

Numerical Methods for a Two-Dimensional "Bad" Boussinesq Equation by Removal of High-frequency Fourier Modes

Arief Anbiya^{1*}

¹Independent Researcher, Indonesia

Abstract. Numerical methods for a two-dimensional "bad" Boussinesq equation: $u_{tt} = u_{xx} + u_{xxx} + u_{yy} - 3(u^2)_{xx}$ are presented with good accuracy. The methods mainly depend on pseudo-spectral Fourier with a trimming of carefully chosen high-frequency Fourier modes. One method also relies on Runge-Kutta fourth order (RK4), and another method relies on Strang operator splitting. Before implementing the two methods, we analyze using Fourier series the linearized version of the equation by removing the nonlinear term $3(u^2)_{xx}$, and found that a particular bound or condition needs to be satisfied to avoid blow-up solution. We found that high-frequency Fourier modes that do not satisfy the condition must be excluded from the Fourier solution. We then apply this condition to the numerical methods for solving the nonlinear Boussinesq equation and found that including only the Fourier modes that satisfy the condition gives stable numerical solution with good accuracy up to $t = 100$. Including even just a few number of Fourier modes that violate the condition can result in a blow-up solution as early as $t = 23.5$. The accuracy of the method is measured by computing the L^∞ error against a soliton exact solution. The errors resulting from Runge-Kutta fourth order and Strang splitting numerical simulations differ slightly for small Δt , while there is a noticeable decrease in performance for the Strang splitting simulation as Δt increases. Using our numerical methods, we also display a simulation with Dirichlet boundary condition to account for wave reflections.

Key words and Phrases: Boussinesq equation, nonlinear PDE, Fourier series, Operator splitting, Numerical method.

*Corresponding author : ariefanbiya@gmail.com

2020 Mathematics Subject Classification: 35G20, 65Mxx, 76B15

Received:19-06-2025, accepted: 10-04-2026.

1. INTRODUCTION

The following one-dimensional classical Boussinesq equation was derived by J. Boussinesq [1] to model the propagation of dispersive long water waves with small amplitude:

$$u_{tt} = u_{xx} + u_{xxxx} + (u^2)_{xx}. \quad (1)$$

Although initially it was not explicitly referred to as “bad”, it is now often referred to as “bad” Boussinesq equation due to that its linearized version, $u_{tt} = u_{xx} + u_{xxxx}$, is not a well-posed problem in the Hadamard sense [2] for bounded domain. The ill-posedness can be revealed by analyzing the linearized version of the equation using Fourier series: for high enough frequencies, the Fourier coefficients can grow exponentially large, which means a small change of the initial condition can give a large change in the solution. Also, the ill-posedness persists in the nonlinear level: there exists a class of solutions of (1) that ‘blow up’ [3, 4]. More recent analysis by the same authors including on its asymptotic solutions can be seen in [5],[6],[7],[8]. On the other hand, if we change the sign of the u_{xxxx} term to negative, we have the ‘good’ Boussinesq equation which is linearly well-posed and not as intricate as (1) to be solved numerically. Papers on numerical solutions for the ‘good’ Boussinesq equation can be seen in [9] which uses a fourth order finite difference method, and in [10] that uses Strang operator splitting with high accuracy. Although the ‘good’ Boussinesq equation has applications and it is linearly well-posed and easier to solve, it does not model the same physical wave phenomenon as the “bad” Boussinesq equation. In [11], a numerical scheme for (1) is presented with good results. The scheme uses Fourier spectral method by trimming the high-frequency Fourier modes that have frequencies exceeding a bound. The bound is carefully chosen such that the solution of the linearized equation, via Fourier spectral method, does not blow-up or grow exponentially. The simple bound also reveals that wider domain, in the case of one-dimensional, allows more high-frequency modes to be included in the Fourier solution, which is consistent for modelling long waves that obviously need wide domain.

Inspired by [11], we want to adapt their method on a two-dimensional Boussinesq equation. A two-dimensional version was derived in 1996 as

$$u_{tt} = u_{xx} + u_{xxxx} + u_{yy} - 3(u^2)_{xx}. \quad (2)$$

The nonlinear equation (2) combines the two-way propagation of the classical Boussinesq equation in one-dimensional space with a weak dependence on the second spatial variable y [12]. As you will see in section 2, the linearized version of (2) can also be shown to be ill-posed using Fourier spectral method: it has high-frequency Fourier modes that can grow exponentially in time. From the linearized version, we then derive the condition that must be satisfied to avoid blow-up solution. First, we derive the ‘trimming’ condition for the case of square domain (6), before also generalizing to rectangular domain (8). Subsequently, we use this condition and apply the Fourier spectral method [13] to solve the nonlinear two-dimensional “bad” Boussinesq equation numerically. Under the ‘trimming’ condition, we can get stable numerical solutions of the nonlinear equation. Note that this

is not proving nonlinear well-posedness, but to show that the ‘trimming’ condition obtained from the linearized equation can also be used in the nonlinear equation for stability.

Similar to [11], our numerical scheme is implemented by first rewriting (2) as a system (12). We then view the approximate solution for u in terms of two-dimensional Fourier series. After substituting with Fourier series, we then must solve a system of ordinary differential equations in the frequency space and “trim” the Fourier modes with high enough frequencies that exceed our bound. We then compare our results by computing the numerical error against the exact solution presented in [14] in the form of soliton. We then also solve (2) numerically with a Dirichlet boundary condition to simulate wave reflections. The numerical scheme is found to provide enough stable and accurate approximation. The numerical results in section 4 show that the L^∞ errors between the numerical and exact solutions are small enough: less than 3% of the absolute initial peak $\max(|U(x, y, 0)|)$ for $t \in [0, 40]$, with spatial steps $\Delta x = \Delta y = 0.25$ and time step $\Delta t = 0.1$.

The solving of the system of PDEs involves two numerical methods: the Runge-Kutta fourth order (for solving ODEs in frequency space) and the Strang operator splitting (with exact solution in frequency space), both are paired with the ‘trimming’ condition (8). The Strang splitting is a method for solving a differential equation by first separating between the linear and nonlinear parts, before combining the solutions from the parts in a particular way. This method has been proven to have second order convergence on the nonlinear Korteweg de Vries (KdV) equation [15] and has been tested numerically on the ‘good’ Boussinesq equation with good accuracy [10]. In this paper, we found that the method can also be used to approximate the two-dimensional Boussinesq equation (2) by first writing it as a first order system. The numerical simulations are implemented in Octave software (<https://octave.org/>), the code for the numerical simulations are publicly available in the author’s Github repository https://github.com/anbarief/Boussinesq2D_Fourier/tree/main. The main contributions of this paper can be summarized as follows:

- (1) We extend and improvise from the numerical method presented in [11] for one-dimensional “bad” Boussinesq equation to a numerical method for a two-dimensional “bad” Boussinesq equation, by trimming some high-frequency Fourier modes in the pseudo-spectral implementation. We derive the trimming condition by analyzing the Fourier series solution of the linearized equation (3). Although the approach is similar to [11], the obtained trimming condition (6) in this paper is different and has more nuance.
- (2) In the pseudo-spectral numerical scheme, we compare two methods for solving the system of ODEs in the frequency space: Runge-Kutta fourth order and Strang operator splitting. Before using both methods, we first rewrite the nonlinear equation (2) as a first-order system. This is a mild improvisation of the work in [10], in which they also similarly rewrite the “good” Boussinesq equation as a first order system.

- (3) Although the trimming condition is derived from the linearized equation (3), it has significance when used in the nonlinear equation (2). We found that the numerical solution of (2) is stable up to $t = 100$ (see Figure 8), if we trim the high frequency modes that violate the stability condition (6). On the other hand, a mild violation of (6) result in a blow-up as early as $t = 23.5$ (see Figure 7). This may indicate that the linear term u_{xxxx} contributes more on the blow-up behavior, rather than the nonlinear term $-3(u^2)_{xx}$.

2. LINEARIZED SOLUTION

We first consider the linearized version of (2) that is attained by removing the term $3(u^2)_{xx}$ on a square domain:

$$u_{tt} = u_{xx} + u_{xxxx} + u_{yy}, \quad (x, y) \in \Omega = [-L, L]^2, \quad t > 0, \quad (3)$$

with initial conditions $u(x, y, 0) = u_0(x, y)$ and $u_t(x, y, 0) = v_0(x, y)$. We consider the $2L$ -periodic solution of (3) in terms of two-dimensional Fourier series on a square domain $[-L, L]^2$. Since the solution is assumed to be periodic by Fourier, then the equation is not conserving mass except when the condition $\int_{\Omega} u_t(x, y, 0) d\Omega = 0$ is met, so we also assume $v_0(x, y)$ satisfies this. To see this, let the total mass at time t be $M(t) = \int_{\Omega} u(x, y, t) d\Omega$, since Fourier-periodicity of $u(x, y, t)$ implies periodicity of its derivatives, then we have

$$\begin{aligned} \int_{-L}^L \left(\int_{-L}^L u_{xx} dx \right) dy &= \int_{-L}^L (u_x(L, y, t) - u_x(-L, y, t)) dy = 0 \\ \int_{-L}^L \left(\int_{-L}^L u_{yy} dy \right) dx &= \int_{-L}^L (u_y(x, L, t) - u_y(x, -L, t)) dx = 0 \\ \int_{-L}^L \left(\int_{-L}^L u_{xxxx} dx \right) dy &= \int_{-L}^L (u_{xxx}(L, y, t) - u_{xxx}(-L, y, t)) dy = 0. \end{aligned}$$

By equation (3), this implies $\int_{\Omega} u_{tt} d\Omega = M''(t) = 0$, so that $M'(t)$ is constant. Thus, condition $M'(0) = \int_{\Omega} u_t(x, y, 0) d\Omega = 0$ implies $M'(t) = 0$ for all t , which in turn implies $M(t)$ always constant. Next, the solution can be written as follows

$$u(x, y, t) = \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \hat{u}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)},$$

where $\omega_j = j/2L$, $\omega_k = k/2L$. By replacing u in the linear equation (3) with Fourier series, we have the following

$$\begin{aligned} \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \hat{u}_{j,k}''(t) e^{i2\pi(\omega_j x + \omega_k y)} &= \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} (i2\pi\omega_j)^2 \hat{u}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)} \\ &+ \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} (i2\pi\omega_j)^4 \hat{u}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)} + \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} (i2\pi\omega_k)^2 \hat{u}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)}. \end{aligned}$$

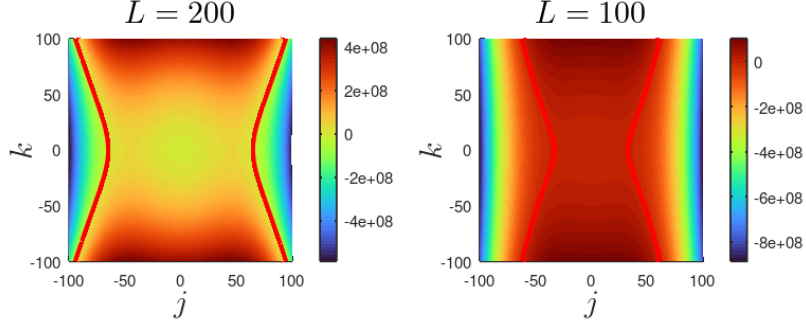


FIGURE 1. Comparison of the function $F(j, k) = L^2 k^2 + L^2 j^2 - \pi^2 j^4$ when $L = 100$ and $L = 200$. The red lines are where $F(j, k) = 0$.

Solving the above is equivalent as solving the following system

$$\hat{u}_{j,k}'' = [(2\pi\omega_j)^4 - (2\pi\omega_k)^2 - (2\pi\omega_j)^2] \hat{u}_{j,k}, \quad \forall (j, k) \in \mathbb{Z}^2, \quad t > 0. \quad (4)$$

The solution of each equation above depends on $(2\pi\omega_j)^4 - (2\pi\omega_k)^2 - (2\pi\omega_j)^2$, which we can categorize into three cases. Let $\lambda_{j,k} = (2\pi\omega_j)^4 - (2\pi\omega_k)^2 - (2\pi\omega_j)^2$ for each pair (j, k) , then we have the following

$$\hat{u}_{j,k} = \begin{cases} \hat{u}_{j,k}'(0)t + \hat{u}_{j,k}(0), & \lambda_{j,k} = 0 \\ \frac{\hat{u}_{j,k}(0) + \lambda_{j,k}^{-1/2} \hat{u}_{j,k}'(0)}{2} e^{\sqrt{\lambda_{j,k}} t} + \left(\hat{u}_{j,k}(0) - \frac{\hat{u}_{j,k}(0) + \lambda_{j,k}^{-1/2} \hat{u}_{j,k}'(0)}{2} \right) e^{-\sqrt{\lambda_{j,k}} t}, & \lambda_{j,k} > 0 \\ \frac{\hat{u}_{j,k}(0) + \frac{\hat{u}_{j,k}'(0)}{i\sqrt{-\lambda_{j,k}}}}{2} e^{i\sqrt{-\lambda_{j,k}} t} + \left(\hat{u}_{j,k}(0) - \frac{\hat{u}_{j,k}(0) + \frac{\hat{u}_{j,k}'(0)}{i\sqrt{-\lambda_{j,k}}}}{2} \right) e^{-i\sqrt{-\lambda_{j,k}} t}, & \lambda_{j,k} < 0. \end{cases} \quad (5)$$

Due to the term $e^{\sqrt{\lambda_{j,k}} t}$ for the case $\lambda_{j,k} > 0$ in (5), we can see that some of the Fourier coefficients can grow exponentially large as t increases. Therefore, the solution u can be unstable and blow-up solutions may occur if we include certain Fourier modes. The case $\lambda_{j,k} = 0$ may also generate Fourier mode that keeps increasing. However, it can happen only when $j = k = 0$ (provided we use integer L), and $\hat{u}_{0,0}(t)$ will be constant due to periodicity and mass conservation:

$$\text{Constant} = \int_{\Omega} u d\Omega = \int_{\Omega} \hat{u}_{0,0}(t) d\Omega + \underbrace{\int_{\Omega} \sum_{(j,k) \in \mathbb{Z}^2 - (0,0)} \hat{u}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)} d\Omega}_{0} = \hat{u}_{0,0}(t) (2L)^2$$

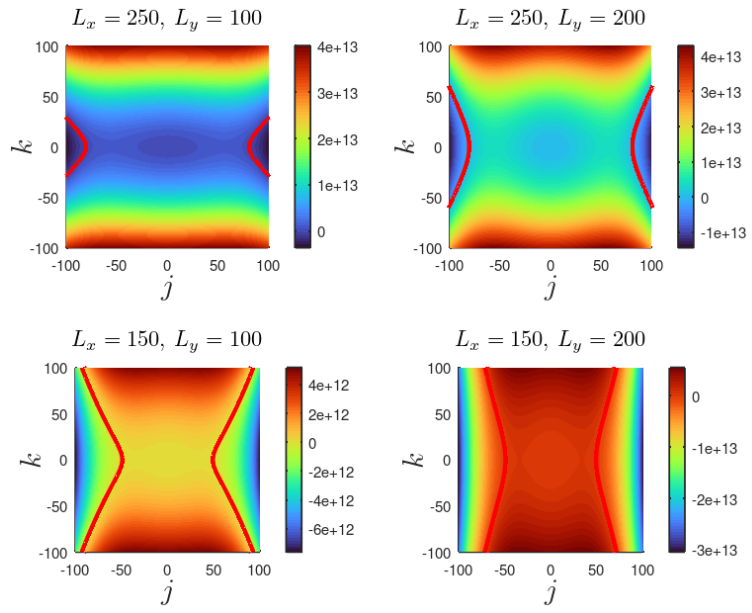


FIGURE 2. Comparison of the function $F_2(j, k) = L_x^4 k^2 + L_y^2 (L_x^2 j^2 - \pi^2 j^4)$ with four combinations of L_x and L_y . The red lines are where $F_2(j, k) = 0$.

Since the case $\lambda_{j,k} \leq 0$ gives stable solution, we attempt to neglect the Fourier modes that result in $\lambda_{j,k} > 0$. The condition for $\lambda_{j,k} \leq 0$ can be derived as

$$\begin{aligned}
 (2\pi\omega_j)^4 - (2\pi\omega_k)^2 - (2\pi\omega_j)^2 &\leq 0 \\
 (2\pi(j/2L))^4 - (2\pi(k/2L))^2 - (2\pi(j/2L))^2 &\leq 0 \\
 (\pi j/L)^4 - (\pi k/L)^2 - (\pi j/L)^2 &\leq 0 \\
 (\pi j)^4/L^2 - (\pi k)^2 - (\pi j)^2 &\leq 0 \\
 (\pi j)^4/L^2 &\leq (\pi k)^2 + (\pi j)^2,
 \end{aligned}$$

which then gives us

$$0 \leq L^2 k^2 + L^2 j^2 - \pi^2 j^4. \quad (6)$$

From (6) we see that using a small domain (small L) will limit the number of ‘stable’ Fourier modes that we can use: the number of combinations (j, k) that we can use is limited. Using wide domain (large L) should give better accuracy due to more ‘stable’ Fourier modes (see Figure 1). We want the following function

$$F(j, k) = L^2 k^2 + L^2 j^2 - \pi^2 j^4$$

to be non-negative. From Figure 1, we can see that this is achieved in between the red lines. Note also that we can have high number for k , so we can have high frequencies without limit for the Fourier modes in y -direction. Note also that this

inhibiting property does not violate the water wave model, since the assumptions for long wave imply that the domain is infinitely large ($L \rightarrow \infty$), in which case the Fourier series solution can be fully used. However, in the numerical approximation, we need to exclude high-frequency modes that do not follow (6), and this can be a good approximation since long waves rely more on low-frequency Fourier modes. We will use the condition (6) for the nonlinear Boussinesq equation (2), even though we derived this from the linearized equation. We will see that using this condition gives stable nonlinear numerical solution. Violating the condition, even only a little, result in a blow-up solution (see Figure 7). Also note that, if we assume u^2 as a $2L$ -periodic Fourier series, then condition $M'(0) = 0$ for mass conservation also applies in the nonlinear equation, since $\int_{\Omega} (u^2)_{xx} d\Omega = 0$.

3. RECTANGULAR DOMAIN

We can extend the method for a rectangular domain. The linearized equation on a rectangular domain is

$$u_{tt} = u_{xx} + u_{xxxx} + u_{yy}, \quad (x, y) \in \Omega = [-L_x, L_x] \times [-L_y, L_y], \quad t > 0, \quad (7)$$

with initial conditions $u(x, y, 0) = u_0(x, y)$ and $u_t(x, y, 0) = v_0(x, y)$. We consider the periodic solution of (7) in terms of two-dimensional Fourier series. By the same approach for the square domain, it is not difficult to see that $M'(0) = 0$ guarantees mass conservation. Plugging the Fourier series into the equation and then solving the ODE in frequency space will give us the same result as (5), but with frequencies defined differently as $\omega_j = j/(2L_x)$, $\omega_k = k/(2L_y)$. The condition for $\lambda_{j,k} \leq 0$ can then be derived as

$$\begin{aligned} (2\pi\omega_j)^4 - (2\pi\omega_k)^2 - (2\pi\omega_j)^2 &\leq 0 \\ (2\pi(j/2L_x))^4 - (2\pi(k/2L_y))^2 - (2\pi(j/2L_x))^2 &\leq 0 \\ (\pi j/L_x)^4 - (\pi k/L_y)^2 - (\pi j/L_x)^2 &\leq 0 \\ \pi^2(j/L_x)^4 - (k/L_y)^2 - (j/L_x)^2 &\leq 0 \\ L_y^2 \pi^2 j^4 - L_x^4 k^2 - (L_x L_y)^2 j^2 &\leq 0, \end{aligned}$$

which then gives us

$$0 \leq L_x^4 k^2 + L_y^2 (L_x^2 j^2 - \pi^2 j^4). \quad (8)$$

We want the following bivariate function to be non-negative:

$$F_2(j, k) = L_x^4 k^2 + L_y^2 (L_x^2 j^2 - \pi^2 j^4).$$

The plots of $F_2(j, k)$ are shown for several combinations of L_x, L_y in Figure 2.

From (8), we can see that the first term on the right hand side, $L_x^4 k^2$, will always be non-negative. Also, k only appears in this first term as a factor k^2 . Thus, large values of $|k|$ will always be welcomed for stability. However, the factor $L_x^2 j^2 - \pi^2 j^4$ of the second term can be negative, depending on how large $|j|$ is relative to L_x . Thus, for a large enough $|j|$, we need to combine them with large values of $|k|$ for stability, thus sacrificing low values of $|k|$. Once the second term

becomes negative, the absolute magnitude of this term is amplified by L_y^2 . Thus, if the value L_x and $|j|$ stay the same, increasing L_y will sacrifice more low values of $|k|$. In this regard, we present Theorem 3.1 on the stability condition (8). By Theorem 3.1, we can conclude that there is no upper bound of $|k|$, but there is a lower bound that depends on j . Also, by Theorem 3.1, we can conclude that $|j|$ has only an upper bound that depends on k .

Theorem 3.1. *Given $L_x, L_y > 0$, the ordered pair of values (j, k) that satisfy (8) must follow the following rules:*

(1) $|k|$ is bounded below. The lower bound depends on j :

$$\sqrt{\frac{L_y^2}{L_x^4} [\pi^2 j^4 - L_x^2 j^2]} \leq |k|, \text{ if } \frac{L_y^2}{L_x^4} [\pi^2 j^4 - L_x^2 j^2] \geq 0 \quad (9)$$

$$k \in \mathbb{R}, \text{ if } \frac{L_y^2}{L_x^4} [\pi^2 j^4 - L_x^2 j^2] < 0,$$

(2) $|j|$ is not bounded below by any positive number, but is bounded above. The upper bound depends only on k :

$$|j| \leq L_x \sqrt{\frac{1}{2\pi^2} + \frac{\sqrt{L_y^2 + 4(k\pi)^2}}{2\pi^2 L_y}}. \quad (10)$$

Proof. To prove the lower bound of $|k|$, we simply rewrite (8) as $\frac{L_y^2}{L_x^4} [\pi^2 j^4 - L_x^2 j^2] \leq k^2$, the result is then obvious. To prove the upper bound of $|j|$, let $z = j^2$, then we must have

$$L_y^2 [\pi^2 z^2 - L_x^2 z] \leq L_x^4 k^2$$

$$\pi^2 z^2 - L_x^2 z - \frac{L_x^4 k^2}{L_y^2} \leq 0$$

The determinant of the above quadratic form is

$$D = L_x^4 \left(\frac{L_y^2 + 4(k\pi)^2}{L_y^2} \right) > 0$$

The roots are

$$z_{1,2} = \frac{1}{2\pi^2} \left(L_x^2 \pm \frac{L_x^2}{L_y} \sqrt{L_y^2 + 4(k\pi)^2} \right)$$

It is easy to see that we can take a very large z -value that is bigger than both of the roots above such that the quadratic will have a positive value. Thus, the quadratic will have non-positive values only when z is between both roots. So we must have

$$\left| z - \frac{L_x^2}{2\pi^2} \right| \leq \frac{L_x^2}{2\pi^2 L_y} \sqrt{L_y^2 + 4(k\pi)^2},$$

which means

$$\frac{L_x^2}{2\pi^2} - \frac{L_x^2}{2\pi^2 L_y} \sqrt{L_y^2 + 4(k\pi)^2} \leq j^2 \leq \frac{L_x^2}{2\pi^2} + \frac{L_x^2}{2\pi^2 L_y} \sqrt{L_y^2 + 4(k\pi)^2}. \quad (11)$$

Notice that j is bounded above by

$$|j| \leq L_x \sqrt{\frac{1}{2\pi^2} + \frac{\sqrt{L_y^2 + 4(k\pi)^2}}{2\pi^2 L_y}}.$$

On the other hand, since $\frac{\sqrt{L_y^2 + 4(k\pi)^2}}{L_y} \geq 1$, then the lower bound is always non-positive:

$$\frac{L_x^2}{2\pi^2} \left(1 - \frac{\sqrt{L_y^2 + 4(k\pi)^2}}{L_y} \right) \leq 0,$$

which means we can neglect the lower bound in (11). $\square$

The bound (9) tells us that the lower bound of $|k|$ increases as j increases. This is consistent with the red lines in Figure 2, on which $F_2(j, k) = 0$. All combinations of (j, k) in between the red lines are allowed. The red lines increase to the right or left, as $|j|$ increases. Moreover, the bound (9) also shows that increasing L_y will increase the lower bound of $|k|$, which means excluding more $|k|$ values from below. As we can see from the figure, the case $L_x = 250, L_y = 100$ noticeably allows more k values than the case $L_x = 250, L_y = 200$. Similar behavior can be seen between the cases $L_x = 150, L_y = 100$ and $L_x = 150, L_y = 200$.

The bound (10) tells us that the range of allowed j values increases as $|k|$ increases. In addition, increasing L_x will allow more j values in combination with lower k values. This is consistent with Figure 2. The plots on the first row of Figure 2 (with $L_x = 250$) has wider range of accepted j values paired with lower k values than the plots on the second row (with $L_x = 150$).

4. NUMERICAL SCHEME 1: RUNGE-KUTTA AND PSEUDO-SPECTRAL

We now present the pseudo-spectral numerical scheme that utilizes the bound (8). We can write (2) as a first-order system by letting $v = u_t$ as follows

$$\begin{bmatrix} u_t \\ v_t \end{bmatrix} = \begin{bmatrix} v \\ u_{xx} + u_{xxxx} + u_{yy} - 3(u^2)_{xx} \end{bmatrix} \quad (12)$$

We view the solutions u, v inside the domain $[-L_x, L_x] \times [-L_y, L_y]$ using Fourier series. Let $x_{n_1} = -L_x + (n_1 - 1)\Delta x$, $y_{n_2} = -L_y + (n_2 - 1)\Delta y$, for $n_1 = 1, \dots, N_1$ and $n_2 = 1, \dots, N_2$, where $\Delta x = \frac{2L_x}{N_1 - 1}$ and $\Delta y = \frac{2L_y}{N_2 - 1}$. Let U, V be the approximate solutions of u, v at each point (x_{n_1}, y_{n_2}, t) . We write the approximate solution U as follows

$$u(x, y, t) = \sum_{k \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \hat{u}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)},$$

$$U(x, y, t) = \sum_{k \in \zeta_2} \sum_{j \in \zeta_1} \widehat{U}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)}, \quad (13)$$

where $\omega_j = j/(2L_x)$, $\omega_k = k/(2L_y)$, and

$$\zeta_1 = (\zeta_{1,1}, \zeta_{1,2}, \dots, \zeta_{1,N_1}) = \begin{cases} (0, 1, \dots, \frac{N_1}{2} - 1, -\frac{N_1}{2}, \dots, -1), & \text{if } N_1 \text{ is even} \\ (0, 1, \dots, \frac{N_1-1}{2}, -\frac{N_1-1}{2}, \dots, -1), & \text{if } N_1 \text{ is odd.} \end{cases} \quad (14)$$

$$\zeta_2 = (\zeta_{2,1}, \zeta_{2,2}, \dots, \zeta_{2,N_2}) = \begin{cases} (0, 1, \dots, \frac{N_2}{2} - 1, -\frac{N_2}{2}, \dots, -1), & \text{if } N_2 \text{ is even} \\ (0, 1, \dots, \frac{N_2-1}{2}, -\frac{N_2-1}{2}, \dots, -1), & \text{if } N_2 \text{ is odd.} \end{cases} \quad (15)$$

Let $w = u^2$, we write w as a different Fourier series. The approximate solutions in Fourier series for v and w are

$$V(x, y, t) = U_t(x, y, t) = \sum_{k \in \zeta_2} \sum_{j \in \zeta_1} \widehat{U}'_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)},$$

$$W(x, y, t) = \sum_{k \in \zeta_2} \sum_{j \in \zeta_1} \widehat{W}_{j,k}(t) e^{i2\pi(\omega_j x + \omega_k y)}.$$

By plugging U, V, W to the system (12), we get the following system of ordinary differential equations for $\forall (j, k) \in \zeta_1 \times \zeta_2$

$$\begin{aligned} \widehat{U}'_{j,k} &= \widehat{V}_{j,k}, \\ \widehat{V}'_{j,k} &= [-(2\pi\omega_j)^2 + (2\pi\omega_j)^4 - (2\pi\omega_k)^2] \widehat{U}_{j,k} + 3(2\pi\omega_j)^2 \widehat{W}_{j,k}. \end{aligned} \quad (16)$$

For all $t \geq 0$, let $\mathbf{U}(t), \mathbf{V}(t), \mathbf{W}(t)$ be matrices of size $N_1 \times N_2$ with elements that are the values of functions U, V, W at points (x_{n_1}, y_{n_2}, t) : the entry at row a and column b is the value of each function at point (x_a, y_b, t) . Additionally, let $\widehat{\mathbf{U}}(t), \widehat{\mathbf{V}}(t), \widehat{\mathbf{W}}(t)$ be matrices of size $N_1 \times N_2$ with elements that are the values of the Fourier coefficients at time t : the entry at row a and column b is the coefficient of the Fourier mode at wave-index $j = \zeta_{1,a}, k = \zeta_{2,b}$. Define $\text{FFT}(\cdot)$ as the operator of two-dimensional discrete Fourier transform. Define the ‘trimming’ matrix $\mathcal{D} = \mathcal{D}(L_x, L_y)$, a matrix of size $N_1 \times N_2$, such that the entry at row a , column b is

$$\mathcal{D}_{a,b} = \begin{cases} 0, & \text{if } \frac{L_x^4 \zeta_{2,b}^2 + (L_x L_y)^2 \zeta_{1,a}^2}{L_y^2 \zeta_{1,a}^4} < \pi^2, \\ 1, & \text{if } \frac{L_x^4 \zeta_{2,b}^2 + (L_x L_y)^2 \zeta_{1,a}^2}{L_y^2 \zeta_{1,a}^4} \geq \pi^2, \text{ or if } (\zeta_{1,a}, \zeta_{2,b}) = (0, 0). \end{cases} \quad (17)$$

Note that every operation of $\text{FFT}(\cdot)$ must be followed by element-wise multiplication by $\mathcal{D}$ to ensure the ‘trimming’ of high-frequency modes. The element-wise multiplication will cancel the high-frequency unstable modes and preserve the stable ones. At time $t = 0$, we set $\widehat{\mathbf{U}}(0) = \mathcal{D} \odot \text{FFT}(\mathbf{U}(0))$ and $\widehat{\mathbf{V}}(0) = \mathcal{D} \odot \text{FFT}(\mathbf{V}(0))$, where $\odot$ represents element-wise multiplication. We then solve (16) numerically using the Runge-Kutta fourth order [16] method in Octave software, manually. At each time step, we update (16) with $\widehat{\mathbf{W}}$ computed as

$$\widehat{\mathbf{W}}(t) = \mathcal{D} \odot \text{FFT} \left(\left[\text{FFT}^{-1}(\widehat{\mathbf{U}}(t)) \right]^2 \right). \quad (18)$$

After each time iteration, we take the inverse at each time point $t_m = m\Delta t$ to retain the solutions U and V :

$$\mathbf{U}(t_m) = \text{FFT}^{-1}(\widehat{\mathbf{U}}(t_m)), \quad \mathbf{V}(t_m) = \text{FFT}^{-1}(\widehat{\mathbf{V}}(t_m)). \quad (19)$$

5. NUMERICAL SCHEME 2: STRANG SPLITTING AND PSEUDO-SPECTRAL

As an alternative scheme other than Runge-Kutta fourth order, we present the Strang Splitting method [10, 15, 17].

5.1. Intro: Strang Splitting on ODE.

The Strang splitting is a method for solving a differential equation by first separating between the linear and nonlinear parts, before combining the solutions of each individual part in a particular way. As a warm-up example, let us consider nonlinear ODEs of the form:

$$y'(t) = A(y) + B(y), \quad y(0) = y_0,$$

where A and B are linear and nonlinear functions of y , respectively. To use Strang splitting method, first we must have solutions, or approximate solutions in case of numerics, for each of the following linear and nonlinear equations with arbitrary initial values:

$$y'(t) = A(y), \quad y(0) = y_0, \quad (20)$$

and

$$y'(t) = B(y), \quad y(0) = y_0. \quad (21)$$

Let the operation $\phi_A(t) \circ y_0$ represents the solution of the linear equation (20) with variable initial value y_0 . Similarly, let the operation $\phi_B(t) \circ y_0$ represents the solution of the nonlinear equation (21) with variable initial value y_0 . Let the time be discretized as $t_m = m\Delta t$, $m = 0, 1, \dots, M$. Let $y(t_m)$ is known for some m , then we can approximate $y(t_{m+1})$ via Strang splitting method. The Strang splitting method has three steps. In the 1st step, we compute the nonlinear solution of (21) at time $t = \Delta t/2$ with $y(t_m)$ as initial value: we compute

$$\phi_B(\Delta t/2) \circ y(t_m). \quad (22)$$

In the 2nd step, we use (22) as the initial value for the linear solution of (20) and compute the solution at time $t = \Delta t$: we compute

$$\phi_A(\Delta t) \circ (\phi_B(\Delta t/2) \circ y(t_m)). \quad (23)$$

Lastly, in the 3rd step we use (23) as the initial value again for the nonlinear solution of (21) and compute the solution at time $t = \Delta t/2$: we compute

$$\phi(\Delta t/2) \circ (\phi_A(\Delta t) \circ (\phi_B(\Delta t/2) \circ y(t_m))).$$

This last value is considered as approximation for $y(t_{m+1})$, that is

$$y(t_{m+1}) \approx \phi_B(\Delta t/2) \circ \phi_A(\Delta t) \circ \phi_B(\Delta t/2) \circ y(t_m).$$

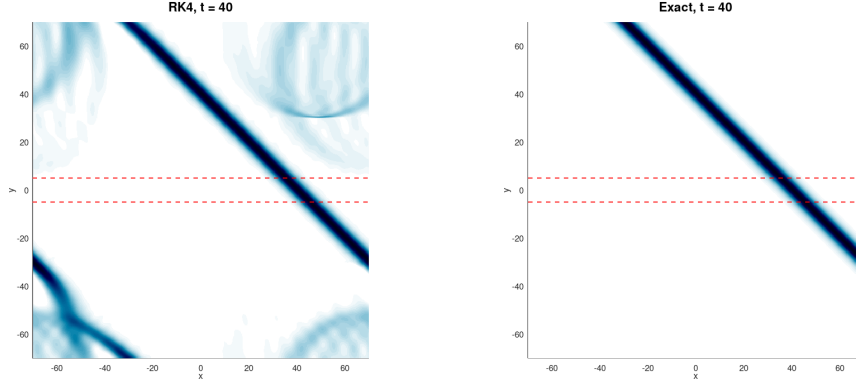


FIGURE 3. The colormap plot of the numerical solution at $t = 40$ for the first simulation via Runge-Kutta method (left). The red dashed lines indicate $y = -5$ and $y = 5$, we only compute the error in between these lines.

5.2. Strang Splitting on “bad” Boussinesq Equation with Pseudo Spectral Fourier.

Moving on from ODE, the Strang splitting method can also be used to numerically solve the two-dimensional “bad” Boussinesq system (12). Here, we use the same notations for Fourier series and matrices mentioned in Section 4. The system (12) can be viewed as

$$Y_t = A(Y) + B(Y), \quad Y(x, y, t) = [u(x, y, t), v(x, y, t)]^T, \quad (24)$$

where

$$A(Y) = \begin{bmatrix} v \\ u_{xx} + u_{xxxx} + u_{yy} \end{bmatrix}, \quad B(Y) = \begin{bmatrix} 0 \\ -3(u^2)_{xx} \end{bmatrix}.$$

Let the operation $\Phi_A(x, y, t) \circ Y_0$ represents the solution (whether exact or numerical) of $Y_t = A(Y)$ with Y_0 as the variable for the initial condition vector. Let the operation $\Phi_B(x, y, t) \circ Y_0$ represents the solution (whether exact or numerical) of $Y_t = B(Y)$ with Y_0 as the variable for the initial condition vector. Similar to the ODE case, the Strang splitting scheme is

$$\begin{aligned} & [u(x, y, t_{m+1}), v(x, y, t_{m+1})]^T \\ & \approx \Phi_B(x, y, \Delta t/2) \circ \Phi_A(x, y, \Delta t) \circ \Phi_B(x, y, \Delta t/2) \circ [u(x, y, t_m), v(x, y, t_m)]^T. \end{aligned} \quad (25)$$

To be able to use (25), we need to know at least how to approximate Φ_A and Φ_B . We start with the linear system:

$$u_t = v, \quad (26)$$

$$v_t = u_{xx} + u_{xxxx} + u_{yy}, \quad (27)$$

with variable initial condition vector $Y_0 = [u_0, v_0]^T$. This linear system is exactly the linearized two-dimensional Boussinesq equation (3), and we can solve it in

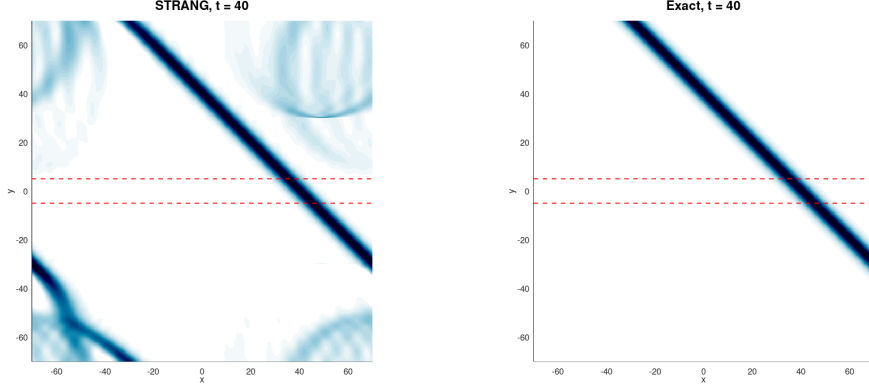


FIGURE 4. The colormap plot of the numerical solution at $t = 40$ for the first simulation via Strang splitting method (right). The red dashed lines indicate $y = -5$ and $y = 5$, we only compute the error in between these lines.

the frequency space with exact solution (5). Numerically, using pseudo-spectral Fourier, we approximate (5) for each point (j, k) as:

$$\hat{U}_{j,k}(t) = \begin{cases} \hat{V}_{j,k}(0)t + \hat{U}_{j,k}(0), & \lambda_{j,k} = 0 \\ \frac{\hat{U}_{j,k}(0) + \lambda_{j,k}^{-1/2} \hat{V}_{j,k}(0)}{2} e^{\sqrt{\lambda_{j,k}} t} + \left(\hat{U}_{j,k}(0) - \frac{\hat{U}_{j,k}(0) + \lambda_{j,k}^{-1/2} \hat{V}_{j,k}(0)}{2} \right) e^{-\sqrt{\lambda_{j,k}} t}, & \lambda_{j,k} > 0 \\ \frac{\hat{U}_{j,k}(0) + \frac{\hat{V}_{j,k}(0)}{i\sqrt{-\lambda_{j,k}}}}{2} e^{i\sqrt{-\lambda_{j,k}} t} + \left(\hat{U}_{j,k}(0) - \frac{\hat{U}_{j,k}(0) + \frac{\hat{V}_{j,k}(0)}{i\sqrt{-\lambda_{j,k}}}}{2} \right) e^{-i\sqrt{-\lambda_{j,k}} t}, & \lambda_{j,k} < 0, \end{cases} \quad (28)$$

$$\hat{V}_{j,k}(t) = \hat{U}'_{j,k}(t).$$

We will only use the case where $\lambda_{j,k} \leq 0$, before performing inverse discrete Fourier transform. The inverse discrete Fourier transform of $\mathcal{D} \odot \hat{U}(t)$ and $\mathcal{D} \odot \hat{V}(t)$ computed from (28) are the approximation for $\Phi_A(x, y, t) \circ Y_0$ on all discrete spatial points (x_{n_1}, y_{n_2}) .

Next, we will approximate Φ_B for the nonlinear system:

$$u_t = 0, \quad (29)$$

$$v_t = -3(u^2)_{xx}, \quad (30)$$

with variable initial condition vector $Y_0 = [u_0, v_0]^T$. Via pseudo-spectral Fourier, the equivalent system in frequency space for each point (j, k) can be approximated as:

$$\begin{aligned} \hat{U}'_{j,k} &= 0 \\ \hat{V}'_{j,k} &= 3(2\pi\omega_j)^2 \hat{W}_{j,k}. \end{aligned} \quad (31)$$

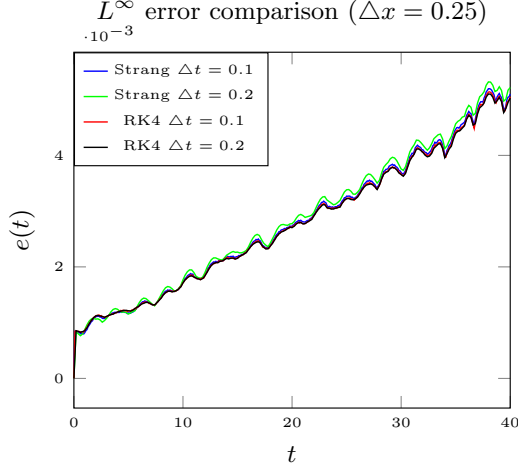


FIGURE 5. The L^∞ error for both numerical methods up to $t = 40$ for the first simulation, using $\Delta x = 0.25$ and two different time steps $\Delta t = 0.1, 0.2$.

Since the time derivative for the first variable is 0, we have constant $\widehat{U}_{j,k}(t) = \widehat{U}_{j,k}(0)$ for each (j, k) . As a consequence, we have $\widehat{\mathbf{W}}(t) = \text{FFT} \left([\text{FFT}^{-1}(\widehat{\mathbf{U}}(0))]^2 \right)$, which means $\widehat{W}_{j,k}(t)$ is constant and therefore $\widehat{V}_{j,k}(t)$ is linear. Therefore, the solution for (31) can be summarized as

$$\begin{aligned} \widehat{\mathbf{U}}(t) &= \widehat{\mathbf{U}}(0), \\ \widehat{\mathbf{V}}(t) &= \widehat{\mathbf{V}}(0) + t \left(3(2\pi\vec{\omega}_x)^2 \odot \text{FFT} \left([\text{FFT}^{-1}(\widehat{\mathbf{U}}(0))]^2 \right) \right), \end{aligned} \quad (32)$$

where $\vec{\omega}_x$ is a matrix with each column vector equals $[\omega_{\zeta_1}, \dots, \omega_{\zeta_{N_1}}]^T$ and $\odot$ is element-wise multiplication. Again, we only consider (32) for the points (j, k) where $\lambda_{j,k} \leq 0$. We then perform inverse discrete Fourier transform of $\mathcal{D} \odot \widehat{\mathbf{U}}(t)$ and $\mathcal{D} \odot \widehat{\mathbf{V}}(t)$ to get the approximation for $\Phi_B(x, y, t) \circ Y_0$ on all discrete spatial points (x_{n_1}, y_{n_2}) .

6. NUMERICAL RESULTS

The numerical computation is conducted using Octave software and especially its built-in two-dimensional fast Fourier transform function `fft2` and the inverse `ifft2`. The exact solution of (2) from [14] is of the form

$$u(x, y, t) = -\frac{2\alpha^4 b_0 e^{\alpha x + \beta y - t\sqrt{\beta^2 + \alpha^2 + \alpha^4}}}{(1 + \alpha^2 b_0 e^{\alpha x + \beta y - t\sqrt{\beta^2 + \alpha^2 + \alpha^4}})^2}. \quad (33)$$

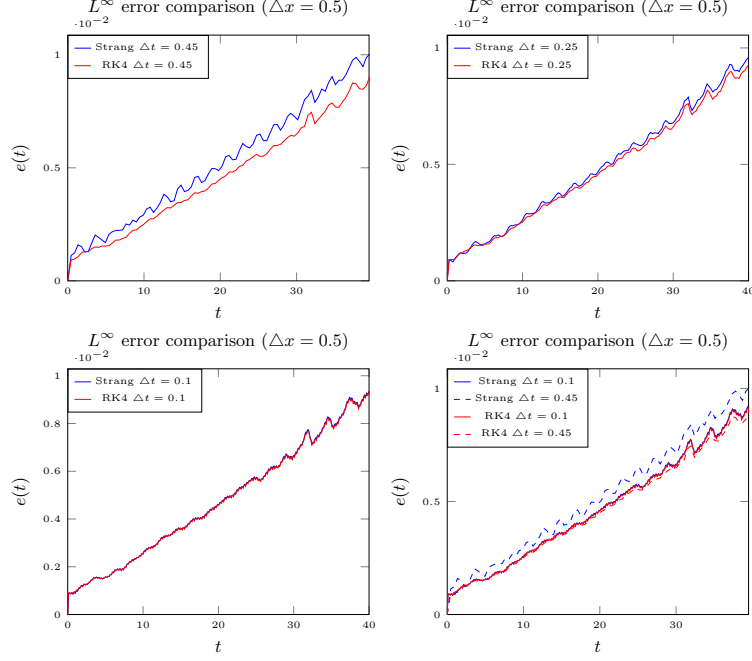


FIGURE 6. The L^∞ error for both numerical methods up to $t = 40$ for the first simulation, using $\Delta x = 0.5$ and three different time steps $\Delta t = 0.1, 0.25, 0.45$.

This solution is a soliton wave travelling at constant speed. Since the exact solution is a travelling soliton, then it is mass conserving, so we do not need to check whether $\int_{\Omega} u_t(x, y, 0) d\Omega = 0$. To measure accuracy at each time point t_m , we use the L^∞ error function

$$e(t) = \sup_{(x,y) \in \tilde{\Omega}} |U(x, y, t) - u(x, y, t)|,$$

where $\tilde{\Omega} = \{x_{n_1}, 1 \leq n_1 \leq N_1\} \times \{y_{n_2}, 1 \leq n_2 \leq N_2\}$.

For the first setting of initial condition and domain, we use initial condition of the form (33) with $\alpha = 0.6$, $\beta = 0.6$, and $b_0 = 10^6$ over the domain $\Omega = [-70, 70]^2$. The exact solution with this setting is a soliton wave heading the northeast direction with constant speed. Using this setting, we numerically solve the Boussinesq system (12) for $t \in [0, 40]$ using both the Runge-Kutta method and Strang splitting with $\Delta x = \Delta y = 0.25$ and $\Delta t = 0.1, 0.2$. Due to the automatic Fourier periodic boundary condition, waves passing through the boundary will reappear again from the opposite side. As a consequence, we only compute the L^∞ error over the subdomain $y \in [-5, 5]$: in this subdomain, the waves from periodic boundary condition will not appear at least before $t = 40$. The plot of the numerical solutions using $\Delta x = \Delta y = 0.25$ and $\Delta t = 0.1$ are shown in Figure 3 and 4. As we can see, the

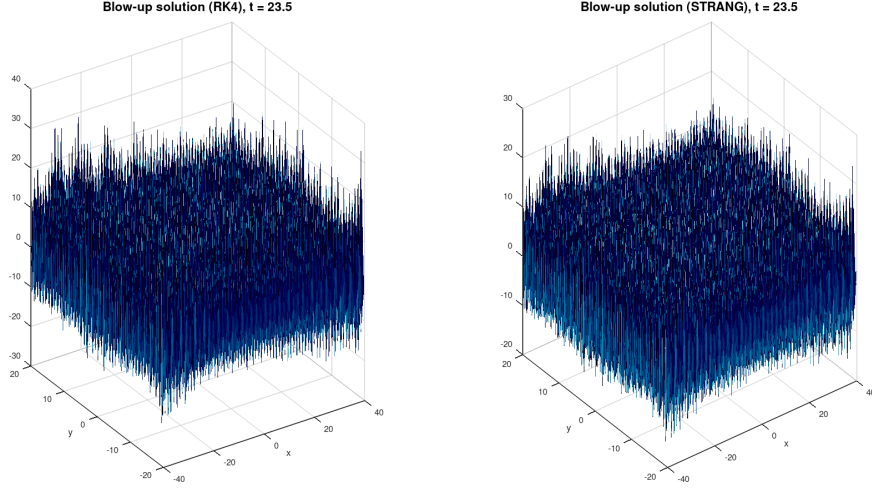


FIGURE 7. Surface plots of the blow-up numerical solutions at $t = 23.5$ when we minorly violate the stable condition (8).

effect of periodic boundary condition can be seen around the corners of the domain. However, both numerical solutions have solitons heading in the same direction as the exact solution, and with approximately the same shape as well. The L^∞ error is plotted in Figure 5 for both time step scenarios $\Delta t = 0.1, 0.2$. We can see that both methods produce good results, with $e(40) \approx 0.005 < 2.8\%$ of the initial condition's absolute peak $\max |U(x, y, 0)| = 0.18$. The error plot shows that the performance of Runge-Kutta fourth order is slightly better than Strang splitting for $t \in [5, 40]$ for $\Delta t = 0.2$. However, for $\Delta t = 0.1$, the performance of both methods are almost indistinguishable. Additionally, we also simulate with $\Delta x = \Delta y = 0.5$ and $\Delta t = 0.1, 0.25, 0.45$ to compare the numerical errors. The L^∞ errors resulting from cases $\Delta x = \Delta y = 0.5$ with $\Delta t = 0.1, 0.25, 0.45$ are shown in Figure 6. From these plots, it is clear that the performance difference between Runge-Kutta fourth order and Strang splitting becomes more noticeable as Δt increases. For $\Delta t = 0.1, 0.25, 0.45$, the Runge-Kutta fourth order performs well and shows similar results for all three time steps, while the Strang splitting becomes approximately as good as Runge-Kutta only when $\Delta = 0.1$.

For the second simulation, we test whether our numerical scheme really depends on the linear ‘trimming’ condition (8). We tried a simulation where we minorly disobey the ‘trimming’ condition: we include the points (j, k) where

$$0 \leq L_x^4 k^2 + (L_x L_y)^2 j^2 - 0.99\pi^2 L_y^2 j^4,$$

therefore allowing a few of the ‘illegal’ high-frequency Fourier modes in our solution. We use initial condition (33) with $\alpha = 0.25$, $\beta = 2$, and $b_0 = 16$, over the domain $(x, y) \in [-40, 40] \times [-20, 20]$. The discrete spatial and time steps are $\Delta x = \Delta y = 0.25$ and $\Delta t = 0.1$. After performing the numerical simulation, it turns out that at

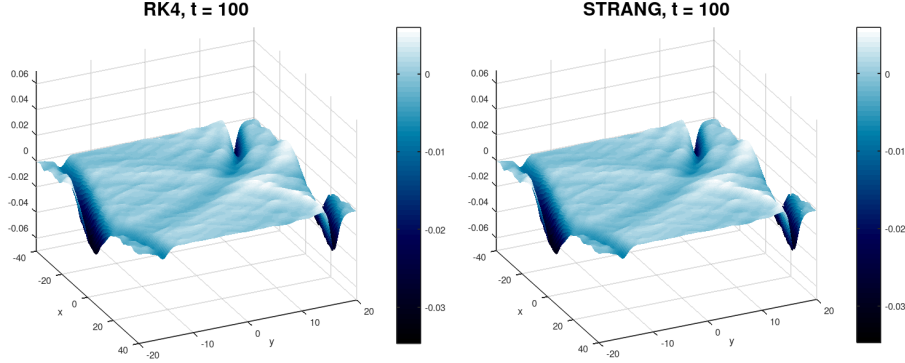


FIGURE 8. Surface plots of the numerical solutions at $t = 100$ via Runge-Kutta and Strang splitting methods by following the stable condition (8).

time $t = 23.5$, a blow-up solution is apparent as shown in Figure 7. However, we also conducted a longer simulation using the same setting but obeying the linear ‘trimming’ condition (8), and the results are stable solutions up to $t = 100$ (see Figure 8). This suggests that the nonlinear Boussinesq equation (2) also depends on the linear condition (8).

Lastly, we perform another numerical simulation (only using Runge-Kutta fourth order method) on a rectangular domain $[-40, 40] \times [-20, 20]$ with Dirichlet boundary condition $U(x, y, t) = 0$ on the set of spatial points

$$\begin{aligned} \partial\Omega_{\text{dirichlet}} = & \{(x, y) | x \in [-40, -40], y \in [-20, -15]\} \\ & \cup \{(x, y) | x \in [-40, 40], y \in [15, 20]\}. \end{aligned}$$

We use the same initial condition (33) as the previous simulation with $\alpha = 0.25$, $\beta = 2$, and $b_0 = 16$. To implement the Dirichlet boundary condition, we define a boundary matrix $\mathbf{B}$ of the same size as $\mathbf{U}$. Matrix $\mathbf{B}$ is such that all the matrix elements corresponding to spatial points $(x, y) \notin \partial\Omega_{\text{dirichlet}}$ have values 1, while the others have value 0. In each time iteration after we inverse the $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ to get the solution $\mathbf{U}$ and $\mathbf{V}$, we perform element-wise multiplication $\mathbf{B} \odot \mathbf{U}$ and $\mathbf{B} \odot \mathbf{V}$, and then Fourier transform again to get the new $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ for Dirichlet boundary condition. Note that this is not a “pure” Fourier spectral treatment of Dirichlet boundary condition, but a simple and convenient way to mimic the condition. This has some potential consequences such as loss of strict mass conservation and possible artifacts at the boundary region, due to inconsistency between the Fourier series periodicity and zero boundary condition. However, the main goal is to simulate and observe the wave reflections under our numerical scheme. The results are shown in Figure 10-12. We can see a substantial part of the soliton has been reflected against the north wall at $t = 18$, indicated by red color, and at $t = 21$ the whole soliton has been reflected. At $t = 30$, we can see the approximate soliton is now heading

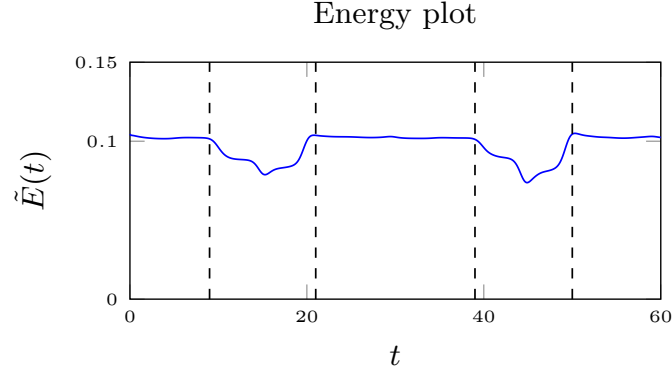


FIGURE 9. The approximate energy plot for the simulation with Dirichlet boundary condition.

south-east, and at $t = 42$ we can see a small part of it has been reflected off the south wall indicated by blue color. To quantitatively see the effects of reflections, we also provide a numerical calculation of the “energy” function $E(t)$ defined by

$$E(t) = \int_{\Omega} u^2 d\Omega,$$

which is also the L^2 -norm of the wave signal. We approximate $E(t)$ for $t \in [0, 60]$ by the following summation

$$\tilde{E}(t) \approx \sum_{(x,y) \in \tilde{\Omega}} U(x, y, t)^2 \Delta x \Delta y,$$

where $\tilde{\Omega}$ consists of all pairs of discrete spatial points used in the simulation. The results are shown in Figure 9. The first two dashed vertical lines represent the approximate time points when the wave first hit the “north wall” boundary (at $t = 9$, see Figure 10) and when the whole soliton has been reflected and travels into the southeast direction (at $t = 21$, see Figure 11), respectively. Similarly, the last two dashed vertical lines represent the approximate time points when the wave first hit the “south wall” boundary (at $t = 39$) and when the whole soliton has been reflected and travels back into the northeast direction (at $t = 50$), respectively. As we can see in Figure 9, during the times when the soliton is reflecting against the walls, the energy is noticeably reduced (see the indented blue curve between the first two dashed lines and between the last two dashed lines). However, during times when the whole soliton is not reflecting and travels only in one direction, the energy is approximately constant.

7. CONCLUSION

We have extended the numerical scheme for one-dimensional “bad” Boussinesq equation introduced in [11] to a numerical scheme for a two-dimensional “bad” Boussinesq equation (2). Our numerical scheme works by using Fourier spectral method and including only certain Fourier modes (removing certain high frequency Fourier modes). The Fourier modes included in the numerical solution have to satisfy the derived condition (8). Under this condition, we also presented an upper bound and a lower bound for the absolute value of Fourier-index j and k , respectively, in Theorem 3.1. The upper bound of $|j|$ depends only on k , while the lower bound of $|k|$ depends only on j .

The implementation of Fourier spectral method is also aided by either Runge-Kutta fourth order or Strang splitting numerical methods for solving differential equations. In a temporal convergence test, we found that the Runge-Kutta fourth order method has smaller L^∞ error than the Strang splitting method. It appears that the difference is more noticeable when Δt is larger (see Figure 5 and 6). Although in this case Runge-Kutta fourth order is seen to be more accurate and simpler than Strang splitting, the latter is an operator splitting method that allows us to treat the linear and nonlinear parts separately. This separation of treatment may be useful for more intricate equations. For both solvers, we found that the numerical solution can be stable up to $t = 100$ (see Figure 8). However, under the same initial condition and discretization settings, minor violation of the “trimming” condition (8) can result in a numerical solution that blows up as early as $t = 23.5$ (see Figure 7).

For future studies, one can also address the standard aliasing issues that appear when using Fourier spectral method on nonlinear terms [18, 19]. Moreover, one can also conduct a rigorous mathematical analysis for the existence of blow-up solutions and other aspects of the two-dimensional “bad” Boussinesq equation (2), similar to what has been done in [5, 6, 7, 8].

Declarations. The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Contributions. The single author wrote the entire manuscript.

Funding Statement. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgment. The author would like to thank Christophe Charlier from Université catholique de Louvain for constructive feedbacks on the manuscript.

The author would like to thank the anonymous reviewers for providing constructive comments on the manuscript.

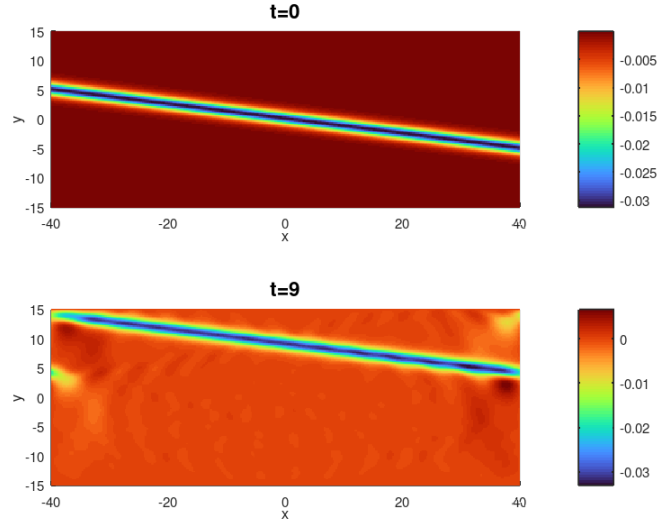


FIGURE 10. Numerical solution at time $t = 0, t = 9$ with Dirichlet boundary condition.

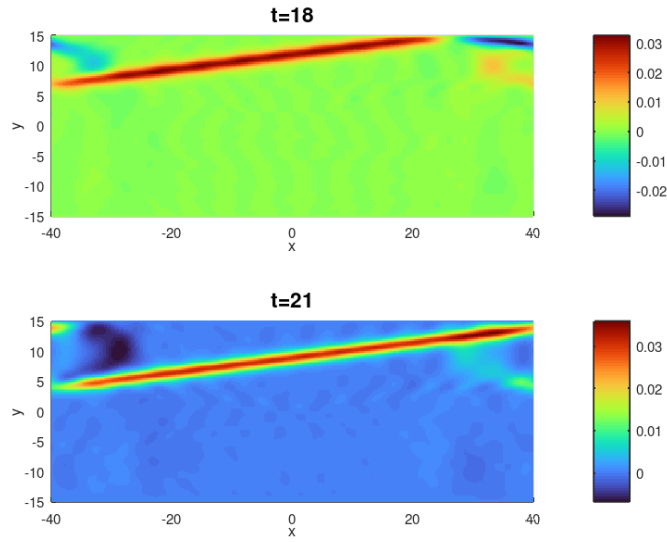


FIGURE 11. Numerical solution at time $t = 18, t = 21$ with Dirichlet boundary condition.

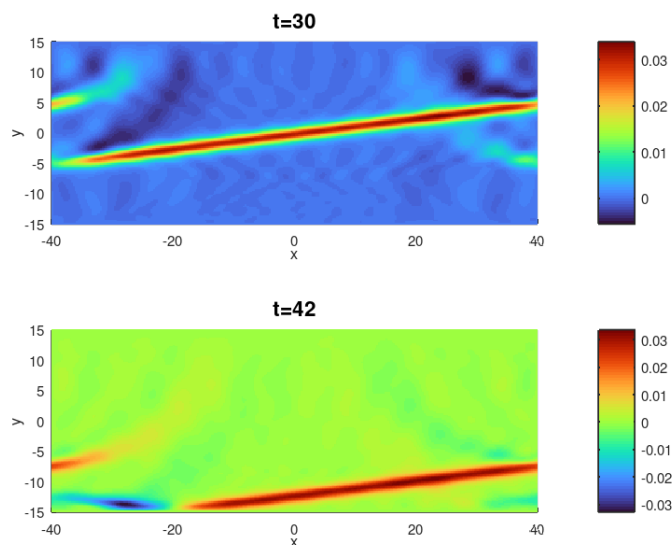


FIGURE 12. Numerical solution at time $t = 30$, $t = 42$ with Dirichlet boundary condition.

REFERENCES

- [1] J. Boussinesq, “Theorie des ondes et des remous qui se propagent le long d’un canal rectangulaire horizontal, en communiquant au liquide contenu dans ce canal des vitesses sensiblement pareilles de la surface au fond,” *J. Math. Pures Appl.*, vol. 17, pp. 55–108, 1872. https://www.numdam.org/item/JMPA_1872_2_17__55_0.pdf.
- [2] J. Hadamard, *Lectures on Cauchy’s Problem in Linear Partial Differential Equations*. Yale Univ. Press, New Haven-London, 1923.
- [3] C. Charlier and J. Lenells, “On boussinesq’s equation for water waves,” 2022. <https://arxiv.org/abs/2204.02365>.
- [4] C. Charlier, “Blow-up solutions of the “bad” boussinesq equation,” 2024. <https://arxiv.org/abs/2405.12210>.
- [5] C. Charlier and J. Lenells, “Boussinesq’s equation for water waves: Asymptotics in sector i,” *Advances in Nonlinear Analysis*, vol. 13, no. 1, 2024. <https://doi.org/10.1515/anona-2024-0022>.
- [6] C. Charlier and J. Lenells, “Boussinesq’s equation for water waves: Asymptotics in sector v,” *SIAM Journal on Mathematical Analysis*, vol. 56, no. 3, pp. 4104–4142, 2024. <https://doi.org/10.1137/23M1587671>.
- [7] C. Charlier and J. Lenells, “The soliton resolution conjecture for the boussinesq equation,” *Journal de Mathématiques Pures et Appliquées*, vol. 191, p. 103621, 2024. <https://www.sciencedirect.com/science/article/pii/S0021782424001193>.
- [8] C. Charlier and J. Lenells, “Boussinesq’s equation for water waves: the soliton resolution conjecture for sector iv,” *Advanced Nonlinear Studies*, vol. 25, no. 1, pp. 106–151, 2025. <https://doi.org/10.1515/ans-2023-0154>.

- [9] M. S. Ismail and F. Mosally, “A fourth order finite difference method for the good boussinesq equation,” *Abstract and Applied Analysis*, vol. 2014. <https://doi.org/10.1155/2014/323260>.
- [10] C. Zhang et al., “A second order operator splitting numerical scheme for the “good” boussinesq equation,” *Applied Numerical Mathematics*, vol. 119, pp. 179–193, 2017. <https://doi.org/10.1016/j.apnum.2017.04.006>.
- [11] C. Charlier et al., “Numerical scheme for the solution of the “bad” boussinesq equation,” *Applied Numerical Mathematics*, vol. 217, 2025. <https://doi.org/10.1016/j.apnum.2025.06.011>.
- [12] R. S. Johnson, “A two-dimensional boussinesq equation for water waves and some of its solutions,” *Journal of Fluid Mechanics*, vol. 323, pp. 65–78, 1996. <https://doi.org/10.1017/S0022112096000845>.
- [13] S. A. Orszag, “Comparison of pseudospectral and spectral approximation,” *Studies in Applied Mathematics*, vol. 51, no. 3, 1972. <https://doi.org/10.1002/sapm1972513253>.
- [14] F. P. Barrera et al., “Exact solutions of the two-dimensional boussinesq and dispersive water waves equations,” *WIT Transactions on Engineering Sciences*, vol. 69, pp. 293–297, 2010. <https://doi.org/10.2495/AFM100251>.
- [15] H. Holden et al., “Operator splitting for the kdv equation,” *Mathematics of Computation*, vol. 80, pp. 821–846, 2011. <https://doi.org/10.1090/S0025-5718-2010-02402-0>.
- [16] E. W. Weisstein, “Runge-kutta method.” <https://mathworld.wolfram.com/Runge-KuttaMethod.html>, accessed 2026-09-11.
- [17] H. Holden et al., *Splitting Methods for Partial Differential Equations with Rough Solutions*. European Mathematical Society, 2010. <https://doi.org/10.4171/078>.
- [18] S. A. Orszag, “On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components,” *Journal of the Atmospheric Sciences*, vol. 28, 1971. [https://doi.org/10.1175/1520-0469\(1971\)028<1074:OTE0AI>2.0.CO;2](https://doi.org/10.1175/1520-0469(1971)028<1074:OTE0AI>2.0.CO;2).
- [19] D. Lindblad et al., “Minimizing aliasing in multiple frequency harmonic balance computations,” *Journal of Scientific Computing*, vol. 91, 2022. <https://doi.org/10.1007/s10915-022-01776-0>.