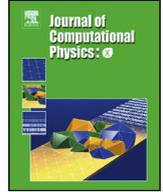


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Journal of Computational Physics: X

journal homepage: www.elsevier.com/locate/jcpx

A conformal mapping approach to modelling two-dimensional stratified flow

Heidi J. Dritschel^{a,b}, David G. Dritschel^{a,*}, Magda Carr^b^a *Mathematical Institute, University of St Andrews, St Andrews, KY16 9SS, United Kingdom*^b *School of Mathematics, Statistics and Physics, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom*

ARTICLE INFO

Article history:

Received 19 October 2022

Received in revised form 17 April 2023

Accepted 25 May 2023

Available online 2 June 2023

Keywords:

PIC

Contour advection

Conformal transformation

Lagrangian methods

Density stratified flows

Turbulent flows

ABSTRACT

Herein we describe a new approach to modelling inviscid two-dimensional stratified flows in a general domain. The approach makes use of a conformal map of the domain to a rectangle. In this transformed domain, the equations of motion are largely unaltered, and in particular Laplace's equation remains unchanged. This enables one to construct exact solutions to Laplace's equation and thereby enforce all boundary conditions.

An example is provided for two-dimensional flow under the Boussinesq approximation, though the approach is much more general (albeit restricted to two-dimensions). This example is motivated by flow under a weir (in a tidal estuary). Here, we discuss how to use a dynamically-evolving conformal map to model changes in the water height on either side of the weir, though the example presented keeps these heights fixed due to limitations in the computational speed to generate the conformal map.

The numerical approach makes use of contour advection, wherein material buoyancy contours are advected conservatively by the local fluid velocity, while a dual contour-grid representation is used for the vorticity in order to account for vorticity generation from horizontal buoyancy gradients. This generation is accurately estimated by using the buoyancy contours directly, rather than a gridded version of the buoyancy field. The result is a highly-accurate, efficient numerical method with extremely low levels of numerical damping.

© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Density stratified flows occur widely in the environment, particularly in the Earth's atmosphere and oceans but also in other planetary atmospheres [20,16]. Stratification provides a natural restoring force, tending to relax flows back to a stable hydrostatic equilibrium. Nonetheless, there are many factors which may disrupt this equilibrium, including surface forcing in the oceans, tides over topography, etc. Such factors commonly result in long-lived propagating internal waves (rather than decay to hydrostatic equilibrium), and these waves can reach large amplitudes, up to hundreds of metres in the oceans (see e.g. Ostrovsky and Stepanyants [15], Helfrich and Melville [12]). They impact natural and man-made submarine structures and contribute significantly to the mixing of heat, salt, chemical and biological constituents.

Over the past few decades, there has been much research conducted to investigate the characteristics, stability, evolution and mixing of internal waves (see Carr et al. [3] and references therein). Experimental, theoretical and computational

* Corresponding author.

E-mail address: david.dritschel@st-andrews.ac.uk (D.G. Dritschel).

approaches all have contributed to our present knowledge of the dynamics of these waves. The present work focuses on a new computational approach enabling one to simulate such waves, and density stratified flows more generally, in arbitrary two-dimensional domains. Such domains may, for example, include bottom topography to examine how internal waves are generated, transformed or scattered from topographic features, a process thought to be generic in the world's oceans (see e.g. Trulsen et al. [19]).

The new approach described in this paper generalises a previous one used to study internal waves in rectangular domains (periodic or aperiodic) [14]. Both approaches use the same underlying method, 'contour advection' [5,6], in which the prognostic fields of density (or buoyancy) and vorticity are represented by a set of material contours, with additional auxiliary gridded fields used for vorticity to allow for vorticity generation from density gradients. The advecting velocity field is obtained by inversion of the vorticity field on a grid (here by spectral methods) after the vorticity contours are converted to gridded values by a fast-fill algorithm developed in Dritschel and Ambaum [5]. Contour advection allows one to resolve a very wide range of scales efficiently and accurately, as has been shown repeatedly (see e.g. Dritschel and Tobias [7] and references therein), and in the present context has led to new insights into the complex nonlinear processes involved in internal wave breaking [3].

The generalisation to arbitrary two-dimensional domains makes use of a conformal map to a rectangle, wherein the equations are solved in transformed coordinates. Remarkably few changes are required to the basic algorithm, for a static domain, and the efficiency is comparable to that enjoyed by the original algorithm. For a time-dependent domain, additional considerations arise, the most important of which is the need to re-generate the conformal mapping at each time step (or when the domain has changed sufficiently). The efficiency of the algorithm in this case is limited by the cost of re-generating the conformal map, which at present is the dominant cost (we use the SToolbox of Driscoll and Trefethen [4]). Nonetheless, we provide the algorithm since it may become viable if improvements in the conformal mapping can be found.

The structure of the paper is as follows. In section 2, we review the governing flow equations, the changes arising from conformal mapping, and the so-called 'inversion problem' in which one recovers the velocity field from the vorticity field and boundary conditions. We also briefly review the numerical method, which is closely similar to that used in [14]. In section 3, we illustrate an example of an exchange flow, a classical and extensively studied problem in density-stratified flows (see Shin et al. [18] and references therein). Here, we look at the much less well studied effect of the domain geometry, in particular obstacles. To this end, we consider a dam-break separating two regions of fluid of different density in a complex domain containing a weir, a sloping bottom boundary and a flat top boundary having different heights either side of the weir. This domain is relevant to investigating tidal flow in an estuary, though one would have to go further and consider time-dependent boundaries with inflow/outflow conditions at the sea and river 'edges'. There are many, potential applications of this method that relate specifically to stratified flows within complex domains. For example, oceanic flow over shelves and slopes [2], topographic effects on exchange flows and gravity currents [2], vertical shear processes in river plumes [1], vortex motion within a circular bay [17], urban atmospheres in complex terrain [8], estuarine circulation [11] and subglacial plumes [13], to name just a few. Further discussion may be found in the conclusions in section 4.

2. Methodology

We consider a two-dimensional stratified flow under the Boussinesq approximation. Taking x to be the horizontal coordinate and y to be the vertical coordinate with $\mathbf{x} = (x, y)$, the flow is governed by the following system of equations in the physical domain (see e.g. Fig. 1):

$$\begin{aligned} \frac{D\zeta}{Dt} &= \frac{\partial b}{\partial x} \\ \frac{Db}{Dt} &= 0 \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \tag{1}$$

where $b(\mathbf{x}, t)$ is the buoyancy, $\zeta(\mathbf{x}, t)$ is the vorticity component pointing out of the plane of view, $\mathbf{u} = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ is the two-dimensional velocity field, and $D/Dt = \partial/\partial t + \mathbf{u} \cdot \nabla$ is the material derivative (see e.g. King et al. [14] and references therein). The vorticity equation comes from taking the curl of the momentum equations (thereby eliminating pressure), while the buoyancy equation expresses conservation of mass in an incompressible fluid for which $\nabla \cdot \mathbf{u} = 0$.

The velocity field may be found in terms of a scalar streamfunction ψ via

$$u = -\frac{\partial \psi}{\partial y}, \quad v = \frac{\partial \psi}{\partial x}, \tag{2}$$

which then automatically ensures $\nabla \cdot \mathbf{u} = 0$. Then using the definition of vorticity $\zeta = \partial v/\partial x - \partial u/\partial y$, we arrive at Poisson's equation for ψ :

$$\nabla^2 \psi = \zeta. \tag{3}$$

This may be solved once we specify the boundary values $\psi = \bar{\psi}$, which are typically zero for a static, closed domain.

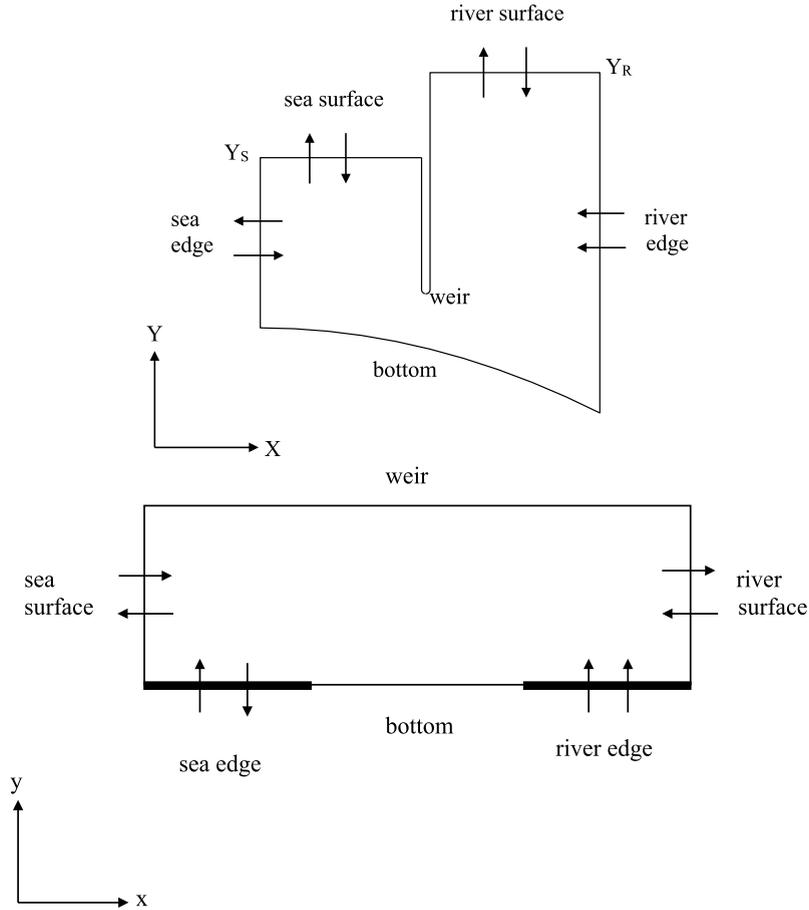


Fig. 1. Schematic of the general domain under consideration (top) and the corresponding conformal domain (bottom).

Below, we describe how to solve these equations after a conformal mapping. We consider first the simpler static domain case, then discuss the new features required for a moving domain.

2.1. Static domain

To simplify notation in the remainder of this paper, we henceforth use (X, Y) to denote a point in the physical domain, and (x, y) to denote a point in the conformal domain. The conformal domain is taken to be a rectangle as in Fig. 1, and without loss of generality we let $0 \leq x \leq w_x$ and $0 \leq y \leq w_y$. Other conformal domain shapes can be chosen (e.g. a circle), but a rectangle is convenient for the present purposes.

2.1.1. Preliminaries

A conformal map consists of the relations

$$X(x, y), \quad Y(x, y) \tag{4}$$

giving a point (X, Y) in the physical domain in terms of the conformal coordinates (x, y) . The map must satisfy the Cauchy relations

$$\frac{\partial X}{\partial x} = \frac{\partial Y}{\partial y}, \quad \frac{\partial X}{\partial y} = -\frac{\partial Y}{\partial x} \tag{5}$$

and moreover (for consistency) must satisfy Laplace's equation in the conformal domain:

$$\nabla^2 X = 0, \quad \nabla^2 Y = 0. \tag{6}$$

To solve Laplace's equation, normally one specifies boundary values, but here both equations must be satisfied simultaneously. The method of solution depends on whether the physical domain boundary is taken to be a polygon or a piecewise smooth curve. Here, we consider the first case in order to exploit the SToolbox software package in MATLAB [4]. This

package enables one to construct general conformal maps for a wide range of polygonal boundaries. Moreover, it provides the derivatives $\partial X/\partial x$ etc. required to recast the fluid dynamical equations (1)–(3) in conformal coordinates.

The advantage of a conformal coordinate transformation is that the material derivative D/Dt remains unchanged in form, as long as the velocity field is defined by $\mathbf{u} = d\mathbf{x}/dt$ as opposed to $\mathbf{U} = d\mathbf{X}/dt$ in the physical domain. Poisson's equation (3) simply picks up a conformal factor λ :

$$\nabla^2 \psi = \lambda \zeta \tag{7}$$

where

$$\lambda = \|\nabla X\|^2 = \|\nabla Y\|^2 \tag{8}$$

and ∇ is the gradient operator now in the conformal domain. Note: λ integrated over the area of the conformal domain gives the area of the physical domain. The X and Y derivatives of any field f transform as

$$\begin{aligned} \frac{\partial f}{\partial X} &= \frac{1}{\lambda} \left(\frac{\partial f}{\partial x} \frac{\partial Y}{\partial y} - \frac{\partial f}{\partial y} \frac{\partial X}{\partial x} \right), \\ \frac{\partial f}{\partial Y} &= \frac{1}{\lambda} \left(\frac{\partial f}{\partial y} \frac{\partial X}{\partial x} + \frac{\partial f}{\partial x} \frac{\partial Y}{\partial y} \right), \end{aligned} \tag{9}$$

where we have used the chain rule

$$\begin{aligned} \frac{\partial f}{\partial x} &= \frac{\partial f}{\partial X} \frac{\partial X}{\partial x} + \frac{\partial f}{\partial Y} \frac{\partial Y}{\partial x}, \\ \frac{\partial f}{\partial y} &= \frac{\partial f}{\partial X} \frac{\partial X}{\partial y} + \frac{\partial f}{\partial Y} \frac{\partial Y}{\partial y}, \end{aligned} \tag{10}$$

together with the Cauchy relations (5) to solve for $\partial f/\partial X$ and $\partial f/\partial Y$ in (9). Finally, the physical velocity field \mathbf{U} also transforms by the chain rule:

$$\begin{aligned} U &= \frac{dX}{dt} = \frac{\partial X}{\partial x} \frac{dx}{dt} + \frac{\partial X}{\partial y} \frac{dy}{dt} = \frac{\partial Y}{\partial y} u - \frac{\partial Y}{\partial x} v, \\ V &= \frac{dY}{dt} = \frac{\partial Y}{\partial x} \frac{dx}{dt} + \frac{\partial Y}{\partial y} \frac{dy}{dt} = \frac{\partial Y}{\partial x} u + \frac{\partial Y}{\partial y} v. \end{aligned} \tag{11}$$

Using (9) and (11) to transform the incompressibility relations (2), we obtain simply

$$u = -\frac{1}{\lambda} \frac{\partial \psi}{\partial y}, \quad v = \frac{1}{\lambda} \frac{\partial \psi}{\partial x}. \tag{12}$$

Note: $\nabla \cdot \mathbf{u} \neq 0$; the conformal transformation does not preserve incompressibility (or local area).

2.1.2. Velocity inversion

The core part of the numerical method consists of solving Poisson's equation (7), needed to obtain the velocity field \mathbf{u} from (12). The rectangular shape of the domain allows one to exploit accurate and efficient spectral methods (Fast Fourier Transforms or FFTs) for this purpose. Considering a grid with n_x and n_y divisions in x and y , fields are expanded either as a finite cosine or a sine series in x and y , with discrete wavenumbers $k_m = m\pi/w_x$ and $l_n = n\pi/w_y$ for integers $m \in [0, n_x]$ and $n \in [0, n_y]$.

While in a static domain it is sufficient to impose $\psi = 0$ on the boundaries, we here consider the more general situation in which $\psi = \bar{\psi}$ on the boundaries, which is relevant to the moving domain case with possible inflow and outflow conditions discussed in the following section.

As ζ and λ in (7) can in general have arbitrary boundary values, the source term $S = \lambda \zeta$ is expanded in a cosine series in x and in y . We therefore seek a streamfunction ψ of the form

$$\psi = \psi_p + \psi_c + \psi_h$$

where the components ψ_p , ψ_c and ψ_h satisfy the following conditions:

- $\nabla^2 \psi_p = S$ and ψ_p has the same cosine series as S but generally does not satisfy $\psi_p = \bar{\psi}$ on the boundaries;
- $\nabla^2 \psi_c = 0$ and $\psi_c = \psi' \equiv \bar{\psi} - \psi_p$ on all corners, i.e.

$$\psi_c = \psi'_{00}(1-\xi)(1-\eta) + \psi'_{10}\xi(1-\eta) + \psi'_{01}(1-\xi)\eta + \psi'_{11}\xi\eta, \tag{13}$$

where respectively ψ'_{00} , ψ'_{10} , ψ'_{01} and ψ'_{11} are the values of ψ' at the lower-left, lower-right, upper-left and upper-right corners of the conformal domain, and where $\xi = x/w_x$ and $\eta = y/w_y$ are scaled coordinates in the range $[0, 1]$;

- $\nabla^2 \psi_h = 0$ and $\psi_h = \tilde{\psi} \equiv \psi' - \psi_c = \bar{\psi} - \psi_p - \psi_c$ on all edges.

This construction ensures that $\psi = \bar{\psi}$ on all boundaries and is compatible with a Fourier series solution, as explained next.

Since the auxiliary field $\tilde{\psi} = \psi' - \psi_c$ vanishes at each corner by construction, the boundary values of $\tilde{\psi}$ can be represented as a sine series (in either x or y) on each edge. Specifically,

$$\begin{aligned} \tilde{\psi}(x, 0) &= \sum_{m=1}^{n_x-1} A_m^- \sin(k_m x), & \tilde{\psi}(x, w_y) &= \sum_{m=1}^{n_x-1} A_m^+ \sin(k_m x) \\ \tilde{\psi}(0, y) &= \sum_{n=1}^{n_y-1} B_n^- \sin(l_n y), & \tilde{\psi}(w_x, y) &= \sum_{n=1}^{n_y-1} B_n^+ \sin(l_n y), \end{aligned} \tag{14}$$

where the coefficients A_m^\pm and B_n^\pm are in practice determined by FFTs. The Nyquist frequency $m = n_x$ or $n = n_y$ is absent because the sine function is zero at all grid points. Consider then the following proposed solution for ψ_h :

$$\begin{aligned} \psi_h &= \sum_{m=1}^{n_x-1} (A_m^- f_m^-(\eta) + A_m^+ f_m^+(\eta)) \sin(k_m x) \\ &+ \sum_{n=1}^{n_y-1} (B_n^- g_n^-(\xi) + B_n^+ g_n^+(\xi)) \sin(l_n y), \end{aligned} \tag{15}$$

where

$$\begin{aligned} f_m^- &= \frac{\sinh \alpha_m (1 - \eta)}{\sinh \alpha_m}, & f_m^+ &= \frac{\sinh \alpha_m \eta}{\sinh \alpha_m} \\ g_n^- &= \frac{\sinh \beta_n (1 - \xi)}{\sinh \beta_n}, & g_n^+ &= \frac{\sinh \beta_n \xi}{\sinh \beta_n} \end{aligned} \tag{16}$$

with $\alpha_m = k_m w_y$ and $\beta_n = l_n w_x$. Each term in both sums in (15) is a solution to Laplace's equation, and hence $\nabla^2 \psi_h = 0$ as required. Moreover, the functions f_m^\pm and g_n^\pm are either 0 or 1 on the boundaries, and it is simple to check that $\psi_h = \tilde{\psi}$ on all edges, as required. Thus the combined solution $\psi = \psi_p + \psi_c + \psi_h$ satisfies $\nabla^2 \psi = S$ together with the required boundary conditions.

The velocity field \mathbf{u} is found by differentiating ψ in (12):

$$\mathbf{u} = -\frac{1}{\lambda} \left(\frac{\partial \psi_p}{\partial y} + \frac{\partial \psi_c}{\partial y} + \frac{\partial \psi_h}{\partial y} \right), \quad \mathbf{v} = \frac{1}{\lambda} \left(\frac{\partial \psi_p}{\partial x} + \frac{\partial \psi_c}{\partial x} + \frac{\partial \psi_h}{\partial x} \right) \tag{17}$$

where the partial derivatives of ψ_p are evaluated in spectral space by wavenumber multiplication followed by inverse FFTs (taking into account the appropriate changes from cosine to sine series), while the partial derivatives of ψ_c are computed analytically as

$$\begin{aligned} \frac{\partial \psi_c}{\partial x} &= \frac{\psi'_{11} - \psi'_{01}}{w_x} \eta + \frac{\psi'_{10} - \psi'_{00}}{w_x} (1 - \eta), \\ \frac{\partial \psi_c}{\partial y} &= \frac{\psi'_{11} - \psi'_{10}}{w_y} \xi + \frac{\psi'_{01} - \psi'_{00}}{w_y} (1 - \xi), \end{aligned} \tag{18}$$

and finally those of ψ_h are computed semi-analytically followed by an appropriate inverse FFT:

$$\begin{aligned} \frac{\partial \psi_h}{\partial x} &= \sum_{m=1}^{n_x-1} (A_m^- f_m^-(\eta) + A_m^+ f_m^+(\eta)) k_m \cos(k_m x) \\ &+ \frac{1}{w_x} \sum_{n=1}^{n_y-1} \left(B_n^- \frac{d g_n^-}{d \xi}(\xi) + B_n^+ \frac{d g_n^+}{d \xi}(\xi) \right) \sin(l_n y), \end{aligned} \tag{19}$$

$$\begin{aligned} \frac{\partial \psi_h}{\partial y} &= \frac{1}{w_y} \sum_{m=1}^{n_x-1} \left(A_m^- \frac{d f_m^-}{d \eta}(\eta) + A_m^+ \frac{d f_m^+}{d \eta}(\eta) \right) \sin(k_m x) \\ &+ \sum_{n=1}^{n_y-1} (B_n^- g_n^-(\xi) + B_n^+ g_n^+(\xi)) l_n \cos(l_n y), \end{aligned} \tag{20}$$

where

$$\begin{aligned} \frac{1}{w_y} \frac{df_m^-}{d\eta} &= -k_m \frac{\cosh(\alpha_m(1-\eta))}{\sinh \alpha_m}, & \frac{1}{w_y} \frac{df_m^+}{d\eta} &= k_m \frac{\cosh(\alpha_m\eta)}{\sinh \alpha_m} \\ \frac{1}{w_x} \frac{dg_n^-}{d\xi} &= -l_n \frac{\cosh(\beta_n(1-\xi))}{\sinh \beta_n}, & \frac{1}{w_x} \frac{dg_n^+}{d\xi} &= l_n \frac{\cosh(\beta_n\xi)}{\sinh \beta_n}. \end{aligned} \tag{21}$$

2.2. Moving domain

We next extend the method described in the previous subsection to allow the shape of the physical domain to vary in time and to allow general inflow/outflow boundary conditions. This permits one to study tides in a river estuary, including changing upstream flow conditions.

Here we restrict attention to the case where the river and sea surfaces, at $Y = Y_R(t)$ and $Y = Y_S(t)$ respectively (see Fig. 1, top panel), are the only parts of the domain which are varying in time t .

We generalise the conformal map (4) to

$$X(\hat{x}, y, t), \quad Y(\hat{x}, y, t) \tag{22}$$

where \hat{x} is a scaled x coordinate defined by $\hat{x} = \kappa(t)x$ where $\kappa(t) \equiv w_{x0}/w_x(t)$ is a dilation factor and w_{x0} is a reference conformal domain length in x . The reason for this construction is that, in the conformal mapping solution, w_x varies with the shape of the physical domain, whereas w_y does not. Moreover, it is convenient for imposing boundary conditions to use a static, modified, conformal domain in \hat{x}, y coordinates (see below).

The objective is to obtain the velocity $\hat{u} = d\hat{x}/dt$, $v = dy/dt$ in the scaled conformal domain in terms of the streamfunction ψ . To this end, we start with the definition of the velocity in the physical domain and use the chain rule:

$$\begin{aligned} U &= \frac{dX}{dt} = \frac{\partial X}{\partial \hat{x}} \frac{d\hat{x}}{dt} + \frac{\partial X}{\partial y} \frac{dy}{dt} + \frac{\partial X}{\partial t} \\ V &= \frac{dY}{dt} = \frac{\partial Y}{\partial \hat{x}} \frac{d\hat{x}}{dt} + \frac{\partial Y}{\partial y} \frac{dy}{dt} + \frac{\partial Y}{\partial t} \end{aligned} \tag{23}$$

where new terms appear due to the time-dependent map. Next using

$$\frac{\partial X}{\partial x} = \frac{\partial X}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} = \kappa \frac{\partial X}{\partial \hat{x}} \quad \& \quad \frac{\partial Y}{\partial x} = \frac{\partial Y}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} = \kappa \frac{\partial Y}{\partial \hat{x}}$$

together with the Cauchy relations (5), we obtain

$$\begin{aligned} U - \frac{\partial X}{\partial t} &= \frac{\partial Y}{\partial y} \frac{\hat{u}}{\kappa} - \frac{\partial Y}{\partial x} v \\ V - \frac{\partial Y}{\partial t} &= \frac{\partial Y}{\partial x} \frac{\hat{u}}{\kappa} + \frac{\partial Y}{\partial y} v. \end{aligned} \tag{24}$$

Inverting these for $\hat{u} = d\hat{x}/dt$ and $v = dy/dt$, we find

$$\begin{aligned} \lambda \frac{\hat{u}}{\kappa} &= \left(V - \frac{\partial Y}{\partial t} \right) \frac{\partial Y}{\partial x} + \left(U - \frac{\partial X}{\partial t} \right) \frac{\partial Y}{\partial y} \\ \lambda v &= \left(V - \frac{\partial Y}{\partial t} \right) \frac{\partial Y}{\partial y} - \left(U - \frac{\partial X}{\partial t} \right) \frac{\partial Y}{\partial x} \end{aligned} \tag{25}$$

where $\lambda = \|\nabla Y\|^2$ is the conformal factor as before.

Next we consider derivatives of the streamfunction ψ . By the chain rule

$$\begin{aligned} \frac{\partial \psi}{\partial \hat{x}} &= \frac{\partial \psi}{\partial X} \frac{\partial X}{\partial \hat{x}} + \frac{\partial \psi}{\partial Y} \frac{\partial Y}{\partial \hat{x}} \\ \Rightarrow \frac{\partial \psi}{\partial x} &= \frac{\partial \psi}{\partial X} \frac{\partial X}{\partial x} + \frac{\partial \psi}{\partial Y} \frac{\partial Y}{\partial x} = V \frac{\partial Y}{\partial y} - U \frac{\partial Y}{\partial x}, \end{aligned} \tag{26}$$

and similarly

$$\frac{\partial \psi}{\partial y} = \frac{\partial \psi}{\partial X} \frac{\partial X}{\partial y} + \frac{\partial \psi}{\partial Y} \frac{\partial Y}{\partial y} = -V \frac{\partial Y}{\partial x} - U \frac{\partial Y}{\partial y} \tag{27}$$

where we have used the Cauchy relations and the definitions of U and V . However, note that the groups of terms representing $\partial\psi/\partial x$ and $\partial\psi/\partial y$ appear also in (25). Thus we can simplify the latter to obtain the sought-after solution for \hat{u} and v :

$$\begin{aligned}\hat{u} &= -\frac{\kappa}{\lambda} \left(\frac{\partial\psi}{\partial y} + \frac{\partial Y}{\partial t} \frac{\partial Y}{\partial x} + \frac{\partial X}{\partial t} \frac{\partial Y}{\partial y} \right) \\ v &= \frac{1}{\lambda} \left(\frac{\partial\psi}{\partial x} + \frac{\partial X}{\partial t} \frac{\partial Y}{\partial x} - \frac{\partial Y}{\partial t} \frac{\partial Y}{\partial y} \right).\end{aligned}\quad (28)$$

Comparing with the static domain case (12), besides the scaling factor $\kappa(t)$, new terms arise due to the time-dependent conformal map. These terms ensure that there is no normal flow across any impermeable boundary, i.e. that the normal velocity in the static, modified, conformal domain is identically zero there. To see this, consider the situation depicted in Fig. 1 (top panel) where the sea surface at $Y = Y_S(t)$ and river surface at $Y = Y_R(t)$ may generally vary in time. These boundaries correspond to $\hat{x} = 0$ and $\hat{x} = w_{x0}$ respectively in the (scaled) conformal domain. We next show that \hat{u} vanishes, as required, on these boundaries. First consider the sea surface. There, X does not vary in time while $\partial Y/\partial t = dY_S/dt = V_S(t)$, the upward velocity of the sea surface. Hence, according to (28),

$$\begin{aligned}-\lambda \frac{\hat{u}}{\kappa} &= \frac{\partial\psi}{\partial y} + \frac{\partial Y}{\partial t} \frac{\partial Y}{\partial x} + \frac{\partial X}{\partial t} \frac{\partial Y}{\partial y} \\ &= \frac{\partial\psi}{\partial y} + V_S \frac{\partial Y}{\partial x} \\ &= \frac{\partial\psi}{\partial y} - V_S \frac{\partial X}{\partial y}\end{aligned}\quad (29)$$

using the Cauchy relations. However, along this boundary, ψ must be chosen to be consistent with the velocity boundary condition there, namely $V = \partial\psi/\partial X = V_S(t)$. This implies $\psi = V_S(t)X + C(t)$, where $C(t)$ is an ignorable constant (as it does not alter the velocity field). Hence,

$$\frac{\partial\psi}{\partial y} = V_S \frac{\partial X}{\partial y},$$

and this is seen to exactly compensate the same term in (29), proving that $\hat{u} = 0$ there as required. The same argument can be used to show that $\hat{u} = 0$ along the river surface at $Y = Y_R(t)$.

3. Results

In this section, we perform a series of tests to demonstrate the validity and accuracy of the methods developed in the previous section. The core numerical problem is obtaining the velocity field from a given interior vorticity field and prescribed boundary velocities. We demonstrate that the method is second-order accurate in grid spacing, as expected. We go on to illustrate an example of the flow evolution resulting from a ‘dam-break’, in a static domain. The moving domain case is deferred to a future study due to the significant added complexity associated with the inflow/outflow boundary conditions.

3.1. Numerical tests

We consider the physical domain illustrated in the top panel in Fig. 1. Taking equally-spaced grid-lines in the conformal domain (the rectangle in the bottom panel of Fig. 1), the image in the physical domain is shown in Fig. 2 at coarse resolution.

3.1.1. Inversion in a static domain

A core part of the numerical algorithm, indeed one which requires the greatest computational resources, is determining the velocity field \mathbf{U} from the vorticity field ζ and prescribed boundary conditions, a process called ‘inversion’. In the domain shown in Fig. 2, we take the vorticity field to be the Gaussian function

$$\zeta = e^{-\|\mathbf{x} - \mathbf{x}_0\|^2/R^2}$$

centred on a prescribed point \mathbf{X}_0 and having a characteristic radius R . We take $\mathbf{X}_0 = (1.5, 1.0)$ and $R = 0.5$, though these specific choices do not affect the rates of convergence of our results. This vorticity field is shown in Fig. 3(a).

Consider first a fixed, static domain with streamfunction $\psi = \text{constant}$ ($= 0$ without loss of generality) along the domain edges. Fig. 3 shows the streamfunction and velocity components at the default resolution, $n_x = 400$ and $n_y = 200$. The flow field is consistent with the prescribed boundary conditions, namely that \mathbf{U} is everywhere tangent to the boundary. Notably, the flow field is well behaved near the lower rounded portion of the weir, where the curvature of the boundary is high.

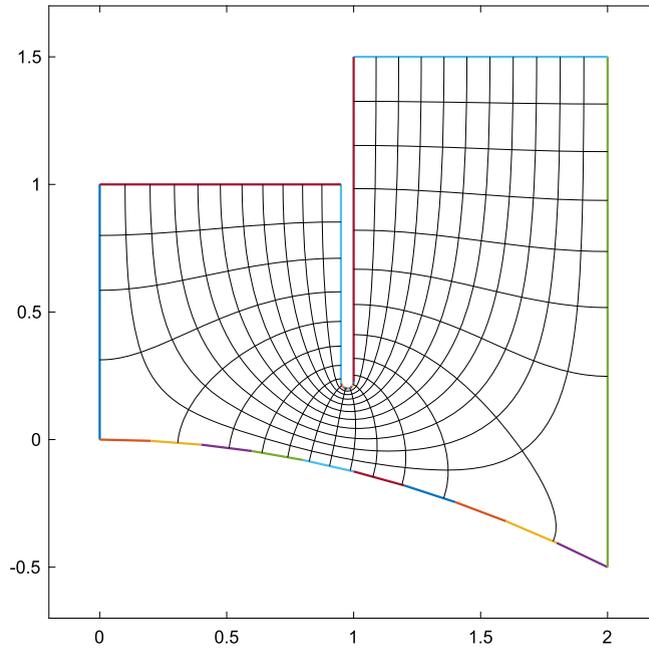


Fig. 2. The image of the conformal map grid-lines in the physical domain depicted in Fig. 1. Note, tests described in this section typically use ten times as many grid-lines in each direction. The colours along the edges of the domain remain constant between specified vertices. Straight lines connect adjacent vertices so that the domain is actually a polygon.

To assess accuracy, we examine next the convergence of the solution with increasing resolution. As an exact solution is not known, we use the solution on a 1600×800 grid as a reference, and compare it to solutions generated on a series of coarser grids having 2^m fewer grid points in each direction for $m = 1, 2$ and 3 . Two measures of error are considered, each for the three fields ψ , u and v . The first measure is the r.m.s. error computed using all common grid points (i.e. every eighth grid point when comparing 200×100 with 1600×800). The second measure is the maximum absolute error over all common grid points.

The results are shown in Fig. 4 in logarithmic scaling. The dashed lines show linear least-squares fits of the errors plotted in solid lines. For ψ , the r.m.s. error decays like $n_x^{-3.07 \pm 0.17}$ while the maximum error decays more slowly, like $n_x^{-1.90 \pm 0.09}$. For u and v , the errors decay like $n_x^{-1.75 \pm 0.04}$ while the maximum error again decays more slowly, like $n_x^{-1.41 \pm 0.17}$. The error in ψ is mainly concentrated in the lower right corner of the domain, where the grid cell size is largest in the physical domain (see Fig. 2), but error also occurs very close to the upper right corners of the domain to the right of the weir, see Fig. 5. The error in u is mainly located at the upper right edge of the domain, while that in v is located at the two right vertical portions of the domain (not shown).

The main conclusion is that the solution converges with increasing resolution. In an r.m.s. measure, the convergence rate is nearly quadratic for the velocity components, and cubic for the streamfunction. Largest errors are found at edges or in regions occupied by the largest grid cells in the physical domain.

3.1.2. Inversion in a moving domain

Consider next inversion in a moving domain where the streamfunction $\bar{\psi}$ may vary along the domain edges. We examine a moment of time when the domain shape is the same as in the previous case. We suppose that there is a uniform flow U_R at the river edge (independent of Y and less than zero if flowing from right to left in Fig. 1). Furthermore the upper surfaces at $Y = Y_R$ and Y_S are moving at speeds V_R and V_S respectively. The flow through the sea edge U_S (also assumed uniform for simplicity) is determined by mass conservation (or volume conservation for the incompressible fluid considered). Let Y_{BR} be the Y coordinate at the bottom of the river edge on the right, and Y_{BS} be the Y coordinate at the bottom of the sea edge on the left. Also, let the river edge be at $X = X_R$ and the sea edge be at $X = X_S$. Finally, let the right (sea) edge of the weir be at $X = X_{WS}$ and the left (river) edge of the weir be at $X = X_{WR}$. Then conservation of volume determines U_S from

$$U_S = \frac{U_R(Y_R - Y_{BR}) + V_R(X_R - X_{WR}) + V_S(X_{WS} - X_S)}{Y_S - Y_{BS}}. \quad (30)$$

These speeds on the various edges are needed to determine $\bar{\psi}$ as a continuous function around the entire domain boundary. Along the (impermeable) bottom of the domain, we take $\bar{\psi} = 0$ without loss of generality. Then along the sea edge, by continuity $\bar{\psi} = U_S(Y_{BS} - Y)$. Going clockwise in Fig. 1, along the sea surface $\bar{\psi} = V_S(X - X_{WS}) + C_W$ where

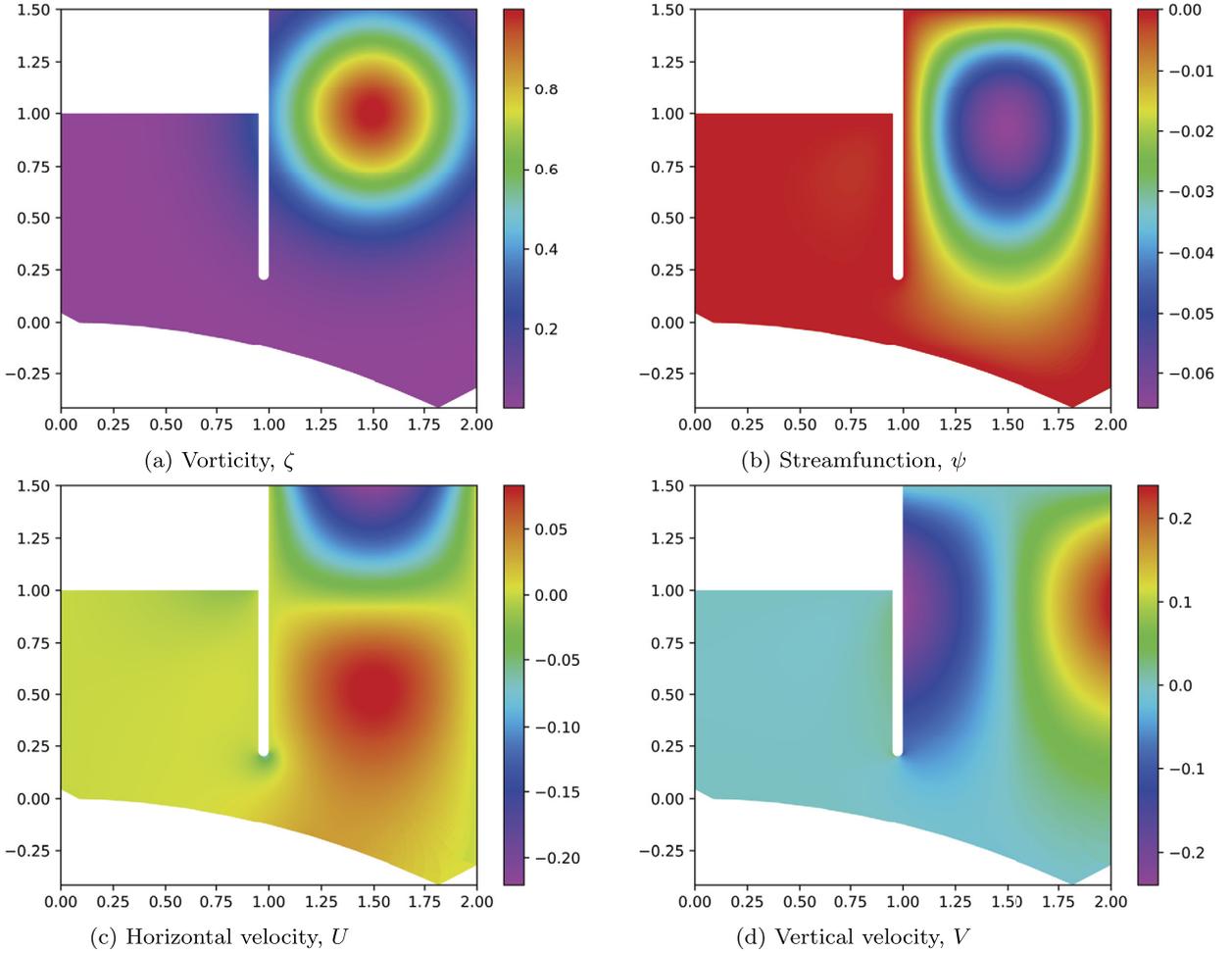


Fig. 3. Various fields as indicated at the default resolution, and found by inverting the vorticity field ζ in panel (a).

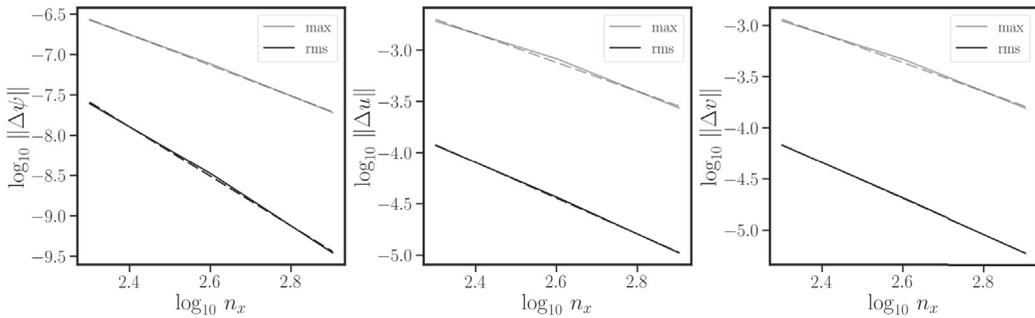


Fig. 4. Errors in streamfunction ψ (left), and the velocity components u (middle) and v (right), together with linear least-square fits in dashed lines. See text for definitions.

$$C_W = -U_R(Y_R - Y_{BR}) - V_R(X_R - X_{WR})$$

is the constant value of $\bar{\psi}$ on the weir (another impermeable boundary). Continuing, on the river surface $\bar{\psi} = V_R(X - X_{WR}) + C_W$. Finally, on the river edge, $\bar{\psi} = U_R(Y_{BR} - Y)$.

Note, we are tacitly assuming that the free surfaces remain flat, free of waves or turbulence (e.g. as induced by a hydraulic jump). This requires sufficiently low flow speeds (small Froude numbers) in practical applications. Otherwise, a two-dimensional model like the one developed would not be suitable, or at least not accurate.

To test the inversion procedure, we choose (arbitrarily) $U_R = -0.09$, $V_R = 0.12$ and $V_S = -0.06$, then determine $U_S \approx -0.1053076$ from (30) and the domain shape parameters $Y_R = 1.5$, $Y_S = 1.0$, $Y_{BR} = -0.3177360$, $Y_{BS} = 0.0447388$, $X_R =$

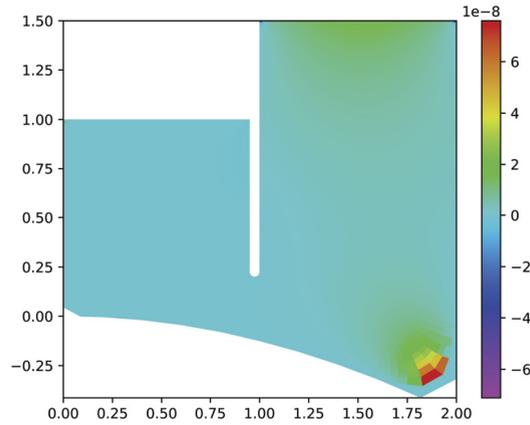


Fig. 5. Difference in the streamfunction, $\Delta\psi$, when computed on the 400×200 grid and the finest 1600×800 grid (plotted at common grid points). A similar spatial pattern is found when comparing other grid resolutions.

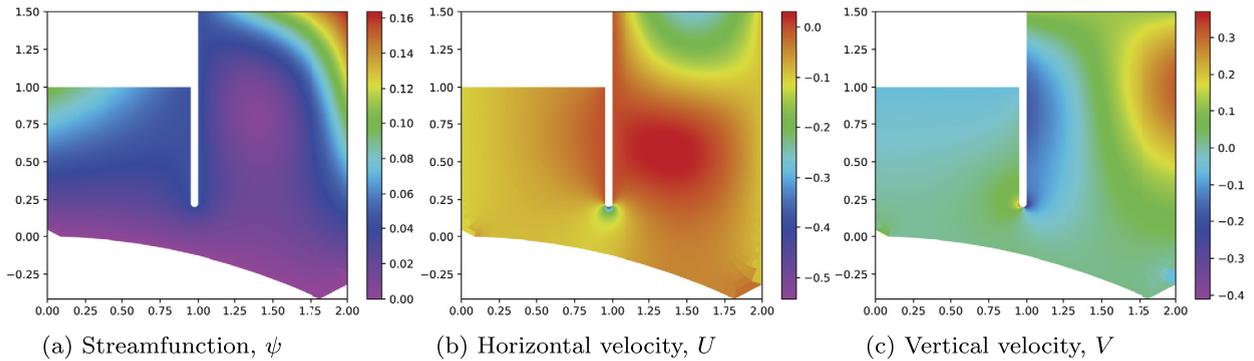


Fig. 6. Various fields as indicated at the default resolution, and found by inverting the vorticity field ζ in Fig. 3(a), here in a moving domain.

2.0 , $X_S = 0$, $X = X_{WR} = 1.0$ and $X = X_{WS} = 0.95$. We also take the bottom of the weir at $Y = Y_W = 0.2$ though this has no effect on U_S . The values of Y_{BR} and Y_{BS} are determined from the discrete conformal mapping after assigning preliminary values $Y_{BR} = -0.5$ and $Y_{BS} = 0$. The mapping rounds off the lower left and lower right corners of the domain. The X coordinate along the left, bottom and right edges of the physical domain monotonically increases from X_R to X_S . Along the left edge, X remains very close to X_S until abruptly increasing (by a factor of about 10^6), and the last point for which $X - X_S < 8 \times 10^{-8}$ determines Y_{BS} . Likewise, along the right edge (moving downwards), X remains very close to X_R until abruptly decreasing, and the last point for which $X_R - X < 8 \times 10^{-8}$ similarly determines Y_{BR} . Only at the highest resolution 1600×800 does the abrupt decrease in X near the river edge X_R not occur at the same Y location – it occurs one grid point lower. However, to compare results between different resolutions, we need to ensure Y_{BR} and Y_{BS} are identical. This is why we chose the specific tolerance 8×10^{-8} in the above procedure.

Fig. 6 shows the streamfunction and velocity components at the default resolution, $n_x = 400$ and $n_y = 200$, for the same vorticity field shown in Fig. 3(a). Now, where fluid is moving across a boundary, the computed solution matches the prescribed boundary velocities to high accuracy (with errors of $\mathcal{O}(10^{-7})$ in U_R , U_S , V_R and V_S). The small errors come from using a finite Fourier series (finite resolution) to represent ψ . Elsewhere, the fluid velocity is tangent to the boundaries, as required. Note that relatively high flow speeds occur near the lower part of the weir where the flow rapidly changes direction to remain tangent to the boundary. If the weir had higher curvature at its lower end, flow speeds would be correspondingly larger, possibly leading to flow separation in reality (i.e. with weak viscosity).

To assess accuracy, we examine next the convergence of the solution with increasing resolution, comparing as before with the solution on the highest resolution (1600×800) grid. The results are shown in Fig. 7 in logarithmic scaling. The dashed lines show linear least-squares fits of the errors plotted in solid lines. For ψ , the r.m.s. error decays like $n_x^{-2.00 \pm 0.17}$ while the maximum error decays more slowly, like $n_x^{-1.17 \pm 0.20}$. For u and v , the errors decay like $n_x^{-1.25 \pm 0.07}$ and $n_x^{-1.60 \pm 0.10}$ while the maximum error again decays more slowly, like $n_x^{-0.264 \pm 0.002}$ and $n_x^{-0.72 \pm 0.22}$. The convergence is not as rapid as in the static domain case, cf. Fig. 4. This is likely due to the discontinuity in velocity along the lower edge of the conformal domain shown in Fig. 1, as well as in the lower left and right hand corners.

This is confirmed in Fig. 8, which shows the differences in ψ , U and V between the solutions on the 400×200 and 1600×800 grids. The errors are mainly located in the lower left and right corners of the domain. The error in ψ vanishes at the boundaries, as enforced by the solution procedure. The error in U has a stippled pattern, likely Gibbs fringes associated

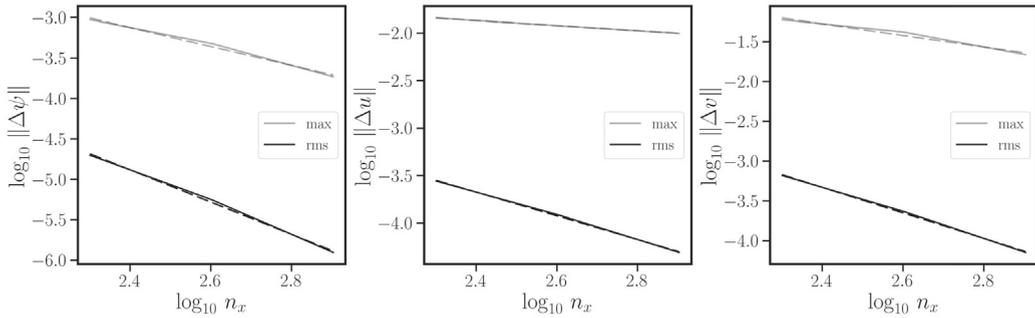


Fig. 7. Errors in streamfunction ψ (left), and the velocity components u (middle) and v (right), together with linear least-square fits in dashed lines. See text for definitions.

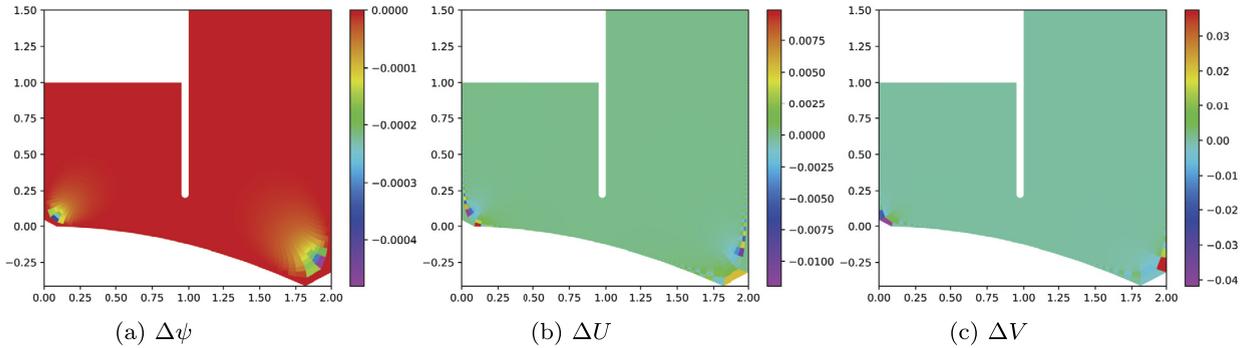


Fig. 8. Differences in ψ , U and V between the 400×200 and 1600×800 grids (plotted at common grid points).

with the discontinuity in U at the corners of the domain. The error in V is more concentrated near the corners and of larger magnitude (four times greater than U).

Nevertheless, the solution converges with increasing resolution, albeit more slowly due to the inflow/outflow boundary conditions. In an r.m.s. measure, the convergence rate is between linear and quadratic for the velocity components, and quadratic for the streamfunction. Largest errors occur near the corners of the domain where the velocity components are discontinuous in the conformal domain.

3.1.3. Contour advection

In the static domain case, the flow evolution is computed in the closed rectangular conformal domain exactly as described in Carr et al. [3], section 3.3. Both vorticity ζ and buoyancy b are represented by a set of *contours* which are advected by the velocity field \mathbf{u} , while ζ is additionally represented by two residual gridded fields designed to uptake the vorticity source $\partial b/\partial x$ in (1) and improve total energy conservation – full details of the method may be found in Dritschel and Fontane [6]. Contour advection allows one to reach extraordinarily high resolution at modest computational expense, or to achieve accuracy much more efficiently than standard numerical methods such as pseudo-spectral. This is demonstrated in Dritschel and Tobias [7] for 2D magneto-hydrodynamics, a system which shares a similar property that the vorticity source is typically fine-scale.

3.2. Illustration: the dam break problem

We next illustrate the complete method for an initial condition consisting of two regions of different density (buoyancy) either side of a vertical line in a flow at rest. Subsequently, the heavier fluid will slump and try to occupy the lower part of the domain, lowering the potential energy. The kinetic energy will grow to compensate, eventually leading to a complex turbulent flow and strong mixing.

We employ the same irregular domain used throughout this paper. In this domain, which extends from $X = 0$ to $X = 2$, we position the initial density transition line at $X = X_0 = 1.5$. To the left, we take the buoyancy $b = 1$, while to the right we take $b = 0$ (the heavier fluid). A small transition is introduced near $X = 1.5$ of width $w = 0.02$ so that the initial buoyancy has the form

$$b = -0.5 \operatorname{erfc}((X - X_0)/w)$$

where $\operatorname{erfc}(z)$ is the complementary error function. The chosen location of the dam break leads to particularly strong turbulent mixing, and also illustrates how the weir largely confines the turbulence to the right-hand portion of the domain.

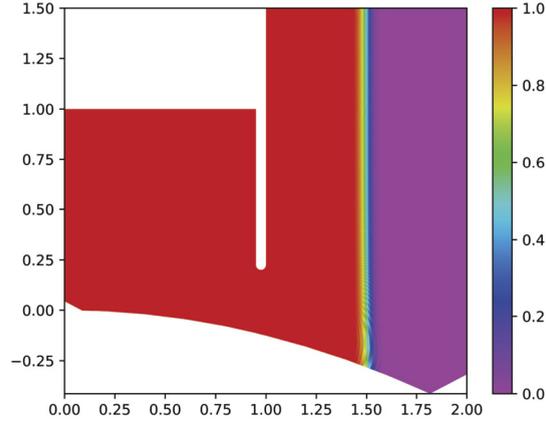


Fig. 9. Initial buoyancy field $b(X, 0)$ used in the dam break simulation. The initial vorticity is identically zero, corresponding to a state of rest.

A dam break positioned under the weir, while perhaps more natural, exhibits much less turbulence. The present situation is a demanding numerical test.

We consider two different resolutions, the default 400×200 , and half resolution 200×100 . The time step Δt is adapted continuously so that

$$\Delta t = \min \left(\frac{\pi}{25|\zeta|_{\max}}, \frac{0.7 \min(\Delta x, \Delta y)}{|\mathbf{u}|_{\max}}, \frac{\min(\Delta x, \Delta y)}{2(b_{\max} - b_{\min})} \right)$$

where Δx and Δy are the grid lengths in the rectangular conformal domain. Above, the number 0.7 is the CFL limit. This ensures both numerical stability for the spectral evolution of vorticity ζ and accuracy in the advection of the b and ζ contours.

Buoyancy contours are equally spaced in b , with 100 contour intervals across the range b_{\min} to b_{\max} (here 0 to 1). This is more than twice the recommended minimum number of intervals [6], but is used here to better represent a continuous variation of b . Vorticity contours are also equally spaced in ζ , but their spacing changes in response to the changing range in ζ (by contrast the range in b never changes as b is materially conserved). We choose the contour interval in ζ from

$$\Delta \zeta = \frac{\sqrt{\langle \zeta^2 \rangle}}{50 \langle |\zeta| \rangle}$$

where $\langle f \rangle$ denotes the domain average of f . In this way, $\Delta \zeta$ increases as the flow develops localised high vorticity values, but the increase is slower than when taking the extreme values of ζ . This choice of $\Delta \zeta$ allows the number of vorticity contour levels to grow to adapt to increasing flow complexity, thus maintaining an adequate resolution of ζ at all times.

Following Carr et al. [3], the same form and magnitude of hyperviscosity is applied to the vorticity equation, to control the inevitable cascade of vorticity to small scales. Without some sort of damping, vorticity can accumulate at small scales, whereas ideally vorticity should cascade to ever smaller scales, below those resolved and be removed. Hyperviscosity is not benign, but it is necessary to prevent this accumulation. (Notably, no damping of any kind is applied to buoyancy.) Hyperviscosity is included by adding a term $\nu_h (-\nabla^2)^p \zeta$ to the right hand side of the vorticity equation in (1). Here, as in Carr et al. [3], we take $p = 3$ and take the hyperviscosity coefficient ν_h proportional to the instantaneous maximum value of $|\zeta|$, specifically $\nu_h = C_h |\zeta|_{\max} / k_{\max}^{2p}$ where k_{\max} is the maximum wavenumber magnitude in the conformal domain, and $C_h = 80$. In spectral space, hyperviscous damping is simply incorporated in the time integration by dividing the undamped spectral tendency by the factor $1 + \nu_h k^{2p} \Delta t / 2$.

We turn next to the results. Fig. 10 shows the fields of buoyancy b and vorticity ζ at a few selected times, for a simulation conducted at the default resolution 400×200 . Note that the effective resolution is 16 times finer in each direction [6]. At early times, the heavier fluid slumps below the lighter fluid, creating a strong shear layer (high vorticity) within the region of strong buoyancy gradients. Recall that vorticity is generated by horizontal buoyancy gradients, see (1). This shear layer then destabilises despite being stretched, rolling up into a series of Kelvin-Helmholtz billows. These interact with each other and with the domain boundaries, and rapidly grow in complexity. This causes the two regions of different buoyancy (density) to partially mix. However, this mixing is largely confined to the region to the right of the weir. Note that the vorticity spreads over an increasing area of the domain, in response to the buoyancy mixing. Maximum vorticity values grow significantly, reaching over 200 in magnitude by the last time shown. (On a coarser grid of dimensions 200×100 , the maximum vorticity magnitude reaches approximately 100.)

Fig. 11 shows the evolution of the energy components, kinetic (red) and potential (blue), together with their sum (black). The dashed curves show results for a coarser 200×100 simulation. Initially, the potential energy is taken to be zero without loss of generality. At first, this decreases sharply as the heavier fluid rushes under the lighter fluid: the centre of mass

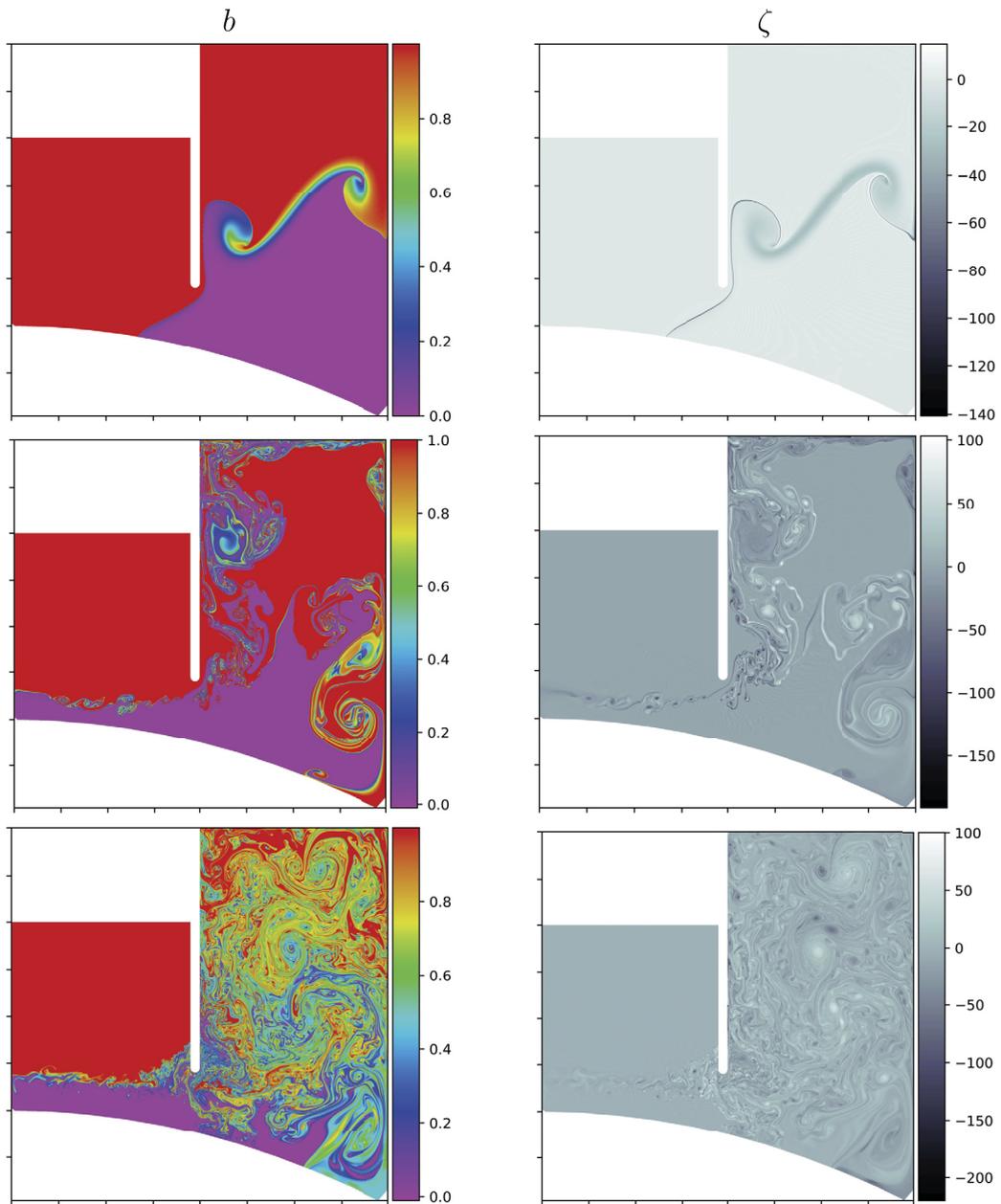


Fig. 10. Buoyancy b (left column) and vorticity ζ (right column) at times $t = 2, 5$ and 10 (top, middle and lower rows) in the dam break simulation conducted on a 400×200 inversion grid. The initial conditions for b are shown in Fig. 9. Note, the fields here are shown on the ultra-fine grid (6400×3200) mapped to the physical domain.

lowers. Then there is a rebound and oscillation around the hydrostatic equilibrium configuration, in which the region of heavier fluid lies below the lighter one. The kinetic energy exhibits the opposite behaviour, consistent with conservation of total energy. Some energy is dissipated here due to the intense small-scale mixing, a generic feature of density-stratified turbulent flows. Notably, the results depend only weakly on resolution, especially at early times ($t < 3$).

Fig. 12 shows the evolution of two buoyancy norms, the linear norm $\|b\|$ (red) and the quadratic $\|b^2\|$ (black) – again for the default resolution and a coarser resolution (solid and dashed lines, respectively). Here, the norms are defined as domain integrals over the physical domain. The method exactly conserves $\|b\|$ – effectively the total mass, an important global quantity. The (monotonic) decay of $\|b^2\|$ is the result of mixing: in the method this is the result of removing very thin filaments of buoyancy by ‘contour surgery’ at $1/16$ of the grid cell size in the conformal domain. Notably, the mixing is not strongly dependent on resolution. This is because mixing is especially vigorous in density-stratified flows, due to the

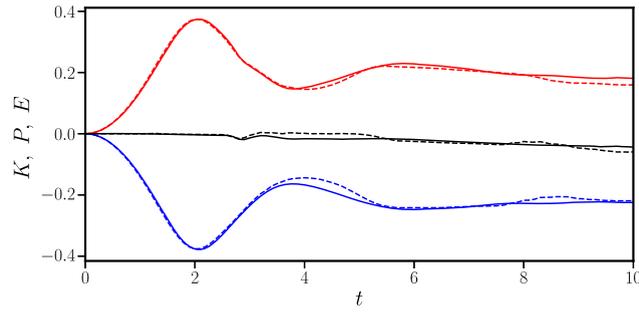


Fig. 11. Kinetic K (red), potential P (blue) and total energy $E = K + P$ (black) as a function of time for the dam break simulation (see text for details). Solid lines correspond to the default resolution (400×200), while dashed lines correspond to a coarser resolution (200×100).

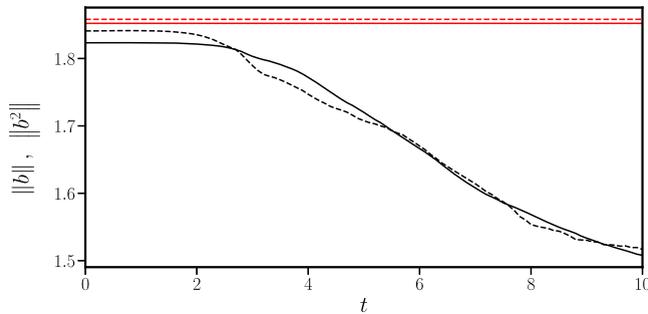


Fig. 12. Linear and quadratic norms of buoyancy: the linear norm $\|b\|$ is shown in red, while the quadratic norm $\|b^2\|$ is shown in black. Solid lines correspond to the default resolution (400×200), while dashed lines correspond to a coarser resolution (200×100).

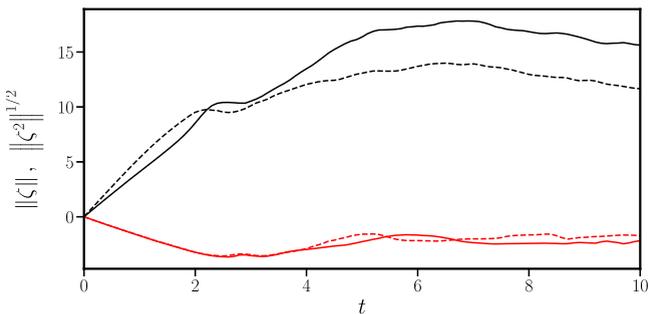


Fig. 13. Vorticity norms $\|\zeta\|$ (red) and $\sqrt{\|\zeta^2\|}$ (black). Solid lines correspond to the default resolution (400×200), while dashed lines correspond to a coarser resolution (200×100).

production of intense vorticity (see below). Note that mixing is slightly delayed at higher resolution (the strong downturn in the black solid occurs later in time), consistent with the greater time it takes for scales to cascade to the surgical scale.

Fig. 13 shows the evolution of the vorticity norms $\|\zeta\|$ (red) and $\sqrt{\|\zeta^2\|}$ (black). (The square root of $\|\zeta^2\|$ is used to be able to plot all results together.) Again results for the default resolution are compared with those obtained at a coarser resolution (solid and dashed lines, respectively). Note that $\|\zeta\|$ is the total circulation within the domain. This is captured well at both resolutions, and there is excellent agreement until $t = 4$. Even $\sqrt{\|\zeta^2\|}$ is not especially sensitive to resolution. As expected, this grows to larger values at higher resolution. Yet, although the default resolution simulation has half the grid spacing of the coarser resolution simulation – and thus the capacity to develop twice greater buoyancy gradients in (1) – the maximum r.m.s. vorticity is only approximately 1.27 times larger. It also slightly delayed in time, consistent with the longer time it takes for scales to reach the grid scale.

4. Conclusions and outlook

We have formulated a method for simulating nearly-inviscid, density-stratified (Boussinesq) flows in complex two-dimensional domains. This is done via conformal mapping, in which the physical domain is mapped into a rectangle. This

mapping has the advantage of largely preserving the structure of the equations, in particular Poisson's equation needed to recover the flow field from the vorticity. As a result, standard methods may be used to perform grid operations.

Here, we also make use of 'contour advection' following previous related work [3]. This is a highly accurate Lagrangian advection scheme that explicitly tracks contours or field isolevels (here of buoyancy and vorticity), as opposed to updating their values on a fixed grid (Eulerian). Contour advection, as a result, has no numerical stability constraints: the time step is chosen purely for accuracy.

We have applied the method to study the 'dam-break' problem, where two regions of different density (buoyancy) are initially placed side by side. We have studied this problem, and assessed the performance of the method, in a highly complex domain. The flow develops great complexity despite the modest grid resolutions used, due to the fact that contour advection faithfully resolves a wide range of subgrid-scale motions.

An extension of the method to moving domains has been presented. Numerical results confirm that the 'inversion' problem, namely that of determining the velocity field from the vorticity field in a moving domain, is accurately solved. What remains is to develop a full model to evolve a flow in moving domains. Here, we are limited by the exceedingly high cost of computing the conformal map. A solution would be to somehow exploit the fact that the map changes slowly from one time step to the next, but the SToolbox software package in MATLAB [4] we have used does not have this capability. Moreover, there appears to be no comparable package in FORTRAN or in Python that we could adapt. We hope this issue can be resolved in the near future to be able to take our approach further. Our original motivation was to model dynamics in a tidal estuary, where the fluid heights on either side of a weir generally vary in time. Another application of the method would be to free-surface flows, where the free-surface is mapped to a straight line.

In the present approach, we are limited to simple surfaces, free of breaking waves, bubbles and turbulence, and thus require sufficiently slow flow speeds. A two-dimensional model would otherwise not be accurate or even capable of representing the actual fluid motion. Regarding three dimensions, conformal mapping is no longer viable. An alternative using Lagrangian advection is to use the 'Elliptical Parcel-In-Cell' (EPIC) method [9,10]. Advection would be carried out in physical space, i.e. around obstacles in the domain. The only challenge is performing the velocity inversion problem in general domains, but this problem is common and there are many methods available.

More immediately, the static two-dimensional method developed in this work could readily be extended to model *rotating* stratified flows, thereby widening its applicability for instance in oceanography. For a domain rotating about the vertical (here Y) axis at rate $\Omega = f/2$ (where f is the Coriolis frequency), the transverse velocity component, say $W(X, Y, t)$, satisfies

$$\frac{DW}{Dt} + fU = 0 \quad \Rightarrow \quad \frac{DW_a}{Dt} = 0$$

where $W_a \equiv W + fX$. Hence, W_a is materially conserved. This follows since there are no transverse variations (in particular no pressure gradient). In the contour advection method developed and illustrated in the present study, it is natural to represent W_a by material contours, just like buoyancy b . This simple extension of the method would enable one to study a much broader range of research problems.

CRediT authorship contribution statement

Heidi J. Dritschel: Investigation, Methodology, Software, Visualization, Writing – original draft. **David G. Dritschel:** Conceptualization, Investigation, Methodology, Software, Supervision, Visualization, Writing – original draft. **Magda Carr:** Supervision, Writing – review & editing.

Declaration of competing interest

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

Heidi Dritschel reports financial support was provided by University of St Andrews. Heidi Dritschel reports financial support was provided by Newcastle University. David Dritschel reports a relationship with Engineering and Physical Sciences Research Council that includes: funding grants.

Data availability

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

Acknowledgements

This research received support from the UK Engineering and Physical Sciences Research Council (grants EP/R511778/1 and EP/R511584/1).

References

- [1] A. Ayouche, X. Carton, G. Charria, Vertical shear processes in river plumes: instabilities and turbulent mixing, *Symmetry* 14 (2022) 217.
- [2] P. Baines, *Topographic Effects in Stratified Flows*, 2nd edition, Cambridge University Press, 2022.
- [3] M. Carr, J. Franklin, S. King, P. Davies, J. Grue, D. Dritschel, The characteristics of billows generated by internal solitary waves, *J. Fluid Mech.* 812 (2017) 541–577.
- [4] T. Driscoll, L. Trefethen, *Schwarz-Christoffel Mapping*. Volume 8, Cambridge University Press, 2002.
- [5] D. Dritschel, M. Ambaum, A contour-advective semi-lagrangian algorithm for the simulation of fine-scale conservative fields, *Q. J. R. Meteorol. Soc.* 123 (1997) 1097–1130.
- [6] D. Dritschel, J. Fontane, The combined Lagrangian advection method, *J. Comput. Phys.* 229 (2010) 5408–5417.
- [7] D. Dritschel, S. Tobias, Two-dimensional magnetohydrodynamic turbulence in the small magnetic Prandtl number limit, *J. Fluid Mech.* 703 (2012) 85–98.
- [8] H. Fernando, Fluid dynamics of urban atmospheres in complex terrain, *Annu. Rev. Fluid Mech.* 42 (2010) 365–389.
- [9] M. Frey, D. Dritschel, S. Böing, EPIC: the elliptical parcel-in-cell method, *J. Comput. Phys. X* (2022) 100109.
- [10] M. Frey, D. Dritschel, S. Böing, The 3D elliptical parcel-in-cell (EPIC) method, *J. Comput. Phys.* (2023), Under review.
- [11] W. Geyer, P. MacCready, The estuarine circulation, *Annu. Rev. Fluid Mech.* 46 (2014) 175–197.
- [12] K. Helfrich, W. Melville, Long nonlinear internal waves, *Annu. Rev. Fluid Mech.* 38 (2006) 395–425.
- [13] I. Hewitt, Subglacial plumes, *Annu. Rev. Fluid Mech.* 52 (2020) 61–91.
- [14] S. King, M. Carr, D. Dritschel, The steady-state form of large-amplitude internal solitary waves, *J. Fluid Mech.* 666 (2011) 477–505.
- [15] L. Ostrovsky, Y. Stepanyants, Do internal solitons exist in the ocean?, *Rev. Geophys.* 27 (1989) 293–310.
- [16] P. Read, E. Pérez, I. Moroz, R. Young, T. von Larcher, P. Williams, General circulation of planetary atmospheres: insights from rotating annulus and related experiments, in: *Modeling Atmospheric and Oceanic Flows: Insights from Laboratory Experiments and Numerical Simulations*, 2014, 205.
- [17] E. Ryzhov, K.V. Koshel, M. Sokolovskiy, X. Carton, Interaction of an along-shore propagating vortex with a vortex enclosed in a circular bay, *Phys. Fluids* 30 (2018) 016602.
- [18] J.O. Shin, S.B. Dalziel, P.F. Linden, Gravity currents produced by lock exchange, *J. Fluid Mech.* 521 (2004) 1–34, <https://doi.org/10.1017/S002211200400165X>.
- [19] K. Trulsen, H. Zeng, O. Gramstad, Laboratory evidence of freak waves provoked by non-uniform bathymetry, *Phys. Fluids* 24 (2012) 097101.
- [20] J. Turner, *Buoyancy Effects in Fluids*. Cambridge Monographs on Mechanics, Cambridge University Press, 1973.