

Preconditioning and linearly implicit time integration for the Serre-Green-Naghdi equations

Linwan Feng, David Shirokoff ^{*}, Wooyoung Choi 

Department of Mathematical Sciences, New Jersey Institute of Technology, University Heights, Newark, NJ, 07102, USA

ARTICLE INFO

2000 MSC:

65L04
65L06
65L07
65M12
76B15
76M22

Keywords:

Preconditioning
Linearly implicit
Serre equations
Green-Naghdi
Dispersive shallow water
High order time integration

ABSTRACT

The treatment of the differential PDE constraint poses a key challenge in computing the numerical solution of the Serre-Green-Naghdi (SGN) equations. In this work, we introduce a constant coefficient preconditioner for the SGN constraint operator and prove rigorous bounds on the preconditioned conditioning number. The conditioning bounds incorporate the effects of bathymetry in two dimensions, are quasi-optimal within a class of constant coefficient operators, highlight fundamental scalings for a loss of conditioning, and ensure mesh independent performance for iterative Krylov methods.

Utilizing the conditioning bounds, we devise and test two time integration strategies for solving the full SGN equations. The first class combines classical explicit time integration schemes (4th order Runge-Kutta and 2nd–4th order Adams-Bashforth) with the new preconditioner. The second is a linearly implicit scheme where the differential constraint is split into a constant coefficient implicit part and remaining (stiff) explicit part. The linearly implicit methods require a single linear solve of a constant coefficient operator at each time step. We provide a host of computational experiments that validate the robustness of the preconditioners, as well as full solutions of the SGN equations including solitary waves traveling over an underwater shelf (in 1d) and a circular bump (in 2d).

1. Introduction

The shallow water equations (SWEs) provide an effective model for simulating coastal waves and large-scale fluid circulation in ocean basins. This includes important physical phenomena such as tides, storm surges, and the effect of extreme weather, such as hurricanes, on wave generation.

From a modeling standpoint, the SWEs arise from an asymptotic expansion of the (depth-averaged) Euler equations truncated to the leading $O(1)$ term in $\beta \equiv \bar{h}/\lambda \ll 1$. Here β is a small parameter measuring the ratio of the mean water depth $\bar{h}$ to the characteristic wavelength λ . Carrying out the asymptotic expansion (in β) to the next order results in dispersive corrections to the SWEs. Physically, the incorporation of dispersion to the SWEs plays an important role in the propagation of long waves at larger (but still small) β . In particular, solitary waves may form through the balancing of dispersion and (steepening effects of) nonlinearity, while dispersion may also regularize the formation of shocks.

The simplest models that incorporate dispersion arise under a weakly nonlinear assumption of small amplitude waves, i.e., $\alpha \equiv a/\bar{h} \ll 1$ with a the amplitude of the wave. These include models for long waves such as the Korteweg de-Vries (KdV) equation [1] and the Boussinesq equations [2] for uni- and bi-directional waves. Several important phenomena such as extreme events observed

^{*} Corresponding author.

E-mail addresses: lf46@njit.edu (L. Feng), david.g.shirokoff@njit.edu (D. Shirokoff), wchoi@njit.edu (W. Choi).

in coastal regions, however, do not satisfy the weakly non-linear assumption, thus motivating the need for extending the models to include high-order nonlinear effects. To this end, Serre [3], Su and Gardner [4] and Green and Naghdi [5] derived a new class of long wave models with the leading-order dispersive terms that appear at $O(\beta^2)$ without any assumption on α . Asymptotically, the Serre/SG/GN equations (often referred to as the SGN equations) modify the shallow water equations by incorporating dispersive terms truncated at an order of β^2 . Therefore, the resulting SGN equations can also be considered as the *dispersive shallow water equations* (DSWEs).

There are numerous asymptotic models closely related to the SGN equations. For instance, the SGN equations for the depth-averaged velocity can be formally extended to any arbitrary order in β^2 , e.g., see the work by Wu and others [6–10]. However, it has been shown that the models truncated to even-orders in β^2 (i.e., the even-order models) are unstable when the wavenumber k is greater than some critical values, at which the growth rate becomes infinity [10,11]. Therefore, the even-order models must be avoided while the odd-order models (including the SGN equations) are useful for numerical studies. In fact, the dispersion relation for the depth-averaged DSWEs shows that the wave frequency is bounded as $k \rightarrow \infty$ so that there is a less severe constraint on the step size for any conditionally-stable time integration scheme.

Related to the SGN equations, Nwogu [12] introduced an alternative weakly nonlinear long wave model aimed at more closely approximating the linearized Euler equations dispersion relation (cf. the model solved in [13]). Following Nwogu, Wei et al. [14] proposed the first-order DSWEs written in terms of the horizontal velocity at a vertical level (as opposed to a depth-averaged velocity). In particular, it was shown that the higher order DSWEs for the horizontal velocity at the bottom are also well-posed at any order of approximation [10].

There are, however, numerical advantages to solving the depth averaged SGN equations over the DSWE equations in terms of the horizontal bottom velocity. For the depth-averaged SGN equations, the mass conservation equation contains a first order spatial derivative (see the detailed discussion in Section 2). On the other-hand, for other DSWEs, high-order spatial derivatives need to be evaluated accurately to avoid numerical instability [10].

For work on the existence of solutions to the three-dimensional water wave equations we refer to [15]; a general summary of the derivation for different equivalent SGN models can be found in [10,16]. The physics of water waves and their derivation of their models have been extensively studied, e.g., [2,10,17–19].

1.1. Prior numerical work on the SGN equations

The dispersive terms in the SGN equation contain mixed space-time derivatives resulting in a differential constraint. This differential constraint poses a key computational challenge that can dominate the computational cost of an SGN solver. In one dimension, the constraint can be handled in a straightforward fashion without significant cost, either via classical iterative methods, direct solvers, or integral equation approaches, for example, in [17,18,20]. In two dimensions the differential constraint results in a couple system for the velocity components with off diagonal blocks introduced by a varying topography.

One approach to time-discretize the SGN equations is to split the equation into a hyperbolic step (which can be treated with hyperbolic solvers) and a step for the elliptic (differential) constraint. These approaches include Strang splitting [13,21,22], and predictor-corrector type approaches [23–26]. A correct treatment of the boundary conditions [26] is then important when splitting methods are used in non-periodic geometries. Most of these aforementioned works focus on practical details of spatial discretizations (e.g., with finite element methods) and structure preservation (e.g., well-balanced and positive preserving schemes).

Approaches that address the SGN elliptic constraint include the development of quadtree multigrid methods [27]. An alternative approach, Lannes and Marche [21], Duran and Marche [28] instead modifies the SGN equation—to an asymptotically consistent one in β —by replacing the elliptic constraint with a block diagonal constant coefficient operator. The constraint can then be solved via two time-independent decoupled scalar elliptic systems. In [29] and [30], the authors carry out a numerical study of the stability of SGN solitary waves in dimensions one and two respectively. To solve the SGN constraint equation, both works use GMRES with a preconditioner based on the derivatives of the constraint operator.

In contrast to splitting methods or predictor-corrector methods, an alternative approach is to introduce a hyperbolic relaxation of the SGN [31] where the differential constraint is replaced by a hyperbolic system. The relaxation approach can then be solved using explicit structure preserving methods [32] that leverage the extensive body of existing hyperbolic PDE solution techniques. The original work of Favrie and Gavrilyuk has since been extended to incorporate the effects of topography [33,34] and unified in a general framework to include the Iordanskii-Kogarko-Wijngaarden (IKW) equations [35]. Numerical solutions for these equations include high order Galerkin spatial discretizations [33] and up to second order in space and time structure preserving methods (e.g., well-balanced, positivity preserving) properties [34]. Practical details regarding convex limiters [36] and schemes that preserve mass and energy [37] have also recently been devised. Structure preservation can improve accuracy over long times, and provide a pathway for ensuring numerical stability.

1.2. Contribution of this work and organization of paper

The differential constraint in the SGN requires solving linear systems involving an operator $\mathcal{G}_{\eta,h}$ with variable coefficients in both space and time. The focus of this paper is on introducing formulas for constant coefficients (σ, α) that define an operator $\mathcal{A} = \sigma I - \alpha \nabla(\nabla \cdot)$, together with quasi-optimal bounds on the spectrum of $\mathcal{A}^{-1} \mathcal{G}_{\eta,h}$. The bounds then enable time-stepping discretizations for the SGN that avoid fully constructing and inverting $\mathcal{G}_{\eta,h}$, namely:

1. For time integration methods that require solving systems involving $\mathcal{G}_{\eta,h}$, the operator $\mathcal{A}$ may be used as a preconditioner in a conjugate gradient method. Linear systems involving $\mathcal{G}_{\eta,h}$ arise in (standard explicit) time-stepping discretizations of the SGN equations, and must be solved in large numbers, making fast solvers desirable.
2. The operator $\mathcal{G}_{\eta,h}$ may be treated in a linearly implicit multi-step method where only $\mathcal{A}$ is treated implicitly. Such linearly implicit multistep methods are efficient as they require only one solves of $\mathcal{A}$ per time-step.

This paper is organized as follows. We first introduced the SGN equations (Section 2) and background for the paper (Section 3). We then present (Section 4) new formulas for the constant coefficient operator $\mathcal{A}$ along with rigorous bounds on the conditioning number and generalized eigenvalues. Utilizing the bounds, Section 5 devises zero-stable linearly implicit multistep time integration schemes. Numerical tests are then performed in (Section 6) showcasing the effectiveness of the preconditioner $\mathcal{A}$ in a suite of progressively challenging test cases. The full SGN equations are then solved in dimensions one and two, comparing the performance and accuracy of different time integration approaches. The Appendices provide proofs of the main theoretical results.

2. The Serre-Green-Naghdi equations

In Section 2.1 we review the derivation of the SGN equations and introduce in Section 2.2 an alternative *constraint* form of the equations (equivalent for smooth solutions). Spatial discretizations of the constraint form will naturally lead to a differential algebraic equations (DAE) which will play a role in the development and stability analysis of the numerical schemes.

2.1. SGN equations as asymptotic models of the Euler equations

As with the shallow water equations (SWE), the SGN equations arise as asymptotic expansions of the depth averaged incompressible Euler equations. In a domain $(\vec{x}, z) \in \mathbb{R}^{D+1}$, where $D = 1$ or $D = 2$, the incompressible Euler equations for a homogeneous fluid layer of constant density ρ take the form:

$$\vec{w}_t + \vec{w} \cdot \nabla_3 \vec{w} = -\frac{1}{\rho} \nabla_3 p - \vec{g}, \quad (2.1a)$$

$$\nabla_3 \cdot \vec{w} = 0. \quad (2.1b)$$

For notational purposes, we separate out the horizontal spatial coordinates $\vec{x} \in \Omega \subseteq \mathbb{R}^D$ from the vertical coordinate $z \in \mathbb{R}$. In (2.1), $\vec{w} = (\vec{u}, w) \in \mathbb{R}^{D+1}$ is the fluid velocity with $\vec{u} \in \mathbb{R}^D$ being the horizontal component, w the vertical component; $p(\vec{x}, z, t)$ is the pressure; $\vec{g}$ is the gravitational force. Meanwhile $\nabla_3 = (\nabla, \partial_z)$ is the (full) gradient in $\mathbb{R}^{D+1}$, where ∇ is the gradient in the variables $\vec{x}$.

As shown in Su and Gardner [4], the SGN (or DSWEs) arise from: (I) integrating the Euler equations over the vertical column of fluid (i.e., z coordinate); and (II) imposing boundary conditions at the free surface $\zeta(\vec{x}, t)$ and the bottom bathymetry $h(\vec{x})$. The derivation results in evolution equations for the new dependent variables which are the surface elevation $\zeta(\vec{x}, t)$, and the depth-averaged horizontal velocity $\bar{\vec{u}}(\vec{x}, t)$ defined by

$$\bar{\vec{u}}(\vec{x}, t) = \frac{1}{\zeta + h} \int_{-h}^{\zeta} \vec{u}(\vec{x}, z, t) \, dz. \quad (2.2)$$

It is worth noting that the process of depth-averaging the Euler equations results in expressions of the form $\overline{\vec{u} \otimes \vec{u}}$ arising from the (depth-averaged) nonlinear terms. To obtain a closed system of PDEs for $\zeta, \bar{\vec{u}}$, the nonlinear term $\vec{u} \otimes \vec{u}$ is approximated in terms of ζ and $\bar{\vec{u}}$ under the following assumptions

$$\beta \equiv \frac{\bar{h}}{\lambda} \ll 1, \quad \zeta/\bar{h} = O(a/\bar{h}) = O(1), \quad |\bar{\vec{u}}|/(g\bar{h})^{1/2} = O(1), \quad \nabla h = O(\bar{h}/\lambda) = O(\beta).$$

Here $\bar{h}$ is the average water depth, while a and λ denote the characteristic amplitude and wavelength of the wave. Typical values for a and λ may be estimated in the initial data $\vec{w}_0$ (see Fig. 1). The assumption $\beta \ll 1$ is referred to as the shallow water, or long wave assumption. Note that the assumptions allow for a “large amplitude” a to be comparable in magnitude to the mean water depth $\bar{h}$. Likewise, the associate fluid motion $|\bar{\vec{u}}|$ relative to $g\bar{h}$ does not need to be asymptotically small in β and may be $O(1)$.

With the asymptotic assumptions, one can then obtain a closed system for ζ and $\bar{\vec{u}}$ correct to $O(\beta^2)$. This yields the (dimensional form) of the Serre-Green-Naghdi (SGN) equations

$$\eta(\vec{x}, t) = h(\vec{x}) + \zeta(\vec{x}, t), \quad (2.3a)$$

$$\eta_t + \nabla \cdot (\eta \bar{\vec{u}}) = 0, \quad (2.3b)$$

$$\bar{\vec{u}}_t + \bar{\vec{u}} \cdot \nabla \bar{\vec{u}} + g \nabla \zeta = \frac{1}{\eta} \nabla \left(\frac{1}{3} \eta^3 G + \frac{1}{2} \eta^2 F \right) - \left(\frac{1}{2} \eta G + F \right) \nabla h, \quad (2.3c)$$

where G and F are defined as

$$G = \nabla \cdot \bar{\vec{u}}_t + \bar{\vec{u}} \cdot \nabla \left(\nabla \cdot \bar{\vec{u}} \right) - \left(\nabla \cdot \bar{\vec{u}} \right)^2, \quad F = \bar{\vec{u}}_t \cdot \nabla h + \left(\bar{\vec{u}} \cdot \nabla \right)^2 h. \quad (2.4)$$

Note that when the right-hand side of the SGN velocity equation is neglected, one recovers the (non-dispersive) shallow water equations. We therefore, also refer to (2.3) as the *dispersive shallow water equations*. If an additional small-amplitude assumption of $\alpha = O(a/\bar{h}) \ll 1$ is adopted, the system (2.3) can be reduced to the weakly nonlinear Boussinesq equations [2].

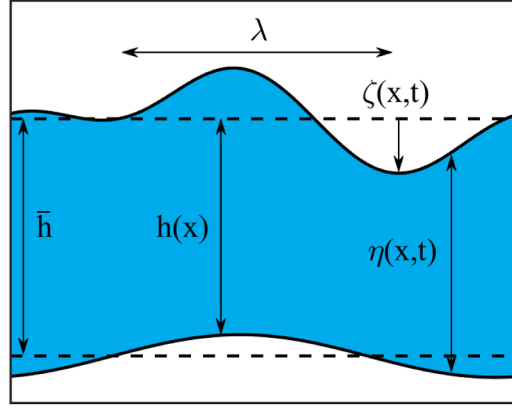


Fig. 1. Domain (not to scale) with length scales and variables.

For the remainder of the paper, we non-dimensionalize all variables with respect to g and $\bar{h}$, or, equivalently, set $g = 1$, $\bar{h} = O(1)$, and drop the bar on $\bar{u}$ and use $\bar{u}$ for the depth-averaged velocity.

2.2. Constraint form of the SGN equations

We now recast (2.3) into two evolution equations plus a PDE constraint. For classical solutions with sufficient regularity, the resulting constraint form is equivalent to (2.3).

First, multiply the velocity equation in (2.3) by η and group terms involving $\bar{u}_t$ on the left to obtain

$$\begin{aligned} \mathcal{G}_{\eta,h} \bar{u}_t + \eta \bar{u} \cdot \nabla \bar{u} + \eta \nabla \zeta &= \nabla \left\{ \frac{1}{3} \eta^3 \left(\bar{u} \cdot \nabla (\nabla \cdot \bar{u}) - (\nabla \cdot \bar{u})^2 \right) + \frac{1}{2} \eta^2 (\bar{u} \cdot \nabla)^2 h \right\} \\ &\quad - \eta \left\{ \frac{1}{2} \eta \left(\bar{u} \cdot \nabla (\nabla \cdot \bar{u}) - (\nabla \cdot \bar{u})^2 \right) + (\bar{u} \cdot \nabla)^2 h \right\} \nabla h, \end{aligned} \quad (2.5)$$

where

$$\mathcal{G}_{\eta,h} \bar{u} \equiv \eta \bar{u} - \nabla \left(\frac{1}{3} \eta^3 \nabla \cdot \bar{u} \right) - \nabla \left(\frac{1}{2} \eta^2 \nabla h \cdot \bar{u} \right) + \frac{1}{2} \eta^2 \nabla h (\nabla \cdot \bar{u}) + \eta \nabla h (\nabla h \cdot \bar{u}). \quad (2.6)$$

Note that the last three terms in $\mathcal{G}_{\eta,h}$ encapsulate the bathymetry effect and vanish when $\nabla h = 0$.

Next, commute the time derivative $\mathcal{G}_{\eta,h}(\bar{u}_t) = (\mathcal{G}_{\eta,h} \bar{u})_t - \mathcal{G}_{\eta,h} \bar{u}_t$ in (2.5), and use the η equation in (2.3) to express $(\mathcal{G}_{\eta,h})_t$ in terms of $\eta, \bar{u}, h$. After introducing the new variable $\bar{U} = \mathcal{G}_{\eta,h} \bar{u}$, one obtains the SGN equations (or DSWEs) in *constraint form*

$$\eta(\vec{x}, t) = h(\vec{x}) + \zeta(\vec{x}, t), \quad (2.7a)$$

$$\eta_t + \nabla \cdot (\eta \bar{u}) = 0, \quad (2.7b)$$

$$\bar{U}_t + \eta \nabla \zeta + \nabla (\eta \bar{u} \otimes \bar{u}) = \bar{F}(\eta, \bar{u}, h), \quad (2.7c)$$

$$\bar{U} - \mathcal{G}_{\eta,h} \bar{u} = 0, \quad (2.7d)$$

where the forcing $\bar{F}(\eta, \bar{u}, h)$ is

$$\begin{aligned} \bar{F}(\eta, \bar{u}, h) &\equiv \nabla \left\{ \frac{1}{3} \eta^3 \left(\bar{u} \cdot \nabla (\nabla \cdot \bar{u}) - (\nabla \cdot \bar{u})^2 \right) + \eta^2 \nabla \cdot (\eta \bar{u}) \nabla \cdot \bar{u} \right. \\ &\quad \left. + \frac{1}{2} \eta^2 (\bar{u} \cdot \nabla)^2 h + \eta \nabla \cdot (\eta \bar{u}) \nabla h \cdot \bar{u} \right\} \\ &\quad - \nabla h \left\{ \frac{1}{2} \eta^2 \left(\bar{u} \cdot \nabla (\nabla \cdot \bar{u}) - (\nabla \cdot \bar{u})^2 \right) + \eta \nabla \cdot (\eta \bar{u}) (\nabla \cdot \bar{u}) \right. \\ &\quad \left. + \eta (\bar{u} \cdot \nabla)^2 h + \nabla \cdot (\eta \bar{u}) (\nabla h \cdot \bar{u}) \right\}. \end{aligned} \quad (2.8)$$

In equation (2.7), the term involving $\bar{u} \otimes \bar{u}$ has components

$$\nabla (\eta \bar{u} \otimes \bar{u}) = \begin{pmatrix} (\eta u u)_x & (\eta u v)_y \\ (\eta u v)_x & (\eta v v)_y \end{pmatrix} \quad \text{where} \quad \bar{u} = (u, v).$$

The presence of the constraint in (2.7) highlights that spatial discretizations (e.g., via method of lines) will lead to an index-1 differential algebraic equation (DAE).

For the numerical computations (Section 6) of the SGN equations, we adopt periodic ($d = 1$) or doubly periodic ($d = 2$) boundary conditions for η and $\bar{u}$. However, the main results for the operator $\mathcal{G}_{\eta,h}$ in Section 4 will be formulated for both periodic and homogeneous Neumann (i.e., $\bar{n} \cdot \bar{u} = 0$ on $\partial\Omega$) boundary conditions.

Remark 1. [Conservation equation when $\nabla h = 0$] When the bathymetry is flat, i.e., $\nabla h = 0$, equations (2.7) yield a conservative system. In this case, $\bar{F} = \nabla W$ reduces to a gradient of a scalar $W(\eta, \bar{u})$, and $\nabla \eta = \nabla \zeta$. The time-evolution equations become

$$\eta_t + \nabla \cdot (\eta \bar{u}) = 0, \quad (2.9a)$$

$$\bar{U}_t + \nabla \cdot \left(\frac{1}{2} \eta^2 + \eta \bar{u} \bar{u} - W \right) = 0, \quad (2.9b)$$

$$\bar{U} - \mathcal{G}_\eta \bar{u} = 0. \quad (2.9c)$$

The system (2.9) conserves the mass m , horizontal momentum $\bar{M}$, and total energy E defined by

$$m = \int_{\Omega} \zeta \, d\vec{x}, \quad \bar{M} = \int_{\Omega} \bar{U} \, d\vec{x}, \quad E = \frac{1}{2} \int_{\Omega} \left[\zeta^2 + \eta (\bar{u} \cdot \bar{u}) + \frac{1}{3} \eta^3 (\nabla \cdot \bar{u})^2 \right] d\vec{x}.$$

♣

3. Mathematical background for the constraint equation

This section collects the mathematical notation and background relevant for the linear operators in the paper. Specifically, we introduce (Section 3.1) the bilinear (weak) form of the operator $\mathcal{G}_{\eta,h}$, and the operator $\mathcal{A}$. Section 3.2 provides background on conditioning numbers relevant for the use in preconditioned conjugate gradient (PCG) methods.

3.1. Linear operators and bilinear forms

Throughout, $\Omega \subseteq \mathbb{R}^2$ is either an open connected domain with Lipschitz boundary $\partial\Omega$, or $\Omega = \mathbb{T}^2$ is the (doubly-periodic) torus with side lengths L_x, L_y . Let

$$\mathcal{A} = \sigma I - \alpha \nabla \cdot \nabla$$

where $\sigma, \alpha > 0$ are real constants (Section 4 will provide formulas for (σ, α)).

The bilinear forms associated to the differential operators $\mathcal{G}_{\eta,h}$ and $\mathcal{A}$ will be defined on the Hilbert space $H_{\text{div}}(\Omega)$ given by

$$H_{\text{div}}(\Omega) \equiv \{ \bar{u} \in [L^2(\Omega)]^2 \mid \nabla \cdot \bar{u} \in L^2(\Omega), \bar{n} \cdot \bar{u} = 0 \text{ on } \partial\Omega \},$$

which is endowed with the norm

$$\|\bar{u}\|_{\text{div}}^2 \equiv \int_{\Omega} [|\bar{u}|^2 + (\nabla \cdot \bar{u})^2] \, d\vec{x}.$$

When $\partial\Omega$ is Lipschitz, the outward boundary unit normal, $\bar{n}(\vec{x})$, is defined for almost every $\vec{x} \in \partial\Omega$, and the normal trace of functions $\bar{u} \in H_{\text{div}}(\Omega)$ is understood to be $\bar{n} \cdot \bar{u} = 0$ in $H^{-1/2}(\partial\Omega)$, e.g., see [[38], Chapter 3.5.2]. When $\Omega = \mathbb{T}^2$, the boundary conditions in $H_{\text{div}}(\Omega)$ (i.e., $\bar{n} \cdot \bar{u} = 0$ on $\partial\Omega$) are replaced by periodic boundary conditions on $\bar{u}$.

Although the SGN equation solution $\eta = \eta(\vec{x}, t)$ is a function of t , for the purposes of preconditioning, t enters only as a parameter in $\mathcal{G}_{\eta,h}$. When discussing the weak forms we therefore suppress the variable t , i.e., treat $\eta(\vec{x})$ as a function of $\vec{x}$ only.

The definition of the bilinear forms will make use of the following regularity and boundedness assumption on the coefficients

$$\eta(\vec{x}), h(\vec{x}), \nabla h(\vec{x}) \in L^\infty(\Omega), \quad \text{and} \quad \eta(\vec{x}) \geq \eta_{\min} > 0. \quad (3.1)$$

Namely, h has one weak derivative bounded in L^∞ , and the essential infimum of η is positive. This second assumption excludes some physically relevant scenarios such as the modeling of “dry” states where $\eta(\vec{x}, t) = 0$ in regions of the domain. Throughout we keep $\sigma > 0, \alpha > 0$ as arbitrary constant values and use σ^*, α^* to refer to our concrete formulas used in the operator $\mathcal{A}$.

The bilinear forms $A : H_{\text{div}}(\Omega) \times H_{\text{div}}(\Omega) \rightarrow \mathbb{R}$, and $B : H_{\text{div}}(\Omega) \times H_{\text{div}}(\Omega) \rightarrow \mathbb{R}$ corresponding to the weak forms of the operators $\mathcal{A}$ and $\mathcal{G}_{\eta,h}$ respectively are:

$$A[\bar{u}, \bar{v}] \equiv \int_{\Omega} [\sigma \bar{u} \cdot \bar{v} + \alpha (\nabla \cdot \bar{v}) (\nabla \cdot \bar{u})] \, d\vec{x} \quad \forall \bar{u}, \bar{v} \in H_{\text{div}}(\Omega), \quad (3.2)$$

and

$$B[\bar{u}, \bar{v}] \equiv \int_{\Omega} \left[\eta \bar{u} \cdot \bar{v} + \frac{1}{3} \eta^3 (\nabla \cdot \bar{u}) (\nabla \cdot \bar{v}) + \frac{1}{2} \eta^2 (\nabla \cdot \bar{v}) (\nabla h \cdot \bar{u}) \right. \\ \left. + \frac{1}{2} \eta^2 (\nabla \cdot \bar{u}) (\nabla h \cdot \bar{v}) + \eta (\nabla h \cdot \bar{u}) (\nabla h \cdot \bar{v}) \right] d\vec{x} \quad \forall \bar{u}, \bar{v} \in H_{\text{div}}(\Omega). \quad (3.3)$$

The bilinear forms arise in the usual fashion from the strong form of the PDE through integration against a test function, e.g., $A[\bar{u}, \bar{v}] = (\bar{u}, \mathcal{A}\bar{v})$ and $B[\bar{u}, \bar{v}] = (\bar{u}, \mathcal{G}_{\eta,h}\bar{v})$, for $\bar{u}, \bar{v}$ (where $(\cdot, \cdot)$ is the L^2 inner product) being twice continuously differentiable with vanishing normal component (cf. [[39], Chapter 12 for additional details on operator extensions via bilinear forms]).

It is clear from the definitions that both $A[\cdot, \cdot]$ and $B[\cdot, \cdot]$ are symmetric, i.e., $A[\bar{u}, \bar{v}] = A[\bar{v}, \bar{u}]$ and $B[\bar{u}, \bar{v}] = B[\bar{v}, \bar{u}]$. Moreover, $A[\cdot, \cdot]$ is positive definite ($A[\bar{u}, \bar{u}] > 0$ for $\bar{u} \neq 0$), and it is the case that $B[\cdot, \cdot]$ is positive definite as well.

We will often write $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$ etc. These expressions are understood via the following definitions

1. (The operator $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$) For $\vec{f} \in H_{\text{div}}(\Omega)$, the function $\vec{w} = \mathcal{A}^{-1}\mathcal{G}_{\eta,h}\vec{f}$ is defined by: Find $\vec{w} \in H_{\text{div}}(\Omega)$ such that

$$A[\vec{w}, \vec{v}] = B[\vec{f}, \vec{v}] \quad \forall \vec{v} \in H_{\text{div}}(\Omega). \quad (3.4)$$

2. (The operator $\mathcal{G}_{\eta,h}^{-1}\mathcal{A}$) For $\vec{f} \in H_{\text{div}}(\Omega)$, the function $\vec{w} = \mathcal{G}_{\eta,h}^{-1}\mathcal{A}\vec{f}$ is defined by: Find $\vec{w} \in H_{\text{div}}(\Omega)$ such that

$$B[\vec{w}, \vec{v}] = A[\vec{f}, \vec{v}] \quad \forall \vec{v} \in H_{\text{div}}(\Omega). \quad (3.5)$$

3. (Generalized eigenvalues) The generalized eigenvalue problem $\mathcal{G}_{\eta,h}\vec{u} = \lambda\mathcal{A}\vec{u}$ is: Find $(\vec{u}, \lambda) \in H_{\text{div}}(\Omega) \times \mathbb{R}$ ($\vec{u} \neq 0$) such that

$$\lambda A[\vec{u}, \vec{v}] = B[\vec{u}, \vec{v}] \quad \forall \vec{v} \in H_{\text{div}}(\Omega). \quad (3.6)$$

The fact that both $\lambda \in \mathbb{R}$ and $\vec{u}$ can be restricted to a real-valued function in $H_{\text{div}}(\Omega)$ in (3.6) follows from the symmetry of the bilinear forms.

3.2. Conjugate gradient and conditioning numbers

Conjugate gradient (CG) is an iterative Krylov method for solving linear matrix systems of the form $A\vec{x} = \vec{f}$ where $A \in \mathbb{R}^n$ is symmetric positive definite (in exact arithmetic, CG converges after a finite number of iterations). A straight-forward, but not necessarily sharp [[40], Chapter 38], bound on the convergence of CG is given in terms of the conditioning number $\kappa(A) \equiv \lambda_{\max}/\lambda_{\min}$ where $\lambda_{\max}$, $\lambda_{\min}$ are the largest and smallest eigenvalues of A :

$$\|\vec{e}_k\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|\vec{e}_0\|_A, \quad \|\vec{x}\|_A \equiv \vec{x}^T A \vec{x}. \quad (3.7)$$

In (3.7) $\vec{e}_k = \vec{x}_k - \vec{x}^*$ is the error after k iterations of CG, $\vec{x}^* = A^{-1}\vec{f}$ is the exact solution.

The convergence of CG may be improved (such as in settings where A has a large conditioning numbers) by solving a preconditioned system with a symmetric positive definite P . Applying CG to the following system, which shares the same solution as $A\vec{x} = \vec{f}$,

$$P^{-1/2}AP^{-1/2}\vec{y} = P^{-1/2}\vec{f} \quad \text{with} \quad \vec{y} = P^{1/2}\vec{x}, \quad (3.8)$$

results in a preconditioned conjugate gradient (PCG) method. It is worth emphasizing that practical implementations of PCG avoid computations of matrix square roots and only require linear system solves involving P . The conditioning number of the PCG matrix in (3.8) is then modified from that of A to be the ratio of maximum to minimum eigenvalues of $P^{-1}A$. Namely, the conditioning number, and upper bound on the convergence rate may be (significantly) reduced if P is “close” to A .

When $\mathcal{A}$ is used as a preconditioner for $\mathcal{G}_{\eta,h}$, the operator analogue of κ for symmetric operators is

$$\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h}) \equiv \frac{\mu_{\max}}{\mu_{\min}}, \quad (3.9)$$

where $\mu_{\min}$ and $\mu_{\max}$ are the (generalized) Rayleigh quotients

$$\mu_{\min} = \inf_{0 \neq \vec{u} \in H_{\text{div}}(\Omega)} \frac{B[\vec{u}, \vec{u}]}{A[\vec{u}, \vec{u}]} \quad \text{and} \quad \mu_{\max} = \sup_{0 \neq \vec{u} \in H_{\text{div}}(\Omega)} \frac{B[\vec{u}, \vec{u}]}{A[\vec{u}, \vec{u}]} \quad (3.10)$$

The quantity $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h})$ measures the ratio of the largest to smallest values in the spectrum of $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$. Not only does $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h})$ play the operator analogue of the PCG matrix conditioning number, but it also appears as a bound in the convergence of operator/Hilbert space versions of preconditioned gradient descent (see [41]) and PCG (e.g., variable coefficient Laplacian [42]). A smaller value of $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h})$ will yield a tighter bound on the convergence of PCG.

Bounds on the continuum operators also carry over to the discrete setting as well.

Remark 2 (Conditioning number bound for spatially discrete problems). When the operator bilinear forms $A[\cdot, \cdot]$ and $B[\cdot, \cdot]$ are discretized via finite element methods, the quantity $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h})$ is a mesh-independent upper bound on the discrete matrix preconditioned conditioning number (see also [41]).

Namely, let $V \subset H_{\text{div}}(\Omega)$ be any finite dimensional subspace with basis $\{v_i\}_{i=1}^n \in V$. The discrete matrices that arise from the weak form (3.6) are symmetric and positive definite

$$A_{ij} = A[v_i, v_j] \quad G_{ij} = B[v_i, v_j] \quad \text{for } i, j = 1, \dots, n.$$

Then $\kappa(A^{-1}G) \leq \kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h})$. In particular, the discrete conditioning number $\kappa(A^{-1}G)$ is bounded independent of the mesh. This ensures that the convergence rate of PCG is bounded independent of the mesh. ♣

4. Preconditioner for the operator $\mathcal{G}_{\eta,h}$

In this section we introduce the constant coefficient preconditioner

$$\mathcal{A}\vec{u} = \sigma^* \vec{u} - \alpha^* \nabla(\nabla \cdot \vec{u}), \quad (4.1)$$

and establish bounds on the generalized eigenvalues and conditioning number of the weak form of the operator $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$ as defined in (3.4). These bounds will play an important role in ensuring convergence for preconditioned conjugate gradient methods or linear stability in semi-implicit time-stepping.

The coefficients in (4.1) for $\mathcal{A}$ are defined in terms of pointwise¹ estimates of η and h

$$|\nabla h|_{\max} \equiv \max_{\vec{x} \in \Omega} |\nabla h(\vec{x})|, \quad \eta_{\min} \equiv \min_{\vec{x} \in \Omega} \eta(\vec{x}), \quad \eta_{\max} \equiv \max_{\vec{x} \in \Omega} \eta(\vec{x}). \quad (4.2)$$

- For flat bathymetry ($|\nabla h|_{\max} = 0$) we set

$$\sigma^* \equiv \eta_{\max}, \quad \alpha^* \equiv \frac{1}{3}\eta_{\max}^3, \quad \kappa_{\text{ub}} \equiv \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3. \quad (4.3)$$

- For variable bathymetry ($|\nabla h|_{\max} \neq 0$) we take

$$\sigma^* \equiv \max_{\vec{x} \in \Omega} \left(\eta(\vec{x}) + \frac{1}{6}(4 + \sqrt{13})\eta(\vec{x})|\nabla h(\vec{x})|^2 \right), \quad \alpha^* \equiv \frac{1}{6}(4 + \sqrt{13})\eta_{\max}^3, \quad (4.4)$$

$$\kappa_{\text{ub}} \equiv \max \left\{ \frac{\sigma^*}{\eta_{\min}}, \left(\frac{4 + \sqrt{13}}{4 - \sqrt{13}} \right) \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3 \right\}. \quad (4.5)$$

Note that $|\nabla h|_{\max}$ quantifies the maximum slope of the bathymetry with $|\nabla h|_{\max} = 0$ corresponding to a flat bottom. When solving the SGN equations, $|\nabla h|_{\max}$ can be determined as a one-time computation since $h(\vec{x})$ is prescribed as part of the problem data.

The significance of the coefficients is highlighted by the next theorem.

Theorem 1. Let $A, B : H_{\text{div}} \times H_{\text{div}} \rightarrow \mathbb{R}$ be the bilinear forms (3.2) and (3.3) corresponding to the operators $\mathcal{A}$ and $\mathcal{G}_{\eta,h}$ with coefficients satisfying (4.3)-(4.4) and (3.1) respectively. Then the condition number $\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})}$ of $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$, as defined in (3.9)-(3.10), is bounded by

$$\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})} \leq \kappa_{\text{ub}}. \quad (4.6)$$

Moreover, the generalized eigenvalues λ of $\mathcal{G}_{\eta,h} \vec{u} = \lambda \mathcal{A} \vec{u}$, as defined by (3.6), are real and lie in the interval

$$(\kappa_{\text{ub}})^{-1} \leq \lambda \leq 1. \quad (4.7)$$

The proof of Theorem 1 is deferred to Appendix A.

The inequalities (4.6) and (4.7) are important for several reasons. Firstly, smaller values of $\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})}$ are desirable as they provide stronger performance bounds for preconditioned conjugate gradient when solving systems involving $\mathcal{G}_{\eta,h}$. Without the use of a preconditioner $\mathcal{A}$, the eigenvalues of $\mathcal{G}_{\eta,h}$ would be unbounded and bounds such as (4.6) and (4.7) would not exist. This would render methods such as conjugate gradient far less effective (or fail entirely) for solving $\mathcal{G}_{\eta,h} \vec{u} = \vec{f}$. While the bounds (4.6) and (4.7) are formulated in terms of PDE operators, they provide mesh-independent PCG convergence rates for the spatially discrete problem as well (see Remark 2). Secondly, when semi-implicit time-stepping approaches are used to solve the SGN equations, the clustering of eigenvalues λ near 1 is important to ensure linear stability.

The formula (4.5) highlights two ways in which the upper bound on $\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})}$ may diverge and lead to a loss in practical effectiveness of $\mathcal{A}$

- (Large contrast ratio) If $\eta_{\max}/\eta_{\min} \gg 1$, and the second term in (4.5) dominates then $\kappa_{\text{ub}} \sim (\eta_{\max}/\eta_{\min})^3$.
- (Large variations in bathymetry) If $\max_{\vec{x} \in \Omega} \eta(\vec{x})|\nabla h(\vec{x})|^2 \gg 1$ so that the first term in (4.5) dominates, then at worst $\kappa_{\text{ub}} \sim \eta_{\max} |\nabla h|_{\max}^2$.

We conclude this section with several remarks.

Remark 3 (Robustness of the coefficients σ^*, α^*). One may ask if alternative formulas may be used for (σ, α) and still yield an upper bound on $\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})}$? It turns out that $\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})}$ is bounded for any positive coefficients $\sigma, \alpha > 0$ in $\mathcal{A}$. Alternative choices of σ, α may however yield larger values of $\kappa_{\mathcal{A}(\mathcal{G}_{\eta,h})}$ and decrease the efficiency of the preconditioner $\mathcal{A}$ or increase the error constant in semi-implicit time integration. In practice, this robustness allows for some flexibility in under or overestimating the formulas for σ^*, α^* . It turns out, see Appendix B, that the choice of $\mathcal{A}$ is *quasi-optimal* within a class of constant coefficient operators. ♣

Remark 4. (Replacing $\eta(\vec{x})|\nabla h(\vec{x})|^2$ with $\eta_{\max} |\nabla h|_{\max}^2$ in $\mathcal{A}$) The coefficient σ^* relies on the maximum pointwise bound for $\eta(\vec{x})|\nabla h(\vec{x})|^2$. In some practical settings, it may be simpler to individually estimate $|\nabla h|_{\max}$ and $\eta_{\max}$ separately. Instead of (4.4), one can take: $\mathcal{A} = \sigma^{**} I - \alpha^{**} \nabla \nabla \cdot$ where

$$\sigma^{**} = \eta_{\max} \left(1 + \frac{1}{6}(4 + \sqrt{13})|\nabla h|_{\max}^2 \right), \quad \alpha^{**} = \frac{1}{6}(4 + \sqrt{13})\eta_{\max}^3. \quad (4.8)$$

These coefficients still ensure the eigenvalue bound (4.7) and conditioning number bound (4.6) where κ_{ub} is replaced with a (less optimal) bound

$$\kappa_{\text{ub}} = \max \left\{ \frac{\eta_{\max}}{\eta_{\min}} \left(1 + \frac{1}{6}(4 + \sqrt{13})|\nabla h|_{\max}^2 \right), \left(\frac{4 + \sqrt{13}}{4 - \sqrt{13}} \right) \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3 \right\}. \quad (4.9)$$

¹ We have written max and min here for simplicity, however, in the general case when $h \in W^{1,\infty}(\Omega)$, $\eta \in L^\infty(\Omega)$, the max and min are taken as essential supremum and infimum.

Unlike (4.4), these coefficients are not *quasi-optimal* in the sense discussed in Appendix B, and can result in a drop in PCG performance (see Section 6.2). However, they are still effective coefficients for both PCG and to define the implicit $\mathcal{A}$ in semi-implicit time-stepping. ♣

Remark 5 (Riesz Preconditioner). The choice of $A[\cdot, \cdot]$ can be viewed as designing a Riesz preconditioner for the operator $B[\cdot, \cdot]$; for a discussion of Riesz preconditioners in the context of bilinear forms see [41–43]. ♣

5. Time-stepping for the SGN equations

This section provides details on the time integration schemes we use to solve (2.7). Sections 5.1 and 5.2 present explicit Runge-Kutta (ERK) and multistep methods that handle the constraint through preconditioned conjugate gradient (PCG) methods. Section 5.3 presents linearly implicit multistep (ImEx-type) schemes that treat $\mathcal{A}$ implicitly and $\mathcal{G}_{\eta,h}$ explicitly.

The linearly implicit multistep schemes admit several advantages—they require only one linear solve involving $\mathcal{A}$ per time step and they do not suffer from order reduction (cf. Remark 7). However, linear implicit schemes do require a precise choice of $\mathcal{A}$ to ensure zero-stability, which will be discussed in Section 5.4.

In this section, it is useful to view the SGN system (2.7) as an evolution equation with a constraint

$$\frac{\partial}{\partial t} \begin{pmatrix} \eta \\ \vec{U} \end{pmatrix} = \begin{pmatrix} -\nabla \cdot (\eta \vec{u}) \\ -\eta \nabla \zeta - \nabla (\eta \vec{u} \otimes \vec{u}) + \vec{F} \end{pmatrix}, \quad (5.1a)$$

$$\mathcal{G}_{\eta,h} \vec{u} = \vec{U}, \quad (5.1b)$$

where $\vec{F}$ is defined in (2.8). Spatial discretizations of (5.1) via a method-of-lines give rise to a differential algebraic equation (DAE)

$$\vec{w}_t = \vec{f}(\vec{w}, \vec{p}), \quad (5.2a)$$

$$\vec{g}(\vec{w}, \vec{p}) = 0, \quad (5.2b)$$

where $\vec{w} \in \mathbb{R}^n$, $\vec{p} \in \mathbb{R}^m$ and $\vec{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $\vec{g} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$. Specifically, the ODE variables are spatial approximations given by $\vec{w} \approx (\eta, \vec{U})^T$, $\vec{p} \approx \vec{u}$, $\vec{f} \approx (-\nabla \cdot (\eta \vec{u}), -\eta \nabla \zeta - \nabla (\eta \vec{u} \otimes \vec{u}) + \vec{F})^T$ and $\vec{g} \approx \mathcal{G}_{\eta,h} \vec{u} - \vec{U}$.

In general, (5.2) will be an *index-1* DAE since the constraint Eq. (5.2b) can be solved to implicitly define $\vec{p}$ as a function of $\vec{w}$. This is because $\mathcal{G}_{\eta,h}$ is an invertible operator (Section 4), and matrix approximations to $\mathcal{G}_{\eta,h}$ constructed by testing $B[\cdot, \cdot]$ with Galerkin or finite element subspaces of H_{div} are also invertible.

For simplicity we adopt constant time steps $\Delta t > 0$ where $t_n = n\Delta t$, however the results for Runge-Kutta methods naturally extend to variable step sizes. The fluid variables at time t_n are written as $\vec{u}_n \approx \vec{u}(t_n)$, $\eta_n \approx \eta(t_n)$, $\zeta_n \approx \zeta(t_n)$, and $\vec{U}_n \approx \vec{U}(t_n)$.

Remark 6. While the focus here is on time discretizations of (2.7), the same discretization approaches can be applied to the non-constraint form of the equations (2.3). ♣

5.1. Explicit Runge-Kutta with preconditioned conjugate gradient

We discretize the SGN equations (2.7) via the classical explicit RK4 scheme defined by the Butcher coefficients [[44], Chapter 5]

$$(a_{ij})_{i,j=1}^4 = \begin{pmatrix} \frac{1}{2} & & & \\ 0 & \frac{1}{2} & & \\ 0 & 0 & 1 & \end{pmatrix}, \quad (b_i)_{i=1}^4 = \left(\frac{1}{6} \quad \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{6} \right)^T, \quad (c_i)_{i=1}^4 = \left(0 \quad \frac{1}{2} \quad \frac{1}{2} \quad 1 \right)^T.$$

This yields the following equations for the stages ($i = 1, \dots, 4$)

$$\eta_n^{(i)} = \eta_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(-\nabla \cdot (\eta_n^{(j)} \vec{u}_n^{(j)}) \right), \quad (5.3a)$$

$$\vec{U}_n^{(i)} = \vec{u}_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \left(-\eta_n^{(j)} \nabla \zeta_n^{(j)} - \nabla (\eta_n^{(j)} \vec{u}_n^{(j)} \otimes \vec{u}_n^{(j)}) + \vec{F}_n^{(j)} \right), \quad (5.3b)$$

$$\mathcal{G}_{\eta_n^{(i)},h} \vec{u}_n^{(i)} = \vec{U}_n^{(i)}, \quad (5.3c)$$

where $\zeta_n^{(i)} = \eta_n^{(i)} - h$. The final update is

$$\eta_{n+1} = \eta_n + \Delta t \sum_{j=1}^4 b_j \left(-\nabla \cdot (\eta_n^{(j)} \vec{u}_n^{(j)}) \right),$$

$$\vec{u}_{n+1} = \vec{u}_n + \Delta t \sum_{j=1}^4 b_j \left(-\eta_n^{(j)} \nabla \zeta_n^{(j)} - \nabla (\eta_n^{(j)} \vec{u}_n^{(j)} \otimes \vec{u}_n^{(j)}) + \vec{F}_n^{(j)} \right).$$

Table 1
Multistep coefficients for r th order Adams-Bashforth (ABr) with $r = 2, 3, 4$, and 2nd order semi-implicit backward differentiation (SBDF2).

		$j = 4$	$j = 3$	$j = 2$	$j = 1$	$j = 0$
AB2	a_j	$\cdot$	$\cdot$	1	-1	0
	b_j	$\cdot$	$\cdot$	0	$\frac{3}{2}$	$-\frac{1}{2}$
AB3	a_j	$\cdot$	1	-1	0	0
	b_j	$\cdot$	0	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$
AB4	a_j	1	-1	0	0	0
	b_j	0	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$
SBDF2	a_j	$\cdot$	$\cdot$	$\frac{3}{2}$	-2	$\frac{1}{2}$
	c_j	$\cdot$	$\cdot$	1	0	0
	b_j	$\cdot$	$\cdot$	0	2	-1

While RK4 is an explicit method, the algebraic constraint in (5.3c) still requires a linear solve for $\mathcal{G}_{\eta,h}$ at each stage. To solve (5.3c), we apply PCG with the preconditioner $\mathcal{A}$ defined by the coefficients (4.4). Since each solve is independent, one can change the coefficients of $\mathcal{A}$ as needed for each stage or step.

Remark 7 (Order reduction in RK schemes). Runge-Kutta methods may exhibit a reduction in order when applied to initial boundary value problems. The reduction often stems from the presence of boundary conditions or a lack of regularity in the solution. Due to the periodic domain, the test cases we employ do not exhibit order reduction. However, the RK schemes presented in this section could exhibit order reduction when applied to problems with time-dependent boundary conditions (multistep methods do not exhibit order reduction). ♣

5.2. Explicit multistep with preconditioned conjugate gradient

Similar to explicit RK methods, we also test explicit Adams-Bashforth with PCG used to solve the constraint equation at each step. A general s -stage linear multistep method discretizes (5.2) with coefficients $(a_j, b_j)_{j=1}^s$ as

$$\frac{1}{\Delta t} \sum_{j=0}^s a_j \vec{w}_{n+j} = \sum_{j=0}^{s-1} b_j \vec{f}(\vec{w}_{n+j}, \vec{p}_{n+j}), \quad (5.4a)$$

$$\vec{g}(\vec{w}_{n+j}, \vec{p}_{n+j}) = 0. \quad (5.4b)$$

The coefficients for Adams-Bashforth (with orders 2, 3 and 4) are included in Table 1.

Applying the discretization to the SGN yields

$$\zeta_n = \eta_n - h, \quad (5.5a)$$

$$\frac{1}{\Delta t} \sum_{j=0}^s a_j \eta_{n+j} = - \sum_{j=0}^{s-1} b_j \nabla \cdot (\eta_{n+j} \vec{u}_{n+j}), \quad (5.5b)$$

$$\frac{1}{\Delta t} \sum_{j=0}^s a_j \vec{U}_{n+j} = \sum_{j=0}^{s-1} b_j \left(-\eta_{n+j} \nabla \zeta_{n+j} - \nabla(\eta_{n+j} \vec{u}_{n+j} \otimes \vec{u}_{n+j}) + \vec{F}(\eta_{n+j}, \vec{u}_{n+j}, h) \right), \quad (5.5c)$$

$$\mathcal{G}_{\eta_{n+j}, h} \vec{u}_{n+j} = \vec{U}_{n+j}. \quad (5.5d)$$

When solving this system, we again use the preconditioner $\mathcal{A}$ with PCG to solve the constraint equation at each time step.

5.3. A linearly implicit method for the SGN equations

In contrast to the previous two subsections, this subsection provides a linearly implicit treatments of $\mathcal{G}_{\eta,h}$ in (5.5). The starting point is to split the algebraic equation by first adding and subtracting a linear term $A\vec{p}$ to obtain

$$A\vec{p} - A\vec{p} + \vec{g}(\vec{w}, \vec{p}) = 0. \quad (5.6)$$

Next, using a set of coefficients $(c_j)_{j=0}^s$, extrapolate the first term in (5.6) at time $t + \tau$,

$$A\vec{p}(t + \tau) \approx \sum_{j=0}^s c_j A\vec{p}_{n+j} + \mathcal{O}(\Delta t^p). \quad (5.7)$$

The coefficient $c_s \neq 0$ ensures the approximation of $A\bar{p}$ will lead to an implicit equation in $\bar{p}$. The remaining term $A\bar{p} - \bar{g}(\bar{w}, \bar{p})$ is extrapolated to time $t + \tau$ via

$$A\bar{p}(t + \tau) - \bar{g}(\bar{w}(t + \tau), \bar{p}(t + \tau)) \approx \sum_{j=0}^{s-1} b_j A\bar{p}_{n+j} - b_j \bar{g}(\bar{w}_{n+j}, \bar{p}_{n+j}) + \mathcal{O}(\Delta t^p). \quad (5.8)$$

The Eq. (5.4b) in (5.4) is then replaced by

$$0 = \sum_{j=0}^s c_j A\bar{p} + \sum_{j=0}^{s-1} b_j \left(\bar{g}(\bar{w}_{n+j}, \bar{p}_{n+j}) - A\bar{p}_{n+j} \right), \quad (5.9)$$

which is obtained from equating (5.7) and (5.8). By construction the scheme (5.4a) with (5.9) is linearly implicit in A ; no solution involving $\bar{g}$ is required.

We use the SBDF2 coefficients for a_j, b_j, c_j presented in Table 1, together with the time-discretization (5.9) where $A \approx \mathcal{A}$ to yield the linearly implicit scheme

$$\zeta_n = \eta_n - h, \quad (5.10a)$$

$$\frac{1}{\Delta t} \sum_{j=0}^s a_j \eta_{n+j} = - \sum_{j=0}^{s-1} b_j \nabla \cdot (\eta_{n+j} \bar{u}_{n+j}), \quad (5.10b)$$

$$\frac{1}{\Delta t} \sum_{j=0}^s a_j \bar{U}_{n+j} = \sum_{j=0}^{s-1} b_j \left(-\eta_{n+j} \nabla \zeta_{n+j} - \nabla(\eta_{n+j} \bar{u}_{n+j} \otimes \bar{u}_{n+j}) + \bar{F}(\eta_{n+j}, \bar{u}_{n+j}, h) \right), \quad (5.10c)$$

$$\sum_{j=0}^s c_j \mathcal{A} \bar{u}_{n+j} = \sum_{j=0}^{s-1} b_j \left(\mathcal{A} \bar{u}_{n+j} - \mathcal{G}_{\eta_{n+j}, h} \bar{u}_{n+j} + \bar{U}_{n+j} \right). \quad (5.10d)$$

For extensions of the scheme to orders higher than 2, linearly implicit SBDF will not in general yield a zero-stable scheme (cf. [45] where the simultaneous choice of $\mathcal{A}$ and scheme coefficients must be used to obtain orders 3 and higher).

Remark 8 (Alternative derivation via ImEx scheme applied to a singular ODE). The scheme (5.10) may be formally derived by applying an implicit explicit multistep method to a singularly perturbed ODE related to the DAE as follows:

First replace the DAE by the singularly perturbed equation (cf. [46], Chapter VI.2])

$$\bar{w}_t = \bar{f}(\bar{w}, \bar{p}) \quad \epsilon \bar{p}_t = A\bar{p} + (\bar{g}(\bar{w}, \bar{p}) - A\bar{p}). \quad (5.11)$$

The scheme (5.9) then arises from an implicit-explicit (ImEx) time-discretization of (5.11) that treats (in the second equation) $A\bar{p}$ implicitly and $\bar{g} - A\bar{p}$ explicitly, followed by the formal limit $\epsilon \rightarrow 0$. As a result, the scheme coefficients (a_j, b_j, c_j) in (5.9) are guaranteed to be a p th order approximation provided they satisfy the order conditions [[47], Equation (7)] of a multistep ImEx method.

♣

5.4. Zero-stability for linear index-1 algebraic equations

In this subsection we establish, in the spirit of absolute stability, conditions for zero-stability of (5.9) applied to linear time-independent problems. Although we focus on constant time-independent problems, the result provides a guide for ensuring zero-stability of the equations (5.10). We say that the scheme is *zero-stable* if the dynamics (5.9) with $\Delta t = 0$ yield bounded iterates; zero-stability is a necessary condition for convergence.

Herein, assume a linear constraint $\bar{g}(\bar{w}, \bar{p}) = G\bar{p} - \bar{w}$, with invertible matrix $G \in \mathbb{R}^{m \times m}$. The scheme (5.9) with $\Delta t = 0$ reads:

$$\sum_{j=0}^s a_j \bar{w}_{n+j} = 0, \quad (5.12a)$$

$$\sum_{j=0}^s c_j A\bar{p}_{n+j} = \sum_{j=1}^{s-1} b_j (A\bar{p}_{n+j} - G\bar{p}_{n+j} + \bar{w}_{n+j}), \quad (5.12b)$$

where $(\bar{p}_j, \bar{w}_j)$ for $j = 1, \dots, s$ are given as initial data. Next, introduce the following polynomials defined in terms of the time-stepping coefficients

$$a(z) = \sum_{j=0}^s a_j z^j, \quad b(z) = \sum_{j=0}^{s-1} b_j z^j, \quad c(z) = \sum_{j=0}^s c_j z^j. \quad (5.13)$$

Eq. (5.12a) defines a stable linear recurrence for $\bar{w}_n$ provided $a(z)$ satisfies the *root condition*: the roots of $a(z)$ lie inside the unit disk with only simple roots on the boundary. Every practical linear multistep method (including both AB2, AB3, AB4 and SBDF2) has an $a(z)$ satisfying the root condition.

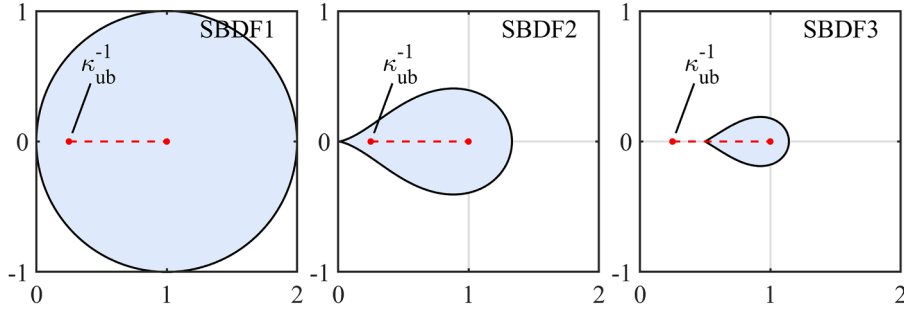


Fig. 2. Figures show stability region D (shaded blue) in relation to the bound on eigenvalues of $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$ from Theorem 1 (red). Note that SBDF1 and SBDF2 are stable, while SBDF3 will not be stable if the largest eigenvalue $\lambda > 2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

With $\vec{w}_n$ completely determined by (5.12a), Eq. (5.12b) defines a forced linear recurrence for $\vec{p}_n$. The stability of $\vec{p}_n$ then depends on the homogeneous Eq. (5.12b) (i.e., $\vec{w}_n \equiv 0$) which takes the form

$$\sum_{j=0}^s c_j A \vec{p}_{n+j} = \sum_{j=1}^{s-1} b_j (A \vec{p}_{n+j} - G \vec{p}_{n+j}). \quad (5.14)$$

A family of eigenvector solutions to (5.14) then follow as: let $(\vec{p}, \lambda)$ satisfy

$$\lambda A \vec{p} = G \vec{p} \quad (A \vec{p} \neq 0). \quad (5.15)$$

Substituting the ansatz $\vec{p}_n = z^n \vec{p}$ into (5.14) yields

$$c(z) - b(z) + \lambda b(z) = 0. \quad (5.16)$$

A sufficient condition for stability of (5.14) is then ensured if each solution of (5.16) has $|z| < 1$. Following in the spirit of absolute stability, introduce the region of zero stability as

$$D \equiv \{ \lambda \in \mathbb{C} \mid |z| < 1 \text{ holds for each } z \text{ satisfying (5.16)} \}. \quad (5.17)$$

Note that D is a property of the time-stepping coefficients only. One then has the following.

Theorem 2. (Sufficient conditions for zero-stability of DAEs with linear constraints) If $a(z)$ satisfies the root condition, and $\lambda \in D$ for each λ satisfying (5.15), then the dynamics (5.12) are stable.

Fig. 2 presents the stability regions D for SBDF2. For reference, the stability regions for SBDF1 and SBDF3, which are not used in Section 6, are shown. Up to a reflection in the imaginary axis and shift, the sets D are identical to those characterized in [48]. The figures show that SBDF1 and SBDF2 always contains the eigenvalues of $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$. In contrast, depending on the variables η and h , it is possible for the largest eigenvalue to lie outside the SBDF3 stability region—rendering SBDF3 unstable. Note that by modifying the SBD coefficients for orders 3, 4 and 5 as in [45], it is possible to always devise time-stepping schemes where the eigenvalues of $\mathcal{A}^{-1}\mathcal{G}_{\eta,h}$ lie in the stability regions D . We do not explore these approaches here.

6. Numerical tests

In this section we numerically test the performance and robustness of $\mathcal{A}$ as a preconditioner in dimensions $d = 1$ (Section 6.2) and $d = 2$ (Section 6.3). We then demonstrate the accuracy and effectiveness of the time integration schemes to solve the SGN in $d = 1$ (Section 6.4) and $d = 2$ (Section 6.5). All numerical computations are performed in MATLAB (Version R2023b) on a Lenovo ThinkPad P14s Gen2 laptop.²

For the purposes of presentation, all computations in this section are done with discretizing the constraint system (2.7). Numerical solutions to the (2.3) are similar but not shown.

The following exact solution will be used in various test cases. In free space with a flat bathymetry, $h(\vec{x}) = h_0$, the SGN equations admit a solitary wave solution of a single parameter $a > 0$ [4,49]:

$$\begin{aligned} \xi(x, y, t) &\equiv x \cos(\theta) + y \sin(\theta) - ct + \xi_0, \\ \eta_s(x, y, t) &= h_0 + a \operatorname{sech}^2(\gamma \xi(x, y)), \end{aligned} \quad (6.1)$$

² Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz; RAM: 32.0 GB; 64-bit operating system.

with velocity

$$\vec{u}_s(x, y, t) = w(x, y, t) (\cos \theta, \sin \theta), \quad (6.2)$$

$$w_s(x, y, t) = c \frac{1}{\eta_s(x, y, t)} (\eta_s(x, y, t) - h_0). \quad (6.3)$$

The speed c and width γ are given in terms of a and h_0 by

$$c = \sqrt{h_0 + a} \quad \text{and} \quad \gamma = \frac{\sqrt{3a}}{2h_0c}.$$

The variables $\theta \in [0, 2\pi)$ and $\xi_0 \in \mathbb{R}$ are independent of a and determine the orientation and center of the wave crest. In $d = 2$, the solitary wave is a one dimensional profile. A solitary wave in one dimension is obtained from the two dimensional wave by taking $\theta = 0$, $y = 0$ and a velocity w .

6.1. Spatial discretizations

Throughout this section we adopt a domain with periodic (in $d = 1$) or doubly-periodic ($d = 2$) boundary conditions, e.g.,

$$\eta(x, y + L_y) = \eta(x + L_x, y) = \eta(x, y), \quad \vec{u}(x, y + L_y) = \vec{u}(x + L_x, y) = \vec{u}(x, y), \quad (6.4)$$

where $\Omega = (0, L_x) \times (0, L_y)$ with L_x and L_y being the periods in the x - and y -directions, respectively. We then discretize space with an equispaced grid of n points in each direction, e.g., in dimension $d = 1$, the grid spacing Δx and mesh points are

$$\Delta x = \frac{L}{n} \quad x_j = j \Delta x \quad 0 = x_0 < x_1 < \dots < x_{n-1} = L - \Delta x.$$

Functions $\vec{u}(x)$ are discretized as $\vec{u}(x_j) \approx u_j$ with

$$\vec{u} = (u_0, u_1, \dots, u_{n-1})^T \in \mathbb{R}^n.$$

Function derivatives are obtained spectrally as

$$u_x(x_j) \approx (\mathbf{D}\mathbf{u})_j \quad \text{and} \quad \mathbf{D} = \mathbf{F}^{-1}(\mathbf{i}\mathbf{K})\mathbf{F},$$

where $\mathbf{F}$ is the discrete Fourier transform matrix and $\mathbf{K}$ is a diagonal matrix consisting of discrete wavenumbers of the form

$$k_j = \frac{2\pi}{L} j \quad j = -\frac{n}{2} + 1, \dots, \frac{n}{2} - 1.$$

We adopt the convention that the $n/2$ wavenumber is 0. The operators $\mathcal{G}_{\eta,h}$ and $\mathcal{A}$ are then discretized with spectrally accurate approximations (we suppress the parameter dependence of η and h on $\mathbf{G}$)

$$(\mathcal{G}_{\eta,h}\vec{u})(x_j) \approx (\mathbf{G}\mathbf{u})_j \quad (\mathcal{A}\vec{u})(x_j) \approx (\mathbf{A}\mathbf{u})_j \quad (j = 0, \dots, n-1). \quad (6.5)$$

For example, the matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \sigma^* \mathbf{I} + \alpha^* \mathbf{D}^\dagger \mathbf{D},$$

where $\mathbf{D}^\dagger = \overline{\mathbf{D}}^T$ is the conjugate transpose of the derivative matrix $\mathbf{D}$. By construction, both $\mathbf{A}$ and $\mathbf{G}$ are real symmetric and positive definite.

In presenting numerical results, we make use of both the (discrete) 2- and ∞ -norms

$$\|\mathbf{u}\|_{2,\Delta x}^2 = \Delta x \sum_{j=1}^n u_j^2 \quad \|\mathbf{u}\|_\infty = \max_{0 \leq i \leq n-1} |u_i|.$$

We also use the 2-norm induced by a matrix $\mathbf{M} > 0$ (which will generally be set to $\mathbf{G}$)

$$\|\mathbf{u}\|_{2,\Delta x,\mathbf{M}}^2 \equiv (\mathbf{u}^T \mathbf{M} \mathbf{u}) \Delta x \quad (\mathbf{u} \in \mathbb{R}^n).$$

With these notations, note that $\|\mathbf{u}\|_{2,\Delta x}^2 = \|\mathbf{u}\|_{2,\Delta x,\mathbf{I}}^2$. For the equation

$$\mathbf{G}\mathbf{u} = \mathbf{b}, \quad (6.6)$$

let $\mathbf{u}^* \equiv \mathbf{G}^{-1} \mathbf{b}$ be the exact solution. In convergence studies where preconditioned conjugate gradient is used to solve (6.6), we track the error

$$\varepsilon(\mathbf{u}) \equiv \frac{\|\mathbf{u} - \mathbf{u}^*\|_{2,\Delta x,\mathbf{G}}}{\|\mathbf{b}\|_{2,\Delta x}} = \frac{\|\mathbf{G}\mathbf{u} - \mathbf{b}\|_{2,\Delta x,\mathbf{G}^{-1}}}{\|\mathbf{b}\|_{2,\Delta x}}. \quad (6.7)$$

Here ε measures the discrete error of $\mathbf{u}$ (in the $\mathbf{G}$ norm), or equivalently the discrete residual $\mathbf{G}\mathbf{u} - \mathbf{b}$ (in the $\mathbf{G}^{-1}$ norm). Note that we use ε as a primary measure of the PCG error, as opposed to the 2-, or ∞ -norm, since PCG minimizes ε over progressively larger Krylov subspaces. This ensures that ε is non-increasing with each iteration of PCG.

The generalized eigenvalues of $\mathbf{G}$ and $\mathbf{A}$ are then

$$\mathbf{G}\mathbf{u}_k = \lambda_k \mathbf{A}\mathbf{u}_k \quad (\mathbf{u}_k \neq 0) \quad (6.8)$$

and, without loss of generality, can be ordered as

$$0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n.$$

For dimension $d = 2$, we extend all the definitions in the section in the natural way via tensor products.

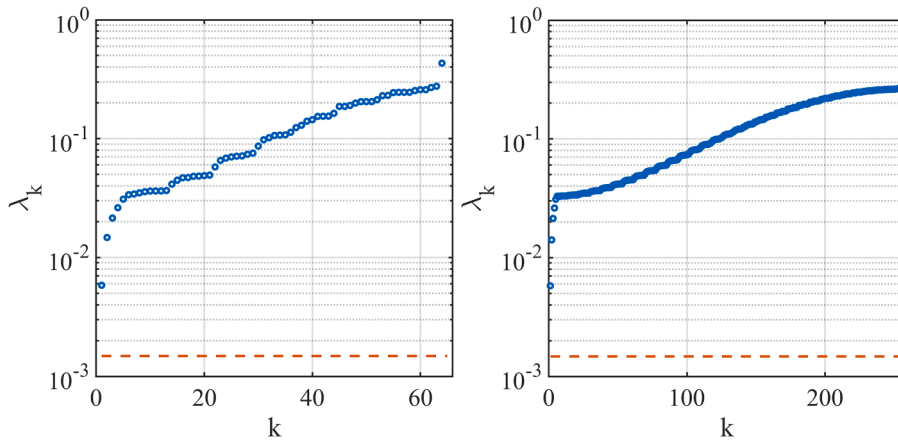


Fig. 3. Plot of the generalized eigenvalues λ_k versus k for the 1d test problem (6.10) with $\eta_0 = h_0 = 1$ and mesh resolutions $n = 64$ (left), $n = 256$ (right), as well as the lower bound κ_{ub}^{-1} (dashed line). The figures provide numerical evidence that the generalized eigenvalues of the operators $\mathcal{G}_{\eta,h}$ and $\mathcal{A}$ have a continuous spectrum confined to an interval $\sim [3 \times 10^{-2}, 3 \times 10^{-1}]$, with ~ 6 isolated eigenvalues outside this interval.

6.2. Preconditioner tests in 1D

In this section we test the performance of the preconditioner $\mathcal{A}$ with coefficients (4.3)-(4.4) using PCG to solve for u on the test problem ($d = 1$, $\Omega = [0, 1]$)

$$\mathcal{G}_{\eta,h}u = b \quad \text{where} \quad b(x) = \cos(4\pi x). \quad (6.9)$$

Here the variable coefficient functions are taken to be

$$h(x) = 1 + h_0 \exp(-(x - 1/2)^2/\sigma^2), \quad \sigma = 1/20 \quad (6.10)$$

$$\eta(x) = 1 + \eta_0 \cos(4\pi x)^2. \quad (6.11)$$

With these functions, we then vary the parameters $\eta_0, h_0 \in (-1, \infty)$ which control the water depth contrast ratio, and bathymetry gradients respectively ($h_0 = 0$ yielding a flat bathymetry). The preconditioner performance bound then depends on

$$\frac{\eta_{\max}}{\eta_{\min}} = \eta_0 + 1 \quad \text{and} \quad |h_x|_{\max} = \frac{\sqrt{2}}{e^{1/2}\sigma} |h_0| \approx 17.2 |h_0|.$$

When η_0 is large relative to $|h_x|_{\max}$, $\kappa_{ub} \sim (\eta_0 + 1)^3$. Alternatively, when $|h_x|_{\max}$ is large relative to η_0 , $\kappa_{ub} \sim (17.2)^2 \eta_0 h_0^2$.

The discrete form of (6.9) then is

$$\mathbf{G}\mathbf{u} = \mathbf{b} \quad \text{where} \quad \mathbf{b}_j = b(x_j) \quad (6.12)$$

with $\mathbf{G}$ defined in (6.5).

When solving (6.12), we initialize PCG with $\mathbf{u}^{(0)} = \mathbf{0}$, so that the initial error is $\varepsilon(\mathbf{0})$. In addition to an upper bound on the continuum operators $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h}) \leq \kappa_{ub}$, it can also be shown that the same upper bound holds for the discrete matrices $\mathbf{A}, \mathbf{G}$ defined by Section 6.1, i.e., $\kappa(\mathbf{A}^{-1}\mathbf{G}) \leq \kappa_{ub}$ (see Remark 2). As a result, if $\{\mathbf{u}^{(j)}\}_{j \geq 0}$ denotes the iterates of PCG, then (3.7) yields the upper bound

$$\varepsilon(\mathbf{u}^{(j)}) \leq 2 \left(\frac{\sqrt{\kappa_{ub}} - 1}{\sqrt{\kappa_{ub}} + 1} \right)^j \varepsilon(\mathbf{0}). \quad (6.13)$$

Fig. 3 plots the eigenvalues (6.8) for the choice of parameters $\eta_0 = h_0 = 1$ and two mesh sizes—a coarse ($n = 64$) and fine ($n = 256$) one. In both cases, the eigenvalues lie in the interval $[\kappa_{ub}^{-1}, 1]$, as expected by Theorem 1.

To test the robustness and performance of the preconditioner $\mathbf{A}$, we use PCG to solve (6.12) for increasingly challenging problems generated from large gradients in bathymetry (varying h_0) and large contrast ratio (varying η_0). As discussed in Section 4 large values of η_0, h_0 yield two independent mechanisms for increasing the conditioning number $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h})$.

Figs. 4 and 5 plot the error $\varepsilon(\mathbf{u}^{(i)})$ and upper bound (6.13) versus iteration i . When $\bar{b}$ is specified as problem data, we use the last PCG output as a reference solution for $\bar{u}^*$ to compute ε . Fig. 4 examines the effect of increasing η_0 holding h_0 fixed, while Fig. 5 varies h_0 holding η_0 fixed. While the number of PCG iterations increase with η_0, h_0 (which is expected), the figures also highlight that $\mathbf{A}$ remains an effective preconditioner over a wide range of η_0, h_0 , with little to no effect due to the mesh resolution n . Fig. 6 demonstrates how the number of PCG iterations, required to achieve a tolerance of $\varepsilon = 10^{-6}$, increases with problem difficulty. Lastly, it is worth noting that the PCG outperforms the (apriori) upper bound (3.7), and is partially explained in the subsequent remark.

Remark 9. The fact that the PCG convergence, e.g. Fig. 4, for test case (6.9) exhibits an “elbow” behavior with two characteristic rates may partially be understood via the (apostori) computation of the generalized eigenvalues, shown in Fig. 3. Fig. 3 provides

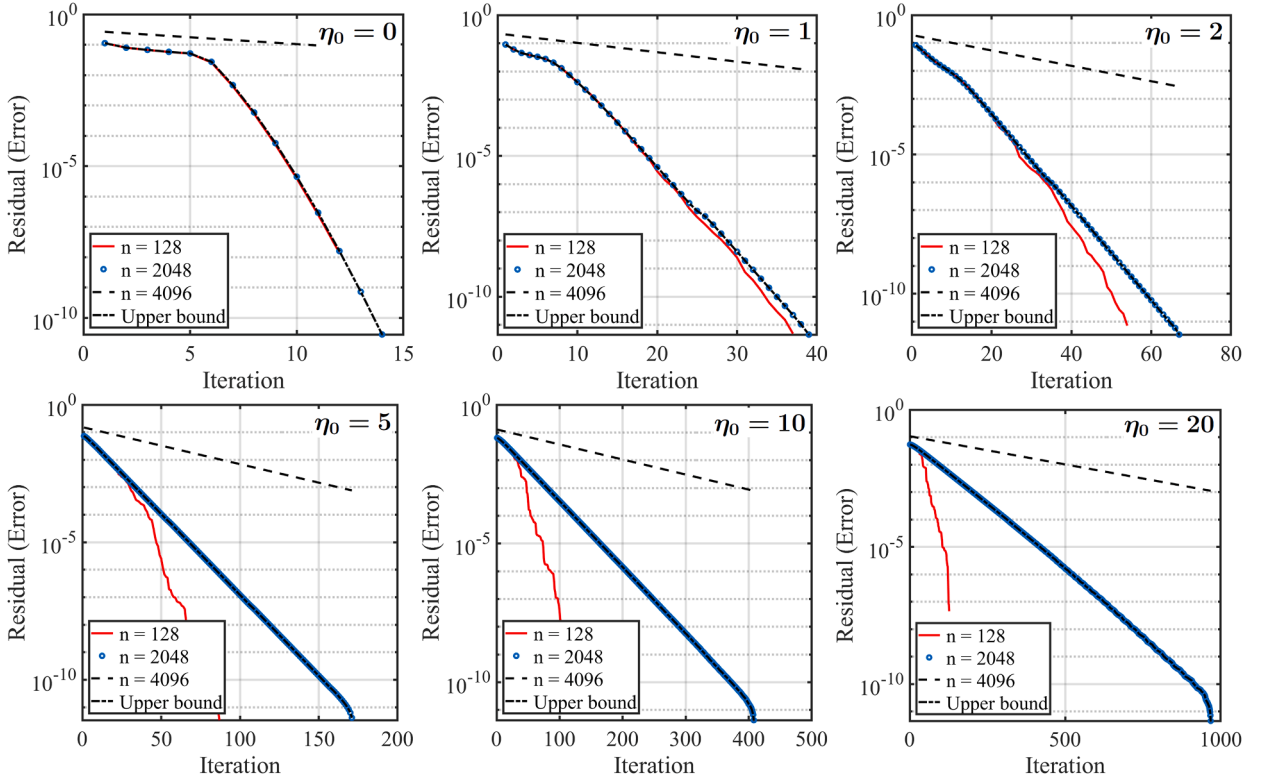


Fig. 4. PCG residual error ε (defined in (6.7)) versus iteration for the 1d problem (6.10). Subfigures show the performance with increasing water depth contrast ratios η_0 , holding $h_0 = 1$ fixed. The iterations increase with η_0 but remain bounded with increasing mesh resolution (n). The upper bound (dashed line) is (3.7) using κ_{ub} .

numerical evidence that the generalized eigenvalues of the operators $\mathcal{G}_{\eta,h}$ and $\mathcal{A}$ admit a continuous spectrum confined to an interval $\sim [3 \times 10^{-2}, 3 \times 10^{-1}]$, with ~ 6 isolated eigenvalues outside of this interval. An improved upper bound for the PCG error can then be used. After roughly 6 iterations (one for each isolated eigenvalue), the PCG error decays with a new effect upper bound (6.13) with $\kappa_{ub} \sim (3 \times 10^{-1})/(3 \times 10^{-2}) = 10$. This tighter upper bound more closely explains the convergence behavior in Figs. 4-5 that occurs after the initial phase. ♣

Fig. 7 examines the clock time required to solve (6.10) with $\eta_0 = h_0 = 1$ at different mesh resolutions ($2^5 \leq n \leq 2^{20}$). The figure demonstrates that clock time scales well with the expected floating point operations of the FFT ($\propto n \log_{10} n$).

Lastly, it is interesting to compare the optimal coefficient (4.3)-(4.4) with the coefficients in Remark 4, which are often easier to estimate for use in a large-scale computation. As preconditioners for conjugate gradient, the two sets of coefficients perform very similarly unless

$$\max_{x \in \Omega} \left(\eta(x) + \lambda_+ \eta(x) |h_x(x)|^2 \right) \ll \eta_{\max} (1 + \lambda_+ |h_x|_{\max}^2) \quad (6.14)$$

i.e., the value of $\eta(x)$ is small in locations where h_x is large. In settings where (6.14) hold, the coefficient σ^{**} in Remark 4 will be larger than σ^* and may lead to a drop in the preconditioner performance. The next test case specifically designs such an occurrence where (6.14) holds.

Set $h_0 = 1$ and $h(x)$ as in (6.10), and take a discontinuous water depth given by a square wave (the coefficients in the weak form of the operators $\mathcal{G}_{\eta,h}$ and $\mathcal{A}$, η need not be continuous)

$$\eta(x) = \begin{cases} 0.1 & \text{if } x \in S \\ 0.1\eta_0 & \text{if } x \notin S \end{cases} \quad \text{where} \quad S = \{x \in \Omega \mid |h_x(x)|^2 \leq 0.2\}.$$

Fig. 8 demonstrates the improved performance of the coefficients (4.4) (red) over the coefficients in Remark 4 (dashed black).

Lastly, Fig. 9 provides a comparison of the new preconditioner against a classic symmetric successive over-relaxation (SSOR) preconditioner

$$\mathbf{P} = (\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{L}^T) \quad \text{where} \quad \mathbf{G} = \mathbf{D} + \mathbf{L} + \mathbf{L}^T.$$

Here $\mathbf{D}$ is the diagonal of $\mathbf{G}$, while $\mathbf{L}^T$ and $\mathbf{L}$ are the strictly upper and lower entries of $\mathbf{G}$. As a result, linear systems involving $\mathbf{P}$ require the solution of two triangular systems. Not only does $\mathbf{A}$ outperform SSOR, it also enables the use of efficient algorithms for constant coefficient operators.

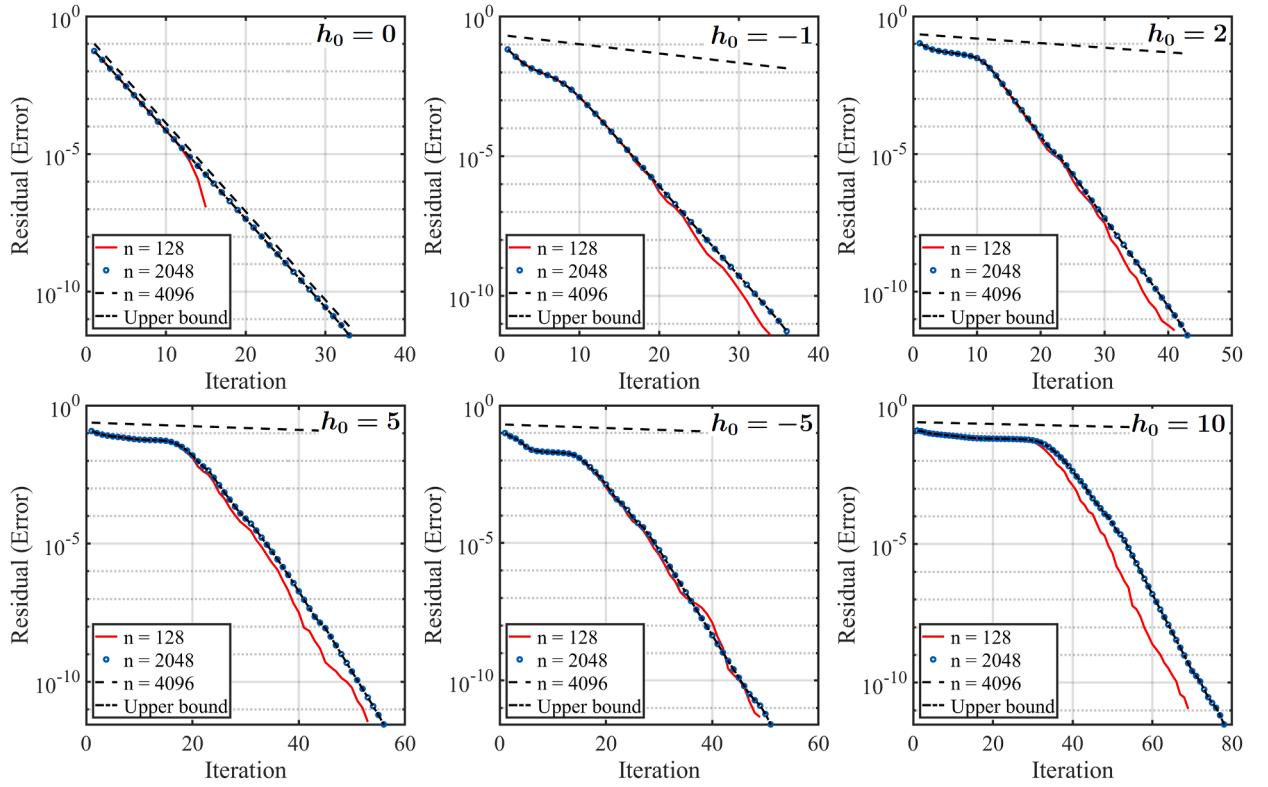


Fig. 5. PCG residual error ϵ (defined in (6.7)) versus iteration for the 1d problem (6.10). Subfigures show the performance with increasing bathymetry gradients h_0 (holding $\eta_0 = 1$ fixed). Again, the iterations increase with h_0 but remain bounded with increasing mesh resolution (n). The upper bound (dashed line) is (3.7) using κ_{ub} .

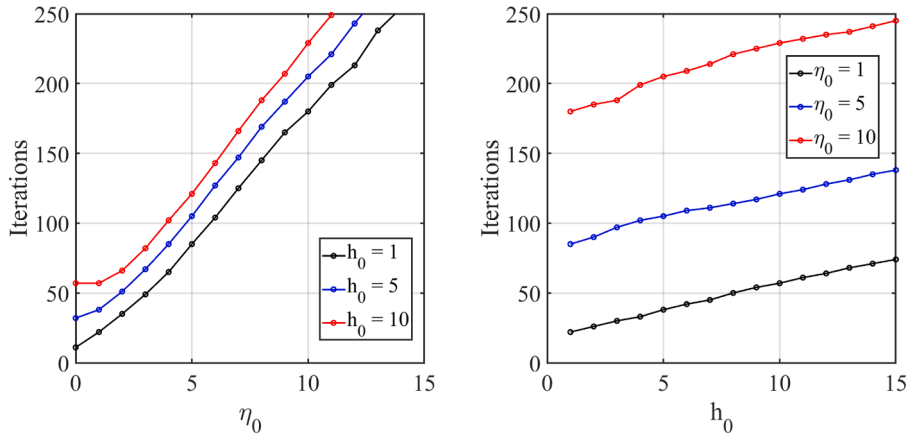


Fig. 6. Number of PCG iterations required for a residual error of $\epsilon < 10^{-6}$ (defined in (6.7)) versus η_0 (left) and h_0 (right) and mesh size $n = 512$. The iterations increase with the conditioning of the problem.

6.3. PCG tests in 2d: Curvature of the bathymetry

We now examine two dimensional bathymetry effects on the PCG performance. As a first test of our code, we verify every result in Section 6.2 by setting $v = 0$ and extending the single variable h, η, u to $d = 2$ as constants in the y -direction.

Next, to provide a (truly) two dimensional test case, we examine an elliptical bump bathymetry with varying aspect ratios

$$h(x, y) = \begin{cases} 1 - \frac{1}{2} \cos^2\left(\frac{\pi r}{2r_0}\right), & |r| \leq r_0, \\ 1, & |r| \geq r_0 \end{cases}, \quad \text{where } r = \sqrt{a^2 x^2 + b^2 y^2}, \quad r_0 = \frac{1}{2} \quad (6.15)$$

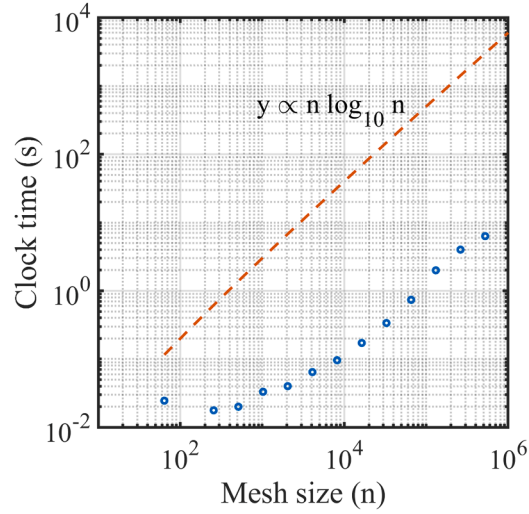


Fig. 7. PCG clock time (s; in seconds) required to solve for a residual of $\varepsilon < 10^{-12}$ versus mesh size n for the 1d test problem (6.10) with parameters $\eta_0 = h_0 = 1$. The reference line (dashed) presents the FLOP scaling for the FFT. The increase in computational time is due to the FFT scaling and not an increase in PCG iterations.

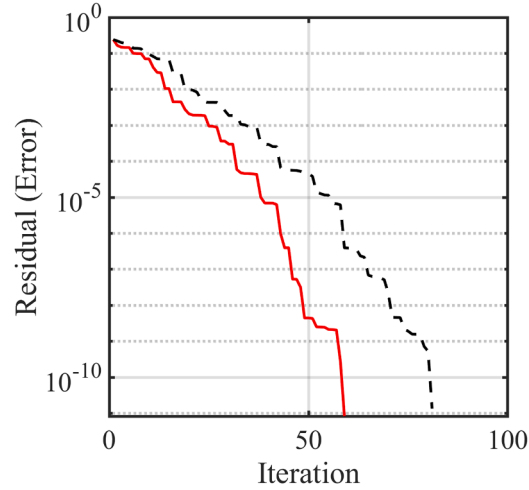


Fig. 8. Comparison of the PCG residual for the coefficients (4.4) (red; solid line) versus the simple (less optimal) coefficients in Remark 4 (black; dash line). To achieve a tolerance $\varepsilon < 10^{-8}$, requires only 49 iterations versus 71. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\eta(x, y) = h(x, y) + \exp(\cos(2\pi x)) + \frac{1}{4} \sin(4\pi y). \quad (6.16)$$

Note that both the magnitude and direction of ∇h vary throughout space to ensure the presence of anisotropic effects in the off-diagonal block entries of $\mathcal{G}_{\eta,h}$. The right hand side vector $\vec{b}$ is set to be the discretization of the velocity $\vec{U} = (\cos(4\pi x), \cos(4\pi y))$.

We test an increasing sequence of aspect ratios in the bathymetry that generate progressively larger gradients ∇h . Larger ∇h values increase the upper bound κ_{ub} , which can be viewed as a proxy for the conditioning of $A^{-1}G$. Fig. 10 plots the PCG convergence versus iteration for three elliptical aspect ratios. Again, the performance decreases with increasing ∇h , however the preconditioner remains effective independent of the mesh size.

6.4. Full solution of the SGN equations in 1d

Building on the studies of the operator $\mathcal{A}$ and $\mathcal{G}_{\eta,h}$ from the previous section, we provide here, numerical test cases and simulations of the full time-dependent SGN equations. We apply two different time integration strategies to solve the constraint form of the SGN equations as discussed in Section 5.

We first validate our codes in a flat bathymetry setting by solving the initial value problem with a solitary wave (6.1)–(6.2) (taking the domain large enough so that the exponential decay of the sech^2 profile reaches machine precision).

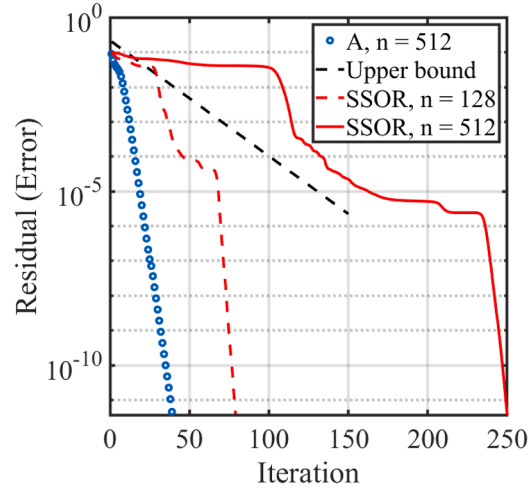


Fig. 9. Comparison of the new preconditioner (blue circles) against an SSOR preconditioner with conjugate gradient (red dashed for $n = 128$ and solid for $n = 512$), and for reference, the upper bound provided by κ_{ub} (black dashed). The test case uses $\eta_0 = 1$, $h_0 = 1$ in (6.9)-(6.10). Note that SSOR exhibits mesh dependence convergence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

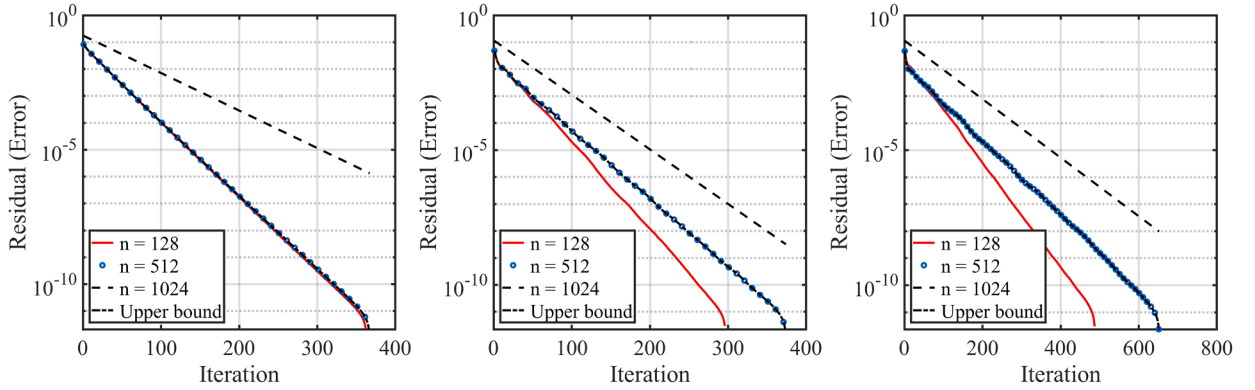


Fig. 10. Test case for a 2d bathymetry (6.15). Plots show the convergence of the PCG residual error ϵ defined in (6.7) with coefficients $(a, b) = (1, 1)$ (left), $(1, 20)$ (middle) and $(1, 40)$ (right). The upper bound (dashed line) is (3.7) using the upper bound κ_{ub} . Note the increase in iterations due to the effect of increasing aspect ratio and ∇h .

Test 1: Manufactured solution. To validate the time-stepping approaches and perform comparative studies in the presence of a variable bathymetry, we test the numerical approaches using the method of *manufactured solutions*. That is, the exact solution (on $L = 1$) is fixed to be

$$h(x) = 2 + \sin(2\pi x), \quad \eta^*(x, t) = 2 + \sin(2\pi x) \sin(10t), \quad u^*(x, t) = \cos(2\pi x) \cos(10t). \quad (6.17)$$

The SGN equations are then solved with an external forcing term chosen so that η^* , u^* solve the PDE.

The solution, initialized to η^* , u^* , is evolved to a final time $t_f = 1$. Since, η^* , u^* are periodic-in-time, the external forcing is also periodic-in-time and induces a Floquet instability in the equations, which we confirmed by computing the spectral radius of the Floquet monodromy matrix. Evolving the SGN equations for a longer time with this external forcing allows the instability to grow in time. The final time $t_f = 1$ in our tests is specifically chosen small enough to test numerical convergence and computational efficiency via clock time while avoiding unnecessary complications from instabilities that could manifest in the equations over longer times.

All numerical computations are run using a fixed mesh with $n = 256$ grid points, which is small enough to ensure spatial discretization errors remain well below temporal errors, i.e., are close to machine precision. For explicit time-stepping, the PCG tolerance is set to 10^{-14} . Fig. 11 provides a convergence plot in L^∞ for η (left), u (middle) demonstrating that the methods converge and do so at the expected rates. The CFL number which dictates the largest values of $\Delta t = C\Delta x$ ($\Delta x = 1/n$) are method dependent, and range from $C = .2$ for RK4 to $C = .05$ for SBDF2. Fig. 11 (right) also provides a clock time comparison. As expected, the high order methods outperform lower order methods, while the linearly implicit SBDF2 outperforms AB2 with PCG.

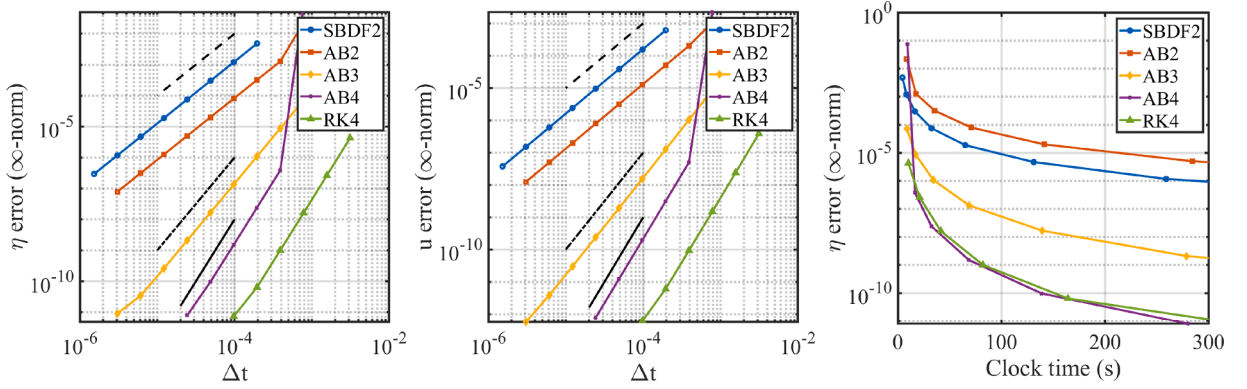


Fig. 11. Figures provide an L^∞ convergence study for η (left), and u (middle) as well as a clock time comparison (right) for the Section 6.4 Test Case 1. The figures compare the linearly implicit SBDF2 approach versus explicit schemes with PCG (AB2, AB3, AB4 and RK4). All methods converge at the expected rates, and higher order methods provide greater accuracy for a given clock time; the linearly implicit SBDF2 outperforms AB2 due to the single linear solve involving $\mathcal{A}$.

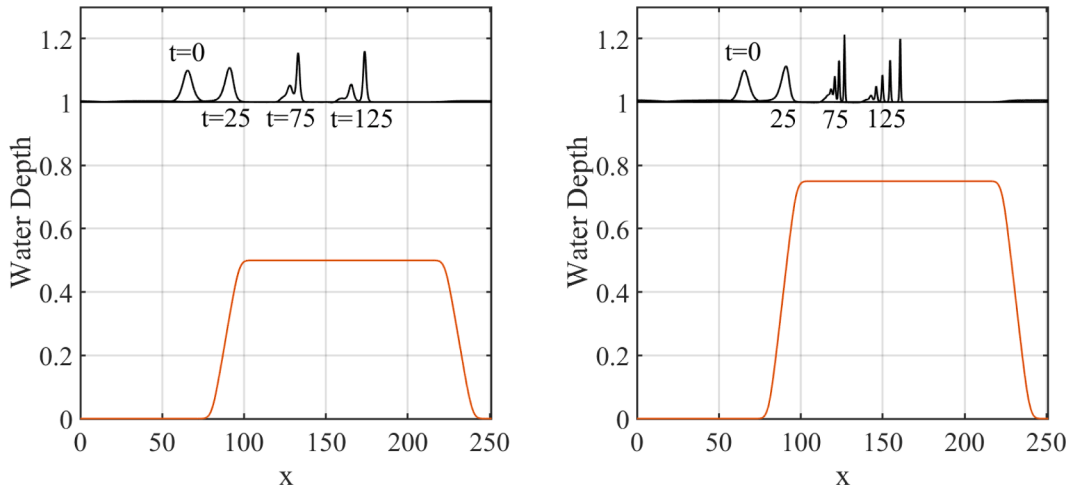


Fig. 12. Solitary wave traveling to the right over an underwater shelf (orange; solid line) with height 0.5 (left) and 0.75 (right). Figs. 13 and 14 show enlarged images of the waves.

Test 2: Simulation of a solitary wave approaching an underwater shelf. In this test, we simulate a solitary wave traveling over an underwater shelf (here $L = 80\pi$). The initial data is set to the traveling solitary wave solution (6.1)–(6.2) with the water depth h_0 set to the height preceding the shelf. The exact form of the shelf is a piecewise linear square wave convolved with a smooth bump function. Fig. 12 visualizes the effect of dispersion for two shelf heights. Figs. 13–14 provide enlarged views of the waves for times $t = 0, 25, 75, 125$, which also shows the presence of a small reflected wave.

Aliasing can present a source of error when spectral methods are used to solve nonlinear problems. In all of our simulations, we have not made use of dealiasing. We found that regardless of the choice of time integration, solitary wave solutions with larger amplitudes can destabilize over long times scales. We also found that dealiasing through the introduction of padding the high frequency wave numbers with additional zeros, can aid in avoiding the instability.

6.5. Full solution of the SGN equations in 2d

This test problem solves the SGN equations on the doubly periodic square with length $L = 1$. To test the codes, we first performed a convergence study on the time integration schemes in two dimensions using a solitary wave solution. The results were analogous to the one dimensional setting.

For the two dimensional test, we set a variable bathymetry to be $h(x, y) = h_0 - b(x, y)$ where $h_0 = 1.5a$ and $b(x, y)$ is the periodic extension of

$$b(x, y) = 0.75a \exp(-r(x, y)^2 / (0.2)^2) \quad r(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}. \quad (6.18)$$

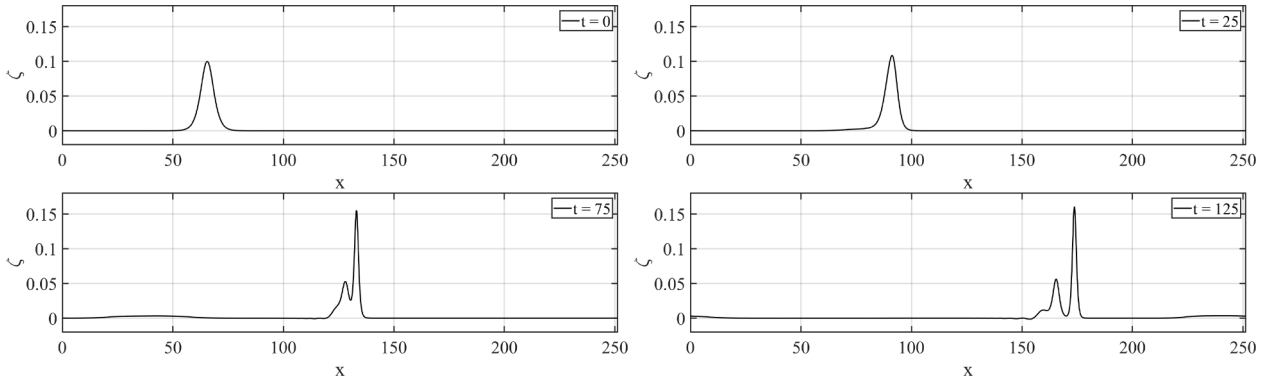


Fig. 13. Enlargements of the solutions plotted in Fig. 12 (left). Note the reflected wave.

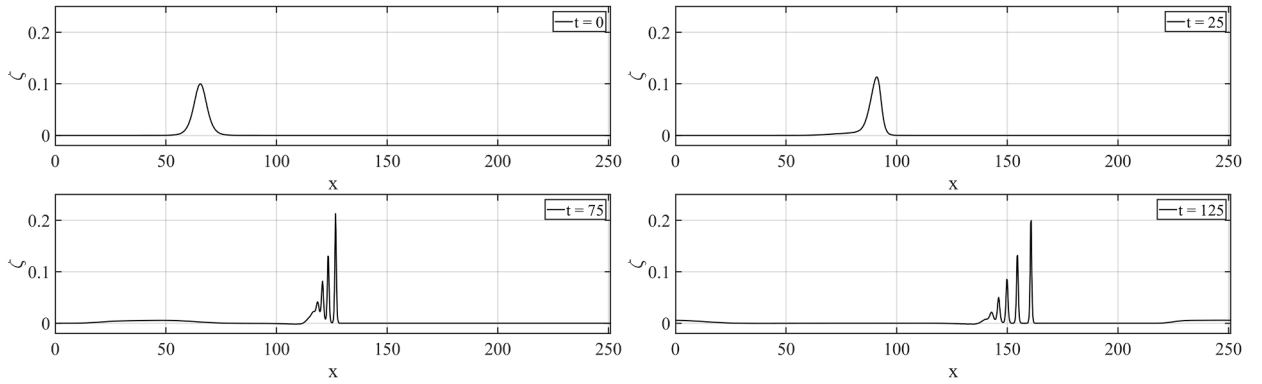


Fig. 14. Enlargements of the solutions plotted in Fig. 12 (right). Note the reflected wave.

The surface η and velocity $\vec{u}$ are initialized to be a traveling solitary wave front with an angle of 45 degrees and $a = 1/100$. If the bathymetry were constant, the solution would remain a solitary wave, however the profile loses its shape upon traveling over the variable $h(x, y)$.

Fig. 15 shows the wave height solution $\zeta(x, y, t)$ at various times using AB2 with PCG. A 255×256 grid and time step $\Delta t = \frac{1}{3}\Delta x$ were used in the computation. Other time integration schemes produce visibly similar results on the scale presented, but yield a greater accuracy.

The semi-implicit SBDF2 approach makes use of a single set of coefficients σ^*, α^* over the entire computation. To ensure that at each time the generalized eigenvalues of $\mathcal{A}$ and $\mathcal{G}_{\eta, h}$ lie in the stability region discussed in Section 5.4, we use the values of σ^* and α^* given by the initial data. These values are close to the largest values provided by the formulas σ^*, α^* over the duration of the simulation. In general, if $\eta^*(x, t)$ is the exact water depth solution of the SGN over the solution time $0 \leq t \leq T$, then the coefficients σ^* and α^* in $\mathcal{A}$ should be taken as the largest values, maximized over time, provided by (4.3)–(4.4).

7. Discussion and conclusions

In this work we have introduced a constant coefficient operator $\mathcal{A}$ as a means to devise time integration approaches for the SGN equations that require only matrix-vector products involving $\mathcal{G}_{\eta, h}$, i.e., are *matrix-free*. The coefficients are rooted in theory, are shown to be quasi-optimal (within a class of constant coefficient operators) and enable mesh-independent performance of preconditioned conjugate gradient methods. Furthermore, the operator $\mathcal{A}$ enables the use of linearly implicit time integration that reduces the total number of linear solves involving $\mathcal{A}$ to one per time step. The computational theory is further supported by a suite of test cases.

Solving linear systems involving the constant coefficient operator $\mathcal{A}$ is advantageous over direct solves involving $\mathcal{G}_{\eta, h}$ since the latter is variable coefficient in both space and time. The formulas for $\mathcal{A}$ and bounds provided here may also provide avenues for other efficient computational approaches. These potentially include successive convolution where the analytic and fast application of the Green's function for $\mathcal{A}$ can be applied; exponential integrators since $\exp(\mathcal{A})$ may be readily evaluated using the Fourier transform, or boundary integral methods. Such approaches could be particularly useful in the context of large scale computations (e.g., modeling the ocean) as the Green's function for $\mathcal{A}$ can be analytically approximated and locally applied thereby avoiding the need to solve large linear systems. This would help to enable the use of physical models that incorporate dispersion for the simulation of waves in oceanography. A constant coefficient preconditioner was also previously tested in [50] for a regularized version of the Miyata-Choi-

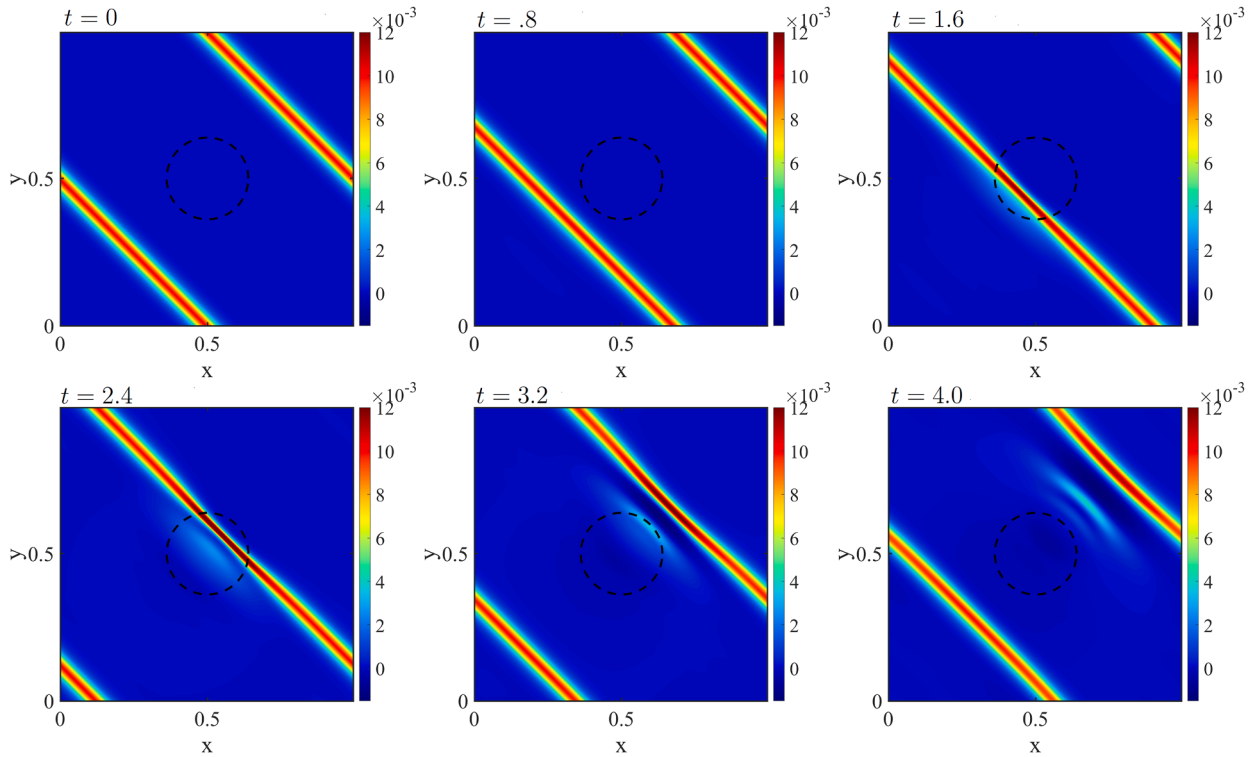


Fig. 15. Contour plots of ζ showing a solitary wave passing over an underwater circular 2d barrier given by (6.18). The dashed line shows the level surface $b(x, y) = \frac{1}{2}0.75a$. Snapshots (top-bottom, left-right) correspond to $t = 0, .8, 1.6, 2.4, 3.2, 4.0$.

Camassa (MCC) model that characterizes internal waves in a two-layer system [51–53]. As the MCC model is a system of coupled SGN equations through the interfacial pressure, it is expected that the results in this work can also be extended to the MCC model.

CRediT authorship contribution statement

Linwan Feng: Formal analysis, Software, Writing - original draft; **David Shirokoff:** Conceptualization, Formal analysis, Software, Writing - original draft; **Wooyoung Choi:** Conceptualization, Writing - original draft.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: David Shirokoff reports financial support was provided by National Science Foundation. Wooyoung Choi reports financial support was provided by National Science Foundation. Linwan Feng reports financial support was provided by National Science Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Benjamin Seibold and Donna Calhoun for helpful conversations. DS would like to gratefully acknowledge the Vietnam Institute for Advanced Study in Mathematics (VIASM) for hosting a visit in June of 2025. This material is based upon work supported by the National Science Foundation under Grants No. DMS-1719693 (LF, DS); DMS-2012268, DMS-2309727 (DS); DMS-2108524 (WC). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Appendix A. Proof of Theorem 1

This subsection provides a theoretical justification for the coefficients (4.3)–(4.5) by proving the convergence bound (4.6). In the proof it will be useful to write both $A[\cdot, \cdot]$ and $B[\cdot, \cdot]$ into block matrix form as

$$A[\vec{u}, \vec{v}] = \int_{\Omega} (\vec{u}^T \nabla \cdot \vec{u}) \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix} \begin{pmatrix} \vec{v} \\ \nabla \cdot \vec{v} \end{pmatrix} d\vec{x}, \quad (\text{A.1})$$

$$B[\vec{u}, \vec{v}] = \int_{\Omega} (\vec{u}^T \nabla \cdot \vec{u}) M(\vec{x}) \begin{pmatrix} \vec{v} \\ \nabla \cdot \vec{v} \end{pmatrix} d\vec{x}, \quad (\text{A.2})$$

where $M(\vec{x})$ is the variable coefficient matrix

$$M(\vec{x}) = \eta(\vec{x}) \begin{pmatrix} I + \nabla h(\vec{x}) \nabla h^T(\vec{x}) & \frac{1}{2} \eta(\vec{x}) \nabla h(\vec{x}) \\ \frac{1}{2} \eta(\vec{x}) \nabla h(\vec{x})^T & \frac{1}{3} \eta^2(\vec{x}) \end{pmatrix}. \quad (\text{A.3})$$

The following is a proof of the main result.

Proof. [Of Theorem 1] Step 1: We first show that M and the coefficients (4.3)–(4.4) satisfy the following matrix inequality

$$(\text{MI}) \quad (\kappa_{\text{ub}})^{-1} \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix} \leq M(\vec{x}) \leq \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix} \quad \text{holds for } \vec{x} \in \Omega.$$

Here I is the 2×2 identity matrix, $M(\vec{x})$ is defined in (A.3), and for square symmetric matrices $X \leq Y$ means $Y - X$ is positive semidefinite.

When $|\nabla h|_{\max} = 0$, corresponding to a flat bathymetry, the matrix M is diagonal so that

$$\begin{pmatrix} \eta_{\min} I & 0 \\ 0 & \frac{1}{3} \eta_{\min}^3 \end{pmatrix} \leq \begin{pmatrix} \eta(\vec{x}) I & 0 \\ 0 & \frac{1}{3} \eta^3(\vec{x}) \end{pmatrix} \leq \begin{pmatrix} \eta_{\max} I & 0 \\ 0 & \frac{1}{3} \eta_{\max}^3 \end{pmatrix} = \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix}. \quad (\text{A.4})$$

Combining (A.4) with (4.3) then yields (MI); the lower bound on M follows since $\eta_{\max} \geq \eta_{\min} > 0$.

Assume now that $|\nabla h|_{\max} \neq 0$. Let $R(\vec{x})$ be the 2×2 Householder matrix that reflects $\nabla h(\vec{x})$ to align with the second coordinate axis (if $\nabla h(\vec{x}) = 0$, take $R = I$), i.e.,

$$W(\vec{x}) = \begin{pmatrix} R(\vec{x})^T & 0 \\ 0 & 1 \end{pmatrix} \quad R(\vec{x}) \nabla h(\vec{x}) = \begin{pmatrix} 0 \\ |\nabla h(\vec{x})| \end{pmatrix} \quad R(\vec{x})^T R(\vec{x}) = I.$$

With this $W(\vec{x})$

$$W(\vec{x})^T M(\vec{x}) W(\vec{x}) = \begin{pmatrix} \eta(\vec{x}) & 0 \\ 0 & \widetilde{M}(\vec{x}) \end{pmatrix}, \quad (\text{A.5})$$

where

$$\widetilde{M}(\vec{x}) = \begin{pmatrix} \eta(\vec{x}) & 0 \\ 0 & 0 \end{pmatrix} + \eta(\vec{x}) \begin{pmatrix} |\nabla h(\vec{x})| & 0 \\ 0 & \eta(\vec{x}) \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} |\nabla h(\vec{x})| & 0 \\ 0 & \eta(\vec{x}) \end{pmatrix}. \quad (\text{A.6})$$

Moreover, from basic properties of semidefinite matrices, (MI) is equivalent to

$$(\text{MI-R}) \quad (\kappa_{\text{ub}})^{-1} \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix} \leq W(\vec{x})^T M(\vec{x}) W(\vec{x}) \leq \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix} \quad \text{for } \vec{x} \in \Omega.$$

The equivalence of (MI) and (MI-R) holds since (i) $A \leq B \Leftrightarrow Q^T A Q \leq Q^T B Q$ for any invertible Q and pair of symmetric matrices A, B ; and (ii) the diagonal matrix $\text{diag}(\sigma^*, \sigma^*, \alpha^*)$ is invariant when multiplied by $W(\vec{x})^T$ and $W(\vec{x})$. Thus, it is sufficient to show that (MI-R) holds.

The matrix of numbers in $\widetilde{M}$ has eigenvalues $\lambda_{\pm}$, i.e.,

$$\lambda_{-} I \leq \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix} \leq \lambda_{+} I \quad \text{where} \quad \lambda_{\pm} = \frac{1}{6} (4 \pm \sqrt{13}) > 0. \quad (\text{A.7})$$

Combining (A.7) with the expression for $\widetilde{M}$ yields the bounds

$$\begin{pmatrix} \eta(\vec{x}) + \lambda_{-} \eta(\vec{x}) |\nabla h(\vec{x})|^2 & 0 \\ 0 & \lambda_{-} \eta^3(\vec{x}) \end{pmatrix} \leq \widetilde{M}(\vec{x}) \leq \begin{pmatrix} \eta(\vec{x}) + \lambda_{+} \eta(\vec{x}) |\nabla h(\vec{x})|^2 & 0 \\ 0 & \lambda_{+} \eta^3(\vec{x}) \end{pmatrix}. \quad (\text{A.8})$$

Inserting (A.8) into the expression for $M(\vec{x})$ in (A.5) and maximizing over $\vec{x}$ yields an upper bound

$$W(\vec{x})^T M(\vec{x}) W(\vec{x}) \leq \begin{pmatrix} \eta_{\max} & \\ & \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \lambda_{+} \eta(\vec{x}) |\nabla h(\vec{x})|^2) \\ & & \lambda_{+} \eta_{\max}^3 \end{pmatrix} \\ \leq \text{diag}(\sigma^*, \sigma^*, \alpha^*).$$

The second line follows since $\lambda_+ \eta(\vec{x}) |\nabla h(\vec{x})|^2 \geq 0$ implies $\eta_{\max} \leq \sigma^*$. Therefore $M(\vec{x})$ satisfies the right hand side of (MI-R) and thus (MI).

In a similar spirit, inserting (A.8) into the expression for $M(\vec{x})$ and minimizing over $\vec{x}$ yields a lower bound

$$\begin{aligned} W(\vec{x})^T M(\vec{x}) W(\vec{x}) &\geq \begin{pmatrix} \eta_{\min} & \\ & \min_{\vec{x} \in \Omega} (\eta(\vec{x}) + \lambda_- \eta(\vec{x}) |\nabla h(\vec{x})|^2) \\ & & \lambda_- \eta_{\min}^3 \end{pmatrix} \\ &\geq \text{diag}(\eta_{\min}, \eta_{\min}, \lambda_- \eta_{\min}^3), \\ &\geq (\kappa_{\text{ub}})^{-1} \text{diag}(\sigma^*, \sigma^*, \alpha^*). \end{aligned}$$

In the last line we have used the fact that

$$(\kappa_{\text{ub}})^{-1} = \min \left\{ \frac{\eta_{\min}}{\sigma^*}, \frac{\lambda_-}{\lambda_+} \left(\frac{\eta_{\min}}{\eta_{\max}} \right)^3 \right\},$$

which simultaneously implies $(\kappa_{\text{ub}})^{-1} \sigma^* \leq \eta_{\min}$ and $(\kappa_{\text{ub}})^{-1} \alpha^* \leq \lambda_- \eta_{\min}^3$. Therefore, $M(\vec{x})$ satisfies the lower bound in (MI) as well.

Step 2: Next, from (MI) the bilinear forms $A[\cdot, \cdot]$, $B[\cdot, \cdot]$ satisfy the inequality

$$(\kappa_{\text{ub}})^{-1} A[\vec{u}, \vec{u}] \leq B[\vec{u}, \vec{u}] \leq A[\vec{u}, \vec{u}] \quad \forall \vec{u} \in H_{\text{div}}(\Omega). \quad (\text{A.9})$$

For if $\vec{u} \in H_{\text{div}}(\Omega)$ then $(\vec{u}^T, \nabla \cdot \vec{u})^T \in (L^2(\Omega))^3$. We can then contract both sides of the left inequality in (MI) with the vector $(\vec{u}^T, \nabla \cdot \vec{u})^T$ to yield

$$(\vec{u}^T, \nabla \cdot \vec{u}) M(\vec{x}) \begin{pmatrix} \vec{u} \\ \nabla \cdot \vec{u} \end{pmatrix} \leq (\vec{u}^T, \nabla \cdot \vec{u}) \begin{pmatrix} \sigma^* I & 0 \\ 0 & \alpha^* \end{pmatrix} \begin{pmatrix} \vec{u} \\ \nabla \cdot \vec{u} \end{pmatrix}, \quad (\text{A.10})$$

which holds for all $\vec{u} \in H_{\text{div}}(\Omega)$ and (almost every) $\vec{x} \in \Omega$. Integrating (A.10) over Ω and combining with (A.1)-(A.2) yields $B[\vec{u}, \vec{u}] \leq A[\vec{u}, \vec{u}]$. The lower bound in (A.9) follows from an identical argument.

Finally, (A.9) proves the desired result since

- The conditioning number satisfies $\kappa_{\mathcal{A}}(\mathcal{G}_{\eta,h}) \leq \kappa_{\text{ub}}$. This follows immediately from dividing (A.9) by $A[\vec{u}, \vec{u}]$ and applying the definitions in (3.10).
- Every eigenvalue λ of $\mathcal{A}^{-1} \mathcal{G}_{\eta,h}$ (equivalently generalized eigenvalue of (3.4)) is real and satisfies $(\kappa_{\text{ub}})^{-1} \leq \lambda \leq 1$. Set $\vec{v} = \vec{u}$ in (3.6), where $(\vec{u}, \lambda)$ is a generalized eigenvector and eigenvalue, and substitute the resulting $B[\vec{u}, \vec{u}] = \lambda A[\vec{u}, \vec{u}]$ into (A.9) to obtain:

$$(\kappa_{\text{ub}})^{-1} A[\vec{u}, \vec{u}] \leq \lambda A[\vec{u}, \vec{u}] \leq A[\vec{u}, \vec{u}]. \quad (\text{A.11})$$

Dividing by $A[\vec{u}, \vec{u}] \neq 0$ yields the result.

□

Appendix B. Quasi-optimality of the conditioning number bound

The proof of the main result relies on $(\sigma^*, \alpha^*, \kappa_{\text{ub}})$ satisfying (MI). One might then wonder if there is a set of coefficients satisfying (MI) that yield a (fundamentally) tighter conditioning number bound on $\mathcal{A}^{-1} \mathcal{G}_{\eta,h}$? To answer this question, consider the optimal conditioning number bound obtained by coefficients satisfying (MI)

$$\begin{aligned} (P) \quad &\text{Minimize: } \gamma \\ &\text{Subject to: } M(\vec{x}) \leq \begin{pmatrix} \sigma I & 0 \\ 0 & \alpha \end{pmatrix} \leq \gamma M(\vec{x}) \quad \text{holds for } \vec{x} \in \Omega, \end{aligned}$$

Note that (P) is a convex problem since the objective function γ is linear and the constraints on σ and α are an intersection of a family of linear matrix inequalities. Denote the solution to (P) as γ_{P}^* .

The answer turns out to be that no other set of coefficients satisfying (MI) can avoid the scalings of ∇h and η in κ_{ub} .

Theorem 3. For any functions $\eta, \nabla h \in L^\infty(\Omega)$ with $\eta_{\min} > 0$, the value of κ_{ub} defined in (4.3)–(4.5) satisfies

$$\gamma_{\text{P}}^* \leq \kappa_{\text{ub}} \leq \left(\frac{4+\sqrt{13}}{4-\sqrt{13}} \right) \gamma_{\text{P}}^*. \quad (\text{B.1})$$

In the case of a flat bathymetry ($\nabla h = 0$) then $\kappa_{\text{ub}} = \gamma_{\text{P}}^*$.

The inequality (B.1) shows that κ_{ub} is always within a constant factor of γ_{P}^* . Thus the fundamental scaling of η and ∇h in κ_{ub} cannot be avoided via other coefficients satisfying (MI).

To establish the result, consider a fixed pair η and ∇h and introduce the feasible constraint set in (P) as

$$\mathcal{S} \equiv \left\{ (\sigma, \alpha, \gamma) \in \mathbb{R}^3 \mid M(\vec{x}) \leq \begin{pmatrix} \sigma I & 0 \\ 0 & \alpha \end{pmatrix} \leq \gamma M(\vec{x}) \right\}, \quad (\text{B.2})$$

$$= \left\{ (\sigma, \alpha, \gamma) \in \mathbb{R}^3 \mid \eta(\vec{x}) \leq \sigma \leq \gamma \eta(\vec{x}), \quad \widetilde{M}(\vec{x}) \leq \begin{pmatrix} \sigma & 0 \\ 0 & \alpha \end{pmatrix} \leq \gamma \widetilde{M}(\vec{x}) \quad \forall \vec{x} \in \Omega \right\}. \quad (\text{B.3})$$

Here $\widetilde{M}(\vec{x})$ is defined in (A.6) and the second line follows from the same Householder reflection in Appendix A. The set S is also non-empty since $\eta_{\max}, |\nabla h|_{\max}$ are assumed to be finite.

Case 1: A flat bathymetry ($\nabla h = 0$). When $\nabla h = 0$ the coefficients $(\sigma^*, \alpha^*, \kappa_{\text{ub}})$ solve (P).

In this case the matrix inequality in S reduces to two linear inequalities defined by the diagonals of $M(\vec{x})$,

$$S = \left\{ (\sigma, \alpha, \gamma) \in \mathbb{R}^3 \mid \eta(\vec{x}) \leq \sigma \leq \gamma \eta(\vec{x}) \quad \text{and} \quad \frac{1}{3} \eta^3(\vec{x}) \leq \alpha \leq \gamma \frac{1}{3} \eta^3(\vec{x}) \quad \forall \vec{x} \in \Omega \right\}, \\ = \left\{ (\sigma, \alpha, \gamma) \in \mathbb{R}^3 \mid \eta_{\max} \leq \sigma \leq \gamma \eta_{\min} \quad \text{and} \quad \frac{1}{3} \eta_{\max}^3 \leq \alpha \leq \gamma \frac{1}{3} \eta_{\min}^3 \right\}. \quad (\text{B.4})$$

The second line follows from taking the (essential) supremum and infimum over $\vec{x}$. Thus, S is polyhedral—it consists of two pairs of linear inequalities.

The coefficients $(\sigma^*, \alpha^*, \kappa_{\text{ub}})$ for $\mathcal{A}$ then exactly solve (P), since

$$\sigma^* = \eta_{\max}, \quad \alpha^* = \frac{1}{3} \eta_{\max}^3, \quad \text{and} \quad \gamma_{\text{P}}^* = \max \left\{ \frac{\eta_{\max}}{\eta_{\min}}, \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3 \right\} = \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3 = \kappa_{\text{ub}}.$$

While the choice of α^* is unique, any value $\sigma \in [\eta_{\max}, \kappa_{\text{ub}} \eta_{\min}]$ yields the optimal κ_{ub} value.

Case 2: A variable bathymetry ($\nabla h \neq 0$). Here the set S is no longer polyhedral, and the coefficients $(\sigma^*, \alpha^*, \kappa_{\text{ub}})$ for $\mathcal{A}$ will be quasi-optimal solutions of (P).

We first bound S from the inside and outside with polyhedral sets.

Proposition B.1 (Polyhedral bounds on S). *For a fixed $\eta(\vec{x})$ and $\nabla h(\vec{x})$, we have:*

(a) (Outer polyhedral bound of S) Any $(\sigma, \alpha, \gamma) \in S$ necessarily satisfies

$$\left. \begin{aligned} \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2) &\leq \sigma \leq \gamma \eta_{\min} \\ \frac{1}{3} \eta_{\max}^3 &\leq \alpha \leq \gamma \frac{1}{3} \eta_{\min}^3 \end{aligned} \right\}. \quad (\text{B.5})$$

(b) (Inner polyhedral bound of S) If (σ, α, γ) satisfies

$$\left. \begin{aligned} \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \lambda_+ \eta(\vec{x}) |\nabla h(\vec{x})|^2) &\leq \sigma \leq \gamma \eta_{\min} \\ \lambda_+ \eta_{\max}^3 &\leq \alpha \leq \gamma \lambda_- \eta_{\min}^3 \end{aligned} \right\}, \quad (\text{B.6})$$

then $(\sigma, \alpha, \gamma) \in S$. Here $\lambda_{\pm}$ are eigenvalues defined in (A.7).

The main idea of the proof is to bound $\widetilde{M}(\vec{x})$ in (A.6) with diagonal matrices.

Proof. (Of Proposition B.1) (a) Let $(\sigma, \alpha, \gamma) \in S$. Using the fact that semidefinite matrices have non-negative diagonal entries, comparing the diagonals in the matrix inequality for $\widetilde{M}(\vec{x})$ in (A.6), shows that (σ, α, γ) satisfy:

$$\left. \begin{aligned} \eta(\vec{x}) &\leq \sigma \leq \gamma \eta(\vec{x}) \\ \eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2 &\leq \sigma \leq \gamma (\eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2) \\ \frac{1}{3} \eta^3(\vec{x}) &\leq \alpha \leq \gamma \frac{1}{3} \eta^3(\vec{x}) \end{aligned} \right\} \quad \forall \vec{x} \in \Omega, \quad (\text{B.7})$$

Note that the first inequality in (B.7) imposes a tighter upper bound on σ while the second inequality in (B.7) imposes a tighter lower bound on σ . Maximizing and minimizing (B.7) over $\vec{x}$, and combining the two σ inequalities together yields (B.5).

(b) Substituting the matrix inequality (A.7) into the definition of $\widetilde{M}(\vec{x})$ shows that $\widetilde{M}(\vec{x})$ (for almost every $\vec{x}$) satisfies the following bound

$$\begin{pmatrix} \eta(\vec{x}) + \lambda_- \eta(\vec{x}) |\nabla h(\vec{x})|^2 & 0 \\ 0 & \lambda_- \eta^3(\vec{x}) \end{pmatrix} \leq \widetilde{M}(\vec{x}) \leq \begin{pmatrix} \eta(\vec{x}) + \lambda_+ \eta(\vec{x}) |\nabla h(\vec{x})|^2 & 0 \\ 0 & \lambda_+ \eta^3(\vec{x}) \end{pmatrix}.$$

Thus, the matrix linear inequality in S for $\widetilde{M}(\vec{x})$ can be ensured if (σ, α, γ) satisfy

$$\begin{pmatrix} \sigma & 0 \\ 0 & \alpha \end{pmatrix} \leq \gamma \begin{pmatrix} \eta(\vec{x}) + \lambda_- \eta(\vec{x}) |\nabla h(\vec{x})|^2 & 0 \\ 0 & \lambda_- \eta^3(\vec{x}) \end{pmatrix} \quad \forall \vec{x} \in \Omega,$$

and

$$\begin{pmatrix} \eta(\vec{x}) + \lambda_+ \eta(\vec{x}) |\nabla h(\vec{x})|^2 & 0 \\ 0 & \lambda_+ \eta^3(\vec{x}) \end{pmatrix} \leq \begin{pmatrix} \sigma & 0 \\ 0 & \alpha \end{pmatrix} \quad \forall \vec{x} \in \Omega.$$

We thus observe that if (σ, α, γ) satisfy

$$\left. \begin{aligned} \eta(\vec{x}) &\leq \sigma \leq \gamma \eta(\vec{x}) \\ \eta(\vec{x}) + \lambda_+ \eta(\vec{x}) |\nabla h(\vec{x})|^2 &\leq \sigma \leq \gamma (\eta(\vec{x}) + \lambda_- \eta(\vec{x}) |\nabla h(\vec{x})|^2) \\ \lambda_+ \eta^3(\vec{x}) &\leq \alpha \leq \gamma \lambda_- \eta^3(\vec{x}) \end{aligned} \right\} \quad \forall \vec{x} \in \Omega, \quad (\text{B.8})$$

then $(\sigma, \alpha, \gamma) \in S$. Just as with, (B.7), the first inequality in (B.8) imposes a tighter upper bound on σ while the second imposes a tighter lower bound. Maximizing and minimizing (B.8) over $\vec{x}$, and combining the two σ inequalities together yields (B.6). $\square$

The coefficients for $\mathcal{A}$ are in fact not only feasible, but yield the tightest conditioning bounds provided by the inner approximation

$$\sigma^* \equiv \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \lambda_+ \eta(\vec{x}) |\nabla h(\vec{x})|^2), \quad \alpha^* \equiv \lambda_+ \eta_{\max}^3, \quad (\text{B.9})$$

along with

$$\kappa_{\text{ub}} \equiv \max \left\{ \frac{\sigma^*}{\eta_{\min}}, \frac{\lambda_+}{\lambda_-} \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3 \right\}. \quad (\text{B.10})$$

Proof. [Of Theorem 3] The lower bound in (B.1) follows immediately since $(\sigma^*, \alpha^*, \kappa_{\text{ub}})$ is a feasible point of (P) and γ_p^* is the global optimum of (P).

The upper bound in (B.1) is a consequence of the polyhedral bound (B.5) in Proposition B.1(a). Namely, every set of points in S , including the optimal value γ_p^* satisfies

$$\gamma_p^* \geq \Gamma \equiv \max \left\{ \frac{1}{\eta_{\min}} \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2), \left(\frac{\eta_{\max}}{\eta_{\min}} \right)^3 \right\}, \quad (\text{B.11})$$

Here Γ is the larger of the two lower bounds on γ provided by (B.5).

Next, note that we can bound κ_{ub} from above by a constant factor of Γ . Since $\lambda_+ > 1 > \lambda_- > 0$ and η is non-negative, we have

$$\sigma^* \leq \lambda_+ \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2) \leq \frac{\lambda_+}{\lambda_-} \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2).$$

Hence,

$$\begin{aligned} \kappa_{\text{ub}} &\leq \max \left\{ \frac{\lambda_+}{\lambda_- \eta_{\min}} \max_{\vec{x} \in \Omega} (\eta(\vec{x}) + \eta(\vec{x}) |\nabla h(\vec{x})|^2), \frac{\lambda_+}{\lambda_-} \frac{\eta_{\max}^3}{\eta_{\min}^3} \right\} \\ &= \frac{\lambda_+}{\lambda_-} \Gamma. \end{aligned}$$

Combining this with (B.11) yields the upper bound in (B.1). $\square$

Remark 10 (Alternatives to $\lambda_{\pm}$). The choice of eigenvalues $\lambda_{\pm}$ for upper and lower bounds in (A.7) is not unique. In fact, the following diagonal matrix bound

$$\frac{1}{2(2 + \sqrt{3})} \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \leq \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} \end{pmatrix} \leq \frac{1}{2}(2 + \sqrt{3}) \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \quad (\text{B.12})$$

can be used to obtain (slightly) tighter inner and outer polyhedral bounds on S . The bound (B.12) can also be used to obtain a set of $(\sigma, \alpha, \gamma) \in S$ with a (slightly) improved guarantee $\gamma \leq (2 + \sqrt{3})^2 \gamma_p^*$. However in practice, we found that there was no practical improvement using these coefficients. ♣

References

- [1] D.J. Korteweg, G. de Vries, On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves, *Phil. Mag.* 39 (1895) 422–443.
- [2] J. Boussinesq, Théorie des ondes et des remous qui se propagent le long d'un canal rectangulaire horizontal, *J. Math. Pures Appl.* 17 (1872) 55–108.
- [3] F. Serre, Contribution à l'étude des écoulements permanents et variables dans les canaux, *Houille Blanche* 39 (1953) 830–872.
- [4] C.H. Su, C.S. Gardner, Korteweg-de vries equation and generalizations. III: derivation of the Korteweg-de vries equation and Burgers equation, *J. Math. Phys.* 10 (1969) 536–539.
- [5] A.E. Green, P.M. Naghdi, Derivation of equations for wave propagation in water of variable depth, *J. of Fluid Mech.* 78 (1976) 237–246.
- [6] T.Y. Wu, Modeling nonlinear dispersive water waves, *J. Eng. Mech.* 11 (1999) 747–755.
- [7] P.A. Madsen, H.B. Bingham, H. Liu, A new approach to high-order Boussinesq models, *J. Fluid Mech.* 462 (2002) 1–30.
- [8] P.A. Madsen, H.A. Schaffer, Higher-order Boussinesq-type equations for surface gravity waves: derivation and analysis, *Proc. R. Soc. London A* 356 (1998) 3123–3184.
- [9] Y. Matsuno, Hamiltonian formulation of the extended Green-Naghdi equations, *Phys. D* 301–302 (2015) 1–7.
- [10] W. Choi, High-order strongly nonlinear long wave approximation and solitary wave solution, *J. Fluid Mech.* 945 (2022) A15.
- [11] Y. Matsuno, Hamiltonian structure for two-dimensional extended Green-Naghdi equations, *Proc. R. Soc. A* 472 (2016) 20160127.
- [12] O. Nwogu, An alternative form of the Boussinesq equations for nearshore wave propagation, *J. Water Port Coastal Ocean Eng.* 119 (1993) 618–638.
- [13] P. Bonneton, F. Chazel, D. Lannes, F. Marche, M. Tissier, A splitting approach for the fully nonlinear and weakly dispersive Green-Naghdi models, *J. Comput. Phys.* 230 (2011) 1479–1498.
- [14] G. Wei, J.T. Kirby, S.T. Grilli, R. Subramanya, A fully nonlinear Boussinesq model for surface waves. Part I. Highly nonlinear unsteady waves, *J. Fluid Mech.* 294 (1995) 71–92.

- [15] B. Alvarez-Samaniego, D. Lannes, Large time existence for 3d water-waves and asymptotics, *Invent. Math.* 171 (2008) 485–541.
- [16] D. Lannes, P. Bonneton, Derivation of asymptotic two-dimensional time-dependent equations for surface water wave propagation, *Phys. Fluids* 21 (2009) 016601.
- [17] Y.A. Li, J.M. Hyman, W. Choi, A numerical study of the exact evolution equations for surface waves in water of finite depth, *Stud. Appl. Math.* 113 (2004) 303–324.
- [18] D. Dutykh, D. Clamond, P. Milewski, D. Mitsotakis, Finite volume and pseudo-spectral schemes for the fully nonlinear 1d Serre equations, *Eur. J. Appl. Math.* 24 (2013) 761–787.
- [19] O.S. Madsen, C.C. Mei, The transformation of a solitary wave over an uneven bottom, *J. Fluid Mech.* 39 (1969) 781–791.
- [20] T. Jang, H. Sung, J. Park, A non-local formulation for simulating the fully nonlinear Serre-Green-Naghdi equations for a solitary wave interaction with a variable slope, *Appl. Ocean Res.* 153 (2024) 104220.
- [21] D. Lannes, F. Marche, A new class of fully nonlinear and weakly dispersive Green-Naghdi models for efficient 2D simulations, *J. Comput. Phys.* 282 (2015) 238–268.
- [22] A. Samii, C. Dawson, An explicit hybridized discontinuous Galerkin method for Serre-Green-Naghdi wave model, *Comput. Methods Appl. Mech. Eng.* 330 (2018) 447–470.
- [23] O. Le Métayer, S. Gavrilyuk, S. Hank, A numerical scheme for the Green-Naghdi model, *J. Comput. Phys.* 229 (2010) 2034–2045.
- [24] M. Parisot, Entropy-satisfying scheme for a hierarchy of dispersive reduced models of free surface flow, *Internat. J. Numer. Methods Fluids* 91 (2019) 509–531.
- [25] S. Gavrilyuk, K.-M. Shyue, 2D Serre-Green-Naghdi equations over topography: elliptic operator inversion method, *J. Hydraulic Eng.* 150 (2023).
- [26] S. Noelle, M. Parisot, T. Tscherpel, A class of boundary conditions for the time-discrete Green-Naghdi equations with bathymetry, *SIAM J. Numer. Anal.* 60 (2022) 2681–2712.
- [27] S. Popinet, A quadtree-adaptive multigrid solver for the Serre-Green-Naghdi equations, *J. Comput. Phys.* 302 (2015) 336–358.
- [28] A. Duran, F. Marche, A discontinuous Galerkin method for a new class of Green-Naghdi equations on simplicial unstructured meshes, *Appl. Math. Model.* 45 (2017) 840–864.
- [29] V. Duchêne, C. Klein, Numerical study of the Serre-Green-Naghdi equations and a fully dispersive counterpart, *Discrete Contin. Dyn. Syst. Ser. B* 27 (2022) 5905–5933.
- [30] S. Gavrilyuk, C. Klein, Numerical study of the Serre-Green-Naghdi equations in 2D, *Nonlinearity* 37 (2024) 045014.
- [31] N. Favrie, S. Gavrilyuk, A rapid numerical method for solving Serre-Green-Naghdi equations describing long free surface gravity waves, *Nonlinearity* 30 (2017) 2718–2736.
- [32] J.-L. Guermond, C. Kees, B. Popov, E. Tovar, Robust explicit relaxation technique for solving the Green-Naghdi equations, *J. Comput. Phys.* 399 (2019) 108917.
- [33] S. Busto, M. Dumbser, C. Escalante, N. Favrie, S. Gavrilyuk, On high order ADER discontinuous Galerkin schemes for first order hyperbolic reformulations of nonlinear dispersive systems, *J. Sci. Comput.* 87 (2021) 48.
- [34] J.-L. Guermond, C. Kees, B. Popov, E. Tovar, Hyperbolic relaxation technique for solving the dispersive the Serre-Green-Naghdi equations with topography, *J. Comput. Phys.* 450 (2022) 110809.
- [35] S. Tkachenko, S. Gavrilyuk, J. Massoni, Extended Lagrangian approach for the numerical study of multidimensional dispersive waves: applications to the Serre-Green-Naghdi equations, *J. Phys.* 477 (2023) 111901.
- [36] J.-L. Guermond, C. Kees, B. Popov, E. Tovar, Well-balanced second-order convex limiting technique for solving the Serre-Green-Naghdi equations, *Water Waves* 4 (2022) 409–445.
- [37] H. Ranocha, M. Ricchiuto, Structure-preserving approximations of the Serre-Green-Naghdi equations in standard and hyperbolic form, *Numer. Methods Partial Differ. Eq.* 41 (2025) e70016.
- [38] P. Monk, *Finite Element Methods For Maxwell's Equations*, Oxford University Press, 2003.
- [39] M. Schechter, *Principles of Functional Analysis*, second ed., American Mathematical Society, 2001.
- [40] L. Trefethen, D. Bau, *Numerical Linear Algebra*, first ed., SIAM, 1997.
- [41] R.C. Kirby, From functional analysis to iterative methods, *SIAM Rev.* 52 (2010) 269–293.
- [42] M. Gilles, A. Townsend, Continuous analogues of Krylov subspace methods for differential operators, *SIAM J. Numer. Anal.* 57 (2019) 899–924.
- [43] J. Málek, Z. Strakoš, Preconditioning and the Conjugate Gradient Method in the Context of Solving PDEs, SIAM, 2015.
- [44] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependence Problems*, first ed., SIAM, 2007.
- [45] B. Seibold, D. Shirokoff, D. Zhou, Unconditional stability for multistep ImEx schemes-practice, *J. Comput. Phys.* 376 (2019) 295–321.
- [46] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II*, Springer Berlin Heidelberg, 1996.
- [47] U. Ascher, S.J. Ruuth, B. Wetton, Implicit-explicit methods for time dependent partial differential equations, *SIAM J. Numer. Anal.* 32 (1995) 797–823.
- [48] R. Rosales, B. Seibold, D. Shirokoff, D. Zhou, Unconditional stability for multistep ImEx schemes-theory, *SIAM J. Numer. Anal.* 55 (2017) 2336–2360.
- [49] J.W.S. Rayleigh, On waves, *Phil. Mag. Ser. 5* 1 (1876) 257–279.
- [50] W. Choi, A. Goullet, T.-C. Jo, An iterative method to solve a regularized model for strongly nonlinear long internal waves, *J. Comput. Phys.* 230 (2011) 2021–2030.
- [51] M. Miyata, Long internal waves of large amplitude, in: *Proceedings of the IUTAM Symposium on Nonlinear Water Waves*, Rome, Italy, (1988), pp. 399–406.
- [52] W. Choi, R. Camassa, Fully nonlinear internal waves in a two-fluid system, *J. Fluid Mech.* 396 (1999) 1–36.
- [53] W. Choi, R. Barros, A. Goullet, A regularized model for strongly nonlinear internal solitary waves, *J. Fluid Mech.* 629 (2009) 73–85.