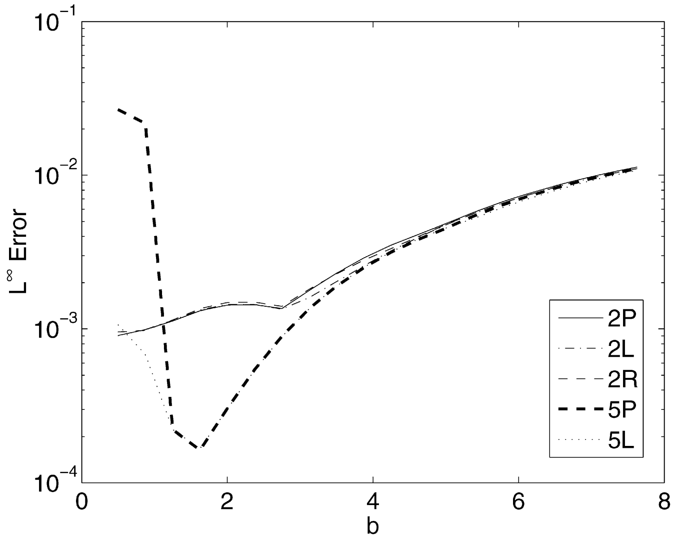

Fig. 10. Plot of convergence of five algorithms to the exact bottom topography given by the Gaussian profile (32a) with varying b. L^∞-error is plotted versus b. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left–Factorization method.)

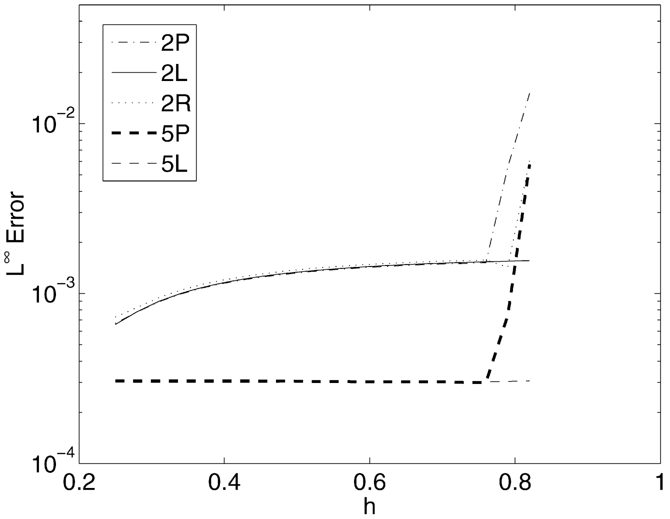

Fig. 11. Plot of convergence of five algorithms to the exact bottom topography given by the Gaussian profile (32a) with varying h. L^∞-error is plotted versus h. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

(1) Gaussian profile: $a = 0.07$, $b = 2$, $h = 0.6$,
(2) Sandbar profile: $a = 0.025$, $b = 2$, $c = 3\pi/5$, $h = 0.6$.

To begin, we again focus upon the Gaussian profile, (32a) and vary the parameters a, b, and h. In Fig. 9 we see the results of our five methods as a is moved from just over 0.3 down towards zero. All methods perform better as a is decreased, however, we note that the two Picard methods do not converge over the entire parameter range and that a must be sufficiently small for them to work. In Fig. 10 we notice somewhat different behavior as b is changed from 8.0 down towards zero; all of the methods seem to work best near the middle of the range. Finally, in Fig. 11 we move the location of the reference depth, h, and study our five algorithms. For the most part all five perform uniformly well for almost all values of h chosen (between 0.25 and 0.8) while there is a precipitous divergence as h nears 0.8 which, presumably, is

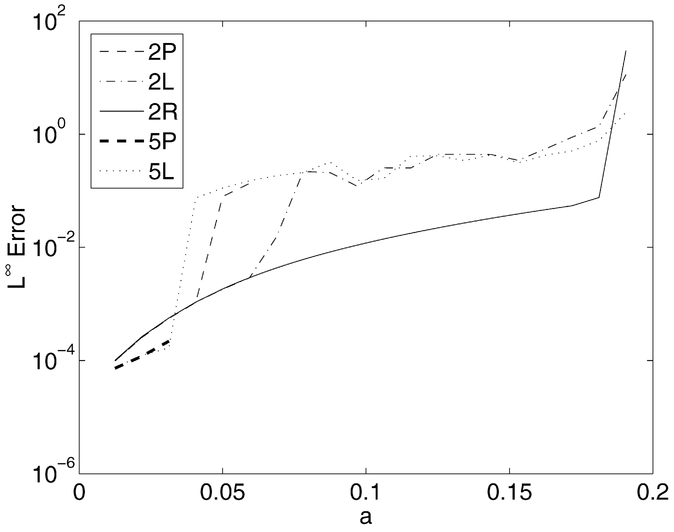

Fig. 12. Plot of convergence of five algorithms to the exact bottom topography given by the sandbar profile (32b) with varying a. L^∞-error is plotted versus a. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

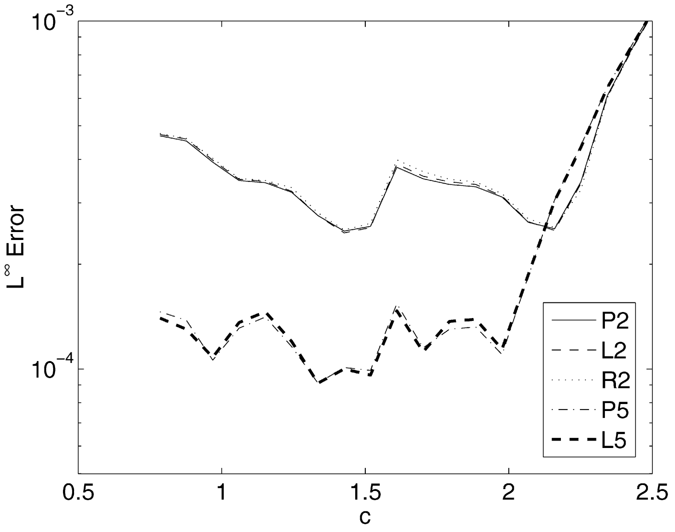

Fig. 14. Plot of convergence of five algorithms to the exact bottom topography given by the sandbar profile (32b) with varying c. L^∞-error is plotted versus c. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

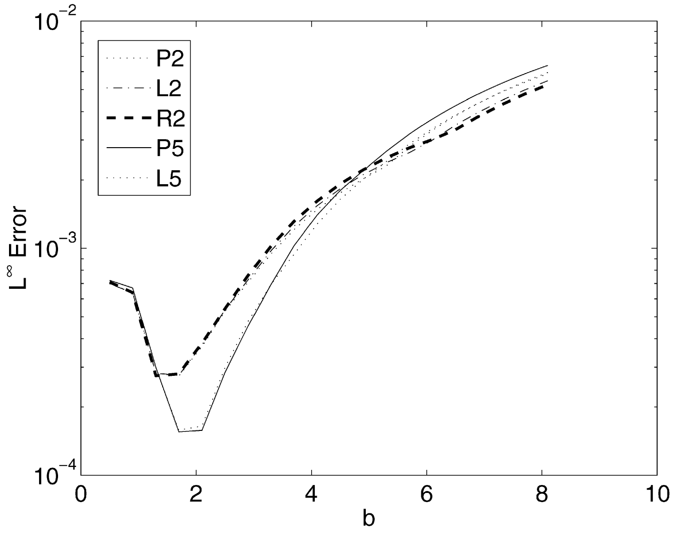

Fig. 13. Plot of convergence of five algorithms to the exact bottom topography given by the sandbar profile (32b) with varying b. L^∞-error is plotted versus b. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

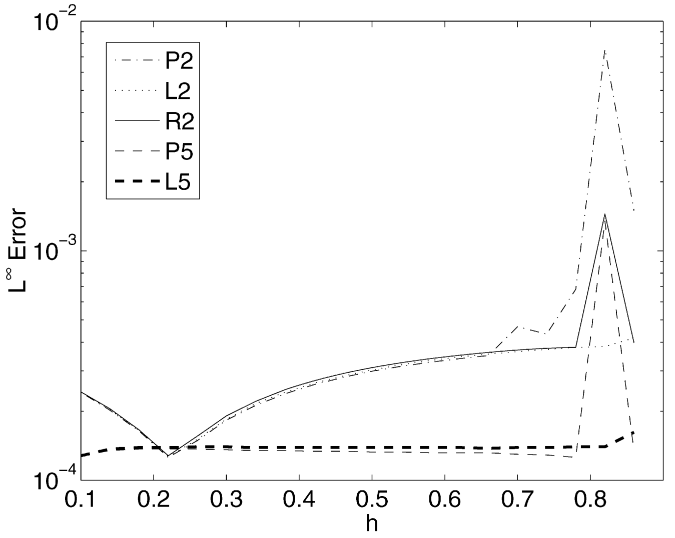

Fig. 15. Plot of convergence of five algorithms to the exact bottom topography given by the sandbar profile (32b) with varying h. L^∞-error is plotted versus h. (2P: Second-order Picard method; 2L: Second-order Left-Factorization method; 2R: Second-order Right-Factorization method; 5P: Fifth-order Picard method; 5L: Fifth-order Left-Factorization method.)

the value of the depth where one starts to enter the deep-water regime.

Finally, we consider the sandbar profile, (32b) and change a, b, c, and h. In Fig. 12 we change a from roughly 0.2 down to zero. Again, all methods improve as a is decreased, but we point out the rather limited range of applicability that the Picard schemes possess. In Fig. 13 we vary the parameter b and notice that all methods are quite comparable (though the fifth-order methods are superior for a certain range) and work best for a moderate value of b. The parameter c variation is depicted in Fig. 14 and here we can see that the fifth-order methods are clearly superior. At last, in Fig. 15 we show how the methods behave for different reference depths. Again, the performance is fairly uniform throughout all depths with a marked deterioration as the value $h = 0.8$ is approached.

5. Conclusion

In this paper we have proposed three families of generalizations to the Craig formula for recovering bottom topography from surface water wave measurements. We have found all to be relatively easy to implement with a Fourier collocation method, all extremely rapid in execution, and all superior to the formula of Craig in terms of accuracy. The simulations of Section 4 show that these methods are highly accurate and that additional accuracy can be realized by use of higher order versions of our formulas. At present these algorithms are not well-suited for direct application to open-ocean measurements as standing waves are difficult to extract from generic wavefield data. It is an ongoing project of the authors to devise new methods to glean this information from "real world" data.

Acknowledgements

DPN gratefully acknowledges support from the NSF through grant No. DMS–0537511.

References

[1] O.P. Bruno, F. Reitich, Numerical solution of diffraction problems: A method of variation of boundaries, J. Opt. Soc. Am. A 10 (6) (1993) 1168–1175.

[2] O.P. Bruno, F. Reitich, Numerical solution of diffraction problems: A method of variation of boundaries. II. Finitely conducting gratings, Padé approximants, and singularities, J. Opt. Soc. Am. A 10 (11) (1993) 2307–2316.

[3] O.P. Bruno, F. Reitich, Numerical solution of diffraction problems: A method of variation of boundaries. III. Doubly periodic gratings, J. Opt. Soc. Am. A 10 (12) (1993) 2551–2562.

[4] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, New York, 1988.

[5] M. Collins, W. Kuperman, Inverse problems in ocean acoustics, Inverse Problems 10 (1994) 1023–1040.

[6] D. Colton, R. Kress, Inverse Acoustic and Electromagnetic Scattering Theory, second ed., Springer-Verlag, Berlin, 1998.

[7] W. Craig, Personal communication, 1998.

[8] W. Craig, P. Guyenne, D.P. Nicholls, C. Sulem, Hamiltonian long wave expansions for water waves over a rough bottom, Proc. Roy. Soc. London, Ser. A 461 (2055) (2005) 839–873.

[9] W. Craig, D.P. Nicholls, Traveling gravity water waves in two and three dimensions, Eur. J. Mech. B Fluids 21 (6) (2002) 615–641.

[10] W. Craig, C. Sulem, Numerical simulation of gravity waves, J. Comput. Phys. 108 (1993) 73–83.

[11] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 26, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1977.

[12] S. Grilli, Depth inversion in shallow water based on nonlinear properties of shoaling periodic waves, Coastal Engineering 35 (1998) 185–209.

[13] P. Guyenne, D.P. Nicholls, Numerical simulation of solitary waves on plane slopes, Math. Comput. Simul. 69 (2005) 269–281.

[14] P. Guyenne, D.P. Nicholls, A high-order spectral method for nonlinear water waves over bottom topography, SIAM J. Sci. Comput. 30 (1) (2007) 81–101.

[15] D. Henderson, Personal communication, 1998.

[16] H. Lamb, Hydrodynamics, sixth ed., Cambridge University Press, Cambridge, 1993.

[17] D.P. Nicholls, M. Taber, Joint analyticity and analytic continuation for Dirichlet–Neumann operators on doubly perturbed domains, J. Math. Fluid Mech. 10 (2) (2008) 238–271.

[18] C. Piotrowski, J. Dugan, Accuracy of bathymetry and current retrievals from airborne optical time-series imaging of shoaling waves, IEEE Trans. on Geoscience and Remote Sensing 40 (12) (2002) 2606–2618.

[19] L. Rayleigh, On the dynamical theory of gratings, Proc. Roy. Soc. London, Ser. A 79 (1907) 399–416.

[20] S.O. Rice, Reflection of electromagnetic waves from slightly rough surfaces, Comm. Pure Appl. Math. 4 (1951) 351–378.

[21] R.A. Smith, An operator expansion formalism for nonlinear surface waves over variable depth, J. Fluid Mech. 363 (1998) 333–347.

[22] M. Taroudakis, G. Makrakis, Inverse Problems in Underwater Acoustics, Springer-Verlag, New York, 2001.

[23] V. Zakharov, Stability of periodic waves of finite amplitude on the surface of a deep fluid, J. Appl. Mech. Technical Phys. 9 (1968) 190–194.