

Journal Pre-proof

Three-dimensional waves under ice computed with novel preconditioning methods

C. Țugulan, O. Trichtchenko and E. Părau

PII: S0021-9991(22)00191-7
DOI: <https://doi.org/10.1016/j.jcp.2022.111129>
Reference: YJCPH 111129

To appear in: *Journal of Computational Physics*

Received date: 10 October 2021
Revised date: 25 February 2022
Accepted date: 3 March 2022

Please cite this article as: C. Țugulan, O. Trichtchenko and E. Părau, Three-dimensional waves under ice computed with novel preconditioning methods, *Journal of Computational Physics*, 111129, doi: <https://doi.org/10.1016/j.jcp.2022.111129>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2022 Published by Elsevier.



Highlights

- Three-dimensional flexural-gravity waves generated by moving pressures are computed.
- A novel hybrid preconditioning technique is derived.
- An iterative Newton-Krylov solver significantly increases the grid refinement.

Three-dimensional waves under ice computed with novel preconditioning methods

C. Țugulan^a, O. Trichtchenko^a, E. Părau^b

^a*Department of Physics and Astronomy, The University of Western Ontario, London, N6G2V4, Canada*

^b*School of Mathematics, University of East Anglia, Norwich, NR47TJ, UK*

Abstract

In this work, we present three-dimensional, nonlinear travelling wave solutions for water waves under a sheet of ice, *i.e.*, flexural-gravity waves. The ice is modelled as a thin elastic plate on top of water of infinite depth and the equations are formulated as a boundary integral method. Depending on the velocity of the moving disturbance generating the flow, different deflection patterns of the floating ice sheet are observed. In order to compute solutions as efficiently as possible, we introduce a novel hybrid preconditioning technique used in an iterative Newton-Krylov solver. This technique is able to significantly increase the grid refinement and decrease the computational time of our solutions in comparison to methods that are presently used in the literature. We show how this approach is generalisable to three-dimensional ice wave patterns in different velocity regimes.

Keywords: three-dimensional waves, flexural-gravity waves, preconditioner

1. Introduction

In this work, we present a novel method for efficiently and accurately computing solutions for three-dimensional waves under ice generated by a moving disturbance. The fluid beneath the sheet of ice is considered to be infinitely deep and we seek travelling wave solutions. The ice is modeled as an elastic sheet and its interaction with the fluid makes the resulting flexural-gravity (or hydroelastic) nonlinear free-surface problem particularly challenging mathematically, as well as computationally expensive [1, 2]. As such, we propose an efficient strategy to overcome this challenge.

The research is motivated by applications to transportation in cold regions, where the ice cover is used for roads or aircraft runways. This is of particular interest in Canada, where ice-covered lakes enable the routing of trucks on winter roads built partially on ice, which allows transport into regions which may not be accessible otherwise [3]. It could also be applied to study waves beneath a very large floating structure, such as the Mega-Float constructed in Japan as a prototype for a floating airport [4]. As these endeavours can be precarious, and occasionally result in the loss of life and equipment, understanding ice and gravity wave interactions has become increasingly important.

For this work, we assume that the ice forms a relatively thin layer on top of the fluid and behaves as a thin elastic plate. Furthermore, we assume that the ice bends with the waves and neglect the friction between the ice plate and the water, as well as the inertia of the thin plate. As such, we will use the bi-Laplacian model, which been used in other studies on three-dimensional flexural-gravity waves [5, 2]. While this model is often appropriate [6], other models can also be considered that take into account viscoelastic effects [7], or ice of finite thickness [8]. Since the focus of this work is to illustrate an efficient method for computing waves under ice, we restrict ourselves to the linear bi-Laplacian model and leave other possible models for future work.

The mathematical study of flexural-gravity waves has a long history, starting in the 19th century with the linear model devised by Greenhill for an ice sheet floating on top of a bulk fluid [9]. Many advances have been made since then, with the development of more sophisticated linear models [10, 11], as well as nonlinear elastic plate/nonlinear flow models [12, 13, 2, 14]. For some of these models the energy is conserved [13], while for others it is not (see [12, 14] for an overview).

It has been shown rigorously that solutions to these hydroelastic models exist for some values of parameters and in the absence of forcing. For example, a Lagrangian formulation for travelling waves has been used to prove the existence of solutions to these models in two-dimensions [15], a variational approach has been employed to show the existence of hydroelastic solitary waves [16], and the bifurcation theory for the existence of periodic waves in two-dimensions [17].

The difficulty involved in solving the set of fully nonlinear, three-dimensional equations with a moving boundary has brought on different approaches to simplify the problem. Here, we restrict our focus to those studies most relevant to our work. Several works have analysed both two and three-

dimensional flexural-gravity waves in weakly-nonlinear regimes. For example, focusing on mainly two-dimensional interactions using a Hamiltonian, a formalism was developed [18], deriving several model equations incorporating effects due to the ice plate. Using a Kirchhoff-Love model for the ice plate, a forced nonlinear Schrödinger equation was derived [19] to study the nonlinear effects of a moving load on a floating ice plate near a critical speed. Using a Hamiltonian formulation and a fully nonlinear model for the ice sheet, two-dimensional forced and unforced solutions were considered in both finite and infinite depth [20, 21]. Moreover, a fifth-order Korteweg-de Vries equation for two-dimensional hydroelastic waves on shallow water in channels was derived [22].

In three dimensions, a generalization of the Kadomtsev-Petviashvili equation was derived to model waves in the presence of additional surface effects [23], and to investigate solitary flexural-gravity waves [24]. Solitary waves in three dimensions have also been studied using a Benney-Roskes-Davey-Stewartson model for a fluid of arbitrary depth covered by an elastic sheet [14]. These asymptotic models show that flexural-gravity solitary waves do not bifurcate from zero amplitude solution in infinite depth. Fully localized three-dimensional solitary wave solutions have also been computed [25] using a fifth-order Hamiltonian model obtained from the fully nonlinear equations of motion. More recently, fully three-dimensional weakly nonlinear flexural-gravity waves were computed via Laplace transform methods, including some time-dependent wave patterns [26].

In this paper we are interested in solving the equations of motion describing travelling waves generated by a pressure distribution moving on the surface of a thin sheet of ice using a boundary integral technique. The localised pressure distribution is an idealized model for a vehicle travelling over a floating ice cover. The boundary integral technique was introduced initially for the problem of uniform flow past an ellipsoid [27], and generalized for three-dimensional free-surface flows past a source [28]. With this reformulation, the three-dimensional variable surface problem effectively reduces to a two-dimensional problem in terms of the variable surface and velocity potential. The resulting integro-differential equations and boundary conditions are discretized over the truncated (x, y) -plane and computed numerically using a mesh of $N \times M$ grid points. For a symmetric pressure distribution centred around the point $(0, 0)$ in the (x, y) -plane, N points in the horizontal x -direction are used to approximate $(-\infty, \infty)$ and M points in the horizontal y -direction approximate $(0, \infty)$. In this way, we exploit the symmetry of the

problem and only compute solutions over half of the truncated domain.

Over a series of works, this formulation has been employed to compute three-dimensional solutions for nonlinear gravity waves [29], forced and solitary capillary-gravity waves [30, 31, 32], as well as forced and solitary flexural-gravity waves using the linear model and the Plotnikov-Toland model to describe the ice sheet [5, 2]. To solve the discretized equations, the authors of these works typically relied on more traditional direct Newton-type methods, involving computations of inverses of a Jacobian matrix [33], which could take anywhere between several hours to several days to converge to a solution. Furthermore, these computations often require large amounts of system memory, thereby limiting the number of mesh points that can be used. This results in reduced accuracy and under-resolved solutions, especially when the pressure is moving close to a critical speed, c_{\min} .

More recently, a variation of the numerical scheme and boundary integral technique was employed by Pethiyagoda *et al.* to compute solutions for the nonlinear gravity-waves problem using a Jacobian-free Newton-Krylov (JFNK) method [34, 35]. This approach was shown to achieve significant performance gains allowing the grid refinement and size of the computational domain to be increased considerably. So far, these Newton-Krylov (NK) solvers have only been used to compute solutions for gravity waves problems [34, 35], such as the wake patterns behind ships, but have yet to be applied to the problem of flexural-gravity waves. A possible reason for this is that an effective preconditioner matrix is typically required for the convergence of NK methods [33]. In order to extend the framework of Pethiyagoda *et al.* [34], developed for gravity waves, to the problem of flexural-gravity waves, the contribution of the added flexural term to the Jacobian of the system must be considered in the construction of the preconditioner. The method of generating the preconditioner matrix for the flexural-gravity waves problem will be discussed in detail in §4.

In this work, we present a versatile method for generating the preconditioner, which will allow the efficient computation of flexural-gravity waves in a variety of parameter regimes and can be easily generalized for different problems and models for ice. To allow this flexibility, we formulate the problem in a general way. Using a novel hybrid preconditioning technique which includes nonzero entries of the linearized gravity waves problem, as well as nonzero entries from finite differences on the term describing the contribution due to the presence of the sheet of ice at the boundary in a Newton-Krylov method, we reduce the run-time of the calculations from several hours to just

minutes for the same number of grid points. This allows us to achieve higher accuracy than was previously possible, which we illustrate using a rigorous numerical error analysis.

The outline of our paper is as follows. In §2 we recast the problem by applying a boundary integral method, so that the governing equations are ultimately satisfied by solving a singular integro-differential equation and Bernoulli's equation subject to some radiation conditions. In §3 we discretize our governing equations to produce a nonlinear system of algebraic equations, which we solve using a NK method. The preconditioner we use is related to the linear problem, as we explain in §4. In §5 we present highly accurate, well resolved numerical solutions for the flexural-gravity problem. We explore grid dependence and analyze the error arising from under-refining solutions and truncating at the edges of the computational domain. The resolution of the mesh we use is up to the standard, and may even surpass, most three-dimensional schemes for flexural-gravity waves published in literature. Finally, we conclude our paper with a summary and discussion of how our methods can be adapted to consider other types of flows or models for ice.

2. Mathematical formulation

2.1. Governing equations

We consider three-dimensional irrotational flow of an inviscid, incompressible fluid of infinite depth and density ρ , bounded above by a continuous sheet of ice. Following [5], we suppose that the flow is due to a pressure distribution moving with constant speed, U , on the surface of a sheet of ice, upon which gravity, g , is also acting. We define the problem in Cartesian coordinates such that the disturbance travels in the negative x -direction and gravity acts in the negative z -direction, as shown in Figure 1.

The vertical deflection of the ice sheet-fluid interface is denoted by $z = \zeta(x, y, t)$ and $z = 0$ corresponds to the undisturbed interface. In terms of the fluid velocity potential $\Phi(x, y, z, t)$, the governing equations of the flow are

$$\nabla^2 \Phi = 0 \text{ on } -\infty < z < \zeta(x, y, t), \quad (1)$$

$$\zeta_t + \Phi_x \zeta_x + \Phi_y \zeta_y = \Phi_z \text{ on } z = \zeta(x, y, t), \quad (2)$$

$$\rho \Phi_t + \frac{\rho}{2} (\Phi_x^2 + \Phi_y^2 + \Phi_z^2) + \rho g \zeta + DP_{\text{flex}} = -p(x, y, t) \text{ on } z = \zeta(x, y, t), \quad (3)$$

$$|\nabla \Phi| \rightarrow 0 \text{ on } z \rightarrow -\infty, \quad (4)$$

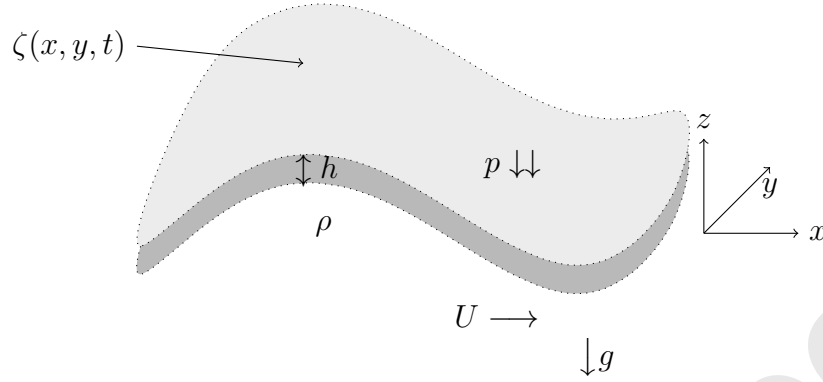


Figure 1: Sketch of the problem given by (1) - (4). The x -axis is aligned with the direction of travel of the pressure distribution. The fluid of infinite depth and density ρ is bounded above by the variable surface $z = \zeta(x, y, t)$ of the ice-fluid interface. The pressure distribution $p = p(x, y, t)$ is acting on the surface of the ice sheet of thickness h to generate the disturbance.

where the subscripts denote partial derivatives. Equation (1) describes the potential flow of an inviscid, irrotational fluid. Equations (2) and (4) are the kinematic boundary condition at ice-fluid interface and the boundary condition at infinite depth, respectively. Equation (3) is the dynamic boundary condition, obtained from the Bernoulli equation describing the pressure balance at the ice-fluid interface, where p is the external pressure distribution exerted on the surface and the term P_{flex} describes the effects due to the presence of the sheet of ice on the surface of the fluid and depends on the model. Radiation conditions are also imposed infinitely far upstream from the disturbance. Equation (4) requires that the flow approaches an undisturbed free stream at infinite depth.

Since the aim of this work is to illustrate an efficient method for computing waves under ice, as a proof of concept we choose to use the simplest linear thin elastic plate model for the pressure, DP_{flex} , exerted by the ice on the fluid (see e.g. [5]), with

$$P_{\text{flex}} = \nabla^4 \zeta = (\partial_{xxxx} + \partial_{yyyy} + 2\partial_{xxyy})\zeta. \quad (5)$$

The constant $D = Eh^3/12(1 - \nu^2)$ is the flexural rigidity of the plate, where E denotes Young's modulus of elasticity, ν is the Poisson ratio describing the effects of transverse strain relative to axial strain, and h is thickness of the ice sheet. There are many other possible models that could be used to define

P_{flex} , including the Kirchhoff–Love and Plotnikov–Toland nonlinear models examined previously (see [2, 13]). However, the linear model (5) is often appropriate [6], and serves as a good approximation when the wave patterns do not exhibit strong nonlinearity [2], such as those considered in §5.

We seek steady-state solutions of the problem (1)–(4) and choose a frame of reference moving with the disturbance. In the moving reference frame, our variables transform according to $\hat{x} = x + Ut$, $\hat{\zeta}(\hat{x}, y) = \zeta(x, y, t)$ and $\hat{\Phi}(\hat{x}, y, z) = \Phi(x, y, z, t) + Ux$. We drop the hat notation from now on for simplicity. We assume that in this moving frame the pressure has the localised form

$$p(x, y) = \begin{cases} P_0 e^{\frac{L^2}{(x^2-L^2)} + \frac{L^2}{(y^2-L^2)}}, & |x| < L \quad \text{and} \quad |y| < L \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where P_0 is a constant and L is the characteristic length scale defining the size of the support of the pressure.

The type of solutions for this problem depends on the speed, U , and form of the moving pressure distribution that models the disturbance, as well as the flexural rigidity, D , of the sheet of ice. From the dispersion relation for linear flexural-gravity waves in infinite depth [6], the value of the critical speed, where the group and phase speed are equal, is given by

$$c_{\min}^2 = \frac{4}{3} \left(\frac{3Dg^3}{\rho} \right)^{\frac{1}{4}}, \quad (7)$$

where ρ is the density of the fluid. When the velocity of the moving load on top of the floating ice plate is less than the minimum phase speed, c_{\min} , the disturbance is localized and the waves do not propagate away from the pressure, so the solution will approach uniform flow in the far-field when $U < c_{\min}$. When the speed of the moving pressure is greater than c_{\min} , a more complicated pattern emerges, with waves of different wavelengths appearing in front of and behind the moving pressure. The linear solution for $U > c_{\min}$ is characterized by trains of waves in the far field, with predominantly gravity waves downstream of the moving pressure and predominantly flexural waves upstream [10]. As such, the boundary and radiation conditions are harder to implement.

It is known that waves under a sheet of ice have properties similar to those of gravity-capillary waves (see [5]), so it is possible to add an artificial dissipative term in the dynamic boundary condition (3) of the form $\tilde{\mu}\Phi$, which

was initially introduced by Rayleigh [36] to analytically calculate linear solutions for the gravity-capillary waves problem. This artificial term represents the effect of small dissipative forces and is based on the assumption that the deviation of fluid particles from the uniform flow state is opposed by a force proportional to the relative velocity. It is not directly related to the real kinematic viscosity and does not change the irrotational character of motion in the bulk of the fluid (see [37], Art.242). This technique was successfully employed to compute forced nonlinear three-dimensional gravity-capillary waves [30] and later for flexural-gravity waves [5].

2.2. Nondimensionalization

The problem (1)-(4) is nondimensionalized by scaling all lengths with respect to L and all speeds with respect to U . This leads to the non-dimensional parameters, $\epsilon = P_0/(\rho U^2)$, the inverse of the length-based Froude number $F = gL/U^2$ which characterizes the nature of the solutions, and $\beta = D/\rho U^2 L^3$ which depends on the flexural rigidity D of the ice. The expression (6) in dimensionless variables becomes

$$p(x, y) = \begin{cases} e^{\frac{1}{(x^2-1)} + \frac{1}{(y^2-1)}}, & |x| < 1 \quad \text{and} \quad |y| < 1 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

In terms of the velocity potential function $\Phi(x, y, z)$ on the variable surface $z = \zeta(x, y)$, the dimensionless problem in the moving reference frame is formulated as follows:

$$\nabla^2 \Phi = 0 \quad \text{and} \quad x, y \in \mathbb{R}, -\infty < z < \zeta(x, y), \quad (9)$$

with the boundary conditions

$$\Phi_x \zeta_x + \Phi_y \zeta_y = \Phi_z \quad \text{on} \quad z = \zeta(x, y), \quad (10)$$

$$\frac{1}{2}(\Phi_x^2 + \Phi_y^2 + \Phi_z^2) + F\zeta + \mu(\Phi - x) + \beta P_{\text{flex}} + \epsilon p = \frac{1}{2} \quad \text{on} \quad z = \zeta(x, y). \quad (11)$$

Equation (11) is the dynamic condition with the artificial non-dimensional damping term $\mu > 0$ included. It can be shown using (7) that the minimum phase speed, $U = c_{\min}$, is reached for $F = gL/c_{\min}^2 = (3/4)\sqrt[3]{1/4\beta}$. We also impose that we have uniform flow both far upstream (the radiation condition)

from the disturbance, and at the bottom (infinitely far below the variable surface), providing the final two conditions

$$(\Phi_x, \Phi_y, \Phi_z) \rightarrow (1, 0, 0), \quad \zeta \rightarrow 0 \quad \text{as} \quad x \rightarrow -\infty \quad (12)$$

$$(\Phi_x, \Phi_y, \Phi_z) \rightarrow (1, 0, 0), \quad \text{as} \quad z \rightarrow -\infty. \quad (13)$$

At the downstream $x \rightarrow \infty$ and lateral $y \rightarrow \pm\infty$ boundaries, we assume that the amplitude of the waves have decayed to zero. The governing equation (9), subject to (10)-(13), with $\mu = 0$ define the exact formulation of the nonlinear problem with no known analytical solution.

2.3. Boundary integral method

In order to solve (9)-(13) numerically, we reformulate the problem using a boundary-integral equation method [28, 29]. For convenience, we define the velocity potential on the variable surface $z = \zeta(x, y)$ by setting

$$\phi(x, y) = \Phi(x, y, \zeta(x, y)). \quad (14)$$

Then, using the chain rule of calculus, the dynamic condition (11) can be rewritten using the kinematic condition (10) as the following local equation in terms of surface variables:

$$\frac{1}{2} \frac{(1 + \zeta_x^2)\phi_y^2 + (1 + \zeta_y^2)\phi_x^2 - 2\zeta_x\zeta_y\phi_x\phi_y}{1 + \zeta_x^2 + \zeta_y^2} + F\zeta + \mu(\phi - x) + \beta P_{\text{flex}} + \epsilon p = \frac{1}{2}, \quad (15)$$

which we refer to as the dynamic condition. Omitting the details and using the notation $\phi^* = \phi(x^*, y^*)$, $\zeta^* = \zeta(x^*, y^*)$ (and same for the derivatives), the reformulation results in the following boundary-integral equation for the variable surface $z = \zeta(x, y)$,

$$\begin{aligned} 2\pi(\phi^* - x^*) = & \int_0^\infty \int_{-\infty}^\infty (\phi - \phi^* - x - x^*) K_1(x, y; x^*, y^*) \, dx \, dy \\ & + \int_0^\infty \int_{-\infty}^\infty [K_2(x, y; x^*, y^*) \zeta_x - \zeta_x^* S_2(x, y; x^*, y^*)] \, dx \, dy \\ & + \zeta_x^* \int_0^\infty \int_{-\infty}^\infty S_2(x, y; x^*, y^*) \, dx \, dy, \end{aligned} \quad (16)$$

which holds for any point (x^*, y^*) in the (x, y) -plane (see [29] for details) and has been desingularized. The kernel functions K_1 and K_2 are

$$K_1(x, y; x^*, y^*) = \frac{\zeta - \zeta^* - (x - x^*)\zeta_x - (y - y^*)\zeta_y}{((x - x^*)^2 + (y - y^*)^2 + (\zeta - \zeta^*)^2)^{3/2}}, \quad (17)$$

$$K_2(x, y; x^*, y^*) = \frac{1}{\sqrt{(x - x^*)^2 + (y - y^*)^2 + (\zeta - \zeta^*)^2}}, \quad (18)$$

and S_2 is

$$S_2(x, y; x^*, y^*) = \frac{1}{\sqrt{A(x - x^*)^2 + B(x - x^*)(y - y^*) + C(y - y^*)^2}}, \quad (19)$$

with

$$A = 1 + \zeta_x^{*2}, \quad B = 2\zeta_x^*\zeta_y^*, \quad C = 1 + \zeta_y^{*2}.$$

The last integral in (16) contains the singularity and can be evaluated analytically in terms of logarithms [29]. Symmetry in the y -direction with $\zeta(x, y) = \zeta(x, -y)$ and $\phi(x, y) = \phi(x, -y)$ implies that we can solve the set of equations given by the dynamic condition (15) and desingularized integro-differential equation (16) on half of the domain. We use the method of images to account for the symmetry, which introduces an additional term in each kernel (17), (18), and in (19) (see [2] for more details). The final form of equations to solve for flexural-gravity waves in infinite depth is thus given by the desingularized integral equation (16) and the dynamic boundary condition (15) with the unknowns ϕ and ζ .

2.4. Linearized Problem

While the focus here is on calculating numerical solutions to the nonlinear problem, analytical approximations can be obtained by linearizing equations (9)-(13) around a uniform flow. The main reason for us to pursue the linearized problem is to construct a preconditioner for the nonlinear problem. The linearized problem can be formulated by substituting $\Phi = x + \phi$ into equations (9)-(13) and then dropping the nonlinear terms in ϕ and ζ . Omitting the details, the linearized problem is to solve Laplace's equation

$$\nabla^2 \phi = 0 \quad \text{for } x, y \in \mathbb{R}, \quad -\infty < z < 0, \quad (20)$$

subject to the linearized kinematic and dynamic conditions (10)-(11), which become

$$\zeta_x = \phi_z \quad \text{on} \quad z = 0, \quad (21)$$

$$\phi_x - 1 + F\zeta + \mu\phi + \beta P_{\text{flex}} + \epsilon p = 0 \quad \text{on} \quad z = 0, \quad (22)$$

and impose the upstream radiation condition and condition at infinite depth, which remain the same as for the nonlinear case.

The solution of the linearized problem can then be found using Laplace transforms [26] or using Fourier transforms for the corresponding free-surface problem [38]. However, we will consider the equivalent boundary-integral approach as that used for the nonlinear problem. The application of Green's second formula gives

$$\begin{aligned} 2\pi(\phi^* - x^*) &= \int_0^\infty \int_{-\infty}^\infty (\zeta_x - \zeta_x^*) K_3(x, y; x^*, y^*) \, dx \, dy \\ &\quad + \zeta_x^* \int_0^\infty \int_{-\infty}^\infty K_3(x, y; x^*, y^*) \, dx \, dy, \end{aligned} \quad (23)$$

where

$$K_3(x, y; x^*, y^*) = \frac{1}{\sqrt{(x - x^*)^2 + (y - y^*)^2}}. \quad (24)$$

We note that the singularity in (23) is dealt with in the same way as with the nonlinear problem and symmetry in the y -direction introduces an additional term in (24).

3. Numerical method

To solve the system of equations (15)-(16), we use the approach introduced in [34]. The numerical discretization leads to a nonlinear system of equations of the form

$$\mathbf{E}(\mathbf{u}) = \mathbf{0}, \quad (25)$$

where \mathbf{u} is a vector of unknowns, which is comprised of the unknown wave profile and surface velocity potential, as well as their derivatives, evaluated at discrete points. The system of equations (25) is then solved with a Newton-Krylov method. For each Newton iteration, the current estimate of the solution, \mathbf{u}^n , is replaced by a new estimate,

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \eta^n \delta \mathbf{u}^n,$$

where $\eta^n \in (0, 1]$ is the damping parameter chosen via line-search [39], and $\delta \mathbf{u}^n$ is the Newton correction. The Krylov iterations find a suitable $\delta \mathbf{u}^n$ which satisfies the linear system

$$\mathbf{J}(\mathbf{u}^n) \delta \mathbf{u}^n = -\mathbf{E}(\mathbf{u}^n), \quad (26)$$

where \mathbf{u}^n is the n th nonlinear iterate in the sequence $\{\mathbf{u}^n\}_{n=0}^{\infty} \rightarrow \mathbf{u}$ approximating the solution and $\mathbf{J}(\mathbf{u}^n) = \partial \mathbf{E}(\mathbf{u}^n) / \partial \mathbf{u}^n$ is the Jacobian matrix [33].

The advantage of using Krylov subspace methods as linear solvers in conjunction with nonlinear Newton iterations, is their ability to compute a Newton iteration without explicitly forming the Jacobian in (26). Instead, the action of the Jacobian is only required in the form of matrix-vector products that build a basis for the Krylov subspace needed to find $\delta \mathbf{u}^n$ [33]. The rate of convergence of Krylov subspace methods depends strongly on the conditioning of the iteration matrix $\mathbf{J}(\mathbf{u}^n)$, which is very ill-conditioned in our case due to the nonlocal nature of the integrals arising from the boundary integral reformulation described in §2.3. As a consequence, the Krylov method converges very poorly, if at all, unless the system is preconditioned appropriately.

The purpose of the preconditioner matrix, \mathbf{P} , is to reduce the number of Krylov iterations by reducing the dimension of the Krylov subspace needed to find a sufficiently accurate value for $\delta \mathbf{u}^n$ [40, 33]. As such, \mathbf{P} should be constructed as an approximation to the full Jacobian of the nonlinear system (25), denoted by $\mathbf{J}_n = \mathbf{J}(\mathbf{u}^n)$, such that $\mathbf{P} \approx \mathbf{J}_n$. In this way, the condition number is reduced for the preconditioned Jacobian with $\mathbf{J}_n \mathbf{P}^{-1} \approx \mathbf{I}$, where \mathbf{I} is the identity matrix. In §4, we discuss how the preconditioning method of Pethiyagoda *et al.* [34], developed for gravity waves, can be extended to include the contribution from the flexural term in the dynamic boundary condition (15), which is required to build an effective preconditioner for the flexural-gravity problem.

3.1. Numerical discretization scheme

Equations (15) and (16) are nonlinear and have no known general analytical solutions. As such, we must solve these equations numerically. To do this, the infinite half plane is truncated such that $-\infty < x < \infty$ becomes $x_1 < x < x_N$ and $0 < y < \infty$ becomes $y_1 < y < y_M$ (where we have taken advantage of symmetry). We introduce the mesh points,

$$x_k = x_1 + (k - 1)\Delta x \quad \text{and} \quad y_\ell = (\ell - 1)\Delta y, \quad (27)$$

for $k = 1, \dots, N$ and $\ell = 1, \dots, M$, where the point x_1 is chosen as a suitably far-upstream value such that the waves are considered to decay to approximately zero amplitude there and solutions are reflected in the $y = y_1$ axis of symmetry. With the mesh points defined according to (27), the functions $\zeta(x, y)$ and $\phi(x, y)$ can be represented by discrete values $\zeta_{k,\ell}$ and $\phi_{k,\ell}$ evaluated at the mesh points (x_k, y_ℓ) . The vector of $2(N+1)M$ unknowns is

$$\mathbf{u} = \left[\phi_{1,1}, (\phi_x)_{1,1}, \dots, (\phi_x)_{N,1}, \phi_{1,2}, (\phi_x)_{1,2}, \dots, \phi_{1,M}, (\phi_x)_{1,M}, \dots, (\phi_x)_{N,M}, \right. \\ \left. \zeta_{1,1}, (\zeta_x)_{1,1}, \dots, (\zeta_x)_{N,1}, \zeta_{1,2}, (\zeta_x)_{1,2}, \dots, \zeta_{1,M}, (\zeta_x)_{1,M}, \dots, (\zeta_x)_{N,M} \right]. \quad (28)$$

The values of the functions $\zeta_{1,\ell}$ and $\phi_{1,\ell}$ at the upstream boundary of the truncated domain are placed before the derivatives of the corresponding slice. The particular ordering of the variables in (28) follows the convention in [34] and allows us to optimize our scheme for a more sparse Jacobian matrix, but other choices are possible.

Given the elements in the vector of unknowns (28), the remaining values of ζ and ϕ are first obtained by trapezoidal-rule integration using the values of ζ_x and ϕ_x . The values ζ_y and ϕ_y are then calculated using second-order accurate finite differences [41]. We evaluate the boundary integral equation (16) at the half-mesh points $(x_{k+\frac{1}{2}}, y_\ell)$, $k = 1, 2, \dots, N-1$ and $\ell = 1, \dots, M$ by using two-point interpolation for the half-mesh points producing $(N-1)M$ equations to avoid vanishing denominators [28]. An additional $(N-1)M$ equations come from evaluating the dynamic condition (15) at the half-mesh points.

The last $4M$ equations enforce the radiation condition (12) on the upstream boundary of the truncated domain and are given by

$$\phi_{1,\ell} = x_1, \quad (\phi_x)_{1,\ell} = 1, \quad \zeta_{1,\ell} = 0, \quad (\zeta_x)_{1,\ell} = 0, \quad (29)$$

for $\ell = 1, \dots, M$. There are various other possible candidates for the radiation conditions (see [42, 34], for example). However, certain radiation conditions may risk eliminating or dampening waves that represent real physical solutions for the flexural-gravity problem. For this reason, we choose the simple radiation conditions in (29), which have also been used in previous studies on three-dimensional flexural-gravity waves [5, 2]. With these conditions, we now have $2(N+1)M$ nonlinear equations that make up the vector \mathbf{E} in (25) for the $2(N+1)M$ unknowns in (28).

Since the system of discretized equations (25) is solved using a Newton-Krylov method, we need a good initial guess, \mathbf{u}^0 in (26). In the regimes we have examined, it was sufficient to use a flat surface as the initial guess, given by

$$\phi_{1,\ell} = x_1, \quad (\phi_x)_{k,\ell} = 1, \quad \zeta_{1,\ell} = 0, \quad (\zeta_x)_{k,\ell} = 0, \quad (30)$$

for $k = 1, \dots, N$ and $\ell = 1, \dots, M$. A variant of the restarted generalised minimum residual method [43] is used to solve the system (26), where we used the default 6×10^{-6} absolute tolerance for the residual.

3.2. Sources of error

In this work, we consider two overarching sources of error. The first is the truncation error introduced when approximating the infinite domain of integration in (16) with a finite domain. This truncation error is particularly obvious in the $U > c_{\min}$ regime, where solutions are characterized by waves in front and behind the source of the disturbance. The second main source of error can be attributed to insufficient grid refinement, which has an affect on the accuracy of the solutions. We examine both of these sources of error in §5.

4. Preconditioner

To speed up the convergence of the numerical methods, we utilise a preconditioner. The preconditioner is a matrix \mathbf{P} , that approximates the Jacobian of the nonlinear problem, which can be applied to reduce the condition number. Its effectiveness can be illustrated by a clustering of the eigenvalues of the preconditioned Jacobian $\mathbf{J}_n \mathbf{P}^{-1} \approx \mathbf{I}$ [40]. There are many ways to construct a relevant preconditioner, but for our work we consider a simplified form of the problem [33]. Following the framework for gravity waves in [34], we construct the preconditioner, \mathbf{P} , as the Jacobian matrix of the linearized problem for flexural-gravity waves, where it becomes necessary to consider the additional effects due to the presence of the ice plate. To construct the Jacobian of the linear problem, we apply the discretization scheme described in §3.1 to the linear problem derived in §2.4 and obtain a system of $2(N+1)M$ equations for the $2(N+1)M$ unknowns in (28). The novelty of our contribution is in our treatment of the flexural term, P_{flex} , in the Jacobian of the system, which is required to build an effective preconditioner.

Different possibilities for the preconditioner are shown in Figure 2, where the Jacobian matrix of the nonlinear problem for flexural-gravity waves is

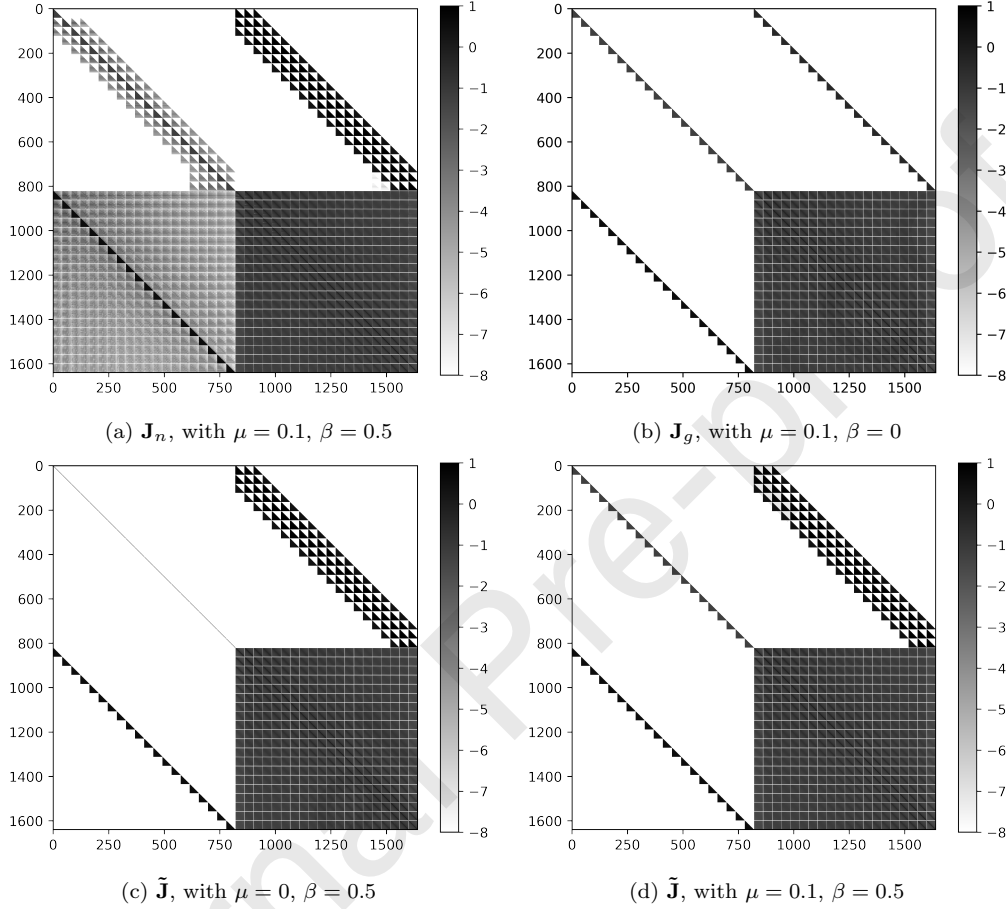


Figure 2: Visualization of the orders of magnitude of entries of the Jacobians on a log scale: $\log_{10} |J_{i,j}|$. Entries with larger magnitudes are represented by darker shades and zero entries are white. These are computed using mesh of size 40×20 with $\Delta x = \Delta y = 0.6$, $x_1 = -12$, and non-dimensional parameters $\epsilon = 1$, $L = 1$, $F = 0.7$. (a) The Jacobian for the nonlinear flexural-gravity problem with $\beta = 0.5$ and $\mu = 0.1$, computed numerically for \mathbf{u}^n using centered differences. (b) The Jacobian of the linearized problem for gravity waves with $\beta = 0$ and $\mu = 0.1$, determined analytically by explicit differentiation of the linearized equations. The Jacobian, $\tilde{\mathbf{J}}$, for the linear flexural-gravity problem is computed using the hybrid approach for \mathbf{u}^0 , with $\beta = 0.5$ when (c) $\mu = 0$ and (d) including artificial viscosity $\mu = 0.1$.

compared to its counterparts for the linear problem, computed with different values of parameters β and μ . The magnitude of the entries of the Jacobians in Figure 2 are shown on a log scale, where the larger magnitude entries are represented by darker shades and the zero entries are white. For each Jacobian in Figure 2, the other non-dimensional parameters are $\epsilon = 1$, $L = 1$, and $F = 0.7$.

We first consider the Jacobian matrix of the nonlinear problem, \mathbf{J}_n , which we wish to approximate. The Jacobian \mathbf{J}_n in Figure 2(a) was computed numerically using centered differences for the solution \mathbf{u}^n satisfying the system of equations (25) with non-dimensional parameters $F = 0.7$, $\beta = 0.5$ and $\mu = 0.1$. This direct method of differentiation is very computationally intensive since it requires each component of \mathbf{E} to be evaluated for at least $2(N+1)M+1$ sets of independent variables. With the particular ordering of equations and unknowns in (28) (see [34] for details), the dynamic boundary condition (15) contributes to the top-half of the Jacobian and is local, yielding a sparse block-diagonal structure. The contribution from the artificial viscosity, μ , is seen in the top-left submatrix. The entries corresponding to the flexural term, P_{flex} , that describes the effects due to the presence of the ice plate, appear in the top-right submatrix and have a relatively large magnitude, which is important for preconditioning purposes. The bottom-half corresponds to the nonlocal boundary integral equations (16) which give a dense submatrix structure, predominantly in the bottom-right submatrix.

Without the flexural contribution from the ice plate (*i.e.*, by setting $\beta = 0$ in (22)), the linear problem derived in §2.4 allows for easy differentiation by hand and all elements of the Jacobian, \mathbf{J}_g , can be determined analytically. This requires considerably less computational time than directly evaluating the differentiation for the entire system numerically. The structure of the fully analytical Jacobian, \mathbf{J}_g , for the linearized problem with $\beta = 0$ and $\mu = 0.1$ is shown in Figure 2(b). While it shares some common structural features with the Jacobian of the nonlinear problem in Figure 2(a) (*e.g.*, lower-right dense submatrix and sparse lower-diagonal blocks), the key difference is in the top-right submatrix which lacks the additional non-zero entries from the flexural contribution that appear above and below the main diagonal.

For flexural-gravity waves, we need to consider the contribution from the flexural term, P_{flex} , to the Jacobian of the nonlinear system, which appears dominantly in the top-right submatrix of \mathbf{J}_n in Figure 2(a). Due to the relatively large magnitude of these entries, the Jacobian, \mathbf{J}_g , for the linearized gravity waves problem is not an adequate preconditioner since it fails to

reduce the dimension of the Krylov subspace, which is illustrated by the spectrum of eigenvalues in Figure 3(b). Instead, we use a hybrid approach to approximate the Jacobian, $\tilde{\mathbf{J}}$, for the linear flexural-gravity waves problem. In the hybrid approach, the flexural contribution to the Jacobian, \mathbf{J}_{flex} , is calculated numerically using centered differences on P_{flex} (for \mathbf{u}^0 corresponding to a flat surface). The flexural contribution, \mathbf{J}_{flex} , is then added to the appropriate entries of the analytical Jacobian, \mathbf{J}_g , of the linearized problem with $\beta = 0$ (*i.e.*, $\tilde{\mathbf{J}} = \mathbf{J}_g + \mathbf{J}_{\text{flex}}$).

Using the hybrid method, the Jacobian, $\tilde{\mathbf{J}}$, of the linearized flexural-gravity problem, computed with $\beta = 0.5$, is shown in Figure 2(d) with artificial viscosity $\mu = 0.1$ and in Figure 2(c) without artificial viscosity $\mu = 0$. Comparison reveals that inclusion of artificial viscosity contributes significantly to the structure of the Jacobian and should be included in the construction of the preconditioner. Comparison of Figure 2(a) and Figure 2(d) reveals the similarity between the Jacobian, \mathbf{J}_n , of the nonlinear problem and the Jacobian, $\tilde{\mathbf{J}}$, of the corresponding linear problem, with the same parameter values $\beta = 0.5$ and $\mu = 0.1$, obtained using the hybrid approach. The visualization shows that $\tilde{\mathbf{J}}$ shares the same structural features as \mathbf{J}_n , while being much easier to form as analytical formulae exist for the \mathbf{J}_g derivatives which are provided in their exact form in §Appendix B and derived following the framework of Pethiyagoda *et al.* [34].

A major advantage of computing the flexural contribution, \mathbf{J}_{flex} , numerically is that other models for ice can easily be considered without needing to modify the code that generates the preconditioner, $\mathbf{P} = \tilde{\mathbf{J}}$. While modest time savings could perhaps be achieved by evaluating the flexural contribution to the Jacobian analytically, the time needed to compute \mathbf{J}_{flex} numerically using a finite difference routine that avoids the nonlocal integral equations is negligible compared to the time required to compute the dense submatrix, $\tilde{\mathbf{J}}_{2,2} = (\mathbf{J}_g)_{2,2}$, analytically. To maintain generalizability to other models where the sparsity structure of \mathbf{J}_{flex} is unknown, we choose to keep $M(N + 1)$ function evaluations in our finite difference routine instead of optimizing for the specific case of the linear model considered here. In our view, the flexibility allowed by computing \mathbf{J}_{flex} numerically, and ease with which the hybrid preconditioning method can be applied to consider different models and differentiation schemes, outweighs the modest performance gains that would be obtained by calculating \mathbf{J}_{flex} analytically, especially when programming labour is taken into account.

An additional advantage is that, since analytical formulae exist for all entries of \mathbf{J}_g , these can be calculated independently and allow for the possibility of parallelization, which we leave for future work. We are also able to exploit the 2×2 submatrix structure of the preconditioner and form each submatrix separately. This is useful as the dense bottom-right submatrix, $\tilde{\mathbf{J}}_{2,2}$, does not depend on the parameters β , μ , and F , which we are interested in changing, and can be reused for a variety of problems. In this way, we remove the burden of computing the most costly part of the preconditioner each time we wish to explore different parameters on a given mesh. Moreover, the submatrices $\tilde{\mathbf{J}}_{1,1}$, $\tilde{\mathbf{J}}_{1,2}$, $\tilde{\mathbf{J}}_{2,1}$ each have an underlying block diagonal structure comprising identical lower-diagonal blocks within each submatrix. These lower diagonal blocks share a common form determined by (B.14), and only differ from each other by a constant coefficient, which allows easy computation and storage. For clarity, we emphasize that the flexural contribution only enters into the top-right submatrix, *i.e.*, $\tilde{\mathbf{J}}_{1,2} = (\mathbf{J}_g)_{1,2} + \mathbf{J}_{\text{flex}}$.

In Figure 3, we check the effectiveness of the preconditioner by comparing the eigenvalues, λ , of the Jacobian, \mathbf{J}_n , for the nonlinear problem without any preconditioning (Figure 3(a)), and those of the preconditioned Jacobian, $\mathbf{J}_n \mathbf{P}^{-1}$, for the different possible \mathbf{P} matrices, where the eigenvalues were computed using Python's `scipy.linalg.eig` module. The result of using a preconditioner that does not consider the flexural component, such as the fully analytical Jacobian, \mathbf{J}_g , for the linear free-surface problem, is illustrated in Figure 3(b) and appears to make the system more ill-conditioned.

The effect of preconditioning with $\mathbf{P} = \tilde{\mathbf{J}}$, the Jacobian of the corresponding linear problem which accounts for the flexural contribution using the hybrid method, is seen in Figure 3(c) and Figure 3(d). The effect is a tight clustering of eigenvalues around $\text{Re}\{\lambda\} = 1$ when the flexural contribution is included in the structure of the preconditioner. By comparing Figure 3(c) when $\mathbf{P} = \tilde{\mathbf{J}}_{\mu=0}$ is computed without artificial viscosity ($\mu = 0$), and Figure 3(d) when $\mathbf{P} = \tilde{\mathbf{J}}_{\mu=0.1}$ includes the contribution from artificial viscosity, $\mu = 0.1$, we see that the tightest clustering around $\text{Re}\{\lambda\} = 1$ occurs when the contribution from artificial viscosity is included in the preconditioner structure. This tight clustering of eigenvalues is desirable since it will result in a smaller Krylov subspace required to find a solution to the Newton correction $\delta \mathbf{u}^n$. The tight clustering of eigenvalues around $\text{Re}\{\lambda\} = 1$ in Figure 3(d) thus confirms the effectiveness of our choice $\mathbf{P} = \tilde{\mathbf{J}}$ for the preconditioner.

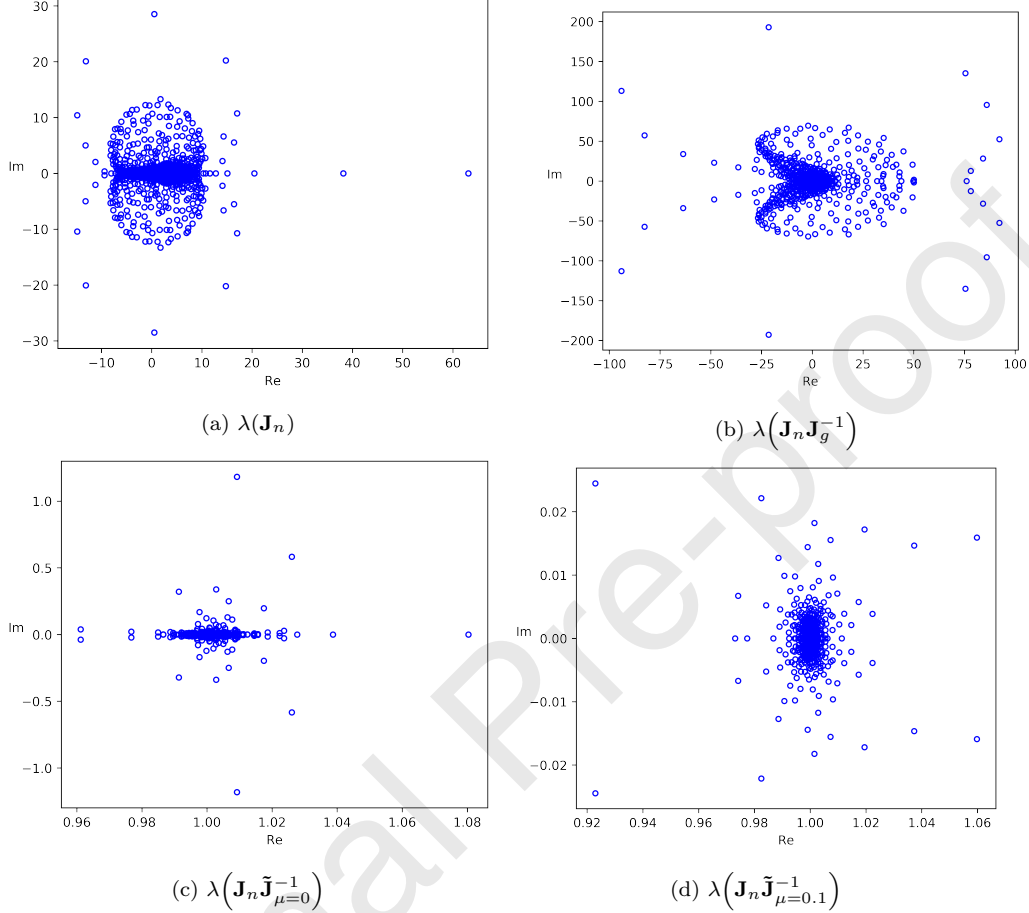


Figure 3: Visualization of the eigenvalues of the Jacobian, \mathbf{J}_n , of the nonlinear problem computed for \mathbf{u}_n (solution) with $\beta = 0.5$ and $\mu = 0.1$ (a) without any preconditioning, and the preconditioned Jacobian $\mathbf{J}_n \mathbf{P}^{-1}$ shown for different \mathbf{P} matrices, all computed for non-dimensional parameters $\epsilon = 1$, L , $F = 0.7$, on a 40×20 mesh with $\Delta x = \Delta y = 0.6$, $x_1 = -12$. (b) Preconditioning with $\mathbf{P} = \mathbf{J}_g$, the analytical Jacobian of the linear gravity waves problem with $\beta = 0$ and $\mu = 0.1$, yields a more disperse spectrum of eigenvalues. The effect of preconditioning with $\mathbf{P} = \tilde{\mathbf{J}}$, the Jacobian of the linear flexural gravity problem approximated using the hybrid method with $\beta = 0.5$, is seen in (c) when $\mathbf{P} = \tilde{\mathbf{J}}_{\mu=0}$ does not include artificial viscosity ($\mu = 0$), and in (d) when $\mathbf{P} = \tilde{\mathbf{J}}_{\mu=0.1}$ includes the contribution from artificial viscosity $\mu = 0.1$. The effect is a tight clustering of eigenvalues around $\text{Re}\{\lambda\} = 1$ when the flexural contribution is included in the structure of the preconditioner. The tightest clustering is obtained when the contribution from μ is also included.

5. Numerical results

We have computed solutions using a standard laptop computer¹ with the code written in Python. In all cases, the `scipy.optimize` implementation of nonlinear solvers from the SciPy library was used. We have computed solutions using both `fsolve` and `newton_krylov`, with the LGMRES method selected for the preconditioned inner Krylov iterations. For the `newton_krylov` solver, we compared two different methods for generating the preconditioner, \mathbf{P} , used in the Krylov iterations. The direct method for generating \mathbf{P} involves computing centred differences for the entire system numerically, which is very computationally intensive. The hybrid preconditioning method is as described in §4 and requires considerably less time to generate the preconditioner compared to the fully numerical direct method. The summary of computation time for these different solvers and methods of generating the preconditioner are given in Table 1, where it is clear that the fastest time is a couple of minutes using the Newton-Krylov method with a hybrid preconditioner compared to the hours it would take using a direct Newton solver (`fsolve`).

Solver (method)	Preconditioner (method)	Time to generate \mathbf{P} (h : min : s)	Time to solve (h : min : s)	Total time (h : min : s)
<code>fsolve</code>	—	—	08:45:02	08:45:32
<code>newton_krylov</code>	direct	08:11:11	00:00:20	08:11:31
<code>newton_krylov</code>	hybrid	00:01:08	00:00:40	00:01:48

Table 1: Performance of different solvers and preconditioning methods for solutions computed on a 80×40 mesh with spacing $\Delta x = \Delta y = 0.6$, and parameters $\epsilon = 1$, $F = 0.7$, $\beta = 0.5$, and $\mu = 0$.

For the results that follow, we recall that an $N \times M$ mesh comprises N grid points in the x -direction and M grid points in the y -direction. The parameter values we focused on are $F = 0.70$ in the $U < c_{\min}$ regime and $F = 0.35$ in the $U > c_{\min}$ regime. The other dimensionless parameters are

¹Intel Core i5-8265U CPU with 1.80 GHz processor and 8 GB of system memory

$\epsilon = 1$ and $L = 1$, which are representative of a moderately nonlinear regime and a symmetric pressure distribution. We choose $\beta = 0.5$ for the ice plate, which corresponds to $F = (3/4)\sqrt[3]{1/2} \approx 0.595$ when $U = c_{\min}$.

In this section, we present some typical computations for forced waves with $U < c_{\min}$, then we discuss some results for forced waves with $U > c_{\min}$. We also consider the accuracy of our results by comparing solutions computed on different $N \times M$ meshes as we vary the number of grid points and size of the grid intervals. In order to explore grid independence, one technique we use is to compute solutions on a given truncated domain and increase the number of grid points between successive iterations of the solver (*e.g.*, twice as many each time), then test whether grid refinement has significantly altered the solution by computing the error as the maximum difference between successive refinements. We employ a similar approach to qualitatively determine the effects the truncated domain has on our numerically computed solution and present the results below.

5.1. Subcritical case

For localized solutions that appear in the regime $U < c_{\min}$, we use the dynamic boundary condition (15), without artificial viscosity (*i.e.*, $\mu = 0$), and parameters $F = 0.7$ and $\beta = 0.5$. For $U \ll c_{\min}$, the deformation of the ice sheet is highly localized near the support of the pressure distribution (Figure 4) and for symmetric pressure distributions with respect to $x = 0$, the solutions are also symmetric. As expected, the steady displacement of the variable surface $\zeta(x, y)$ of the ice-fluid interface is similar to the static deformation of an ice plate under a load [6].

To explore the extent of grid dependence, we compare the solution shown in Figure 4 to the same parameter solution computed on both a coarser and finer mesh with the same number of grid points, as shown in Figure 5. Comparison of the centrelines in Figure 5 reveals that the solutions are clearly grid dependent and the amplitudes of the waves are overestimated by solutions computed on the coarser grid. The size of the truncated domain appears to have a comparatively minor effect on solutions in this regime since the disturbance is localized and the waves do not propagate away from the pressure. We also note that there appears to be very good agreement between the solutions computed for $\Delta x = \Delta y = 0.3$ and $\Delta x = \Delta y = 0.15$. The effect of grid spacing on the amplitude of the waves for a fixed truncated domain is illustrated in more detail in Figure 6.

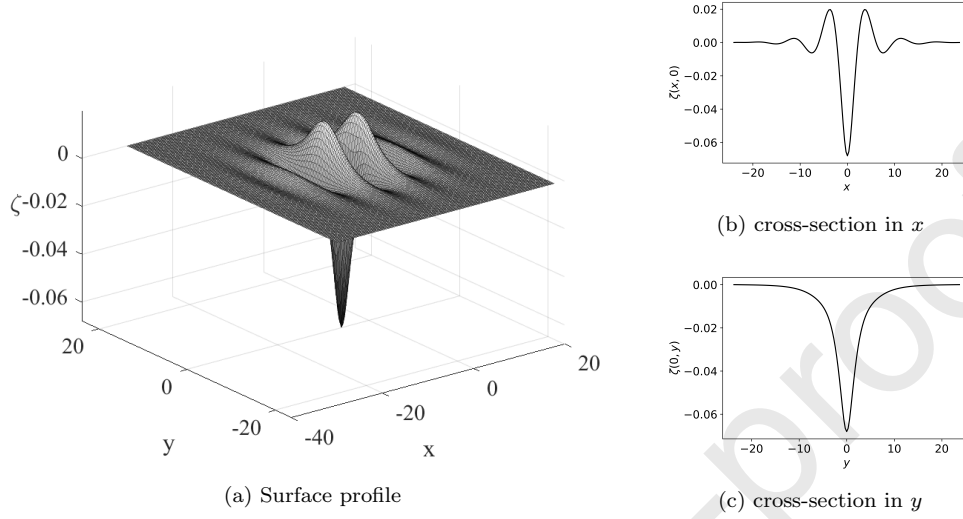


Figure 4: Forced wave solution when $U < c_{\min}$ computed on a 160×80 mesh with spacing $\Delta x = \Delta y = 0.3$, and $x_1 = -24$. The dimensionless parameters are $F = 0.7$ and $\beta = 0.5$.

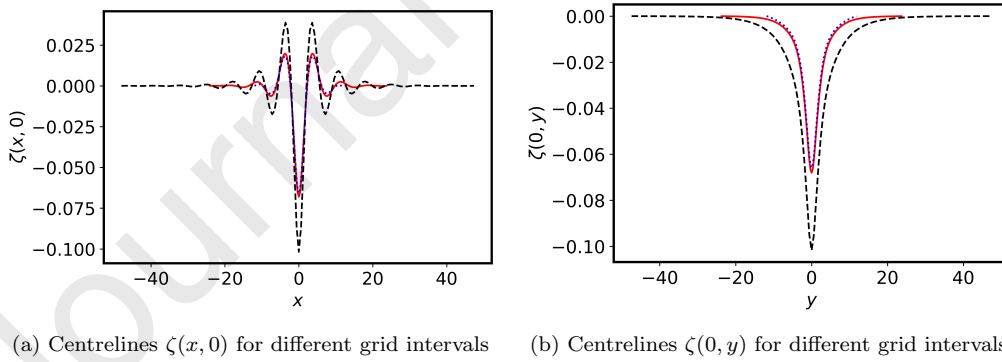


Figure 5: Comparison between forced waves solutions, with $\Delta x = \Delta y = 0.15$, $x_1 = -12$ (dotted line), $\Delta x = \Delta y = 0.3$, $x_1 = -24$ (solid line) and $\Delta x = \Delta y = 0.6$, $x_1 = -48$ (dashed line), when $U < c_{\min}$ computed on a 160×80 mesh. For both solutions, the dimensionless parameters are $F = 0.7$ and $\beta = 0.5$.

The characteristic localized nature of solutions in this regime makes it ideal for exploring the effect of mesh refinement on the accuracy of solutions, since solutions can be computed on a smaller truncated domain without significantly altering their surface profile. The accuracy of solutions in this regime is determined by computing solutions on a fixed truncated domain: $-12 < x < 12$, $0 < y < 12$ (chosen to capture the main features of the solution and impose the radiation conditions accurately) and doubling the number of grid points between successive solutions. Solutions are computed for the following mesh sizes: 40×20 , 80×40 , 160×80 , 320×160 , corresponding to uniform grid spacing $\Delta x = \Delta y$: 0.6, 0.3, 0.15, 0.075, respectively.

The error is plotted in Figure 7, with each point corresponding to the L_∞ -norm of the difference between successive solutions. For a given uniform grid spacing denoted by $\Delta_k \equiv \Delta x = \Delta y$, the error is calculated as

$$e_k = \max_{ij} |\zeta_{\Delta_{k+1}}(x_i, y_j) - \zeta_{\Delta_k}(x_i, y_j)|,$$

where the difference is taken between two consecutive refinements (*i.e.*, $\Delta_k/\Delta_{k+1} = 2$) of the solutions ζ evaluated at the same (x_i, y_j) coordinates of the coarse grid with spacing Δ_k . The semi-log plot of the L_∞ -norm of differences between solutions in Figure 7 shows that the error decays exponentially. This exponential decrease in error suggests that we may be able to approach grid independence of our solutions and is distinguishable from the error due to our second-order accurate finite difference schemes, which would be expected to follow a power law.

5.2. Supercritical case

In this regime, the artificial viscosity $\mu = 0.1$ is included in the dynamic boundary condition (15), and its effect has already been discussed in previous works [5]. A typical solution for $U > c_{\min}$ is presented in Figure 8 showing the two different wave patterns expected: one ahead of the pressure with shorter waves, and behind the pressure with longer waves. We show only the results for $F = 0.35$, but it is known from previous works that the waves behind the pressure are confined to an angle that varies with the velocity of the moving pressure [5, 10]. Figure 8 illustrates how the waves in front of the moving load are mainly flexural waves and are curved around the support of the pressure. The highest point of positive elevation is shown in Figure 8(b) to occur just in front of the pressure.

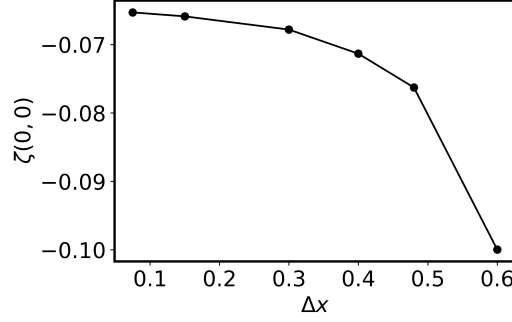


Figure 6: Comparison of solutions for different uniform grid spacing $\Delta x = \Delta y$ shows variation in the maximum amplitude $\zeta(0,0)$ of solutions $F = 0.7$ computed on the fixed truncated domain $-12 < x < 12$, $0 < y < 12$. This suggests larger grid spacings overestimate the amplitude of the solution.

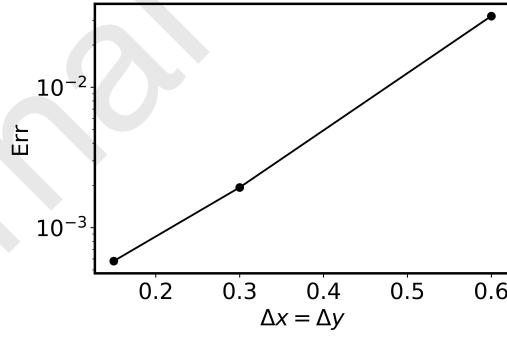


Figure 7: Error calculated as L_∞ -norm of differences between solutions computed on the fixed truncated domain: $-12 < x < 12$, $0 < y < 12$ for $N \times M$: 40×20 , 80×40 , 160×80 , 320×160 meshes (doubling N, M between successive solutions) with uniform grid spacing $\Delta x = \Delta y$: 0.6, 0.3, 0.15, 0.075, respectively.

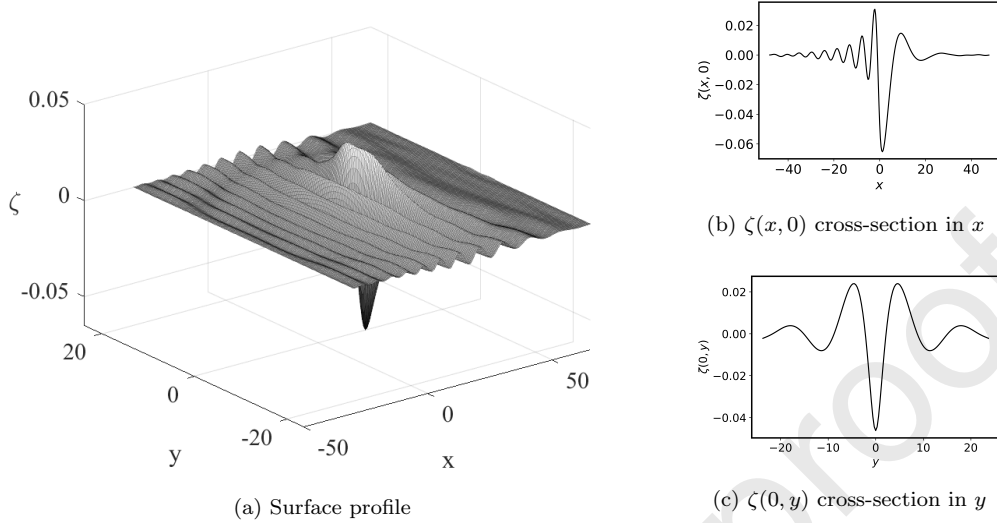


Figure 8: A Forced solution ($U > c_{\min}$) with artificial viscosity ($\mu = 0.1$) computed on a 320×80 mesh with spacing $\Delta x = \Delta y = 0.3$, and $x_1 = -48$. The dimensionless parameters are $F = 0.35$ and $\beta = 0.5$

The effects of the truncated domain are more evident in this regime since we have waves of different wavelengths that appear before and after the pressure so the radiation condition imposed at the upstream boundary may not be accurately enforced if x_1 is chosen too close to the pressure. Similarly, choosing the lateral boundary y_M too close to the pressure may lead to non-physical reflections at the boundary. Figure 9 compares centrelines of solutions in this regime, computed for the same parameter values but different lengths of the truncated domain, varying the number of N grid points for fixed uniform grid spacing $\Delta x = \Delta y = 0.3$. The grid spacing is chosen to be sufficiently small such that the effects from truncation may be considered the dominant source of error. In Figure 9, we can see some agreement around the support of the pressure but find more pronounced discrepancies further upstream and downstream as the domain is truncated closer to the source of the disturbance.

6. Discussion

In this work we developed a novel preconditioning method to efficiently compute forced flexural-gravity travelling wave solutions in three dimensions. Using the nonlinear governing equations (1)-(4), and following [5, 28], we

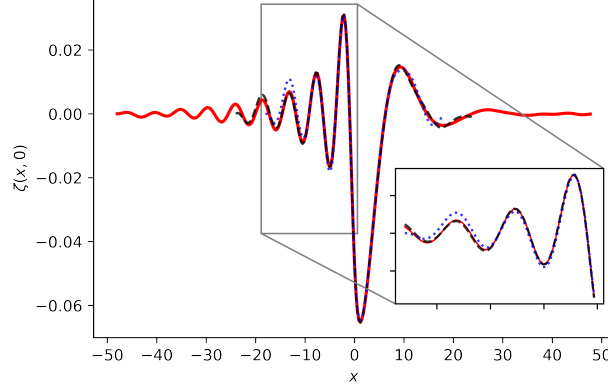


Figure 9: Comparison of centrelines $\zeta(x, 0)$ for forced wave solutions on different truncated domains $N \times M$: 320×80 with $x_1 = -48$ (solid line), 160×80 with $x_1 = -24$ (dashed line), 120×80 with $x_1 = -18$ (dotted line), with uniform grid spacing $\Delta x = \Delta y = 0.3$. For all solutions, the dimensionless parameters are $F = 0.35$, $\beta = 0.5$ and $\mu = 0.1$. The inset shows a close up of this comparison upstream of $x = 0$.

employed a boundary-integral method to obtain a desingularized integro-differential set of equations for the velocity potential $\phi(x, y)$ and variable surface $\zeta(x, y)$. The resulting discretized equations are solved using an iterative Newton-Krylov method that relies on a suitable preconditioner to converge. The hybrid preconditioning method we have developed for flexural-gravity waves is an extension of the framework devised for gravity waves generated by a point source [34]. The novelty of our contribution is the treatment of the flexural term in the Jacobian of the system, which is required to build an effective preconditioner. The hybrid preconditioner is so called as it combines analytical and numerical differentiation to approximate the Jacobian of the linear problem for flexural-gravity waves. This versatile method of generating the preconditioner has proven to be highly effective, and has allowed us to reduce the computational time dramatically from several hours to a few minutes for contemporary mesh sizes.

As mentioned in §2.1, the form of the solutions for flexural-gravity waves is known to depend on the speed of the moving pressure distribution U relative to the critical speed c_{\min} [5]. With the hybrid preconditioning method, we computed representative solutions in both $U < c_{\min}$ and $U > c_{\min}$ regimes that agree with previously published results. The performance gains achieved using the hybrid preconditioning method persist for finer meshes with more grid-points making it possible to analyze the error arising from under-refining

solutions and truncating at the edges of the computational domain. We showed that the error decays exponentially as the mesh is refined and when a larger computational domain is considered. In turn, this implies that with this new method, we are able to approach grid independence of our solutions. To further increase the grid refinement and decrease the run-time of our solutions, we wish to explore the possibility of utilizing graphics processing unit acceleration in future work, as was done in [34, 44].

The dense submatrix of the preconditioner, $\tilde{\mathbf{J}}_{2,2}$, is the most computationally intensive part and is considered to be the limiting factor for the number of grid points we can use. However, we take advantage of the fact that the construction of the dense submatrix only depends on the mesh parameters $(N, M, \Delta x, \Delta y, x_1)$. As such, it can be stored and reused to probe different parameter regimes (*e.g.*, varying $F, \beta, \epsilon, L, \mu$) on a given mesh without recomputing it each time. Moreover, the flexibility allowed by calculating the flexural contribution to the Jacobian numerically can be adapted easily for different models for ice, adding very little computational complexity to find solutions in a wider variety of physical regimes.

For our purposes, we have chosen the linear biharmonic model for the ice sheet and an axially symmetric pressure distribution as an idealized problem to illustrate our method, but our method can easily be extended to different models for ice and features of the disturbance. Without any modification to the preconditioner, it is also possible to compute elevation waves by the sign of P_0 in (8). We plan to consider different nonlinear models such as Kirchhoff–Love and Plotnikov–Toland in future work and expect that our hybrid preconditioning method will generalise in a straightforward manner.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2020-06417].

Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [numéro de référence RGPIN-2020-06417].

Appendix A. Biharmonic finite difference scheme

The flexural term, P_{flex} , is evaluated at the midpoints of the two-dimensional $N \times M$ mesh described in §3.1 and is defined as

$$(P_{\text{flex}}^*)_{k,\ell} \equiv \nabla^4 \zeta_{k,\ell}^* = (\nabla^4 \zeta_{k+1,\ell} + \nabla^4 \zeta_{k,\ell})/2, \quad (\text{A.1})$$

which involves fourth-order partial derivatives. Since applying finite difference equations for higher order derivatives can become rather involved, and different schemes are possible, we wish to discuss the biharmonic term in (A.1) in more detail here to remove any ambiguity. The biharmonic operator $\nabla^4 \zeta_{k,\ell}$ is discretized using centered finite differences in the x - and y -directions with 13-points [41]. Figure A.10 shows a visualization of the dependence of one point in the biharmonic term $(P_{\text{flex}}^*)_{k,\ell} = \nabla^4 \zeta_{k,\ell}^*$ on the nearby 17 points.

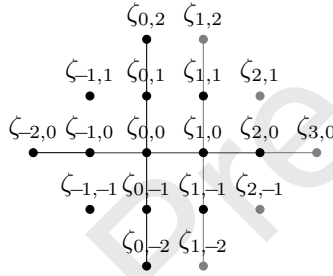


Figure A.10: 18-point stencil for $(P_{\text{flex}}^*)_{0,0}$ with symmetric grid spacing, $\Delta x = \Delta y$

To allow for asymmetric spacing in the x - and y -directions,

$$(P_{\text{flex}})_{k,\ell} = \nabla^4 \zeta_{k,\ell} = \left(\frac{\partial^4 \zeta}{\partial x^4} + 2 \frac{\partial^4 \zeta}{\partial x^2 \partial y^2} + \frac{\partial^4 \zeta}{\partial y^4} \right)_{k,\ell}$$

is computed in the interior of the domain using the following second-order accurate centered finite difference formulas

$$\begin{aligned} \left(\frac{\partial^4 \zeta}{\partial x^4} \right)_{k,\ell} &= (\zeta_{k-2,\ell} - 4\zeta_{k-1,\ell} + 6\zeta_{k,\ell} - 4\zeta_{k+1,\ell} + \zeta_{k+2,\ell})/(\Delta x)^4 + \mathcal{O}(\Delta x^2) \\ \left(\frac{\partial^4 \zeta}{\partial x^2 \partial y^2} \right)_{k,\ell} &= (\zeta_{k+1,\ell+1} + \zeta_{k-1,\ell+1} + \zeta_{k+1,\ell-1} + \zeta_{k-1,\ell-1} + 4\zeta_{k,\ell})/(\Delta x \Delta y)^2 \\ &\quad - 2(\zeta_{k+1,\ell} + \zeta_{k,\ell+1} + \zeta_{k-1,\ell} + \zeta_{k,\ell-1})/(\Delta x \Delta y)^2 + \mathcal{O}(\Delta x \Delta y) \\ \left(\frac{\partial^4 \zeta}{\partial y^4} \right)_{k,\ell} &= (\zeta_{k,\ell-2} - 4\zeta_{k,\ell-1} + 6\zeta_{k,\ell} - 4\zeta_{k,\ell+1} + \zeta_{k,\ell+2})/\Delta y^4 + \mathcal{O}(\Delta y^2), \end{aligned} \quad (\text{A.2})$$

which reduces to the 13-point formula when $\Delta x = \Delta y$ [41]. At the boundaries of the truncated domain, the finite difference formula (A.2) depends on points outside the domain, which require additional considerations. At the upstream $x = x_1$ boundary we use second-order accurate forward differences for differentiation in the x -direction. At the $y = 0$ boundary, points outside the domain are obtained by symmetry. At the downstream, $x = x_N$, and lateral, $y = y_M$, boundaries we apply centered differences and use extrapolation for points outside the domain. Developing boundary conditions for a finite computational domain without any spurious reflections can be a challenge [5, 45], and we leave this consideration for future work.

Appendix B. The Jacobian of the linear free-surface problem

Appendix B.1. Algebraic equations for the linearized problem

The analytical part of the preconditioner is given by the Jacobian of the linear problem with $\beta = 0$ in (22), following the approach in [34]. The linear problem presented in §2.4 is discretized in the same manner as the nonlinear problem, using the numerical scheme outlined in §3.1 and evaluated on the half-mesh points $(x_k^*, y_\ell^*) = ((x_k + x_{k+1})/2, y_\ell)$ for $k = 1, \dots, N-1$, $\ell = 1, \dots, M$. The singularity in (23) is dealt with in the same way as with the nonlinear problem by noting that

$$\iint_{\mathbb{R}^2} \zeta_x K_3 \, dx \, dy = \iint_{\mathbb{R}^2} K_3 (\zeta_x - \zeta_x^*) \, dx \, dy + \zeta_x^* \iint_{\mathbb{R}^2} K_3 \, dx \, dy,$$

and using symmetry in the y -direction to compute solutions on half of the domain which introduces an additional term in the kernel.

The resulting vector function, \mathbf{E} , for the linear flexural-gravity problem

with artificial viscosity is

$$\mathbf{E}_{1(k,\ell)} = \phi_{x(k,\ell)}^* - 1 + F\zeta_{(k,\ell)}^* + \mu\phi_{(k,\ell)}^* + \beta P_{\text{flex}(k,\ell)}^* + p(x_k^*, y_\ell^*) \quad (\text{B.1})$$

$$\begin{aligned} \mathbf{E}_{2(k,\ell)} = & 2\pi(\phi_{(k,\ell)}^* - x_k^*) - \zeta_{x(i,j)}^* \int_{y_1}^{y_M} \int_{x_1}^{x_N} K_3 \, dx \, dy \\ & - \sum_{i=1}^N \sum_{j=1}^M w(i,j) (\zeta_{x(i,j)} - \zeta_{x(k,\ell)}^*) K_{3(i,j,k,\ell)} \end{aligned} \quad (\text{B.2})$$

$$\mathbf{E}_{3(\ell)} = \phi_{(1,\ell)} - x_1 \quad (\text{B.3})$$

$$\mathbf{E}_{4(\ell)} = \phi_{x(1,\ell)} - 1 \quad (\text{B.4})$$

$$\mathbf{E}_{5(\ell)} = \zeta_{(1,\ell)} \quad (\text{B.5})$$

$$\mathbf{E}_{6(\ell)} = \zeta_{x(1,\ell)}, \quad (\text{B.6})$$

for $k = 1, \dots, N-1$, $\ell = 1, \dots, M$ and $K_{3(i,j,k,\ell)} = K_3(x_i, y_j; x_k^*, y_\ell^*)$, which can be integrated exactly in terms of logarithms

$$\iint \frac{ds \, dt}{\sqrt{s^2 + t^2}} = t \ln(2s + 2\sqrt{s^2 + t^2}) + s \ln(2t + 2\sqrt{s^2 + t^2}).$$

The weighting function $w(i,j)$ for numerical integration is defined for the trapezoidal-rule such that $w(i,j) = \Delta x \Delta y$ on the interior of the domain $i = 2, \dots, N-1$, $j = 2, \dots, M-1$, each $i = 1, N$ and $j = 1, M$ brings in an additional factor of $1/2$, such that $w(i,j) = (\Delta x \Delta y)/2$ for values along the edges, and $w(i,j) = (\Delta x \Delta y)/4$ in the corners.

Appendix B.2. Ordering the equations

The left-hand side of (25) is a vector valued function made up of the six equations (B.1)-(B.6) evaluated at the half-mesh points $(x_{k+1/2}, y_\ell)$, for $k = 1, \dots, N-1$ and $\ell = 1, \dots, M$. These equations are ordered according to the framework developed by Pethiyagoda *et al.* [34], and are as follows

$$\mathbf{E} = [\mathbf{E}_{3(1)}, \mathbf{E}_{4(1)}, \mathbf{E}_{1(1,1)}, \dots, \mathbf{E}_{1(N-1,1)}, \dots, \mathbf{E}_{3(M)}, \mathbf{E}_{4(M)}, \mathbf{E}_{1(1,M)}, \dots, \mathbf{E}_{1(N-1,M)}, \mathbf{E}_{5(1)}, \mathbf{E}_{6(1)}, \mathbf{E}_{2(1,1)}, \dots, \mathbf{E}_{2(N-1,1)}, \dots, \mathbf{E}_{5(M)}, \mathbf{E}_{6(M)}, \mathbf{E}_{2(1,M)}, \dots, \mathbf{E}_{2(N-1,M)}]^T, \quad (\text{B.7})$$

resulting in the Jacobian structure illustrated in Figure 2. The purpose of this ordering is to make the structure of the Jacobian more sparse and easier to invert.

Appendix B.3. Explicit differentiation

Finally, to calculate the Jacobian of the linear problem with $\beta = 0$ analytically, the system (B.1)-(B.6) can be easily differentiated by hand with respect to the unknowns $\phi_{(1,m)}$, $\phi_{x(n,m)}$, $\zeta_{(1,m)}$, and $\zeta_{x(n,m)}$, for $n = 1 \dots N$, $m = 1 \dots M$. Note, the contribution from P_{flex} is discussed in more detail in Appendix A.

Equations (B.3)-(B.6) are already in terms of the unknowns (28), so we can immediately determine the expressions for their explicit differentiation:

$$\begin{aligned} \frac{\partial \mathbf{E}_{3(\ell)}}{\partial \phi_{(1,m)}} &= \frac{\partial \mathbf{E}_{5(\ell)}}{\partial \zeta_{(1,m)}} = \begin{cases} 1 & \text{for } m = \ell \\ 0 & \text{otherwise} \end{cases} \\ \frac{\partial \mathbf{E}_{4(\ell)}}{\partial \phi_{x(n,m)}} &= \frac{\partial \mathbf{E}_{6(\ell)}}{\partial \zeta_{x(n,m)}} = \begin{cases} 1 & \text{for } n = 1, m = \ell \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.8})$$

The other partial derivatives of $\mathbf{E}_{3(\ell)}$, $\mathbf{E}_{4(\ell)}$, $\mathbf{E}_{5(\ell)}$ and $\mathbf{E}_{6(\ell)}$ with respect to the unknowns $\phi_{(1,m)}$, $\phi_{x(n,m)}$, $\zeta_{(1,m)}$, and $\zeta_{x(n,m)}$ are all equal to zero. Since the only nonzero partial derivatives of $\mathbf{E}_{3(\ell)}$ and $\mathbf{E}_{4(\ell)}$ are with respect to $\phi_{(1,m)}$ and $\phi_{x(n,m)}$, the contribution from \mathbf{E}_3 and \mathbf{E}_4 only appears in the top-left submatrix of \mathbf{P} . Similarly, the only nonzero partial derivatives of $\mathbf{E}_{5(\ell)}$ and $\mathbf{E}_{6(\ell)}$ are with respect to $\zeta_{(1,m)}$ and $\zeta_{x(n,m)}$. Hence, the contribution from \mathbf{E}_5 and \mathbf{E}_6 only appears in the bottom-right (dense) submatrix of \mathbf{P} . Notice that the expressions provide identical contributions to the corresponding entries in each block.

Next, we need to express (B.1)-(B.2) in terms of the unknowns (28). This involves determining how ϕ , ϕ^* , ζ and ζ^* depend on the unknowns in (28). To do this, we need the values for ζ obtained by trapezoidal-rule integration using the values of ζ_x :

$$\zeta_{(k+1,\ell)} = \zeta_{(k,\ell)} + \frac{\Delta x}{2} (\zeta_{x(k,\ell)} + \zeta_{x(k+1,\ell)}), \quad \ell = 1, \dots, M, \quad k = 1, \dots, N-1, \quad (\text{B.9})$$

and apply a similar formula to compute the values of ϕ using ϕ_x . Expanding the trapezoidal-rule integration of ζ and ϕ gives

$$\begin{aligned} \zeta_{(k,\ell)} &= \zeta_{(1,\ell)} + \frac{\Delta x}{2} \zeta_{x(1,\ell)} + \Delta x \sum_{i=2}^{k-1} \zeta_{x(i,\ell)} + \frac{\Delta x}{2} \zeta_{x(k,\ell)}, \\ \phi_{(k,\ell)} &= \phi_{(1,\ell)} + \frac{\Delta x}{2} \phi_{x(1,\ell)} + \Delta x \sum_{i=2}^{k-1} \phi_{x(i,\ell)} + \frac{\Delta x}{2} \phi_{x(k,\ell)}, \end{aligned} \quad (\text{B.10})$$

for $k = 2, \dots, N$, $\ell = 1, \dots, M$. The values for ζ^* and ϕ^* at the half-mesh points are obtained using two-point interpolation

$$\zeta_{(k,\ell)}^* = \frac{1}{2}(\zeta_{(k,\ell)} + \zeta_{(k+1,\ell)}), \quad \phi_{(k,\ell)}^* = \frac{1}{2}(\phi_{(k,\ell)} + \phi_{(k+1,\ell)}). \quad (\text{B.11})$$

From the result of (B.10) and the two-point interpolation in (B.11), we obtain the expressions for ζ^* and ϕ^* in terms of the variables in (28):

$$\zeta_{(k,\ell)}^* = \zeta_{(1,\ell)} + \frac{\Delta x}{2}\zeta_{x(1,\ell)} + \Delta x \sum_{i=2}^{k-1} \zeta_{x(i,\ell)} + \frac{3\Delta x}{4}\zeta_{x(k,\ell)} + \frac{\Delta x}{4}\zeta_{x(k+1,\ell)} \quad (\text{B.12a})$$

$$\phi_{(k,\ell)}^* = \phi_{(1,\ell)} + \frac{\Delta x}{2}\phi_{x(1,\ell)} + \Delta x \sum_{i=2}^{k-1} \phi_{x(i,\ell)} + \frac{3\Delta x}{4}\phi_{x(k,\ell)} + \frac{\Delta x}{4}\phi_{x(k+1,\ell)}, \quad (\text{B.12b})$$

for $k = 1, \dots, (N-1)$, $\ell = 1, \dots, M$. We use the equivalent of (B.11) for ζ_x^* and ϕ_x^* to get

$$\zeta_{x(k,\ell)}^* = \frac{1}{2}(\zeta_{x(k,\ell)} + \zeta_{x(k+1,\ell)}), \quad \phi_{x(k,\ell)}^* = \frac{1}{2}(\phi_{x(k,\ell)} + \phi_{x(k+1,\ell)}). \quad (\text{B.13})$$

We substitute the expressions (B.12) for $\zeta_{k,\ell}^*$, $\phi_{k,\ell}^*$, and the two-point interpolation (B.13) for their derivatives $\phi_{x(k,\ell)}^*$, $\zeta_{x(k,\ell)}^*$ into (B.1) and (B.2) to obtain the expressions for $\mathbf{E}_{1(k,\ell)}$ and $\mathbf{E}_{2(k,\ell)}$ in terms of the unknowns in (28).

To find the expressions for the differentiation of (B.1)-(B.2), it is helpful to note that

$$\begin{aligned} \frac{\partial \zeta_{(k,\ell)}^*}{\partial \zeta_{(1,m)}} &= \frac{\partial \phi_{(k,\ell)}^*}{\partial \phi_{(1,m)}} = \begin{cases} 1 & \text{for } m = \ell \\ 0 & \text{otherwise} \end{cases} \\ \frac{\partial \zeta_{(k,\ell)}^*}{\partial \zeta_{x(n,m)}} &= \frac{\partial \phi_{(k,\ell)}^*}{\partial \phi_{x(n,m)}} = \begin{cases} \frac{\Delta x}{4} & \text{for } n = 1, k = 1 \text{ and } m = \ell \\ \frac{\Delta x}{2} & \text{for } n = 1, k > 1 \text{ and } m = \ell \\ \Delta x & \text{for } n < k, \text{ and } m = \ell \\ \frac{3\Delta x}{4} & \text{for } n = k, \text{ and } m = \ell \\ \frac{\Delta x}{4} & \text{for } n = k + 1, \text{ and } m = \ell \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.14})$$

These expressions form repeating blocks on the diagonal when $m = \ell = 1, \dots, M$.

The expressions for the explicit differentiation of (B.1)-(B.2) with $\beta = 0$ are as follows

$$\begin{aligned} \frac{\partial \mathbf{E}_{1(k,\ell)}}{\partial \phi_{(1,m)}} &= \begin{cases} \mu & \text{for } m = \ell \\ 0 & \text{otherwise} \end{cases} \\ \frac{\partial \mathbf{E}_{1(k,\ell)}}{\partial \phi_{x(n,m)}} &= \begin{cases} \frac{\mu \Delta x}{4} & \text{for } n = 1, k = 1 \text{ and } m = \ell \\ \frac{\mu \Delta x}{2} & \text{for } n = 1, k > 1 \text{ and } m = \ell \\ \mu \Delta x & \text{for } n < k, \text{ and } m = \ell \\ \frac{3\mu \Delta x}{4} + \frac{1}{2} & \text{for } n = k, \text{ and } m = \ell \\ \frac{\mu \Delta x}{4} + \frac{1}{2} & \text{for } n = k + 1, \text{ and } m = \ell \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.15})$$

$$\begin{aligned} \frac{\partial \mathbf{E}_{1(k,\ell)}}{\partial \zeta_{(1,m)}} &= \begin{cases} F & \text{for } m = \ell \\ 0 & \text{otherwise} \end{cases} \\ \frac{\partial \mathbf{E}_{1(k,\ell)}}{\partial \zeta_{x(n,m)}} &= \begin{cases} \frac{F \Delta x}{4} & \text{for } n = 1, k = 1 \text{ and } m = \ell \\ \frac{F \Delta x}{2} & \text{for } n = 1, k > 1 \text{ and } m = \ell \\ F \Delta x & \text{for } n < k, \text{ and } m = \ell \\ \frac{3F \Delta x}{4} & \text{for } n = k, \text{ and } m = \ell \\ \frac{F \Delta x}{4} & \text{for } n = k + 1, \text{ and } m = \ell \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.16})$$

$$\begin{aligned} \frac{\partial \mathbf{E}_{2(k,\ell)}}{\partial \phi_{(1,m)}} &= \begin{cases} 2\pi & \text{for } m = \ell \\ 0 & \text{otherwise} \end{cases} \\ \frac{\partial \mathbf{E}_{2(k,\ell)}}{\partial \phi_{x(n,m)}} &= \begin{cases} \frac{\pi \Delta x}{2} & \text{for } n = 1, k = 1 \text{ and } m = \ell \\ \pi \Delta x & \text{for } n = 1, k > 1 \text{ and } m = \ell \\ 2\pi \Delta x & \text{for } n < k, \text{ and } m = \ell \\ \frac{3\pi \Delta x}{2} & \text{for } n = k, \text{ and } m = \ell \\ \frac{\pi \Delta x}{2} & \text{for } n = k + 1, \text{ and } m = \ell \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.17})$$

$$\begin{aligned}
\frac{\partial \mathbf{E}_{2(k,\ell)}}{\partial \zeta_{(1,m)}} &= 0 \\
\frac{\partial \mathbf{E}_{2(k,\ell)}}{\partial \zeta_{x(n,m)}} &= \begin{cases} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M w(i,j) K_{3(i,j,k,\ell)} - \frac{1}{2} I - w(n,m) K_{3(n,m,k,\ell)}, & \text{for } n = k, k+1 \text{ and } m = \ell \\ -w(n,m) K_{3(n,m,k,\ell)}, & \text{otherwise} \end{cases}
\end{aligned} \tag{B.18}$$

The preconditioner is formed by ordering these Jacobian entries in the manner described in Appendix B.2.

References

- [1] A. Korobkin, E. I. Părău, J.-M. Vanden-Broeck, The mathematical challenges and modelling of hydroelasticity, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 369 (2011) 2803–2812.
- [2] O. Trichtchenko, E. I. Părău, J.-M. Vanden-Broeck, P. Milewski, Solitary flexural-gravity waves in three dimensions, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376 (2129) (2018) 20170345.
- [3] H. Babaei, J. Van Der Sanden, N. Short, P. Barrette, Lake ice cover deflection induced by moving vehicles: comparing theoretical results with satellite observations, *Proceedings of TAC* (2016).
- [4] K. Kobayashi, S. Okada, C. Sato, Dynamic responses of the mega-float phase-ii 1000 m model., in: *ISOPE-2002: Twelfth(2002) International Offshore and Polar Engineering Conference, 2002*, pp. 245–252.
- [5] E. I. Părău, J.-M. Vanden-Broeck, Three-dimensional waves beneath an ice sheet due to a steadily moving pressure, *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 369 (1947) (2011) 2973–2988.
- [6] V. A. Squire, R. J. Hosking, A. D. Kerr, P. J. Langhorne, *Moving loads on ice plates*, The Netherlands: Kluwer, Dordrecht, 1996.

- [7] R. J. Hosking, A. D. Sneyd, D. W. Waugh, Viscoelastic response of a floating ice plate to a steadily moving load, *Journal of Fluid Mechanics* 196 (1988) 409–430.
- [8] J. Strathdee, W. H. Robinson, E. M. Haines, Moving loads on ice plates of finite thickness, *Journal of Fluid Mechanics* 226 (1991) 37–61.
- [9] A. G. Greenhill, Wave motion in hydrodynamics, *American Journal of Mathematics* (1886) 62–96.
- [10] J. W. Davys, R. J. Hosking, A. D. Sneyd, Waves due to a steadily moving source on a floating ice plate, *Journal of Fluid Mechanics* 158 (1985) 269–287.
- [11] F. Milinazzo, M. Shinbrot, N. W. Evans, A mathematical analysis of the steady response of floating ice to the uniform motion of a rectangular load, *Journal of Fluid Mechanics* 287 (1995) 173–197.
- [12] V. A. Squire, Of ocean waves and sea-ice revisited, *Cold Regions Science and Technology* 49 (2) (2007) 110–133.
- [13] P. I. Plotnikov, J. F. Toland, Modelling nonlinear hydroelastic waves, *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 369 (1947) (2011) 2942–2956.
- [14] P. A. Milewski, Z. Wang, Three dimensional flexural-gravity waves, *Studies in Applied Mathematics* 131 (2) (2013) 135–148.
- [15] J. F. Toland, Steady periodic hydroelastic waves, *Archive for rational mechanics and analysis* 189 (2) (2008) 325–362.
- [16] M. D. Groves, B. Hewer, E. Wahlén, Variational existence theory for hydroelastic solitary waves, *Comptes Rendus Mathématique* 354 (11) (2016) 1078–1086.
- [17] B. F. Akers, D. M. Ambrose, D. W. Sulton, Periodic traveling interfacial hydroelastic waves with or without mass, *Zeitschrift für angewandte Mathematik und Physik* 68 (6) (2017) 1–27.
- [18] A. V. Marchenko, V. I. Shrira, Theory of two-dimensional nonlinear waves in liquid covered by ice, *Fluid dynamics* 26 (4) (1991) 580–587.

- [19] E. I. Părău, F. Dias, Nonlinear effects in the response of a floating ice plate to a moving load, *Journal of Fluid Mechanics* 460 (2002) 281–305.
- [20] P. Guyenne, E. I. Părău, Computations of fully nonlinear hydroelastic solitary waves on deep water, *Journal of Fluid Mechanics* 713 (2012) 307–329.
- [21] P. Guyenne, E. I. Părău, Finite-depth effects on solitary waves in a floating ice sheet, *Journal of Fluids and Structures* 49 (2014) 242–262.
- [22] X. Xia, H. T. Shen, Nonlinear interaction of ice cover with shallow water waves in channels, *Journal of Fluid Mechanics* 467 (2002) 259.
- [23] M. Hărăguș-Courcelle, A. Il'ichev, Three-dimensional solitary waves in the presence of additional surface effects, *European Journal of Mechanics-B/Fluids* 17 (5) (1998) 739–768.
- [24] P. Guyenne, E. I. Părău, Asymptotic modeling and numerical simulation of solitary waves in a floating ice sheet, in: *The Twenty-Fifth International Ocean and Polar Engineering Conference, OnePetro*, 2015, pp. 467–475.
- [25] Z. Wang, P. A. Milewski, J.-M. Vanden-Broeck, Computation of three-dimensional flexural-gravity solitary waves in arbitrary depth, *Procedia IUTAM* 11 (2014) 119–129.
- [26] E. Dinvey, H. Kalisch, E. I. Părău, Fully dispersive models for moving loads on ice sheets, *Journal of Fluid Mechanics* 876 (2019) 122–149.
- [27] L. Landweber, M. Macagno, Irrotational flow about ship forms, *Tech. Rep. 123, IIHR* (1969).
- [28] L. K. Forbes, An algorithm for 3-dimensional free-surface problems in hydrodynamics, *Journal of Computational Physics* 82 (2) (1989) 330 – 347.
- [29] E. I. Părău, J.-M. Vanden-Broeck, Nonlinear two- and three-dimensional free surface flows due to moving disturbances, *European Journal of Mechanics - B/Fluids* 21 (6) (2002) 643–656.

- [30] E. I. Părău, J.-M. Vanden-Broeck, M. J. Cooker, Three-dimensional capillary-gravity waves generated by a moving disturbance, *Physics of fluids* (1994) 19 (8) (2007) 82–102.
- [31] E. I. Părău, J.-M. Vanden-Broeck, M. J. Cooker, Nonlinear three-dimensional gravity-capillary solitary waves, *Journal of Fluid Mechanics* 536 (2005) 99–105.
- [32] E. I. Părău, J.-M. Vanden-Broeck, M. J. Cooker, Three-dimensional gravity-capillary solitary waves in water of finite depth and related problems, *Physics of Fluids* 17 (12) (2005) 122101.
- [33] D. A. Knoll, D. E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, *Journal of Computational Physics* 193 (2) (2004) 357–397.
- [34] R. Pethiyagoda, S. W. McCue, T. J. Moroney, J. M. Back, Jacobian-free Newton–Krylov methods with GPU acceleration for computing nonlinear ship wave patterns, *Journal of Computational Physics* 269 (2014) 297–313.
- [35] N. R. Buttle, R. Pethiyagoda, T. J. Moroney, S. W. McCue, Three-dimensional free-surface flow over arbitrary bottom topography, *Journal of Fluid Mechanics* 846 (2018) 166–189.
- [36] J. M. S. Rayleigh, The form of standing waves on the surface of running water, *Proceedings of the London Mathematical Society* 1 (1) (1883) 69–78.
- [37] H. Lamb, *Hydrodynamics*, 6th Edition, Cambridge University Press, 1932, Ch. 7, pp. 224–228, art. 156.
- [38] F. Noblesse, Alternative integral representations for the green function of the theory of ship wave resistance, *Journal of Engineering Mathematics* 15 (4) (1981) 241–265.
- [39] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics, SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, 1995.

- [40] Y. Saad, M. H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing* 7 (3) (1986) 856–869.
- [41] M. Abramowitz, I. A. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, Applied mathematics series, Dover Publications, 1965.
- [42] D. C. Scullen, *Accurate computation of steady nonlinear free-surface flows*, Phd thesis, The University of Adelaide (1998).
- [43] A. H. Baker, E. R. Jessup, T. Manteuffel, A technique for accelerating the convergence of restarted GMRES, *SIAM Journal on Matrix Analysis and Applications* 26 (4) (2005) 962–23.
- [44] R. Pethiyagoda, T. J. Moroney, S. W. McCue, Efficient computation of two-dimensional steady free-surface flows, *International Journal for Numerical Methods in Fluids* 86 (9) (2018) 607–624.
- [45] D. Givoli, Non-reflecting boundary conditions, *Journal of computational physics* 94 (1) (1991) 1–29.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

--